UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-2013-31
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 18/11/2013 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Naipeng Dong
Born on 7 September 1982 in Shandong (P. R. China)

Enforced Privacy: from Practice to Theory

Dissertation defense committee

Dr. Ir. Hugo Jonker (University of Luxembourg)
Prof. Dr. Pierre Kelsen (University of Luxembourg)
Dr. Pascal Lafourcade (Clermont Université, Université d'Auvergne, LIMOS)
Prof. Dr. Sjouke Mauw (University of Luxembourg)
Dr. Jun Pang (University of Luxembourg)
Dr. Michael Rusinowitch (LORIA–INRIA)

Fonds National de la Recherche Luxembourg

Summary

Privacy protection is an important requirement in both everyday life and the Internet. As the Internet is an open network, adversaries can observe and manipulate data flowing over it. To ensure privacy in communications over open networks, cryptographic protocols have been widely used, and thus, proposing such protocols has become a popular research area. However, design of cryptographic protocols is difficult and error-prone. Thus, verification of such protocols is a necessary step before implementation. Formal analysis has shown its strength in proving or disproving privacy properties of cryptographic protocols, contrary to informal verification of cryptographic protocols which is not suitable for finding subtle privacy flaws.

To formally verify whether a protocol satisfies a property, there are usually three steps: 1) formally model the protocol, 2) formalise the property and 3) decide whether the formalised property is satisfied on the formal model. Depending on the differences on the formalisms used to model the protocol and the property, there are various formal approaches. Once a formal approach is chosen, that is, the first and third steps are determined, one only needs to focus on the second step. In this thesis, we use a formalism called the applied pi calculus. The applied pi calculus provides an intuitive way to model cryptographic protocols. In addition, the applied pi calculus is equipped with proof techniques for privacy properties modelled as equivalences of processes. Furthermore, the verification of a protocol modelled in the applied pi calculus is supported by an automatic verification tool ProVerif.

Many privacy properties have been proposed, most of which are with respect to an adversary controlling the network. Recently, a stronger privacy property was identified in the e-voting domain. This privacy property assumes that the adversary can perform extra actions, namely bribing or coercing voters, to obtain additional information. To distinguish such strong privacy properties, we from here refer to these as *enforced privacy*, capturing the idea that the system enforces privacy, even if users try to reveal themselves due to bribery or coercion. Properties such as receipt-freeness and coercion-resistance, have been formalised in e-voting, to formally verify whether a protocol satisfies enforced privacy. The leading work of formalising enforced privacy in the applied pi calculus is the DKR framework, proposed by Delaune et al.

Following studies of enforced privacy in e-voting, protocols ensuring enforced privacy have also been proposed in other domains, e-auctions and e-health. However, such protocols have not been formally analysed. Thus, we first formally define enforced privacy in the e-auction and e-health domains. To validate the formal definitions, we formally verify enforced privacy in a case study in each domain.

First, we study the enforced privacy requirements in the e-auction and e-health domains. In both, enforced privacy is important. In the e-auction domain, there exists a type of auction, sealed-bid e-auction, where bidders seal their bids. In such auctions, it is required that a bidder should not be able to prove his bid to a coercer, so that malicious bidders cannot win with an unreasonable low price by controlling other bidder's bidding prices. In the e-health domain, it is required that doctors should not be able to prove prescriptions of their patients to pharmaceutical companies, to prevent pharmaceutical companies bribing or coercing doctors to favour their medicine.

Second, we formally define enforced privacy properties in e-auctions and e-health. To do so, we first study whether we could apply the formalisations in voting to e-auctions and e-health. We found that we cannot directly adopt these existing formalisations. The first reason is that classical privacy, on which enforced privacy is built on, differs from domain to domain. In e-voting, the voting result is published. When the result is unanimous, the adversary knows every voter's vote. Consequently, DKR defines classical privacy property in voting as: The adversary cannot detect any difference when two voters swap their votes. In effect, the definition uses a counter-balancing vote. In sealed-bid e-auction, a non-winning bidder's bid is not revealed. Thus, we do not need a counter-balancing bid. Classical privacy in e-auction can be defined, in a classical way, as: The adversary cannot detect any difference when the bidder changes his bid. However, we do need a higher bid, so that the target bid does not win. The second reason that we cannot directly adopt the existing formalisations in e-voting to e-auctions and e-health, is that the amount of information the adversary bribes or coerces differs. In e-voting, each voter normally only votes once. Similarly, in sealed-bid e-auctions, each bidder bids only once. However, in e-health, a doctor uses the same protocol multiple times to prescribe medicine for patients. Classical privacy in e-health requires that the adversary cannot detect any difference when two doctors swap their prescriptions. This requirements focus on a specific session of a doctor. The corresponding enforced privacy focuses on the specific session as well. That is, not all the doctor's information is revealed to the adversary. Therefore, we need to model doctors only partially communicating with the adversary, contrasting with in e-voting and e-auctions, voters/bidders fully communicating with the adversary.

Third, besides enforced privacy, we identify a new privacy property, independency of privacy in e-health. Unlike e-voting and e-auction systems, where roles can be naturally divided into two types: voters/bidders and authorities, e-health systems involve much more roles: doctors, patients, pharmacists, medical administrations, insurance companies. Some of these roles have access to sensitive data, for example, pharmacists have access to prescriptions. However, these roles may not be trustworthy, for example, the pharmacists may be bribed to reveal the link between a doctor and his prescriptions. To prevent this, it requires the pharmacist be independent of a doctor's privacy. To capture this requirement, we formally define the property *independency of prescribing-privacy*. This property has also been verified in a case study.

Based on the experience of domain-specific formalisation of enforced privacy, we propose a formal framework, in which enforced privacy properties can be defined in a domain-independent manner. Inspired by the demand that only part of the user process communicates with the adversary, we extend formalisation of bribery and coercion in e-voting, to allow us to specify the shared information with the adversary. This captures a variety of collaborations between users and the adversary. Each enforced privacy property and independency privacy property is parametrised with a collaboration specification. Specific enforced privacy and independency of privacy properties can be expressed by instantiating the parameter.

Inspired by the requirement in e-health that a pharmacist should not be able to help break a doctor's privacy, we notice that third parties may influence a target user's privacy. Here, pharmacists, cooperate with the adversary to *break* a target user's privacy. In such cases, we say that third parties have negative influence on the target user's privacy. On the other hand, third parties may help *strengthen* the target user's privacy, by cooperating with the target user. We say that third parties have positive influence. For instance, a voter can swap receipts with a coerced voter, if the voter votes as the coercer wants, so that the coerced voter can vote freely. We say that the third parties and the target user form a coalition. A coalition can be specified in a similar way as the collaboration. The difference is that in collaboration, we do not need to specify the behaviour of the adversary because the adversary is modelled as the environment; in coalition, both the third parties and the target user's behaviour need to be specified. We can formally define a coalition using a coalition specification. Correspondingly, we formalise privacy properties parametrised with coalitions. As an additional benefit, we can use coalitions to model various domain-specific classical privacy properties. For instance, vote-privacy, where a counter-balancing voter is required, can be considered as a coalition privacy with a coalition of the target voter and a counter-balancing voter. Finally, we prove the relations between the privacy properties defined in the formal framework.

Acknowledgements

I wish to thank many people for their support in my four years at the University of Luxembourg.

First of all, I would like to express my deepest gratitude to Dr. Jun Pang, Dr. Hugo Jonker and Prof. Dr. Sjouke Mauw, my research supervisors, for their guidance, enthusiastic encouragement and useful critiques of this work.

Dr. Jun Pang introduced me to this research field. I learned a lot form him – how to do research, how to write papers and how to give presentations, amongst other things. His passion and attitude on research influenced me and encouraged me to continue research.

Dr. Hugo Jonker helped me a lot on various aspects, research ideas, language, how to improve documents, and so on. His willingness to give his time and insight have been very much appreciated. No matter how busy he is, “there is always time for Naipeng”.

Without their great effort, there would not be this thesis. They tried different ways to teach me and took effect to find the way suits me the best. They did what they can to save my time. Even in the hard times, they still believe in me and encourage me. I enjoyed the time working with them, chatting with them, sharing food and nice music.

I also own my deepest gratitude to Prof. Dr. Sjouke Mauw. Without him, I would not have the opportunity to have done this work. His advice is straightforward and directly to the point. Advice given by him has been a great help in both research and daily life. He is a humorous and nice group leader who creates very comfortable atmosphere in the group, even to the point where he complained “I want my PhD student to take me seriously”.

I also would like to thank Prof. Dr. Pierre Kelsen for joining my CET committee, with his valuable and constructive suggestions during the planning and development of this work.

I wish to express my appreciation to my defence committee, Dr. Hugo Jonker, Prof. Dr. Pierre Kelsen, Dr. Pascal Lafourcade, Prof. Dr. Sjouke Mauw, Dr. Jun Pang, and Dr. Michael Rusinowitch for their effort in reading the thesis, and taking the time to provide me with valuable feedback.

To my colleagues, Baptiste Alcalde, Xihui Chen, Ton van Deursen, Wojtek Jamroga, Barbara Kordy, Piotr Kordy, Simon Kramer, Matthijs Melissen, Andrzej Mizera, Tim Muller, Georgios Pitsilis, Saša Radomirović, Patrick Schweitzer, Rolando Trujillo Rasúa, Chenyi Zhang and Yang Zhang, I also owe gratitude. I benefited a lot from seminars given by them, comments from them

and discussions with them. I enjoyed the group lunches and group activities.

Special thanks should be given to my family for their devoted love and my friends for their support and encouragement throughout my study.

Finally, I wish to thank to my boyfriend Tim Muller for his love and his family for their hospitality. I had great weekends and holidays in the Netherlands.

Naipeng Dong

Contents

1 Introduction **1**

1.1 Privacy and enforced privacy 1

1.2 Formal approach 4

1.3 Research questions 5

1.4 Thesis overview 8

2 Related work **11**

2.1 Privacy properties of cryptographic protocols 11

2.1.1 Classical privacy properties 12

2.1.2 Enforced privacy properties 14

2.2 Formalisations of privacy properties 15

3 Preliminaries **19**

3.1 The applied pi calculus 19

3.1.1 Syntax 19

3.1.2 Operational semantics 22

3.1.3 Equivalences 22

3.2 ProVerif 24

3.3 The DKR framework 26

4 Enforced privacy in e-auctions **29**

4.1 Introduction to e-auctions 29

4.2 Privacy and enforced privacy in e-auctions 30

4.3 Formalisation of privacy notions 31

4.3.1 Bidding-price-secrecy 31

4.3.2 Receipt-freeness 32

4.4 Case study: the AS02 protocol 33

4.4.1 Introduction 33

4.4.2 Physical assumptions 33

4.4.3 Settings 33

4.4.4 Description of the protocol 33

4.4.5 Claimed privacy properties 35

4.5 Modelling AS02 35

4.6 Analysis of AS02 41

4.6.1 Bidding-price-secrecy 42

4.6.2 Receipt-freeness 43

4.7 Conclusions 46

5 Enforced privacy in e-health 49

5.1 Introduction to e-health 49

5.2 Privacy and enforced privacy in e-health 50

5.2.1 Patient privacy 51

5.2.2 Doctor privacy 52

5.3 Formalisation of privacy notions 53

5.3.1 Prescribing-privacy 54

5.3.2 Enforced prescribing-privacy 55

5.3.3 Independency of prescribing-privacy 56

5.3.4 Independency of enforced prescribing-privacy 56

5.3.5 Anonymity and strong anonymity 57

5.3.6 Untraceability and strong untraceability 57

5.4 Case study: the DLV08 protocol 58

5.4.1 Introduction 58

5.4.2 Cryptographic primitives 58

5.4.3 Settings 59

5.4.4 Description of the protocol 60

5.4.5 Claimed privacy properties 63

5.5 Modelling DLV08 64

5.5.1 Underspecification of the DLV08 protocol 64

5.5.2 Modelling cryptographic primitives 65

5.5.3 Modelling the DLV08 protocol 68

5.6 Analysis of DLV08 78

5.6.1 Secrecy of patient and doctor information 79

5.6.2 Patient and doctor authentication 80

5.6.3 (Strong) patient and doctor anonymity 81

5.6.4 (Strong) patient and doctor untraceability 82

5.6.5 Prescribing-privacy 84

5.6.6 Enforced prescribing-privacy 84

5.6.7 Independency of (enforced) prescribing-privacy 87

5.7 Addressing the flaws of the DLV08 protocol 88

5.8 Conclusions 92

6 Enforced privacy in the presence of others 95

6.1 Privacy notions 95

6.2 Formal framework 97

6.2.1 Well-formed protocols 97

6.2.2 Data-privacy 98

6.2.3 Modelling collaboration with the adversary 100

6.2.4 Modelling user coalitions 102

6.3 Formalising the privacy notions 104

6.3.1 Enforced-privacy 104

6.3.2 Independency-of-privacy 105

6.3.3 Independency-of-enforced-privacy 106

6.3.4 Coalition privacy properties 107

6.3.5 An example of coalition privacy 109

6.4 Relations between the privacy notions 111

6.5 Application 113

6.6 Conclusions 114

7 Conclusions and future work 117

7.1 Summary of contributions 118

7.2 Future work 119

Appendix 120

A Full proofs in Chapter 4 121

A.1 Full proof of receipt-freeness of AS02 121

B Full proofs in Chapter 6 127

B.1 Auxiliary lemmas 127

B.2 Theorem 6.20 129

B.3 Theorem. 6.21 139

B.4 Theorem. 6.22 147

Publications 153

Bibliography **161**

Index of subjects **165**

List of Symbols **169**

List of Figures

2.1 Relations between anonymity and unobservability. 13

2.2 Relations of classical privacy properties. 14

3.1 Terms in the applied pi calculus. 19

3.2 Processes in the applied pi calculus. 20

3.3 Structural equivalence in the applied pi calculus. 22

3.4 Internal reduction in the applied pi calculus. 22

3.5 Labelled reduction in the applied pi calculus. 23

4.1 MSC: AS02 protocol 34

4.2 Functions in AS02. 37

4.3 Equational theory in AS02. 37

4.4 The main process. 38

4.5 The key distribution process. 38

4.6 The bidder process. 39

4.7 The auctioneer process. 40

4.8 The process *readinfo*. 41

4.9 The process $\mathit{checknextb}^{\mathsf{p}_j}_{\mathsf{b}_i}$. 41

4.10 The process $\mathit{checknextbnp}^{\mathsf{p}_j}_{\mathsf{b}_i}$. 42

4.11 The context $\mathcal{C}_{AS02}[\_]$. 42

4.12 The process P_f. 44

4.13 A sketch proof of receipt-freeness in AS02. 45

5.1 MSC: Doctor-Patient sub-protocol. 61

5.2 MSC: Patient-Pharmacist sub-protocol. 62

5.3 MSC: Pharmacist-MPA sub-protocol. 63

5.4 MSC: MPA-HII sub-protocol. 64

5.5 Functions in DLV08. 68

5.6 Equational theory in DLV08 part I: non-zero-knowledge part. 68

5.7 Equational theory in DLV08 part II: zero-knowledge part. 69

5.8 The doctor process P_{dr}. 70

5.9 The patient process part I: $P_{pt}\_p_1$. 71

5.10 The patient process part II: $P_{pt}\_p_2$. 72

5.11 The pharmacist process part I: $P_{ph}\_p_1$. 73

5.12 The MPA process part I: $P_{mpa}\_p_1$. 74

5.13 The pharmacist process part II: $P_{ph}\_p_2$. 75

5.14 The MPA process part II: $P_{mpa}\_p_2$. 75

5.15 The HII process P_{hii}. 76

5.16 Overview of the DLV08 protocol. 77

5.17 The process for role patient R_{pt}. 77

5.18 The process for role pharmacist R_{ph}. 78

5.19 The process for role MPA R_{mpa}. 78

5.20 The process for the DLV08 protocol. 78

5.21 The process for role doctor R_{dr}. 78

5.22 The process for role HII R_{hii}. 79

5.23 The doctor process P'_{dr} (using untappable channels). 89

6.1 Relations of the privacy notions 111

A.1 The process $\nu\mathtt{chc}.(P_f \mid !\mathsf{in}(\mathtt{chc}, x))$. 121

A.2 Transitions of P' 122

A.3 The bidder process $P_b\{\mathtt{c}/p_b\}$. 122

A.4 Transitions of Q'. 123

List of Tables

2.1 Definitions of classical privacy properties. 14

3.1 Different notations in ProVerif and applied pi. 26

5.1 Verification results of secrecy for patients and doctors. 80
5.2 Verification results of authentication of patients and doctors. 81
5.3 Verification results of privacy properties and revised assumptions. . 88

6.1 Privacy notions 96

1

Introduction

In the physical world, privacy is assured by means such as locks and curtains. Communication over the Internet introduces new privacy risks. For example, a malicious user may spy on another user's e-mails. As the usage of Internet-based services continues to grow, privacy on the Internet is becoming more and more important. This has led to a proliferation of privacy notions, including notions focused on communication privacy and those focused on data-bases, etc. Recently, a strong privacy notion has been proposed for e-voting. This new notion includes the assumption that adversaries can bribe or coerce users. This thesis studies the privacy effect of such an assumption in other domains and proposes a domain-independent notion, using a formal approach.

In the introduction, we discuss the privacy issues and privacy requirements in Internet-based services in Section 1.1. The approach that we use to address the privacy issues is introduced in Section 1.2. Next, detailed research challenges are presented in Section 1.3. Finally, we describe the organisation of the thesis in Section 1.4.

1.1 Privacy and enforced privacy

Privacy has been considered as an important requirement in daily life. For instance, curtains are used to keep people's private life beyond the general public's observation. In some cases, privacy is even required by law. For example, the U.S. HIPAA Privacy Rule (The Health Insurance Portability and Accountability Act of 1996) states that a patient's medical records are required to be confidential in health care. Even in cases where privacy is not a legal requirement, privacy may remain a desirable and marketable property. For instance, people may want to publish an article anonymously to avoid problems, if the article is against a group of people's benefits. Consequently, in the analogous Internet-based activities, privacy protection is required as well. In addition, in order to attract users, providers for Internet-based services where privacy is optional may offer privacy as a distinguishing feature. For instance, privacy protection is an aimed feature in the Internet search engine *DuckDuckGo*.

However, there have been more and more high-profile privacy incidents in Internet-based services, such as, private messages sent between users and Bloomberg's financial terminals having leaked [6], a user's activities being viewed by people who should not be able to do so in facebook [4]. These incidents cause more and more potential privacy breaking worries, for example, when a user posts a photo on facebook, additional data, such as the time, date, and place are recorded, the user can be traced using the collected data by facebook [2]. In order to register a service, a user often needs to provide personal information to the service provider. Normally the requested information is used for the purpose of authenticating users or providing better services. However, this provides the service providers an opportunity to know more than the user expected, from the collected data. For example,

- Google can track search terms via cookies and IP addresses, thus can build a profile of you based on your gmail account and your search queries, for example, who you are, where you go and what you do.

  > "We are moving to a Google that knows more about you."
  >
  > – Google CEO Eric Schmidt, 10 February 2005 [3]

- Mobile phone companies can log users locations and associations between users' credit card companies and other financial organisations [7].
- Internet service providers can log all transactions, monitor email, web accesses, etc. [5]

Even if the service providers are all trustworthy on not revealing privacy, sending information over the Internet may also leak information due to eavesdroppers on the Internet. If the transmitted information is not well protected, an observer on the Internet can obtain private information of users. For example, by observing the location attached to a service query "what is the nearest coffee shop", the observer knows the location of the user.

All these privacy concerns lead to research on how to develop systems satisfying a desired privacy property. Various techniques have been applied to ensure privacy. For instance,

- Laws and regulations exist to ensure privacy on an administrative level. For example the European Union's data protection directive (95/46/EC) regulates the processing of personal data.
- Access control exists to ensure privacy in the sense that only authorised roles can access protected data. Such an example is role-based access control [RCHS03].
- Cryptographic protocols have been employed to ensure privacy on communications over open networks (such as, the Internet). For example protocols using zero-knowledge proofs, allow an individual to prove a statement without revealing his secrets.

We observe that privacy controls focus on different interests. A breach of privacy on any domain affects privacy in general. Hence, ensuring privacy requires that privacy is assured on each individual domain.

For each domain, there are specific approaches towards ensuring privacy. And in general, an approach used to ensure privacy in one domain does not translate to another. For example, zero-knowledge proofs cannot be applied on laws. In addition, for each domain, there are specific privacy requirements. For instance, in cryptographic protocols:

- Adversaries, observing and manipulating messages over open networks, need to be considered. The use of computers and networks, in addition to facilitate honest users, provides adversaries stronger computing ability and the ability to control the networks.
- Privacy of encryption keys, which may not even be mentioned in some access controls (such as laws), may be critical in cryptographic protocols.
- Moreover, data can be easily spread over networks. And once the data is revealed, it is hardly possible to destroy it, as it may has been stored multiple times by many users. Thus, privacy breach, such as revealing private data, in the digital world may be more harmful than in the real world.

Therefore, domain specific expertise is required to design or evaluate specific privacy controls. In this thesis, we focus on privacy on the level of cryptographic protocols.

To capture privacy requirements in cryptographic protocols, privacy properties are proposed. In order to define a privacy property precisely, the adversary abilities need to be defined first. The best known adversary model is the Dolev-Yao adversary, who controls the whole network (observing, blocking, modifying, injecting messages on the network). Depending on protected items in various domains, many privacy properties have been defined, e.g., anonymity, untraceability, unlinkability, etc. A common feature for such privacy properties is that they assume users want to keep their information private. In other words, a system satisfies privacy under the assumption that users honestly follow the system execution.

However, this assumption has been shown not suitable in some cases. For instance, in e-voting, a voter may want to sell his vote to a vote-buyer. As vote-selling harms the e-voting system, e-voting systems require that a voter should not be able to prove his vote to others. In addition, a coercer may be able to force (using a gun or a threat) a voter to vote a certain candidate or vote in a certain way. Hence, e-voting systems also require coercion-resistance. Similar requirements have also been identified in other domains. For instance, some e-auction systems require that the adversary should not be able to coerce a non-winning bidder to show his bid. In e-health, a doctor should not be able to prove his prescriptions to a pharmaceutical company even with the help of pharmacists.

To capture these requirements, privacy properties have been proposed, for instance, receipt-freeness in e-voting and e-auctions, coercion-resistance in e-voting.

- Receipt-freeness in e-voting: a voter should not have a receipt to prove his vote to any vote-buyer;
- Receipt-freeness in sealed-bid e-auctions: a non-winning bidder should not be able to prove his bid to other bidders;
- Coercion-resistance in e-voting: a coercer should not be able to coerce a voter to vote in a certain way.

These properties all follow the same idea: If a system is designed in such a way that a user can lie to the adversary about the target information, a vote or a bid, and the adversary cannot tell whether the user has lied, then the system satisfies this type of privacy.

We call this type of privacy, where users are assumed to reveal information to the adversary (by bribery/coercion or other methods), *enforced privacy*, meaning that the system enforces privacy upon its users instead of users desiring privacy. Enforced privacy properties have recently been identified. Compared to the classical privacy properties where users do not cooperate with the adversary, research on enforced privacy properties is less mature. This thesis focuses on enforced privacy.

1.2 Formal approach

As stated in the previous section, information sharing between users, which are distributed over open networks, follows communication protocols. Communication among users is open to interference by adversaries controlling the network. To ensure security and privacy in communications over open networks, cryptography is widely used in protocols. Hence, proposing cryptographic protocols has become a popular research area.

However, equipping protocols with cryptography is not a guarantee for security or privacy. In fact, the design of cryptographic protocols is notoriously difficult and error-prone. For instance, the Needham-Schroeder protocol [NS78] was found to allow the adversary to reuse an old and compromised session key [DS81]. An RFID protocol [HM04] with an untraceability claim was found flawed [Avo05].

At the beginning, a claimed property of a protocol is empirically verified. Researchers study a protocol in detail and decide whether the protocol satisfies a claimed property. As flaws in cryptographic protocols are often subtle and counter-intuitive, it is easy to make undetected mistakes in such informal verification. Thus, informal analysis is too prone to error to reliably verify cryptographic protocols, especially when the protocols are getting more and more complex. Protocols tend to involve increasing utilisation of cryptographic primitives and increasing concurrent distributed programs which can be executed by a large population of agents.

As informal reasoning was getting less reliable and efficient, people turned to formal approaches, a mathematically based technique, to help detect flaws. The use of formal approaches was initiated by Dolev and Yao to analyse secrecy properties of protocols. The main ideas are that

1. the adversary controls the whole network; and
2. encryptions are perfect, meaning that the adversary cannot undo an encryption and find the plain text.

Since the adversary controls the whole network, and communicated messages all pass through the network, the adversary can be modelled as a buffer between users. The messages from an honest user are considered to be sent to the adversary and the messages received by an honest user are considered to be sent by the adversary. In addition, the adversary can block, redirect and alter messages, and generate fresh messages. Security properties and privacy properties of a protocol are modelled as properties of such a model of the protocol.

Formal verification has been successfully used in proving or disproving the satisfaction of a claimed property. In several cases, formal verification found security or privacy flaws in protocols which were thought to be secure or privacy-preserving, e.g., see [Low96, CKS04, DKR09]. Therefore, before using a cryptographic protocol, formal verification is an important step.

To formally verify whether a protocol satisfies a property, there are usually three steps:

1. formally model the protocol,
2. formalise the property, and
3. decide whether the formalised property is satisfied on the formal model.

Depending on the differences on the formalisms used to model the protocol and the property, and verification algorithms used to decide whether a property is satisfied on a formal model, there are various formal approaches. In general, in the literature, the study of specification languages for modelling a protocol and its properties, and verification algorithms is rather mature. In contrast to this, the specification of privacy properties is still in its infancy. Formalisation of privacy properties is not as standard as other properties, like security. New privacy notions arise from time to time. In addition, privacy requirements differ from domain to domain. Thus, we use an existing formalism, the applied pi calculus, and focus on formalising privacy properties. The applied pi calculus provides an intuitive way to model cryptographic protocols. In addition, the applied pi calculus is equipped with proof techniques for privacy properties modelled as equivalences of processes. Furthermore, the verification of a protocol modelled in the applied pi calculus is supported by an automatic verification tool ProVerif. Moreover, a leading work in formalising enforced privacy properties in e-voting, receipt-freeness and coercion-resistance, uses the applied pi calculus. Therefore, in this thesis, we focus on studying and formalising enforced privacy using the applied pi calculus, learning from the formalisations of enforced privacy in e-voting.

1.3 Research questions

Current research on enforced privacy focuses on the e-voting domain. However, bribery and coercion, in general, are domain independent. Thus, the adversary can bribe or coerce users in any domain. Therefore, a natural question to ask is: Are privacy requirements against bribery and coercion relevant in other domains? If the answer is yes, then how do the requirements in other domains differ from those in e-voting?

We answer these questions by studying enforced privacy requirements in e-voting, e-auctions and e-health. We observe that enforced privacy is indeed required in other domains. In e-auctions, it requires that a non-winning bidder should not be able to prove his bid to a coercer; in e-health, it requires that a doctor should not be able to prove his prescriptions to a pharmaceutical company.

Enforced privacy requirements differ in domains. Despite the difference in the protected items, privacy requirements may also differ in the way a user cooperates with the adversary. For example, in e-voting, the adversary may instruct a voter to vote for a certain candidate, to vote the same as another voter (copy another's vote), to abstain etc. In general, theses instructions may also be given by the adversary in e-auctions. However, in some sealed-bid auctions where each bidder is required to submit a bid, the adversary, in order to achieve the goal of controlling the price, would not give the abstaining instruction to bidders. In e-health, abstention is not a realistic instruction as well, since it conflicts the adversary's goal. In addition, privacy notions may differ for the various kinds of users. For example, in e-voting, enforced privacy needs to be satisfied for all voters; in some sealed bid e-auctions, enforced privacy only needs to be satisfied for non-winning bidders. Finally, privacy requirements may differ in the influence of third parties. In e-voting and e-auctions, in general, two roles are involved and one of them is assumed to be honest. In e-health, many roles are involved, typically including: doctors, patients, pharmacists, insurance companies, etc. Some of these roles are in general not trustworthy, e.g., pharmacists. Thus, it is required that pharmacists should not be able to help prove a doctor's prescriptions.

Due to these differences in domains, formalisations of privacy properties in e-voting, in general, cannot be directly adopted in other domains; for example, new privacy properties taking pharmacists into consideration need to be formalised in e-health.

Since enforced privacy is required and the formalisation of privacy properties is a necessary and important step in deciding whether a protocol satisfies such a privacy requirement, a research question arises:

Research question 1: How should enforced privacy be formalised in the e-auction and e-health domain?

Enforced privacy notions, e.g., receipt-freeness and coercion-resistance, have been formalised in e-voting. The first piece of work is the formalisation of receipt-freeness using the applied pi calculus. Later, following the direction of the applied pi calculus, a general formal framework for enforced privacy in e-voting has been developed – the DKR formal framework. Other formal frameworks, using other formal approaches or focusing on variations of enforced privacy properties, have also been developed in e-voting.

In e-auctions, enforced privacy properties have been discussed and protocols preserving them have been proposed. However, these properties are not formalised, protocols are not formally verified. Similarly, enforced privacy requirements can be found in the e-health domain, but no formalisation has been developed. Thus, a research task is to formally define enforced privacy in such domains. To do so, we first want to see whether the existing formalisations in e-voting can be reused. Among the few formal frameworks, we choose to follow the DKR formal framework, due to the convenience of the applied pi calculus.

We observe that although the formal definitions of enforced privacy in e-voting cannot be adopted due to the variety in classical privacy properties on which enforced privacy is built, the formalisation of bribery and coercion can be reused in general. Following the DKR style of defining the cooperation between bribed or coerced users and the adversary, we formalise enforced privacy in e-auction and e-health protocols in the applied pi calculus. This formalisation has been applied to two case studies – a sealed-bid e-auction protocol proposed by Abe and Suzuki

(AS02) and an e-health protocol proposed by de Decker et al. (DLV08).

It is not convenient to formalise enforced privacy in every domain where it is required. And the formalisations in the three domains follow the same idea and only differ in minor details. These lead us to consider the generalisation of formal definitions for enforced privacy.

Research question 2: How can we develop a generic formal framework in which we are able to define various enforced privacy notions?

We learn from the DKR framework in e-voting and are inspired by a formal framework for defining generic classical privacy in the applied pi calculus. With the help of the knowledge and experience of formalising enforced privacy in e-auctions and e-health, we propose a formal framework. This framework presents a standard form of protocols which enables us to formalise privacy in a generic manner. We notice that the formalisation of bribery and coercion in e-voting, e-auctions and e-health are instances of various types of cooperation between users and the adversary and cooperation is not limited to bribery and coercion. To be able to formalise more types of cooperation, a generic way of modelling cooperating users is proposed. By allowing a precise specification of which information has been shared with the adversary, the formalisation of bribery, coercion and many other types of cooperation can be instantiated.

In addition, in this formal framework, we formally define a classical privacy notion – data privacy. This formal definition is domain-independent and covers many privacy requirements. It serves as the foundational privacy notion on top of which enforced privacy notions can be built. Using the formalisation of coercion, we are able to formalise various enforced privacy properties.

However, since data privacy does not cover all classical privacy formalisations, e.g., vote-privacy cannot be instantiated as data privacy, enforced privacy notions based on data privacy inherently have the same limitation, e.g., receipt-freeness defined in DKR framework cannot be instantiated. We observe that those classical privacy formalisations which cannot be instantiated as data privacy, normally require certain behaviour of users who are not the target user, e.g., vote-privacy requires a voter besides a target voter to counter balance the target voter's vote, bid-privacy requires a bidder who bids higher than the target bidder. We consider that in these privacy formalisations, the target user needs third parties' help. On the other hand, third parties helping to break a target user's privacy has been formalised in e-health. A natural research direction is to formalise privacy and enforced privacy which takes third parties' influence into account.

Research question 3: How do we formally define privacy notions taking third parties' influence into account in our framework?

We notice that the influence of a third party can be positive and negative. The positive influence includes both active help and passive help from third parties. In some cases, although third parties do not actively help, they do assist to create an environment in which privacy may be satisfied. The pharmacists in e-health can be considered as instances of such negative third parties.

The negative influence of third parties is modelled as the third parties cooperating with the adversary. Using the formalisation of cooperation between users and the adversary mentioned in the previous question, we are able to formalise the privacy notions taking third parties' negative influence into account. The positive influence of third parties is modelled as a coalition of a target user and the third parties. A coalition is formalised as a transformation of the original target user and third party processes in the formal framework. Using the formalisation of coalition, we are able to formalise the privacy notions taking third parties' positive influence into account. Finally, since many privacy notions have been formalised, we provide a hierarchy of these privacy notions and prove the relations between them.

1.4 Thesis overview

The first research question is answered in Chapter 4 and Chapter 5. And the last two research questions are answered in Chapter 6. The remaining part of the thesis is organised as follows.

Chapter 2: Related work In this chapter, we discuss the adversary model, privacy notions and enforced privacy notions in the literature.

Chapter 3: Preliminaries In this chapter, we focus on the applied pi calculus. We motivate the use of the applied pi calculus, briefly introduce the language, and present how enforced privacy in e-voting is formalised in the applied pi calculus.

Chapter 4: Enforced privacy in e-auctions This chapter focuses on enforced privacy in the e-auction domain. We first briefly introduce e-auctions. Next, privacy and enforced privacy requirements in e-auction systems are discussed. Then, we formalise a classical privacy notion – bid-privacy for non-winning bidders and an enforced privacy notion – receipt-freeness for non-winning bidders in the applied pi calculus. Finally, a case study – the AS02 protocol is introduced, modelled and analysed.
The main contribution of this chapter is the formalisation of enforced privacy in e-auctions and the formal analysis of the AS02 e-auction protocol.
This chapter is based on work with Hugo Jonker and Jun Pang, published in [FAST10].

Chapter 5: Enforced privacy in e-health This chapter focuses on enforced privacy in the e-health domain. We briefly discuss privacy issues in e-health. Differing from e-auctions and e-voting, where enforced privacy is well identified, there is only few work on enforced privacy in e-health. Thus, enforced privacy in this domain remains a challenge. We contribute to this challenge in formally defining privacy notions which capture the enforced privacy requirements. Enforced privacy for doctor's prescription behaviour is formalised. In addition, we formally define the privacy notions which capture the requirement to enforce a third party (pharmacist) to respect a target user's privacy or enforced privacy (doctor prescribing privacy). Finally, a case study is performed to validate the formalisations of privacy notions. This case study includes a description of the DLV08 e-health protocol, the formal modelling of the protocol and the formal analysis. Few privacy flaws have been identified due to the ambiguous assumptions of the protocol and suggestions to address these flaws are proposed.

The main contribution of this chapter is as follows: 1) We formalised enforced privacy in e-health; 2) identified and formalised independency of (enforced) privacy in e-health; and 3) formally analysed the DLV08 e-health protocol.

This chapter is based on two published papers [FHIES11, ESORICS12].

Chapter 6: Enforced privacy in the presence of others In this chapter, we generalise the domain specific enforced privacy. In addition, we take third parties' influence on a target user's privacy into account. Two types of influences are considered: positive (helping to maintain privacy) and negative (helping to break privacy). In order to formally define the above privacy in a domain-independent manner, we propose a formal framework. In the framework, a standard form of protocols is defined using the applied pi calculus. Based on the standard form, a classical privacy notion – data privacy, is formally defined which is used as the foundation of enforced privacy notions. In addition, a formalisation of a variety of ways for cooperation between users and the adversary is proposed. Using this formalisation, enforced privacy is defined based on data privacy. The behaviour of third parties is formalised in the formal framework as well: the negative behaviour is modelled as the cooperation between the third parties and the adversary; the positive behaviour is modelled as a coalition between the third parties and the target user. Using this formalisation, we formalise privacy notions with third party influences in the framework. Finally, a hierarchy of enforced privacy properties is built to show the relations between these notions. The relations are proved in the appendix.

The main contribution of this chapter is as follows: 1) We formalised collaboration between users and the adversary and the collaboration between users, in a formal framework. 2) We identified coalition privacy and formalised them in the formal framework, as well as other (enforced) privacy properties. 3) Finally, we proved the relations between the formalised privacy properties.

This chapter is based on a paper with Hugo Jonker and Jun Pang published in [ESORICS13].

Chapter 7: Conclusions and future remarks Finally, the thesis is concluded in Chapter 7.

2

Related work

This chapter is organised as follows. Section 2.1 introduces privacy properties of cryptographic protocols previously proposed in the literature, and Section 2.2 discusses existing formalisations of these privacy properties.

2.1 Privacy properties of cryptographic protocols

In general, privacy hides the correspondence between users and the users' items (such as identities and votes) from the adversary. To define a privacy property, first of all, we need to define the adversary ability. As 1) our main focus is privacy of cryptographic protocols and 2) in such protocols, the use of the network is essential, the adversary from which privacy is protected is from the network. Many such adversary models exist, for instance, passive adversaries which only observe the network, and active adversaries which also actively communicate with the participants. Among those adversary models, the best-known is the Dolev-Yao adversary.

Dolev-Yao adversary [DY83]. The adversary has the following abilities:

- Controlling the network: The adversary is able to eavesdrop, block and inject messages on the network.
- Computational ability: The adversary can extract data from messages and compose new messages from known data.
- Initiating data and sessions: The adversary can generate fresh data as needed and can initiate a conversation with any user.
- Initial knowledge: The adversary has a set of initial knowledge containing public information, such as public keys.

This adversary is considered as the strongest adversary controlling the whole network [Cer01]. When a system satisfies privacy with respect to the Dolev-Yao adversary, it also satisfies privacy with respect to weaker adversaries. Thus, in this thesis, we only consider privacy properties with respect to the Dolev-Yao adversary, that is, the variations of privacy properties due to weaker adversary models are neglected.

We show privacy properties of cryptographic protocols in the rest of this section. These properties are classified into two categories by whether the assumption that the adversary bribes or coerces users is made. Privacy properties without this assumption, named *classical* privacy properties, are introduced in Section 2.1.1. Privacy properties with the assumption, named *enforced* privacy properties, are presented in Section 2.1.2.

2.1.1 Classical privacy properties

Classical privacy has been studied for long time [Cha85]. Classical privacy properties have been proposed for systems in various domains, such as, e-mail, digital cash, e-voting, e-auction, RFID (radio-frequency identification), location based services, etc. These properties can be distinguished by the protected items. Protected items naturally vary in domains. Even in the same domain, protected items can be different. For example, in e-mail systems, three privacy properties are distinguished:

- Sender anonymity: the adversary cannot identify the sender of a given message [SD02].
- Recipient anonymity: the adversary cannot identify the recipient of a given message [SD02].
- Relation anonymity: the adversary cannot link a sender to a receiver [PK00].

Variations of anonymity can be found in other systems, for instance, payment anonymity, where the adversary cannot identify the initiator of a payment.

Another typical privacy property is untraceability – the adversary cannot tell whether two choices (sending message A versus B, or voting for candidate C versus D) are made by the same user. This property was first proposed to capture the requirement that a bank should not be able to trace payment records to the same account [Cha88]. Later, this privacy property is showed to be highly desired in location-based services [DDS11] and RFID systems [HMZH08, vMR08]. It has been shown that anonymity and untraceability are not comparable with respect to which one is stronger ([ACRR10])[1].

In addition to the above mentioned privacy properties, a large number of other privacy properties have also been proposed [BMW03, CL01, HM08]. With the number of properties increasing, different names were used to capture the same requirement, and the same name was used to capture different requirements by different authors (for examples, see [BP11]). Pfitzmann et al. proposed terminology of privacy properties which distinguishes *anonymity*, *unlinkability*, *unobservability* and *pseudonymity* [PK00].

- Anonymity is defined as not being identifiable within a set of subjects;
- Unlinkability is the state that two or more items are no more and no less related after the adversary's observation than they are related concerning the adversary's a-priori knowledge;

[1] Note that the authors of [ACRR10] use a different terminology for untraceability, namely unlinkability.

- Unobservability is defined as items being indistinguishable from any other items;

- Pseudonymity is defined as the state of using a pseudonym as identity.

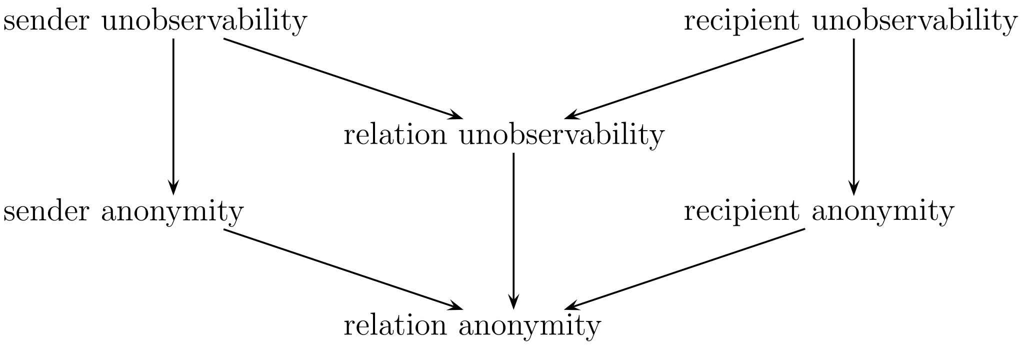

Figure 2.1: Relations between anonymity and unobservability [PK00].

The relations of these privacy properties are as follows: unlinkability is a sufficient condition for anonymity, but not a necessary condition; unobservability implies anonymity. In the context where sending and receiving of messages are the items of interest, anonymity can be defined as unlinkability of an item and identifier of a subject, and the relations between anonymity and unobservability are as shown in Figure 2.1.

Note that other generalisations of privacy properties exist. For instance, Chothia et al., distinguish player anonymity and choice anonymity as follows [COPD06].

- Player anonymity is defined as the adversary being unable to distinguish who makes a particular choice (data or action).

- Choice anonymity is defined as the adversary being unable to distinguish which choice a player made.

Differences between the various privacy properties are often subtle, since they are used to capture similar requirements. In addition, due to the differences in context, privacy properties are often not easy to compare. For example, some authors state that anonymity and unlinkability are actually the same, while others distinguish them [BP11]. Bohli and Pashalidis proposed a framework in which privacy properties are defined in a consistent and comparable manner [BP11]. This framework considers systems having finite runs. Each run is uniquely associated with a user. They modelled the correspondence between the runs and the set of its users as a function f. Privacy aims to ensure properties of f. Privacy properties considered in the framework describe potentially different degrees to which f remains hidden from the adversary. The privacy properties are distinguished by the information potentially revealed to the adversary:

- the set of user identifiers ($\mathtt{U}_f$);

- the number of runs corresponding to each participant ($\mathtt{Q}_f$);

- the partition of runs that is induced by f ($\mathtt{P}_f$);

- the multiset of equivalence class sizes with respect to $\mathtt{P}_f$ ($\mathtt{C}_f$).

The following privacy properties are considered: strong anonymity (SA), strong unlinkability with participation hiding (SUP), strong unlinkability with usage hiding (SUU), weak unlinkability with participation hiding (WUP), weak unlinkability with usage hiding (WUU), weak unlinkability (WU), pseudonymity (PS), anonymity (AN), and weak anonymity (WA). The properties are defined in Table 2.1. The relations between the privacy properties are shown in the hierarchy in Figure 2.2.

privacy property	**A system providing the property hides f except for**
SA	
SUP	$\lvert\mathtt{U}_f\rvert$
SUU	$\mathtt{U}_f$
WUP	$\mathtt{C}_f$
WUU	$\mathtt{U}_f$ and $\mathtt{C}_f$
WU	$\mathtt{Q}_f$
PS	$\mathtt{P}_f$
AN	$\mathtt{P}_f$ and $\mathtt{U}_f$
WA	$\mathtt{P}_f$ and $\mathtt{Q}_f$

Table 2.1: Definitions of privacy properties [BP11].

$$\begin{tikzcd}
SA \arrow[dr] & & & \\
& SUP \arrow[r] \arrow[d] & WUP \arrow[r] \arrow[d] & PS \arrow[d] \\
& SUU \arrow[r] \arrow[dr] & WUU \arrow[r] \arrow[d] & AN \arrow[d] \\
& & WU \arrow[r] & WA
\end{tikzcd}$$

Figure 2.2: Relations of privacy properties [BP11].

2.1.2 Enforced privacy properties

A common feature of the above privacy properties is that users are assumed to keep their information private. Recently, it has been identified that in some cases users may reveal their private information due to bribery, blackmail or extortion, and such revealing information is different from the revealing information by the compromised users. Normally, compromised users are considered as part of the adversary, as the adversary fully controls the compromised users. However, the bribed or coerced users may lie to the adversary if it is possible. Thus, bribed or coerced users are not fully trusted by the adversary.

Privacy properties that account for revealing information from bribed or coerced users (enforced privacy properties) are first proposed in e-voting. To prevent vote-buying, a property named *receipt-freeness* is proposed [BT94]. An e-voting protocol is receipt-free if a voter cannot prove his vote to a vote-buyer. Later, a stronger property *coercion-resistance* is proposed to prevent coercion in e-voting [Oka97]. An e-voting protocol is coercion-resistant if a voter cannot be forced to vote in a certain way. Coercion-resistance is stronger than receipt-freeness because a vote-buyer is not assumed to interrupt the voting process, while a coercer can communicate with the voter during the voting process[2]. Inspired by the enforced privacy properties in e-voting, similar properties are proposed in e-auctions [SM00, AS02]. In particular, in sealed-bid e-auctions, receipt-freeness for non-winning bidders is required to prevent the adversary from trying to control the winning price of the auction. Similar requirements have been found in the e-health domain, for example, a doctor should not be able to prove his prescriptions to a pharmaceutical company [dDLVV08].

2.2 Formalisations of privacy properties

In order to verify a claimed privacy property of a protocol, precise definitions of the property are required. A privacy property can be defined in different manners. For instance, we distinguish binary privacy from quantitative privacy.

- Binary privacy: A protocol either satisfies a privacy property or not.
- Quantitative privacy: It defines to which extent a protocol satisfies a claimed privacy property. For example, sender anonymity can be quantified by the number of participants from which the adversary cannot identify the sender [Cha88].

Quantitative enforced privacy properties have been defined for e-voting in a formal framework proposed by Jonker, Pang and Mauw – the JMP framework [JPM09]. In this framework, the enforced privacy property, coercion-resistance, is quantified using the size of a candidate set. If a voter is coerced to vote a candidate in the set, the voter can vote for another candidate in the set and lie to the adversary, without the adversary detecting whether the voter lied. Many ways to quantify privacy can be found [Cha88, RR98, BPS00]. How to quantify (enforced) privacy is beyond our concern. This thesis focuses on the first category, binary privacy.

Definitions of a privacy property also vary depending on the verification techniques used to prove the satisfaction of the definition. We distinguish directly proving a privacy property from proving a privacy property in a symbolic model. Directly proving (e.g., using game-based provable security) is normally achieved by showing that the adversary cannot solve the underlying hard problem in order to break the property, for example, integer factoring, discrete logarithm, 3-SAT, etc.

- Game-based provable security: A privacy property is defined as a game of the adversary and a hypothetical challenger. The privacy property is sat-

[2] This relation is based on the formal definitions of the two properties proposed by Delaune et al. [DKR09]. In informal definitions, the distinction between the two properties are not so clear.

isfied if no polynomially bounded adversary has a non-negligible advantage against the challenger in the game. Enforced privacy properties in e-voting have been defined in this way: receipt-freeness for a specific voting protocol (Prêt à Voter) [KR11] and a generic coercion-resistance for the e-voting domain [KTV10].

- Symbolic model: Typically, the Dolev-Yao assumption is adopted: Cryptographic primitives are assumed to be perfect, e.g., the adversary cannot undo an encryption; and messages are considered to be abstract, e.g., data are expressed as symbols instead of bit-strings.

As verification in symbolic models is easier to carry out and is often supported by tools. This thesis lies in the symbolic category.

In this category, formalisations of privacy properties vary depending on the used formal models. For instance,

- using epistemic model [SS99, HO05]: Protocols are modelled as knowledge of users and the adversary. Epistemic logic is used to reason about knowledge. Privacy properties are formalised as epistemic formulas. Enforced privacy property in e-voting have been formalised based on epistemic logic, for instance, receipt-freeness by Jonker and Pieters [JP06] and a framework for coercion-resistance proposed by Küsters and Truderung – the KT framework [KT09].
- using process algebra: The behaviour of a system can be intuitively modelled as a process. Privacy properties are typically modelled as relations of processes.

Compared to epistemic logic, process algebra is better at modelling the behaviour of protocols. Process algebras are designed for specifying concurrent systems, and thus are very suitable to model e-services in which users are often highly distributed. In addition, process algebras are often equipped with proof techniques for process equivalences and some of them are supported by automatic verification tools. In this thesis, privacy properties are formalised using process algebra.

Many process algebras are used to model cryptographic protocols and formalise privacy properties, for example, CSP (communicating sequential processes) [Hoa78, Sch96, SS96, OC02], spi calculus [AG97] and the applied pi calculus [AF01, KR05, DKR09]. Enforced privacy properties were first formalised using the applied pi calculus for a specific e-voting protocol [KR05]. Later, a framework for e-voting was proposed using the applied pi calculus – the DKR framework [DKR09]. In addition, enforced privacy properties for weighted voting were proposed using the applied pi calculus as well – the DLL framework proposed by Dreier, Lafourcade and Lakhnech [DLL12]. The formalisation using the applied pi calculus is rather mature. Furthermore, compared to other process algebras, the applied pi calculus provides an intuitive way to model cryptographic protocols. And the verification of many properties is supported by the verification tool – ProVerif. Thus, in this thesis, we use the applied pi calculus to model protocols and formalise enforced privacy properties. In order to formalise enforced privacy properties, the cooperation between users and the adversary needs to be formalised first. So far in the literature, the cooperation is only formalised in the context of e-voting as bribery or coercion using the applied pi calculus. This formalisation is proposed in a framework by Delaune, Kremer and Ryan – the DKR framework and is applied in many formal definitions of enforced privacy properties [KR05, BHM08, DKR09, DLL12]. Thus, we follow the methodology of the DKR framework.

The DKR framework. In this framework, e-voting protocols are modelled as processes in which voter processes and authority processes run in parallel, using the applied pi calculus. To model the enforced privacy property – receipt-freeness, the bribery of a voter is formalised as the voter forwarding all private information to the adversary over a fresh channel. The process for a bribed voter is constructed from the original voter process – whenever a private data is generated or read-in in the original voter process, the data is sent to the adversary in the bribed voter process. Receipt-freeness is defined as follows: an e-voting protocol is receipt-free if there exists a process in which a voter can lie to the adversary and the adversary cannot distinguish the process from the constructed process in which the voter genuinely forwards information to the adversary. The coercion of a voter is stronger: The adversary is able to communicate with the coerced voter during the voting procedure. This is modelled by giving the adversary the ability to prepare information for the coerced voter. The coerced voter process is constructed from the original process in a similar way as constructing the bribed voter process. The main difference is that in the coerced voter process, the voter reads in information from the adversary over a particular channel if there exists information sending in the original voter process, and sends the information prepared by the adversary instead. Since the adversary prepares information for a coerced voter, the vote may be decided by the adversary instead of the voter. Thus, coercion-resistance is defined as: for any prepared information by the adversary (possibly empty) such that the intended vote is a (no matter it is decided by the adversary or the voter), there is a process in which the voter can vote for c, and the adversary cannot distinguish the process from the process in which the voter genuinely shares information with the adversary and uses the information prepared by the adversary. Detailed formalisations are introduced in the next chapter (Chapter 3).

3

Preliminaries

As stated in Chapter 1, the design of cryptographic protocols is error-prone and formal verification has shown its strength in proving or disproving correctness of cryptographic protocols. To verify whether a protocol satisfies a claimed property, a necessary step is to specify the protocol. Therefore, in this chapter, we first introduce a formal language for modelling protocols – the applied pi calculus – in Section 3.1. Given a protocol modelled in the applied pi calculus, many properties can be automatically verified using the tool ProVerif. Thus, we introduce ProVerif in the next section (Section 3.2). As stated in Chapter 2, we focus on formalising enforced privacy following the DKR formal framework of modelling bribery and coercion. Thus, we show several definitions in the DKR framework in Section 3.3.

3.1 The applied pi calculus

The applied pi calculus is a language for modelling and analysing concurrent systems, in particular cryptographic protocols. The following briefly introduces its syntax, semantics and equivalence relations. It is mainly based on [AF01, RS10].

3.1.1 Syntax

The calculus assumes an infinite set of *names*, which are used to model communication channels or other atomic data, an infinite set of *variables*, which are used to model received messages, and a signature Σ consisting of a finite set of *function symbols*, which are used to model cryptographic primitives. Each function symbol has an arity. A function symbol with arity zero is a constant. *Terms* (which are used to model messages) are defined as names, variables, or function symbols applied to terms (see Figure 3.1).

$$\begin{array}{ll}
M, N, T ::= & \texttt{terms} \\
\quad \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{m}, \mathsf{n}, \ldots & \texttt{names} \\
\quad x, y, z & \texttt{variables} \\
\quad \mathsf{f}(M_1, \ldots, M_\ell) & \texttt{function application}
\end{array}$$

Figure 3.1: Terms in the applied pi calculus.

Example 3.1 (function symbols and terms). *Typical function symbols are* enc *with arity 2 for encryption,* dec *with arity 2 for decryption. The term for encrypting* x *with a key* k *is* $\mathsf{enc}(x, k)$.

The applied pi calculus assumes a sort system for terms. Terms can be of a base type (e.g., Key or a universal base type Data) or type $\mathsf{Channel}\langle\omega\rangle$ where ω is a type. A variable and a name can have any type. A function symbol can only be applied to and return, terms of base type. Terms are assumed to be well-sorted and substitutions preserve types.

Terms are often equipped with an equational theory E – a set of equations on terms. The equational theory is normally used to capture features of cryptographic primitives. The equivalence relation induced by E is denoted as $=_E$.

Example 3.2 (equational theory)**.** *The behaviour of symmetrical encryption and decryption can be captured by the following equation:*

$$\mathsf{dec}(\mathsf{enc}(x,k),k) =_E x,$$

where x, k are variables.

Systems are described as processes: plain processes and extended processes (see Figure 3.2). In Figure 3.2, M, N are terms, $\mathtt{n}$ is a name, x is a variable and v is a metavariable, standing either for a name or a variable. The null process 0 does nothing. The parallel composition $P \mid Q$ represents the sub-process P and the sub-process Q running in parallel. The replication $!P$ represents an infinite number of process P running in parallel. The name restriction $\nu\mathtt{n}.P$ bounds the name $\mathtt{n}$ in the process P, which means the name $\mathtt{n}$ is secret to the adversary. The conditional evaluation $M =_E N$ represents equality over the equational theory rather than strict syntactic identity. The message input $\mathsf{in}(v,x).P$ reads a message from channel v, and bounds the message to the variable x in the following process P. The message output $\mathsf{out}(v,M).P$ sends the message M on the channel v, and then runs the process P. Extended processes add restriction on variables and active substitutions. The variable restriction $\nu x.A$ bounds the variable x in the process A. The active substitution $\{M/x\}$ replaces variable x with term M in any process that it contacts with.

$P, Q, R ::=$	`plain processes`
0	`null process`
$P \mid Q$	`parallel composition`
$!P$	`replication`
$\nu\mathtt{n}.P$	`name restriction`
`if` $M =_E N$ `then` P `else` Q	`conditional`
$\mathsf{in}(v,x).P$	`message input`
$\mathsf{out}(v,M).P$	`message output`
$A, B, C ::=$	`extended processes`
P	`plain process`
$A \mid B$	`parallel composition`
$\nu\mathtt{n}.A$	`name restriction`
$\nu x.A$	`variable restriction`
$\{M/x\}$	`active substitution`

Figure 3.2: Processes in the applied pi calculus.

Example 3.3 (process and protocol). *In a protocol, A generates a nonce* $\mathtt{m}$*, encrypts the message with a secret key* $\mathtt{k}$*, then sends the encrypted message to B. The process modelling the behaviour of A is:*

$$Q_A := \nu\mathtt{m}.\mathsf{out}(\mathtt{ch}, \mathsf{enc}(\mathtt{m}, \mathtt{k})).$$

where $\mathtt{ch}$ *is a public channel.*
The process modelling the behaviour B is:

$$Q_B := \mathsf{in}(\mathtt{ch}, x).$$

The process Q models the complete protocol, composed of A and B.

$$Q := \nu\mathtt{k}.(Q_A \mid Q_B).$$

Names and variables have scopes. A name is *bound* if it is under restriction. A variable is *bound* by restrictions or inputs. Names and variables are *free* if they are not delimited by restrictions or by inputs. The sets of free names, free variables, bound names and bound variables of a process A are denoted as $\mathsf{fn}(A)$, $\mathsf{fv}(A)$, $\mathsf{bn}(A)$ and $\mathsf{bv}(A)$, respectively. A term is *ground* when it does not contain variables. A process is *closed* if it does not contain free variables.

Example 3.4 (scopes of names and variables). *In Example 3.3,* $\mathtt{ch}$ *is a free name representing a public channel. Name* $\mathtt{k}$ *is bound in process Q; name* $\mathtt{m}$ *is bound in process Q_A. Variable x is bound in process Q_B.*

A *frame* is defined as an extended process built up from 0 and active substitutions by parallel composition and restrictions. The active substitutions in extended processes allow us to map an extended process A to its frame $\mathsf{frame}(A)$ by replacing every plain process in A with 0.

Example 3.5 (frame). *The frame of the process $\nu\mathtt{m}.(\mathsf{out}(\mathtt{ch}, x)) \mid \{\mathtt{m}/x\})$, denoted as $\mathsf{frame}(\nu\mathtt{m}.(\mathsf{out}(\mathtt{ch}, x)) \mid \{\mathtt{m}/x\}))$ is $\nu\mathtt{m}.(\{\mathtt{m}/x\})$.*

The *domain* of a frame B, denoted as $\mathsf{domain}(B)$, is the set of variables for which the frame defines a substitution.

Example 3.6 (domain). *The domain of the frame in Example 3.5, denoted as $\mathsf{domain}(\nu\mathtt{m}.(\{\mathtt{m}/x\}))$ is $\{x\}$.*

A *context* $\mathcal{C}[\_]$ is defined as a process with a hole, which may be filled with any process. An evaluation context is a context whose hole is not under a replication, a condition, an input or an output.

Example 3.7 (context). *Process $\nu\mathtt{k}.(Q_A \mid \_)$ is and evaluation context. When we fill the hole with process Q_B, we obtain the process in Example 3.3.*

Finally, we abbreviate the process $\nu\mathtt{n}_1 \cdots \nu\mathtt{n}_n$ as $\nu\tilde{\mathtt{n}}$, abbreviate the process $\nu\mathtt{n}_1 \cdots \nu\mathtt{n}_{i-1}.\nu\mathtt{n}_{i+1}. \cdots .\nu\mathtt{n}_n$ as $\nu\tilde{\mathtt{n}}/\mathtt{n}_i$ (erasing process $\nu\mathtt{n}_i$ from $\nu\tilde{\mathtt{n}}$), and abbreviate $\{M_1/x_1\} \cdots \{M_n/x_n\}$ as $\{M_1/x_1, \cdots, M_n/x_n\}$.

3.1.2 Operational semantics

The operational semantics of the applied pi calculus is defined by: 1) structural equivalence ($\equiv$), 2) internal reduction ($\rightarrow$), and 3) labelled reduction ($\xrightarrow{\alpha}$) of processes.

1) Informally, two processes are structurally equivalent if they model the same thing but differ in structure. Formally, structural equivalence of processes is the smallest equivalence relation on extended process that is closed by α-conversion on names and variables, by application of evaluation contexts as shown in Figure 3.3.

$$
\begin{array}{llcll}
\texttt{PAR}-\texttt{0} & A \mid 0 & \equiv & A & \\
\texttt{PAR}-\texttt{A} & A \mid (B \mid C) & \equiv & (A \mid B) \mid C & \\
\texttt{PAR}-\texttt{C} & A \mid B & \equiv & B \mid A & \\
\texttt{REPL} & !P & \equiv & P \mid !P & \\
\texttt{SUBST} & \{M/x\} \mid A & \equiv & \{M/x\} \mid A\{M/x\} & \\
\texttt{NEW}-\texttt{0} & \nu u.0 & \equiv & 0 & \\
\texttt{NEW}-\texttt{C} & \nu u.\nu v.A & \equiv & \nu v.\nu u.A & \\
\texttt{NEW}-\texttt{PAR} & A \mid \nu v.B & \equiv & \nu v.(A \mid B) & \texttt{if } v \notin \mathsf{fn}(A) \cup \mathsf{fv}(A) \\
\texttt{ALIAS} & \nu x.\{M/x\} & \equiv & 0 & \\
\texttt{REWRITE} & \{M/x\} & \equiv & \{N/x\} & \texttt{if } M =_E N
\end{array}
$$

Figure 3.3: Structural equivalence in the applied pi calculus.

2) Internal reduction is the smallest relation on extended processes closed under structural equivalence, application of evaluation of contexts as shown in Figure 3.4.

$$
\begin{array}{lll}
\texttt{COMM} & \mathsf{out}(\mathsf{c}, x).P \mid \mathsf{in}(\mathsf{c}, x).Q \ \rightarrow \ P \mid Q \\
\texttt{THEN} & \texttt{if } N =_E N \texttt{ then } P \texttt{ else } Q \ \rightarrow \ P \\
\texttt{ELSE} & \texttt{if } M =_E N \texttt{ then } P \texttt{ else } Q \ \rightarrow \ Q \\
 & \quad \texttt{for ground terms } M, N \texttt{ where } M \neq_E N
\end{array}
$$

Figure 3.4: Internal reduction in the applied pi calculus.

3) The labelled reduction models the environment interacting with the processes. It defines a relation $A \xrightarrow{\alpha} A'$ as in Figure 3.5. The label α is either reading a term from the process's environment, or sending a name or a variable of base type to the environment.

3.1.3 Equivalences

The applied pi calculus defines *observational equivalence* and *labelled bisimilarity* to model the indistinguishability of two processes by the adversary. It is proved that the two relations coincide [AF01, Liu11]. We mainly use the labelled bisimilarity for the convenience of proofs. Labelled bisimilarity is based on *static equivalence*: labelled bisimilarity compares the dynamic behaviour of processes, while static equivalence compares their static states (as represented by their frames).

`IN`	$\mathsf{in}(\mathsf{c}, x).P \xrightarrow{\mathsf{in}(\mathsf{c},M)} P\{M/x\}$
`OUT − ATOM`	$\mathsf{out}(\mathsf{c}, v).P \xrightarrow{\mathsf{out}(\mathsf{c},v)} P$
`OPEN − ATOM`	$\dfrac{A \xrightarrow{\mathsf{out}(\mathsf{c},v)} A' \quad v \neq \mathsf{c}}{\nu v.A \xrightarrow{\nu v.\mathsf{out}(\mathsf{c},v)} A'}$
`SCOPE`	$\dfrac{A \overset{\alpha}{\rightarrow} A' \quad v \texttt{ does not occur in } \alpha}{\nu v.A \overset{\alpha}{\rightarrow} \nu v.A'}$
`PAR`	$\dfrac{A \overset{\alpha}{\rightarrow} A' \quad \mathsf{bv}(\alpha) \cup \mathsf{fv}(B) = \mathsf{bn}(\alpha) \cap \mathsf{fn}(B) = \emptyset}{A \mid B \overset{\alpha}{\rightarrow} A' \mid B}$
`STRUCT`	$\dfrac{A \equiv B \quad B \overset{\alpha}{\rightarrow} B' \quad A' \equiv B'}{A \overset{\alpha}{\rightarrow} A'}$

Figure 3.5: Labelled reduction in the applied pi calculus.

Definition 3.8 (static equivalence)**.** *Two terms M and N are equal in the frame B, written as $(M =_E N)B$, iff there exists a set of restricted names $\tilde{\mathtt{n}}$ and a substitution σ such that $B \equiv \nu\tilde{\mathtt{n}}.\sigma$, $M\sigma =_E N\sigma$ and $\tilde{\mathtt{n}} \cap (\mathsf{fn}(M) \cup \mathsf{fn}(N)) = \emptyset$.*

Closed frames B and B' are statically equivalent, denoted as $B \approx_s B'$, if
(1) $\mathsf{domain}(B) = \mathsf{domain}(B')$;
(2) $\forall$ terms M, N: $(M =_E N)B$ iff $(M =_E N)B'$.

Extended processes A, A' are statically equivalent, denoted as $A \approx_s A'$, if their frames are statically equivalent: $\mathsf{frame}(A) \approx_s \mathsf{frame}(A')$.

Example 3.9 (equivalence of frames [AF01])**.** *The frame B and the frame B', are equivalent. However, the two frames are not equivalent to frame B'', because the adversary can discriminate B'' by testing $y =_E \mathsf{f}(x)$.*

$$\begin{aligned} B &:= \nu M.\{M/x\} \mid \nu N.\{N/y\} \\ B' &:= \nu M.(\{\mathsf{f}(M)/x\} \mid \{\mathsf{g}(M)/y\}) \\ B'' &:= \nu M.(\{M/x\} \mid \{\mathsf{f}(M)/y\}) \end{aligned}$$

where f and g are two function symbols without equations.

Example 3.10 (static equivalence)**.** *Process $\{M/x\} \mid Q_1$ is static equivalent to process $\{M/x\} \mid Q_2$ where Q_1 and Q_2 are two closed plain process, because the frame of the two processes are the statically equivalent, i.e., $\{M/x\} \approx_s \{M/x\}$.*

Definition 3.11 (labelled bisimilarity)**.** Labelled bisimilarity ($\approx_\ell$) *is the largest symmetric relation $\mathcal{R}$ on closed extended processes, such that $A \mathcal{R} B$ implies:*
(1) $A \approx_s B$;
(2) if $A \rightarrow A'$ then $B \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B';
*(3) if $A \overset{\alpha}{\rightarrow} A'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(B) = \emptyset$; then $B \rightarrow^* \overset{\alpha}{\rightarrow} \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B', where * denotes zero or more.*

3.2 ProVerif

The verification of protocols modelled in the applied pi calculus is supported by an automatic verification tool ProVerif [Bla01, Bla02, Bla04]. The tool has been used to verify many secrecy, authentication and privacy properties, e.g., see [AB05, ABF07, BC08].

ProVerif takes a protocol and a property modelled in the applied pi calculus as input, returns a proof of correctness or flaws as output. A protocol modelled in the applied pi calculus is translated to Horn clauses [Hor51]. The adversary's capabilities are modelled as Horn clauses as well. Using these clauses, verification of secrecy and authentication is equivalent to determining whether a particular state is reachable in the execution of the protocol.

Secrecy of a term is defined as the adversary cannot obtain the term by communicating with the protocol and/or applying cryptography on the output of the protocol [AB05]. The secrecy property is modelled as a predicate in ProVerif: the query of secrecy of term M is "`attack:M`" [Bla01]. ProVerif determines whether the term M can be inferred from the Horn clauses representing the adversary knowledge.

Authentication is captured by correspondence properties of events in processes: if one event happens the other event must have happened [ABF07, Bla09]. Events are tags which mark important stages reached by the protocol. Events have arguments, which allow us to express relationships between the arguments of events. A correspondence property is a formula of the form: $f\langle M\rangle \rightsquigarrow g\langle N\rangle$. That is, in any process if event $f\langle M\rangle$ has been executed then the event $g\langle N\rangle$ must have been previously executed and any relationship between M and N must be satisfied. To capture stronger authentication where an injective relationship between executions of participants is required, an injective correspondence property $f\langle M\rangle \rightsquigarrow_{inj} g\langle N\rangle$ is defined: in any process if event $f\langle M\rangle$ is executed, there is a distinct earlier occurrence of the event $g\langle N\rangle$ being executed and the relationship between M and N is satisfied.

A correspondence property $f\langle M\rangle \rightsquigarrow g\langle N\rangle$ can be translated into a secrecy property, thus can be verified in ProVerif [Bla02]. The translation is as follows: the begin event "$g\langle N\rangle.P$" is replaced with "if $N = N'$ then 0 else P" and the end event "$f\langle M\rangle.P'$" is replaced with "$\mathsf{out}(\mathsf{c}, \mathsf{auth}(M)).P'$". For any closed term N', if the modified process preserves the secrecy of $\mathsf{auth}(N')$, the process satisfies the correspondence property. The intuition is that when the process does not execute event $g\langle N\rangle$, the event $f\langle M\rangle$ cannot be executed, thus, the event $f\langle M\rangle$ should be secret. This intuition is reflected in the translated process: given a term N' as the parameter of the end event,

- if $N = N'$, the process stops, meaning that if the original process tries to execute $g\langle N\rangle$, since the begin event is executed, the secrecy of the end event does not reflect the verification result of the correspondence property, thus the process stops;

- when $N' \neq N$, the original process does not execute $g\langle N\rangle$, if the correspondence property is satisfied, the secrecy of $f\langle N'\rangle$ in the translated process should hold, if the translated process does not satisfy secrecy of $f\langle N'\rangle$, then the trace in which the secrecy of $f\langle N'\rangle$ is flawed shows that the end event is executed without the begin event being executed before, thus the correspondence property is not satisfied.

The predicate to query non-injective correspondence property $f\langle M\rangle \rightsquigarrow g\langle N\rangle$ is `ev:f(M) ==> ev:g(N)` and the predicate to query injective correspondence property $f\langle M\rangle \rightsquigarrow_{inj} g\langle N\rangle$ is `evinj:f(M) ==> evinj:g(N)`.

Secrecy and authentication are properties expressed as predicates on system behaviours. However, not all properties can be expressed as predicates on system behaviours. Many of such properties can be expressed as equivalences of processes, for example, strong secrecy which is defined as the adversary's inability to distinguish when the secret changes. Therefore, in addition, ProVerif provides automatic verification of labelled bisimilarity of two processes which differ only in the choice of some terms [BAF08]. An operation "$\mathsf{choice}[M, N]$" is introduced to model the different choices of a term in the two processes. Using this operation, the two processes can be written as one process – a *bi-process*.

Example 3.12. *To verify the equivalence*

$$\nu \mathsf{a}.\nu \mathsf{b}.\mathsf{out}(\mathtt{ch}, \mathsf{a}).\mathsf{out}(\mathtt{ch}, \mathsf{e}) \approx_{\ell} \nu \mathsf{a}.\nu \mathsf{b}.\mathsf{out}(\mathtt{ch}, \mathsf{b}).\mathsf{out}(\mathtt{ch}, \mathsf{d})$$

where `ch` *is a public channel,* `e` *and* `d` *are two free names, we can query the following bi-process in ProVerif:*

$$P := \nu \mathsf{a}.\nu \mathsf{b}.\mathsf{out}(\mathtt{ch}, \mathsf{choice}[\mathsf{a}, \mathsf{b}]).\mathsf{out}(\mathtt{ch}, \mathsf{choice}[\mathsf{e}, \mathsf{d}]).$$

Using the first parameter of all "*choice*" operations in a bi-process P, we obtain one side of the equivalence (denoted as $\mathsf{fst}(P)$); using the second parameters, we obtain the other side (denoted as $\mathsf{snd}(P)$).

Example 3.13. *For the bi-process in Example 3.12, using the first parameter to replace each "choice" operation, we obtain* $\nu \mathsf{a}.\nu \mathsf{b}.\mathsf{out}(\mathtt{ch}, \mathsf{a}).\mathsf{out}(\mathtt{ch}, \mathsf{e})$*, which is the left hand side of the equivalence in Example 3.12; using the second parameter to replace each "choice" operation, we obtain* $\nu \mathsf{a}.\nu \mathsf{b}.\mathsf{out}(\mathtt{ch}, \mathsf{b}).\mathsf{out}(\mathtt{ch}, \mathsf{d})$*, which is the right hand side of the equivalence.*

Given a bi-process P, ProVerif determines whether $\mathsf{fst}(P)$ is labelled bisimilar to $\mathsf{snd}(P)$. The fundamental idea is that ProVerif tries to prove that the executions of the bi-process are uniform: when $\mathsf{fst}(P)$ can do a reduction to some Q_1, it implies that the bi-process can do a reduction to some bi-process Q, such that $\mathsf{fst}(Q) \equiv Q_1$ and symmetrically for $\mathsf{snd}(P)$ taking a reduction to Q_2.

ProVerif uses slightly different notations. We use both the notions in ProVerif and the applied pi calculus for the convenience of expression. The notations representing the same thing is presented in Table 3.1.

	Notations in ProVerif	**Notations in applied pi**
Equational theory	*reduc* /*equation* $M = N$.	$M =_E N$
Substitution	let $x = M$ in P	$P\{M/x\}$
Defining process	let $P = \cdots$.	$P := \cdots$
Generating names/variables	new $\mathtt{n}/x$	$\nu\mathtt{n}/x$

Table 3.1: Different notations in ProVerif and applied pi.

3.3 The DKR framework

ProVerif has been successfully used for helping analyse enforced privacy properties which are formalised for specific e-voting protocols, using the applied pi calculus [KR05, BHM08]. The specific formalisations of enforced privacy have been generalised in the DKR framework [DKR09]. We introduce some definitions in the framework which are mentioned in this thesis.

In the framework, a voting protocol with n_v of voters and n_{ad} of authorities is modelled as $P_{vote} := \nu chandata.(P_K \mid P_{v1} \mid \cdots \mid P_{vn_v} \mid P_{ad1} \mid \cdots \mid P_{adn_{ad}})$ where P_{vi} is an instance of a voter, P_{adj} is an instance of an authority, *chandata* is a set of private channel names and data, and P_K generates and distributes keys.

One requirement in e-voting is to protect the link between a voter and his vote. This requirement is captured by the privacy property *vote-privacy* as in Definition 3.14. Since the final result of a voting is normally published, when voters vote unanimously, every voter's vote is revealed. To avoid such situation, in the formalisation of vote-privacy, two voters are modelled and they need to vote differently.

Definition 3.14 (vote-privacy [DKR09]). *A voting protocol P_{vote} respects vote-privacy if $\mathcal{C}_v[P_{vA}\{a/vote\} \mid P_{vB}\{c/vote\}] \approx_\ell \mathcal{C}_v[P_{vA}\{c/vote\} \mid P_{vB}\{a/vote\}]$ for all possible a and c.*

In the definition, $\mathcal{C}_v[\_] := \nu chandata.(P_K \mid P_{v1} \mid \cdots \mid P_{vn_v-2} \mid \_ \mid P_{ad1} \mid \cdots \mid P_{adn_{ad}})$, $P_{vA}\{a/vote\}$ is a voter process in which the voted candidate *vote* is a, similarly, $P_{vB}\{c/vote\}$ is another voter process in which the voted candidate *vote* is c. On the right-hand side, the two voter processes swap voted candidates.

On top of vote-privacy, two enforced privacy properties are formalised: receipt-freeness and coercion-resistance. Receipt-freeness ensures that a voter cannot prove his vote to a vote-buyer. Coercion-resistance is defined to capture the requirement that a coercer should not be able to enforce a voter to vote in a certain way. In order to formalise the two enforced privacy properties, the behaviour of a bribed or coerced user needs to be formalised first.

The bribery of a voter is formalised as the voter forwarding all private information to the adversary over a fresh channel. Formally, the bribed behaviour of voter P_v is given by $P_v^{\mathtt{chc}}$ which is defined as follows:

Definition 3.15 (process $P^{\mathtt{chc}}$ [DKR09]). *Let P be a plain process and* $\mathtt{chc}$ *a fresh channel name. $P^{\mathtt{chc}}$, the process that shares all of P's secrets, is defined as:*

- $0^{\mathtt{chc}} \hat{=} 0$,

- $(P \mid Q)^{\mathsf{chc}} \hat{=} P^{\mathsf{chc}} \mid Q^{\mathsf{chc}}$,
- $(\nu \mathtt{n}.P)^{\mathsf{chc}} \hat{=} \nu \mathtt{n}.\mathsf{out}(\mathtt{chc}, \mathtt{n}).P^{\mathsf{chc}}$ *when* n *is a name of base type,*
- $(\nu \mathtt{n}.P)^{\mathsf{chc}} \hat{=} \nu \mathtt{n}.P^{\mathsf{chc}}$ *otherwise,*
- $(\mathsf{in}(v, x).P)^{\mathsf{chc}} \hat{=} \mathsf{in}(v, x).\mathsf{out}(\mathtt{chc}, x).P^{\mathsf{chc}}$ *when x is a variable of base type,*
- $(\mathsf{in}(v, x).P)^{\mathsf{chc}} \hat{=} \mathsf{in}(v, x).P^{\mathsf{chc}}$ *otherwise,*
- $(\mathsf{out}(v, M).P)^{\mathsf{chc}} \hat{=} \mathsf{out}(v, M).P^{\mathsf{chc}}$,
- $(!P)^{\mathsf{chc}} \hat{=} !P^{\mathsf{chc}}$,
- $(if\ \ M =_E N\ then\ P\ else\ Q)^{\mathsf{chc}} \hat{=} if\ \ M =_E N\ then\ P^{\mathsf{chc}}\ else\ Q^{\mathsf{chc}}$.

Delaune et al. also define the process transformation $A^{\backslash \mathsf{out}(\mathtt{chc}, \cdot)}$, which can be considered as the process A hides $\mathsf{out}(\mathtt{chc}, \cdot)$ (the outputs on the channel chc).

Definition 3.16 (process $A^{\backslash \mathsf{out}(\mathtt{chc}, \cdot)}$ [DKR09])**.** *Let A be an extended process. We define the process $A^{\backslash \mathsf{out}(\mathtt{chc}, \cdot)}$ as $\nu \mathtt{chc}.(A \mid !\mathsf{in}(\mathtt{chc}, x))$.*

Using the above two definitions, receipt-freeness is defined as follows: There exists a process in which the bribed/coerced voter can lie to the adversary and the adversary cannot tell whether the voter lied.

Definition 3.17 (receipt-freeness [DKR09])**.** *A voting protocol P_{vote} is receipt-free if there exists a closed plain process P_f such that*

- ${P_f}^{\backslash \mathsf{out}(\mathtt{chc}, \cdot)} \approx_\ell P_{vA}\{a/vote\}$,
- $\mathcal{C}_v[P_{vA}\{c/vote\}^{\mathsf{chc}} \mid P_{vB}\{a/vote\}] \approx_\ell \mathcal{C}_v[P_f \mid P_{vB}\{c/vote\}]$

Process P_f is the process in which the voter lies to the adversary The first equivalence shows the real behaviour of process P_f. The second equivalence shows that the adversary cannot tell whether the voter lied.

The coercion of a voter is defined as that the adversary is able to communicate with the coerced voter during the voting procedure. This is modelled by adding to the bribery behaviour the ability for the adversary to prepare information for the coerced voter to use. Formally, the coerced behaviour of P_v is given by ${P_v}^{\mathsf{c}_{out}, \mathsf{c}_{in}}$ which is defined in Definition 3.18.

Definition 3.18 (process $P^{\mathsf{c}_{out}, \mathsf{c}_{in}}$ [DKR09])**.** *Let P be a plain process and c_{out}, c_{in} be channel names. We define $P^{\mathsf{c}_{out}, \mathsf{c}_{in}}$ as follows:*

- $0^{\mathsf{c}_{out}, \mathsf{c}_{in}} \hat{=} 0$,
- $(P \mid Q)^{\mathsf{c}_{out}, \mathsf{c}_{in}} \hat{=} P^{\mathsf{c}_{out}, \mathsf{c}_{in}} \mid Q^{\mathsf{c}_{out}, \mathsf{c}_{in}}$,
- $(\nu \mathtt{n}.P)^{\mathsf{c}_{out}, \mathsf{c}_{in}} \hat{=} \nu \mathtt{n}.\mathsf{out}(\mathsf{c}_{out}, \mathtt{n}).P^{\mathsf{c}_{out}, \mathsf{c}_{in}}$ *when* n *is a name of base type,*
- $(\nu \mathtt{n}.P)^{\mathsf{c}_{out}, \mathsf{c}_{in}} \hat{=} \nu \mathtt{n}.P^{\mathsf{c}_{out}, \mathsf{c}_{in}}$ *otherwise,*

- $(\mathsf{in}(v,x).P)^{\mathsf{c}_{out},\mathsf{c}_{in}} \hat{=} \mathsf{in}(v,x).\mathsf{out}(\mathsf{c}_{out},x).P^{\mathsf{c}_{out},\mathsf{c}_{in}}$ *when* x *is a variable of base type,*
- $(\mathsf{in}(v,x).P)^{\mathsf{c}_{out},\mathsf{c}_{in}} \hat{=} \mathsf{in}(v,x).P^{\mathsf{c}_{out},\mathsf{c}_{in}}$ *otherwise,*
- $(\mathsf{out}(v,M).P)^{\mathsf{c}_{out},\mathsf{c}_{in}} \hat{=} \mathsf{out}(v,M).P^{\mathsf{c}_{out},\mathsf{c}_{in}}$,
- $(!P)^{\mathsf{c}_{out},\mathsf{c}_{in}} \hat{=} !P^{\mathsf{c}_{out},\mathsf{c}_{in}}$,
- $(\mathit{if}\ M =_E N\ \mathit{then}\ P\ \mathit{else}\ Q)^{\mathsf{c}_{out},\mathsf{c}_{in}} \hat{=} \mathit{if}\ M =_E N\ \mathit{then}\ P^{\mathsf{c}_{out},\mathsf{c}_{in}}\ \mathit{else}\ Q^{\mathsf{c}_{out},\mathsf{c}_{in}}$.

Coercion-resistance is defined similarly to receipt-freeness. The difference is that in coercion, the coerced voter's vote may be decided by the adversary, since the adversary may prepare information for the coerced voter. For the simplicity of modelling, the adversary's ability of providing information for the coerced voter is separated from the ability to distinguish two processes. The adversary's ability to provide information is modelled as a context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_ \mid P)$ where P models the adversary's behaviour, c_{out} and c_{in} are the two communication channels between the adversary and the coerced voter. What the adversary wants the coerced voter to vote is modelled by an equivalence $\mathcal{C}_v[\mathcal{C}[P_{vA}\{?/vote\}] \mid P_{vB}\{a/vote\}] \approx_\ell \mathcal{C}_v[(P_{vA}\{c/vote\})^{\mathsf{chc}} \mid P_{vB}\{a/vote\}]$. The right side of the equivalence decides the vote. On the left side, symbol "?" is used to represent any candidate, as no matter what the symbol is replaced with, it does not affect the vote.

Definition 3.19 (coercion-resistance [DKR09]). *A voting protocol P_{vote} satisfies coercion-resistance if there exists a closed plain process P_f such that for any $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_ \mid P)$ satisfying $\tilde{n} \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and $\mathcal{C}_v[\mathcal{C}[P_{vA}\{?/vote\}] \mid P_{vB}\{a/vote\}]$ $\approx_\ell \mathcal{C}_v[(P_{vA}\{c/vote\})^{\mathsf{chc}} \mid P_{vB}\{a/vote\}]$, we have*

- $\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{chc},\cdot)} \approx_\ell P_{vA}\{a/vote\}$
- $\mathcal{C}_v[\mathcal{C}[P_{vB}\{?/vote\}^{\mathsf{c}_{out},\mathsf{c}_{in}}] \mid P_{vj}\{a/vote\}] \approx_\ell \mathcal{C}_v[\mathcal{C}[P_f] \mid P_{vj}\{c/vote\}]$

Intuitively, a protocol satisfies coercion-resistance, if for any successful coercion (the coerced voter follows the adversary's instructions and successfully votes the candidate desired by the adversary), there exists a process P_f in which the coerced voter can diverge from the coerced behaviour and vote a different candidate; in addition, the adversary cannot tell whether the coerced voter genuinely follows his instruction.

4

Enforced privacy in e-auctions

Besides e-voting, enforced privacy requirements have also been identified in the e-auction domain. Unlike in the e-voting domain, enforced privacy has not been formally defined in e-auctions. In this chapter, we briefly introduce e-auctions, and then discuss privacy and enforced privacy requirements in the e-auction domain. Next we formalise privacy and enforced privacy in e-auctions. Finally we verify the formalised privacy properties of an e-auction protocol in a case study.

4.1 Introduction to e-auctions

Auctions are ways to negotiate exchange of goods and services. We use *e-auctions* to refer to auctions over the Internet. A typical (e-)auction works as follows: a seller offers items to bid, then bidders submit bids, finally auctioneers decide the winner. Compared to the traditional auctions, where bidders attend the auction in person, e-auctions attract more participants, as users with the Internet can join an auction. Real-life examples are websites like *eBay, eBid, Yahoo!auctions* and so on.

There are different types of (e-)auctions. For instance, depending on whether the bids are public, there are sealed-bid auctions and open-bid auctions.

- *Sealed-bid auctions*: There are two phases in an auction: the bidding phase and the opening phase. Bidders can only submit bids in the bidding phase. All bids are sealed in the bidding phase and opened in the opening phase.
- *Open-bid auctions*: Bids are broadcast to all participants.

Other criteria to classify (e-)auctions exist. For example, depending on the bidding price increases or decreases, there are English auctions (A bid needs to be higher than the previous one; the winning bid is the final bid) and Dutch auctions (The bidding price decreases until a bid is submitted); depending on the calculation of payment, there are first-price auctions (The winner pays the price he bid, the highest price) and Vickrey auctions (The winner pays the second highest price). Different auctions are suitable for different types of negotiations, e.g., English auctions are often used in real estate, Dutch auctions are often used in flower selling, and Vickrey auctions are favoured by economists as that Vickrey auctions are better at encouraging bidders to express their real estimation on the value of the items to bid on [Tre07].

This chapter is based on published work [FAST10]

4.2 Privacy and enforced privacy in e-auctions

Many security issues have been identified in e-auctions, such as, a bidder may falsely claim or forge bids, a seller may not deliver goods, the auctioneer may corrupt with other bidders [Tre05]. Beside security issues, an important problem with existing e-auction systems is privacy. The link between a bidder and his bids needs to be protected as such information can be used to target a bidder with unsolicited junk mails or other malicious purposes, e.g., *bid shielding*[1]. A major challenge of designing a protocol is to ensure the functionality of the protocol. In addition to that, a challenge for designing a privacy preserving e-auction protocol is that too much anonymity may allow bidders to repudiate bids: if nothing can be learned about the bids, the winning bidder can simply not claim to have won, and no one can identify him. On the other hand, insufficient anonymity may allow bidders to be profiled.

Depending on different types of auctions, privacy may have varying levels. For instance, in sealed-bid auctions, all bids are sealed until the winner is determined, thus, if auctioneers can decide the winners without knowing the non-winning bidder's bids, sealed-bid auctions can offer bidding-price secrecy for non-winning bidders; while in open-bid auctions, all the bids are published. Some auctions require that the auctioneer cannot link a bidder to his bids, whereas some others do not. The arguments of this are made according to the following lines. In Vickery auctions, a bidder's bid reflects the bidder's valuation of the item bid on. Knowing a bidder's bid, an auctioneer knows the bidder's valuation. Since the winning bidder pays the second highest price, the auctioneer could enter a bid just under the bidder's valuation, to increase the auction's revenue [Tre07]. Contrarily in English auctions, a bidder's previous bids reveal less information of the bidder's future bid, thus, that the auctioneer knows the link between a bidder and his previous bids is less harmful [Tre07]. In general, sealed-bid e-auctions require that the non-winning bidder's bidder-bid relation should be kept secret.

In addition to the above privacy notions, a stronger privacy notion – enforced privacy – has also been identified. In sealed-bid e-auctions, a bidder may be coerced to bid a low price, so that the coercer can win an auction with an unreasonably low price. The phenomenon that a coercer tries to control the winning price by coercion is call bid-rigging. Note that the traditional auctions do not suffer from bid-rigging, as the bidders do not have receipts on submitting a bid [HGP09]. Inspired by the requirement of receipt-freeness in e-voting that a voter should not be able to prove his vote to a voter-buyer, the requirement of receipt-freeness for fighting against bid-rigging has been identified [SM00].

In general, the following two privacy notions are required in sealed-bid e-auctions:

Bidding-price-secrecy for non-winning bidders: An e-auction protocol preserves bidding-price-secrecy for non-winning bidders if the adversary cannot determine the bidding price of any non-winning bidder.

[1] A dishonest bidder submits a higher price to deter other bidders with lower valuations, when it approaches the close time of the auction, the dishonest bidder withdraws his bid in order to win with another lower bid from him.

Receipt-freeness for non-winning bidders: An e-auction protocol is receipt-free for non-winning bidders if a non-winning bidder cannot prove how he bids to the adversary.

4.3 Formalisation of privacy notions

We formalise the above two privacy notions using the applied pi calculus in the context of sealed-bid e-auctions. An e-auction protocol normally involves two roles: bidders and auctioneers. An e-auction protocol with n_b bidders and n_{au} auctioneers can be modelled as:

$$P_{bid} := \nu chandata.(P_K \mid P_{b1} \mid \cdots \mid P_{b n_b} \mid P_{a1} \mid \cdots \mid P_{a n_{au}}),$$

where P_{bi} is an instance of a bidder process, P_{aj} is an instance of an auctioneer process, P_K is the key distribution process, and *chandata* models private data and private channels. The corresponding context in where two bidder processes are replaced with a hole is formalised as:

$$\mathcal{C}_b[\_] := \nu chandata.(P_K \mid P_{b1} \mid \cdots \mid P_{b n_b - 2} \mid \_ \mid P_{a1} \mid \cdots \mid P_{a n_{au}}).$$

4.3.1 Bidding-price-secrecy

Bidding-price-secrecy for non-winning bidders can be formalised in two levels: standard bidding-price-secrecy and strong bidding-price-secrecy. Standard bidding-price-secrecy is formalised as the adversary cannot derive the bidding price of a non-winning bidder. Strong bidding-price-secrecy is formalised as the adversary cannot even distinguish between the case when a bidder bids for price a and the case when the bidder bids for price c. In other words, the adversary cannot tell whether a bidder changes his bidding price from a to c.

Formalisation similar to strong bidding-price-secrecy has been used, e.g., vote-privacy [DKR09]: a process in which voter v_A votes for a ($P_{vA}\{a/vote\}$) and voter v_B votes for c ($P_{vB}\{c/vote\}$) is observationally equivalent to a process where v_A votes for c ($P_{vA}\{c/vote\}$) and v_B votes for a ($P_{vB}\{a/vote\}$). The idea is that even if all other voters reveal how they voted, the adversary cannot deduce the votes of voter v_A and voter v_B, given voter v_A and voter v_B counterbalance each other. Different from privacy in voting where the voting result is published, in sealed-bid e-auction protocols, normally a non-winning bidder's bidding price is not published. Therefore, we do not need a counterbalancing process. Instead, we need a process in which a bidder bids for a higher price so that non-winning bids are not revealed in the opening phase. Therefore, strong bidding-price-secrecy is formalised as follows:

Definition 4.1 (strong bidding-price-secrecy for non-winning bidders)**.** *An auction protocol P_{bid}, with a bidder sub-process represented as P_b, satisfies strong bidding-price-secrecy for non-winning bidders, if for all possible bidders b_A and b_B we have:*

$$\mathcal{C}_b[P_{bA}\{^a/_{p_b}\} \mid P_{bB}\{^d/_{p_b}\}] \approx_\ell \mathcal{C}_b[P_{bA}\{^c/_{p_b}\} \mid P_{bB}\{^d/_{p_b}\}]$$

with $a < d$ and $c < d$.

The context $\mathcal{C}_b[\_]$ is used to capture the assumption made on the checked protocol, usually it includes the other honest participants in the protocol. The process P_{bA} is a non-winning bidder process executed by bidder b_A. The process P_{bB} is a bidder process in which the bidder b_B bids for a higher price d. The intuition is that the adversary cannot determine whether a non-winning bidder bids for price a or price c, provided there exists another bidder who bids for a higher price d.

4.3.2 Receipt-freeness

Similar to receipt-freeness in e-voting, when modelling receipt-freeness for non-winning bidders in e-auctions, we also need to model the situation in which a bidder wants to provide his secret information to the adversary. We use the definition in e-voting (Definition 3.15) directly in our model. Intuitively, a bidder, who is willing to share information with the adversary, sends any input of base type, any freshly generated names of base type to the adversary through a fresh public channel chc. Note that public channels are under the adversary's control.

Now, we can define receipt-freeness for sealed-bid e-auction protocols. Again, we need a bidder process P_{bB} in which bidder b_B bids for a higher price d, so that non-winning bids are not revealed. Intuitively, if a non-winning bidder has a strategy to cheat the adversary, and the adversary cannot tell the difference between whether the bidder cheats or not, then the protocol is receipt-free.

Definition 4.2 (receipt-freeness for non-winning bidders). *An auction protocol P_{bid}, with a bidder sub-process P_b, satisfies receipt-freeness for non-winning bidders, if there exists a closed plain process P_f such that:*

1. $P_f^{\backslash out(\mathtt{chc},\cdot)} \approx_\ell P_{bA}\{^c/_{p_b}\}$,

2. $\mathcal{C}_b[P_{bA}\{^a/_{p_b}\}^{\mathtt{chc}} \mid P_{bB}\{^d/_{p_b}\}] \approx_\ell \mathcal{C}_b[P_f \mid P_{bB}\{^d/_{p_b}\}]$

with $a < d$ and $c < d$.

Process P_f is a bidder process in which bidder b_A bids for price c but communicates with the adversary and tells the adversary he bids for price a. Process $P_{bA}\{^c/_{p_b}\}$ is a bidder process in which bidder b_A bids for price c. Process $P_{bA}\{^a/_{p_b}\}^{chc}$ is a bidder process in which bidder b_A bids for price a and shares his secret with the adversary. Process P_{bB} is a bidder process in which bidder b_B bids for a higher price d. The first equivalence says that ignoring the outputs bidder b_A makes on the channel chc to the adversary, P_f looks like a normal process in which b_A bids for price c. The second equivalence says that the adversary cannot tell the difference between the situation in which b_A obeys the adversary's commands and bids for price a, and the situation in which b_A pretends to cooperate but actually bids for price c, provided there is a bidding process P_{bB} that bids higher, ensuring that bidding processes P_{bA} and P_f are not winners. Receipt-freeness is a stronger property than bidding-price-secrecy, for the same reason as receipt-freeness in e-voting is stronger than vote-privacy (as shown [DKR09]).

4.4 Case study: the AS02 protocol

After receipt-freeness has been identified in sealed-bid e-auctions. Abe and Suzuki proposed the first protocol which aims to prevent bid-rigging – the AS02 protocol [AS02]. In this section, we analyse both *bidding-price-secrecy* and *receipt-freeness* for non-winning bidders in the AS02 protocol. ProVerif code is available at [1].

4.4.1 Introduction

This protocol is a sealed-bid e-auction protocol. The protocol involves n bidders $\mathsf{b}_1, \ldots, \mathsf{b}_n$ and k auctioneers $\mathsf{a}_1, \ldots, \mathsf{a}_k$. The list of all prices allowed for bids is published before the protocol. During the protocol, each bidder sends a commit for *every* price in the price list: a 'bid' if he wants to bid that price, and a 'non-bid' otherwise. Auctioneers work together to open the commitments of all bidders from the highest price down until the winning bid(s) is/are found.[2]

4.4.2 Physical assumptions

In order to ensure privacy of bidders, the protocol has two physical assumptions:

- **a1:** a bidding booth for the bidders, and
- **a2:** a one-way untappable channel from every bidder to every auctioneer.

The bidding booth enables a bidder to privately submit a bid free from control or observation of the adversary. The untappable channels ensure no adversary can see messages sent.

4.4.3 Settings

Before starting the protocol, one auctioneer publishes an increasing price list $\mathsf{p}_1, \ldots, \mathsf{p}_m$, a message M_{yes} for "I bid", a message M_{no} for "I do not bid", a generator g of subgroup of $\mathbb{Z}_p^*$ with order q, where q, p are large primes with $p = 2q + 1$.

4.4.4 Description of the protocol

The protocol consists of two phases: bidding and opening.

Bidding phase. A bidder in the bidding booth chooses a secret key x, publishes his public key $h = g^x$ with a predetermined signature. Then the bidder chooses a series of random numbers $r_1, \ldots, r_m$ as secret seeds, one random number for each price, and decides a price p_b to bid for. Then he generates either a "bid" or

[2] The protocol does not specify how to resolve the case where there are fewer bidding items than winners.

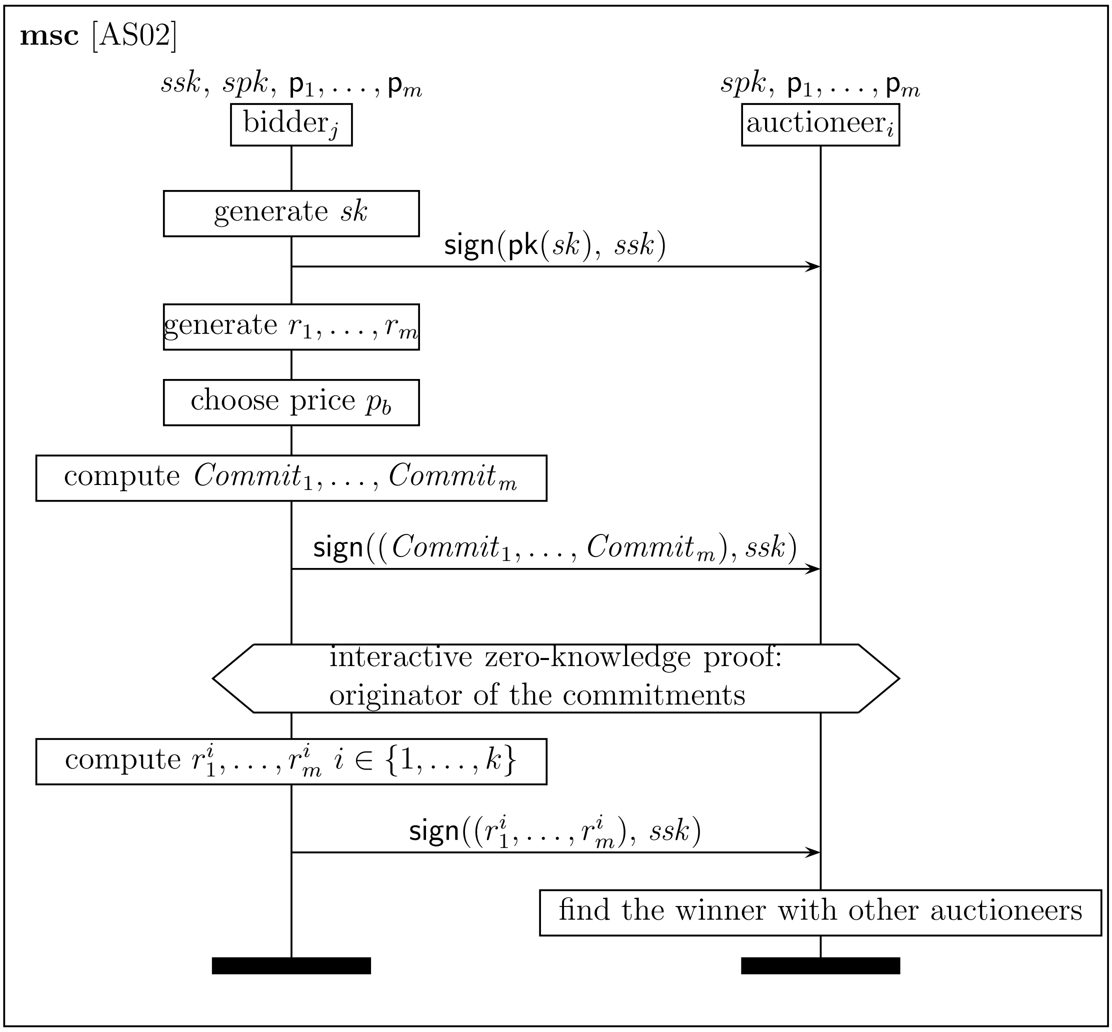
msc [AS02]
ssk, spk, $\mathsf{p}_1, \ldots, \mathsf{p}_m$
bidder$_j$
spk, $\mathsf{p}_1, \ldots, \mathsf{p}_m$
auctioneer$_i$
generate sk
$\mathsf{sign}(\mathsf{pk}(sk), ssk)$
generate $r_1, \ldots, r_m$
choose price p_b
compute $Commit_1, \ldots, Commit_m$
$\mathsf{sign}((Commit_1, \ldots, Commit_m), ssk)$
interactive zero-knowledge proof: originator of the commitments
compute $r_1^i, \ldots, r_m^i$ $i \in \{1, \ldots, k\}$
$\mathsf{sign}((r_1^i, \ldots, r_m^i), ssk)$
find the winner with other auctioneers

Figure 4.1: The AS02 protocol.

a “non-bid” bit-commitment for each price p_ℓ $(1 \leq \ell \leq m)$, using the following formula:

$$Commit_\ell = \begin{cases} g^{M_{yes}} h^{r_\ell} & \text{if } \mathsf{p}_\ell = p_b \quad \text{(a bid for price } \mathsf{p}_\ell\text{)} \\ g^{M_{no}} h^{r_\ell} & \text{if } \mathsf{p}_\ell \neq p_b \quad \text{(a non-bid for price } \mathsf{p}_\ell\text{)} \end{cases}$$

Next, the bidder publishes the sequence of the bit-commitments with his signature. Then he proves to each auctioneer that he knows the secret key $\log_g h = x$ and the discrete logs $(\log_g Commit_1, \ldots, \log_g Commit_m)$ using interactive zero-knowledge proofs. Finally, he computes t-out-of-k[3] secret shares r_ℓ^i for each secret seed r_ℓ and each auctioneer a_i, and then sends the signed secret share r_ℓ^i over the one-way untappable channel to the auctioneer a_i (see Figure 4.1).

Opening phase. Auctioneers together iterate the following steps for each price $\mathsf{p}_\ell = \mathsf{p}_m, \mathsf{p}_{m-1}, \ldots, \mathsf{p}_1$ until the winning bid is determined.

Each auctioneer a_i publishes secret shares r_ℓ^i (the ℓ-th secret share of a bidder sent to auctioneer a_i) of all bidders. For each bidder, all auctioneers work together to

[3] t is a threshold, k is the number of auctioneers, it means only more than t auctioneers together can reconstruct the secret seeds.

reconstruct the secret seed r_ℓ, and check for each bidder whether

$$Commit_\ell \overset{?}{=} g^{M_{yes}} h^{r_\ell}.$$

If there exist some bidders for which the above equivalences are satisfied, the auctioneers finish checking the current price and then stop. In this case, the price p_ℓ is the winning price, those bidders are winning bidders. If there is no equivalence existing, which means there is no bidder bidding for the price p_ℓ, the auctioneers repeat the above process on the next lower price.

4.4.5 Claimed privacy properties

The authors claim the following properties: bidding-price-secrecy and receipt-freeness for non-winning bidders. Intuitively, the bidding price of each bidder is sealed in the bidding phase, and only the winning bidder's bidding price is revealed in the opening phase, thus the adversary does not know the bidding price for non-winning bidders, thus standard bidding-price-secrecy is satisfied. The strong bidding-price-secrecy is satisfied mainly due to the random number used in calculating the bit-commitments.

Informal reasoning of receipt-freeness. We use M to represent either M_{yes} or M_{no}, the formula for computing $Commit_\ell$ is of the following form:

$$Commit_\ell = g^M \cdot h^{r_\ell} = g^M \cdot (g^x)^{r_\ell} = g^{M+xr_\ell} ,$$

since $h = g^x$. Thus, $\log Commit_\ell = M + xr_\ell$. By using interactive zero-knowledge proofs, a bidder is proved to know his secret key x and discrete logs $\log Commit_\ell$. An interesting property of chameleon bit-commitments is that if the bidder bids for price p_ℓ,

$$\log Commit_\ell = M_{yes} + xr_\ell$$

he can calculate a fake r'_ℓ such that:

$$\log Commit_\ell = M_{no} + xr'_\ell \quad \text{and} \quad r'_\ell = (M_{yes} + xr_\ell - M_{no})/x.$$

Using the fake r'_ℓ, the bidder can show that the bit-commitment $Commit_\ell$ is opened as message M_{no}, which means bidder did not bid for price p_ℓ. Using the same method, a bidder can open a 'no' bit-commitment as a 'yes' bit-commitment. Thus, the commit leaks no information concerning the bid, thus the bidder cannot prove how he bid, e.g. receipt-freeness is satisfied.

4.5 Modelling AS02

We use the applied pi calculus to model the AS02 protocol. We use two simplifications:

- **s1:** one honest auctioneer; and
- **s2:** perfect zero knowledge proofs.

In the protocol, auctioneers are cooperating to find the winning bid. It takes at least t auctioneers to decide the winner, thus guaranteeing t-out-of-k secrecy. As we focus on bidder privacy, we need to consider only one honest auctioneer. Thus, we simplify the model to have only one honest auctioneer. The AS02 protocol uses interactive zero knowledge proofs to guarantee that each bidder knows his secret key and the discrete logs of bit-commitments. However, the details of these proofs are left unspecified, and thus we did not include them in the model. We simply assume that each bidder knows his secret key and discrete logs of bit-commitments.

In addition, the AS02 does not specify how the auctioneers tell the signed public key from the signed commitments generated by the same bidder. In order for the auctioneer to distinguish the two messages, in our modelling,

- **s3:** we use a symbol k in the signed public key messages.

Signature and equational theory. The signatures and the equational theory model cryptographic primitives used in the protocol. We fix a list of bidders $(\mathsf{b}_1, \ldots, \mathsf{b}_n)$ and an ordered list of prices $(\mathsf{p}_1, \ldots, \mathsf{p}_m)$, which are modelled as functions with arity 0. We define function nextbidder to find the next bidder in the bidder list, and function nextprice to find the next lower price in the price list. Function checksign is used to check whether the public signature key is the right one for the signed message, and we use function getmsg to get the original message from a signed message. Particularly, chameleon bit-commitments are modelled as a function commit with arity 3 (a random number, public key of the bidder and message M either M_{yes} or M_{no}). The relevant properties of chameleon bit-commitments are captured in the following equational theory.

$$\begin{aligned}
\mathsf{commit}(r, \mathsf{pk}(sk_b), \mathsf{M}_{yes}) &=_E \mathsf{commit}(\mathsf{f}(r), \mathsf{pk}(sk_b), \mathsf{M}_{no}) \\
\mathsf{commit}(r, \mathsf{pk}(sk_b), \mathsf{M}_{no}) &=_E \mathsf{commit}(\mathsf{f}(r), \mathsf{pk}(sk_b), \mathsf{M}_{yes}) \\
\mathsf{open}(\mathsf{commit}(r, pk, m), r, pk) &=_E m
\end{aligned}$$

Constants M_{no} and M_{yes} represent “I do not bid” and “I bid”, respectively. The parameter $\mathsf{pk}(sk_b)$ is the public key of a bidder, and r is the secret seed the bidder chooses. Function $\mathsf{f}(r)$ returns the fake secret seed of a secret seed r. We can model the function f by just giving one parameter - the real secret seed. Because we assume that each bidder knows his secret key and discrete logs of bit-commitments, he can compute the fake secret seed for each real secret seed, as explained in the previous section. The first equivalence means that if a bidder chooses a secret seed r, bids for a price, and calculates the bit-commitment $\mathsf{commit}(r, \mathsf{pk}(sk_b), \mathsf{M}_{yes})$, he can compute a fake secret seed $\mathsf{f}(r)$, and by using this fake secret seed, the bit-commitment can be opened as message M_{no}, which means “I do not bid”. The second equivalence shows that the opposite situation also holds. A bidder can also open a bit-commitment as if he bids for that price, when actually he does not. All functions defined in this model are shown in Figure 4.2 and the equational theory is shown in Figure 4.3. Recall that *fun* is used to denote function in ProVerif, and *reduc* and *equation* are used to denote the equational theory in ProVerif.

Main process. For each bidder b_j, the main process (see Figure 4.4) generates two private channels $\mathtt{privch}_{\mathsf{b}_j}$ (**m1**) and $\mathtt{privcha}_{\mathsf{b}_j}$ (**m2**). These channel are used for

$$\begin{array}{l}
\mathit{fun}\ \mathsf{b}_1/0, \ldots, \mathit{fun}\ \mathsf{b}_n/0, \mathit{fun}\ \mathsf{p}_1/0, \ldots, \mathit{fun}\ \mathsf{p}_m/0, \mathit{fun}\ \mathsf{M}_{yes}/0, \mathit{fun}\ \mathsf{M}_{no}/0, \\
\mathit{fun}\ \mathsf{true}/0, \mathit{fun}\ \mathsf{pk}/1, \mathit{fun}\ \mathsf{commit}/3, \mathit{fun}\ \mathsf{sign}/2, \mathit{fun}\ \mathsf{f}/1, \mathit{fun}\ \mathsf{k}/0.
\end{array}$$

Figure 4.2: Functions.

$$\begin{array}{lrl}
\mathit{reduc} & \mathsf{nextbidder}(\mathsf{b}_1) & = \mathsf{b}_2. \\
\ldots & & \\
\mathit{reduc} & \mathsf{nextbidder}(\mathsf{b}_{n-1}) & = \mathsf{b}_n. \\
\mathit{reduc} & \mathsf{nextbidder}(\mathsf{b}_n) & = \bot. \\
\mathit{reduc} & \mathsf{nextprice}(\mathsf{p}_m) & = \mathsf{p}_{m-1}. \\
\ldots & & \\
\mathit{reduc} & \mathsf{nextprice}(\mathsf{p}_2) & = \mathsf{p}_1. \\
\mathit{reduc} & \mathsf{nextprice}(\mathsf{p}_1) & = \top. \\
\mathit{reduc} & \mathsf{checksign}(\mathsf{pk}(\mathit{sk}), \mathsf{sign}(m, \mathit{sk})) & = \mathsf{true}. \\
\mathit{reduc} & \mathsf{getmsg}(\mathsf{sign}(m, \mathit{sk})) & = m. \\
\mathit{equation} & \mathsf{commit}(r, \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{yes}) & = \mathsf{commit}(\mathsf{f}(r), \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{no}). \\
\mathit{equation} & \mathsf{commit}(r, \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{no}) & = \mathsf{commit}(\mathsf{f}(r), \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{yes}). \\
\mathit{reduc} & \mathsf{open}(\mathsf{commit}(r, \mathit{pk}, m), r, \mathit{pk}) & = m
\end{array}$$

Figure 4.3: Equational theory.

instantiating a bidder process. In particular, a bidder receives his secret signing key from channel $\mathtt{privch}_{\mathsf{b}_j}$; and the auctioneer receives the corresponding public key from channel $\mathtt{privcha}_{\mathsf{b}_j}$. In addition, the main process generates an untappable channel $\mathtt{untapch}_{\mathsf{b}_j}$ for bidders b_j (**m3**). The untappable channel is shared between each bidder and the auctioneer. The private channels $\mathtt{synch}_{\mathsf{b}_1}, \ldots, \mathtt{synch}_{\mathsf{b}_n}$ are generated for modelling convenience (**m4**). These channels are used by the auctioneer to collect all necessary information before moving to the opening phase. The main process launches a key generating process P_K (**m5**), n instantiations of the bidder process (**m5-m8**) and an instance of the auctioneer process (**m8**). Four variables need to be instantiated in an instance of bidder process: the bidding price p_b, the untappable channel *untapch*, the private channel *privch* and the public channel for that bidder *ch*. Note that $p_{\mathsf{b}_1}, \ldots, p_{\mathsf{b}_n}$ are parameters, each of these parameters has to be instantiated with a constant in the published price list $\mathsf{p}_1, \ldots, \mathsf{p}_m$. For the simplicity of modelling, each bidder b_j has a distinct public channel $\mathtt{ch}_{\mathsf{b}_j}$. The correspondence between $\mathtt{privcha}_{\mathsf{b}_j}$, $\mathtt{untapch}_{\mathsf{b}_j}$ and $\mathtt{ch}_{\mathsf{b}_j}$ allows the auctioneer to distinguish messages from the same bidder. In this way, we avoid modelling the auctioneer classifying messages by bidders (by checking signatures).

Key distribution process. This process generates and distributes keying material modelling a PKI – public key infrastructure (Figure 4.5). This process first generates n secret keys (**k1**). Each bidder b_j has one secret key $\mathit{ssk}_{\mathsf{b}_j}$ for signing messages. Each secret key corresponds to a public key (**k2-k4**). Each secret key is assigned to a bidder process by being sent to the bidder over the private channel $\mathtt{privch}_{\mathsf{b}_j}$ for that bidder (**k5**). The corresponding public key is sent to the auctioneer over the private channel $\mathtt{privcha}_{\mathsf{b}_j}$ (**k6**) and is published over the pub-

```
     $P_{AS02} :=$
m1.     $\nu \mathtt{privch}_{\mathtt{b}_1}.\nu \mathtt{privch}_{\mathtt{b}_2}.\cdots.\nu \mathtt{privch}_{\mathtt{b}_n}.$
m2.     $\nu \mathtt{privcha}_{\mathtt{b}_1}.\nu \mathtt{privcha}_{\mathtt{b}_2}.\cdots.\nu \mathtt{privcha}_{\mathtt{b}_n}.$
m3.     $\nu \mathtt{untapch}_{\mathtt{b}_1}.\nu \mathtt{untapch}_{\mathtt{b}_2}.\cdots.\nu \mathtt{untapch}_{\mathtt{b}_n}.$
m4.     $\nu \mathtt{synch}_{\mathtt{b}_1}.\nu \mathtt{synch}_{\mathtt{b}_2}.\cdots.\nu \mathtt{synch}_{\mathtt{b}_n}.$
m5.     $(P_K \mid (\mathtt{let}\ p_b = p_{\mathtt{b}_1}\ \mathtt{in}\ \mathtt{let}\ \mathit{untapch} = \mathtt{untapch}_{\mathtt{b}_1}\ \mathtt{in}$
m6.         $\mathtt{let}\ \mathit{privch} = \mathtt{privch}_{\mathtt{b}_1}\ \mathtt{in}\ \mathtt{let}\ \mathit{ch} = \mathtt{ch}_{\mathtt{b}_1}\ \mathtt{in}\ P_b) \mid$
m7.      $\cdots \mid (\mathtt{let}\ p_b = p_{\mathtt{b}_n}\ \mathtt{in}\ \mathtt{let}\ \mathit{untapch} = \mathtt{untapch}_{\mathtt{b}_n}\ \mathtt{in}$
m8.         $\mathtt{let}\ \mathit{privch} = \mathtt{privch}_{\mathtt{b}_n}\ \mathtt{in}\ \mathtt{let}\ \mathit{ch} = \mathtt{ch}_{\mathtt{b}_n}\ \mathtt{in}\ P_b) \mid P_a)$
```

Figure 4.4: The main process.

lic channel $\mathtt{ch}_{\mathtt{b}_j}$ such that the adversary knows the keys (**k7**). Therefore, only a bidder knows his own secret key, and everyone including the adversary knows each bidder's public key. Sending each public key to the auctioneer over a private channel, models the following protocol setting: There are fix number of bidders in sealed-bid auctions, and the auctioneer knows each bidder's public signing key as predetermined knowledge.

```
     $P_K :=$
k1.     $\nu \mathtt{ssk}_{\mathtt{b}_1}.\nu \mathtt{ssk}_{\mathtt{b}_2}.\cdots.\nu \mathtt{ssk}_{\mathtt{b}_n}.$
k2.     $\mathtt{let}\ \mathit{spk}_{\mathtt{b}_1} = \mathsf{pk}(\mathtt{ssk}_{\mathtt{b}_1})\ \mathtt{in}$
k3.     $\cdots$
k4.     $\mathtt{let}\ \mathit{spk}_{\mathtt{b}_n} = \mathsf{pk}(\mathtt{ssk}_{\mathtt{b}_n})\ \mathtt{in}$
k5.     $(\mathsf{out}(\mathtt{privch}_{\mathtt{b}_1}, \mathtt{ssk}_{\mathtt{b}_1}) \mid \cdots \mid \mathsf{out}(\mathtt{privch}_{\mathtt{b}_n}, \mathtt{ssk}_{\mathtt{b}_n}) \mid$
k6.      $\mathsf{out}(\mathtt{privcha}_{\mathtt{b}_1}, \mathit{spk}_{\mathtt{b}_1}) \mid \cdots \mid \mathsf{out}(\mathtt{privcha}_{\mathtt{b}_n}, \mathit{spk}_{\mathtt{b}_n}) \mid$
k7.      $\mathsf{out}(\mathtt{ch}_{\mathtt{b}_1}, \mathit{spk}_{\mathtt{b}_1}) \mid \cdots \mid \mathsf{out}(\mathtt{ch}_{\mathtt{b}_n}, \mathit{spk}_{\mathtt{b}_n}))$
```

Figure 4.5: The key distribution process.

Bidder process. The applied pi calculus process for a bidder P_b is given in Figure 4.6. First, a bidder receives his secret signature key from his private channel (**b1**). Next, the bidder generates his secret key $\mathtt{sk}_b$, signs the corresponding public key and publishes the signed message (**b2**). To indicate that this message contains a key, we add k into the message (see **s3**). In addition, the bidder chooses a series of random numbers $\mathtt{r}_1, \ldots, \mathtt{r}_m$ as secret seeds (**b3**). The bidder then computes each bit-commitment $\mathit{cmt}^{\mathsf{p}_\ell}$ as described in Section 4.4.4. For each price, the bidder computes a commitment: if the price is the bidding price, then the bidder commits 'yes' with M_{yes}, otherwise, the bidder commits 'no' with M_{no} (**b4-b9**). Finally, the bidder publishes the series of bit-commitments $\mathit{cmt}^{\mathsf{p}_1}, \ldots, \mathit{cmt}^{\mathsf{p}_m}$ with his signature (**b10**), and sends the signed series of secret seeds to the auctioneer through the untappable channel (**b11**). As we assume there is only one honest auctioneer in the model, we do not need to model secret shares.

Auctioneer process. During the bidding phase, the auctioneer launches n copies of sub-process *readinfo* to gather information from each bidder b_j (**a1**).

```
        $P_b :=$
b1.         $\mathsf{in}(privch, ssk_b)$.
b2.         $\nu\mathtt{sk}_b.\mathsf{out}(\mathtt{ch}, \mathsf{sign}((\mathsf{pk}(\mathtt{sk}_b), \mathsf{k}), ssk_b))$.
b3.         $\nu\mathtt{r}_1. \cdots .\nu\mathtt{r}_m$.
b4.         if $\mathsf{p}_1 = p_b$
b5.         then let $cmt^{\mathsf{p}_1} = \mathsf{commit}(\mathtt{r}_1, \mathsf{pk}(\mathtt{sk}_b), \mathsf{M}_{yes})$ in
b6.         else let $cmt^{\mathsf{p}_1} = \mathsf{commit}(\mathtt{r}_1, \mathsf{pk}(\mathtt{sk}_b), \mathsf{M}_{no})$ in
            ...
b7.         if $\mathsf{p}_m = p_b$
b8.         then let $cmt^{\mathsf{p}_m} = \mathsf{commit}(\mathtt{r}_m, \mathsf{pk}(\mathtt{sk}_b), \mathsf{M}_{yes})$ in
b9.         else let $cmt^{\mathsf{p}_m} = \mathsf{commit}(\mathtt{r}_m, \mathsf{pk}(\mathtt{sk}_b), \mathsf{M}_{no})$ in
b10.        $\mathsf{out}(\mathtt{ch}, \mathsf{sign}((cmt^{\mathsf{p}_1}, \cdots, cmt^{\mathsf{p}_m}), ssk_b))$.
b11.        $\mathsf{out}(untapch, \mathsf{sign}((\mathtt{r}_1, \cdots, \mathtt{r}_m), ssk_b))$
```

Figure 4.6: The bidder process.

In details, the auctioneer collects public signature key spk_{b_j} (**r1**) and the signed committing public key *signedpk* (supposed to be $\mathsf{sign}((\mathsf{pk}(sk_{b_j}), \mathsf{k}), ssk_{b_j}))$ (**r2**). The auctioneer verifies whether the committing public key is signed with the right signature (**r3**) and obtain the committing public key from *signedpk* (**r4**). Next, the auctioneer reads in the signed commitments $signedcommit_{\mathsf{b}_j}$ of bidder b_j (**r5**) and verifies the signature (**r6**). If the commitments are correctly signed, the auctioneer obtains the series of bit-commitments $cmt_{\mathsf{b}_j}^{\mathsf{p}_1}, \ldots, cmt_{\mathsf{b}_j}^{\mathsf{p}_m}$ (**r7**), then the auctioneer reads in the signed secret seeds *sr* from the untappable channel of bidder b_j (**r8**). The auctioneer verifies the signature (**r9**). If the secret seeds are correctly signed, the auctioneer obtains the secret seeds $ss_{\mathsf{b}_j}^{\mathsf{p}_1}, \ldots, ss_{\mathsf{b}_j}^{\mathsf{p}_m}$ (**r10**). Finally, the auctioneer sends the signal that information collecting for bidder b_j has finished, over the channel *synch* (**r9**). In addition, the collected information (the committing public key, the commitments, the secret seeds) is sent to the sub-process in which the winning bidder is determined.

Next the auctioneer needs to synchronise with all bidders (**a2**). The auctioneer process is not allowed to continue until all bidders reach the end of the bidding phase. In the opening phase, the auctioneer evaluates $cmt_{\mathsf{b}_j}^{\mathsf{p}_m} \stackrel{?}{=} \mathsf{commit}(ss_{\mathsf{b}_j}^{\mathsf{p}_m}, pk_{\mathsf{b}_j}, \mathsf{M}_{yes})$ for each bidder (**a3, a7, a12**). If the two values are equivalent for the first bidder b_1 (**a3**), bidder b_1 has bid for that price, otherwise, bidder b_1 has not bid for that price. When bidder b_1 has bid for that price, the auctioneer publishes the bidder together with the price over the public channel *winnerch* (**a4**), then the auctioneer checks the evaluation for the next bidder (if exists) (**a7**). Once the auctioneer has evaluated for every bidder (**a5**) and has determined the set of winning bidders (**a4**), he stops the process (**a6**). When bidder b_1 has not bid for that price, the auctioneer checks the evaluation for the next bidder (if exists) (**a12**). Once the auctioneer has evaluated for every bidder and no winner has been found (**a8**), the auctioneer repeats the evaluation steps for each bidder at the next lower price (**a11**). If the next lower price does not exist (**a9**), the process stops (**a10**) and no bidder has bid for any price. In a similar way, the sub-process $checknextb_{\mathsf{b}_i}^{\mathsf{p}_j}$ is used to evaluate the bid of a bidder b_i at price p_j, if there are already some winners before bidder b_i. And the sub-process $checknextbnp_{\mathsf{b}_i}^{\mathsf{p}_j}$ is used to check the next

```
     $P_a :=$
a1.       let $ch = \mathsf{ch}_{\mathsf{b}_1}$ in let $privcha = \mathtt{privcha}_{\mathsf{b}_1}$ in
          let $synch = \mathsf{synch}_{\mathsf{b}_1}$ in let $untapch = \mathtt{untapch}_{\mathsf{b}_1}$ in $readinfo$ |
          $\cdots$ |
          let let $ch = \mathsf{ch}_{\mathsf{b}_n}$ in let $privcha = \mathtt{privcha}_{\mathsf{b}_n}$ in
          let let $synch = \mathsf{synch}_{\mathsf{b}_n}$ in $untapch = \mathtt{untapch}_{\mathsf{b}_n}$ in $readinfo$ |
a2.       $\mathsf{in}(\mathsf{synch}_{\mathsf{b}_1}, (pk_{\mathsf{b}_1}, cmt^{\mathsf{p}_1}_{\mathsf{b}_1}, \ldots, cmt^{\mathsf{p}_m}_{\mathsf{b}_1}, ss^{\mathsf{p}_1}_{\mathsf{b}_1}, \ldots, ss^{\mathsf{p}_m}_{\mathsf{b}_1}))$.
          $\cdots$ .
          $\mathsf{in}(\mathsf{synch}_{\mathsf{b}_n}, (pk_{\mathsf{b}_n}, cmt^{\mathsf{p}_1}_{\mathsf{b}_n}, \ldots, cmt^{\mathsf{p}_m}_{\mathsf{b}_n}, ss^{\mathsf{p}_1}_{\mathsf{b}_n}, \ldots, ss^{\mathsf{p}_m}_{\mathsf{b}_n}))$.
a3.       if $cmt^{\mathsf{p}_m}_{\mathsf{b}_1} = \mathsf{commit}(ss^{\mathsf{p}_m}_{\mathsf{b}_1}, pk_{\mathsf{b}_1}, \mathsf{M}_{yes})$
a4.       then $\mathsf{out}(\mathtt{winnerch}, (\mathsf{p}_m, \mathsf{b}_1))$.
a5.           if $\mathsf{nextbidder}(\mathsf{b}_1) = \bot$
a6.           then 0
a7.           else $checknextb^{\mathsf{p}_m}_{\mathsf{nextbidder}(\mathsf{b}_1)}$
a8.       else if $\mathsf{nextbidder}(\mathsf{b}_1) = \bot$
a9.          then if $\mathsf{nextprice}(\mathsf{p}_m) = \top$
a10.              then 0
a11.              else $checknextbnp^{\mathsf{nextprice}(\mathsf{p}_m)}_{\mathsf{b}_1}$
a12.         else $checknextbnp^{\mathsf{p}_m}_{\mathsf{nextbidder}(\mathsf{b}_1)}$
```

Figure 4.7: The auctioneer process.

bidder at price p_j, if there is no winner before that bidder. We use $\bot$ and $\top$ to represent the end of the bidder list and price list, respectively.

In the sub-process $checknextb^{\mathsf{p}_j}_{\mathsf{b}_i}$, the auctioneer checks whether the bidder b_i has bid for price p_j (**n1**). If the bidder b_i has bid for p_j, b_i is a winning bidder. The auctioneer publishes the winning bidder b_i and the winning price p_j (**n2**). Note that since there already exists one or more winning bidders, b_i is not the first winner. The auctioneer checks whether the bidder b_i is the last bidder (**n3**). If b_i is the last bidder, the auctioneer has found all winning bidders, thus stops the opening process (**n4**); otherwise, the auctioneer checks the evaluation for the next bidder at the same price (i.e., whether the next bidder is also a winner) (**n5**).

In the sub-process $checknextbnp^{\mathsf{p}_j}_{\mathsf{b}_i}$, the auctioneer first checks whether the bidder b_i has bid for price p_j (**p1**). If the bidder b_i has bid for p_j, b_i is a winner. The auctioneer publishes the bidder b_i and the winning price p_j (**p2**). Since there is no winning bidder found before, b_i is the first winner. Then the auctioneer checks whether the bidder b_i is the last bidder (**p3**). If b_i is the last bidder, bidder b_i is the only winner. Since the auctioneer has found all winners, he stops the opening process (**p4**). Otherwise, the auctioneer checks whether the next bidder is also a winner (**p5**). Note that since there is already a winner b_i, the auctioneer use the process $checknextb^{\mathsf{p}_j}_{\mathsf{nextbidder}(\mathsf{b}_i)}$. If the bidder b_i has not bid for p_j, the auctioneer checks whether the bidder is the last bidder (**p6**). If b_i is the last bidder, since there is no bidder bid for price p_j before b_i and b_i has not bid for p_j, there is no bidder bid for price p_j. Thus, the auctioneer checks the evaluations for every bidder at the next lower price p_{j-1}. To do so, the auctioneer first checks whether p_{j-1} is the bottom (whether p_j is already the lowest price in the price list) (**p7**).

```
      $readinfo :=$
r1.        $\mathsf{in}(privcha, spk)$.
r2.        $\mathsf{in}(ch, signedpk)$.
r3.        if $\mathsf{checksign}(signedpk, spk) = \mathsf{true}$
r4.        then let $(pk, = \mathsf{k}) = \mathsf{getmsg}(signedpk)$ in
r5.        $\mathsf{in}(ch, signedcommit)$.
r6.        if $\mathsf{checksign}(signedcommit, spk) = \mathsf{true}$
r7.        then let $(cmt_{\mathsf{b}_j}^{\mathsf{p}_1}, \ldots, cmt_{\mathsf{b}_j}^{\mathsf{p}_m}) = \mathsf{getmsg}(signedcommit)$ in
r8.        $\mathsf{in}(untapch, sr)$.
r9.        if $\mathsf{checksign}(sr, spk) = \mathsf{true}$
r10.       then let $(ss^{\mathsf{p}_1}, \ldots, ss^{\mathsf{p}_m}) = \mathsf{getmsg}(sr)$ in
r11.       $\mathsf{out}(synch, (pk, cmt_{\mathsf{b}_j}^{\mathsf{p}_1}, \ldots, cmt_{\mathsf{b}_j}^{\mathsf{p}_m}, ss^{\mathsf{p}_1}, \ldots, ss^{\mathsf{p}_m}))$
```

Figure 4.8: The process *readinfo*.

```
      $checknextb_{\mathsf{b}_i}^{\mathsf{p}_j} :=$
n1.        if $cmt_{\mathsf{b}_i}^{\mathsf{p}_j} = \mathsf{commit}(ss_{\mathsf{b}_i}^{\mathsf{p}_j}, pk_{\mathsf{b}_i}, \mathsf{M}_{yes})$
n2.        then $\mathsf{out}(\mathtt{winnerch}, (\mathsf{p}_j, \mathsf{b}_i))$.
n3.            if $\mathsf{nextbidder}(\mathsf{b}_i) = \perp$
n4.            then 0
n5.            else $checknextb_{\mathsf{nextbidder}(\mathsf{b}_i)}^{\mathsf{p}_j}$
```

Figure 4.9: The process $checknextb_{\mathsf{b}_i}^{\mathsf{p}_j}$.

If p_{j-1} is the bottom, since the auctioneer has not found a winner, there does not exist a winner. That is, the auctioneer has checked the evaluations for all bidders at all prices, and no one has bid for any price. Thus, the opening process stops (**p8**). If p_{j-1} is not the bottom, the auctioneer checks the evaluation for the first bidder at the next lower price p_{j-1}. Note that since b_1 is the first bidder checked for price p_{j-1}, there is no winning bidder found before, the process for checking b_1 is $checknextbnp_{\mathsf{b}_1}^{\mathsf{nextprice}(\mathsf{p}_j)}$ (**a9**). If b_i has not bid for p_j and b_i is not the last bidder, the auctioneer checks the evaluation for the next bidder at the same price (**p10**). Note that since there is no winning bid found, the process is $checknextbnp_{\mathsf{nextbidder}(\mathsf{b}_i)}^{\mathsf{p}_j}$.

4.6 Analysis of AS02

After modelling the protocol in the previous section, we formally analyse bidding-price-secrecy and receipt-freeness for bidders. In the AS02 protocol, the winning bid is published, and thus bidding-price-secrecy and receipt-freeness for the winning bidders are not satisfied. Particularly, if all bidders bid for the same price, then all bidders are winners, i.e., no bidder is a non-winning bidder, thus bidding-price-secrecy is not satisfied in this case. From here on, when we refer to bidding-price-secrecy and receipt-freeness, we mean only with respect to non-winning bidders.

```
$checknextbnp^{\mathsf{p}_j}_{\mathsf{b}_i}$ :=
        if $cmt^{\mathsf{p}_j}_{\mathsf{b}_i} = \mathsf{commit}(ss^{\mathsf{p}_j}_{\mathsf{b}_i}, pk_{\mathsf{b}_i}, \mathsf{M}_{yes})$
        then out(winnerch, $(\mathsf{p}_j, \mathsf{b}_i)$).
            if $\mathsf{nextbidder}(\mathsf{b}_i) = \bot$
            then $0$
            else $checknextb^{\mathsf{p}_j}_{\mathsf{nextbidder}(\mathsf{b}_i)}$
        else if $\mathsf{nextbidder}(\mathsf{b}_i) = \bot$
           then if $\mathsf{nextprice}(\mathsf{p}_j) = \top$
               then $0$
               else $checknextbnp^{\mathsf{nextprice}(\mathsf{p}_j)}_{\mathsf{b}_1}$
           else $checknextbnp^{\mathsf{p}_j}_{\mathsf{nextbidder}(\mathsf{b}_i)}$
```

Figure 4.10: The process $checknextbnp^{\mathsf{p}_j}_{\mathsf{b}_i}$.

4.6.1 Bidding-price-secrecy

In general, bidding-price-secrecy can be formalised in two levels: standard bidding-price-secrecy and strong bidding-price-secrecy. Standard bidding-price-secrecy is defined as no matter how the adversary interacts with the protocol, he cannot derive a non-winning bidder's bidding price. Thus, it aims to keep the price secret. However, since the AS02 protocol publishes the bidding price list, the adversary initially knows all the prices. No matter which price a bidder bids for, the bidding price is not a secret to the adversary. Therefore, a bidder's bidding price is not a secret. In fact, what the AS02 protocol aims to protect is the link between bidders and the price he bid, instead of the price itself. Therefore, bidding-price-secrecy of the AS02 protocol is captured by strong bidding-price-secrecy.

Strong bidding-price-secrecy ensures the anonymity of the link between a non-winning bidder and the price he bids for. It is formalised as that the adversary cannot distinguish between the case when a bidder bids for price a and the case when the bidder bids for price c. This property is formally defined in Definition 4.1.

```
$\mathcal{C}_{AS02}[\_]$ :=
        $\nu$privch$_{\mathsf{b}_1}$.$\nu$privch$_{\mathsf{b}_2}$. $\cdots$ .$\nu$privch$_{\mathsf{b}_n}$.
        $\nu$privcha$_{\mathsf{b}_1}$.$\nu$privcha$_{\mathsf{b}_2}$. $\cdots$ .$\nu$privcha$_{\mathsf{b}_n}$.
        $\nu$untapch$_{\mathsf{b}_1}$.$\nu$untapch$_{\mathsf{b}_2}$. $\cdots$ .$\nu$untapch$_{\mathsf{b}_n}$.
        $\nu$synch$_{\mathsf{b}_1}$.$\nu$synch$_{\mathsf{b}_2}$. $\cdots$ .$\nu$synch$_{\mathsf{b}_n}$.
        $(P_K \mid$
        (let $p_b = p_{\mathsf{b}_1}$ in let $untapch$ = untapch$_{\mathsf{b}_1}$ in
         let $privch$ = privch$_{\mathsf{b}_1}$ in let $ch$ = ch$_{\mathsf{b}_1}$ in $P_b) \mid$
        ...
        (let $p_b = p_{\mathsf{b}_{n-2}}$ in let $untapch$ = untapch$_{\mathsf{b}_{n-2}}$ in
         let $privch$ = privch$_{\mathsf{b}_{n-2}}$ in let $ch$ = ch$_{\mathsf{b}_{n-2}}$ in $P_b) \mid$
        $\_ \mid$
        $P_a)$
```

Figure 4.11: The context $\mathcal{C}_{AS02}[\_]$.

In the verification, we assume all the participants in the context are honest. Thus, the context $\mathcal{C}_{AS02}[\_]$ is defined as the auction process P_{AS02} with a hole (**c9**) instead of two bidder processes (see Figure 4.11). We denote the two bidder process as P_{bA} and P_{bB}. Sub-process **c5** to **c8** models the other $n-2$ bidder processes. To verify strong bidding-price-secrecy is to verify the following equivalence

$$
\begin{array}{rl}
\mathcal{C}_{AS02}[& (\text{let } p_b = \mathbf{a} \text{ in let } untapch = \mathtt{untapch}_{\mathsf{b}_A} \text{ in}\\
& \;\text{let } privch = \mathtt{privch}_{\mathsf{b}_A} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_b) \mid \\
& (\text{let } p_b = \mathbf{d} \text{ in let } untapch = \mathtt{untapch}_{\mathsf{b}_B} \text{ in}\\
& \;\text{let } privch = \mathtt{privch}_{\mathsf{b}_B} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_B} \text{ in } P_b)] \\
\approx_\ell \mathcal{C}_{AS02}[& (\text{let } p_b = \mathbf{c} \text{ in let } untapch = untapch_{\mathsf{b}_A} \text{ in}\\
& \;\text{let } privch = privch_{\mathsf{b}_A} \text{ in let } \mathsf{ch} = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_b) \mid \\
& (\text{let } p_b = \mathbf{d} \text{ in let } untapch = untapch_{\mathsf{b}_B} \text{ in}\\
& \;\text{let } privch = privch_{\mathsf{b}_B} \text{ in let } \mathsf{ch} = \mathtt{ch}_{\mathsf{b}_B} \text{ in } P_b)]
\end{array}
$$

where $\mathbf{a}, \mathbf{c}, \mathbf{d}$ are from the list $\mathsf{p}_1, \dots, \mathsf{p}_m$ with $\mathbf{a} < \mathbf{d}$ and $\mathbf{c} < \mathbf{d}$.

Strong secrecy properties can be verified, using ProVerif, by querying *noninterf*. Note that ProVerif is sensitive to evaluations of statements in the *if-then-else* constructs. ProVerif reports false attacks when directly querying the following predicate: *noninterf* p_b *among* $\mathsf{p}_1, \dots, \mathsf{p}_{d-1}$. To be able to check *noninterf* of a bidding price in ProVerif, we modify the bidder process by replacing *if-then-else* constructions with choices of a list of variables $vp_1, \dots, vp_{n-1}$. For example, lines (**b4-b6**) in Figure 4.6 is changed to "let $cmt^{\mathsf{p}_1} = \mathsf{commit}(\mathtt{r}_1, \mathsf{pk}(\mathtt{sk}_b), vp_1)$ in". Each variable vp_i corresponds to a price p_i and can be assigned to two possible values, either M_{yes} or M_{no}. If the variable is assigned M_{yes}, the bidder bids that price, otherwise, not. By querying "*noninterf* vp_1 *among* $(\mathsf{M}_{yes}, \mathsf{M}_{no}), \dots, vp_{n-1}$ *among* $(\mathsf{M}_{yes}, \mathsf{M}_{no})$", the variable vp_i is replaced with M_{yes} or M_{no}, resulting into different versions of the bidder process with different bidding prices (possibly bidding for multi-prices). ProVerif gives a positive result, which means that these process versions are all observationally equivalent. In this way, we prove that the protocol satisfies strong bidding-price-secrecy.

4.6.2 Receipt-freeness

Receipt-freeness is formally defined in Definition 4.2. To prove the AS02 protocol satisfies receipt-freeness, we need to find a process P_f which satisfies both equivalences in the definition of receipt-freeness. The process P_f is shown in Figure 4.12. The two equivalences are shown as follows (**eq1** and **eq2**).

eq1

$$
\begin{array}{rl}
& (\text{let } untapch = \mathtt{untapch}_{\mathsf{b}_A} \text{ in}\\
& \;\text{let } privch = \mathtt{privch}_{\mathsf{b}_A} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_f^{\backslash \mathsf{out}(\mathtt{chc}, \cdot)}) \\
\approx_\ell & (\text{let } p_b = \mathbf{c} \text{ in let } untapch = \mathtt{untapch}_{\mathsf{b}_A} \text{ in}\\
& \;\text{let } privch = \mathtt{privch}_{\mathsf{b}_A} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_b),
\end{array}
$$

eq2

$$
\begin{aligned}
&\mathcal{C}_{AS02}[(\text{let } p_b = \mathbf{a} \text{ in let } \mathit{untapch} = \mathtt{untapch}_{\mathtt{b}_A} \text{ in} \\
&\qquad \text{let } \mathit{privch} = \mathtt{privch}_{\mathtt{b}_A} \text{ in let } \mathit{ch} = \mathtt{ch}_{\mathtt{b}_A} \text{ in } P_b)^{\mathsf{chc}} \mid \\
&\qquad (\text{let } p_b = \mathbf{d} \text{ in let } \mathit{untapch} = \mathtt{untapch}_{\mathtt{b}_B} \text{ in} \\
&\qquad \text{let } \mathit{privch} = \mathtt{privch}_{\mathtt{b}_B} \text{ in let } \mathit{ch} = \mathtt{ch}_{\mathtt{b}_B} \text{ in } P_b)] \\
\approx_\ell\, &\mathcal{C}_{AS02}[(\text{let } \mathit{untapch} = \mathtt{untapch}_{\mathtt{b}_A} \text{ in} \\
&\qquad \text{let } \mathit{privch} = \mathtt{privch}_{\mathtt{b}_A} \text{ in let } \mathit{ch} = \mathtt{ch}_{\mathtt{b}_A} \text{ in } P_f) \mid \\
&\qquad (\text{let } p_b = \mathbf{d} \text{ in let } \mathit{untapch} = \mathtt{untapch}_{\mathtt{b}_B} \text{ in} \\
&\qquad \text{let } \mathit{privch} = \mathtt{privch}_{\mathtt{b}_B} \text{ in let } \mathit{ch} = \mathtt{ch}_{\mathtt{b}_B} \text{ in } P_b)]
\end{aligned}
$$

with $\mathbf{a} < \mathbf{d}$ and $\mathbf{c} < \mathbf{d}$.

```
      $P_f :=$
f1.   $\mathsf{in}(\mathit{privch}, \mathit{ssk}_b).\ \mathsf{out}(\mathsf{chc}, \mathit{ssk}_b).$
f2.   $\nu \mathsf{sk}_b.\ \mathsf{out}(\mathsf{chc}, \mathsf{sk}_b).$
f3.   $\mathsf{out}(\mathit{ch}, \mathsf{sign}((\mathsf{pk}(\mathsf{sk}_b), \mathsf{k}), \mathit{ssk}_b)).$
f4.   $\nu \mathtt{r}_1. \cdots .\nu \mathtt{r}_a. \cdots .\nu \mathtt{r}_c. \cdots .\nu \mathtt{r}_m.$
f5.   $\mathsf{out}(\mathsf{chc}, (\mathtt{r}_1, \dots, \mathsf{f}(\mathtt{r}_a), \dots, \mathsf{f}(\mathtt{r}_c), \dots, \mathtt{r}_m)).$
f6.   let $\mathit{cmt}^{\mathsf{p}_1} = \mathsf{commit}(\mathtt{r}_1, \mathsf{pk}(\mathsf{sk}_b), \mathsf{M}_{no})$ in
f7.   $\dots$
f8.   let $\mathit{cmt}^{\mathsf{p}_a} = \mathsf{commit}(\mathtt{r}_a, \mathsf{pk}(\mathsf{sk}_b), \mathsf{M}_{no})$ in
f9.   $\dots$
f10.  let $\mathit{cmt}^{\mathsf{p}_c} = \mathsf{commit}(\mathtt{r}_c, \mathsf{pk}(\mathsf{sk}_b), \mathsf{M}_{yes})$ in
f11.  $\dots$
f12.  let $\mathit{cmt}^{\mathsf{p}_m} = \mathsf{commit}(\mathtt{r}_m, \mathsf{pk}(\mathsf{sk}_b), \mathsf{M}_{no})$ in
f13.  $\mathsf{out}(\mathit{ch}, \mathsf{sign}((\mathit{cmt}^{\mathsf{p}_1}, \dots, \mathit{cmt}^{\mathsf{p}_m}), \mathit{ssk}_b)).$
f14.  $\mathsf{out}(\mathit{untapch}, \mathsf{sign}((\mathtt{r}_1, \dots, \mathtt{r}_a, \dots, \mathtt{r}_c, \dots, \mathtt{r}_m), \mathit{ssk}_b))$
```

Figure 4.12: The process P_f.

According to the properties of chameleon bit-commitments, the bidder can send a sequence of fake secret seeds to the adversary, and sends the series of real secret seeds to the auctioneer through an untappable channel. The adversary opens the bit-commitments as the bidder bids for price a, using the fake secret seeds he received, while the auctioneer opens the same bit-commitments as the bidder bids for price c, using the secret seeds the auctioneer received through an untappable channel. Thus, the bidder could execute the process P_f as shown in Figure 4.12 to lie to the adversary. The bidder in this process communicates with the adversary through channel chc, sending the adversary his secret signature key ssk_b (**f1**) and his secret key sk_b (**f2**). Later the bidder sends the auctioneer $\mathtt{r}_1, \dots, \mathtt{r}_m$ through an untappable channel (**f14**), and send the adversary the same list except changing $\mathtt{r}_a$ and $\mathtt{r}_c$ to $\mathsf{f}(\mathtt{r}_a)$ and $\mathsf{f}(\mathtt{r}_c)$, respectively (**f5**). The untappable channel ensures the adversary cannot learn anything about the differences.

Proof sketch: To prove the first equivalence, we can simply consider $P_f{}^{\backslash \mathsf{out}(\mathsf{chc},\cdot)}$ as process P_f without communication on the channel chc. Since the process $P_f{}^{\backslash \mathsf{out}(\mathsf{chc},\cdot)}$ works exactly the same as the process $P_b\{c/p_b\}$, the first equivalence (**eq1**) is satisfied. For detailed proofs, see Appendix A.1. To show the second equivalence (**eq2**), we need to consider all the transitions of each side. On both

$$
\begin{aligned}
P &\xrightarrow{\mathsf{in}(privch,ssk_b)} \xrightarrow{\mathsf{in}(privchb,bssk_b)} \xrightarrow{\nu x_1.\mathsf{out}(\mathtt{chc},x_1)} P_1 \mid \{ssk_b/x_1\} \\
&\xrightarrow{\nu x_2.\mathsf{out}(\mathtt{chc},x_2)} \nu\tilde{n}.\ (P_2 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\}) \\
&\xrightarrow{\nu x_3.\mathsf{out}(\mathtt{ch},x_3)} \\
&\xrightarrow{\nu x_4.\mathsf{out}(\mathtt{chc},x_4)} \nu\tilde{n}.\ (P_3 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\} \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{sk}_b), ssk_b)/x_3\} \\
&\qquad\qquad \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{bsk}_b), bssk_b)/x_4\}) \\
&\xrightarrow{\nu x_5.\mathsf{out}(\mathtt{chc},x_5)} \nu\tilde{n}.\ (P_4 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\} \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{sk}_b), ssk_b)/x_3\} \\
&\qquad\qquad \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{bsk}_b), bssk_b)/x_4\} \mid \{\mathtt{r}_1, \ldots, \mathtt{r}_m/x_5\} \\
&\xrightarrow{\nu x_6.\mathsf{out}(\mathtt{ch},x_6)} \\
&\xrightarrow{\nu x_7.\mathsf{out}(\mathtt{chc},x_7)} \nu\tilde{n}.\ (P_5 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\} \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{sk}_b), ssk_b)/x_3\} \\
&\qquad\qquad \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{bsk}_b), bssk_b)/x_4\} \\
&\qquad\qquad \mid \{\mathtt{r}_1, \ldots, \mathtt{r}_m/x_5\} \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1}, \ldots, cmt^{\mathsf{p}_m}), ssk_b)/x_6\} \\
&\qquad\qquad \mid \{\mathsf{sign}((bcmt^{\mathsf{p}_1}, \ldots, bcmt^{\mathsf{p}_m}), bssk_b)/x_7\})
\end{aligned}
$$

$$
\begin{aligned}
Q &\xrightarrow{in(privch,ssk_b)} \xrightarrow{in(privchb,bssk_b)} \xrightarrow{\nu x_1.\mathsf{out}(\mathtt{chc},x_1)} Q_1 \mid \{ssk_b/x_1\} \\
&\xrightarrow{\nu x_2.\mathsf{out}(\mathtt{chc},x_2)} \nu\tilde{n}.\ (Q_2 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\}) \\
&\xrightarrow{\nu x_3.\mathsf{out}(\mathtt{ch},x_3)} \\
&\xrightarrow{\nu x_4.\mathsf{out}(\mathtt{ch},x_4)} \nu\tilde{n}.\ (Q_3 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\} \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{sk}_b), ssk_b)/x_3\} \\
&\qquad\qquad \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{bsk}_b), bssk_b)/x_4\}) \\
&\xrightarrow{\nu x_5.\mathsf{out}(\mathtt{chc},x_5)} \nu\tilde{n}.\ (Q_4 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\} \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{sk}_b), ssk_b)/x_3\} \\
&\qquad\qquad \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{bsk}_b), bssk_b)/x_4\} \\
&\qquad\qquad \mid \{\mathtt{r}_1, \ldots, \mathsf{f}(\mathtt{r}_a), \ldots, \mathsf{f}(\mathtt{r}_c), \ldots, \mathtt{r}_m/x_5\}) \\
&\xrightarrow{\nu x_6.\mathsf{out}(\mathtt{ch},x_6)} \\
&\xrightarrow{\nu x_7.\mathsf{out}(\mathtt{ch},x_7)} \nu\tilde{n}.\ (Q_5 \mid \{ssk_b/x_1\} \mid \{\mathtt{sk}_b/x_2\} \mid \{\mathsf{sign}(\mathsf{pk}(\mathtt{sk}_b), ssk_b)/x_3\} \\
&\qquad\qquad \mid \{\mathsf{sign}((\mathsf{pk}(\mathtt{bsk}_b), bssk_b)/x_4\} \\
&\qquad\qquad \mid \{\mathtt{r}_1, \ldots, \mathsf{f}(\mathtt{r}_a), \ldots, \mathsf{f}(\mathtt{r}_c), \ldots, \mathtt{r}_m/x_5\} \\
&\qquad\qquad \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1}, \ldots, cmt^{\mathsf{p}_m}), ssk_b)/x_6\} \\
&\qquad\qquad \mid \{\mathsf{sign}((bcmt^{\mathsf{p}_1}, \ldots, bcmt^{\mathsf{p}_m}), bssk_b)/x_7\})
\end{aligned}
$$

Figure 4.13: A sketch proof of receipt-freeness in AS02.

sides, the process P_K only distributes keys, and all the bidder processes in the context follow the same process. For the sake of simplicity, we ignore the outputs in the process P_K and those bidder processes in the context. During the bidding phase the auctioneer process only reads information and synchronises on the private channel *synch*. There is no output on public channels in the auctioneer process. We denote the sequence of names $\mathtt{sk}_b, \mathtt{r}_1, \ldots, \mathtt{r}_m, \mathtt{bsk}_b, \mathtt{br}_1, \ldots, \mathtt{br}_m$ by $\tilde{n}$ ($\mathtt{sk}_b, \mathtt{r}_1, \ldots, \mathtt{r}_m$ are bound names in the non-winning bidder process, and $\mathtt{bsk}_b, \mathtt{br}_1, \ldots, \mathtt{br}_m$ are bound names in the winning bidder process P_{bB}). After the key distribution, we want to see whether the behaviour of the process $P_b\{a/p_b\}^{\mathtt{chc}} \mid P_{bB}\{d/p_b\}$ is observationally equivalent to $P_f \mid P_{bB}\{d/p_b\}$. For this purpose, we need to consider all possible executions of these two processes. Here, we consider a particular execution and only show the interesting part of the two frames after each step of execution by the two processes. Let $P := P_{bA}\{a/p_b\}^{\mathsf{chc}} \mid P_{bB}\{d/p_b\}$ and $Q := P_f \mid P_{bB}\{d/p_b\}$, we have their labelled transitions as shown in Figure 4.13.

The frames we obtained at the end of P and Q are statically equivalent. In particular, as the adversary knows the bit-commitments the bidder submits, the public key of the bidder, and the secret seeds, the adversary can open all the commitments of the bidder. The only functions the adversary can use are getmsg and open, to get extra information. By applying these two functions, the adversary can additional get the public key of the bidder represented as $x_{msg} = \mathsf{getmsg}(x_3, x_1)$ and a series of opened messages from bit-commitments. Since x_3 and x_1 are the same for both P and Q, x_{msg} is the same for both processes as well. Particularly, $P_{bA}\{a/p_b\}$ bids for price a. The adversary opens the commitments $cmt^{p_a} = \mathsf{commit}(r_a, \mathsf{pk}(sk_b), \mathsf{M}_{yes})$ and $cmt^{p_c} = \mathsf{commit}(r_c, \mathsf{pk}(sk_b), \mathsf{M}_{no})$ as follows:

$$\mathsf{open}(cmt^{p_a}, r_a, \mathsf{pk}(sk_b)) =_E \mathsf{M}_{yes} \quad \mathsf{open}(cmt^{p_c}, r_c, \mathsf{pk}(sk_b)) =_E \mathsf{M}_{no}$$

For the process Q, the process P_f bids for price c. The adversary has a sequence of secret seeds, in which two of them are fake: $\mathsf{f}(r_a)$ and $\mathsf{f}(r_c)$. According to the equational theory of chameleon bit-commitments (see Section 4.5), the adversary opens $cmt^{p_a} = \mathsf{commit}(r_a, \mathsf{pk}(sk_b), \mathsf{M}_{no}) =_E \mathsf{commit}(\mathsf{f}(r_a), \mathsf{pk}(sk_b), \mathsf{M}_{yes})$ and $cmt^{p_c} = \mathsf{commit}(r_c, \mathsf{pk}(sk_b), \mathsf{M}_{yes}) =_E \mathsf{commit}(\mathsf{f}(r_c), \mathsf{pk}(sk_b), \mathsf{M}_{no})$ as follows:

$$\mathsf{open}(cmt^{p_a}, \mathsf{f}(r_a), \mathsf{pk}(sk_b)) =_E \mathsf{M}_{yes} \quad \mathsf{open}(cmt^{p_c}, \mathsf{f}(r_c), \mathsf{pk}(sk_b)) =_E \mathsf{M}_{no}$$

All other secret seeds and bit-commitments are the same in both P and Q, hence the adversary gets the same series of opened messages for both P and Q as well.

Next, we consider the opening phase, the auctioneer process is the only active process. According to the protocol, the auctioneer process stops after finding the winning bids. Therefore, non-winning bids are not revealed. Since we have assumed the auctioneer is honest, the information that the auctioneer process reveals is the opened bit-commitments of all bidders at prices no lower than the winning price, and the winning bidders. Only the winning bid is opened as M_{yes}, others are opened as M_{no}. Due to the existence of a higher bid (d in the process $P_{bB}\{d/p_b\}$) on both sides of the equivalence, the bid made by the bidder b_A will never be published, hence the information the auctioneer process reveals is the same. Now, we can conclude that the protocol satisfies receipt-freeness. □

4.7 Conclusions

In this chapter, we first discussed privacy and enforced privacy issues in the e-auction domain. An enforced privacy property, receipt-freeness for non-winning bidders, is required in sealed-bid e-auctions. To formally define this property, we first formalised the classical privacy property, strong bidding-price-secrecy for non-winning bidders. On top of that, the enforced privacy property was formalised. As the requirement for privacy is different in e-auction and e-voting, the privacy, strong bidding-price-secrecy for non-winning bidders (Definition 4.1), is formalised differently from vote-privacy (Definition 3.14). Due to this difference, the enforced privacy property, receipt-freeness for non-winning bidders (Definition 4.2), is formalised differently from receipt-freeness in e-voting (Definition 3.17). In addition, we only model the cooperating bidder forwarding information to the adversary in e-auctions, because of the setting – the bidding booth in which bidders cannot communicate with the adversary. In situations without this setting, a similar property like coercion-resistance can be defined in e-auctions. Finally, the proposed formalisations of strong bidding-price-secrecy and receipt-freeness for non-winning bidders are validated in the case study. The AS02 protocol is modelled in the applied pi calculus, and we successfully verified that it does indeed satisfy bidding-price-secrecy and receipt-freeness for non-winning bidders. The protocol achieves this mainly based on the use of chameleon bit-commitments and an untappable channel, much like Okamoto used in his e-voting protocol [Oka96].

Future directions. The case study focused on a protocol relying on chameleon bit-commitments and an untappable channel. It would be interesting to use the proposed formalisation to verify correctness of enforced privacy claims of an auction protocol that uses other cryptographic primitives, such as the auction protocol by Chen et al. [CLK03], that relies on homomorphic encryption.

One of the main motivation for studying enforced privacy in auctions was the similarity between auction protocols and voting protocols. (For example, in both, roles can be divided into two types: participants and authorities.) Having successfully lifted the concept of enforced privacy to a similar domain, we are now ready to take matters one step further and study enforced privacy in a more complex setting.

5

Enforced privacy in e-health

In the previous chapter, we formalised enforced privacy in the e-auction domain. In this chapter, we study enforced privacy in the e-health domain. We identify and formalise enforced privacy in e-health in a similar way as in e-voting and e-auctions. Additionally, in e-health, it is required that pharmacists should not be able to prove a doctor's prescription behaviour to the adversary. We formalise a privacy property capturing this requirement. Furthermore, we formalise a privacy property capturing the conjunction of enforced privacy and pharmacist revealing information to the adversary. Finally, we verify the formalised privacy properties of an e-health protocol in a case study.

5.1 Introduction to e-health

E-health systems are health care systems using distributed electronic devices which communicate via the network, typically the Internet. E-health systems aim to support secure sharing of information and resources across different health care settings and workflows among different health care providers. The services of such systems for the general public are intended to be more secure, more effective, more efficient and more timely.

An e-health system normally consists of at least patients and doctors. Doctors prescribe medicine to patients according to examinations. In some e-health systems, this procedure can be done entirely using electronic devices. For instance, in home-care systems, patients do examinations at home. The examination results are sent to doctors via the network (automatically or manually). Then a doctor sends the prescription back. Whereas, in some other e-health systems, only storing information, such as patient's medical records, is digitalised. Consequently, e-health systems may have different privacy requirements. For instance, ensuring privacy in systems, which only involve electronic devices for storing data, mainly requires local protection. In e-health systems involving using open networks for communication, adversaries from the network need to be taken into consideration.

In addition, e-health systems may involve more roles. Once a patient obtains a prescription from doctors, the patient needs to get medicine according to the prescription. Therefore, pharmacists are involved. During this procedure, a pharmacist may have access to some private information, for example, prescriptions. In some systems, a pharmacist is even allowed to change one type of medicine to another type with similar functions. And in emergency cases, the pharmacist needs to be able to contact the doctor who has prescribed certain medicine. However,

This chapter is based on published work [FHIES11] and [ESORICS12] pharmacists, in general, are not always trustworthy. Hence, pharmacists may not be reliable about not revealing private information. In fact, pharmacists are allowed to offer certain information to others in particular cases, like doing research. Therefore, pharmacists may influence privacy in e-health.

Furthermore, depending on systems, even more other parties can be involved. For instance, medical administration and social insurance, which are normally assumed to trustworthy, and nurses, who need to access private information of patients. In some e-health systems, emergency situations are considered. In emergency situations, privacy becomes subtle since patient's information is normally revealed.

Hence, we can see that roles and relations in e-health systems are complex compared to e-voting and e-auction systems. In e-voting and e-auctions, there is a natural division into two types of roles: participants (voters, bidders) and authorities (who run the election/auction). E-health systems have to deal with a far more complex constellation of roles. This may cause complexity of privacy in e-health.

5.2 Privacy and enforced privacy in e-health

Due to the sensitive nature of health care data, privacy is one of the foremost challenges raised by adopting electronic storage and communication. In health care, it is often necessary to collect information for statistical study, like, collecting locations of patients having a certain type of disease to study the correspondence between locations and the disease. Thus, one privacy challenge is to maintain user anonymity even if information is revealed for statistical study. In addition, given the sensitive nature of health care data, handling this data must meet strict privacy requirements. Traditionally, data in health care (e.g., patient records) is stored on paper files. Privacy is relatively easily satisfied by controlling access to the physical documents. Those who had access could be considered trusted not to violate privacy of the data. With the advent of e-health systems – systems that digitally store and exchange health care data – health care data is open to inference by adversaries controlling the network or manipulating digital devices.

Indeed, privacy in e-health has been recognised as an important requirement necessary for adoption by the general public [MRS06, KAB09]. Moreover, due to the complexity of e-health systems, existing privacy control techniques from domains such as e-voting (e.g., [DKR09]) and e-auctions (e.g., [FAST10]) does not carry over straightforwardly. E-health systems involve more complex roles. Each of these roles has access to different private information and has different privacy concerns. As existing approaches from other domains are not properly equipped to handle such a diverse array of roles, privacy must be tailored to the e-health domain. In addition, roles may not be trustworthy, although some of them may be able to access sensitive data of others, for example, pharmacists normally have access to prescriptions. Thus, third parties may influence a target participant's privacy. Therefore, we shall consider privacy taking into account of third parties' influences.

Depending on the protected role, privacy in e-health can be classified into patient privacy and doctor privacy. Privacy of other roles, e.g., pharmacists, insurance companies, medical administrations, is beyond our consideration, as they are public entities.

5.2.1 Patient privacy

The importance of patient privacy in e-health is traditionally seen as vital to establishing a good doctor-patient relationship. This is even more pertinent with the emergence of the Electronic Patient Record [And96]. As in most of the literature, a necessary early stage of e-health is to transform the paper-based health care process into a digital process. The most important changes in this stage are made to patient information processing, mainly health care records. To properly express privacy requirements for such patient records, privacy policies are considered the de facto standard. There are three main approaches to implement these requirements of patient privacy: access control, architectural design, and the use of cryptography.

Patient privacy by access control. To preserve privacy of electronic health care records, one necessary part is to limit access to these records to allowed parties. The need for access control is supported by several privacy threats to personal health information listed by Anderson [And96]. Many access control approaches designed for patient privacy can be found in the literature, including access rules proposed by Anderson [And96], consent-based access rules [Lou98], role-based access control (RBAC) [RCHS03], organisation based access control [KBM$^+$03], etc.

Patient privacy by architectural design. E-health systems cater to a number of different roles, including doctors, patients, pharmacists, insurers, etc. Each such role has its own sub-systems or components. As such, e-health systems can be considered as a large network of systems, including administrative system components, laboratory information systems, radiology information systems, pharmacy information systems, and financial management systems. Diligent architectural design is an essential step to make such a complex system function correctly. Since privacy is important in e-health systems, keeping privacy in mind when designing the architecture of such systems is a promising path towards ensuring privacy [SV09]. Examples of architectures taking privacy in mind include the architecture of wireless sensor networks in e-health [KLS$^+$10], the architecture for e-health systems proposed by Maglogiannis et al. [MKDH09], the architecture for cross-institution image sharing in e-health [CHCK07], etc.

Cryptographic approaches to patient privacy. Cryptography is a necessary tool for privacy in e-health systems, especially communications between components of systems [BB96]. For example, Van der Haak et al. [vWB$^+$03] use digital signatures and public-key authentication (for access control) to satisfy legal requirements for cross-institutional exchange of electronic patient records. Ateniese et al. [ACdD03] use pseudonyms to preserve patient anonymity, and enable a user to transform statements concerning one of his pseudonyms into statements concerning one of his other pseudonyms (e.g., transforming a prescription for the pseudonym used with his doctor to a prescription for the pseudonym used with the pharmacist). Layouni et al. [LVS+09] consider communication between health monitoring equipment at a patient's home and the health care centre. They propose a protocol using wallet-based credentials (a cryptographic primitive) to let patients control when and how much identifying information is revealed by the monitoring equipment. More recently, De Decker et al. [dDLVV08] propose a health care system for communication between insurance companies and administrative bodies as well as patients, doctors and pharmacists. Their system relies on various cryptographic primitives to ensure privacy, including zero-knowledge proofs, signed proofs of knowledge (a signature scheme which uses zero-knowledge proofs to sign a message), and bit-commitments. Their system is explained in more detail in Section 5.4.

5.2.2 Doctor privacy

A relatively understudied privacy aspect is that of doctor privacy. Matyáš [Mat98] investigates the problem of enabling analysis of prescription information while ensuring doctor privacy. His approach is to group doctors, and release the data per group, hiding who is in the group. He does not motivate a need for doctor privacy, however. Two primary reasons for doctor privacy have been identified in the literature: (1) (Ateniese et al. [ACdD03]) to safeguard doctors against administrators setting specific efficiency metrics on their performance (e.g., requiring the cheapest medicine be used, irrespective of the patient's needs). To address this, Ateniese et al. [Ad02, ACdD03] propose an anonymous prescription system that uses group signatures for privacy; (2) (De Decker et al. [dDLVV08]) to prevent a pharmaceutical company from bribing a doctor to prescribe their medicine. A typical scenario can be described as follows. A pharmaceutical company seeks to persuade a doctor to favour a certain kind of medicine by bribing or coercing. To prevent this, a doctor should not be able to prove which medicine he is prescribing to this company (in general, to the adversary). This implies that doctor privacy must be enforced by e-health systems. De Decker et al. also note that preserving doctor privacy is not sufficient to prevent bribery: pharmacists could act as *go-betweens*, revealing the doctor's identity to the briber, as pharmacists often have access to prescriptions, and thus know something about the prescription behaviour of a doctor. This leads us to formulate the requirement of *independency of prescribing-privacy*: no third party should be able to help the adversary link a doctor to his prescription.

Observations

In the above overview, we observe that current approaches to privacy in e-health mostly focus on patient privacy and try to solve it as an access control or authentication problem. Doctor privacy is also required, but research on doctor privacy is in its infancy. We believe that doctor privacy is as important as patient privacy and should be studied in more depth. It is clear from the analysis that privacy in e-health systems needs to be addressed at different layers: access control ensures privacy at the service layer; privacy by architecture design addresses privacy concerns at the system/architecture layer; use of cryptography guarantees privacy at the communication layer. Since e-health systems are complex [TGC09] and rely on correct communications between many sub-systems, we strongly advocate to study privacy in e-health as a communication problem. In fact, message exchanges in communication protocols may leak information which leads to a privacy breach.

Classical privacy notions which are well-studied in the literature, attempt to ensure that privacy can be *enabled.* However, enabling privacy is far from enough. In many cases, a system must *enforce* user privacy instead of allowing the user to pursue it. To avoid doctor bribery, we take into account enforced privacy for doctors. In addition, we consider that one party's privacy may depend on another party (e.g., in the case of a pharmacist revealing prescription behaviour of a doctor). Our opinion is that offering privacy is insufficient if privacy can be reduced in such ways.

In summary, we focus on the following privacy notions for doctors in communication protocols in the e-health domain.

prescribing-privacy: A protocol preserves prescribing-privacy if the adversary cannot link a doctor to his prescriptions.

enforced prescribing-privacy: A protocol satisfies enforced prescribing-privacy if a doctor cannot prove his prescriptions to the adversary.

independency of prescribing-privacy: A protocol preserves independency of prescribing-privacy if third parties cannot help the adversary to link a doctor to the doctor's prescriptions.

independency of enforced prescribing-privacy: A protocol ensures independency of enforced prescribing-privacy if a doctor cannot prove his prescriptions to the adversary given that third parties sharing information with the adversary.

5.3 Formalisation of privacy notions

In order to formally verify privacy notions of a protocol, the first step is to give precise definitions of the privacy notions. The privacy notions in the previous section are formalised in the applied pi calculus. These notions focus on protecting doctor's prescription behaviour against bribery. Such kinds of privacy notions have not been studied formally so far. In the end, we briefly show the definitions of anonymity, strong anonymity, untraceability and strong untraceability for e-health protocols, as such notions have been formally studied in the literature (e.g., [SS96, vMR08, BHM08, KT09, ACRR10, KTV10]), which can be lifted to the e-health domain.

In the following discussions, we model an e-health protocol P_{eh} as an n-role well-formed [ACRR10] protocol of the form:

$$P_{eh} := \nu chandata.init.(!R_1 \mid \ldots \mid !R_n).$$

Unlike in e-voting and e-auctions where the number of participates is determined, in e-health, participants need not to be predetermined. Essentially, this formalisation allows us to model an unbounded number of users (modelled by the exclamation mark in front of each role) and represent each user as an instance of a role. In particular, we have a doctor role R_{dr} of the form:

$$R_{dr} := \nu \mathtt{Id}_{dr}.init_{dr}.!P_{dr},$$

where

$$P_{dr} := \nu \mathtt{presc}.main_{dr}.$$

We focus on the behaviour of a doctor, since we aim to formalise privacy notions for doctors. Each doctor is associated with an identity and can execute an infinite number of sessions (modelled by the exclamation mark in front of P_{dr}). Within each session, the doctor will create a prescription. Processes *init* and $init_{dr}$ model the initialisation of the protocol and the doctor role. Process P_{dr} models a session of the doctor role. Furthermore, we use $\mathcal{C}_{eh}[\_]$ to denote a context (a process with a hole) consisting of honest users,

$$\mathcal{C}_{eh}[\_] := \nu chandata.init.(!R_1 \mid \ldots \mid !R_n \mid \_).$$

Finally, Id_{dr} and *presc* are free variables; $\mathtt{d_A}$ and $\mathtt{d_B}$ are free names, representing doctor identities known to the adversary; and $\mathtt{p_A}$ and $\mathtt{p_B}$ are two free names, representing two different prescriptions.

5.3.1 Prescribing-privacy

Prescribing-privacy aims to protect doctors' prescription behaviour, which can be captured by the unlinkability of a doctor and his prescriptions. Considering that doctors' prescriptions are revealed eventually, e.g., in the DLV08 e-health protocol, unlinkability is modelled as indistinguishability when two honest users swap their actions (or items), e.g., see the formalisation of vote-privacy 3.14. Thus, prescribing-privacy is modelled as the equivalence of two doctor processes: in the first process, an honest doctor $\mathtt{d_A}$ prescribes $\mathtt{p_A}$ in one of his sessions and another honest doctor $\mathtt{d_B}$ prescribes $\mathtt{p_B}$ in one of his sessions; in the second one, $\mathtt{d_A}$ prescribes $\mathtt{p_B}$ and $\mathtt{d_B}$ prescribes $\mathtt{p_A}$.

Definition 5.1 (prescribing-privacy). *A well-formed e-health protocol P_{eh} with a doctor role R_{dr}, satisfies prescribing-privacy if for all possible doctors $\mathtt{d_A}$ and $\mathtt{d_B}$ ($\mathtt{d_A} \neq \mathtt{d_B}$) we have*

$$\begin{aligned}
&\mathcal{C}_{eh}[\big(init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_A}/presc\})\big) \mid \\
&\quad \big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_B}/presc\})\big)] \\
\approx_\ell\ &\mathcal{C}_{eh}[\big(init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_B}/presc\})\big) \mid \\
&\quad \big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_A}/presc\})\big)],
\end{aligned}$$

where $\mathtt{p_A}$ and $\mathtt{p_B}$ ($\mathtt{p_A} \neq \mathtt{p_B}$) are two prescriptions.

Process $init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_A}/presc\})$ models an instance of a doctor, with identity $\mathtt{d_A}$. The sub-process $main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_A}/presc\})$ models a prescribing session in which $\mathtt{d_A}$ prescribes $\mathtt{p_A}$ for a patient. The sub-process $!P_{dr}\{\mathtt{d_A}/Id_{dr}\}$ models other prescribing sessions of $\mathtt{d_A}$. Similarly, process $init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_B}/presc\})$ models another doctor $\mathtt{d_B}$. On the right hand of the equivalence, the two doctors, $\mathtt{d_A}$ and $\mathtt{d_B}$, swap their prescriptions, $\mathtt{p_A}$ and $\mathtt{p_B}$.

5.3.2 Enforced prescribing-privacy

De Decker et al. [dDLVV08] identify the need to prevent a pharmaceutical company from bribing a doctor to favour their medicine. Hence, doctor's prescribing-privacy should be enforced by protocols to prevent doctor bribery. This means that intuitively, even if a doctor collaborates, the adversary cannot be certain that the doctor has followed his instructions. Bribed users cannot be modelled as part of the adversary, as they are not trusted by the adversary. Inspired by the formalisation of receipt-freeness in e-voting 3.17 and e-auction 4.2, we define enforced prescribing-privacy to be satisfied if there exists a process where the bribed doctor does not follow the adversary's instruction (e.g., prescribing a particular medicine), which is indistinguishable from a process where she does.

Modelling this notion necessitates modelling a doctor who genuinely reveals all her private information to the adversary. This is achieved by process transformation $P^{\mathtt{chc}}$, which transforms a plain process P into one which shares all private information over the channel $\mathtt{chc}$ with the adversary (see Definition 3.15). In addition, we also use the transformation $P^{\setminus \mathsf{out}(\mathtt{chc},\cdot)})$ (see Definition 3.16). This models a process P which erases all outputs on channel $\mathtt{chc}$. Formally, $P^{\setminus \mathsf{out}(\mathtt{chc},\cdot)} := \nu \mathtt{chc}.(P \mid !\mathsf{in}(\mathtt{chc}, x))$.

Definition 5.2 (enforced prescribing-privacy)**.** *A well-formed e-health protocol P_{eh} with a doctor role R_{dr}, satisfies enforced prescribing-privacy for a doctor $\mathtt{d}_\mathtt{A}$, if there exist processes $init'_{dr}$ and P'_{dr}, such that:*

1. $$\begin{aligned} & \mathcal{C}_{eh}\big[\big(init'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\})\big) \mid \\ & \qquad \big(init_{dr}\{\mathtt{d}_\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d}_\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d}_\mathtt{B}/Id_{dr}, \mathtt{p}_\mathtt{A}/presc\})\big)\big] \\ \approx_\ell\ & \mathcal{C}_{eh}\big[\big((init_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\})^{\mathtt{chc}}.(!P_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\} \mid (main_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}, \mathtt{p}_\mathtt{A}/presc\})^{\mathtt{chc}})\big) \mid \\ & \qquad \big(init_{dr}\{\mathtt{d}_\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d}_\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d}_\mathtt{B}/Id_{dr}, \mathtt{p}_\mathtt{B}/presc\})\big)\big]; \end{aligned}$$

2. $$\begin{aligned} & init'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\}^{\setminus \mathsf{out}(\mathtt{chc},\cdot)}.(P'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\}^{\setminus \mathsf{out}(\mathtt{chc},\cdot)}) \\ \approx_\ell\ & init_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\}.(main_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}, \mathtt{p}_\mathtt{B}/presc\}), \end{aligned}$$

where $init'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\})$ is a closed plain process, $\mathtt{chc}$ is a fresh channel name, $\mathtt{p}_\mathtt{A}$ and $\mathtt{p}_\mathtt{B}$ ($\mathtt{p}_\mathtt{A} \neq \mathtt{p}_\mathtt{B}$) are two prescriptions, and $\mathtt{d}_\mathtt{B}$ ($\mathtt{d}_\mathtt{A} \neq \mathtt{d}_\mathtt{B}$) is a doctor identity.

In the definition, the process $init'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{d}_\mathtt{A}/Id_{dr}\})$ models the process in which the doctor $\mathtt{d}_\mathtt{A}$ lies to the adversary about one of his prescriptions. The real prescription behaviour of $\mathtt{d}_\mathtt{A}$ is modelled by the second equivalence. The first equivalence shows that the adversary cannot distinguish whether $\mathtt{d}_\mathtt{A}$ lied, given a counter-balancing doctor $\mathtt{d}_\mathtt{B}$.

The difference between this formalisation and receipt-freeness in e-voting and in e-auctions is that in this definition only a part of the doctor process (the initiation sub-process and a prescribing session) shares information with the adversary. In e-voting, each voter only votes once. In the contrast, a doctor prescribes multiple times for various patients. As patients and situations of patients vary, a doctor cannot prescribe medicine from the bribing pharmaceutical company all the time. Therefore, only part of the doctor process shares information with the adversary. We model only one bribed prescribing session, as that if a doctor can cheat the adversary in one session, he can easily cheat in multiple similar sessions. This definition can be easily extended to model multiple prescribing sessions being bribed.

5.3.3 Independency of prescribing-privacy

Usually, e-health systems have to deal with a complex constellation of roles: doctors, patients, pharmacists, insurance companies, medical administration, etc. Each of these roles has access to different private information and has different privacy concerns. An untrusted role may be bribed to reveal private information to the adversary such that the adversary can break another roles' privacy. De Decker et al. [dDLVV08] note that pharmacists may have sensitive data which can be revealed to the adversary to break a doctor's prescribing-privacy. To prevent a party (not a doctor) to do this, e-health protocols are required to satisfy *independency of prescribing-privacy*, meaning that even if another party R_i reveals their information (i.e., $R_i^{\mathtt{chc}}$), the adversary should not be able to break a doctor's prescribing-privacy.

Definition 5.3 (independency of prescribing-privacy). *A well-formed e-health protocol P_{eh} with a doctor role R_{dr}, satisfies* prescribing-privacy independent of role *R_i, if for all possible doctors $\mathtt{d_A}$ and $\mathtt{d_B}$ ($\mathtt{d_A} \neq \mathtt{d_B}$) we have*

$$\begin{aligned}
&\mathcal{C}_{eh}[!R_i{}^{\mathtt{chc}} \mid \big(init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_A}/presc\})\big) \mid \\
&\qquad\qquad \big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_B}/presc\})\big)] \\
\approx_\ell\ &\mathcal{C}_{eh}[!R_i{}^{\mathtt{chc}} \mid \big(init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_B}/presc\})\big) \mid \\
&\qquad\qquad \big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_A}/presc\})\big)].
\end{aligned}$$

where $\mathtt{p_A}$ and $\mathtt{p_B}$ ($\mathtt{p_A} \neq \mathtt{p_B}$) are two prescriptions, and R_i is a non-doctor role.

Note that we assume a worst situation in which role R_i genuinely cooperates with the adversary. For example, the pharmacist forwards all information obtained from channels hidden from the adversary. The equivalence shows that no matter how role R_i cooperates with the adversary, the adversary cannot link a doctor to the doctor's prescriptions.

5.3.4 Independency of enforced prescribing-privacy

We have discussed two situations where a doctor's prescription behaviour can be revealed when either the doctor or another different party cooperates with the adversary. It is natural to consider the conjunction of these two, i.e., a situation in which the adversary coerces both a doctor and another party (not a doctor). Since the adversary obtains more information, this constitutes a stronger attack on doctor's prescribing-privacy. To address this problem, we define *independency of enforced prescribing-privacy*, which is satisfied when a doctor's prescribing-privacy is preserved even if both the doctor and another party reveal their private information to the adversary.

Definition 5.4 (independency of enforced prescribing-privacy). *A well-formed e-health protocol P_{eh} with a doctor role R_{dr}, satisfies* enforced prescribing-privacy independent of role *R_i for a doctor $\mathtt{d_A}$, if there exist processes $init'_{dr}$ and P'_{dr}, such* *that:*

1. $$\begin{aligned} &\mathcal{C}_{eh}[!R_i^{\mathtt{chc}} \mid \big(init'_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{d_A}/Id_{dr}\})\big) \mid \\ &\qquad \big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_A}/presc\})\big)] \\ \approx_\ell\ &\mathcal{C}_{eh}[!R_i^{\mathtt{chc}} \mid \big((init_{dr}\{\mathtt{d_A}/Id_{dr}\})^{\mathtt{chc}}. \\ &\qquad (!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid (main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_A}/presc\})^{\mathtt{chc}})\big) \mid \\ &\qquad \big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_B}/presc\})\big)]; \end{aligned}$$

2. $$\begin{aligned} &init'_{dr}\{\mathtt{d_A}/Id_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}.(P'_{dr}\{\mathtt{d_A}/Id_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}) \\ \approx_\ell\ &init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(main_{dr}\{\mathtt{d_A}/Id_{dr}, \mathtt{p_B}/presc\}), \end{aligned}$$

where $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{d_A}/Id_{dr}\})$ *is a closed plain process,* R_i *is a non-doctor role,* $\mathtt{chc}$ *is a fresh channel name,* $\mathtt{p_A}$ *and* $\mathtt{p_B}$ *(*$\mathtt{p_A} \neq \mathtt{p_B}$*) are two prescriptions, and* $\mathtt{d_B}$ *(*$\mathtt{d_A} \neq \mathtt{d_B}$*) is a doctor identity.*

We conjecture that independency of enforced prescribing-privacy implies independency of prescribing-privacy and enforced prescribing-privacy, each of which also implies prescribing-privacy.

5.3.5 Anonymity and strong anonymity

Anonymity is a privacy notion that protects users' identities. We model anonymity as indistinguishability of processes initiated by two different users.

Definition 5.5 (doctor anonymity)**.** *A well-formed e-health protocol* P_{eh} *with a doctor role* R_{dr} *satisfies doctor anonymity for a doctor* $\mathtt{d_A}$ *if there exists another doctor* $\mathtt{d_B}$*, such that*

$$\mathcal{C}_{eh}[init_{dr}\{\mathtt{d_A}/Id_{dr}\}.!P_{dr}\{\mathtt{d_A}/Id_{dr}\}] \approx_\ell \mathcal{C}_{eh}[init_{dr}\{\mathtt{d_B}/Id_{dr}\}.!P_{dr}\{\mathtt{d_B}/Id_{dr}\}].$$

A stronger notion of anonymity is defined in [ACRR10], capturing the situation that the adversary cannot even find out whether a user (with identity $\mathtt{d_A}$) has participated in a session of the protocol or not.

Definition 5.6 (strong doctor anonymity [ACRR10])**.** *A well-formed e-health protocol* P_{eh} *with a doctor role* R_{dr} *satisfies strong doctor anonymity, if*

$$P_{eh} \approx_\ell \nu chandata.init.\big(!R_1 \mid \ldots \mid !R_n \mid (init_{dr}\{\mathtt{d_A}/Id_{dr}\}.!P_{dr}\{\mathtt{d_A}/Id_{dr}\})\big).$$

Similarly, we can define anonymity and strong anonymity for patient and other roles in an e-health protocol, by replacing the doctor role with a different role.

5.3.6 Untraceability and strong untraceability

Untraceability is a notion preventing the adversary from tracing a user, meaning that he cannot tell whether two executions are initiated by the same user.

Definition 5.7 (doctor untraceability)**.** *A well-formed e-health protocol* P_{eh} *with a doctor role* R_{dr} *satisfies doctor untraceability if, for any two doctors* $\mathtt{d_A}$ *and* $\mathtt{d_B} \neq \mathtt{d_A}$*,*

$$\begin{aligned} &\mathcal{C}_{eh}[init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid P_{dr}\{\mathtt{d_A}/Id_{dr}\})] \\ \approx_\ell\ &\mathcal{C}_{eh}[(init_{dr}\{\mathtt{d_A}/Id_{dr}\}.P_{dr}\{\mathtt{d_A}/Id_{dr}\}) \mid (init_{dr}\{\mathtt{d_B}/Id_{dr}\}.P_{dr}\{\mathtt{d_B}/Id_{dr}\})]. \end{aligned}$$

A stronger notion of untraceability is proposed in [ACRR10] that captures the adversary's inability to distinguish the situation where one user executes the protocol multiple times from no user executing the protocol more than once.

Definition 5.8 (strong doctor untraceability [ACRR10])**.** *A well-formed e-health protocol P_{eh} with a doctor role R_{dr} being the j-th role, satisfies strong doctor untraceability, if*

$$P_{eh} \approx_{\ell} \nu chandata.init.\big(!R_1 \mid \ldots \mid !R_{j-1} \mid !R_{j+1} \mid !R_n \mid !(\nu \mathtt{Id}_{dr}.init_{dr}.P_{dr})\big).$$

Similarly, we can define untraceability and strong untraceability for patient and other roles in a protocol, by replacing the doctor role with a different role.

5.4 Case study: the DLV08 protocol

In this section, we apply the above formal definitions for doctor privacy in a case study as a validation of the definitions. We choose to analyse the DLV08 e-health protocol proposed by De Decker et al. since it claims enforced privacy for doctors. However, the analysis is not restricted to doctor privacy. We provide a rather complete analysis of the protocol including patient anonymity, patient untraceability, patient/doctor information secrecy and patient/doctor authentication as well. ProVerif code is available at [1].

5.4.1 Introduction

The DLV08 protocol is a complex health care protocol for the Belgium situation [dDLVV08], which captures most aspects of the current Belgian health care practice and aims to provide a strong guarantee of privacy for patients and doctors. The protocol involves five roles: doctor, patient, pharmacist, medicine prescription administrator (MPA) and health insurance institute (HII); and it works as follows: a doctor prescribes medicine to a patient; next the patient obtains medicine from a pharmacist according to the prescription; following that, the pharmacist forwards the prescription to his MPA, the MPA checks the prescription and refunds the pharmacist; finally, the MPA sends invoices to the patient's HII and is refunded. As we do not focus on properties such as revocability and reimbursement, we do not consider the other two roles: public safety organisation (PSO) and social security organisation (SSO).

5.4.2 Cryptographic primitives

To ensure security and privacy properties, the DLV08 protocol employs several special cryptographic primitives, besides the classical ones, like encryption. We briefly introduce the following special cryptographic primitives.

Bit-commitments. The bit-commitments scheme consists of two phases, committing phase and opening phase. On the committing phase, a message sender makes a commitment on a message. It can be considered as putting the message into a box, and sending the box to the receiver. Later in the opening phase, the sender sends the key of the box to the receiver. The receiver opens the box and obtains the message.

Zero-knowledge proofs. A zero-knowledge proof is a cryptographic scheme which can be used for one party (prover) to prove to another party (verifier) that a statement is true, without leaking secret information of the prover. A zero-knowledge proof scheme can be interactive or non-interactive. We consider the non-interactive zero-knowledge proofs in this protocol.

Digital credentials. A digital credential is like a certificate, which can be used to prove that the owner qualifies some requirements. Unlike some paper certificates such as passport which gives out the owner's identity, a digital credential could be used to authenticate the owner anonymously. For example, a digital credential can be used to prove that a driver is old enough to drive without showing the age of the driver.

Anonymous authentication. Anonymous authentication is a scheme for authenticating a user anonymously. The procedure of an anonymous authentication is actually a zero-knowledge proof, with the digital credential being the public information of the prover. In the scheme, a user's digital credential is used as the public key in the public key authentication structure. A verifier can check whether a message is signed correctly by the prover, while the verifier cannot identify the prover. Thus, this ensures anonymous authentication.

Verifiable encryptions. A verifiable encryption is a zero-knowledge proof as well. A prover encrypts a message, and uses zero-knowledge proofs to prove that the encrypted message satisfies some properties without showing the original message.

Signed proofs of knowledge. Signed proofs of knowledge is using proofs of knowledge as a digital signature scheme (for details see [Bra00]). Intuitively, a prover signs a message using some secret information, which can be considered as a secret signing key. And the prover uses proofs of knowledge to convince the verifier that he has the secret signing key corresponding to the public key.

5.4.3 Settings

Every participant of the protocol is equipped with some initial information.

- A doctor has an identity ($\mathtt{Id}_{dr}$), a pseudonym ($\mathtt{Pnym}_{dr}$), and an anonymous doctor credential (Cred_{dr}) issued by trusted authorities.
- A patient has an identity ($\mathtt{Id}_{pt}$), a pseudonym ($\mathtt{Pnym}_{pt}$), an HII ($\mathtt{Hii}$), a social security status ($\mathtt{Sss}$), a health expense account ($\mathtt{Acc}$) and an anonymous patient credential (Cred_{pt}) issued by trusted authorities.

- Pharmacists, MPA, and HII are public entities, each of which has an identity ($\mathtt{Id}_{ph}$, $\mathtt{Id}_{mpa}$, $\mathtt{Id}_{hii}$), a secret key ($\mathtt{sk}_{ph}$, $\mathtt{sk}_{mpa}$, $\mathtt{sk}_{hii}$) and an authorised public key certificate ($\mathtt{pk}_{ph}$, $\mathtt{pk}_{mpa}$, $\mathtt{pk}_{hii}$) issued by trusted authorities.

5.4.4 Description of the protocol

The DLV08 protocol consists of four sub-protocols: doctor-patient sub-protocol, patient-pharmacist sub-protocol, pharmacist-MPA sub-protocol, and MPA-HII sub-protocol. We describe the sub-protocols one by one.

Doctor-patient sub-protocol

The doctor authenticates himself to a patient using the authorised doctor credential. The patient verifies the doctor credential. If the verification passes, the patient authenticates himself to the doctor using the patient credential, sends the bit-commitments on his identity to the doctor, and proves to the doctor that the identity used in the credential is the same as in the bit-commitments. After verifying the patient credential, the doctor generates a prescription, computes a prescription identity, computes the doctor bit-commitments. Then the doctor combines these computed messages with the received patient bit-commitments; signs these messages using a signed proof of knowledge, which proves that the doctor's pseudonym used in the doctor credential is the same as in the doctor bit-commitments. Together with the proof, the doctor sends the open information of the doctor bit-commitments. The communication in the doctor-patient sub-protocol is shown as a message sequence chart (MSC) in Figure 5.1.

Patient-Pharmacist sub-protocol

The pharmacist authenticates himself to the patient. The patient verifies the authentication and obtains, from the authentication, the pharmacist's identity and the pharmacist's MPA. Then the patient anonymously authenticates himself to the pharmacist, and proves his social security status. Next, the patient computes verifiable encryptions vc_1, vc_2, vc_3, vc_3', vc_4, vc_5, where

- vc_1 encrypts the patient's HII using the MPA's public key and proves that the HII encrypted in vc_1 is the same as the one in the patient's credential.

- vc_2 encrypts the doctor's pseudonym using the MPA's public key and proves that the doctor's pseudonym encrypted in vc_2 is the same as the one in the doctor commitment embedded in the prescription.

- vc_3 encrypts the patient's pseudonym using the public safety organisation's public key and proves that the pseudonym encrypted in vc_3 is the same as the one in the patient's commitment.

- vc_3' encrypts the patient's HII using the social security organisation's public key and proves that the content encrypted in vc_3' is the same as the HII in the patient's credential.

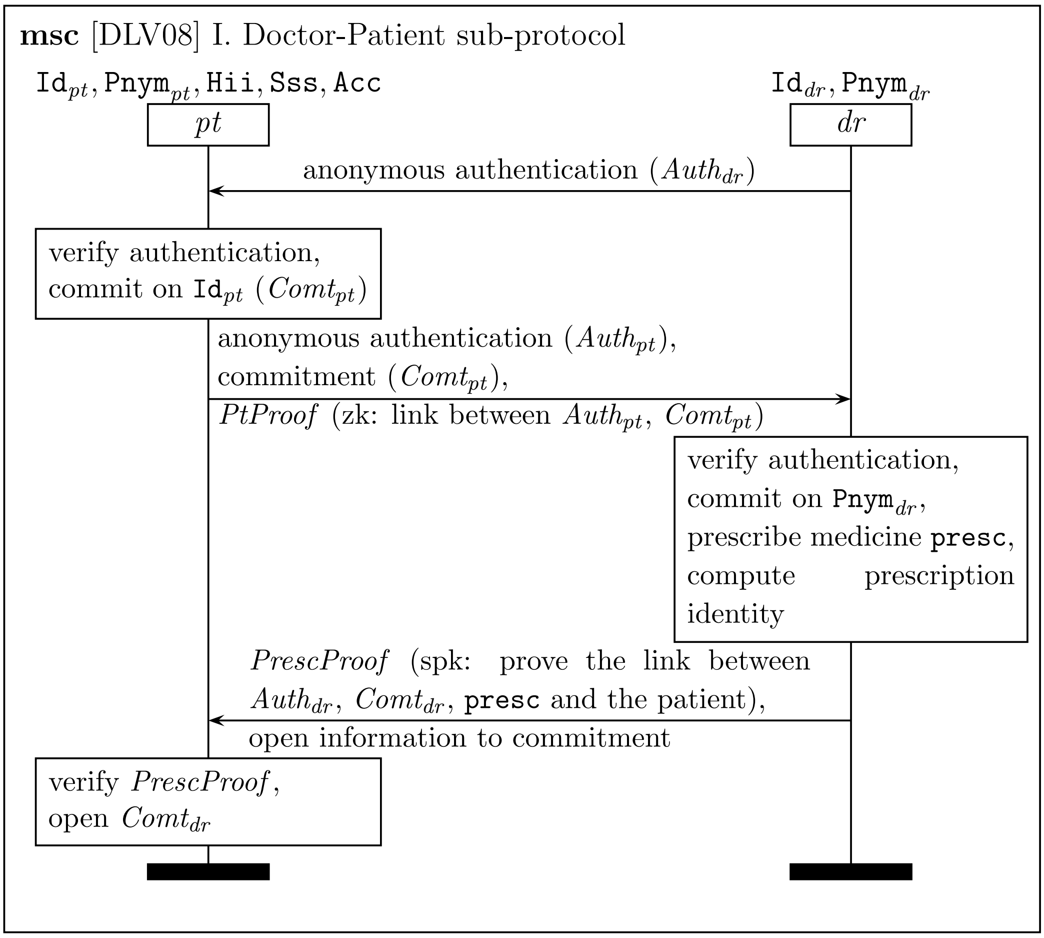
msc [DLV08] I. Doctor-Patient sub-protocol
$\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}$
$\mathtt{Id}_{dr}, \mathtt{Pnym}_{dr}$
pt
dr
anonymous authentication ($Auth_{dr}$)
verify authentication, commit on $\mathtt{Id}_{pt}$ ($Comt_{pt}$)
anonymous authentication ($Auth_{pt}$), commitment ($Comt_{pt}$), $PtProof$ (zk: link between $Auth_{pt}$, $Comt_{pt}$)
verify authentication, commit on $\mathtt{Pnym}_{dr}$, prescribe medicine $\mathtt{presc}$, compute prescription identity
$PrescProof$ (spk: prove the link between $Auth_{dr}$, $Comt_{dr}$, $\mathtt{presc}$ and the patient), open information to commitment
verify $PrescProof$, open $Comt_{dr}$

Figure 5.1: Doctor-Patient sub-protocol.

- vc_4 encrypts the patient's pseudonym using the MPA's public key and proves that the patient's pseudonym encrypted in vc_4 is the same as the one in the patient's credential.

- vc_5 encrypts the patient's pseudonym using his HII's public key and proves that the patient's pseudonym encrypted in vc_5 is the same as the one in the patient's credential.

- c_5 encrypts vc_5 using the MPA's public key.

The patient sends the received prescription to the pharmacist and proves to the pharmacist that the patient's identity in the prescription is the same as in the patient credential. The patient sends vc_1, vc_2, vc_3, vc_3', vc_4, c_5 as well. The pharmacist verifies the correctness of all the received messages. If every message is correctly formatted, the pharmacist charges the patient, and delivers the medicine. Then the pharmacist generates an invoice and sends it to the patient. The patient computes a receipt *ReceiptAck*: signing a message (consists of the prescription identity, the pharmacist's identity, vc_1, vc_2, vc_3, vc_3', vc_4, vc_5) using a signed proof of knowledge and proving that he can construct the patient credential. This receipt proves that the patient has received his medicine. The pharmacist verifies the correctness of the receipt. The communication in the patient-Pharmacist sub-protocol is shown in Figure 5.2.

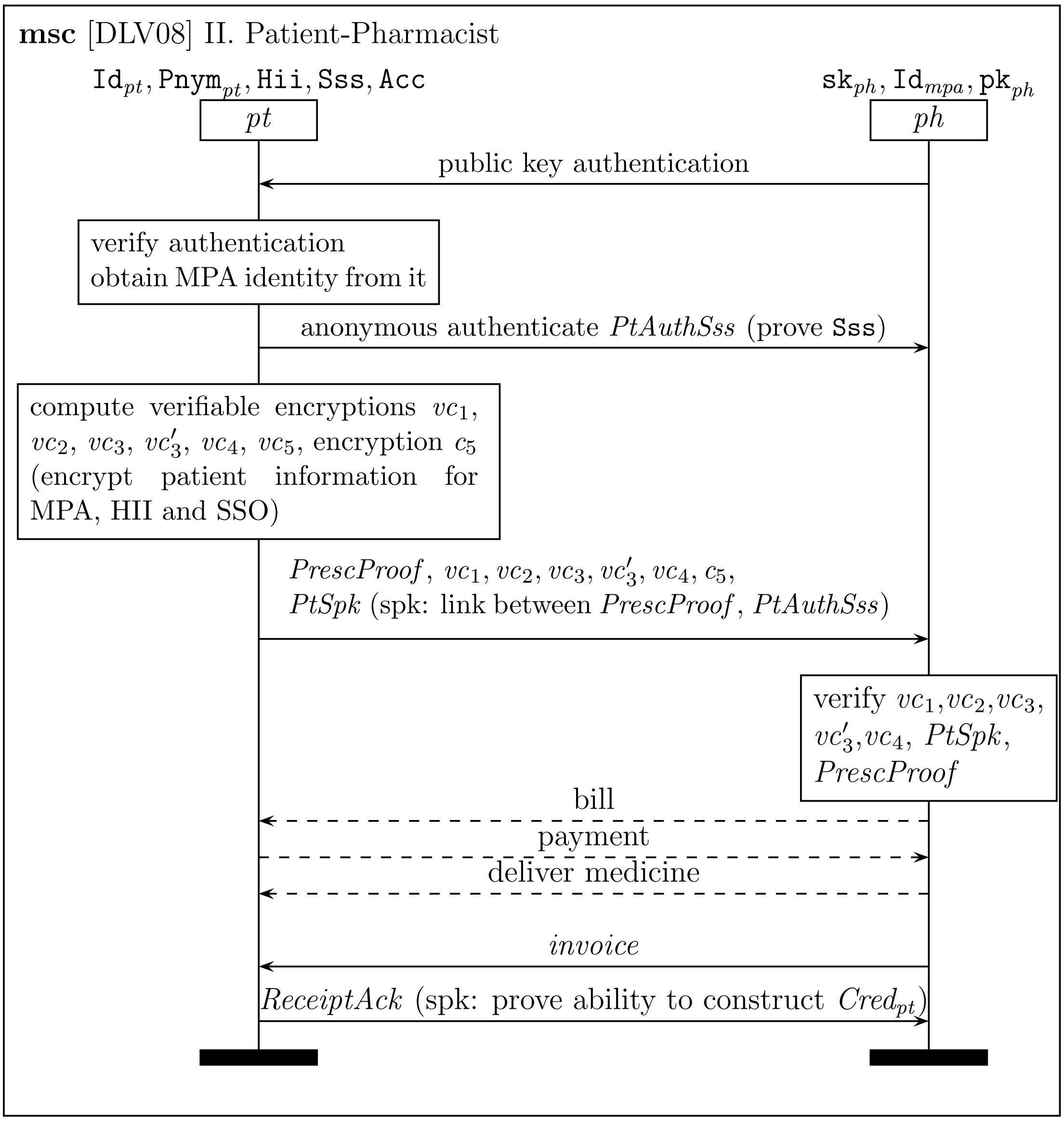
msc [DLV08] II. Patient-Pharmacist
$\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}$
$\mathtt{sk}_{ph}, \mathtt{Id}_{mpa}, \mathtt{pk}_{ph}$
pt
ph
public key authentication
verify authentication obtain MPA identity from it
anonymous authenticate *PtAuthSss* (prove Sss)
compute verifiable encryptions vc_1, vc_2, vc_3, vc'_3, vc_4, vc_5, encryption c_5 (encrypt patient information for MPA, HII and SSO)
PrescProof, $vc_1, vc_2, vc_3, vc'_3, vc_4, c_5$, *PtSpk* (spk: link between *PrescProof*, *PtAuthSss*)
verify vc_1,vc_2,vc_3, vc'_3,vc_4, *PtSpk*, *PrescProof*
bill
payment
deliver medicine
invoice
ReceiptAck (spk: prove ability to construct $Cred_{pt}$)

Figure 5.2: Patient-Pharmacist sub-protocol.

Pharmacist-MPA sub-protocol

The pharmacist and the MPA first authenticate each other using public key authentication. Then the pharmacist sends the received prescription and the receipt *ReceiptAck*, together with vc_1, vc_2, vc_3, vc'_3, vc_4, c_5, to the MPA. The MPA verifies correctness of the received information. Then, the MPA decrypts vc_1, vc_2, vc_4 and c_5, which provide the patient's HII, the doctor's pseudonym, the patient's pseudonym, and vc_5. The communication in the pharmacist-MPA sub-protocol is shown in Figure 5.3.

MPA-HII sub-protocol

The MPA and the patient's HII first authenticate each other using public key authentication. Then the MPA sends the receipt *ReceiptAck* to the patient's HII as well as the verifiable encryption vc_5 which encrypts the patient's pseudonym

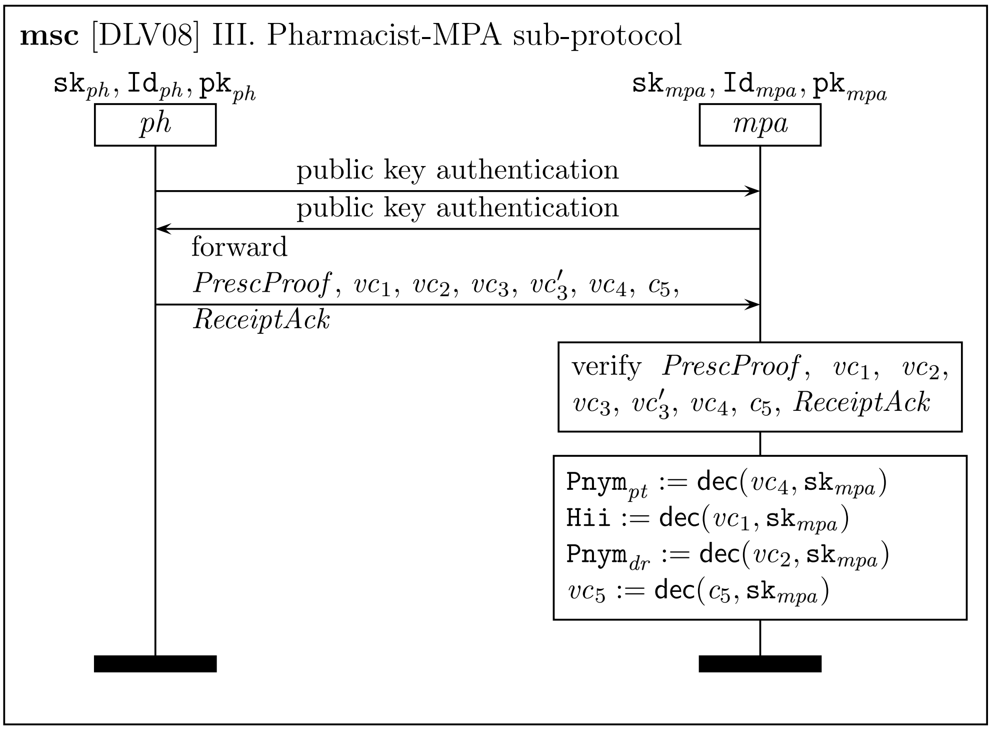
msc [DLV08] III. Pharmacist-MPA sub-protocol
$\mathtt{sk}_{ph}, \mathtt{Id}_{ph}, \mathtt{pk}_{ph}$
ph
$\mathtt{sk}_{mpa}, \mathtt{Id}_{mpa}, \mathtt{pk}_{mpa}$
mpa
public key authentication
public key authentication
forward
$PrescProof$, vc_1, vc_2, vc_3, vc_3', vc_4, c_5,
$ReceiptAck$
verify $PrescProof$, vc_1, vc_2, vc_3, vc_3', vc_4, c_5, $ReceiptAck$
$\mathtt{Pnym}_{pt} := \mathsf{dec}(vc_4, \mathtt{sk}_{mpa})$
$\mathtt{Hii} := \mathsf{dec}(vc_1, \mathtt{sk}_{mpa})$
$\mathtt{Pnym}_{dr} := \mathsf{dec}(vc_2, \mathtt{sk}_{mpa})$
$vc_5 := \mathsf{dec}(c_5, \mathtt{sk}_{mpa})$

Figure 5.3: Pharmacist-MPA sub-protocol.

with the patient's HII's public key. The patient's HII checks the correctness of *ReceiptAck*, decrypts vc_5 and obtains the patient's pseudonym. From the patient pseudonym, the HII obtains the identity of the patient; then updates the patient's account and pays the MPA. The MPA pays the pharmacist when he receives the payment. The communication in the MPA-HII sub-protocol is shown in Figure 5.4.

5.4.5 Claimed privacy properties

The DLV08 protocol is claimed to satisfy the following properties:

- Secrecy of patient and doctor information: No other party should be able to know a patient or a doctor's information, unless the information is intended to be revealed in the protocol.
- Authentication: All parties should properly authenticate each other.
- Patient anonymity: No party should be able to determine a patient's identity.
- Patient untraceability: Prescriptions issued to the same patient should not be linkable to each other.
- Prescribing-privacy: The protocol protects a doctor's prescription behaviour.
- Enforced prescribing-privacy: The protocol prevents bribery between doctors and pharmaceutical companies.
- Independency of prescribing-privacy: Pharmacists should not be able to provide evidence to pharmaceutical companies about doctors' prescription.

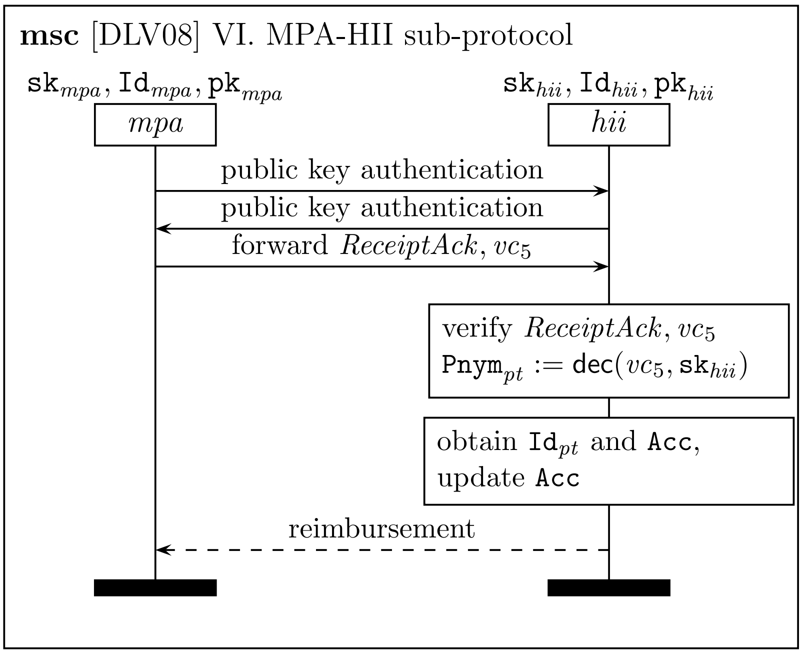
msc [DLV08] VI. MPA-HII sub-protocol
$\mathtt{sk}_{mpa}, \mathtt{Id}_{mpa}, \mathtt{pk}_{mpa}$
$\mathtt{sk}_{hii}, \mathtt{Id}_{hii}, \mathtt{pk}_{hii}$
mpa
hii
public key authentication
public key authentication
forward *ReceiptAck*, vc_5
verify *ReceiptAck*, vc_5
$\mathtt{Pnym}_{pt} := \mathtt{dec}(vc_5, \mathtt{sk}_{hii})$
obtain $\mathtt{Id}_{pt}$ and $\mathtt{Acc}$,
update $\mathtt{Acc}$
reimbursement

Figure 5.4: MPA-HII sub-protocol.

5.5 Modelling DLV08

We model the DLV08 protocol in the applied pi calculus. Since the description of the protocol is not clear in some details, before modelling the protocol, a few ambiguities need to be settled. Next we explain the modelling of a few cryptographic primitives, since security and privacy rely heavily on these cryptographic primitives in the protocol. Then, we illustrate the modelling of the protocol.

5.5.1 Underspecification of the DLV08 protocol

The DLV08 protocol leaves the following issues unspecified:

- **a1** whether a zero-knowledge proof is transferable;
- **a2** whether an encryption is probabilistic;
- **a3** whether a patient/doctor uses a fresh identity and/or pseudonym for each session;
- **a4** whether credentials are freshly generated in each session;
- **a5** what a patient's social security status is and how it can be modified;
- **a6** how many HIIs exist and whether a patient can change his HII;
- **a7** whether a patient/doctor can obtain a credential by requesting one;
- **a8** what type of communication channels are used (public or untappable).

To be able to discover potential flaws on privacy, we make the following (weakest) assumptions in our modelling of the DLV08 protocol:

- **s1** the zero-knowledge proofs used are non-interactive and transferable;
- **s2** encryptions are not probabilistic;
- **s3** a patient/doctor uses the same identity and pseudonym in every session;
- **s4** a patient/doctor has the same credential in every session;
- **s5** a patient's social security status is the same in every session;
- **s6** there are many HIIs, different patients may have different HIIs, and a patient's HII is fixed and cannot be changed;
- **s7** a patient/doctor's credential can be obtained by requesting one;
- **s8** the communication channels are public.

5.5.2 Modelling cryptographic primitives

The cryptographic primitives are modelled in the applied pi calculus using function symbols and equational theory. All functions and equational theory are shown in Figure 5.5, Figure 5.6 and Figure 5.7.

Bit-commitments. The bit-commitments scheme is modelled as two functions: function commit, modelling the committing phase, and function open, modelling the opening phase. Function commit creates a commitment with two parameters: a message m and a random number r. A commitment can only be opened with the correct opening information r, thus reveals the message m.

$$\begin{array}{rl} \textit{fun} & \mathsf{commit}/2. \\ \textit{reduc} & \mathsf{open}(\mathsf{commit}(m,r),r) = m. \end{array}$$

Zero-knowledge proofs. Non-interactive zero-knowledge proofs can be modelled as function $\mathsf{zk}(\mathit{secrets}, \mathit{pub\_info})$ (a function with two parameters: a tuple of secret information $\mathit{secrets}$, and a tuple of public information $\mathit{pub\_info}$) inspired by [BMU08]. The verifying information and the secret information satisfies a relation. Since the secret information is only known by the prover, only the prover can construct the zero-knowledge proof. To verify a zero-knowledge proof is to check whether the relation between the secret formation and the verifying formation is satisfied. The verification of a zero-knowledge proof is modelled as function $\mathsf{Vfy\text{-}zk}(\mathsf{zk}(\mathit{secrets}, \mathit{pub\_info}), \mathit{verif\_info})$, in which two parameters are: a zero-knowledge proof to be verified $\mathsf{zk}(\mathit{secrets}, \mathit{pub\_info})$ and the verification information $\mathit{verif\_info}$. Compared to that in [BMU08], we define each zero-knowledge specifically, as only a limited number of zero-knowledge proofs are used in the protocol. We specify each verification rule as in Figure 5.7. Since the $\mathit{pub\_info}$ and $\mathit{verif\_info}$ happens to be the same in all the zero-knowledge proofs verifications in this protocol, the generic structure of verification rule is as

$$\mathsf{Vfy\text{-}zk}(\mathsf{zk}(\mathit{secrets}, \mathit{pub\_info}), \mathit{pub\_info}) = \mathsf{true},$$

where true is a constant. The specific function to check a zero-knowledge proof of type z is denoted as $\mathsf{Vfy\text{-}zk}_z$, e.g., verification of a patient's anonymous authentication is modelled by function $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}$.

Digital credentials. A digital credential is issued by trusted authorities. We assume the procedure of issuing a credential is perfect, which means that the adversary cannot forge a credential nor obtain one by impersonation. We model digital credentials as a private function (declaimed by key word *private fun* in ProVerif) which is only usable by honest users. In the DLV08 protocol, a credential can have several attributes; we model these as parameters of the credential function.

$$\textit{private fun } \mathsf{drcred}/2. \qquad \textit{private fun } \mathsf{ptcred}/5.$$

There are two credentials in the DLV08 protocol: a doctor credential which is modelled as $\mathit{Cred}_{dr} := \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})$, and a patient credential which is modelled as $\mathit{Cred}_{pt} := \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})$.

Anonymous authentication. The procedure of anonymous authentication is a zero-knowledge proof using the digital credential as public information. The anonymous authentication of a doctor is modelled as

$$\mathit{Auth}_{dr} := \mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),$$

and the verification of the authentication is modelled as

$$\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{dr}}(\mathit{Auth}_{dr}, \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})).$$

The equational theory for the verification is

$$\begin{aligned} \textit{reduc}\ \ \mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{dr}}(\ & \mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),\\ & \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})) = \mathsf{true}. \end{aligned}$$

The verification implies that the creator of the authentication is a doctor, because only doctors can use the function drcred, and thus create a valid proof. The adversary can observe a credential $\mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})$, but does not know secrets $\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}$, and thus cannot forge a valid zero-knowledge proof. If the adversary forges a zero-knowledge proof with fake secret information $\mathtt{Pnym}_{dr}'$ and $\mathtt{Id}_{dr}'$, the fake zero-knowledge proof will not pass verification. For the same reason, a validated proof proves that the credential belongs to the creator of the zero-knowledge proof. Similarly, an anonymous authentication of a patient is modelled as

$$\begin{aligned} \mathit{Auth}_{pt} := \mathsf{zk}(\ & (\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\ & \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})), \end{aligned}$$

and the verification rule is modelled as

$$\begin{aligned} \textit{reduc}\ \mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}(\ & \mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\ & \quad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})),\\ & \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})) = \mathsf{true}. \end{aligned}$$

Verifiable encryptions. A verifiable encryption is modelled as a zero-knowledge proof. The encryption is embedded in the zero-knowledge proof as public function. The receiver can obtain the cipher text from the proof. For example, a patient wants to prove that he encrypted a secret s using a public key k to a pharmacist, while the pharmacist does not know the corresponding secret key for k. The pharmacist cannot open the cipher text to test whether it uses the public key k to encrypt. However, the zero-knowledge proof can prove that the cipher text is encrypted using k, while not revealing s. The general structure of the verification of a verifiable encryption is

$$\textsf{Vfy-venc}(\textsf{zk}(\mathit{secrets}, (\mathit{pub\_info}, \mathit{cipher})), \mathit{verif\_info}) = \textsf{true},$$

where *secrets* is private information, *pub_info* and *cipher* consist public information, *verif_info* is the verification information.

Signed proofs of knowledge. A signed proof of knowledge is a scheme which signs a message, and proves a property of the signer. For the DLV08 protocol, this proof only concerns equality of attributes of credentials and commitments (e.g. the identify of this credential is the same as the identity of that commitment). To verify a signed proof of knowledge, the verifier must know which credentials/commitments are considered. Hence, this information must be obtainable from the proof, and thus is included in the model. In general, a signed proof of knowledge is modelled as function

$$\textsf{spk}(\mathit{secrets}, \mathit{pub\_info}, \mathit{msg}),$$

which models a signature using private value(s) *secrets* on the message *msg*, with public information *pub_info* as settings. What knowledge is proven, depends on the specific instance of the proof and is captured by the verification functions for the specific proofs. For example, to prove that a user knows a) all fields of a (simplified) credential, b) all fields of a commitment to an identity, and c) that the commitment concerns the same identity as the commitment, he generates the following proof:

$$\begin{array}{lll}
\textsf{spk}(& (\texttt{Id}_{pt}, \texttt{Pnym}_{pt}, \texttt{r}_{pt}), & (*\mathit{secrets}*) \\
& (\textsf{ptcred}(\texttt{Id}_{pt}, \texttt{Pnym}_{pt}), \textsf{commit}(\texttt{Id}_{pt}, \texttt{r}_{pt})), & (*\mathit{pubublic\_info}*) \\
& \mathit{msg}). & (*\mathit{message}*)
\end{array}$$

These proofs are verified by checking that the signature is correct, given the signed message and the verification information. E.g., the above example proof can be verified as follows:

$$\begin{array}{lll}
\mathit{reduc}\ \textsf{Vfy-spk}(& \textsf{spk}(\ (\texttt{Id}_{pt}, \texttt{Pnym}_{pt}, \texttt{r}_{pt}), & \\
& \quad\ (\textsf{ptcred}(\texttt{Id}_{pt}, \texttt{Pnym}_{pt}), \textsf{commit}(\texttt{Id}_{pt}, \texttt{r}_{pt})), & \\
& \quad\ \mathit{msg}\ ), & (*\mathit{signed\_message}*) \\
& (\ \textsf{ptcred}(\texttt{Id}_{pt}, \texttt{Pnym}_{pt}), \textsf{commit}(\texttt{Id}_{pt}, \texttt{r}_{pt})\ ), & (*\mathit{verify\_info}*) \\
& \mathit{msg} & (*\mathit{message}*) \\
&) = \textsf{true}. &
\end{array}$$

fun	true/0.	*fun*	hash/3.	*fun*	pk/1.
fun	enc/2.	*fun*	commit/2.	*fun*	sign/2.
private fun	drcred/2.	*private fun*	ptcred/5.	*fun*	zk/2.
fun	spk/3.	*fun*	invoice/1.	*fun*	key/1.
fun	host/1.				

Figure 5.5: Functions.

reduc	$\mathsf{dec}(\mathsf{enc}(m, \mathsf{pk}(sk)), sk) = m.$
reduc	$\mathsf{open}(\mathsf{commit}(x, y), y) = x.$
reduc	$\mathsf{Vfy\text{-}sign}(\mathsf{sign}(x, y), \mathsf{pk}(y)) = \mathsf{true}.$
reduc	$\mathsf{getsignmsg}(\mathsf{sign}(x, y), \mathsf{pk}(y)) = x.$
reduc	$\mathsf{getpublic}(\mathsf{zk}(x, y)) = y.$
reduc	$\mathsf{getmsg}(\mathsf{spk}(x, y, z)) = z.$
reduc	$\mathsf{getSpkVinfo}(\mathsf{spk}(x, y, z)) = y.$
equation	$\mathsf{key}(\mathsf{host}(x)) = x.$
equation	$\mathsf{host}(\mathsf{key}(x)) = x.$

Figure 5.6: Equational theory part I: non-zero-knowledge part.

Other cryptographic primitives. Hash functions, encryptions and signing messages are modelled by functions hash, enc, and sign, respectively (Figure 5.5). Correspondingly, decryption and retrieving the message from a signature are modelled as functions dec and getsignmsg (Figure 5.6). Functions getpublic, getSpkVinfo and getmsg model retrieving public information from a zero-knowledge proof, from a signed proof of knowledge, and obtaining the message from a signed proof of knowledge, respectively (Figure 5.6).

5.5.3 Modelling the DLV08 protocol

Modelling the doctor-patient sub-protocol. This sub-protocol is used for a doctor, whose steps are labelled **d***i* in Figure 5.8, to prescribe medicine for a patient, whose steps are labelled **t***i* in Figure 5.9.

First, the doctor anonymously authenticates to the patient using credential $Cred_{dr}$ (**d1**). The patient reads in the doctor authentication (**t1**), obtains the doctor credential (**t2**), and verifies the authentication (**t3**). If the verification in step (**t3**) succeeds, the patient anonymously authenticates himself to the doctor using his credential (**t5**, the first zk function), generates a nonce $\mathtt{r}_{pt}$ (**t4**), computes a commitment with the nonce as opening information, and proves that the patient identity used in the patient credential is the same as in the commitment, thus linking the patient commitment and the patient credential (**t5**, the second zk).

The doctor reads in the patient authentication as rcv_Auth_{pt} and the patient proof as $rcv\_PtProof$ (**d2**), obtains the patient credential from the patient authentication (**d3**), obtains the patient commitment c_Comt_{pt} and the patient credential from the patient proof, tests whether the credential matches the one embedded in the patient authentication (**d4**), then verifies the authentication (**d5**) and the patient

$$
\begin{aligned}
&\mathit{reduc}\quad \mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{dr}}(\mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),\\
&\qquad\qquad \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}(\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}zk}_{\mathsf{PtProof}}(\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad (\mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}),\\
&\qquad\qquad\quad\ \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}))),\\
&\qquad\qquad \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}spk}_{\mathsf{PrescProof}}(\mathsf{spk}((\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}),\\
&\qquad\qquad\quad (\mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),\\
&\qquad\qquad\quad (\mathtt{presc}, \mathit{PrescriptID}, \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}),\\
&\qquad\qquad\quad\ \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}))),\\
&\qquad\qquad \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathtt{presc}, \mathit{PrescriptID},\\
&\qquad\qquad \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt})) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}zk}_{\mathsf{PtAuthSss}}(\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad (\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathtt{Sss}),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathtt{Sss}) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}spk}_{\mathsf{PtSpk}}(\mathsf{spk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}, \mathtt{r}_{pt}),\\
&\qquad\qquad\quad (\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt})),\\
&\qquad\qquad\quad \mathtt{nonce}),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}), \mathtt{nonce}) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}venc}_{\mathsf{Hii}}(\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad (\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad\ \mathsf{enc}(\mathtt{Hii}, pk_x))),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathsf{enc}(\mathtt{Hii}, pk_x), pk_x) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}venc}_{\mathsf{DrnymMpa}}(\mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}),\\
&\qquad\qquad\quad (\mathsf{spk}((\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}),\\
&\qquad\qquad\qquad (\mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),\\
&\qquad\qquad\qquad (\mathtt{presc}, \mathit{PrescriptID},\\
&\qquad\qquad\qquad\ \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), c_{ph}_Comt_{pt})),\\
&\qquad\qquad\quad\ \mathsf{enc}(\mathtt{Pnym}_{dr}, pk_x))),\\
&\qquad\qquad \mathsf{spk}((\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}),\\
&\qquad\qquad\quad (\mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),\\
&\qquad\qquad\quad (\mathtt{presc}, \mathit{PrescriptID},\\
&\qquad\qquad\quad\ \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), c_{ph}_Comt_{pt})),\\
&\qquad\qquad \mathsf{enc}(\mathtt{Pnym}_{dr}, pk_x), pk_x) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}venc}_{\mathsf{Ptnym}}(\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad (\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathsf{enc}(\mathtt{Pnym}_{pt}, pk_x))),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad \mathsf{enc}(\mathtt{Pnym}_{pt}, pk_x), pk_x) = \mathsf{true}.\\
&\mathit{reduc}\quad \mathsf{Vfy\text{-}spk}_{\mathsf{ReceiptAck}}(\mathsf{spk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad\quad (c_PrescriptID, c_{pt}_Id_{ph}, vc_1, vc_2, vc_3, vc'_3, vc_4, c_5)),\\
&\qquad\qquad \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\
&\qquad\qquad c_PrescriptID, c_{pt}_Id_{ph}, vc_1, vc_2, vc_3, vc'_3, vc_4, c_5) = \mathsf{true}.
\end{aligned}
$$

Figure 5.7: Equational theory part II: zero-knowledge part.

```
        $P_{dr} :=$
d1.     $\mathsf{out}(\mathtt{ch}, \mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})))$.
d2.     $\mathsf{in}(\mathtt{ch}, (\mathit{rcv\_Auth}_{pt}, \mathit{rcv\_PtProof}))$.
d3.     let $c\_Cred_{pt} = \mathsf{getpublic}(\mathit{rcv\_Auth}_{pt})$ in
d4.     let $(c\_Comt_{pt}, = c\_Cred_{pt}) = \mathsf{getpublic}(\mathit{rcv\_PtProof})$ in
d5.     if $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}(\mathit{rcv\_Auth}_{pt}, c\_Cred_{pt}) = \mathtt{true}$ then
d6.     if $\mathsf{Vfy\text{-}zk}_{\mathsf{PtProof}}(\mathit{rcv\_PtProof}, (c\_Comt_{pt}, c\_Cred_{pt})) = \mathtt{true}$ then
d7.     $\nu\mathtt{presc}$.
d8.     $\nu\mathtt{r}_{dr}$.
d9.     let $\mathit{PrescriptID} = \mathsf{hash}(\mathtt{presc}, c\_Comt_{pt}, \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}))$ in
d10.    $\mathsf{out}(\mathtt{ch}, (\mathsf{spk}((\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}),$
                    $(\mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})),$
                    $(\mathtt{presc}, \mathit{PrescriptID}, \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), c\_Comt_{pt})),$
                $\mathtt{r}_{dr}))$
```

Figure 5.8: The doctor process P_{dr}.

proof (**d6**). If the verification in the previous item succeeds, the doctor generates a prescription presc (**d7**), generates a nonce $\mathtt{r}_{dr}$ (**d8**), computes a prescription identity *PrescriptID* (**d9**), and computes a commitment $Comt_{dr}$ using the nonce as opening information (**d10**). Note that a medical examination of the patient is not part of the DLV08 protocol. Next, the doctor signs the message (presc, *PrescriptID*, $Comt_{dr}$, c_Comt_{pt}) using a signed proof of knowledge. This proves the pseudonym used in the credential $Cred_{dr}$ is the same as in the commitment $Comt_{dr}$, thus linking the prescription to the credential. The doctor sends the signed proof of knowledge together with the open information of the doctor commitment $\mathtt{r}_{dr}$ (**d10**).

The patient reads in the prescription as *rcv_PrescProof* and the opening information of the doctor commitment (**t6**), obtains the prescription *c_presc*, prescription identity *c_PrescriptID*, doctor commitment c_Comt_{dr}, and tests the patient commitment signed in the receiving message (**t7**). Then the patient verifies the signed proof of prescription (**t8**). If the verification succeeds, the patient obtains the doctor's pseudonym c_Pnym_{dr} by opening the doctor commitment (**t9**).

Modelling the patient-pharmacist sub-protocol. This sub-protocol is used for a patient, whose steps are labelled **t***i* in Figure 5.10, to obtain medicine from a pharmacist, whose steps are labelled **h***i* in Figure 5.11.

First, the pharmacist authenticates to the patient using a public key authentication (**h1**). Note that the pharmacist does not authenticate anonymously, and that the pharmacists's MPA identity is embedded. The patient reads in the pharmacist authentication rcv_Auth_{ph} (**t10**) and verifies the authentication (**t11**). If the verification succeeds, the pharmacist obtains the pharmacist's MPA identity from the authentication (**t12**), thus obtains the public key of MPA (**t13**). Then the patient anonymously authenticates himself to the pharmacist, and proves his social security status using the proof *PtAuthSss* (**t14**). The patient generates a nonce which will be used as a message in a signed proof of knowledge (**t15**), and

```
$P_{pt\_}p_1 :=$
t1.   $\mathsf{in}(\mathtt{ch}, rcv\_Auth_{dr})$.
t2.   let $c\_Cred_{dr} = \mathsf{getpublic}(rcv\_Auth_{dr})$ in
t3.   if $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{dr}}(rcv\_Auth_{dr}, c\_Cred_{dr}) = \mathtt{true}$ then
t4.   $\nu \mathtt{r}_{pt}$.
t5.   $\mathsf{out}(\mathtt{ch}, (\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),$
                  $\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})),$
              $\mathsf{zk}((\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),$
                  $(\mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}),$
                   $\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc})))))$.
t6.   $\mathsf{in}(\mathtt{ch}, (rcv\_PrescProof, rcv\_r_{dr}))$.
t7.   let $(c\_presc, c\_PrescriptID, c\_Comt_{dr}, = \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}))$
         $= \mathsf{getmsg}(rcv\_PrescProof)$ in
t8.   if $\mathsf{Vfy\text{-}spk}_{\mathsf{PrescProof}}(rcv\_PrescProof, (c\_Cred_{dr}, c\_presc,$
           $c\_PrescriptID, c\_Comt_{dr}, \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt}))) = \mathtt{true}$ then
t9.   let $c\_Pnym_{dr} = \mathsf{open}(c\_Comt_{dr}, rcv\_r_{dr})$ in 0
```

Figure 5.9: The patient process in the doctor-patient sub-protocol $P_{pt\_}p_1$.

computes verifiable encryptions vc_1, vc_2, vc_3, vc_3', vc_4 and vc_5 (**t16-t21**). These divulge the patient's HII, the doctor's pseudonym, and the patient's pseudonym to the MPA, the patient's pseudonym to the HII, and the patient pseudonym and HII to the social safety organisation, respectively. The patient encrypts vc_5 with MPA's public key as c_5 (**t22**). The patient computes a signed proof of knowledge

$$\begin{aligned} PtSpk = \mathsf{spk}(&(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}),\\ &(\mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt})),\\ &\mathtt{nonce}) \end{aligned}$$

which proves that the patient identity embedded in the prescription is the same as in his credential. In the prescription, this identity is contained in a commitment. For simplicity, we model the proof using the commitment instead of the prescription. The link between commitment and prescription is ensured when the proof is verified (**h10**).

The patient sends the prescription *rcv_PrescProof*, the signed proof *PtSpk*, and $vc_1, vc_2, vc_3, vc_3', vc_4, c_5$ to the pharmacist (**t23**). The pharmacist reads in the authentication *rcv_PtAuthSss* (**h2**), obtains the patient credential and his social security status (**h3**), verifies the authentication (**h4**). If the verification succeeds, the pharmacist reads in the patient's prescription $rcv_{ph}\_PrescProof$, the signed proof of knowledge rcv_{ph}_PtSpk, the verifiable encryptions rcv_vc_1, rcv_vc_2, rcv_vc_3, rcv_vc_3', rcv_vc_4, and cipher text rcv_c_5 (**h5**); and verifies $rcv_{ph}\_PrescProof$ (**h6-h8**), rcv_{ph}_PtSpk (**h9-h10**), and rcv_vc_1, rcv_vc_2, rcv_vc_3, rcv_vc_3', rcv_vc_4 (**h11-h20**). If all the verifications succeed, the pharmacist charges the patient, and delivers the medicine (neither are modelled as they are out of DLV08's scope). Then the pharmacist generates an invoice with the prescription identity embedded in it and sends the invoice to the patient (**h21**).

The patient reads in the invoice (**t24**), computes a receipt: a signed proof of knowledge *ReceiptAck* which proves that he receives the medicine (**t25**); and sends

```
      $P_{pt\_p_2} :=$
t10.     in(ch, $rcv\_Auth_{ph}$).
t11.     if Vfy-sign($rcv\_Auth_{ph}$, $rcv_{pt}\_pk_{ph}$) = true then
t12.     let (= $c_{pt}\_Id_{ph}$, $c_{pt}\_Id_{mpa}$)
                         = getsignmsg($rcv\_Auth_{ph}$, $rcv_{pt}\_pk_{ph}$) in
t13.     let $c_{pt}\_pk_{mpa}$ = key($c_{pt}\_Id_{mpa}$) in
t14.     out(ch, zk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                 (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc), Sss))).
t15.     $\nu$nonce.
t16.     let $vc_1$ = zk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                   (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                    enc(Hii, $c_{pt}\_pk_{mpa}$))) in
t17.     let $vc_2$ = zk(($c\_Pnym_{dr}$, $rcv\_r_{dr}$),
                   ($rcv\_PrescProof$, enc($c\_Pnym_{dr}$, $c_{pt}\_pk_{mpa}$))) in
t18.     let $vc_3$ = zk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                   (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                    enc(Pnym$_{pt}$, pk$_{sso}$))) in
t19.     let $vc_3'$ = zk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                   (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                    enc(Hii, pk$_{sso}$))) in
t20.     let $vc_4$ = zk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                   (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                    enc(Pnym$_{pt}$, $c_{pt}\_pk_{mpa}$))) in
t21.     let $vc_5$ = zk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                   (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                    enc(Pnym$_{pt}$, $c_{pt}\_pk_{hii}$))) in
t22.     let $c_5$ = enc($vc_5$, $c_{pt}\_pk_{mpa}$) in
t23.     out(ch, ($rcv\_PrescProof$, spk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
         (ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc), commit(Id$_{pt}$, r$_{pt}$)), nonce),
                                        $vc_1$, $vc_2$, $vc_3$, $vc_3'$, $vc_4$, $c_5$)).
t24.     in(ch, $rcv\_Invoice$).
t25.     let $ReceiptAck$ = spk((Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
                         ptcred(Id$_{pt}$, Pnym$_{pt}$, Hii, Sss, Acc),
              ($c\_PrescriptID$, $c_{pt}\_Id_{ph}$, $vc_1$, $vc_2$, $vc_3$, $vc_3'$, $vc_4$, $c_5$)) in
t26.     out(ch, $ReceiptAck$)
```

Figure 5.10: The patient process in the patient-pharmacist sub-protocol $P_{pt\_p_2}$.

the signed proof of knowledge to the patient (**t26**). The pharmacist reads in the receipt *rcv_ReceiptAck* (**h22**) and verifies its correctness (**h23**).

Modelling the pharmacist-MPA sub-protocol. The pharmacist-MPA sub-protocol is used for the pharmacist, whose steps are labelled **h***i* in Figure 5.13 to report the received prescriptions to the MPA, whose steps are labelled **m***i* in Figure 5.12.

As the pharmacist mostly forwards the information supplied by the patient, this protocol greatly resembles the patient-pharmacist protocol described above. Each

```
$P_{ph}\_p_1 :=$
h1.   $\mathsf{out}(\mathsf{ch}, \mathsf{sign}((\mathtt{Id}_{ph}, c_{ph}\_Id_{mpa}), \mathtt{sk}_{ph})).$
h2.   $\mathsf{in}(\mathsf{ch}, rcv\_PtAuthSss).$
h3.   $\mathtt{let}\ (c_{ph}\_Cred_{pt}, c_{ph}\_Sss) = \mathsf{getpublic}(rcv\_PtAuthSss)\ \mathtt{in}$
h4.   $\mathtt{if}\ \mathsf{Vfy\text{-}zk}_{\mathsf{PtAuthSss}}(rcv\_PtAuthSss, (c_{ph}\_Cred_{pt}, c_{ph}\_Sss))$
                                                        $= \mathtt{true}\ \mathtt{then}$
h5.   $\mathsf{in}(\mathsf{ch}, (rcv_{ph}\_PrescProof, rcv_{ph}\_PtSpk,$
          $rcv\_vc_1, rcv\_vc_2, rcv\_vc_3, rcv\_vc'_3, rcv\_vc_4, rcv\_c_5)).$
h6.   $\mathtt{let}\ (c_{ph}\_Comt_{dr}, c_{ph}\_Cred_{dr})$
                               $= \mathsf{getSpkVinfo}(rcv_{ph}\_PrescProof)\ \mathtt{in}$
h7.   $\mathtt{let}\ (c_{ph}\_presc, c_{ph}\_PrescriptID, = c_{ph}\_Comt_{dr}, c_{ph}\_Comt_{pt})$
                                  $= \mathsf{getmsg}(rcv_{ph}\_PrescProof)\ \mathtt{in}$
h8.   $\mathtt{if}\ \mathsf{Vfy\text{-}spk}_{\mathsf{PrescProof}}(rcv_{ph}\_PrescProof, (c_{ph}\_Cred_{dr}, c_{ph}\_presc,$
            $c_{ph}\_PrescriptID, c_{ph}\_Comt_{dr}, c_{ph}\_Comt_{pt})) = \mathtt{true}\ \mathtt{then}$
h9.   $\mathtt{let}\ c\_msg = \mathsf{getmsg}(rcv_{ph}\_PtSpk)\ \mathtt{in}$
h10.  $\mathtt{if}\ \mathsf{Vfy\text{-}spk}_{\mathsf{PtSpk}}(rcv_{ph}\_PtSpk,$
                     $(c_{ph}\_Cred_{pt}, c_{ph}\_Comt_{pt}, c\_msg)) = \mathtt{true}\ \mathtt{then}$
h11.  $\mathtt{let}\ (= c_{ph}\_Cred_{pt}, c\_Enc_1) = \mathsf{getpublic}(rcv\_vc_1)\ \mathtt{in}$
h12.  $\mathtt{if}\ \mathsf{Vfy\text{-}venc}_{\mathsf{Hii}}(rcv\_vc_1, (c_{ph}\_Cred_{pt}, c\_Enc_1, rcv_{ph}\_pk_{mpa}))$
                                                        $= \mathtt{true}\ \mathtt{then}$
h13.  $\mathtt{let}\ (= rcv_{ph}\_PrescProof, c\_Enc_2) = \mathsf{getpublic}(rcv\_vc_2)\ \mathtt{in}$
h14.  $\mathtt{if}\ \mathsf{Vfy\text{-}venc}_{\mathsf{DrnymMpa}}(rcv\_vc_2, (rcv_{ph}\_PrescProof,$
                                  $c\_Enc_2, rcv_{ph}\_pk_{mpa})) = \mathtt{true}\ \mathtt{then}$
h15.  $\mathtt{let}\ (= c_{ph}\_Cred_{pt}, c\_Enc_3) = \mathsf{getpublic}(rcv\_vc_3)\ \mathtt{in}$
h16.  $\mathtt{if}\ \mathsf{Vfy\text{-}venc}_{\mathsf{Ptnym}}(rcv\_vc_3, (c_{ph}\_Cred_{pt}, c\_Enc_3, \mathsf{pk}_{sso}))$
                                                        $= \mathtt{true}\ \mathtt{then}$
h17.  $\mathtt{let}\ (= c_{ph}\_Cred_{pt}, c\_Enc'_3) = \mathsf{getpublic}(rcv\_vc'_3)\ \mathtt{in}$
h18.  $\mathtt{if}\ \mathsf{Vfy\text{-}venc}_{\mathsf{Hii}}(rcv\_vc'_3, (c_{ph}\_Cred_{pt}, c\_Enc'_3, \mathsf{pk}_{sso})) = \mathtt{true}\ \mathtt{then}$
h19.  $\mathtt{let}\ (= c_{ph}\_Cred_{pt}, c\_Enc_4) = \mathsf{getpublic}(rcv\_vc_4)\ \mathtt{in}$
h20.  $\mathtt{if}\ \mathsf{Vfy\text{-}venc}_{\mathsf{Ptnym}}(rcv\_vc_4,$
                     $(c_{ph}\_Cred_{pt}, c\_Enc_4, rcv_{ph}\_pk_{mpa})) = \mathtt{true}\ \mathtt{then}$
h21.  $\mathsf{out}(\mathsf{ch}, \mathsf{inv}(c_{ph}\_PrescriptID)).$
h22.  $\mathsf{in}(\mathsf{ch}, rcv\_ReceiptAck).$
h23.  $\mathtt{if}\ \mathsf{Vfy\text{-}spk}_{\mathsf{ReceiptAck}}(rcv\_ReceiptAck, (c_{ph}\_Cred_{pt}, c_{ph}\_PrescriptID,$
        $\mathtt{Id}_{ph}, rcv\_vc_1, rcv\_vc_2, rcv\_vc_3, rcv\_vc'_3, rcv\_vc_4, rcv\_c_5)) = \mathtt{true}$
       $\mathtt{then}\ 0$
```

Figure 5.11: The pharmacist process in the patient-pharmacist sub-protocol $P_{ph}\_p_1$.

step is modelled in details as follows: The pharmacist authenticates himself to his MPA by sending his identity and the signed identities of the pharmacist and the MPA (**h24**). The MPA stores this authentication in $rcv_{mpa}\_Auth_{ph}$, and stores the pharmacist's identity in $c_{mpa}\_Id_{ph}$ (**m1**). From the pharmacist's identity, the MPA obtains the pharmacist's public key (**m2**). Then the MPA verifies the pharmacist's authentication against the pharmacist's public key (**m3**). If the verification succeeds, according to the corresponding rule in the equational theory,

```
$P_{mpa}\_p_1 :=$
m1.   $\mathsf{in}(\mathsf{ch}, (rcv_{mpa}\_Auth_{ph}, c_{mpa}\_Id_{ph}))$.
m2.   let $rcv_{mpa}\_pk_{ph} = \mathsf{key}(c_{mpa}\_Id_{ph})$ in
m3.   if $\mathsf{Vfy\text{-}sign}(rcv_{mpa}\_Auth_{ph}, rcv_{mpa}\_pk_{ph}) = \mathsf{true}$ then
m4.   let $(= c_{mpa}\_Id_{ph}, = \mathtt{Id}_{mpa})$
                  $= \mathsf{getsignmsg}(rcv_{mpa}\_Auth_{ph}, rcv_{mpa}\_pk_{ph})$ in
m5.   $\mathsf{out}(\mathsf{ch}, \mathsf{sign}(\mathtt{Id}_{mpa}, \mathtt{sk}_{mpa}))$.
m6.   $\mathsf{in}(\mathsf{ch}, (rcv_{mpa}\_PrescProof, rcv_{mpa}\_vc_1, rcv_{mpa}\_vc_2, rcv_{mpa}\_vc_3,$
          $rcv_{mpa}\_vc'_3, rcv_{mpa}\_vc_4, rcv_{mpa}\_c_5, rcv_{mpa}\_ReceiptAck))$.
m7.   let $(c_{mpa}\_Comt_{dr}, c_{mpa}\_Cred_{dr})$
                  $= \mathsf{getSpkVinfo}(rcv_{mpa}\_PrescProof)$ in
m8.   let $(c_{mpa}\_presc, c_{mpa}\_PrescriptID, = c_{mpa}\_Comt_{dr}, c_{mpa}\_Comt_{pt})$
                  $= \mathsf{getmsg}(rcv_{mpa}\_PrescProof)$ in
m9.   if $\mathsf{Vfy\text{-}spk}_{\mathsf{PrescProof}}(rcv_{mpa}\_PrescProof, (c_{mpa}\_Cred_{dr}, c_{mpa}\_presc,$
          $c_{mpa}\_PrescriptID, c_{mpa}\_Comt_{dr}, c_{mpa}\_Comt_{pt})) = \mathsf{true}$ then
m10.  let $(= c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc_1) = \mathsf{getpublic}(rcv_{mpa}\_vc_1)$ in
m11.  if $\mathsf{Vfy\text{-}venc}_{\mathsf{Hii}}(rcv_{mpa}\_vc_1,$
                  $(c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc_1, \mathsf{pk}_{mpa})) = \mathsf{true}$ then
m12.  let $c_{mpa}\_Hii = \mathsf{dec}(c_{mpa}\_Enc_1, \mathtt{sk}_{mpa})$ in
m13.  let $(= rcv_{mpa}\_PrescProof, c_{mpa}\_Enc_2)$
                  $= \mathsf{getpublic}(rcv_{mpa}\_vc_2)$ in
m14.  if $\mathsf{Vfy\text{-}venc}_{\mathsf{DrnymMpa}}(rcv_{mpa}\_vc_2,$
                  $(rcv_{mpa}\_PrescProof, c_{mpa}\_Enc_2, \mathsf{pk}_{mpa})) = \mathsf{true}$ then
m15.  let $c_{mpa}\_Pnym_{dr} = \mathsf{dec}(c_{mpa}\_Enc_2, \mathtt{sk}_{mpa})$ in
m16.  let $(= c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc3) = \mathsf{getpublic}(rcv_{mpa}\_vc_3)$ in
m17.  if $\mathsf{Vfy\text{-}venc}_{\mathsf{Ptnym}}(rcv_{mpa}\_vc_3,$
                  $(c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc_3, \mathsf{pk}_{sso})) = \mathsf{true}$ then
m19.  let $(= c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc'_3) = \mathsf{getpublic}(rcv_{mpa}\_vc'_3)$ in
m20.  if $\mathsf{Vfy\text{-}venc}_{\mathsf{Hii}}(rcv_{mpa}\_vc'_3,$
                  $(c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc'_3, \mathsf{pk}_{sso})) = \mathsf{true}$ then
m21.  let $(= c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc_4) = \mathsf{getpublic}(rcv_{mpa}\_vc_4)$ in
m22.  if $\mathsf{Vfy\text{-}venc}_{\mathsf{Ptnym}}(rcv_{mpa}\_vc_4,$
                  $(c_{mpa}\_Cred_{pt}, c_{mpa}\_Enc_4, \mathsf{pk}_{mpa})) = \mathsf{true}$ then
m23.  let $c_{mpa}\_Pnym_{pt} = \mathsf{dec}(c_{mpa}\_Enc_4, \mathtt{sk}_{mpa})$ in
m24.  if $\mathsf{Vfy\text{-}spk}_{\mathsf{ReceiptAck}}(rcv_{mpa}\_ReceiptAck, (c_{mpa}\_Cred_{pt},$
          $c_{mpa}\_PrescriptID, c_{mpa}\_Id_{ph}, rcv_{mpa}\_vc_1, rcv_{mpa}\_vc_2,$
          $rcv_{mpa}\_vc_3, rcv_{mpa}\_vc'_3, rcv_{mpa}\_vc_4, rcv_{mpa}\_c_5)) = \mathsf{true}$ then 0
```

Figure 5.12: The MPA process in the pharmacist-MPA sub-protocol $P_{mpa}\_p_1$.

and the MPA verifies that he is indeed the pharmacist's MPA (**m4**), the MPA then authenticates itself to the pharmacist by sending the signature of his identity (**m5**). The pharmacist reads in the MPA's authentication in rcv_Auth_{mpa} (**h25**), and verifies the authentication (**h26**). If the verification succeeds, the pharmacist sends the following to the MPA: prescription $rcv_{ph}\_PrescProof$, received receipt $rcv\_ReceiptAck$, and verifiable encryptions rcv_vc_1, rcv_vc_2, rcv_vc_3, $rcv\_vc'_3$, rcv_vc_4, rcv_c_5 (**h27**). The MPA reads in the information (**m6**) and ver-

```
        $P_{ph}\_p_2 :=$
h24.        $\mathsf{out}(\mathtt{ch}, (\mathsf{sign}((\mathtt{Id}_{ph}, c_{ph}\_Id_{mpa}), \mathtt{sk}_{ph}), \mathtt{Id}_{ph})).$
h25.        $\mathsf{in}(\mathtt{ch}, rcv\_Auth_{mpa}).$
h26.        if $\mathsf{Vfy\text{-}sign}(rcv\_Auth_{mpa}, rcv_{ph}\_pk_{mpa}) = \mathtt{true}$ then
h27.        $\mathsf{out}(\mathtt{ch}, (rcv_{ph}\_PrescProof,$
                $rcv\_vc_1, rcv\_vc_2, rcv\_vc_3, rcv\_vc_3', rcv\_vc_4, rcv\_c_5,$
                $rcv\_ReceiptAck))$
```

Figure 5.13: The pharmacist process in the pharmacist-MPA sub-protocol $P_{ph}\_p_2$.

ifies their correctness (**m7-m23**). If the verifications succeed, the MPA decrypts the corresponding encryptions ($c_{mpa}\_Enc_1$, $c_{mpa}\_Enc_2$, and $c_{mpa}\_Enc_4$) embedded in $rcv_{mpa}\_vc_1$, $rcv_{mpa}\_vc_2$, $rcv_{mpa}\_vc_4$, and obtains the patient's HII (**m12**), the doctor pseudonym (**m15**), the patient pseudonym (**m23**). The storing information to database by the MPA is beyond our concern.

Modelling the MPA-HII sub-protocol. This protocol covers the exchange of information between the pharmacist's MPA, whose steps are labelled **m***i* in Figure 5.14 and the patient's HII, whose steps are labelled **i***i* in Figure 5.15.

```
        $P_{mpa}\_p_2 :=$
m25.        $\mathsf{out}(\mathtt{ch}, (\mathsf{sign}(\mathtt{Id}_{mpa}, \mathtt{sk}_{mpa}), \mathtt{Id}_{mpa})).$
m26.        $\mathsf{in}(\mathtt{ch}, rcv_{mpa}\_Auth_{hii}).$
m27.        let $c_{mpa}\_pk_{hii} = \mathsf{key}(c_{mpa}\_Hii)$ in
m28.        if $\mathsf{Vfy\text{-}sign}(rcv_{mpa}\_Auth_{hii}, c_{mpa}\_pk_{hii}) = \mathtt{true}$ then
m29.        if $\mathsf{getsignmsg}(rcv_{mpa}\_Auth_{hii}, c_{mpa}\_pk_{hii}) = c_{mpa}\_Hii$ then
m30.        $\mathsf{out}(\mathtt{ch}, (rcv_{mpa}\_ReceiptAck, \mathsf{dec}(rcv_{mpa}\_c_5, \mathtt{sk}_{mpa}))).$
m31.        $\mathsf{in}(\mathtt{ch}, rcv_{mpa}\_Invoice)$
```

Figure 5.14: The MPA process in the MPA-HII sub-protocol $P_{mpa}\_p_2$.

The MPA sends his identity to the HII and authenticates to the HII using public key authentication (**m25**). The HII stores the MPA's identity in $rcv_{hii}\_Id_{mpa}$ and stores the authentication in $rcv_{hii}\_Auth_{mpa}$ (**i1**). From the MPA's identity, the HII obtains the MPA's public key (**i2**). Then the HII verifies the MPA's authentication (**i3**). If the verification succeeds, the HII authenticates to the MPA using public key authentication (**i4**). The MPA stores the authentication in $rcv_{mpa}\_Auth_{hii}$ (**m26**). Then the MPA obtains the HII's public key from the HII's identity (**m27**) and verifies the HII's authentication (**m28**). If the verification succeeds, and the MPA verifies that the authentication is from the intended HII (**m29**), the MPA sends the receipt $rcv_{mpa}\_PrescProof$ and the patient pseudonym encrypted for the HII – verifiable encryption $rcv_{mpa}\_vc_5 = \mathsf{dec}(rcv_{mpa}\_c_5, \mathtt{sk}_{mpa})$ (**m30**). The HII receives the receipt as $rcv_{hii}\_ReceiptAck$ and the encrypted patient pseudonym for the HII as $c_{hii}\_vc_5$ (**i5**). The HII verifies the above two pieces of information (**i6-i10**). If the verifications succeed, the HII decrypts the encryption $c_{hii}\_Enc_5$ which is embedded in $c_{hii}\_vc_5$, and obtains the patient's pseudonym (**i11**). Finally, the HII sends an invoice of the prescription identity to the MPA (**i12**). The MPA stores the invoice

```
      $P_{hii} :=$
i1.     $\mathsf{in}(\mathsf{ch}, (rcv_{hii}\_Auth_{mpa}, rcv_{hii}\_Id_{mpa}))$.
i2.     let $c_{hii}\_pk_{mpa} = \mathsf{key}(rcv_{hii}\_Id_{mpa})$ in
i3.     if $\mathsf{Vfy\text{-}sign}(rcv_{hii}\_Auth_{mpa}, c_{hii}\_pk_{mpa}) = \mathtt{true}$ then
i4.     $\mathsf{out}(\mathsf{ch}, \mathsf{sign}(\mathtt{Id}_{hii}, \mathtt{sk}_{hii}))$.
i5.     $\mathsf{in}(\mathsf{ch}, (rcv_{hii}\_ReceiptAck, c_{hii}\_vc_5))$.
i6.     let $c_{hii}\_Cred_{pt} = \mathsf{getSpkVinfo}(rcv_{hii}\_ReceiptAck)$ in
i7.     let $(c_{hii}\_PrescriptID, c_{hii}\_Id_{ph}, c_{hii}\_vc_1, c_{hii}\_vc_2, c_{hii}\_vc_3, c_{hii}\_vc'_3,$
                        $c_{hii}\_vc_4, c_{hii}\_c_5) = \mathsf{getmsg}(rcv_{hii}\_ReceiptAck)$ in
i8.     if $\mathsf{Vfy\text{-}spk}_{\mathsf{ReceiptAck}}(rcv_{hii}\_ReceiptAck, (c_{hii}\_Cred_{pt},$
            $c_{hii}\_PrescriptID, c_{hii}\_Id_{ph}, c_{hii}\_vc_1, c_{hii}\_vc_2, c_{hii}\_vc_3, c_{hii}\_vc'_3,$
            $c_{hii}\_vc_4, c_{hii}\_c_5)) = \mathtt{true}$ then
i9.     let $(= c_{hii}\_Cred_{pt}, c_{hii}\_Enc_5) = \mathsf{getpublic}(c_{hii}\_vc_5)$ in
i10.    if $\mathsf{Vfy\text{-}venc}_{\mathsf{Ptnym}}(c_{hii}\_vc_5, (c_{hii}\_Cred_{pt}, c_{hii}\_Enc_5, \mathsf{pk}_{hii})) = \mathtt{true}$ then
i11.    let $c_{hii}\_Pnym_{pt} = \mathsf{dec}(c_{hii}\_Enc_5, \mathsf{sk}_{hii})$ in
i12.    $\mathsf{out}(\mathsf{ch}, \mathsf{invoice}(c_{hii}\_PrescriptID))$
```

Figure 5.15: The HII process P_{hii}.

in $rcv_{mpa}\_Invoice$ (**m31**). Afterwards, the HII pays the MPA and updates the patient account. As before, handling payment and storing information are beyond the scope of the DLV08 protocol and therefore, we do not model this stage.

Composition. Finally, we compose the two parts of patient processes, the two parts of pharmacist processes and the two parts of the MPA processes. Since the DLV08 protocol works as four sub-protocols executing in order, as shown in Figure 5.16, we simply connect each two parts to compose them. We compose $P_{pt}\_p_1$ and $P_{pt}\_p_2$ to obtain P_{pt} (as shown in Figure 5.17), compose $P_{ph}\_p_1$ and $P_{ph}\_p_2$ to obtain P_{ph} (as shown in Figure 5.18), and compose $P_{mpa}\_p_1$ and $P_{mpa}\_p_2$ to obtain P_{mpa} (as shown in Figure 5.19).

The protocol. The DLV08 protocol is modelled as the five roles R_{dr}, R_{pt}, R_{ph}, R_{mpa}, and R_{hii} running in parallel.

$$P_{DLV08} := \nu\tilde{m}.init.(!R_{pt} \mid !R_{dr} \mid !R_{ph} \mid !R_{mpa} \mid !R_{hii})$$
$$init := \text{let } \mathtt{pk}_{sso} = \mathsf{pk}(\mathtt{sk}_{sso}) \text{ in } \mathsf{out}(\mathsf{ch}, \mathtt{pk}_{sso})$$

where $\nu\tilde{m}$ represents global secrets $\mathtt{sk}_{sso}$ and private channels $\mathtt{ch}_{hp}$, $\mathtt{ch}_{mp}$, $\mathtt{ch}_{phpt}$; process *init* initialises the settings of the protocol. Each role is of the form $R_i := init_i.!P_i$.

In summary, the DLV08 protocol is composed as shown in Figure 5.20. In the protocol model, the roles R_{dr}, R_{hii}, R_{pt}, R_{ph}, R_{mpa} are shown as in Figure 5.21, Figure 5.22, Figure 5.17, Figure 5.18, and Figure 5.19, respectively.

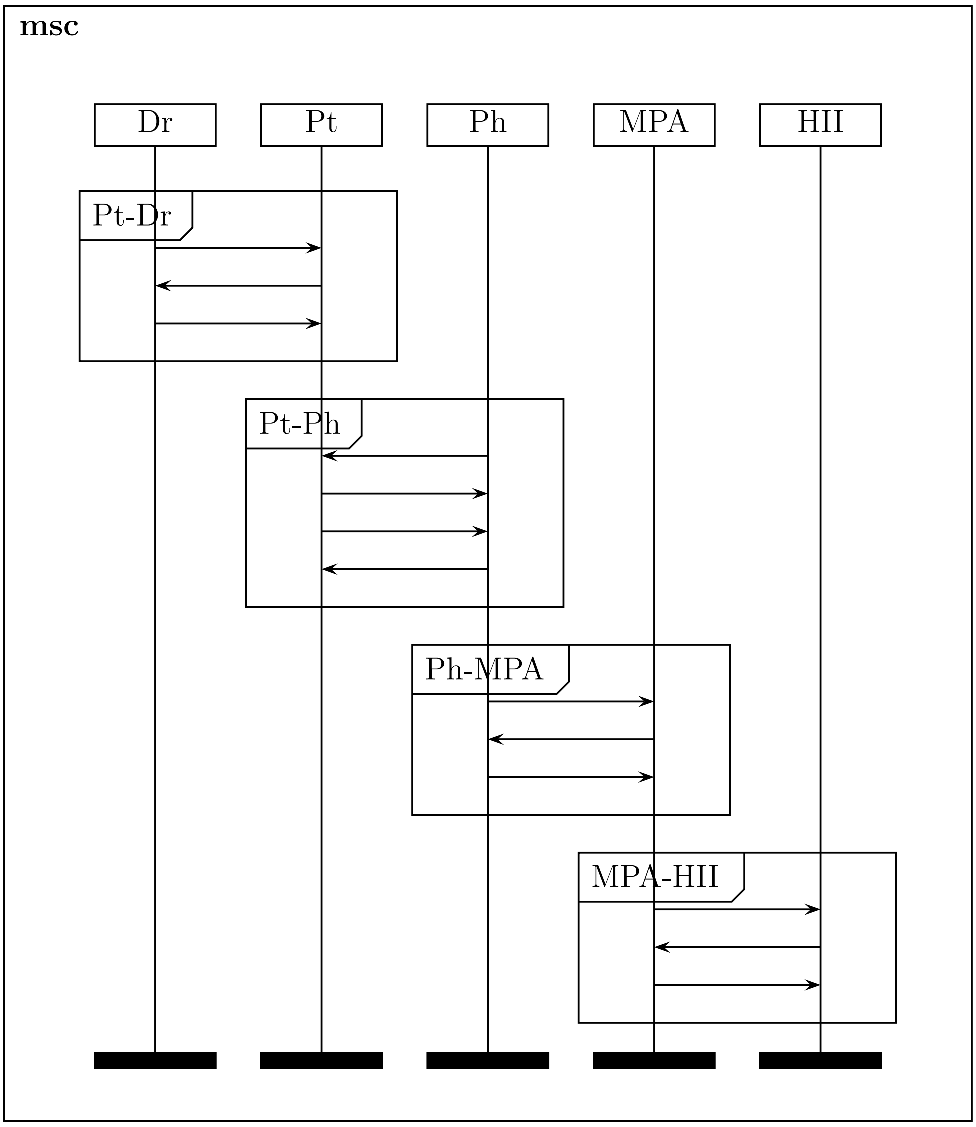

Figure 5.16: Overview of the DLV08 protocol.

$$
\begin{array}{l}
R_{pt} := \nu \mathtt{Id}_{pt}. \\
\qquad \left.\begin{array}{l} \nu\mathtt{Pnym}_{pt}.\nu\mathtt{Sss}.\nu\mathtt{Acc}. \\ \mathsf{in}(\mathsf{ch}_{hp},\mathtt{Hii}).\mathtt{let}\ c_{pt}\_pk_{hii} = \mathsf{key}(\mathtt{Hii})\ \mathtt{in} \end{array}\right\} init_{pt} \\
\qquad \left.\begin{array}{l} !(\mathsf{in}(\mathsf{ch}_{phpt}, rcv\_pk_{ph}). \\ \ \mathtt{let}\ rcv_{pt}\_pk_{ph} = rcv\_pk_{ph}\ \mathtt{in}\ \mathtt{let}\ \mathtt{Id}_{ph} = \mathsf{host}(rcv\_pk_{ph})\ \mathtt{in} \\ \ \left.\begin{array}{l} (\ldots \\ \mathtt{let}\ c\_Pnym_{dr} = \mathsf{open}(c\_Comt_{dr}, rcv\_r_{dr})\ \mathtt{in} \end{array}\right\} P_{pt}\_p_1 \\ \ \left.\begin{array}{l} \mathsf{in}(\mathsf{ch}, rcv\_Auth_{ph}). \\ \ldots) \end{array}\right\} P_{pt}\_p_2 \end{array}\right\} P_{pt}
\end{array}
$$

Figure 5.17: The process for role patient R_{pt}.

$$
\begin{array}{l}
R_{ph} := \nu \mathtt{sk}_{ph}.\\
\qquad \mathtt{let}\ \mathtt{pk}_{ph} = \mathsf{pk}(\mathtt{nsk}_{ph})\ \mathtt{in}\\
\qquad \mathtt{let}\ \mathtt{Id}_{ph} = \mathsf{host}(\mathtt{pk}_{ph})\ \mathtt{in}\\
\qquad (!\mathsf{out}(\mathtt{ch}, \mathtt{pk}_{ph})\ |!\mathsf{out}(\mathtt{ch}_{phpt}, \mathtt{pk}_{ph})\ |\\
\qquad \left.\begin{array}{l}
!(\mathsf{in}(\mathtt{ch}_{mp}, rcv_{ph}\_pk_{mpa}).\\
\quad \mathtt{let}\ c_{ph}\_Id_{mpa} = \mathsf{host}(rcv_{ph}\_pk_{mpa})\ \mathtt{in}\\
\quad \left.\begin{array}{l}
(\ldots\\
\ \mathtt{if}\ \mathsf{Vfy\text{-}spk}_{\mathsf{ReceiptAck}}(rcv\_ReceiptAck, (c_{ph}\_Cred_{pt},\\
\qquad c_{ph}\_PrescriptID, \mathtt{Id}_{ph}, rcv\_vc_1, rcv\_vc_2, rcv\_vc_3,\\
\qquad rcv\_vc'_3, rcv\_vc_4, rcv\_c_5)) = \mathsf{true}\ \mathtt{then}
\end{array}\right\} P_{ph}\text{-}p_1\\
\quad \left.\begin{array}{l}
\mathsf{out}(\mathtt{ch}, (\mathsf{sign}((\mathtt{Id}_{ph}, c_{ph}\_Id_{mpa}), \mathtt{sk}_{ph}), \mathtt{Id}_{ph})).\\
\ldots)))
\end{array}\right\} P_{ph}\text{-}p_2
\end{array}\right\} P_{ph}
\end{array}
$$

Figure 5.18: The process for role pharmacist R_{ph}.

$$
\begin{array}{l}
R_{mpa} := \nu \mathtt{sk}_{mpa}.\\
\qquad \left.\begin{array}{l}
\mathtt{let}\ \mathtt{pk}_{mpa} = \mathsf{pk}(\mathtt{sk}_{mpa})\ \mathtt{in}\\
\mathtt{let}\ \mathtt{Id}_{mpa} = \mathsf{host}(\mathtt{pk}_{mpa})\ \mathtt{in}\\
(!\mathsf{out}(\mathtt{ch}, \mathtt{pk}_{mpa})\ |!\mathsf{out}(\mathtt{ch}_{mp}, \mathtt{pk}_{mpa})\ |\\
\left.\begin{array}{l}
\ !(\ldots\\
\quad \mathtt{if}\ \mathsf{Vfy\text{-}spk}_{\mathsf{ReceiptAck}}(rcv_{mpa}\_ReceiptAck,\\
\qquad (c_{mpa}\_Cred_{pt}, c_{mpa}\_PrescriptID, c_{mpa}\_Id_{ph},\\
\qquad\qquad rcv_{mpa}\_vc_1, rcv_{mpa}\_vc_2, rcv_{mpa}\_vc_3,\\
\qquad\qquad rcv_{mpa}\_vc'_3, rcv_{mpa}\_vc_4, rcv_{mpa}\_c_5))\\
\qquad = \mathsf{true}\ \mathtt{then}
\end{array}\right\} P_{mpa}\text{-}p_1\\
\left.\begin{array}{l}
\quad \mathsf{out}(\mathtt{ch}, (\mathsf{sign}(\mathtt{Id}_{mpa}, \mathtt{sk}_{mpa}), \mathtt{Id}_{mpa})).\\
\quad \ldots))
\end{array}\right\} P_{mpa}\text{-}p_2
\end{array}\right\} P_{mpa}
\end{array}
$$

Figure 5.19: The process for role MPA R_{mpa}.

$$
\begin{array}{ll}
P_{DLV08} := \nu \mathtt{sk}_{sso}.\nu \mathtt{ch}_{hp}.\nu \mathtt{ch}_{mp}.\nu \mathtt{ch}_{phpt}. & \}\ \nu\tilde{m}\\
\qquad \left.\begin{array}{l}
\mathtt{let}\ \mathtt{pk}_{sso} = \mathsf{pk}(\mathtt{sk}_{sso})\ \mathtt{in}\\
\mathsf{out}(\mathtt{ch}, \mathtt{pk}_{sso}).
\end{array}\right\} & init\\
\qquad (!(R_{dr})\ |!(R_{pt})\ |!(R_{ph})\ |!(R_{mpa})\ |!(R_{hii})) &
\end{array}
$$

Figure 5.20: The process for the DLV08 protocol.

$$
\begin{array}{ll}
R_{dr} := \nu \mathtt{Id}_{dr}. & \\
\qquad \nu \mathtt{Pnym}_{dr}. & \}\ init_{dr}\\
\qquad !(P_{dr}) &
\end{array}
$$

Figure 5.21: The process for role doctor R_{dr}.

5.6 Analysis of DLV08

In this section, We analyse the claimed properties, including secrecy of patient and doctor information, authentication, (strong) patient and doctor anonymity,

$$\begin{array}{l} R_{hii} := \nu \mathtt{sk}_{hii}. \\ \qquad \mathtt{let}\ \mathtt{pk}_{hii} = \mathsf{pk}(\mathtt{sk}_{hii})\ \mathtt{in} \\ \qquad \mathtt{let}\ \mathtt{Id}_{hii} = \mathsf{host}(\mathtt{pk}_{hii})\ \mathtt{in} \\ \qquad (!\mathsf{out}(\mathtt{ch}, \mathtt{pk}_{hii})\ |!\mathsf{out}(\mathtt{ch}_{hp}, \mathtt{Id}_{hii})\ |!(P_{hii})) \end{array}$$

Figure 5.22: The process for role HII R_{hii}.

(strong) patient and doctor untraceability, (enforced) prescribing-privacy and independence of (enforced) prescribing-privacy of the DLV08 protocol. The two properties, doctor anonymity and untraceability, are not required by the protocol but are still interesting to analyse.

5.6.1 Secrecy of patient and doctor information

The DLV08 protocol is claimed to satisfy the following requirement: any party involved in the prescription processing workflow should not know the information of a patient and a doctor unless the information is intended to be revealed in the protocol. In [dDLVV08], this requirement is considered as an access control requirement. We argue that ensuring the requirement with access control is not enough when a party is dishonest. Since it is not clearly described whether the involved parties are honest, assuming a party is dishonest by observing the network and manipulating the protocol, we found that the party may obtain information which he should not access.

We analyse secrecy of patient and doctor information, respecting to a dishonest party who has the Dolev-Yao adversary ability. Private information of patients and doctors, which needs to be protected, is as follows: patient identity ($\mathtt{Id}_{pt}$), doctor identity ($\mathtt{Id}_{dr}$), patient pseudonym ($\mathtt{Pnym}_{pt}$), doctor pseudonym ($\mathtt{Pnym}_{dr}$), a patient's social security status (Sss), a patient's health insurance institute (Hii). Although it is not required, the health expense account Acc of a patient has also been verified.

Verification result. We query the standard secrecy of the set of private information using ProVerif [Bla01]. The verification result (see Table 5.1) shows that a patient's identity, pseudonym, health expense account, health insurance institute and identity of a doctor ($\mathtt{Id}_{pt}$, $\mathtt{Pnym}_{pt}$, Hii Acc, $\mathtt{Id}_{dr}$) satisfy standard secrecy; a patient's social security status Sss and a doctor's pseudonym $\mathtt{Pnym}_{dr}$ do not satisfy standard secrecy. The Sss is revealed by the proof of social security status from the patient to the pharmacist. The $\mathtt{Pnym}_{dr}$ is revealed by the revealing of both the commitment of the patient's pseudonym and the open key to the commitment during the communication between the patient and the doctor.

To fix secrecy of a patient's social security status, it requires that the proof of social security status only reveals the status to the pharmacist. Since how a social security status is represented and what the pharmacist needs to verify, are not clear, we cannot give explicit suggestions. However, if the social security status is a number, and the pharmacist only needs to verify that the number is higher than a certain threshold, the patient can prove it using zero-knowledge proof without

checked Security property	initial model	cause(s)	improvement
Secrecy of $\mathtt{Id}_{pt}$	✓		
Secrecy of $\mathtt{Pnym}_{pt}$	✓		
Secrecy of $\mathtt{Sss}$	×	revealed	session key
Secrecy of $\mathtt{Hii}$	✓		
Secrecy of $\mathtt{Acc}$	✓		
Secrecy of $\mathtt{Id}_{dr}$	✓		
Secrecy of $\mathtt{Pnym}_{dr}$	×	revealed	session key

Table 5.1: Verification results of secrecy for patients and doctors.

revealing the number; if the pharmacist needs to verify the exact value of the status, one way to fix its secrecy is that the pharmacist and the patient agree on a session key and the status is encrypted using the key. Similarly, a way to fix the secrecy of $\mathtt{Pnym}_{dr}$ is to encrypt the open information using the agreed session key.

5.6.2 Patient and doctor authentication

The protocol claims that all parties should be able to properly authenticate each other. Compared to authentications between public entities, pharmacists, MPA and HII, we focus on authentications between patients and doctors, as patients and doctors use anonymous authentication. Authentications between patients and pharmacists are sketched as well.

The authentication from a patient to a doctor is defined as when the doctor finishes his process and believes that he prescribed medicine for a patient, then the patient did ask the doctor for prescription. Similarly, the authentication from a doctor to a patient is defined as when the patient believes that he visited a doctor, the doctor did prescribe medicine for the patient.

Authentications are modelled as correspondence properties. To verify the authentication of a patient, we add an event $\mathsf{EndDr}\langle c\_Cred_{pt}, c\_Comt_{pt}\rangle$ at the end of the doctor process, meaning the doctor believes that he prescribed medicine for a patient who has a credential c_Cred_{pt} and committed c_Comt_{pt}; and add an event $\mathsf{StartPt}\langle \mathsf{ptcred}(\mathtt{Id}_{pt}, \mathtt{Pnym}_{pt}, \mathtt{Hii}, \mathtt{Sss}, \mathtt{Acc}), \mathsf{commit}(\mathtt{Id}_{pt}, \mathtt{r}_{pt})\rangle$ in the patient process, meaning that the patient did ask for a prescription. Then we query $\mathtt{ev(inj)}:\mathsf{EndDr}(\mathtt{x,y})\ \mathtt{==>}\ \mathtt{ev(inj)}:\mathsf{StartPt}(\mathtt{x,y})$, meaning that when the event EndDr is executed, there is an (unique) event $\mathsf{StartPt}$ has been executed before. Similarly, to authenticate a doctor, we add to the patient process an event $\mathsf{EndPt}\langle c\_Cred_{dr}, c\_Comt_{dr}, c\_presc, c\_PrescriptID\rangle$ at the end, and add an event $\mathsf{StartDr}\langle \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathtt{presc}, PrescriptID\rangle$ in the doctor process, then query $\mathtt{ev(inj)}:\mathsf{EndPt}(\mathtt{x,y,z,t})\ \mathtt{==>}\ \mathtt{ev(inj)}:\mathsf{StartDr}(\mathtt{x,y,z,t})$.

Verification result. The queries are verified using the tool ProVerif [Bla02]. The verification result shows that the doctor authentication, both injective and non-injective, succeed; the non-injective patient authentication succeeds and injective patient authentication fails. The failure is caused by the a replay attack from the adversary. That is, the adversary can impersonate a patient by replaying old messages from the patient. This authentication flaw leads to termination of the

checked Auth	initial model	cause(s)	improvement
dr to pt (inject)	√		
dr to pt (non-inject)	√		
pt to dr (inject)	×	replay attack	add challenge
pt to dr (non-inject)	√		
ph to pt (inject)	×	adv. can reply 1st message, compute 2nd message	sign the invoice
ph to pt (non-inject)	×	adv. can reply 1st message, compute 2nd message	sign the invoice
pt to ph (inject)	×	replay attack	add challenge
pt to ph (non-inject)	√		

Table 5.2: Verification results of authentication of patients and doctors.

successive procedure, the patient-pharmacist sub-process. We verified the authentications between patients and pharmacists as well. Non-injective patient authentication succeeds, and injective patient authentication fails. This means that the messages received by a pharmacist are from a patient, but not necessarily from the patient who the pharmacist is communicating with. Neither non-injective nor injective pharmacist authentication succeeds. This means that the adversary can record messages from a pharmacist and pretend to be that pharmacist. The verification results are summarised in Table 5.2.

The reason that injective patient authentications to doctors and pharmacists fail is that they suffer from replay attack. To fix them, one way is to add a challenge step from the doctor or pharmacist to the patient, and when the patient authenticates to the doctor or pharmacist, the patient includes the challenge in the proofs. This assures that the proof is freshly generated. Thus it prevents the adversary replaying old messages. The reason that the (injective and non-injective) authentication from a pharmacist to a patient fails is that the adversary can generate the invoice to replace the one from the real pharmacist. To fix the flaw, one way is for the pharmacist to sign the invoice.

5.6.3 (Strong) patient and doctor anonymity

The DLV08 protocol claims that no party should be able to determine the identity of a patient. We define (strong) patient anonymity to capture the requirement. Note that in the original paper of the DLV08 protocol, the terminology of the privacy notion for capturing this requirement is patient untraceability. Our definition of untraceability (Definition 5.7) has different meaning from theirs (for details, see Section 5.6.4). Also note that the satisfaction of standard secrecy of patient identity does not fully capture this requirement, as the adversary can still guess about it.

Patient and doctor anonymity. Doctor anonymity is defined as in Definition 5.5. Patient anonymity can be defined in a similar way by replacing the role of doctor with the role of patient.

$$\mathcal{C}_{eh}[init_{pt}\{\mathtt{t_A}/Id_{pt}\}.!P_{pt}\{\mathtt{t_A}/Id_{pt}\}] \approx_{\ell} \mathcal{C}_{eh}[init_{pt}\{\mathtt{t_B}/Id_{pt}\}.!P_{pt}\{\mathtt{t_B}/Id_{pt}\}].$$

To verify doctor/patient anonymity, is to check the satisfiability of the corresponding equivalence between processes in the definition. This is done by modelling the two processes on two sides of the equivalence as a bi-processes, and verify the bi-process using ProVerif. Recall that a bi-process models two processes sharing the same structure and differing only in terms or destructors. The two processes are written as one process with choice-constructors which tells ProVerif the spots where the two processes differ. For example, $\mathsf{choice}[x, y]$ means that the first process uses x to replace $\mathsf{choice}[x, y]$ while the second process uses y.

The bi-process for verifying doctor anonymity is

$$\nu\tilde{m}.init.(!R_{pt} \;|!R_{dr} \;|!R_{ph} \;|!R_{mpa} \;|!R_{hii} \;|\; (\nu\mathtt{Pnym}_{dr}.\text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_A}, \mathtt{d_B}] \text{ in } !P_{dr}),$$

and the bi-process for verifying patient anonymity is

$$\begin{aligned}\nu\tilde{m}.init.(!R_{pt} \;|!R_{dr} \;|!R_{ph} \;|!R_{mpa} \;|!R_{hii} \;|\; (&\text{let } \mathtt{Id}_{pt} = \mathsf{choice}[\mathtt{t_A}, \mathtt{t_B}] \text{ in}\\ &\nu\mathtt{Pnym}_{pt}.\nu\mathtt{Sss}.\nu\mathtt{Acc}.\mathsf{in}(\mathtt{ch}_{hp}, \mathtt{Hii}).\text{let } c_{pt}\text{-}pk_{hii} = \mathsf{key}(\mathtt{Hii}) \text{ in } !P_{pt}).\end{aligned}$$

Strong patient and doctor anonymity. Strong doctor anonymity is defined as in Definition 5.6. By replacing the role of doctor with the role of patient, we obtain the definition of strong patient anonymity.

The bi-process for verifying strong doctor anonymity is

$$\begin{aligned}&\textit{free } \mathtt{d_A};\\ &\nu\tilde{m}.init.(!R_{pt} \;|!R_{dr} \;|!R_{ph} \;|!R_{mpa} \;|!R_{hii} \;|\\ &\qquad (\nu\mathtt{d_B}.\nu\mathtt{nPnym}_{dr}.\text{let } \mathtt{Pnym}_{dr} = \mathtt{nPnym}_{dr} \text{ in } !(\text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_B}, \mathtt{d_A}] \text{ in } P_{dr}))),\end{aligned}$$

and the bi-process for verifying strong patient anonymity is

$$\begin{aligned}&\textit{free } \mathtt{t_A};\\ &\nu\tilde{m}.init.(!R_{pt} \;|!R_{dr} \;|!R_{ph} \;|!R_{mpa} \;|!R_{hii} \;|\; (\nu\mathtt{t_B}.\nu\mathtt{Pnym}_{pt}.\nu\mathtt{Sss}.\nu\mathtt{Acc}.\\ &\qquad \mathsf{in}(\mathtt{ch}_{hp}, \mathtt{Hii}).\text{let } c_{pt}\text{-}pk_{hii} = \mathsf{key}(\mathtt{Hii}) \text{ in } !(\text{let } \mathtt{Id}_{pt} = \mathsf{choice}[\mathtt{t_B}, \mathtt{t_A}] \text{ in } P_{pt}))).\end{aligned}$$

Verification result. The bi-processes are verified using ProVerif. The verification results show that patient anonymity and strong patient anonymity are satisfied and neither doctor anonymity nor strong doctor anonymity is not satisfied. For strong doctor anonymity, the adversary can distinguish a process initiated by an unknown doctor and a known doctor. Given a doctor process, where the doctor has identity $\mathtt{d_A}$, pseudonym $\mathtt{Pnym}_{dr}$, and credential $\mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{d_A})$, $\mathtt{Pnym}_{dr}$ and $\mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{d_A})$ are revealed. We assume that the adversary knows another doctor identity $\mathtt{d_B}$. The adversary can fake an anonymous authentication by faking the zero-knowledge proof as $\mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{d_B}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{d_A}))$. If the zero-knowledge proof passes the corresponding verification $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{dr}}$ by the patient, then the adversary knows that the doctor process is executed by the doctor $\mathtt{d_B}$. Otherwise, not. For the same reason, doctor anonymity fails the verification. Both flaws can be fixed by requiring a doctor to generate a new credential in each session (**s4'**).

5.6.4 (Strong) patient and doctor untraceability

Even if a user's identity is not revealed, the adversary may be able to trace a user by telling whether two executions are done by the same user. The DLV08 protocol claims that prescriptions issued to the same patient should not be linkable to each other. In other words, the situation in which a patient executes the protocol twice should be indistinguishable from the situation in which two different patients execute the protocol individually. To satisfy this requirement, patient untraceability is required. Note that patient untraceability is claimed as patient unlinkability by the authors of the DLV08 protocol, we use different terminology.

Patient and doctor untraceability. Doctor untraceability has been defined in Definition 5.7, and patient untraceability can be defined in a similar style.

The bi-process for verifying doctor untraceability is

$$\begin{array}{l}\nu\tilde{m}.init.(!R_{pt}\ |!R_{dr}\ |!R_{ph}\ |!R_{mpa}\ |!R_{hii}\ |\ (\nu\mathtt{nPnym}_{dr}.\nu\mathtt{wPnym}_{dr}.\\ ((\text{let } \mathtt{Id}_{dr} = \mathtt{d_A} \text{ in let } \mathtt{Pnym}_{dr} = \mathtt{nPnym}_{dr} \text{ in } P_{dr})\ |\\ \ (\text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_A}, \mathtt{d_B}] \text{ in let } \mathtt{Pnym}_{dr} = \mathsf{choice}[\mathtt{nPnym}_{dr}, \mathtt{wPnym}_{dr}] \text{ in } P_{dr})))),\end{array}$$

and the bi-process for verifying patient untraceability is

$$\begin{array}{l}\nu\tilde{m}.init.(!R_{pt}\ |!R_{dr}\ |!R_{ph}\ |!R_{mpa}\ |!R_{hii}\ |\\ (\nu\mathtt{nPnym}_{pt}.\nu\mathtt{nSss}.\nu\mathtt{nAcc}.\nu\mathtt{wPnym}_{pt}.\nu\mathtt{wSss}.\nu\mathtt{wAcc}.\\ \ \mathsf{in}(\mathtt{ch}_{hp}, nHii).\mathsf{in}(\mathtt{ch}_{hp}, wHii).\\ \ \text{let } c_{pt}\text{-}npk_{hii} = \mathsf{key}(nHii) \text{ in let } c_{pt}\text{-}wpk_{hii} = \mathsf{key}(wHii) \text{ in}\\ (\text{let } \mathtt{Hii} = nHii \text{ in let } c_{pt}\text{-}pk_{hii} = c_{pt}\text{-}npk_{hii} \text{ in let } \mathtt{Id}_{pt} = \mathtt{t_A} \text{ in}\\ \ \text{let } \mathtt{Pnym}_{pt} = \mathtt{nPnym}_{pt} \text{ in let } \mathtt{Sss} = \mathtt{nSss} \text{ in let } \mathtt{Acc} = \mathtt{nAcc} \text{ in } P_{pt})\ |\\ (\text{let } \mathtt{Hii} = \mathsf{choice}[nHii, wHii] \text{ in let } c_{pt}\text{-}pk_{hii} = \mathsf{choice}[c_{pt}\text{-}npk_{hii}, c_{pt}\text{-}wpk_{hii}] \text{ in}\\ \ \text{let } \mathtt{Id}_{pt} = \mathsf{choice}[\mathtt{t_A}, \mathtt{t_B}] \text{ in let } \mathtt{Pnym}_{pt} = \mathsf{choice}[\mathtt{nPnym}_{pt}, \mathtt{wPnym}_{pt}] \text{ in}\\ \ \text{let } \mathtt{Sss} = \mathsf{choice}[\mathtt{nSss}, \mathtt{wSss}] \text{ in let } \mathtt{Acc} = \mathsf{choice}[\mathtt{nAcc}, \mathtt{wAcc}] \text{ in } P_{pt}))).\end{array}$$

Strong patient and doctor untraceability. Strong untraceability is modelled as a patient executing the protocol repeatedly is indistinguishable from different patients executing the protocol each once. Strong doctor untraceability is defined as in Definition 5.8 and strong patient untraceability can be defined in the same manner.

The bi-process for verifying strong doctor untraceability is

$$\begin{array}{l}\nu\tilde{m}.init.(!R_{pt}\ |!R_{ph}\ |!R_{mpa}\ |!R_{hii}\ |!(\nu\mathtt{nId}_{dr}.\nu\mathtt{nPnym}_{dr}.!(\nu\mathtt{wId}_{dr}.\nu\mathtt{wPnym}_{dr}.\\ \ \text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{nId}_{dr}, \mathtt{wId}_{dr}] \text{ in let } \mathtt{Pnym}_{dr} = \mathsf{choice}[\mathtt{nPnym}_{dr}, \mathtt{wPnym}_{dr}] \text{ in } P_{dr}))),\end{array}$$

and the bi-process for verifying strong patient untraceability is

$$\begin{array}{l}\nu\tilde{m}.init.(!R_{dr}\ |!R_{ph}\ |!R_{mpa}\ |!R_{hii}\ |!(\nu\mathtt{nId}_{pt}.\nu\mathtt{nPnym}_{pt}.\nu\mathtt{nSss}.\nu\mathtt{nAcc}.\mathsf{in}(\mathtt{ch}_{hp}, nHii).\\ \ !(\nu\mathtt{wId}_{pt}.\nu\mathtt{wPnym}_{pt}.\nu\mathtt{wSss}.\nu\mathtt{wAcc}.\\ \ \ \text{let } \mathtt{Id}_{pt} = \mathsf{choice}[\mathtt{nId}_{pt}, \mathtt{wId}_{pt}] \text{ in let } \mathtt{Pnym}_{pt} = \mathsf{choice}[\mathtt{nPnym}_{pt}, \mathtt{wPnym}_{pt}] \text{ in}\\ \ \ \text{let } \mathtt{Sss} = \mathsf{choice}[\mathtt{nSss}, \mathtt{wSss}] \text{ in let } \mathtt{Acc} = \mathsf{choice}[\mathtt{nAcc}, \mathtt{wSss}] \text{ in}\\ \ \ \mathsf{in}(\mathtt{ch}_{hp}, wHii).\text{let } \mathtt{Hii} = \mathsf{choice}[nHii, wHii] \text{ in let } c_{pt}\text{-}pk_{hii} = \mathsf{key}(\mathtt{Hii}) \text{ in } P_{pt}))).\end{array}$$

Verification result. The bi-processes are verified using ProVerif. The verification results show that the DLV08 protocol does not satisfy patient/doctor untraceability, nor strong untraceability. The strong doctor untraceability fail because the adversary can distinguish sessions initiated by one doctor and by different doctors. The doctor's pseudonym is revealed and a doctor uses the same pseudonym in all sessions. Sessions with the same doctor pseudonyms are initiated by the same doctor. For the same reason, doctor untraceability also fails. Both of them can be fixed by requiring a doctor to freshly generate his pseudonym in each session (**s3'**). For strong patient untraceability, the adversary can distinguish sessions initiated by one patient (with identical social security statuses) and initiated by different patients (with different social security statuses). Second, the adversary can distinguish sessions initiated by one patient (with identical cipher texts $\mathsf{enc}(\mathtt{Pnym}_{pt}, \mathtt{pk}_{sso})$ and identical cipher texts $\mathsf{enc}(\mathtt{Hii}, \mathtt{pk}_{sso})$) and initiated by different patients (with different cipher texts $\mathsf{enc}(\mathtt{Pnym}_{pt}, \mathtt{pk}_{sso})$ and different cipher texts $\mathsf{enc}(\mathtt{Hii}, \mathtt{pk}_{sso})$). Third, since the patient credential is the same in all sessions and is revealed, the adversary can also trace a patient by the patient's credential. Fourth, the adversary can distinguish sessions using the same HII and sessions using different HIIs. For the same reasons, patient untraceability fails. Both flaws can be fixed by revising the assumptions (**s5'**, **s2'**, **s4"** and **s6'**).

5.6.5 Prescribing-privacy

Prescribing-privacy has been defined in Definition 5.1. To verify the prescribing-privacy is to check the satisfaction of the equivalence in the definition. The bid-process for verifying the equivalence is

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{free}\ \mathtt{d_A}.\mathit{free}\ \mathtt{d_B}.\mathit{free}\ \mathtt{p_A}.\mathit{free}\ \mathtt{p_B}.} \\
\nu\tilde{m}.\mathit{init}.(!R_{pt}\ |!R_{dr}\ |!R_{ph}\ | & (\nu\mathtt{nPnym}_{dr}.\nu\mathtt{wPnym}_{dr}. \\
& \quad \text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_A}, \mathtt{d_B}] \text{ in} \\
& \quad \text{let } \mathtt{Pnym}_{dr} = \mathsf{choice}[\mathtt{nPnym}_{dr}, \mathtt{wPnym}_{dr}] \text{ in} \\
& \quad \text{let } \mathtt{presc} = \mathtt{p_A} \text{ in } \mathit{main}_{dr})\ | \\
& (\nu\mathtt{nPnym}_{dr}.\nu\mathtt{wPnym}_{dr}. \\
& \quad \text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_B}, \mathtt{d_A}] \text{ in} \\
& \quad \text{let } \mathtt{Pnym}_{dr} = \mathsf{choice}[\mathtt{nPnym}_{dr}, \mathtt{wPnym}_{dr}] \text{ in} \\
& \quad \text{let } \mathtt{presc} = \mathtt{p_B} \text{ in } \mathit{main}_{dr})).
\end{array}$$

Verification result. The verification, using ProVerif, shows that the DLV08 protocol does not satisfy prescribing-privacy, i.e., the adversary can distinguish whether a prescription is prescribed by doctor $\mathtt{d_A}$ or doctor $\mathtt{d_B}$. In the prescription proof, a prescription is linked to a doctor credential. And a doctor credential is linked to a doctor identity. Thus, the adversary can link a doctor to his prescription. To break the link, one way is to make sure that the adversary cannot link a doctor credential to a doctor identity. This can be achieved by adding randomness to the credential (**s4'**).

5.6.6 Enforced prescribing-privacy

The definition of enforced prescribing-privacy is modelled as the existence of a process P'_{dr}, such that the two equivalences in Definition 5.2 are satisfied. Due to the existential quantification, we cannot verify the property directly using ProVerif.

Examining the DLV08 protocol, we find an attack on enforced prescribing-privacy, even after fixing prescribing privacy (with assumption **s4'**). A bribed doctor is able to prove to the adversary of his prescription as follows:

1. A doctor communicates with the adversary to agree on a bit-commitment that he will use, which links the doctor to the commitment.

2. The doctor uses the agreed bit-commitment in the communication with his patient. This links the bit-commitment to a prescription.

3. Later, when the patient uses this prescription to get medicine from a pharmacist, the adversary can observe the prescription being used. This proves that the doctor has really prescribed the medicine.

Formally, using ProVerif, we can show that if a doctor reveals all his information to the adversary, the doctor's prescribing-privacy is broken. To prove that there exist no alternative precesses for a doctor to cheat the adversary, we assume that there exists a process P'_{dr} which satisfies the definition of enforced prescribing-privacy, and then derive some contradiction. A bribed doctor reveals the nonces used in the commitment and the credential to the adversary. Thus, the adversary links a bribed doctor to his commitment and credential. In the prescription proof, a prescription is linked to a doctor's commitment and credential. Suppose there exists a process P'_{dr} in which the doctor lies to the adversary that he prescribed $\mathtt{p_A}$, while the adversary observes that the commitment or the credential is linked to $\mathtt{p_B}$. The adversary can detect that the doctor has lied.

Proof. Assume there exists a process P'_{dr} which satisfies the definition of enforced prescribing-privacy. i.e., $\exists\ P'_{dr}$ such that

$$
\begin{array}{ll}
1. & \mathcal{C}_{eh}\big[\big(\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\} \mid P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\})\big) \mid \\
& \qquad \big(\mathit{init}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\} \mid \mathit{main}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}, \mathtt{p_A}/\mathit{presc}\})\big)\big] \\
& \approx_\ell \mathcal{C}_{eh}\big[\big((\mathit{init}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\})^{\mathtt{chc}}.(!P_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\} \mid (\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_A}/\mathit{presc}\})^{\mathtt{chc}})\big) \mid \\
& \qquad \big(\mathit{init}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\} \mid \mathit{main}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}, \mathtt{p_B}/\mathit{presc}\})\big)\big]; \\[2ex]
2. & \mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathsf{out}(\mathtt{chc},\cdot)}.(P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathsf{out}(\mathtt{chc},\cdot)}) \\
& \approx_\ell \mathit{init}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.(\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_B}/\mathit{presc}\}).
\end{array}
$$

According to the definition of labelled bisimilarity and static equivalence, if two processes are labelled bisimilar, we have that if $M =_E N$ at any state of one process, then there exist some states of the other process, such that $M =_E N$. Due to the first equivalence, we have that if $M =_E N$ on the right hand side of the first equivalence, then $M =_E N$ on the left hand side.

On the right hand side of the first equivalence, there exists an output of a prescription proof $\mathit{PrescProof}^{\mathtt{r}}$, over public channels, from a process initiated by doctor $\mathtt{d_A}$ (process $(\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_A}/\mathit{presc}\})^{\mathtt{chc}}$). The adversary can obtain the prescription $\mathtt{p_A}$ from $\mathit{PrescProof}^{\mathtt{r}}$.

$$(\mathtt{p_A}, \mathit{PrescriptID}^r, \mathit{Comt}^r_{dr}, \mathit{c\_Comt}^r_{pt}) = \mathsf{getmsg}(\mathit{PrescProof}^{\mathtt{r}})$$

On the left hand side of the first equivalence, there should also exist an output of a prescription proof $\mathit{PrescProof}^{l}$ over public channels, from which the adversary can obtain a prescription $\mathtt{p_A}$.

$$(\mathtt{p_A}, \mathit{PrescriptID}^{l}, \mathit{Comt}^{l}_{dr}, c\_\mathit{Comt}^{l}_{pt}) = \mathsf{getmsg}(\mathit{PrescProof}^{l})$$

Next, we prove that prescription proof $\mathit{PrescProof}^{l}$ is the prescription proof in process $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$. On the right hand side of the first equivalence, the term $\mathit{PrescProof}^{\mathtt{r}}$ is computed as

$$\begin{aligned}\mathit{PrescProof}^{\mathtt{r}} = \mathsf{spk}((&\mathtt{Pnym^r}_{dr}, \mathtt{r}^{r}_{dr}, \mathtt{d_A}, \mathtt{n}^{r}_{dr}),\\ &(\mathsf{commit}(\mathtt{Pnym^r}_{dr}, \mathtt{r}^{r}_{dr}), \mathsf{drcred}(\mathtt{Pnym^r}_{dr}, \mathtt{d_A}, \mathtt{n}^{r}_{dr})),\\ &(\mathtt{p_A}, \mathit{PrescriptID}^{r}, \mathsf{commit}(\mathtt{Pnym^r}_{dr}, \mathtt{r}^{r}_{dr}), c\_\mathit{Comt}^{r}_{pt})\end{aligned}$$

where the doctor pseudonym $\mathtt{Pnym^r}_{dr}$ and the nonce for doctor commitment $\mathtt{r}^{r}_{dr}$ and the nonce for doctor credential $\mathtt{n}^{r}_{dr}$ (used for assumption **s4'**) are revealed to the adversary on `chc` channel. On the left hand side of the first equivalence, to be equivalent to the right side, there should also exist sub-processes which output messages on `chc` channel. Such sub-processes can only be $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ and $P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$, because there is no output on `chc` in other sub-processes in the left hand side process. Suppose $\mathtt{Pnym^l}_{dr}$, $\mathtt{r}^{l}_{dr}$ and $\mathtt{n}^{l}_{dr}$ are the doctor pseudonym and the nonces sent to the adversary over `chc` on the left hand side (corresponding to $\mathtt{Pnym^r}_{dr}$, $\mathtt{r}^{r}_{dr}$ and $\mathtt{n}^{r}_{dr}$ on the right hand side). Since the process $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ knows $\mathtt{Pnym^l}_{dr}$, $\mathtt{r}^{l}_{dr}$, $\mathtt{n}^{l}_{dr}$ and $\mathtt{d_A}$, the process is able to compute and thus output a prescription proof $\mathit{PrescProof}^{l}$, where the prescription is $\mathtt{p_A}$.

$$\begin{aligned}\mathit{PrescProof}^{l} = \mathsf{spk}((&\mathtt{Pnym^l}_{dr}, \mathtt{r}^{l}_{dr}, \mathtt{d_A}, \mathtt{n}^{l}_{dr}),\\ &(\mathsf{commit}(\mathtt{Pnym^l}_{dr}, \mathtt{r}^{l}_{dr}), \mathsf{drcred}(\mathtt{Pnym^l}_{dr}, \mathtt{d_A}, \mathtt{n}^{l}_{dr})),\\ &(\mathtt{p_A}, \mathit{PrescriptID}^{l}, \mathsf{commit}(\mathtt{Pnym^l}_{dr}, \mathtt{r}^{l}_{dr}), c\_\mathit{Comt}^{l}_{pt})\end{aligned}$$

On the other hand, sub-processes on the left hand side of the first equivalence, except processes $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ and $P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$, cannot generate such a prescription as $\mathit{PrescProof}^{l}$ with $\mathtt{Pnym^l}_{dr}$, $\mathtt{r}^{l}_{dr}$ and $\mathtt{n}^{l}_{dr}$ shared with the adversary.

1) The sub-processes except doctor sub-processes honestly follow the protocol and thus do not generate a signed proof knowledge which matches the pattern of $\mathit{PrescProof}^{l}$.

2) The prescription proof $\mathit{PrescProof}^{l}$ cannot be generated in doctor sub-processes except the sub-processes $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ and $P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$.

   - First, the doctor sub-processes do not generate $\mathit{PrescProof}^{l}$ using outside information. The doctor sub-processes honestly follow the protocol and thus do not purposely read in the secret values ($\mathtt{Pnym^l}_{dr}$, $\mathtt{r}^{l}_{dr}$ and $\mathtt{n}^{l}_{dr}$) from the `chc` channel to generate $\mathit{PrescProof}^{l}$. Even the adversary feeds the sub-processes with $\mathtt{Pnym^l}_{dr}$, $\mathtt{r}^{l}_{dr}$ and $\mathtt{n}^{l}_{dr}$ over public channels, the sub-processes cannot generate $\mathit{PrescProof}^{l}$. Because the honest doctor sub-processes do not use read in values to generate a prescription proof.
   - Second, the honest doctor sub-processes can not generate $\mathit{PrescProof}^{l}$ using their initial knowledge.

- Since in honest doctor sub-processes, $\mathtt{Pnym}_{dr}$, $\mathtt{r}_{dr}$ and $\mathtt{n}_{dr}$ are freshly generated in each session of a doctor sub-process, the honest doctor sub-processes do not generate $\mathit{PrescProof}^{l}$ using globally shared information.
- The honest doctor sub-processes can not generate $\mathit{PrescProof}^{l}$ using self-generated information. Suppose an honest doctor sub-process generates $\mathit{PrescProof}^{l}$ using self-generated information. In order to do so, the honest doctor sub-process needs to generates $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^{l}_{dr}$ and $\mathtt{n}^{l}_{dr}$. As we stated earlier, $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ or $P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ need to know $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^{l}_{dr}$ and $\mathtt{n}^{l}_{dr}$, so that the three values can be sent to the adversary over $\mathtt{chc}$. However, there is no way for process $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ or $P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ to know $\mathtt{n}^{l}_{dr}$ from the honest doctor sub-process, because $\mathtt{n}_{dr}$ is never revealed by an honest doctor sub-process. Therefore, the honest doctor processes do not generate $\mathtt{n}^{l}_{dr}$ and thus cannot generate $\mathit{PrescProof}^{l}$ using self-generated information.

Therefore, the process which outputs the prescription proof $\mathit{PrescProof}^{l}$ on the left hand side of the first equivalence is $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ and/or $P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ Thus, the prescription in process $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ is $\mathtt{p_A}$.

However, in the right hand side process of the second equivalence, there is a prescription proof output over public channels from process $\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_B}/\mathit{presc}\}$. The adversary obtains prescription $\mathtt{p_B}$ from the prescription proof. To satisfy the second equivalence, the left hand side process of the second equivalence (the process $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathtt{out(chc,\cdot)}}.(P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathtt{out(chc,\cdot)}}))$ should also output a prescription proof (denoted as $\mathit{PrescProof}'$) where the prescription is $\mathtt{p_B}$. Since the process $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathtt{out(chc,\cdot)}}.(P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathtt{out(chc,\cdot)}})$ outputs $\mathit{PrescProof}'$, its corresponding process, $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$, in which communication over $\mathtt{chc}$ is not hidden, should also output $\mathit{PrescProof}'$. Therefore, the prescription in $\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}$ is $\mathtt{p_B}$. Since $\mathtt{p_A} \neq \mathtt{p_B}$, contradiction is found. □

5.6.7 Independency of (enforced) prescribing-privacy

The doctor's prescribing-privacy independent of the pharmacist is modelled by replacing R_i with R_{ph} in Definition 5.3. The bi-process for verifying the property is

$$
\begin{aligned}
&\mathit{free}\ \mathtt{d_A}.\mathit{free}\ \mathtt{d_B}.\mathit{free}\ \mathtt{p_A}.\mathit{free}\ \mathtt{p_B}.\\
&\nu\tilde{m}.\mathit{init}.(!R_{pt}\ |!R_{dr}\ |!(R_{ph})^{\mathtt{chc}}\ |\ (\nu\mathtt{nPnym}_{dr}.\nu\mathtt{wPnym}_{dr}.\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_A}, \mathtt{d_B}] \text{ in}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{let } \mathtt{Pnym}_{dr} = \mathsf{choice}[\mathtt{nPnym}_{dr}, \mathtt{wPnym}_{dr}] \text{ in}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{let } \mathtt{presc} = \mathtt{p_A} \text{ in } \mathit{main}_{dr})\ |\\
&\qquad\qquad\qquad\qquad\qquad\qquad\ \ (\nu\mathtt{nPnym}_{dr}.\nu\mathtt{wPnym}_{dr}.\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{let } \mathtt{Id}_{dr} = \mathsf{choice}[\mathtt{d_B}, \mathtt{d_A}] \text{ in}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{let } \mathtt{Pnym}_{dr} = \mathsf{choice}[\mathtt{nPnym}_{dr}, \mathtt{wPnym}_{dr}] \text{ in}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{let } \mathtt{presc} = \mathtt{p_B} \text{ in } \mathit{main}_{dr})).
\end{aligned}
$$

The verification, using ProVerif, shows that the protocol (after fixing the flaw on prescribing-privacy with assumption **s4'**) satisfies this property.

checked privacy notion	**initial model**	**cause(s)**	**improvement**	**revised model**
prescribing-privacy	×	**s4**	**s4’**	✓
enforced presc.-priv.	× (with **s4’**)		**s8’**	✓
ind. of presc.-priv.	✓ (with **s4’**)			✓
ind. of enf. presc.-priv.	×(with **s4’**)		**s8’**	×
patient anonymity	✓			✓
strong patient anonymity	✓			✓
doctor anonymity	×	**s4**	**s4’**	✓
strong doctor anonymity	×	**s4**	**s4’**	✓
patient untraceability	×	**s2**, **s4**, **s5**, **s6**	**s2’**, **s4”**, **s5’**, **s6’**	✓
strong patient untrace.	×	**s2**, **s4**, **s5**, **s6**	**s2’**, **s4”**, **s5’**, **s6’**	✓
doctor untraceability	×	**s3**	**s3’**	✓
strong doctor untrace.	×	**s3**	**s3’**	✓

Table 5.3: Verification results of privacy properties and revised assumptions.

Similarly, the doctor’s enforced prescribing-privacy independent of pharmacist is defined as replacing R_i with R_{ph} in Definition 5.4. The flaw described in Section 5.6.6 is also applied here. Intuitively, when a doctor can prove his prescription without the pharmacist sharing information with the adversary, the doctor can prove it when the pharmacist genuinely cooperates with the adversary.

The verification results for privacy properties are summarised in Table 5.3.

5.7 Addressing the flaws of the DLV08 protocol

To summarise, we modify assumptions in Section 5.5.1 to fix the flaws found in our analysis of the privacy notions.

- **s2’** The encryptions are probabilistic.
- **s3’** A doctor’s pseudonym is freshly generated in every session.
- **s4’** A doctor freshly generates an unpredictable credential in each session. We model this with another parameter (a random number) of the credential. Following this, anonymous authentication using these credentials proves knowledge of the used randomness.
- **s4”** A patient freshly generates a credential in each session.
- **s5’** A patient’s social security status is different in each session.
- **s6’** All patients share the same HII.

The modified protocol was verified again using ProVerif. The verification results show that the protocol with revised assumptions satisfies doctor anonymity, strong doctor anonymity, and prescribing-privacy, as well as untraceability and strong untraceability for both patient and doctor.

To make the protocol satisfies enforced prescribing-privacy, we apply the following assumption on communication channels.

- **s8'** The communication channels are untappable, except that the communication channels for authentications remain public.

Our model of the protocol is accordingly modified as follows: replacing channel ch in lines **d10**, **t6** with an untappable channel $\mathtt{ch}_{dp}$, replacing channel ch in lines **t23**, **t26**, **h5**, **h22** with an untappable channel $\mathtt{ch}_{ptph}$, and replacing channel ch in lines **t24**, **h21** with an untappable channel $\mathtt{ch}_{phpt}$. The untappable channels are modelled as global private channels. We prove that the protocol (with **s4'** and **s8'**) satisfies enforced prescribing-privacy by showing the existence of a process P'_{dr} (as shown in Figure 5.23) such that the equivalences in Definition 5.2 are satisfied. The Equivalences are verified using ProVerif.

```
$init_{dr}\{\mathtt{d_A}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{d_A}/Id_{dr}\}) :=$
let $\mathtt{Id}_{dr} = \mathtt{d_A}$ in $\nu\mathtt{Pnym}_{dr}$.
$(!P_{dr}$ (*the $P_{dr}$ has assumptions s4' and s8'*)
    $\mid$ (out(chc, $\mathtt{Id}_{dr}$).
      out(chc, $\mathtt{Pnym}_{dr}$).
      $\nu\mathtt{n}_{dr}$.out(chc, $\mathtt{n}_{dr}$).
      (*s4': creating a nonce and adding it in zk and spk*)
      out(ch, zk(($\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}, \mathtt{n}_{dr}$), drcred($\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}, \mathtt{n}_{dr}$))).
      in(ch, ($rcv\_Auth_{pt}$, $rcv\_PtProof$)).
      out(chc, ($rcv\_Auth_{pt}$, $rcv\_PtProof$)).
      let $c\_Cred_{pt}$ = getpublic($rcv\_Auth_{pt}$) in
      let ($c\_Comt_{pt}$, $= c\_Cred_{pt}$) = getpublic($rcv\_PtProof$) in
      if $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}$($rcv\_Auth_{pt}$, $c\_Cred_{pt}$) = true then
      if $\mathsf{Vfy\text{-}zk}_{\mathsf{PtProof}}$($rcv\_PtProof$, ($c\_Comt_{pt}$, $c\_Cred_{pt}$)) = true then
      out(chc, $\mathtt{p_A}$).
      $\nu\mathtt{r}_{dr}$.
      out(chc, $\mathtt{r}_{dr}$).
      let $PrescriptID$ = hash($\mathtt{p_B}$, $c\_Comt_{pt}$, commit($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}$)) in
      out($\mathtt{ch}_{dp}$, (spk(($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}, \mathtt{n}_{dr}$),
                     (commit($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}$), drcred($\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}, \mathtt{n}_{dr}$)),
                     ($\mathtt{p_B}$, $PrescriptID$, commit($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}$), $c\_Comt_{pt}$)),
                 $\mathtt{r}_{dr}$)).
      out(chc, (spk(($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}, \mathtt{n}_{dr}$),
                    (commit($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}$), drcred($\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}, \mathtt{n}_{dr}$)),
                    ($\mathtt{p_A}$, hash($\mathtt{p_A}$, $c\_Comt_{pt}$, commit($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}$)),
                    commit($\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}$), $c\_Comt_{pt}$)),
                $\mathtt{r}_{dr}$))))
```

Figure 5.23: The doctor process P'_{dr} (using untappable channels).

However, with the above assumptions the DLV08 protocol does not satisfy independency of enforced prescribing-privacy. We first show that P'_{dr} is not sufficient for proving this with ProVerif. Then we prove (analogous to the proof in Section 5.6.6) that there is no alternative process P'_{dr} which satisfies Definition 5.4. Intuitively, all information sent over untappable channels are received by pharmacists and can be genuinely revealed to the adversary by the pharmacists (do not lie by assumption). Hence, there still exist links between a doctor, his nonces, his commitment, his credential and his prescription, when the doctor is bribed/coerced to reveal the nonces used in the commitment and the credential to the adversary.

Proof. Suppose, there exists a process P'_{dr} which satisfies the definition of independency of enforced prescribing-privacy. That is, $\exists\, P'_{dr}$ such that the two equivalences in Definition 5.4 are satisfied.

$$\begin{array}{rl} 1. & \mathcal{C}_{eh}[!R_{ph}^{\mathtt{chc}} \mid \big(\mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\} \mid P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\})\big) \mid \\ & \qquad\qquad \big(\mathit{init}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\} \mid \mathit{main}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}, \mathtt{p_A}/\mathit{presc}\})\big)] \\ \approx_\ell & \mathcal{C}_{eh}[!R_{ph}^{\mathtt{chc}} \mid \big((\mathit{init}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\})^{\mathtt{chc}}. \\ & \qquad\qquad (!P_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\} \mid (\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_A}/\mathit{presc}\})^{\mathtt{chc}})\big) \mid \\ & \qquad\qquad \big(\mathit{init}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\} \mid \mathit{main}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}, \mathtt{p_B}/\mathit{presc}\})\big)]; \\ 2. & \mathit{init}'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}.(P'_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}) \\ \approx_\ell & \mathit{init}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}\}.(\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_B}/\mathit{presc}\}). \end{array}$$

We prove that this assumption leads to contradictions.

According to the definition of labelled bisimilarity and static equivalence, if two processes are labelled bisimilar, we have that if $M =_E N$ at any state of one process, then there exist some states of the other process, such that $M =_E N$. Due to the first equivalence, we have that if $M =_E N$ on the right hand side of the first equivalence, then $M =_E N$ on the left hand side of the first equivalence.

In the right hand side process of the first equivalence, the doctor $\mathtt{d_A}$ computed a prescription proof $\mathit{PrescProof}^{\mathtt{r}}$ in sub-process $(\mathit{main}_{dr}\{\mathtt{d_A}/\mathit{Id}_{dr}, \mathtt{p_A}/\mathit{presc}\})^{\mathtt{chc}}$. This prescription proof eventually is revealed to the adversary on the chc channel by a pharmacist. The adversary can obtain a prescription $\mathtt{p_A}$ and a doctor commitment Comt^r_{dr} from it using

$$(\mathtt{p_A}, \mathit{PrescriptID}^r, \mathit{Comt}^r_{dr}, c\_\mathit{Comt}^r_{pt}) = \mathsf{getmsg}(\mathit{PrescProof}^{\mathtt{r}}).$$

The adversary can also obtain the doctor credential Cred^r_{dr} and the same doctor commitment Comt^r_{dr} from the prescription proof using

$$(\mathit{Comt}^r_{dr}, \mathit{Cred}^r_{dr}) = \mathsf{getSpkVinfo}(\mathit{PrescProof}^{\mathtt{r}}).$$

In the left hand side process of the first equivalence, there should also exist an output of a prescription proof $\mathit{PrescProof}^l$ on the chc channel by a pharmacist, such that

$$(\mathtt{p_A}, \mathit{PrescriptID}^l, \mathit{Comt}^l_{dr}, c\_\mathit{Comt}^l_{pt}) = \mathsf{getmsg}(\mathit{PrescProof}^l),$$

and

$$(\mathit{Comt}^l_{dr}, \mathit{Cred}^l_{dr}) = \mathsf{getSpkVinfo}(\mathit{PrescProof}^l).$$

Furthermore, in the right hand side process of the first equivalence, the term Comt^r_{dr} is computed as

$$\mathit{Comt}^r_{dr} = \mathsf{commit}(\mathtt{Pnym^r}_{dr}, \mathtt{r}^r_{dr}),$$

where $\mathtt{Pnym^r}_{dr}$, $\mathtt{r}^r_{dr}$ are revealed to the adversary on $\mathtt{chc}$ channel, and the term $Cred^r_{dr}$ is computed as

$$Cred^r_{dr} = \mathsf{drcred}(\mathtt{d_A}, \mathtt{Pnym^r}_{dr}, \mathtt{n}^r_{dr}),$$

where $\mathtt{Pnym^r}_{dr}$ and $\mathtt{n}^r_{dr}$ are revealed to the adversary on $\mathtt{chc}$ channel. In the left hand side process of the first equivalence, there should also exist sub-processes which output messages on $\mathtt{chc}$ channel. Such sub-processes can only be $!R^{\mathtt{chc}}_{ph}$, $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}$ and $P'_{dr}\{\mathtt{d_A}/Id_{dr}\}$.

Suppose the corresponding doctor pseudonym, the nonce used for doctor commitments and the nonce used in doctor credentials on the left hand side, which are sent to the adversary over $\mathtt{chc}$, are $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ and $\mathtt{n}^l_{dr}$. The adversary obtains the bribed doctor's commitment on the left hand side as

$$Comt^l_{dr} = \mathsf{commit}(\mathtt{Pnym^1}_{dr}, \mathtt{r}^l_{dr}),$$

and the bribed doctor's credential of the left hand side as

$$Cred^l_{dr} = \mathsf{drcred}(\mathtt{d_A}, \mathtt{Pnym^1}_{dr}, \mathtt{n}^l_{dr}).$$

According to the protocol and the transformation $P^{\mathtt{chc}}$, messages output over $\mathtt{chc}$ by process $!R^{\mathtt{chc}}_{ph}$ cannot be used as $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ or $\mathtt{n}^l_{dr}$, because the relations between the output messages and the prescription proof $PrescProof^l$ cannot be satisfied. Thus, the processes that output $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ and $\mathtt{n}^l_{dr}$ over $\mathtt{chc}$ are $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}$ and/or $P'_{dr}\{\mathtt{d_A}/Id_{dr}\}$.

Next, we show that $PrescProof^l$ is indeed the prescription proof generated by the process $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}.P'_{dr}\{\mathtt{d_A}/Id_{dr}\}$. First, $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}.P'_{dr}\{\mathtt{d_A}/Id_{dr}\}$ is able to generate such a prescription proof, with the knowledge of $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$, $\mathtt{n}^l_{dr}$ and $\mathtt{d_A}$.

$$\begin{aligned} PrescProof^l = \mathsf{spk}(&(\mathtt{Pnym^1}_{dr}, \mathtt{r}^l_{dr}, \mathtt{d_A}, \mathtt{n}^l_{dr}),\\ &(\mathsf{commit}(\mathtt{Pnym^1}_{dr}, \mathtt{r}^l_{dr}), \mathsf{drcred}(\mathtt{Pnym^1}_{dr}, \mathtt{d_A}, \mathtt{n}^l_{dr})),\\ &(\mathtt{p_A}, PrescriptID^l, \mathsf{commit}(\mathtt{Pnym^1}_{dr}, \mathtt{r}^l_{dr}), c\_Comt^l_{pt}), \end{aligned}$$

Second, no sub-processes, except $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}$ and $P'_{dr}\{\mathtt{d_A}/Id_{dr}\}$, can generate a prescription with $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ and $\mathtt{n}^l_{dr}$, which are shared with the adversary over $\mathtt{chc}$.

- Sub-processes, except doctor sub-processes, on the left hand side (including $!R^{\mathtt{chc}}_{ph}$) honestly follow the protocol (or transformed process) and thus do not compute a signed proof of knowledge which matches the pattern of prescription proof $PrescProof^l$.

- Doctor sub-processes, except $init'_{dr}\{\mathtt{d_A}/Id_{dr}\}$ and $P'_{dr}\{\mathtt{d_A}/Id_{dr}\}$, cannot generate a prescription with $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ and $\mathtt{n}^l_{dr}$. As honest doctor processes, $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ and $\mathtt{n}^l_{dr}$ are not purposely read in by the processes over $\mathtt{chc}$. Even $\mathtt{Pnym^1}_{dr}$, $\mathtt{r}^l_{dr}$ or $\mathtt{n}^l_{dr}$ is feed by the adversary over public channels, honest doctor sub-processes will not use the read in values to generate a prescription proof. In addition, the nonce $\mathtt{n}_{dr}$ is freshly generated in each session of an honest doctor sub-process, meaning that $\mathtt{n}^l_{dr}$ is not globally shared between sessions of doctors. Therefore, the doctor sub-processes, except $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$ and $P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$, cannot generate $\mathit{PrescProof}^l$ using read in information and globally shared information. Furthermore, the doctor sub-processes, except $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$ and $P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$, cannot generate $\mathit{PrescProof}^l$ using self-generated information. Because if the $\mathtt{n}^l_{dr}$ is generated by an honest doctor sub-process, there is no way for $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$ or $P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$ to know it, since $\mathtt{n}^l_{dr}$ is not revealed by the honest doctor sub-process.

Therefore, the prescription proof in $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$ is $\mathit{PrescProof}^l$, meaning that the doctor prescribes $\mathtt{p}_\mathtt{A}$ in process $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$.

However, on the second equivalence, in the right hand side process, the doctor prescribes $\mathtt{p}_\mathtt{B}$. Thus, on the left hand side, the doctor prescribes $\mathtt{p}_\mathtt{B}$ in process $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}^{\backslash\mathtt{out(chc,\cdot)}}.P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}^{\backslash\mathtt{out(chc,\cdot)}}$. Therefore, in corresponding process where $\mathtt{chc}$ is not hidden (process $\mathit{init}'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}.P'_{dr}\{\mathtt{d}_\mathtt{A}/\mathit{Id}_{dr}\}$), the doctor prescribes $\mathtt{p}_\mathtt{B}$, because showing the communication over $\mathtt{chc}$ does not affect the communication between the doctor and his communication partner. Since $\mathtt{p}_\mathtt{A} \neq \mathtt{p}_\mathtt{B}$, contradiction is found.

Intuitively, a bribed doctor is linked to the nonces he sent to the adversary. The nonces are linked to the doctor's prescription in a prescription proof. A doctor's prescription proof is sent over untappable channels first to a patient and later from the patient to a pharmacist. Malicious pharmacists reveal the prescription proof to the adversary. If a bribed doctor lied about his prescription, the adversary can detect it by checking the doctor's corresponding prescription proof revealed by the pharmacist. The untappable channel assumption enables the protocol to satisfy enforced prescribing-privacy but not independency of enforced prescribing-privacy because untappable channels enable a bribed doctor to hide his prescription proof and thus allow the doctor to lie about his prescription, however the pharmacist gives the prescription proof away, from which the adversary can detect whether the doctor lied about the prescription. □

5.8 Conclusions

In this chapter, we studied enforced privacy in the e-health domain. We identified the requirement that doctor privacy should be enforced to prevent doctor bribery by (e.g.) the pharmaceutical industry. To capture this requirement, we first formalised the classical privacy property prescribing-privacy, and its enforced privacy counterpart, enforced prescribing-privacy. The cooperation between the bribed doctor and the adversary is formalised in the same way as in receipt-freeness in e-voting. However, the formalisation of enforced prescribing-privacy differs from receipt-freeness in e-voting, due to the domain requirement that only part of the doctor's process needs to share information with the adversary.

Next, we noted that e-health systems involve untrusted third parties, such as pharmacists. Such parties should also not be able to assist an adversary in breaking doctor privacy. To capture this requirement, we formally defined independency of prescribing-privacy. This requirement must hold, even if the doctor is forced to help the adversary. To capture that, we formally defined independency of enforced prescribing-privacy. These formalisations were validated in a case study of the DLV08 protocol. The protocol was modelled in the applied pi calculus and verified with the help of the ProVerif tool. In addition to the (enforced) doctor privacy notions, we analysed secrecy, authentication, anonymity and untraceability for both patients and doctors. Ambiguities which may lead to flaws were found and addressed.

The experience gained in the domain-specific studies in e-auctions (Chapter 4) and e-health (this chapter), and the differences between the formalisations enable us to generalise from these formalisations and develop a domain-independent formal framework for verifying enforced privacy properties in the next chapter.

6

Enforced privacy in the presence of others

In the previous chapters, we proposed domain-specific formalisations of enforced privacy in e-auctions and e-health. However, the adversary's ability to bribe or coerce does not depend on any specific domain. In order to address the enforced privacy concerns domain-independently, in this chapter, we propose a generic property – *enforced privacy*: a user's privacy is preserved even if the user collaborates with the adversary by sharing information.

Enforced privacy only prevents the target user from undoing his own privacy. However, a third party may help to break user privacy (*collaboration*), e.g., pharmacist may help prove a doctor's prescription behaviour, revealing your vote may enable the adversary to deduce another voter's vote. On the other hand, we identify that a third party can help maintain privacy (*coalition*), e.g., a non-coerced voter (who votes as the adversary desires) can swap receipts with a coerced voter, providing the coerced voter "proof" of compliance while being free to vote as he pleases. Accounting for the privacy effect of third parties is particularly necessary in domains where many non-trusted roles are involved. For example, pharmacists in e-health may be able to help reveal prescription behaviour of doctors. In order to ensure doctor prescribing-privacy, an e-health system must prevent this [dDLVV08, FHIES11]. This requirement has been expressed and formalised in e-health [ESORICS12] and e-voting [DLL11]. We generalise these formalisations as *independency of privacy*: the help of a set of third parties does not enable the adversary to break a target user's privacy. To capture the converse situation – the privacy effect of third parties helping the target user by sharing information with the target user, we propose a new notion of *coalition privacy*: a target user's privacy is preserved with the help of a set of third parties sharing information with the target user. In particular, we use this notion to also capture the situation where third parties are involved but no information is shared between the target user and third parties. In this case, the mere *existence* of the third parties can help to create a situation where privacy is preserved.

6.1 Privacy notions

We distinguish between two classes of privacy-affecting behaviour: the target user (collaborating with the adversary or not), and the behaviour of third parties. Third parties may be *neutral*, collaborating with the adversary (*attacking*), or collaborating with the target user (*defending*) – thus we also consider the situation where some are attacking and some are defending. A target user who collaborates with

This chapter is based on published work [ESORICS13]

target user collaborates with adversary	**third parties**			
	all neutral	*some attacking*	*some defending*	*some defending* *some attacking*
no	priv	ipriv	cpriv	cipriv
yes	epriv	iepriv	cepriv	ciepriv

Table 6.1: Privacy notions

the adversary is not under the adversary's direct control, contrary to a compromised user who genuinely shares initial private information with the adversary. A *neutral* third party, like an honest user, follows the protocol specification exactly. Thus, such a third party neither actively helps nor actively harms the target user's privacy. A *defending* third party helps the target user to preserve his privacy. An *attacking* third party communicates with the adversary to break the target user's privacy. Note that we do not consider a third party that attacks and defends the target user simultaneously. Given this classification, a target user will find himself one of the following four situations w.r.t. third parties: 1) all are neutral; 2) some are attacking; 3) some are defending; and 4) some are attacking, some are defending. In the latter three cases, the remaining third parties (if any) are considered neutral. Combining the various behaviours of the third parties with those of the target user gives rise to eight privacy properties (see Table 6.1). In the table, privacy properties are abbreviated. The property, data-privacy, where the target user does not collaborate with the adversary and third parties are all neutral, is abbreviated as priv. Abbreviations of other properties can be distinguished by their prefix to priv. In particular, the prefix e indicates that the target user collaborates with the adversary. Prefix i indicates a set of third parties collaborates with the adversary. And prefix c indicates that a set of third parties cooperate with the target user. The properties listed in Table 6.1 cover all combinations of prefixes.

Examples of each property listed in Table 6.1 are as follows:

- data-privacy (priv): the adversary cannot link the contents of an encrypted email to the user;

- enforced-privacy (epriv): a voter should not be able to prove to a vote-buyer how he voted;

- independency-of-privacy (ipriv): in e-health the adversary cannot link a doctor to his prescriptions, despite the help of a pharmacist;

- independency-of-enforced-privacy (iepriv): the adversary should not be able to link a doctor to his prescriptions (to prevent bribes), even when both the pharmacist and the doctor are helping him;

- coalition-privacy (cpriv): in location-based services, the user's real location is hidden amongst the locations of the helping users;

- coalition-enforced-privacy (cepriv): in anonymous routing, a sender remains anonymous if he synchronises with a group of senders, even if he seems to collaborate;

- coalition-independency-of-privacy (cipriv): the adversary cannot link an RFID chip to its identity, even though some malicious readers are helping the adversary, provided other RFID tags behave exactly as the target one;

- coalition-independency-of-enforced-privacy (ciepriv): in electronic road pricing, other users may hide a user's route from the adversary, even if the user seems to collaborate and malicious routers relay information on passing cars to the adversary.

The examples above illustrate that similar privacy concerns arise in many different domains – e-voting, e-health, location-based services, RFID, etc. So far, attempts at formalising privacy have usually been domain-specific (e.g., [KR05, vMR08, DKR09, ACRR10, BP11, DDS11, DLL12, FAST10, ESORICS12]). We advocate a domain-independent approach to privacy, and develop a formal framework to achieve this in Section 6.2.

6.2 Formal framework

In this section, we present a formal framework which allows us to give domain-independent formalisations. We define a standard form of protocols which is able to represent any protocol. To formally define enforced privacy properties and independency of privacy properties, we model *collaboration* between users and the adversary. The collaboration allows us to precisely specify which information is shared and how it is shared, thus provides the necessary flexibility for modelling various types of collaboration. To model coalition privacy properties, we propose the notion of *coalition* in our framework to formally capture the behaviour and shared information among a target user and a set of third parties.

6.2.1 Well-formed protocols

In the applied pi calculus, a protocol is normally modelled as a plain process. For the simplicity of formalising privacy properties, we define a standard form of a protocol [ACRR10] and any protocol can be written in this form.

Definition 6.1 (well-formed protocols). *A protocol with p roles is well-formed if it is a closed plain process P_w of the form:*

$$\begin{aligned}&P_w := \nu\tilde{\mathtt{c}}.(genkey \mid !R_1 \mid \cdots \mid !R_p)\\&R_i := \nu\mathtt{id}_i.\nu\mathtt{data}_i.init_i.!(\nu\mathtt{s}_i.\nu\mathtt{sdata}_i.sinit_i.main_i) \quad (\forall i \in \{1, \cdots, p\})\end{aligned}$$

where

1. *P_w is canonical [ACRR10]: names and variables in the process never appear both bound and free, and each name and variable is bound at most once;*

2. *data is typed, channels are ground, private channels are never sent on any channel;*

3. *$\nu\tilde{\mathtt{c}}$, $\nu\mathtt{data}_i$ and $\nu\mathtt{sdata}_i$ may be null;*

4. *$init_i$ and $sinit_i$ are sequential processes;*

5. *genkey, $init_i$, $sinit_i$ and $main_i$ can be any process (possibly null) such that P_w is a closed plain process.*

In process P_w, $\tilde{\mathtt{c}}$ are channel names; *genkey* is a sub-process in which shared data (e.g., keys shared between two roles) are generated and distributed; R_i $(1 \leq i \leq p)$ is a role. To distinguish instances taking the same role R_i, each instance is dynamically associated with a distinct identity $\nu\mathtt{id}_i$; $\mathtt{data}_i$ is private data of an instance; $init_i$ models the initialisation of an instance; $(\nu\mathtt{s}_i.\nu\mathtt{sdata}_i.sinit_i.main_i)$ models a session of an instance. To distinguish sessions of the same instance, each session is dynamically associated to a distinct identity $(\nu\mathtt{s}_i)$; $\mathtt{sdata}_i$ is private data of a session; $sinit_i$ models the initialisation of a session; $main_i$ models the behaviour of a session.

Note that this standard form does not limit the type of protocols we consider. A role may include a number of sub-roles so that a user may take more than one part in a protocol. The identities do not have to be used in the process. All of $\nu\tilde{\mathtt{c}}$, $\nu\mathtt{data}_i$ and $\nu\mathtt{sdata}_i$ may be null and *genkey*, $init_i$, $sinit_i$ and $main_i$ can be any process (possibly null) such that P_w is a closed plain process. Any process can be written in a canonical form by α-conversion [ACRR10]. Thus, any protocol can be written as a well-formed protocol.

Example 6.2 (well-formed protocol). *A very simple anonymous proxy service could work via an intermediary TTP: the user sends her service requests to the TTP, which forwards them under its own identity to APP. Each request is distinct from others. In more detail, the system can be formalised in the following form (well-formed):*

$$\begin{aligned}
P \quad &:= \nu\mathtt{c}_p.(!R_U \mid !R_{TTP}) \\
R_U \quad &:= \nu\mathtt{id}.!(\nu\mathtt{req}.\mathsf{out}(\mathtt{c}_p, (\mathtt{id}, \mathtt{req}))) \\
R_{TTP} &:= \mathsf{in}(\mathtt{c}_p, (\mathtt{id}_x, x)).\mathsf{out}(\mathtt{ch}, (\mathtt{id}_{TTP}, x))
\end{aligned}$$

A user process is R_U and the TTP process is R_{TTP}. $\mathtt{c}_p$ is a private channel between a user and TTP which models the assumption that APP does not know the message between a user and TTP. $\mathtt{ch}$ is a public channel over which information is sent to the APP.

6.2.2 Data-privacy

We formally define the property data-privacy that acts as the foundation upon which other properties are built. To do so, we need to make explicit *which data* is protected. Thus, the property data-privacy always specifies the target data. When there is no ambiguity of the target data, we use data-privacy for short. In process P_w, the target data τ is a bound name which belongs to a role (the target role R_i), i.e., $\tau \in \mathsf{bn}(R_i)$. For the sake of simplicity, we (re)write the role R_i in the form of

$$R_i := \nu\mathtt{id}_i.\nu\tau.\hat{R}_i,$$

where $\hat{R}_i$ is a plain process which has two variables id_i and τ. Note that by α-conversion we can always transform any role R_i into the above form. When

$\tau \in \mathtt{data}_i$,

$$\hat{R}_i := \nu\mathtt{data}_i/\tau.init_i.!(\nu\mathtt{s}_i.\nu\mathtt{sdata}_i.sinit_i.main_i).$$

When τ is session data in session $\mathtt{s}$, i.e., $\tau \in \mathtt{sdata}'_i$,

$$\hat{R}_i := \nu\mathtt{data}_i.init_i.(!(\nu\mathtt{s}_i.\nu\mathtt{sdata}_i.sinit_i.main_i) \mid (\nu\mathtt{s}.\nu\mathtt{sdata}'_i/\tau.sinit'_i.main'_i)).$$

In case that only information in session $\mathtt{s}$ is shared with the adversary or third parties, we require that $\mathtt{s} \notin \mathsf{bn}(P_w)$, $\nu\mathtt{sdata}'_i/\tau.sinit'_i.main'_i$ is obtained by applying α-conversion on bound names and variables in the original process $\nu\mathtt{sdata}_i/\tau.$ $sinit_i.main_i$.

Intuitively, data-privacy w.r.t. τ of protocol P_w, is the unlinkability of an honest user taking role R_i and his instantiation of the target data τ. An honest user taking role R_i is modelled as process R_i. We denote a particular user – the *target user process*, as $\check{R}_i\{\mathsf{id}/id_i\}$ where $R_i := \nu\mathtt{id}_i.\check{R}_i$, variable id_i is instantiated with a name or constant id. $\hat{R}_i\{\mathsf{id}/id, t/\tau\}$ denote an instance of the target user in which the target user instantiates the target data with t where t denotes any data which can be used to replace the target data. The unlinkability is modelled as strong secrecy [Bla04] of the target data: the adversary cannot distinguish an execution of R_i where $\tau = \mathtt{t}_1$ from an execution where $\tau = \mathtt{t}_2$, for $\mathtt{t}_1 \neq \mathtt{t}_2$.

Definition 6.3 (priv). *A well-formed protocol P_w satisfies data-privacy (priv) w.r.t. data τ ($\tau \in \mathsf{bn}(R_i)$), if*

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}].$$

In the definition, id is a name or constant, $\mathtt{t}_1$ and $\mathtt{t}_2$ are free names. Since $R_i := \nu\mathtt{id}_i.\nu\tau.\hat{R}_i$, process $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}$ is an instance of role R_i where the identity is id and the target data is $\mathtt{t}_1$. The context $\mathcal{C}_{P_w}[\_]$ models honest third parties. Thus, $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]$ is an instance of the protocol P_w, similarly for $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}]$. The only difference between these two instances is the instantiation of the target data τ. Thus, this definition captures data-privacy by using the relation $\approx_\ell$: the adversary cannot distinguish a user process with different target data.

Example 6.4 (data-privacy). *We study whether the system in Example 6.2 ensures request anonymity. Request anonymity is defined as the adversary cannot distinguish a user sending request $\mathtt{req}_1$ from sending request $\mathtt{req}_2$. This property can be formalised as data-privacy with respect to a request $\mathtt{req}_i$. In order to specify a request, we write the target user process in the following form:*

$$R_U := \nu\mathtt{id}.\nu\mathtt{req}_i.\big(\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req}_i)) \mid !(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req})))\big).$$

Thus we have, $\hat{R}_U$ defined as follows:

$$\hat{R}_U := \big(\mathsf{out}(\mathsf{c}_p, (id, req_i)) \mid !(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (id, \mathtt{req})))\big).$$

Data-privacy w.r.t. $\mathtt{req}_i$ is formalised as

$$\begin{aligned} &\mathcal{C}_{ex}[\big(\mathsf{out}(\mathsf{c}_p, (\mathsf{id}, \mathtt{req}_1)) \mid !(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathsf{id}, \mathtt{req})))\big)] \\ \approx_\ell\, &\mathcal{C}_{ex}[\big(\mathsf{out}(\mathsf{c}_p, (\mathsf{id}, \mathtt{req}_2)) \mid !(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathsf{id}, \mathtt{req})))\big)], \end{aligned}$$

where $\mathcal{C}_{ex}[\_] := \nu\mathsf{c}_p.(!R_U \mid !R_{TTP} \mid \_)$.
The system does not satisfy request anonymity, because on the left hand side process of the equivalence, the adversary observes $\mathtt{req}_1$ which does not appear on the right hand side process.

6.2.3 Modelling collaboration with the adversary

In order to define enforced privacy properties where the target user collaborates with the adversary and independency privacy properties where a set of third parties collaborate with the adversary, we need to model *collaboration* of users (a target user/third parties) with the adversary.

The process of a set of users is modelled as processes of each user in parallel. Since a user process is modelled as a role in a well-formed protocol and each user process can be any role, the set of users of a well-formed protocol P_w is formally defined as a plain process $R_U := R_{u_1} \mid \cdots \mid R_{u_m}, \forall i \in \{1, \ldots, m\}, R_{u_i} \in \{R_1, \ldots, R_p\}$.

Inspired by the formal definition of coercion in [DKR09], the collaboration between a user and the adversary is formalised as a transformation of the user process. We extend it as a transformation of the process of a set of users. Note that a user need not always share *all* his information, e.g., a bribed user in a social network may reveal his relation with another user, but not his password. To be able to specify which information is shared, we formally define the set of information that a user has. Information of a user is expressed as a set of terms in the user process. Since the user processes are canonical in a well-formed protocol, bound names and variables are different in each user process. Thus, we can express information of a set of users as a set of terms appearing in the process of the set of users. Terms appearing in a plain process R_U are given by $\mathsf{Term}(R_U)$.

$$\begin{array}{ll}
\mathsf{Term}(0) = \emptyset & \mathsf{Term}(P \mid Q) = \mathsf{Term}(P) \cup \mathsf{Term}(Q) \\
\mathsf{Term}(!P) = \mathsf{Term}(P) & \mathsf{Term}(\nu \mathtt{n}.P) = \{\mathtt{n}\} \cup \mathsf{Term}(P) \\
\mathsf{Term}(\mathsf{in}(v, x).P) = \{x\} \cup \mathsf{Term}(P) & \mathsf{Term}(\mathsf{out}(v, M).P) = \{M\} \cup \mathsf{Term}(P) \\
\multicolumn{2}{l}{\mathsf{Term}(\text{if } M =_E N \text{ then } P \text{ else } Q) = \mathsf{Term}(P) \cup \mathsf{Term}(Q)}
\end{array}$$

A collaboration specification then specifies which terms of a process are shared and how they are shared.

Definition 6.5 (collaboration specification)**.** *A* collaboration specification *of a process R_U is a tuple $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$. $\Psi \subseteq \mathsf{Term}(R_U)$ denotes the set of terms sent to the adversary each of which is of base type, $\Phi \subseteq \mathsf{Term}(R_U)$ represents terms to be replaced by information provided by the adversary, c_{out} is a fresh channel for sending information to the adversary, and c_{in} is a fresh channel for reading information from the adversary, i.e., $\mathsf{c}_{out}, \mathsf{c}_{in} \notin \mathsf{fn}(R_U) \cup \mathsf{bn}(R_U)$.*

Given a plain process R_U and a collaboration specification $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$ of the process, the transformation of R_U is given by $R_U^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}$. Note that we use $\mathsf{c}_{in} = \epsilon$ to denote that the adversary neither prepares information for the coerced users nor controls the conditional evaluations of the users.

Definition 6.6 (collaboration behaviour)**.** *Let R_U be a plain process, and $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$ be a collaboration specification of R_U. Collaboration behaviour* *of R_U according to $\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ is defined as:*

- $\mathit{0}^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}\; 0,$
- $(P \mid Q)^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}\; P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \mid Q^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle},$
- $(!P)^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}\; !P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle},$
- $(\nu \mathtt{n}.P)^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}\; \begin{cases} \nu \mathtt{n}.\mathsf{out}(\mathsf{c}_{out}, \mathtt{n}).P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} & \textit{if}\ \mathtt{n} \in \mathit{\Psi}, \\ \nu \mathtt{n}.P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} & \textit{otherwise}, \end{cases}$
- $(\mathsf{in}(v, x).P)^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}\; \begin{cases} \mathsf{in}(v, x).\mathsf{out}(\mathsf{c}_{out}, x)).P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} & \textit{if}\ x \in \mathit{\Psi}, \\ \mathsf{in}(v, x).P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} & \textit{otherwise}, \end{cases}$
- $(\mathsf{out}(v, M).P)^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}\; \begin{cases} \mathsf{in}(\mathsf{c}_{in}, x).\mathsf{out}(v, x).P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} & \textit{if}\ M \in \mathit{\Phi} \\ \quad \wedge \mathsf{c}_{in} \neq \epsilon,\ \textit{where } x \textit{ is a fresh variable}, & \\ \mathsf{out}(v, M).P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} & \textit{otherwise}, \end{cases}$
- $(\textit{if}\ M =_E N\ \textit{then}\ P\ \textit{else}\ Q)^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \;\hat{=}$

$$\begin{cases} \mathsf{in}(\mathsf{c}_{in}, x).\textit{if}\ x = \mathsf{true}\ \textit{then}\ P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \textit{else}\ Q^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\ \textit{if}\ \mathsf{c}_{in} \neq \epsilon, \\ \qquad\qquad \textit{where } x \textit{ is a fresh variable and } \mathsf{true} \textit{ is a constant}, \\ \textit{if}\ M =_E N\ \textit{then}\ P^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\ \textit{else}\ Q^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} \qquad \textit{otherwise}. \end{cases}$$

Note that we only specify user behaviour in a collaboration with the adversary. The adversary's behaviour may be omitted, as in the applied pi calculus the adversary is considered as the environment and does not need to be explicitly modelled. Our approach to reasoning about the adversary's behaviour in a collaboration (e.g., enforcing a voter to cast a particular vote) follows the line of the definition of coercion-resistance in [DKR09]. Namely, a context $\mathcal{C}[\_] := \nu \mathsf{c}_{out}.\nu \mathsf{c}_{in}(\_|Q)$ models a specific way of collaboration of the adversary, where Q models the the adversary's behaviour in the context. In this way, we separate the adversary's behaviour of distinguishing two processes, which is modelled by the environment, from the behaviour of collaborating with users which is modelled by the context.

Example 6.7 (collaboration). *In the system in Example 6.2, suppose a user collaborates with the adversary in the following way. The user reveals his request* $\mathtt{req}_i$ *in a specific session with session identity i, reads in an identity and a request from the adversary and then forwards the received message in session j.*

In order to specify the collaboration, we need to specify the session i and j. The user process needs to be of the following form:

$$\begin{aligned} R_U := \nu \mathtt{id}.\big((\nu \mathtt{req}_i.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req}_i))) \mid (\nu \mathtt{req}_j.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req}_j))) \\ |!(\nu \mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req})))\big). \end{aligned}$$

The term sent to the adversary is $\mathtt{req}_i$*, i.e.,* $\mathit{\Psi} = \{\mathtt{req}_i\}$*. The replaced term is* $(\mathtt{id}, \mathtt{req}_j)$*, i.e.,* $\mathit{\Phi} = \{(\mathtt{id}, \mathtt{req}_j)\}$*. Therefore the collaboration is specified as* $\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle = \langle \{\mathtt{req}_i\}, \{(\mathtt{id}, \mathtt{req}_j)\}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$*.*

Following Definition 6.6, the collaboration behaviour of the user is modelled as:

$$\begin{aligned} R_U^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle} := \nu \mathtt{id}.\big(\ & (\nu \mathtt{req}_i.\mathsf{out}(\mathsf{c}_{out}, \mathtt{req}_i).\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req}_i))) \\ & \mid (\nu \mathtt{req}_j.\mathsf{in}(\mathsf{c}_{in}, x).\mathsf{out}(\mathsf{c}_p, x)) \\ & |!(\nu \mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req})))\big). \end{aligned}$$

6.2.4 Modelling user coalitions

To define coalition privacy properties, we need to formally define a *coalition* between a target user and a set of defending third parties. The notion collaboration from the previous section cannot be adopted directly, as it does not specify the adversary's behaviour, whereas a coalition must specify the behaviour of *all* involved users.

Given a set of users $R_U := R_{u_1} \mid \cdots \mid R_{u_m}$, a coalition of the users specifies communication between (potentially) each pair of users. For every communication, a coalition specification needs to make explicit who the sender and receiver are (unlike collaboration). Similar to the specification of collaboration, a coalition specification makes explicit which data is sent on which channel. To make the behaviour of both communicating parties explicit, we need to specify how the term in a communication is referred to in the receiver's process. A communication in a coalition is specified as a tuple $\langle R_{u_i}, R_{u_j}, M, \mathsf{c}, y\rangle$ where $R_{u_i}, R_{u_j} \in \{R_{u_1}, \ldots, R_{u_m}\}$ $(R_{u_i} \neq R_{u_j})$ are the sender and receiver process, respectively; $M \in \mathsf{Term}(R_{u_i})$ is the data sent in the communication; $\mathsf{c} \notin \mathsf{fn}(R_U) \cup \mathsf{bn}(R_U)$ is a fresh channel used in the communication; $y \notin \mathsf{fv}(R_U) \cup \mathsf{bv}(R_U)$ is the variable used by the receiver to refer to the term M. A coalition specifies a set of communications of this type (denoted as Θ). For the simplicity of modelling, we assume that for each communication, the coalition uses a distinct channel and distinct variable, i.e., $\forall\ \langle R_{u_i}, R_{u_j}, M, \mathsf{c}, y\rangle \in \Theta$ and $\langle R'_{u_i}, R'_{u_j}, M', \mathsf{c}', y'\rangle \in \Theta$ we have $\mathsf{c} \neq \mathsf{c}' \wedge y \neq y'$.

A coalition specifies a set of terms which are communicated by the originating user process and are replaced in the coalition. In addition, a coalition needs to define how a term is replaced. In a collaboration, the adversary is assumed to be able to compute and prepare this, but in a coalition, no user can compute and prepare information for other users. Thus, this ability has to be explicitly specified in a coalition as a set of substitutions $\Delta = \{\{N/M\} \mid M \in \mathsf{Term}(R_U)\}$. The new term N are calculated from a set of terms $N_1, \ldots, N_n$ which are generated by the user, read in by the original process, or read in from coalition members. A successful coalition requires that there are no such situations where N cannot be calculated in the user process when M needs to be replaced.

Moreover, in a coalition, we allow the coalition to decide values of conditional evaluations (similar to collaboration, where the adversary decides this). Since no user in a coalition has the ability to specify the values of evaluations, these need to be assigned specifically. In addition, to add more flexibility, we allow a coalition to specify which evaluations are decided by the coalition and which are not. The evaluations of a plain user process R_U is $\mathsf{Eval}(R_U)$. The assignments of evaluations are specified as a set $\Pi \subseteq \{(e, b) \mid e \in \mathsf{Eval}(R_U) \wedge b \in \{\mathsf{true}, \mathsf{false}\}\}$.

$$\begin{array}{ll}
\mathsf{Eval}(0) = \emptyset & \mathsf{Eval}(P \mid Q) = \mathsf{Eval}(P) \cup \mathsf{Eval}(Q) \\
\mathsf{Eval}(!P) = \mathsf{Eval}(P) & \mathsf{Eval}(\nu \mathtt{n}.P) = \mathsf{Eval}(P) \\
\mathsf{Eval}(\mathsf{in}(v, x).P) = \mathsf{Eval}(P) & \mathsf{Eval}(\mathsf{out}(v, M).P) = \mathsf{Eval}(P) \\
\multicolumn{2}{l}{\mathsf{Eval}(\text{if } \ M =_E N \text{ then } P \text{ else } Q) = \{M =_E N\} \cup \mathsf{Eval}(P) \cup \mathsf{Eval}(Q)}
\end{array}$$

Definition 6.8 (coalition specification). *A coalition of a set of users R_U is specified as a tuple $\langle \Theta, \Delta, \Pi\rangle$ where Θ is a set of communication, Δ is a set of substitutions and Π is an assignment for a set of evaluations.*

Note that this model does not include the coalition strategies in which the target users and defending third parties are able to generate new data, initiate new sessions, establishing new secrets, etc.

With the above setting, given a set of users R_U and a coalition specification $\langle \Theta, \Delta, \Pi \rangle$ on users, the behaviour of a user in the coalition is modelled as a coalition transformation of the user's original process.

Definition 6.9 (coalition behaviour). *Let $R_U := R_{u_1} \mid \cdots \mid R_{u_m}$ be a plain process of a set of users, $\langle \Theta, \Delta, \Pi \rangle$ be a coalition specification of process R_U, $R \in \{R_{u_1}, \cdots, R_{u_m}\}$ be a plain user process, the transformation of the process R in the coalition is given by $R^{\langle \Theta, \Delta, \Pi \rangle}$:*

$$R^{\langle \Theta, \Delta, \Pi \rangle} \mathrel{\hat{=}} \nu\eta.(R^{\langle \Gamma, \Delta, \Pi \rangle} \mid \mathsf{in}(\mathsf{c}_1, y'_1).!\mathsf{out}(\mathsf{c}'_1, y'_1) \mid \cdots \mid \mathsf{in}(\mathsf{c}_\ell, y'_\ell).!\mathsf{out}(\mathsf{c}'_\ell, y'_\ell))$$

where $\Gamma = \{\langle R, R_{u_j}, M, \mathsf{c}, y \rangle \mid \langle R, R_{u_j}, M, \mathsf{c}, y \rangle \in \Theta\}$, $\eta = \{\mathsf{c}'_1, \ldots, \mathsf{c}'_\ell\}$, c'_1, …, c'_ℓ are fresh, $\{\mathsf{c}_1, \ldots, \mathsf{c}_\ell\} = \{\mathsf{c} \mid \langle R_{u_i}, R, M, \mathsf{c}, y \rangle \in \Theta\}$, $y'_1, \ldots, y'_\ell$ are fresh variables, $\xi = \{(\mathsf{c}_1, y'_1, \mathsf{c}'_1), \ldots, (\mathsf{c}_\ell, y'_\ell, \mathsf{c}'_\ell)\}$ defines the association of channels and variables in process $\mathsf{in}(\mathsf{c}_1, y'_1).!\mathsf{out}(\mathsf{c}'_1, y'_1) \mid \cdots \mid \mathsf{in}(\mathsf{c}_\ell, y'_\ell).!\mathsf{out}(\mathsf{c}'_\ell, y'_\ell)$, and $R^{\langle \Gamma, \Delta, \Pi \rangle}$ is given by:

- $0_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}} 0,$
- $(P \mid Q)_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}} P_F^{\langle \Gamma, \Delta, \Pi \rangle} \mid Q_F^{\langle \Gamma, \Delta, \Pi \rangle},$
- $(!P)_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}} !P_F^{\langle \Gamma, \Delta, \Pi \rangle},$
- $(\nu \mathtt{n}.P)_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}} \begin{cases} \nu\mathtt{n}.\mathsf{out}(\mathsf{c}_1, \mathtt{n}).\ldots.\mathsf{out}(\mathsf{c}_\ell, \mathtt{n}).P_F^{\langle \Gamma, \Delta, \Pi \rangle} \\ \qquad \textit{if } \{\mathsf{c}_1, \ldots, \mathsf{c}_\ell\} = \{\mathsf{c} \mid \langle R, R_{u_j}, \mathtt{n}, \mathsf{c}, y \rangle \in \Gamma\}, \\ \nu\mathtt{n}.P_F^{\langle \Gamma, \Delta, \Pi \rangle} \qquad \textit{otherwise}, \end{cases}$
- $(\mathsf{in}(v, x).P)_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}} \begin{cases} \mathsf{in}(v, x).\mathsf{out}(\mathsf{c}_1, x).\ldots.\mathsf{out}(\mathsf{c}_\ell, x).P_F^{\langle \Gamma, \Delta, \Pi \rangle} \\ \qquad \textit{if } \{\mathsf{c}_1, \ldots, \mathsf{c}_\ell\} = \{\mathsf{c} \mid \langle R, R_{u_j}, x, \mathsf{c}, y \rangle \in \Gamma\}, \\ \mathsf{in}(v, x).P_F^{\langle \Gamma, \Delta, \Pi \rangle} \qquad \textit{otherwise}, \end{cases}$
- $(\mathsf{out}(v, M).P)_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}} \begin{cases} \mathsf{in}(\mathsf{c}'_1, y_1).\cdots.\mathsf{in}(\mathsf{c}'_\ell, y_\ell).\mathsf{out}(v, N).P_{F \setminus \{y_1, \ldots, y_\ell\}}^{\langle \Gamma, \Delta, \Pi \rangle} \\ \qquad \textit{if } \{N/M\} \in \Delta, \{y_1, \ldots, y_\ell\} \subseteq F \cup N, \\ \qquad \forall i \in \{1, \ldots, \ell\}, \\ \qquad \langle R_i, R, \mathsf{c}_i M, y_i \rangle \in \Theta \wedge (\mathsf{c}_i, y'_i, \mathsf{c}'_i) \in \xi, \\ \mathsf{out}(v, M).P_F^{\langle \Gamma, \Delta, \Pi \rangle} \qquad \textit{otherwise}, \end{cases}$
- $(\textit{if } M =_E N \textit{ then } P \textit{ else } Q)_F^{\langle \Gamma, \Delta, \Pi \rangle} \mathrel{\hat{=}}$
  $\begin{cases} P_F^{\langle \Gamma, \Delta, \Pi \rangle} & \textit{if } (M =_E N, \mathsf{true}) \in \Pi, \\ Q_F^{\langle \Gamma, \Delta, \Pi \rangle} & \textit{if } (M =_E N, \mathsf{false}) \in \Pi, \\ \textit{if } M =_E N \textit{ then } P_F^{\langle \Gamma, \Delta, \Pi \rangle} \textit{ else } Q_F^{\langle \Gamma, \Delta, \Pi \rangle} & \textit{otherwise.} \end{cases}$

with F initially equals to $\{y_1, \ldots, y_\ell \mid \langle R_{u_i}, R, M, \mathsf{c}, y \rangle \in \Theta\}$.

Process $\mathsf{in}(\mathsf{c}_1, y'_1).!\mathsf{out}(\mathsf{c}'_1, y'_1) \mid \cdots \mid \mathsf{in}(\mathsf{c}_\ell, y'_\ell).!\mathsf{out}(\mathsf{c}'_\ell, y'_\ell)$ models the receiving behaviour of process R in the coalition. The coalition specifies which channel is use to receive data. The received data on a channel are referred to as a distinct fresh variable. The received data is sent out over a distinct private channel. The association of channels and variables is modelled in ξ. This sending behaviour is used for the process $R^{\langle \Gamma, \Delta, \Pi \rangle}$ to read the data when it is needed. Process $R^{\langle \Gamma, \Delta, \Pi \rangle}$

models the sending behaviour, substitution of terms, assignments of evaluations. F captures the variables which are in $\{y_1, \ldots, y_\ell\}$ and has not been read in yet.

Given a set of users R_U and a coalition specification $\langle \Theta, \Delta, \Pi \rangle$ for them, the coalition is now modelled as $R_U^{\langle \Theta,\Delta,\Pi \rangle} \mathrel{\hat{=}} \nu\Omega.(R_{u_1}^{\langle \Theta,\Delta,\Pi \rangle} \mid \cdots \mid R_{u_m}^{\langle \Theta,\Delta,\Pi \rangle})$ where $\Omega = \{\mathsf{c} \mid \langle R_{u_i}, R_{u_j}, M, \mathsf{c}, y \rangle \in \Theta\}$.

Example 6.10 (coalition)**.** *In the system in Example 6.2, suppose two users U_i and U_j cooperate in the following way: U_i sends his own request and the request from U_j; U_j sends his own request and the request from U_i.*

$$\begin{aligned} R_{U_i} := {} & \nu\mathtt{id}_i.\big((\nu\mathtt{req}_i.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_i, \mathtt{req}_i))) \mid (\nu\mathtt{req}'_i.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_i, \mathtt{req}'_i))) \\ & |!(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_i, \mathtt{req})))\big) \\ R_{U_j} := {} & \nu\mathtt{id}_j.\big((\nu\mathtt{req}_j.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_j, \mathtt{req}_j))) \mid (\nu\mathtt{req}'_j.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_j, \mathtt{req}'_j))) \\ & |!(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req})))\big). \end{aligned}$$

The coalition can be specified as $\Theta = \{\langle R_{u_j}, R_{u_i}, \mathtt{req}_j, \mathsf{c}_j, y_j \rangle, \langle R_{u_i}, R_{u_j}, \mathtt{req}_i, \mathsf{c}_i, y_i \rangle\}$, $\Delta = \{\{(\mathtt{id}_i, y_j)/(\mathtt{id}_i, \mathtt{req}'_i)\}, \{(\mathtt{id}_j, y_i)/(\mathtt{id}_j, \mathtt{req}'_j)\}\}$ and $\Pi = \emptyset$.

According to Definition 6.15, the process $(R_{U_i} \mid R_{U_j})^{\langle \Theta,\Delta,\Pi \rangle}$ is as follows:

$$\begin{aligned} & \nu\mathsf{c}'_i.\nu\mathsf{c}'_j. \\ & \Big(\big(\nu\mathtt{id}_i.((\nu\mathtt{req}_i.\mathsf{out}(\mathsf{c}_i, \mathtt{req}_i).\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_i, \mathtt{req}_i))) \mid (\nu\mathtt{req}'_i.\mathsf{in}(\mathsf{c}'_j, y_j).\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_i, y_j))) \\ & \qquad |!(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_i, \mathtt{req}))))\big) \mid \\ & \big(\nu\mathtt{id}_j.((\nu\mathtt{req}_j.\mathsf{out}(\mathsf{c}_j, \mathtt{req}_j).\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_j, \mathtt{req}_j))) \mid (\nu\mathtt{req}'_j.\mathsf{in}(\mathsf{c}'_i, y_i).\mathsf{out}(\mathsf{c}_p, (\mathtt{id}_j, y_i))) \\ & \qquad |!(\nu\mathtt{req}.\mathsf{out}(\mathsf{c}_p, (\mathtt{id}, \mathtt{req}))))\big) \mid \\ & \mathsf{in}(\mathsf{c}_i, y'_i).!\mathsf{out}(\mathsf{c}'_i, y'_i) \mid \mathsf{in}(\mathsf{c}_j, y'_j).!\mathsf{out}(\mathsf{c}'_i, y'_j)\Big). \end{aligned}$$

6.3 Formalising the privacy notions

In our framework, the foundational property data-privacy, is formalised in a classical way as strong secrecy: equivalence of two processes where a variable is instantiated differently [Bla04]. Based on this property, we formalise enforced-privacy, independency-of-privacy and independency-of-enforced-privacy using the formalisation of collaboration. Using the formalisation of coalition, four corresponding coalition privacy properties are formalised.

6.3.1 Enforced-privacy

Enforced-privacy is the unlinkability of a target user to his data even when the user collaborates with the adversary. Different collaborations impact privacy differently, so when we say a protocol satisfies enforced-privacy, it always refers to a specific collaboration specification.

Similar as in receipt-freeness and coercion-resistance in e-voting [DKR09], when a protocol P_w satisfies enforced-privacy w.r.t. a target data τ (which belongs to role R_i) and a collaboration specification $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$ defined on process $\hat{R}_i$ (where $R_i := \nu\mathtt{id}_i.\nu\tau.\hat{R}_i$), there exists a process P_f for the target user to execute, such that the adversary cannot distinguish between real collaboration with $\tau = \mathtt{t}_1$ and fake collaboration (by means of process P_f) with $\tau = \mathtt{t}_2$. In the epistemic notion of coercion-resistance, enforced-privacy can be defined as the existence of a *counter-strategy* for the target user to achieve his own goal, but the adversary cannot distinguish it from the target user following the adversary's instructions [KT09].

Definition 6.11 (epriv). *A well-formed protocol P_w satisfies enforced-privacy (epriv) w.r.t. target data τ and collaboration specification $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, if there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu \mathsf{c}_{out}.\nu \mathsf{c}_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \varPsi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]$, we have*

1. $\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out}, \cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}$,
2. $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f]]$,

where $\tau \in \mathsf{bn}(R_i)$, $R_i := \nu \mathtt{id}_i.\nu\tau.\hat{R}_i$, $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ is defined on $\hat{R}_i$, t is a free name representing a piece of data, and $\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out}, \cdot)} := \nu \mathsf{c}'_{out}.(\mathcal{C}[P_f] \mid !\mathsf{in}(\mathsf{c}'_{out}, x))$.

The process $\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}$ models the behaviour of the collaborating target user. The behaviour of the adversary in the collaboration is implicitly modelled as Q in the context $\mathcal{C}[\_] := \nu \mathsf{c}_{out}.\nu \mathsf{c}_{in}.(\_ \mid Q)$. Thus a specific collaboration is modelled as $\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]$. Note that sometimes the target data in the collaboration is not decided by $\{t/\tau\}$, but by the context $\mathcal{C}[\_]$. The target data is actually instantiated by $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \varPsi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]$. The first equivalence shows that even if the context $\mathcal{C}[\_]$ is able to decide the target data, the target user can still actually instantiate the target data with $\mathtt{t}_2$ by executing the process P_f. The second equivalence shows that the adversary cannot distinguish the target user following the collaboration in process $\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}$ from executing the process P_f, in the context of the adversary collaboration $\mathcal{C}[\_]$.

6.3.2 Independency-of-privacy

Next, we account for attacking third parties. Based on data-privacy, we define independency-of-privacy to capture privacy when a set of third parties collaborate with the adversary. As different sets of third parties may differently influence the target user's privacy, and since different collaboration amongst the same third parties leads to different privacy properties, independency-of-privacy is defined with respect to a set of third parties and a collaboration specification between them and the adversary.

Definition 6.12 (third parties). *Given a well-formed protocol P_w and an instance of the target user $\hat{R}_i\{\mathsf{id}/id, t/\tau\}$, a set of third parties is defined as a set of users $R_U := R_{u_1} \mid \cdots \mid R_{u_m}$ where $\forall i \in \{1, \cdots, m\}, R_{u_i} \neq \hat{R}_i\{\mathsf{id}/id, t/\tau\}$. We use R_T to denote a set of attacking third parties and R_D to denote a set of defending third parties.*

The collaboration between a set of attacking third parties R_T and the adversary is expressed as a collaboration specification $\langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle$ defined on process R_T.

The behaviour of the third parties in the collaboration is modelled as $R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}$. Inspired by the domain-specific formal definitions, vote-independence [DLL11] in e-voting and independency-of-prescribing-privacy [ESORICS12] in e-health, the generic property independency-of-privacy is defined as follows: a well-formed protocol P_w satisfies independency-of-privacy w.r.t. $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle)$ and $\tau \in \mathsf{bn}(R_i)$, if the adversary cannot distinguish the honest target user executing role R_i with $\tau = \mathtt{t}_1$ from the same user with $\tau = \mathtt{t}_2$, even when the set of third parties R_T collaborates with the adversary according to collaboration specification $\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle$.

Definition 6.13 (ipriv). *A well-formed protocol P_w satisfies independency-of-privacy (*ipriv*) w.r.t. data τ and attacking third parties $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle)$ if*

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}],$$

where $\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle$ is a collaboration specification of process R_T.

If the equivalence holds, then despite this collaboration, adversary cannot distinguish $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}$ in which the target user uses $\tau = \mathtt{t}_1$ from $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}$ in which the target user uses $\tau = \mathtt{t}_2$.

6.3.3 Independency-of-enforced-privacy

We define independency-of-enforced-privacy (iepriv) based on epriv in a similar fashion as ipriv. More precisely, iepriv of a protocol P_w is defined w.r.t. target data $\tau \in \mathsf{bn}(R_i)$, a collaboration specification $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$ defined on process $\hat{R}_i$ with $R_i := \nu \mathtt{id}_i.\nu\tau.\hat{R}_i$, and a set of attacking third parties together with a collaboration specification defined on the third parties processes $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle)$. A well-formed protocol P_w satisfies iepriv w.r.t. $\tau, \langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle, (R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle)$, if there exists a closed plain process P_f for the target user to execute, such that, despite the help of third parties R_T according to $\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle$, the adversary cannot distinguish between the target user collaborating with $\tau = \mathtt{t}_1$, and him really using $\tau = \mathtt{t}_2$ but faking collaboration for $\tau = \mathtt{t}_1$ by P_f.

Definition 6.14 (iepriv). *A well-formed protocol P_w satisfies independency-of-enforced-privacy (*iepriv*) w.r.t. τ, $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$, and $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle)$, if there exists a closed plain process P_f, s.t. for any $\mathcal{C}[\_] := \nu \mathsf{c}_{out}.\nu \mathsf{c}_{in}.(\_ | Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon \rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T]$, we have*

1. $\mathcal{C}[P_f]^{\setminus \mathsf{out}(\mathsf{c}_{out}, \cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}$,
2. $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]$,

where $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$ is a collaboration specification for $\hat{R}_i$, and $\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle$ is a collaboration specification of process R_T.

This formalisation adds the collaboration of third parties $R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}$ to Definition 6.11.

6.3.4 Coalition privacy properties

Corresponding to each privacy property defined above, we define coalition privacy properties which take into account defending third parties.

Definition 6.15 (defensive coalition). *Given an instance of the target user* $\hat{R}_i\{\mathsf{id}/id, t/\tau\}$*, a set of defending third parties* R_D*, and a coalition specification* $\langle \Theta, \Delta, \Pi \rangle$ *defined on* $\hat{R}_i\{\mathsf{id}/id, t/\tau\}|R_D$*, the coalition is modelled as* $\nu\Omega.(\hat{R}_i\{\mathsf{id}/id, t/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi \rangle}$*, where* $\Omega = \{\mathsf{c} \mid \langle R_{u_i}, R_{u_j}, M, \mathsf{c}, y \rangle \in \Theta\}$*. The target user's behaviour in the coalition is* $\hat{R}_i\{\mathsf{id}/id, t/\tau\}^{\langle \Theta, \Delta, \Pi \rangle} \mathrel{\hat{=}} \nu\eta.((\hat{R}_i\{\mathsf{id}/id, t/\tau\})^{\langle \Gamma, \Delta, \Pi \rangle}|P_\gamma)$*, where* η*is a set of fresh channels* $\{\mathsf{c}'_i, \ldots, \mathsf{c}'_\ell\}$*,* $\Gamma = \{\langle \hat{R}_i\{\mathsf{id}/id, t/\tau\}, R_{u_j}, M, \mathsf{c}, y \rangle \mid \langle \hat{R}_i\{\mathsf{id}/id, t/\tau\}, R_{u_j}, M, \mathsf{c}, y \rangle \in \Theta\}$*, and* $P_\gamma := \mathsf{in}(\mathsf{c}_1, y'_1).!\mathsf{out}(\mathsf{c}'_1, y'_1) \mid \cdots \mid \mathsf{in}(\mathsf{c}_\ell, y'_\ell).!\mathsf{out}(\mathsf{c}'_\ell, y'_\ell)$ *with* $\{y'_1, \ldots, y'_\ell\}$ *being fresh variables,* $\{(\mathsf{c}_1, \ldots, \mathsf{c}_\ell\} = \{\mathsf{c} \mid \langle R_{u_i}, \hat{R}_i\{\mathsf{id}/id, t/\tau\}, M, \mathsf{c}, y \rangle \in \Theta\}$*. The third parties' behaviour in the coalition is* $R_D^{\langle \Theta, \Delta, \Pi \rangle}$.

Coalition-privacy Intuitively, coalition-privacy means that a target user's privacy is preserved due to the cooperation of a set of defending third parties. A well-formed protocol P_w satisfies coalition-privacy w.r.t. data $\tau \in \mathsf{bn}(R_i)$ and $(R_D, \langle \Theta, \Delta, \Pi \rangle)$, where $\langle \Theta, \Delta, \Pi \rangle$ is defined on $\hat{R}_i \mid R_D$ and $R_i := \nu\mathtt{id}_i.\nu\tau.\hat{R}_i$, if the adversary cannot distinguish an honest user in role R_i using $\tau = \mathtt{t_1}$ from the user actually using $\tau = \mathtt{t_2}$ while helped by a set of defending third parties.

Definition 6.16 (cpriv). *A well-formed protocol* P_w *satisfies coalition-privacy (*cpriv*) w.r.t. data* τ *and coalition* $(R_D, \langle \Theta, \Delta, \Pi \rangle)$ *if*

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t_1}/\tau\} \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t_2}/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi \rangle}],$$

where $\langle \Theta, \Delta, \Pi \rangle$ *is a coalition specification defined on* $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t_2}/\tau\} \mid R_D$.

In the definition, the coalition is modelled as $\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t_2}/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi \rangle}$, where the target user instantiates the target data with $\mathtt{t_2}$. The equivalence shows that the adversary cannot distinguish the target user instantiating the target data with $\mathtt{t_2}$ in the coalition from the target user instantiating the target data with $\mathtt{t_1}$. Thus, coalition-privacy captures privacy when there exists a set of third parties cooperating with the target user following a pre-defined coalition specification.

Coalition-enforced-privacy Taking into account defending third parties, we define coalition-enforced-privacy based on enforced-privacy. As before, coalition-enforced-privacy specifies a target data τ and a collaboration specification of the target user $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle$. Similar as in coalition-privacy, coalition-enforced-privacy specifies a set of defending third parties R_D and a coalition specification $\langle \Theta, \Delta, \Pi \rangle$ as well. In coalition-enforced-privacy, the target user both cooperates with the adversary and defending third parties. Similar to enforced-privacy, we assume that the target user lies to the adversary if it is possible. We do not assume that the target user lies to the defending third parties, as they help the target user maintain privacy.

Intuitively, coalition-enforced-privacy means that a target user is able to lie to the adversary about his target data when helped by defending third parties – the adversary cannot tell whether the user lied. This property is modelled as the combination of coalition-privacy and enforced-privacy: a protocol P_w satisfies coalition-enforced-privacy w.r.t $\tau \in \mathsf{bn}(R_i), \langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and $(R_D, \langle \varTheta, \varDelta, \varPi\rangle)$, for $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ a collaboration specification defined on $\hat{R}_i$ ($R_i := \nu\mathtt{id}_i.\nu\tau.\hat{R}_i$), and $\langle \varTheta, \varDelta, \varPi\rangle$ a coalition specification defined on the target user and R_D, if there exists a process P_f, such that the adversary cannot distinguish between genuine collaboration with $\tau = \mathtt{t}_1$ and faking collaboration using P_f with the help of the coalition for $\tau = \mathtt{t}_2$.

Definition 6.17 (cepriv). *A well-formed protocol P_w satisfies coalition-enforced-privacy (cepriv) w.r.t. data τ, $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and $(R_D, \langle \varTheta, \varDelta, \varPi\rangle)$, if there exists a closed plain process P_f, such that for any $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \varPsi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D]$, we have*

1. $\nu\varOmega.(\nu\eta.(\mathcal{C}[P_f]^{\setminus \mathsf{out}(\mathsf{c}'_{out}, \cdot)} \mid P_\gamma) \mid R_D^{\langle \varTheta, \varDelta, \varPi\rangle}) \approx_\ell \nu\varOmega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \varTheta, \varDelta, \varPi\rangle}$,
2. $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\nu\varOmega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle \varTheta, \varDelta, \varPi\rangle})]$,

where $\varOmega$, η, P_γ are defined in Definition 6.15, $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ is defined on $\hat{R}_i$, $\langle \varTheta, \varDelta, \varPi\rangle$ is a coalition specification defined on $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$.

The collaboration between the target user and the adversary instantiating the target data with $\mathtt{t}_1$ is modelled by the equivalence $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \varPsi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D]$. The target user's actual behaviour of instantiating the target data with $\mathtt{t}_2$ in process P_f is modelled as the first equivalence. The second equivalence shows that the adversary cannot distinguish the process in which the target user follows the collaboration with the adversary from the process in which the target user lies to the adversary with the help of defending third parties.

Coalition-independency-of-privacy Similarly, we define coalition-independency -of-privacy with respect to a target data τ, a set of attacking third parties with a collaboration specification $(R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, and a set of defending third parties R_D with a coalition specification $\langle \varTheta, \varDelta, \varPi\rangle$. Note that we require that there is no intersection between attacking third parties and defending third parties, i.e., $R_T \cap R_D = \emptyset$, as we assume a third party cannot be both attacking and defending at the same time. A well-formed protocol P_w satisfies coalition-independency-of -privacy w.r.t. τ, $(R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$ and $(R_D, \langle \varTheta, \varDelta, \varPi\rangle)$, if the adversary, even with the collaboration of a set of attacking third parties, cannot distinguish the target user instantiating $\tau = \mathtt{t}_1$ from the target user actually instantiating $\tau = \mathtt{t}_2$ in the coalition with the help of defending third parties.

Definition 6.18 (cipriv). *A well-formed protocol P_w satisfies coalition-independency-of-privacy (cipriv) w.r.t. data τ, $(R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, and $(R_D, \langle \varTheta, \varDelta, \varPi\rangle)$, if*

$$\begin{aligned} &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\varOmega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \varTheta, \varDelta, \varPi\rangle}) \mid R_T^{\langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}], \end{aligned}$$

where $\langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle$ is a collaboration specification of process R_T, $\langle \varTheta, \varDelta, \varPi\rangle$ is a coalition specification defined on $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$.

Coalition-independency-of-enforced-privacy Finally, we consider the case combining all situations together: the target user collaborates with the adversary following $\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, a set of attacking third parties R_T collaborate with the adversary following $\langle \mathit{\Psi}^t, \mathit{\Phi}^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle$, and a set of defending third parties R_D and a coalition $\langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle)$. We formally define coalition-independency-of-enforced-privacy below.

Definition 6.19 (ciepriv). *A well-formed protocol P_w satisfies coalition-independency-of-enforced-privacy (*ciepriv*) w.r.t. τ, $\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, $(R_T, \langle \mathit{\Psi}^t, \mathit{\Phi}^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$ and $(R_D, \langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle)$, if there exists a closed plain process P_f such that for any context $\mathcal{C}[\_] := \nu \mathsf{c}_{out}.\nu \mathsf{c}_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \mathit{\Psi}, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\} \mid R_T \mid R_D]$, we have*

1. $\nu\mathit{\Omega}.(\nu\eta.(\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out}, \cdot)} \mid P_\gamma) \mid R_D^{\langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle}) \approx_\ell \nu\mathit{\Omega}.((\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_2/\tau\} | R_D)^{\langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle})$,
2. $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T^{\langle \mathit{\Psi}^t, \mathit{\Phi}^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]$
$$\approx_\ell \mathcal{C}_{P_w}[\nu\mathit{\Omega}.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle}) \mid R_T^{\langle \mathit{\Psi}^t, \mathit{\Phi}^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}],$$

where $\mathit{\Omega}$, η, P_γ are defined in Definition 6.15, $\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ is a collaboration specification defined on $\hat{R}_i$, $\langle \mathit{\Psi}^t, \mathit{\Phi}^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle$ is a collaboration specification defined on R_T, $\langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle$ is a coalition specification defined on $\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_2/\tau\} \mid R_D$.

As certain coalitions may fail to maintain privacy, the coalition privacy properties can be generalised by requiring the existence of a successful coalition. The general version of coalition privacy properties allow us to reason about the existence of a coalition (a strategy) such that a user's privacy is preserved. How to find such a coalition is an interesting topic for studying coalition privacy properties. Each property defined in the above can be instantiated in many different forms by specifying the parameters of the property (such as target data, collaboration, coalition). Furthermore, only the target user is allowed to lie to the adversary – we do not consider lying third parties. Properties, ipriv, iepriv, cipriv and ciepriv, can be extended by allowing third parties to lie.

6.3.5 An example of coalition privacy

We modelled a system in Example 6.2 and defined a coalition $\langle \mathit{\Psi}, \mathit{\Phi}, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ of two users in the system in Example 6.10. According to Definition 6.16, the system does not satisfy cpriv w.r.t. target data `req` and coalition $\langle \mathit{\Theta}, \mathit{\Delta}, \mathit{\Pi}\rangle$. The reason is that without coalition, all the requests the adversary observed are different; with the coalition, the adversary can detect that two requests are sent twice.

We show that the system in Example 6.2 may satisfy cpriv with respect to a different coalition. For instance, the following coalition specification in which the two users swap their requests.

$$\langle \mathit{\Theta}', \mathit{\Delta}', \mathit{\Pi}'\rangle = \langle \emptyset, \{\{(\mathtt{id}_i, \mathtt{req}_j)/(\mathtt{id}_i, \mathtt{req}_i\}, \{(\mathtt{id}_j, \mathtt{req}_i)/(\mathtt{id}_j, \mathtt{req}_j\}), \emptyset\rangle.$$

According to Definition 6.15, the coalition process is as follows.

$$\begin{aligned}\nu\Omega.(R_{U_i} \mid R_{U_j})^{\langle\Theta,\Delta,\Pi\rangle} \;\hat{=}\; & \Big(\nu\mathtt{id}_i.\big((\nu\mathsf{req}_i.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_j))) \mid \\ & \qquad !(\nu\mathsf{req}.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}))))\Big) \mid \\ & \Big(\nu\mathtt{id}_j.\big((\nu\mathsf{req}_j.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_j,\mathtt{req}_i))) \mid \\ & \qquad !(\nu\mathsf{req}.\mathsf{out}(\mathsf{c}_p,(\mathtt{id},\mathtt{req}))))\Big).\end{aligned}$$

Since the adversary cannot detect who sent which request, a user's request is anonymous. This coalition prevents the situation, in which one only user is involved and thus his request is known by the adversary.

Now consider the following collaboration specification in which the user U_i needs to forward the private message to the adversary.

$$\langle\Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle = \langle\{(\mathtt{id}_i, \mathtt{req}_i)\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle.$$

The collaboration behaviour of U_i is modelled as follows:

$$\begin{aligned}R_{U_i}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle} \;\hat{=}\; \nu\mathtt{id}_i.(\, & (\nu\mathsf{req}_i.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_i)).\mathsf{out}(\mathsf{c}_{out},(\mathtt{id}_i,\mathtt{req}_i))) \\ & |!(\nu\mathsf{req}.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req})))).\end{aligned}$$

The system does not satisfy cepriv w.r.t. $\mathtt{req}$, collaboration $\langle\Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and coalition $(R_{U_j}, \langle\Theta, \Delta, \Pi\rangle)$, since the system does not satisfies cpriv w.r.t. $\mathtt{req}$ and coalition $(R_{U_j}, \langle\Theta, \Delta, \Pi\rangle)$. However, the system satisfies cepriv w.r.t. $\mathtt{req}$, collaboration $\langle\Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and coalition $(R_{U_j}, \langle\Theta', \Delta', \Pi'\rangle)$. Because there exists a process

$$\begin{aligned}P_f := (&\nu\mathtt{id}_i.(\nu\mathsf{req}_i.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_j)).\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_i))) \mid \\ & !(\nu\mathsf{req}.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req})))).\end{aligned}$$

such that the two equivalence in Definition 6.17 are satisfied. That is, context $\mathcal{C}[\_]$ satisfies

$$\mathcal{C}_{ex}[\mathcal{C}[R_{U_i}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_{U_j}] \approx_\ell \mathcal{C}_{ex}[R_{U_i}^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_{U_j}],$$

Since U_i does not read information from the adversary, we have the context being empty and

$$R_{U_i}^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} := R_{U_i}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}$$

with $\mathsf{c}_{out} = \mathsf{c}'_{out}$.

Accordingly, $\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) :=$

$$\begin{aligned}&\big(\nu\mathtt{id}_i.(\nu\mathsf{req}_i.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_j)).\mathsf{out}(\mathsf{c}_{out},(\mathtt{id}_i,\mathtt{req}_i)))^{\backslash\mathsf{out}(\mathsf{c}_{out},\cdot)} \mid \\ &\qquad\qquad !(\nu\mathsf{req}.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}))))\big) \mid \\ &\big(\nu\mathtt{id}_j.(\nu\mathsf{req}_j.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_j,\mathtt{req}_i))) \;|!(\nu\mathsf{req}.\mathsf{out}(\mathsf{c}_p,(\mathtt{id},\mathtt{req}))))\big).\end{aligned}$$

Since $(\nu\mathsf{req}_i.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_j)).\mathsf{out}(\mathsf{c}_{out},(\mathtt{id}_i,\mathtt{req}_i)))^{\backslash\mathsf{out}(\mathsf{c}_{out},\cdot)}$ works the same as $(\nu\mathsf{req}_i.\mathsf{out}(\mathsf{c}_p,(\mathtt{id}_i,\mathtt{req}_j)))$, we have the first equivalence,

$$\nu\Omega.\big(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_{U_j}^{\langle\Theta,\Delta,\Pi\rangle}\big) \approx_\ell \nu\Omega.(R_{U_i} \mid R_{U_j})^{\langle\Theta,\Delta,\Pi\rangle}.$$

Similarly, since

$$R_{U_i}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle} \mid R_{U_j} \approx_\ell P_f \mid R_{U_j}^{\langle\Theta,\Delta,\Pi\rangle}$$

we have the second equivalence,

$$\mathcal{C}_{ex}[R_{U_i}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle} \mid R_{U_j}] \approx_\ell \mathcal{C}_{ex}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_{U_j}^{\langle\Theta,\Delta,\Pi\rangle})].$$

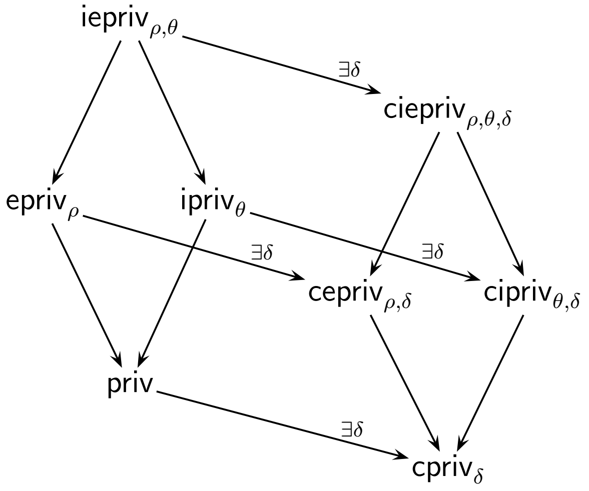
$\mathsf{iepriv}_{\rho,\theta}$
$\exists\delta$
$\mathsf{ciepriv}_{\rho,\theta,\delta}$
epriv_{ρ}
ipriv_{θ}
$\exists\delta$
$\exists\delta$
$\mathsf{cepriv}_{\rho,\delta}$
$\mathsf{cipriv}_{\theta,\delta}$
priv
$\exists\delta$
cpriv_{δ}

Figure 6.1: Relations of the privacy notions

6.4 Relations between the privacy notions

We show the relations between the privacy properties in Figure 6.1: we use ρ to denote the specification of a target user's collaboration with the adversary $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, θ to denote the specification of a set of attacking third parties and their collaboration with the adversary $(R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, and δ to denote the specification of a set of defending third parties and their coalition with the target user $(R_D, \langle \varTheta, \varDelta, \varPi\rangle)$.

The left diamond in Figure 6.1 shows the relations between privacy properties which do not consider defending third parties while the right diamond shows the relations between privacy properties which consider defending third parties. In the left diamond, epriv_ρ and ipriv_θ are stronger than priv, meaning that if a protocol satisfies epriv_ρ or ipriv_θ, then the protocol satisfies priv. Intuitively, if the adversary cannot break privacy with the help from the target user (in epriv_ρ) or from a set of attacking third parties (in ipriv_θ), the adversary cannot break privacy without any help (in priv). Similarly, if the adversary cannot break privacy with the help from both target user and attacking third parties (in $\mathsf{iepriv}_{\rho,\theta}$), the adversary cannot break privacy with the help from only one of them (in epriv_ρ and ipriv_θ). Thus, $\mathsf{iepriv}_{\rho,\theta}$ is stronger than both enforced-privacy$_\rho$ and ipriv_θ. This is described as Theorem. 6.20.

Theorem 6.20. *(1) $\forall\theta$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{epriv}_\rho$, (2) $\forall\rho$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{ipriv}_\theta$, (3) $\forall\rho$, $\mathsf{epriv}_\rho \implies \mathsf{priv}$, and (4) $\forall\theta$, $\mathsf{ipriv}_\theta \implies \mathsf{priv}$.*

Proof sketch: The proof of $\forall\rho$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{ipriv}_\theta$ and $\forall\rho$, $\mathsf{epriv}_\rho \implies \mathsf{priv}$ follows the strategy of how to prove coercion-resistance $\implies$ receipt-freeness $\implies$ vote-privacy given by Delaune et al. [DKR09]. For all ρ, when a protocol satisfies epriv_ρ, for an adversary context $\mathcal{C}[\_]$, three equivalences in Definition 6.11 hold. From the equivalences, we can deduce that $\mathcal{C}_{P_w}[\hat{R}_i^{\langle \varPsi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f]]$. By applying the evaluation context $\nu \mathsf{c}'_{out}.(\_ \;|!\mathsf{in}(\mathsf{c}'_{out}, x))$ on both side of the equivalence, we prove that $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)}]$. Because of the first equivalence in Definition 6.11: $\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_2/\tau\}$, we deduce the equivalence $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}]$. This coincides with the equivalence in Definition 6.3. Thus we prove that $\forall\rho$, $\mathsf{epriv}_\rho \implies \mathsf{priv}$. Similarly we prove $\forall\rho$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{ipriv}_\theta$.

$\forall\theta$, $\mathsf{ipriv}_\theta \implies \mathsf{priv}$ can be proved as follows: for an adversary context $\mathcal{C}[\_] := \nu \mathsf{c}^t_{out}.\nu \mathsf{c}^t_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset \wedge \mathcal{C}_{P_w}[\mathcal{C}[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]] \approx_\ell \mathcal{C}_{P_w}[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]$]), we show that $\mathsf{ipriv}_\theta \implies \mathsf{priv}$. By applying $\mathcal{C}[\_]$ and the evaluation context $\nu \mathsf{c}^t_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}^t_{out}, x))$ on both side of the equivalence in Definition 6.13, we have $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T]$. By applying rule $!P \equiv P \mid !P$, the third parties' behaviour R_T is absorbed by the environment. Thus, the equivalence in Definition 6.3 is satisfied. Similarly reasoning holds for proving $\forall\theta$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{epriv}_\rho$. Precise proofs are available in Appendix B.2. □

Moreover, the implication relations in Theorem. 6.20 are uni-directional, in the sense that we can disprove the opposite directions by presenting counter-examples (see details in Appendix B.2). We can apply the same technique to prove the relations in the right diamond. Thus we have the following theorem. Precise proofs are available in Appendix B.3.

Theorem 6.21. *(1) $\forall\theta$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \implies \mathsf{cepriv}_{\rho,\delta}$, (2) $\forall\rho$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \implies \mathsf{cipriv}_{\theta,\delta}$, (3) $\forall\rho$, $\mathsf{cepriv}_{\rho,\delta} \implies \mathsf{cpriv}_\delta$, and (4) $\forall\theta$, $\mathsf{cipriv}_{\theta,\delta} \implies \mathsf{cpriv}_\delta$.*

Each privacy property in the left diamond has a weaker corresponding property in the right diamond, meaning that if a protocol satisfies a privacy property in the left diamond, there exists a coalition such that the property satisfies the corresponding coalition privacy property in the right diamond. Intuitively, if a protocol preserves privacy of a target user without any help from third parties, the protocol can still preserve his privacy with the help from others.

Theorem 6.22. *(1) $\mathsf{ciepriv}_{\rho,\theta} \implies \exists\delta$, $\mathsf{ciepriv}_{\rho,\theta,\delta}$, (2) $\mathsf{epriv}_\rho \implies \exists\delta$,$\mathsf{cepriv}_{\rho,\delta}$, (3) $\mathsf{ipriv}_\theta \implies \exists\delta$, $\mathsf{cipriv}_{\theta,\delta}$, and (4) $\mathsf{priv} \implies \exists\delta$, cpriv_δ.*

Proof sketch: When a protocol satisfies priv, the equivalence in Definition 6.3 holds. It is easy to see that the equivalence in Definition 6.3 coincides with the one in Definition 6.16 when the coalition is set empty. The same reasoning holds for proving other relations in the theorem. Precise proofs are available in Appendix B.4. □

Generally, given a set of defending third parties R_D, when a protocol satisfies priv, the requirement that the protocol also satisfies cpriv_δ is the following equivalence: $\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$. When the coalition is of the form $\langle\Theta, \emptyset, \emptyset\rangle$, this requirement is satisfied. However, not all coalition specifications defined on R_D can satisfy the requirement. Therefore, even when a protocol satisfies priv, some coalition specification may fail to satisfy cpriv_δ. The observation holds for other relations in Theorem. 6.22 as well. Note that the requirement '$\exists\delta$' makes the coalition privacy properties in Theorem. 6.22 coincide with their general extensions as discussed previously in Section 6.3.4.

6.5 Application

Privacy notions modelled as strong secrecy can be captured by data-privacy. For instance, doctor anonymity (see Definition 5.5) is data-privacy where the target data is a user's identity. Various domain-specific properties, which capture privacy in domains where data-privacy is too strong to be satisfied, can be instantiated by coalition-privacy. For instance,

- strong bidding-price-secrecy for non-winning bidders (see Definition 4.1) in sealed-bid e-auctions is defined as the adversary cannot determine a bidder's bidding-price, assuming the existence of a winning bid. This can be instantiated as coalition-privacy where the target data is a bid p_b, the defending third party is the winning bidder $P_{bB}\{^d/_{p_b}\}$ and the coalition specification is $\langle\emptyset,\emptyset,\emptyset\rangle$.
- Prescribing-privacy (see Definition 5.1) is defined as the adversary cannot determine a doctor's prescription with the existence of a counter-balancing doctor. This can be instantiated as coalition-privacy where the target data is a prescription *presc*, the defending third party is the counter-balancing doctor $\big(init_{dr}\{\mathtt{d_B}/Id_{dr}\}.(!P_{dr}\{\mathtt{d_B}/Id_{dr}\} \mid main_{dr}\{\mathtt{d_B}/Id_{dr}, \mathtt{p_B}/presc\})\big)$ and the coalition specification is $\langle\emptyset,\Delta,\emptyset\rangle$ where the substitution Δ specifies how to replace the counter-balancing doctor's prescription $\{\mathtt{p_A}/\mathtt{p_B}\}$.
- Vote-privacy (see Definition 3.14) is defined as the adversary cannot determine a voter's vote with the existence of a counter-balancing voter. This can be instantiated as coalition-privacy where the target data is a vote *vote*, the defending third party is the counter-balancing voter $P_{vB}\{c/vote\}$ and the coalition specification is $\langle\emptyset,\Delta,\emptyset\rangle$ where the substitution Δ specifies how to replace the counter-balancing voter's vote $\{a/c\}$.

Enforced privacy notions like receipt-freeness or coercion-resistance in e-voting can be captured by either enforced-privacy or coalition-enforced-privacy.

- Receipt-freeness in e-voting (see Definition 3.17) can be instantiated by coalition-enforced-privacy w.r.t. *vote* and $(P_{vB}\{c/vote\},\langle\emptyset,\{\{a/c\}\},\emptyset\rangle)$ (the target data and the coalition are the same as in vote-privacy), and the collaboration specification is $\langle\Psi,\emptyset,\mathsf{c}_{out},\epsilon\rangle$ where Ψ contains all private terms generated and read-in in the target voter process. Ψ in a process R is given by $\mathsf{OutTerm}(R)$.

$$\begin{aligned}
\mathsf{OutTerm}(0) &= \emptyset\\
\mathsf{OutTerm}(P \mid Q) &= \mathsf{OutTerm}(P) \cup \mathsf{OutTerm}(Q)\\
\mathsf{OutTerm}(!P) &= \mathsf{OutTerm}(P)\\
\mathsf{OutTerm}(\nu\mathtt{n}.P) &= \{\mathtt{n}\} \cup \mathsf{OutTerm}(P) \quad \text{when } \mathtt{n} \text{ is name of base type,}\\
\mathsf{OutTerm}(\nu\mathtt{n}.P) &= \mathsf{OutTerm}(P) \text{ otherwise}\\
\mathsf{OutTerm}(\mathsf{in}(v,x).P) &= \{x\} \cup \mathsf{OutTerm}(P) \quad \text{when } \mathtt{n} \text{ is name of base type,}\\
\mathsf{OutTerm}(\mathsf{in}(v,x).P) &= \mathsf{OutTerm}(P) \text{ otherwise}\\
\mathsf{OutTerm}(\mathsf{out}(v,M).P) &= \mathsf{OutTerm}(P)\\
\mathsf{OutTerm}(\mathit{if}\ M =_E N\ \mathit{then}\ P\ \mathit{else}\ Q)&\\
&= \mathsf{OutTerm}(P) \cup \mathsf{OutTerm}(Q)
\end{aligned}$$

- In a similar way, coercion-resistance in e-voting (see Definition 3.19) is an instance of coalition-enforced-privacy w.r.t. *vote* and coalition specification $(P_{vB}\{c/vote\}, \langle\emptyset, \{\{a/c\}\}, \emptyset\rangle)$ (the target data and the coalition are the same as in vote-privacy), and the cooperation specification is $\langle\, \Psi,\, \Phi,\, \mathsf{c}_{out},\, \mathsf{c}_{in}\rangle$ where Ψ contains all private terms generated and read-in in the target voter process and Φ contains all the send out terms. Φ in a process R is given by $\mathsf{ReplaceTerm}(R)$.

$$\begin{aligned}
\mathsf{ReplaceTerm}(0) &= \emptyset \\
\mathsf{ReplaceTerm}(P \mid Q) &= \mathsf{ReplaceTerm}(P) \cup \mathsf{ReplaceTerm}(Q) \\
\mathsf{ReplaceTerm}(!P) &= \mathsf{ReplaceTerm}(P) \\
\mathsf{ReplaceTerm}(\nu \mathtt{n}.P) &= \mathsf{ReplaceTerm}(P) \\
\mathsf{ReplaceTerm}(\mathsf{in}(v, x).P) &= \mathsf{ReplaceTerm}(P) \\
\mathsf{ReplaceTerm}(\mathsf{out}(v, M).P) &= \{M\} \cup \mathsf{ReplaceTerm}(P) \\
\mathsf{ReplaceTerm}(\mathit{if}\ M =_E N\ \mathit{then}\ P\ \mathit{else}\ Q)& \\
&= \mathsf{ReplaceTerm}(P) \cup \mathsf{ReplaceTerm}(Q)
\end{aligned}$$

The two independency of privacy properties, i.e., independency-of-prescribing-privacy and independence-vote-privacy are instances of coalition-independency-of-privacy.

- Independency of prescribing-privacy (see Definition 5.3) can be instantiated as coalition-independency-of-privacy w.r.t. target data *presc*, defending third parties $\big(\mathit{init}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\}.(!P_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}\} \mid \mathit{main}_{dr}\{\mathtt{d_B}/\mathit{Id}_{dr}, \mathtt{p_B}/\mathit{presc}\})\big)$, coalition specification $\langle\emptyset, \{\{\mathtt{p_A}/\mathtt{p_B}\}\}, \emptyset\rangle$ with the attacking third parties R_i and third party collaboration $\langle\, \Psi, \emptyset, \mathsf{c}_{out}, \epsilon\rangle$ where Ψ contains all private terms generated and read-in in R_i, i.e., $\Psi = \mathsf{OutTerm}(R_i)$.
- Vote-independence [DLL11]: A voting process respects vote-independence if for all votes a and c

$$\begin{aligned}
&\mathcal{C}_v[P_{vA}\{a/vote\} \mid P_{vB}\{c/vote\} \mid P_{vC}{}^{\mathsf{c}_{out},\mathsf{c}_{in}}] \\
\approx_\ell\ &\mathcal{C}_v[P_{vA}\{a/vote\} \mid P_{vB}\{c/vote\} \mid P_{vC}{}^{\mathsf{c}_{out},\mathsf{c}_{in}}].
\end{aligned}$$

  This property can also be considered as an instance of coalition-independency -of-privacy, where the target data is and the coalition are the same as in vote-privacy, the set of attacking third parties is a third voter P_{vC}, and the collaboration specification of the third voter is $\langle\, \Psi,\, \Phi,\, \mathsf{c}_{out},\, \mathsf{c}_{in}\rangle$ where Ψ are all generated $\mathsf{OutTerm}(P_{vC})$ and Φ are all read-in terms $\mathsf{ReplaceTerm}(P_{vC})$ in the third voter process.

6.6 Conclusions

In this chapter, we considered both negative and positive influences of third parties to a target user's privacy. Furthermore, we identified a new privacy notion: coalition privacy, where third parties help defend the target user's privacy. In addition, we presented a formal framework which allows us to give domain-independent formalisations of the privacy notions accounting for third parties. We defined a standard form of protocols in which any protocol may be expressed.

To formally define enforced privacy properties and independency of privacy properties, we modelled *collaboration* between users and the adversary. A collaboration specifies precisely what information is shared and how it is shared. As such, this modelling provides the necessary flexibility for expressing various types of collaboration. To model coalition privacy properties, we proposed the notion of *coalition* in our framework. This notion formally expresses the behaviour of and information shared between a target user and a set of third parties.

In our framework, the basic privacy property data-privacy was formalised in a classical way as strong secrecy: equivalence of two processes where a variable is instantiated differently [Bla04]. Based on this property, we formalised enforced-privacy, independency-of-privacy and independency-of-enforced-privacy using the formalisation of collaboration. Using the formalisation of coalition, four corresponding coalition privacy properties were formalised.

Finally, we discussed the relations between the defined properties. We proved that these properties are hierarchically related, and we showed that various privacy definitions from literature are instances of properties in our hierarchy.

7

Conclusions and future work

As motivated in the introduction (Chapter 1), privacy is a desirable requirement in services based on the Internet. To ensure privacy against adversaries controlling the network, cryptography is widely used in protocols. However, the design and verification of cryptographic protocols are well known to be error-prone. Formal approaches have shown to be effective in addressing this problem. Therefore we argue that formalising privacy notions is a necessary step to verify privacy claims of a protocol. In particular, a strong privacy notion – enforced privacy was proposed recently to ensure privacy against bribery and coercion. In this thesis, we studied enforced privacy formally using a process algebra – the applied pi calculus.

We first studied existing requirements and formalisations of enforced privacy (Chapter 2). We found that enforced privacy is required in domains such as e-voting, e-auctions and e-health. However, formalisations of enforced privacy focused on the e-voting domain. We formalised enforced privacy in other domains: e-auctions and e-health. We formalised enforced privacy in the e-auction domain in a similar way as the formalisations in the e-voting domain (Chapter 4). We performed a case study to validate our formalisations. E-auction systems are similar to e-voting systems in the sense that roles can be naturally divided into two types: participants and authorities. In contrast, e-health systems involve a far more complex constellation of roles, some of which may not be trustworthy. Hence, protecting a user's privacy when third parties cooperating with the adversary was required in e-health. Therefore, in addition to enforced privacy, we formalised an independency of privacy property of e-health systems to capture such third parties' cooperating behaviour. Furthermore, we verified the formalised properties of a Belgian e-health protocol which was proposed for practical use. Using the experience of formalising domain-specific enforced privacy, we proposed a general formalisation of enforced privacy in a formal framework. In addition, we took third parties' influence into account. On the one hand, a third party can influence a target user's privacy negatively by cooperating with the adversary to break the user's privacy. On the other hand, a third party can influence a target user's privacy positively by cooperating with the target user to maintain the user's privacy. We generalised the formalisations of privacy properties taking into account negative third parties in the framework. In addition, we proposed privacy properties taking into account positive third parties and formalised the properties in the framework as well. Finally, we proved the relations between the formalised privacy properties in the framework.

7.1 Summary of contributions

In this thesis, we studied enforced privacy: privacy with respect to an adversary who can bribe or coerce users in addition to control the network. More specifically, we

1. studied enforced privacy in e-auctions (Chapter 4),
2. studied enforced privacy in e-health (Chapter 5), and
3. developed a formal verification framework in which we generalised enforced privacy and studied enforced privacy in the presence of others (Chapter 6).

Chapter 4: Enforced privacy in e-auctions In the study of enforced privacy in e-auctions, our main contribution is that we proposed formalisations of two privacy properties of auction protocols: bidding-price-secrecy and receipt-freeness, following definitions of vote-privacy and receipt-freeness in e-voting [DKR09]. We have modelled the AS02 protocol in the applied pi calculus, verified bidding-price-secrecy of the protocol automatically, using ProVerif, and receipt-freeness of the protocol manually.

Chapter 5: Enforced privacy in e-health The main contribution in the study of enforced privacy in e-health is that we identified three enforced privacy properties of e-health systems: enforced prescribing-privacy, independency of prescribing-privacy, independency of enforced prescribing-privacy. In addition, we are the first to provide formal definitions for them. Furthermore, we developed an in-depth applied pi model of the DLV08 e-health protocol [dDLVV08] which is rather complicated and aims for practical use in Belgium. Furthermore, we formally analysed privacy and enforced privacy properties of the protocol, as well as regular security and privacy properties. We have found ambiguities in the protocol which potentially lead to flaws on privacy, and proposed suggestions for fixing them. The formal analysis of the DLV08 protocol, together with the analysis of the AS02 protocol, provide insights on the design of protocols preserving enforced privacy.

Chapter 6: Enforced privacy in the presence of others We generalised domain-specific enforced privacy properties of the e-voting, e-auction and e-health systems. Inspired by the requirement of independency of privacy in e-health systems, we took into account privacy properties where third parties cooperate with the adversary and generalised them as independency of privacy properties. In addition, we considered privacy properties where third parties cooperate with the target user, and proposed coalition privacy. We also formalised the privacy properties in a new formal framework and formally prove their relations. In the framework, we defined a standard form of protocols which is able to represent any protocol. To formally define enforced privacy properties and independency of privacy properties, we model collaboration between users and the adversary. The collaboration allows us to precisely specify which information is shared and how it is shared, thereby providing the necessary flexibility for modelling various types of collaboration. To model coalition privacy properties, we proposed the property of coalition in our framework to formally capture the behaviour and shared information among a target user and a set of third parties. The formal framework allows us to give domain-independent formalisations of the privacy properties. We formalised data-privacy, enforced-privacy, independency-of-privacy and independency -of-enforced-privacy using the formalisation of collaboration. Using the formalisation of coalition, four corresponding coalition privacy properties are formalised. Finally, we formally discussed how the formalised privacy properties are related in a privacy hierarchy. In addition, we showed that many existing formalisations are instances of properties in our hierarchy. It appears that the formalisations of enforced privacy in other domains can benefit from our framework.

7.2 Future work

Enforced privacy is still rather new compared to classical privacy properties such as anonymity and untraceability. Enforced privacy requirements have only been formally studied in a few domains, e-voting, e-auctions and e-health. The adversary's ability to bribe or coerce users does not depend on domains. As such, enforced privacy needs to be studied in other domains. For instance, in online social networks, normally a user maintains the link between the identities and pseudonyms of his friends. The following scenario that a user may be coerced to reveal the link has been identified in social networks [BMP11]. Thus, the following requirement has been identified: a user should be able to give fake associations between pseudonyms and identities of his friends to a coercer [BMP11]. Once the requirements are identified, designing systems enforcing user's privacy is needed. Development of systems providing enforced privacy will benefit from privacy-enforcing techniques used to guarantee receipt-freeness and coercion-resistance in e-voting, e-auctions and e-health, for example chameleon bit commitments, untappable channels and zero-knowledge proofs, as used in the AS02 e-auction protocol. As design of cryptographic protocols is well-known to be error-prone, a claimed enforced privacy property needs to be formally verified. To do so, formalising such a property is a necessary step. The formalisations of a claimed enforced privacy property will benefit from our formal framework.

In the two domains where enforced privacy has been studied in this thesis, e-auctions and e-health, there is still interesting research that needs to be carried out. For instance, in e-auctions, we only formalised receipt-freeness in sealed-bid e-auctions as it is required in the literature. Privacy notions due to different types of cooperation between a bribed or coerced user and the adversary have not been formalised. Thus, an interesting research direction is to formalise such privacy notions, for example, coercion-resistance, in e-auctions. In addition, the privacy notions we defined in e-auctions (bidding-price-secrecy and receipt-freeness) aim to protect privacy for non-winning bidders. Chen et al. proposed an auction protocol which can ensure the winner's privacy as well [CLK03]. It is interesting to formalise and verify privacy and enforced privacy for this protocol.

Compared to the study of enforced privacy in the e-auction and e-voting domains, where many systems providing enforced privacy have been proposed, the study of enforced privacy in the e-health domain is in its infancy. Especially, e-health systems require a strong privacy, the combination of enforced privacy and independency of privacy, for instance, the DLV08 e-health protocol requires doctor privacy even if both pharmacists and doctors cooperate with the adversary. How to ensure such strong privacy properties in e-health remains a challenge, for instance, improving the DLV08 protocol to satisfy the independency of enforced prescribing-privacy.

Moreover, as we have already mentioned in Chapter 6, there are some interesting research directions based on the formal framework. A coalition privacy notion formalised in the framework is always with respect to a specific coalition. A coalition between a target user and third parties may fail to maintain privacy of the target user. How to find a coalition and synthesise a strategy for the coalition to satisfy some coalition privacy properties for a protocol is an interesting research direction. In addition, in the formal framework, we did not consider that the third parties cooperating with the adversary may lie to the adversary. Taking this third parties' lying into account will lead to new privacy notions. Thus, how to extend our privacy hierarchy to capture situations where a third party is coerced but has a strategy to lie to the adversary needs to be addressed.

Our work focuses on formalising enforced privacy notions. The verification of enforced privacy properties in our case studies benefit from verification techniques provided by the tool ProVerif. However, this tool may report false attacks, or not terminate. As we can see, equivalences in the enforced privacy formalisations are often complex. Developing verification techniques which are efficient for verifying such equivalences is an interesting direction. Furthermore, ProVerif can only help prove whether an equivalence hold, it cannot, for example, prove whether there exists a process in which a bribed or coerced user can successfully cheat the adversary. How to find such processes automatically remains a challenge.

A

Full proofs in Chapter 4

A.1 Full proof of receipt-freeness of AS02

We show the detailed proof of the equivalence **eq1** in Section 4.6.2.

$$\begin{aligned}
&(\text{let } \mathit{untapch} = \mathtt{untapch}_{\mathsf{b}_A} \text{ in let } \mathit{privch} = \mathtt{privch}_{\mathsf{b}_A} \text{ in}\\
&\ \text{let } \mathit{ch} = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_f{}^{\backslash \mathsf{out}(\mathtt{chc},\cdot)})\\
\approx_\ell\ &(\text{let } p_b = \mathbf{c} \ \text{ inlet } \mathit{untapch} = \mathtt{untapch}_{\mathsf{b}_A} \text{ in}\\
&\ \text{let } \mathit{privch} = \mathtt{privch}_{\mathsf{b}_A} \text{ in let } \mathit{ch} = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_b),
\end{aligned}$$

where $P_f{}^{\backslash \mathsf{out}(\mathtt{chc},\cdot)} \mathrel{\hat{=}} \nu\mathtt{chc}.(P_f \mid !\mathsf{in}(\mathtt{chc}, x))$, P_f is defined in Figure 4.12, and P_b is defined as in Figure 4.6.

In order to prove the equivalence, we can prove that $P_f{}^{\backslash \mathsf{out}(\mathtt{chc},\cdot)} \approx_\ell P_b\{\mathbf{c}/p_b\}$, i.e., $(\nu\mathtt{chc}.(P_f \mid !\mathsf{in}(\mathtt{chc}, x)) \approx_\ell P_b\{\mathbf{c}/p_b\}$.

Let $P' := (\nu\mathtt{chc}.(P_f \mid !\mathsf{in}(\mathtt{chc}, x))$ and $Q' := P_b\{\mathbf{c}/p_b\}$. Process P' is shown in Figure A.1, and process Q' is shown in Figure A.3.

$P' := \nu\mathtt{chc}.($

p1. $\mathsf{in}(\mathit{privch}, \mathit{ssk}_b).$

p2. $\mathsf{out}(\mathtt{chc}, \mathit{ssk}_b).$

p3. $\nu\ \mathit{sk}_b.\ \mathsf{out}(\mathtt{chc}, \mathit{sk}_b).$

p4. $\mathsf{out}(\mathtt{ch}, \mathsf{sign}(\mathsf{pk}(\mathit{sk}_b), \mathit{ssk}_b)).$

p5. $\nu\ r_1.\ \cdots.\ \nu\ r_a.\ \cdots.\ \nu\ r_c.\ \cdots.\ \nu\ r_m.$

p6. $\mathsf{out}(\mathtt{chc}, (r_1, \ldots, \mathsf{f}(r_a), \ldots, \mathsf{f}(r_c), \ldots, r_m)).$

p7. $\mathtt{let}\ \ \mathit{cmt}^{\mathsf{p}_1} = \mathsf{commit}(r_1, \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{no})\ \ \mathtt{in}$

p8. $\ldots$

p9. $\mathtt{let}\ \ \mathit{cmt}^{\mathsf{p}_a} = \mathsf{commit}(r_a, \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{no})\ \ \mathtt{in}$

p10. $\ldots$

p11. $\mathtt{let}\ \ \mathit{cmt}^{\mathsf{p}_c} = \mathsf{commit}(r_c, \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{yes})\ \ \mathtt{in}$

p12. $\ldots$

p13. $\mathtt{let}\ \ \mathit{cmt}^{\mathsf{p}_m} = \mathsf{commit}(r_m, \mathsf{pk}(\mathit{sk}_b), \mathsf{M}_{no})\ \ \mathtt{in}$

p14. $\mathsf{out}(\mathtt{ch}, \mathsf{sign}((\mathit{cmt}^{\mathsf{p}_1}, \ldots, \mathit{cmt}^{\mathsf{p}_m}), \mathit{ssk}_b)).$

p15. $\mathsf{out}(\mathit{untapch}, \mathsf{sign}((r_1, \ldots, r_a, \ldots, r_c, \ldots, r_m), \mathit{ssk}_b))$

p16. $\mid !\mathsf{in}(\mathtt{chc}, x))$

Figure A.1: The process $\nu\mathtt{chc}.(P_f \mid !\mathsf{in}(\mathtt{chc}, x))$.

The transitions of process P' are shown in Figure A.2.

In Figure A.2 $P'_1, \ldots, P'_7$ are sub-processes in Figure A.1.

$$\begin{array}{rl}
P' \xrightarrow{in(privch,y)} & \nu\mathsf{chc}.P'_1\{y/ssk_b\} \\
\to (\text{COMM}) & \nu\mathsf{chc}.P'_2\{y/ssk_b\} \\
\to (\text{COMM}) & \nu\mathsf{chc}.\nu sk_b.P'_3\{y/ssk_b\} \\
\xrightarrow{\nu\ x_1.\ \mathsf{out}(\mathsf{ch},x_1)} & \nu\mathsf{chc}.\nu sk_b.(P'_4 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\} \\
\to (\text{COMM}) & \nu\mathsf{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(P'_5 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\} \\
\xrightarrow{\nu\ x_2.\ \mathsf{out}(\mathsf{ch},x_2)} & \nu\mathsf{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(P'_6 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \\
 & \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1},\ldots,cmt^{\mathsf{p}_m}), ssk_b)/x_2\})\{y/ssk_b\} \\
\xrightarrow{\nu\ x_3.\ \mathsf{out}(\mathsf{chc},x_3)} & \nu\mathsf{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(P'_7 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \\
 & \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1},\ldots,cmt^{\mathsf{p}_m}), ssk_b)/x_2\} \\
 & \mid \{\mathsf{sign}((r_1,\ldots,r_m), ssk_b)/x_3\})\{y/ssk_b\}
\end{array}$$

Figure A.2: Transitions of P'

- P'_1 is the sub-process **p2** to **p16**.
- P'_2 is the sub-process **p3** to **p16**.
- P'_3 is the sub-process **p4** to **p16**.
- P'_4 is the sub-process **p5** to **p16**.
- P'_5 is the sub-process **p7** to **p16**.
- P'_6 is the sub-process **p15** to **p16**.
- P'_7 is the sub-process **p16**.

```
         $Q' :=$
q1.      $\mathsf{in}(ch_{priv}, ssk_b)$.
q2.      $\nu\ sk_b.\mathsf{out}(\mathsf{ch}, \mathsf{sign}(\mathsf{pk}(sk_b), ssk_b))$.
q3.      $\nu\ r_1.\cdots.\nu\ r_m$.
q4.      let  $cmt^{\mathsf{p}_1} = \mathsf{commit}(r_1, \mathsf{pk}(sk_b), \mathsf{M}_{no})$  in
q5.      ...
q6.      let  $cmt^{\mathsf{p}_a} = \mathsf{commit}(r_a, \mathsf{pk}(sk_b), \mathsf{M}_{no})$  in
q7.      ...
q8.      let  $cmt^{\mathsf{p}_c} = \mathsf{commit}(r_c, \mathsf{pk}(sk_b), \mathsf{M}_{yes})$  in
q9.      ...
q10.     let  $cmt^{\mathsf{p}_m} = \mathsf{commit}(r_m, \mathsf{pk}(sk_b), \mathsf{M}_{no})$  in
q11.     $\mathsf{out}(\mathsf{ch}, \mathsf{sign}((cmt^{\mathsf{p}_1}, \cdots, cmt^{\mathsf{p}_m}), ssk_b))$.
q12.     $\mathsf{out}(untapch, \mathsf{sign}((r_1, \cdots, r_m), ssk_b))$
```

Figure A.3: The bidder process $P_b\{\mathbf{c}/p_b\}$.

The transitions of process Q' are shown in Figure A.4.

In Figure A.4, $Q'_1, \ldots, Q'_4$ are sub-processes in Figure A.3.

- Q'_1 is the sub-process **q2** to **q12**.

$$\begin{array}{lll} Q' & \xrightarrow{in(privch,y)} & Q'_1\{y/ssk_b\} \\ & \xrightarrow{\nu\ x_1.\ \mathsf{out}(\mathtt{chc},x_1)} & \nu sk_b.(Q'_2 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\} \\ & \xrightarrow{\nu\ x_2.\ \mathsf{out}(\mathtt{chc},x_2)} & \nu sk_b.\nu\ r_1.\cdots.\nu\ r_m.\ (Q'_3 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \\ & & \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1}, \cdots, cmt^{\mathsf{p}_m}), ssk_b)/x_2\})\{y/ssk_b\} \\ & \xrightarrow{\nu\ x_3.\ \mathsf{out}(\mathtt{ch},x_3)} & \nu sk_b.\nu\ r_1.\cdots.\nu\ r_m.\ (Q'_4 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \\ & & \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1}, \cdots, cmt^{\mathsf{p}_m}), ssk_b)/x_2\} \\ & & \mid \{\mathsf{sign}((r_1, \cdots, r_m), ssk_b)/x_3\})\{y/ssk_b\} \end{array}$$

Figure A.4: Transitions of Q'.

- Q'_2 is the sub-process **q3** to **q12**.
- Q'_3 is the sub-process **q12**.
- Q'_4 is the sub-process 0.

We build a relation $\mathcal{R}$ as follows: $P' \mathcal{R} Q'$, $P^1 \mathcal{R} Q^1$, $P^2 \mathcal{R} Q^1$, $P^3 \mathcal{R} Q^1$, $P^4 \mathcal{R} Q^2$, $P^5 \mathcal{R} Q^2$, $P^6 \mathcal{R} Q^3$, where

$$\begin{array}{l} P^1 := (\nu\mathtt{chc}.P'_1\{y/ssk_b\}) \\ P^2 := (\nu\mathtt{chc}.P'_2\{y/ssk_b\}) \\ P^3 := (\nu\mathtt{chc}.\nu sk_b.P'_3\{y/ssk_b\}) \\ P^4 := (\nu\mathtt{chc}.\nu sk_b.(P'_4 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\}) \\ P^5 := (\nu\mathtt{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(P'_5 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\}) \\ P^6 := (\nu\mathtt{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(P'_6 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \\ \qquad\qquad\qquad\qquad\qquad\qquad \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1}, \ldots, cmt^{\mathsf{p}_m}), ssk_b)/x_2\})\{y/ssk_b\}) \\ Q^1 := Q'_1\{y/ssk_b\}, \\ Q^2 := (\nu sk_b.(Q'_2 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\}) \\ Q^3 := (\nu sk_b.\nu\ r_1.\cdots.\nu\ r_m.\ (Q'_3 \mid \{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \\ \qquad\qquad\qquad\qquad\qquad\qquad \mid \{\mathsf{sign}((cmt^{\mathsf{p}_1}, \cdots, cmt^{\mathsf{p}_m}), ssk_b)/x_2\})\{y/ssk_b\}). \end{array}$$

We prove that $P' \approx_s Q'$. Since $\mathsf{frame}(P') := \nu sk_b.\nu\ r_1.\cdots.\nu\ r_m$ and $\mathsf{frame}(Q') := \nu\mathtt{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m$, we have that $\mathsf{domain}(P') = \mathsf{domain}(Q')$ and $M =_E N$ in $\mathsf{frame}(P')$ iff $M =_E N$ in $\mathsf{frame}(Q')$, thus, we have $\mathsf{frame}(P') \approx_s \mathsf{frame}(Q')$. Therefore, $P' \approx_s Q'$.

Similarly, from the above we can prove that $P^1 \approx_s Q^1$, $P^2 \approx_s Q^1$, $P^3 \approx_s Q^1$, $P^4 \approx_s Q^2$, $P^5 \approx_s Q^2$, $P^6 \approx_s Q^3$.
(1) $\mathsf{frame}(P^1) := (\nu\mathtt{chc}.0)$ and $\mathsf{frame}(Q^1) := 0$. We can see that $\mathsf{domain}(P^1) = \mathsf{domain}(Q^1)$. According to rule NEW $-$ 0, we have that $\mathsf{frame}(P^1) \approx_s \mathsf{frame}(Q^1)$, and thus, $M =_E N$ in $\mathsf{frame}(P^1)$ iff $M =_E N$ in $\mathsf{frame}(Q^1)$. Therefore, $P^1 \approx_s Q^1$.

(2) Since $\mathsf{frame}(P^2) := (\nu\mathtt{chc}.0)$ as well, from (1) we have that $P^2 \approx_s Q^1$.

(3) Since $\mathsf{frame}(P^3) := \nu\mathtt{chc}.\nu sk_b.0$, we can see that $\mathsf{domain}(P^1) = \mathsf{domain}(Q^1)$. According to rule NEW $-$ 0, we have that $\mathsf{frame}(P^3) \approx_s \mathsf{frame}(Q^1)$ as well. Therefore, $M =_E N$ in $\mathsf{frame}(P^1)$ iff $M =_E N$ in $\mathsf{frame}(Q^1)$, and thus, $P^3 \approx_s Q^1$.

(4) $\mathsf{frame}(P^4) := \nu\mathtt{chc}.\nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\})$ and $\mathsf{frame}(Q^2) := \nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\}$. We can see that $\mathsf{domain}(p^4) = \mathsf{domain}(Q^2)$.

According to rule PAR − 0, we have that

$$\mathsf{frame}(P^4) \equiv \nu\mathtt{chc}.\nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \mid 0)\{y/ssk_b\}).$$

According to rule NEW − PAR, we have that

$$\begin{aligned}&\nu\mathtt{chc}.\nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \mid 0)\{y/ssk_b\})\\ \equiv\ &\nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \mid \nu\mathtt{chc}.0)\{y/ssk_b\}).\end{aligned}$$

According to rule NEW − 0, we have that $\nu\mathtt{chc}.0 \equiv 0$ and thus

$$\begin{aligned}&\nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \mid \nu\mathtt{chc}.0)\{y/ssk_b\})\\ \equiv\ &\nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\}).\end{aligned}$$

Therefore, $\mathsf{frame}(P^4) \equiv \nu sk_b.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\})$.
That is $\mathsf{frame}(P^4) \equiv \mathsf{frame}(Q^2)$. Therefore, we have $\mathsf{frame}(P^4) \approx_s \mathsf{frame}(Q^2)$, and thus, $M =_E N$ in $\mathsf{frame}(P^4)$ iff $M =_E N$ in $\mathsf{frame}(Q^2)$. Therefore, $P^4 \approx_s Q^2$.

(5) $\mathsf{frame}(P^5) := \nu\mathtt{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\})\{y/ssk_b\}$. We can see that $\mathsf{domain}(p^5) = \mathsf{domain}(Q^2)$. Similar as in (3), we can prove that $\mathsf{frame}(P^5) \equiv \mathsf{frame}(P^4)$. Therefore, $\mathsf{frame}(P^5) \approx_s \mathsf{frame}(Q^2)$ as well. Thus, $M =_E N$ in $\mathsf{frame}(P^5)$ iff $M =_E N$ in $\mathsf{frame}(Q^2)$. Hence, we have $P^5 \approx_s Q^2$.

(6) We have the two frames, $\mathsf{frame}(P^6)$ and $\mathsf{frame}(Q^3)$, defined as follows:

$$\begin{aligned}\mathsf{frame}(P^6) := \nu\mathtt{chc}.\nu sk_b.\nu r_1.\cdots\nu r_m.(\ &\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \mid\\ &\{\mathsf{sign}((cmt^{\mathsf{p}_1}, \ldots, cmt^{\mathsf{p}_m}), ssk_b)/x_2\})\{y/ssk_b\}.\end{aligned}$$

$$\begin{aligned}\mathsf{frame}(Q^3) := \nu sk_b.\nu\ r_1.\cdots.\nu\ r_m.\ (\ &\{\mathsf{sign}(\mathsf{pk}(sk_b), ssk_b)/x_1\} \mid\\ &\{\mathsf{sign}((cmt^{\mathsf{p}_1}, \cdots, cmt^{\mathsf{p}_m}), ssk_b)/x_2\})\{y/ssk_b\}.\end{aligned}$$

We can see that $\mathsf{domain}(P^6) = \mathsf{domain}(Q^3)$. Similar as in (1), we prove that $\mathsf{frame}(P^6) \approx_s \mathsf{frame}(Q^3)$. Thus, $M =_E N$ in $\mathsf{frame}(P^6)$ iff $M =_E N$ in $\mathsf{frame}(Q^3)$. Hence, we have $P^6 \approx_s Q^3$.

We can see that the relation $\mathcal{R}$ satisfies the definition of labelled bisimilarity (see Definition 3.11). If P' does a internal reduction to P^1, then we have $P^1\ \mathcal{R}\ Q'$. If Q' does a labelled transition to Q^1, then P' can do three internal reductions and a labelled transition to P^4, and $Q^2\ \mathcal{R}\ P^4$.

□

Proving the second equivalence holds is similar. Below, we explain how the proof runs.

Recall the proof obligation:

$$\begin{aligned}&\mathcal{C}_{AS02}[\ (\text{let } p_b = \mathbf{a} \text{ in let } untapch = \mathtt{untapch}_{\mathsf{b}_A} \text{ in}\\ &\qquad \text{let } privch = \mathtt{privch}_{\mathsf{b}_A} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_b)^{\mathtt{chc}} \mid\\ &\qquad (\text{let } p_b = \mathbf{d} \text{ in let } untapch = \mathtt{untapch}_{\mathsf{b}_B} \text{ in}\\ &\qquad \text{let } privch = \mathtt{privch}_{\mathsf{b}_B} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_B} \text{ in } P_b)]\\ \approx_\ell\ &\mathcal{C}_{AS02}[\ (\text{let } untapch = \mathtt{untapch}_{\mathsf{b}_A} \text{ in}\\ &\qquad \text{let } \ privch = \mathtt{privch}_{\mathsf{b}_A} \text{ in let } \ ch = \mathtt{ch}_{\mathsf{b}_A} \text{ in } P_f) \mid\\ &\qquad (\text{let } p_b = \mathbf{d} \text{ in let } untapch = \mathtt{untapch}_{\mathsf{b}_B} \text{ in}\\ &\qquad \text{let } privch = \mathtt{privch}_{\mathsf{b}_B} \text{ in let } ch = \mathtt{ch}_{\mathsf{b}_B} \text{ in } P_b)]\end{aligned}$$

Since the processes on each side of the equivalence consists of several concurrent processes, including the key distribution process, n bidder processes and an auctioneer process. In addition, the key distribution process and the auctioneer process contain sub-processes which are in parallel. Thus, the transitions of the process on each side are too complicated to be showed here. The proof of this equivalence is similar to the proof of the first equivalence as shown above. The difference is that for proving the second equivalence, the states and transitions are more complicated.

Generally speaking, since the contexts are the same on both sides, if the process on the left side can do a labelled transition by one of the parallel sub-processes in the context, or a internal reduction between sub-processes in the context, the corresponding sub-processes on the right hand side context can do the same transition. And Vice versa.

The transitions caused by the two bidder processes, $P_{bA}\{a/p_b\}^{\texttt{chc}}$ and $P_{bB}\{d/p_b\}$ on the left side, P_f and $P_{bB}\{d/p_b\}$ on the right side, are showed in Figure 4.13. From Figure 4.13, we can see that if the left side can do a transition by the two bidder processes, the right side can do a corresponding transition by the two bidder processes as well. Figure 4.13 also shows the change of adversary knowledge caused by transitions of the two bidder processes. We can see, form Figure 4.13, that the change of adversary knowledge is the same on both sides for most transitions. The only exception is x_5. In the proof sketch in Section 4.6.2, we showed that the adversary cannot distinguish the difference of x_5 between left and right side.

Therefore, for a state on the left side, there is a corresponding state on the right side. We build a relation between the corresponding states. The relations satisfies the second and third items in the definition of labelled bisimilarity (see Definition 3.11). Then we prove that the frames of the corresponding states are static equivalence, in a similarly way as we prove the first equivalence. Generally speaking, most frames are the same, due to the similarity of the processes on both sides. In such a way, the second equivalence is proved. □

B

Full proofs in Chapter 6

B.1 Auxiliary lemmas

Theorem 1. If $A \approx_\ell B$ and $B \approx_\ell C$, then $A \approx_\ell C$.

Proof. 1. Since $A \approx_\ell B$, according to Definition 3.11, we have $A \approx_s B$. Similarly, since $B \approx_\ell C$, we have $B \approx_s C$.

Accoding to Definition 3.8, since $A \approx_s B$, we have $\mathsf{frame}(A) \approx_s \mathsf{frame}(B)$. That is,
1) $\mathsf{domain}(\mathsf{frame}(A)) = \mathsf{domain}(\mathsf{frame}(B))$ and
2) $\forall$ terms M, N: $(M =_E N)$ in frame $\mathsf{frame}(A)$ iff $(M =_E N)$ in frame $\mathsf{frame}(B)$.
Similarly, since $B \approx_s C$, we have $\mathsf{frame}(B) \approx_s \mathsf{frame}(C)$. Thus,
3) $\mathsf{domain}(\mathsf{frame}(B)) = \mathsf{domain}(\mathsf{frame}(C))$ and
4) $\forall$ terms M, N: $(M =_E N)$ in frame $\mathsf{frame}(B)$ iff $(M =_E N)$ in frame $\mathsf{frame}(C)$.

Because of 1) and 3), we have $\mathsf{domain}(\mathsf{frame}(A)) = \mathsf{domain}(\mathsf{frame}(C))$. Because of 2) and 4), we have $\forall$ terms M, N: $(M =_E N)$ in frame $\mathsf{frame}(A)$ iff $(M =_E N)$ in frame $\mathsf{frame}(C)$. Therefore, $\mathsf{frame}(A) \approx_s \mathsf{frame}(C)$, and thus, $A \approx_s C$.

2. We build a relation $\mathcal{R}$: $A \mathrel{\mathcal{R}} C$ if $A \approx_\ell P$ and $P \approx_\ell C$ for some P.

2.1 Since $A \approx_\ell B$, according to Definition 3.11, we have that if $A \rightarrow A'$ then $B \rightarrow^* B'$ and $A' \approx_\ell B'$ for some B'. Similarly, since $B \approx_\ell C$, we have that if $B \rightarrow B$" then $C \rightarrow^* C$" and B" $\approx_\ell C$" for some C". Therefore, if $B \rightarrow^* B'$, then $C \rightarrow^* C'$ and $B' \approx_\ell C'$ for some C'. From the above, we can see, if $A \rightarrow A'$ then $C \rightarrow^* C'$ and $A' \approx_\ell B'$ and $B' \approx_\ell C'$ for some B' and C'. Thus, we have if $A \rightarrow A'$ then $C \rightarrow^* C'$ and $A' \mathrel{\mathcal{R}} C'$ for some C'. Vice versa, we prove that if $C \rightarrow C'$ then $A \rightarrow^* A'$ and $A' \mathrel{\mathcal{R}} C'$ for some A', in a simlar way.

2.2 Since $A \approx_\ell B$, according to Definition 3.11, we have that if $A \xrightarrow{\alpha} A'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(B) = \emptyset$; then $B \rightarrow^* B_1 \xrightarrow{\alpha}\rightarrow^* B'$ and $A' \approx_\ell B'$ for some B'.

5) Since $A \approx_\ell B$, we have $\mathsf{domain}(A) = \mathsf{domain}(B)$ (see second paragraph). Therefore, From the condition $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$, we have $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(B)$.

6) Since $B \approx_\ell C$, we have that if $B \rightarrow^* B_1$ then $C \rightarrow^* C_1$ and $B_1 \approx_\ell C_1$ for some C_1 (see fourth paragrph).
Since internal reduction does not introduce active substitutions, thus does not change the domain, i.e., $\mathsf{domain}(B) = \mathsf{domain}(B_1)$. Form 5), we get $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(B_1)$.
Assume $\mathsf{bn}(\alpha) \cap \mathsf{fn}(C) = \emptyset$, since internal reduction does not introduce free names, we have $\mathsf{bn}(\alpha) \cap \mathsf{fn}(C_1) = \emptyset$.
Since $B_1 \approx_\ell C_1$, we have that if $B_1 \xrightarrow{\alpha} B'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(B_1)$ and $\mathsf{bn}(\alpha) \cap$

$\mathsf{fn}(C_1) = \emptyset$; then $C_1 \rightarrow^* \xrightarrow{\alpha} \rightarrow^* C'$ and $B' \approx_\ell C'$ for some C'.

Combining 5) and 6), we have that if $A \xrightarrow{\alpha} A'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(C) = \emptyset$; then $C \rightarrow^* C_1 \rightarrow^* \xrightarrow{\alpha} \rightarrow^* C'$ and $A' \approx_\ell B'$ and $B' \approx_\ell C'$ for some C'.

Therefore, if $A \xrightarrow{\alpha} A'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(C) = \emptyset$; then $C \rightarrow^* \xrightarrow{\alpha} \rightarrow^* C'$ and $A' \mathcal{R} C'$ for some C'.

In a similar way, we can prove that if $C \xrightarrow{\alpha} C'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(C)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(A) = \emptyset$; then $A \rightarrow^* \xrightarrow{\alpha} \rightarrow^* A'$ and $A' \mathcal{R} C'$ for some A'.

From the above step 1, 2.1, and 2.2, we prove that $A \approx_\ell C$. □

Theorem 2. If $A \equiv B$ and $C \equiv D$, and $A \approx_\ell C$ then $B \approx_\ell D$.

Proof. From the following lemma in [AF01]: Static equivalence is closed by structural equivalence, we have that if $A \equiv B$, then $A \approx_s B$.
That is, accoding to Definition 3.8, if $A \approx_s B$, then
1) $\mathsf{domain}(\mathsf{frame}(A)) = \mathsf{domain}(\mathsf{frame}(B))$ and
2) $\forall$ terms M, N: $(M =_E N)$ in frame $\mathsf{frame}(A)$ iff $(M =_E N)$ in frame $\mathsf{frame}(B)$.

According to Definition 3.11, since $A \approx_\ell C$, we have, $A \approx_s C$, that is,
3) $\mathsf{domain}(\mathsf{frame}(A)) = \mathsf{domain}(\mathsf{frame}(C))$ and
4) $\forall$ terms M, N: $(M =_E N)$ in frame $\mathsf{frame}(A)$ iff $(M =_E N)$ in frame $\mathsf{frame}(C)$.

Combining 1) and 3), we have $\mathsf{domain}(\mathsf{frame}(B)) = \mathsf{domain}(\mathsf{frame}(C))$.
Combining 2) and 4), we have $\forall$ terms M, N: $(M =_E N)$ in frame $\mathsf{frame}(B)$ iff $(M =_E N)$ in frame $\mathsf{frame}(C)$. Thus, $B \approx_s C$.

We build a relation $\mathcal{R}$ as $B \mathcal{R} C$ if $B \approx_\ell C$.

Since $A \approx_\ell C$, we have that if $C \rightarrow C'$ then $A \rightarrow^* A'$ and $A' \approx_\ell C'$ for some A'.
Since internal reduction is closed under structural equivalence, we have that if $A \equiv B$ and $A \rightarrow^* A'$, then $B \rightarrow^* B'$ and $B' = A'$.
Therefore, if $C \rightarrow C'$ then $B \rightarrow^* B'$ and $B' \approx_\ell C'$ for some B'. That is, if $C \rightarrow C'$ then $B \rightarrow^* B'$ and $B' \mathcal{R} C'$ for some B'.
In a similar way, we can prove that if $B \rightarrow B'$ then $C \rightarrow^* C'$ and $B' \mathcal{R} C'$ for some C'.

5) Assume $B \xrightarrow{\alpha} B'$. According to rule STRUCT, we have that if $B \xrightarrow{\alpha} B'$ and $A \equiv B$ then $A \xrightarrow{\alpha} B'$.
6) Since $A \approx_\ell C$, we have that if $A \xrightarrow{\alpha} B'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(C) = \emptyset$; then $C \rightarrow^* \xrightarrow{\alpha} \rightarrow^* C'$ and $B' \approx_\ell C'$ for some C'.
7) Since we have $\mathsf{domain}(A) = \mathsf{domain}(B)$ (see first paragraph), we have if $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(A)$ then $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(B)$.

Combining 5) 6) and 7), we have if $B \xrightarrow{\alpha} B'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(B)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(C) = \emptyset$; then $C \rightarrow^* \xrightarrow{\alpha} \rightarrow^* C'$ and $B' \approx_\ell C'$ for some C'.

In a similar way, we can prove that if $C \xrightarrow{\alpha} C'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{domain}(C)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(B) = \emptyset$; then $B \rightarrow^* \xrightarrow{\alpha} \rightarrow^* B'$ and $B' \approx_\ell C'$ for some B'.

From the above, we prove that $B \approx_\ell C$. Simialrly, we can prove that $A \approx_\ell D$. By lemma 1, we have if $A \approx_\ell C$, $B \approx_\ell C$, then $A \approx_\ell B$. Similarly, if $A \approx_\ell B$ and $A \approx_\ell D$ then $B \approx_\ell D$. □

Theorem 3. Let Q be a closed plain process and c'_{out} be a channel name such that $\mathsf{c}'_{out} \notin \mathsf{fn}(Q) \cup \mathsf{bn}(Q)$. Let $\mathcal{C}_h[\_] := \nu \mathsf{c}'_{out}.(\_ \,|!\mathsf{in}(\mathsf{c}'_{out}, x))$. We have $Q^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon \rangle \backslash \mathsf{out}(\mathsf{c}'_{out}, \cdot)}$

$\hat{=} \nu \mathsf{c}'_{out}.(Q^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle} \,|!\mathsf{in}(\mathsf{c}'_{out}, x)) \approx_\ell Q$ [DKR09]

Theorem 4. Let $\mathcal{C}_1[\_] = \nu\tilde{u_1}.(\_ \mid B_1)$ and $\mathcal{C}_2[\_] = \nu\tilde{u_2}.(\_ \mid B_2)$ be two evaluation contexts such that $\tilde{u_1} \cap (\mathsf{fv}(B_2) \cup \mathsf{fv}(B_2)) = \emptyset$ and $\tilde{u_2} \cap (\mathsf{fv}(B_1) \cup \mathsf{fv}(B_1)) = \emptyset$. We have that $\mathcal{C}_1[\mathcal{C}_2[A]] \equiv \mathcal{C}_2[\mathcal{C}_1[A]]$ for any extended process A [DKR09].

Theorem 5. Let $A \mid B$ be a process, $\mathtt{ch}$ be a channel name in A, $\mathtt{ch}$ never appears in B. $(A \mid B)^{\backslash\mathsf{out}(\mathtt{ch},\cdot)} \equiv A^{\backslash\mathsf{out}(\mathtt{ch},\cdot)} \mid B$.

Proof.

$$(A \mid B)^{\backslash\mathsf{out}(\mathtt{ch},\cdot)} := \nu\mathtt{ch}.((A \mid B) \,|!\mathsf{in}(\mathtt{ch}, x))$$

$$A^{\backslash\mathsf{out}(\mathtt{ch},\cdot)} \mid B := (\nu\mathtt{ch}.(A \,|!\mathsf{in}(\mathtt{ch}, x))) \mid B$$

Since $\mathtt{ch}$ never appears in B, we have (rule `NEW-PAR`)

$$(\nu\mathtt{ch}.(A \,|!\mathsf{in}(\mathtt{ch}, x))) \mid B \equiv \nu\mathtt{ch}.((A \,|!\mathsf{in}(\mathtt{ch}, x)) \mid B),$$

Because of rule `PAR-C` and rule `PAR-A`, we have

$$(A \mid B) \,|!\mathsf{in}(\mathtt{ch}, x) \equiv A \,|!\mathsf{in}(\mathtt{ch}, x) \mid B,$$

Thus,

$$\nu\mathtt{ch}.((A \mid B) \,|!\mathsf{in}(\mathtt{ch}, x)) \equiv \nu\mathtt{ch}.((A \,|!\mathsf{in}(\mathtt{ch}, x)) \mid B).$$

By transitivity of structural equivalence, we have

$$(A \mid B)^{\backslash\mathsf{out}(\mathtt{ch},\cdot)} \equiv A^{\backslash\mathsf{out}(\mathtt{ch},\cdot)} \mid B.$$

□

B.2 Theorem 6.20

(3) $\forall\rho$, $\mathsf{epriv}_\rho \Longrightarrow \mathsf{priv}$

We prove the statement in the following two directions: 1. $\forall\rho$, $\mathsf{epriv}_\rho \Longrightarrow \mathsf{priv}$ 2. $\exists\rho$, $\mathsf{priv} \not\Longrightarrow \mathsf{epriv}_\rho$

1. $\forall\rho$, when a protocol satisfies epriv_ρ, we prove that the protocol also satisfies priv.

   For a collaboration $\rho = \langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, when a well-formed protocol P_w satisfies epriv w.r.t. τ and $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eq1:

   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}],$$

   we have
   eq2:

   $$\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\},$$

and
eq3:

$$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i,\, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f]].$$

1) According to Lemma 1 (transitivity of $\approx_\ell$), combining ($\mathsf{eq1}$) and ($\mathsf{eq3}$), we have
eq4:

$$\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f]].$$

2) By applying the evaluation context $\mathcal{C}_h[\_] := \nu \mathsf{c}'_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}'_{out}, x))$ (x is a fresh variable) on both sides of ($\mathsf{eq4}$), we have
eq5:

$$\mathcal{C}_h[\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]] \approx_\ell \mathcal{C}_h[\mathcal{C}_{P_w}[\mathcal{C}[P_f]]].$$

3) According to Lemma 4, by swapping position of context $\mathcal{C}_h[\_]$ and $\mathcal{C}_{P_w}[\_]$, the left side of ($\mathsf{eq5}$) is structural equivalent to

$$\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]],$$

and the right side of ($\mathsf{eq5}$) is structural equivalent to $\mathcal{C}_{P_w}[\mathcal{C}_h[\mathcal{C}[P_f]]]$. According to Lemma 2, the above two processes are bisimilar, that is
eq6:

$$\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}_h[\mathcal{C}[P_f]]].$$

4) By Lemma 3, we have the following equivalence

$$\mathcal{C}_h[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}.$$

By applying the context $\mathcal{C}_{P_w}[\_]$ on both sides of the above equivalence, we have
eq7:

$$\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}].$$

That is, the left side of ($\mathsf{eq6}$) is equivalent to $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}]$.
5) By Lemma 3, we have $\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} := \mathcal{C}_h[\mathcal{C}[P_f]]$. Thus, we can replace the process $\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)}$ in ($\mathsf{eq2}$) with $\mathcal{C}_h[\mathcal{C}[P_f]]$. That is, $\mathcal{C}_h[\mathcal{C}[P_f]] \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}$. By applying context $\mathcal{C}_{P_w}[\_]$ on both sides of the above equivalence, we have
eq8:

$$\mathcal{C}_{P_w}[\mathcal{C}_h[\mathcal{C}[P_f]]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}].$$

That is, the right side of ($\mathsf{eq6}$) is equivalent to $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}]$.
6) According to Lemma 2, combining ($\mathsf{eq6}$), ($\mathsf{eq7}$) and ($\mathsf{eq8}$), we have
eq9:

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}].$$

The equivalence ($\mathsf{eq9}$) coincides with the equivalence in Def. 6.3. Thus, the protocol P_w satisfies priv. □

2. There exists ρ such that $\mathsf{priv} \not\Longrightarrow \mathsf{epriv}_\rho$.

   We prove the statement by showing an example in which a protocol satisfies priv but not epriv_ρ for some ρ as in Ex. B.1.

Example B.1. *Protocol $Q := \nu\mathtt{r}.\nu\mathtt{s}.\mathsf{out}(\mathtt{ch}, \mathsf{enc}(\mathtt{s}, \mathtt{r}))$ where $\mathtt{ch}$ is a public channel, satisfies* priv *w.r.t.* $\mathtt{s}$, *but not* epriv *w.r.t.* $\mathtt{s}$ *and* $\langle\{\mathtt{r}\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle$. *The adversary cannot distinguish* $\mathsf{enc}(\mathtt{s}_1, \mathtt{r})$ *and* $\mathsf{enc}(\mathtt{s}_2, \mathtt{r})$, *thus the protocol satisfies* priv *w.r.t* $\mathtt{s}$. *However, when Q is coerced to reveal* $\mathtt{r}$, *there is no way for Q to cheat the adversary. Because of the perfect encryption assumption, any other nonce cannot be used to decrypted* $\mathsf{enc}(\mathtt{s}, \mathtt{r})$, *thus, the adversary will find out whether the user lied.*

(4) $\forall\theta$, $\mathsf{ipriv}_\theta \implies \mathsf{priv}$

Note that in ipriv_θ, we assume the existence of a set of attacking third parties R_T. Thus, when we consider priv, we have the same assumption that there exists the same set of third parties R_D.

We prove the statement in the following two directions: 1. $\forall\theta$, $\mathsf{ipriv}_\theta \implies \mathsf{priv}$ 2. $\exists\theta$, $\mathsf{priv} \not\Longrightarrow \mathsf{ipriv}_\theta$

1. $\forall\theta = (R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a protocol satisfies ipriv_θ, we prove that the protocol also satisfies priv with the existence of R_T.

   For a collaboration of third parties $\theta = (R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a well-formed protocol P_w satisfies ipriv w.r.t. τ and $(R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, the following equivalence holds.
   eqi1:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\ &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   Similar as in definitions of enforced privacy properties like epriv, we separate the adversary's ability of coercing from distinguishing differences of two processes, and model the ability of providing information for collaborating users as a context. Since for all contexts of the adversary which provides information for the collaborating third parties, the protocol satisfies ipriv_θ, thus, for the following context $\mathcal{C}_t[\_]$, which supplies information needed by the collaborating third parties, the protocol satisfies ipriv_θ.

   $$\mathcal{C}_t[\_] := \nu\mathsf{c}^t_{out}.\nu\mathsf{c}^t_{in}.(\_ \mid Q)$$

   satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}_t[\_]) = \emptyset$ and
   eqi2:
   $$\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell R_T^{\langle\Psi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle},$$

   2) By applying the context $\mathcal{C}_t[\_]$ on both sides of ($\mathtt{eqi1}$), we have
   eqi3:
   $$\begin{aligned}&\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]\\ \approx_\ell\ &\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]].\end{aligned}$$

   3) By applying the evaluation context $\mathcal{C}^t_h[\_] := \nu\mathsf{c}^t_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}^t_{out}, x))$ (x is a fresh variable), on both sides of ($\mathtt{eqi3}$), we have
   eqi4:
   $$\begin{aligned}&\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\ &\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

4) According to Lemma 4, by swapping contexts $\mathcal{C}_h^t[\_]$ and $\mathcal{C}_{P_w}[\_]$, the left side of (eqi4) is structural equivalent to

$$\mathcal{C}_{P_w}[\mathcal{C}_h^t[\mathcal{C}_t[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]$$

That is,
eqi5:

$$\begin{aligned}&\mathcal{C}_h^t[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \equiv\, &\mathcal{C}_{P_w}[\mathcal{C}_h^t[\mathcal{C}_t[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\end{aligned}$$

Since c^t_{out} and c^t_{in} are fresh channel names, they do not appear in $\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}$. According to Lemma 5, we have are able to move the position of the context $\mathcal{C}_h^t[\_]$, thus have
eqi6:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}_h^t[\mathcal{C}_t[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \equiv\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

Thus, combining (eqi5) and (eqi6), we have
eqi7:

$$\begin{aligned}&\mathcal{C}_h^t[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \equiv\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

5) Similarly, the right side of (eqi4) satisfies the following equivalence,
eqi8:

$$\begin{aligned}&\mathcal{C}_h^t[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \equiv\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

6) According to Lemma 2, combining (eqi7), (eqi8) and (eqi4), we have
eqi9:

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

7) By applying the context $\mathcal{C}_h^t[\_]$ on both sides of (eqi2), we obtain

$$\mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]] \approx_\ell \mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}].$$

According to Lemma 3, from the above equivalence, we have

$$\mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}] \approx_\ell R_T.$$

By Lemma 1 (transitivity of the above two equivalences), we have
eqi10:

$$\mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]] \approx_\ell R_T.$$

8) Thus, the left side of (eqi9) satisfies the following equivalence (by applying context $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \_]$ on both sides of (eqi10))

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T].$$

The right side of (eqi9) satisfies the following equivalence (by applying context $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid \_]$ on both sides of (eqi10))

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T].$$

According to Lemma 1 (transitivity), from (eqi9), we have
eqi11:

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T]$$

9) According to the definition of third parties – third parties are third party processes running in parallel. The context $\mathcal{C}_{P_w}[\_]$ has the following form

$$\mathcal{C}_{P_w}[\_] := \nu\tilde{\mathsf{c}}.(genkey \mid !R_1 \mid \ldots \mid !R_p \mid \_).$$

Thus, according to rule

$$!P \equiv P \mid !P,$$

R_T can be absorbed by the context. Thus, $\mathcal{C}_{P_w}[\_ \mid R_T]$ is a type of context where there requires R_T to be present. We define $\mathcal{C}'_{P_w}[\_] := \mathcal{C}_{P_w}[\_ \mid R_T]$, where R_T has to be present in the context, we have
eqi12:

$$\mathcal{C}'_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \mathcal{C}'_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}]$$

Therefore, the protocol satisfies priv w.r.t. τ with the existence of R_T. □

2. There exists θ such that $\mathsf{priv} \not\Longrightarrow \mathsf{ipriv}_\theta$.

   We prove the statement by showing an example in which a protocol satisfies priv but not ipriv_θ for some θ as in Ex. B.2.

   Example B.2. *The following protocol*

$$\begin{aligned} P &:= \nu\mathtt{untapch}.(Q \mid Q') \\ Q &:= \nu\mathtt{s}.\mathsf{out}(\mathtt{untapch}, \mathtt{s}) \\ Q' &:= \mathsf{in}(\mathtt{untapch}, x) \end{aligned}$$

   where $\mathtt{untapch}$ *is an untappable channel, satisfies* priv *w.r.t.* $\mathtt{s}$*, but not* ipriv *w.r.t.* $\mathtt{s}$ *and* $(Q', \langle\{x\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle)$*. Since the communication is untappable, the adversary cannot distinguish* $\mathtt{s}_1$ *from* $\mathtt{s}_2$*, thus the protocol satisfies* priv *w.r.t.* $\mathtt{s}$*. However, when the communication partner* Q' *reveals the secret information he reads in on the untappable channel,* $\mathtt{s}$ *is revealed.*

(2) $\forall\rho$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{ipriv}_\theta$

Similar as proving $\forall\rho$, $\mathsf{epriv}_\rho \implies \mathsf{priv}$, we prove the statement in the following two directions: 1. $\forall\rho$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{ipriv}_\theta$ 2. $\exists\rho, \theta$, $\mathsf{ipriv}_\theta \not\Longrightarrow \mathsf{iepriv}_{\rho,\theta}$

1. $\forall\rho$, when a protocol satisfies $\mathsf{iepriv}_{\rho,\theta}$ for some θ, we prove that the protocol also satisfies ipriv_θ.

   For a collaboration $\rho = \langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, when a well-formed protocol P_w satisfies iepriv w.r.t. τ, $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$ there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqie1:

$$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T],$$

we have
eqie2:

$$\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\},$$

and
eqie3:

$$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \\ \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].$$

We first prove the following statement: If a context which provides information for the collaborating target user $\mathcal{C}'[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q')$ satisfies $\mathsf{bn}(P_w) \cap \mathsf{fn}((\mathcal{C}[\_]) = \emptyset$ and
eqie4:

$$\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \\ \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}],$$

then this context satisfies (`eqie1`) when R_T exists.

Proof. Since (`eqie4`) holds for any context of the adversary which provides information for the collaborating third parties, for a specific context $\mathcal{C}_t[\_]$ of the adversary providing information for the collaborating third parties, (`eqie4`) should hold.

$$\mathcal{C}_t[\_] := \nu\mathsf{c}^t_{out}.\nu\mathsf{c}^t_{in}.(\_ \mid Q)$$

satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}_t[\_]) = \emptyset$ and
eqie41:

$$\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell R_T^{\langle\Psi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle},$$

Since (`eqie4`) holds in context $\mathcal{C}_t[\_]$, we apply context $\mathcal{C}_t[\_]$ and evaluation context $\mathcal{C}^t_h[\_] := \nu\mathsf{c}^t_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}^t_{out}, x))$ (x is a fresh variable) on both sides of (`eqie4`), we have
eqie42:

$$\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]] \\ \approx_\ell \mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]].$$

Similar as proving $\forall\theta$, $\mathsf{ipriv}_\theta \Longrightarrow \mathsf{priv}$, by Lemma 5, we move the position of the contexts $\mathcal{C}_t[\_]$ and $\mathcal{C}^t_h[\_]$, and have
eqie43:

$$\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]] \\ \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]$$

By applying context $\mathcal{C}^t_h[\_]$ on both sides of (`eqie41`) we have
eqie44:

$$\mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]] \approx_\ell \mathcal{C}^t_h[R_T^{\langle\Psi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].$$

According to Lemma 3, we have

$$\mathcal{C}^t_h[R_T^{\langle\Psi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell R_T$$

Thus, by transitivity, combining the above equivalence and (`eqie44`), we have
eqie45:

$$\mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]] \approx_\ell R_T$$

By applying context $\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \_]$ on both sides of (`eqie45`), we have

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T]\end{aligned}$$

By applying context $\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \_]$ on both sides of (`eqie45`), we have

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T]\end{aligned}$$

Because of (`eqie43`), combining the above two equivalences, we have

$$\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T]$$

Thus, the statement is proved. □

1) Since the context $\mathcal{C}'[\_]$ satisfies $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
eqie51: (replacing $\mathcal{C}[\_]$ with $\mathcal{C}'[\_]$ in (`eqie1`))

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T],\end{aligned}$$

for $\mathcal{C}'[\_]$, (`eqie2`) and (`eqie3`) should hold by replacing $\mathcal{C}[\_]$ with $\mathcal{C}'[\_]$.
eqie52:

$$\mathcal{C}'[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\},$$

eqie53:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}'[P_f] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

2) Combining (`eqie4`) and (`eqie53`), we have
eqie6:

$$\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}'[P_f] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].$$

3) By applying evaluation context $\mathcal{C}_h[\_] := \nu \mathsf{c}'_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}'_{out}, x))$ (x is a fresh variable) on both sides of (`eqie6`), we have
eqie7:

$$\begin{aligned}&\mathcal{C}_h[\mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_h[\mathcal{C}_{P_w}[\mathcal{C}'[P_f] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]].\end{aligned}$$

4) By Lemma 4 and Lemma 5, we move the position of context $\mathcal{C}_h[\_]$ and have

eqie8:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i,\mathtt{t}_1/\tau\}] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}_h[\mathcal{C}'[P_f]] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

6) Because of Lemma 3,

$$\mathcal{C}_h[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i,\mathtt{t}_1/\tau\}] \approx_\ell \hat{R}_i\{\mathsf{id}/id_i,\mathtt{t}_1/\tau\},$$

thus we have that the left side of (`eqie8`) is equivalent to

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i,\mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]$$

Because of (`eqie52`), we have

$$\mathcal{C}_h[\mathcal{C}'[P_f]] := \mathcal{C}'[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i,\mathtt{t}_2/\tau\}.$$

Thus, by applying context $\mathcal{C}_{P_w}[\_ \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]$ on both sides of the equivalence, we have that the right side of (`eqie8`) is equivalent to

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i,\mathtt{t}_2/\tau\} \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]$$

By Lemma 1 (transitivity), we have

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i,\mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i,\mathtt{t}_2/\tau\} \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\end{aligned}$$

The above equivalence coincides with the equivalence in ipriv (Def: 6.13). Thus, the protocol satisfies ipriv_θ. □

There exists ρ, θ such that $\mathsf{ipriv}_\theta \not\Longrightarrow \mathsf{iepriv}_{\rho,\theta}$.

We prove the statement by showing an example in which a protocol satisfies ipriv_θ but not $\mathsf{iepriv}_{\rho,\theta}$ for some ρ as in Ex. B.3.

Example B.3. *Protocol*

$$\begin{aligned}P &:= Q \mid Q'\\ Q &:= \nu\mathtt{r}.\nu\mathtt{s}.\mathsf{out}(\mathtt{ch},\mathsf{enc}(\mathtt{s},\mathtt{r}))\\ Q' &:= \mathsf{in}(\mathtt{ch},x)\end{aligned}$$

where `ch` *is a public channel, satisfies* ipriv *w.r.t.* `s` *and* $(Q', \langle\{x\},\emptyset,\mathsf{c}_{out},\epsilon\rangle)$*, but not* iepriv *w.r.t.* `s`, $\langle\{\mathtt{r}\},\emptyset,\mathsf{c}_{out},\epsilon\rangle$ *and* $(Q', \langle\{x\},\emptyset,\mathsf{c}_{out},\epsilon\rangle)$.

When the third party Q' is coerced for the information he read. The revealing of information from Q' does not help increase the adversary's knowledge. Therefore, the adversary still cannot distinguish $\mathsf{enc}(\mathtt{s}_1,\mathtt{r})$ *and* $\mathsf{enc}(\mathtt{s}_2,\mathtt{r})$*, even when Q' reveals information, thus the protocol satisfies* ipriv *w.r.t.* `s` *and* $(Q', \langle\{x\},\emptyset,\mathsf{c}_{out},\epsilon\rangle)$.

However, when Q is also coerced to reveal `r`*, the adversary can decrypt the message* $\mathsf{out}(\mathtt{ch},\mathsf{enc}(\mathtt{s},\mathtt{r}))$ *and find* `s`*. In addition, there is no way for Q to cheat the adversary. Due to the perfect encryption assumption, any other nonce cannot be used to decrypt* $\mathsf{enc}(\mathtt{s},\mathtt{r})$*, thus, the adversary will find out whether the user lied. Therefore, the protocol does not satisfy* iepriv *w.r.t.* `s`, $\langle\{r\},\emptyset,\mathsf{c}_{out},\epsilon\rangle$ *and* $(Q', \langle\{x\},\emptyset,\mathsf{c}_{out},\epsilon\rangle)$.

(1) $\forall\theta$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{epriv}_{\rho}$

We prove the statement in the following two directions: 1. $\forall\theta$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{epriv}_{\rho}$ 2. $\exists\rho,\theta$, $\mathsf{epriv}_{\rho} \;\not\Longrightarrow\; \mathsf{iepriv}_{\rho,\theta}$

1. $\forall\theta = (R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a protocol satisfies $\mathsf{iepriv}_{\rho,\theta}$ for some ρ, we prove that the protocol also satisfies epriv_{ρ} with the existence of R_T.

   For a collaboration of third parties $\theta = (R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a well-formed protocol P_w satisfies iepriv w.r.t. target data τ, $\langle \varPsi, \varPhi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and $(R_T, \langle \varPsi^t, \varPhi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqiee1:

   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi,\varPhi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \varPsi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T],$$

   we have
   eqiee2:

   $$\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\},$$

   eqiee3:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi,\varPhi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   1) Since for any context of the adversary which provides information for the collaborating third parties, the equivalence (`eqiee3`) holds. Thus, for the following context $\mathcal{C}_t[\_]$ of the adversary, the equivalence still holds. $\mathcal{C}_t[\_] := \nu\mathsf{c}^t_{out}.\nu\mathsf{c}^t_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}_t[\_]) = \emptyset$ and
   eqiee4:

   $$\mathcal{C}_t[R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell R_T^{\langle \varPsi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}.$$

   That is, by applying the context $\mathcal{C}_t[\_]$ on both sides of (`eqiee3`), we have,
   eqiee5:

   $$\begin{aligned}&\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi,\varPhi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]\\ \approx_\ell\, &\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]].\end{aligned}$$

   2) By applying the evaluation context $\mathcal{C}^t_h[\_] := \nu\mathsf{c}^t_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}^t_{out}, x))$ (x is a fresh variable), on both sides of (`eqiee5`), we have
   eqiee6:

   $$\begin{aligned}&\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi,\varPhi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

   3) By Lemma 4 and Lemma 5, we move the position of the contexts $\mathcal{C}^t_h[\_]$ and $\mathcal{C}_t[\_]$ in (`eqiee6`) and have
   eqiee7:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \varPsi,\varPhi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle \varPsi^t,\varPhi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

4) By applying context $\mathcal{C}_h^t[\mathcal{C}_{P_w}[\mathcal{C}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \_]]]$ on both sides of (eqiee4), we have
eqiee8:

$$\begin{aligned} &\mathcal{C}_h^t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}]]]\\ \approx_\ell\, &\mathcal{C}_h^t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \Psi^t, \emptyset, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}]]. \end{aligned}$$

5) By Lemma 5, we move the position of context $\mathcal{C}_h^t[\_]$ and have
eqiee9:

$$\begin{aligned} &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}]]. \end{aligned}$$

6) By Lemma 1 (transitivity), combining (eqiee7) and (eqiee9),we have
eqiee10:

$$\begin{aligned} &\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}]]. \end{aligned}$$

7) According to Lemma 3 (hide on channel), we have

$$\mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}] \approx_\ell R_T.$$

8) By Lemma 1 (transitivity), combining the above equivalence and (eqiee4), we have

$$\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}_{out}^t, \mathsf{c}_{in}^t \rangle}] \approx_\ell R_T.$$

9) Thus, by applying context $\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid \_]$ on both sides of the above equivalence, the left side of (eqiee10) is bisimilar to

$$\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T]$$

and by applying context $\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid \_]$ on both sides of the above equivalence, the right side of (eqiee10) is bisimilar to

$$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T].$$

Thus,
eqiee11:

$$\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T].$$

10) Because of rule

$$!P \equiv P \mid !P,$$

R_T can be absorbed by the context. That is, $\mathcal{C}_{P_w}[\_ \mid R_T]$ is a type of context where there requires R_T to be present. We define $\mathcal{C}_{P_w}^{'}[\_] := \mathcal{C}_{P_w}[\_ \mid R_T]$, where R_T has to be present in the context, Thus, we have
eqiee12:

$$\mathcal{C}_{P_w}^{'}[\mathcal{C}[P_f]] \approx_\ell \mathcal{C}_{P_w}^{'}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}]].$$

From (eqiee1), by replacing the context $\mathcal{C}_{P_w}[\_]$ with $\mathcal{C}_{P_w}^{'}[\_]$, we have
eqiee13:

$$\mathcal{C}_{P_w}^{'}[\mathcal{C}[\hat{R}_i^{\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in} \rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}^{'}[\hat{R}_i^{\langle \Psi, \emptyset, \mathsf{c}_{out}', \epsilon \rangle}\{\mathsf{id}/id_i, \mathsf{t_1}/\tau\}],$$

Therefore, for any context $\mathcal{C}[\_]$ satisfying (eqiee13), (eqiee2) and (eqiee12) hold. Thus, the protocol satisfies epriv_ρ. □

2. There exists θ, ρ such that $\mathsf{epriv}_\rho \;\not\Longrightarrow\; \mathsf{iepriv}_{\rho,\theta}$.

   We prove the statement by showing an example in which a protocol satisfies epriv_ρ but not $\mathsf{iepriv}_{\rho,\theta}$ for some θ as in Ex. B.4.

   Example B.4. *The following protocol*

   $$\begin{aligned} P &:= \nu\mathtt{untapch}.(Q \mid Q') \\ Q &:= \nu\mathtt{s}.\mathsf{out}(\mathtt{untapch}, \mathtt{s}) \\ Q' &:= \mathsf{in}(\mathtt{untapch}, x) \end{aligned}$$

   where $\mathtt{untapch}$ *is an untappable channel, satisfies* epriv *w.r.t.* $\mathtt{s}$ *and collaboration* $\langle\{\mathtt{s}\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle$*, but not* iepriv *w.r.t.* $\mathtt{s}$*, collaboration* $\langle\{\mathtt{s}\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle$ *and third party collaboration* $(Q', \langle\{x\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle)$*.*

   Since the communication is untappable, Q can lie about $\mathtt{s}$ *to be* $\mathtt{s}'$*, the adversary cannot detect whether Q lied, thus the protocol satisfies* epriv *w.r.t.* $\mathtt{s}$ *and* $\langle\{\mathtt{s}\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle$*. However, when the communication partner Q' reveals the secret information that he reads in on the untappable channel,* $\mathtt{s}$ *is revealed. Thus, the protocol does not satisfies* iepriv *w.r.t.* $\mathtt{s}$*,* $\langle\{\mathtt{s}\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle$ *and* $(Q', \langle\{x\}, \emptyset, \mathsf{c}_{out}, \epsilon\rangle)$*.*

B.3 Theorem. 6.21

(3) $\forall\rho$, $\mathsf{cepriv}_{\rho,\delta} \Longrightarrow \mathsf{cpriv}_\delta$

With the above assumption, we prove the statement in the following two directions: 1. $\forall\rho$, $\mathsf{cepriv}_{\rho,\delta} \Longrightarrow \mathsf{cpriv}_\delta$ 2. $\exists\rho, \delta$, $\mathsf{cpriv}_\delta \;\not\Longrightarrow\; \mathsf{cepriv}_{\rho,\delta}$

1. $\forall\rho$, when a protocol satisfies $\mathsf{cepriv}_{\rho,\delta}$ for some δ, we prove that the protocol also satisfies cpriv_δ.

   For a collaboration $\rho = \langle\Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, when a well-formed protocol P_w satisfies cepriv w.r.t. τ, $\langle\Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and $(R_D, \langle\Theta, \Delta, \Pi\rangle)$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqc1:

   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}],$$

   we have
   eqc2:

   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}|R_D)^{\langle\Theta,\Delta,\Pi\rangle},$$

   eqc3:

   $$\begin{aligned} &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.((\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma)) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle})]. \end{aligned}$$

   1) By applying context $\mathcal{C}_h[\_]$ on both side of $(\mathsf{eqc3})$, we have
   eqc4:

   $$\begin{aligned} &\mathcal{C}_h[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D]] \\ \approx_\ell\, &\mathcal{C}_h[\mathcal{C}_{P_w}[\nu\Omega.((\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma)) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle})]]. \end{aligned}$$

2) By Lemma 5, we move the position of $\mathcal{C}_h[\_]$, and have
eqc5:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}_h[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}]] \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.((\nu\eta.(\mathcal{C}_h[\mathcal{C}[P_f]] \mid P_\gamma)) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle})].\end{aligned}$$

3) The context $\mathcal{C}_{P_w}[\_]$ has the following form:

$$\mathcal{C}_{P_w}[\_] := \nu\tilde{\mathsf{c}}.(genkey \mid !R_1 \mid \ldots \mid !R_p \mid \_).$$

Because of ($\mathtt{eqc1}$) and rule $!P \equiv P \mid !P$, we have
eqc6:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_D].\end{aligned}$$

4) By applying $\mathcal{C}_h[\_]$ on both side of ($\mathtt{eqc6}$), we have
eqc7:

$$\begin{aligned}&\mathcal{C}_h[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D]]\\ \approx_\ell\, &\mathcal{C}_h[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_D]].\end{aligned}$$

5) By Lemma 5, we move the position of $\mathcal{C}_h[\_]$ and have
eqc8:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}_h[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}]] \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}] \mid R_D].\end{aligned}$$

6) By Lemma 1, combining ($\mathtt{eqc5}$) and ($\mathtt{eqc8}$), we have
eqc9:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}] \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.((\nu\eta.(\mathcal{C}_h[\mathcal{C}[P_f]] \mid P_\gamma)) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle})].\end{aligned}$$

7) By Lemma 3, we have

$$\mathcal{C}_h[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}] \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}$$

Thus, we have (by applying context $\mathcal{C}_{P_w}[\_ \mid R_D]$ on the above equivalence)
eqc10:

$$\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}] \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D].$$

That is, the left side of ($\mathtt{eqc9}$) is equivalent to

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D].$$

8) According to Lemma 3, we have

$$\nu\Omega.((\nu\eta.(\mathcal{C}_h[\mathcal{C}[P_f]] \mid P_\gamma)) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle}) \mathrel{\hat{=}} \nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle})$$

Because of ($\mathtt{eqc2}$), we have
eqc11:

$$\nu\Omega.((\nu\eta.(\mathcal{C}_h[\mathcal{C}[P_f]] \mid P_\gamma)) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle}) \approx_\ell \nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} | R_D)^{\langle \Theta,\Delta,\Pi\rangle}.$$

9) By applying context $\mathcal{C}_{P_w}[\_]$ on both sides of ($\mathtt{eqc11}$), we have
eqc12:

$$\begin{aligned}&\mathcal{C}_{P_w}[\nu\Omega.((\nu\eta.(\mathcal{C}_h[\mathcal{C}[P_f]] \mid P_\gamma)) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle})]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} | R_D)^{\langle \Theta,\Delta,\Pi\rangle}].\end{aligned}$$

That is, the right side of (`eqc9`) is equivalent to

$$\mathcal{C}_{P_w}[\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}|R_D)^{\langle\Theta,\Delta,\Pi\rangle}].$$

10) Combining (`eqc10`) and (`eqc12`), we have

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}|R_D)^{\langle\Theta,\Delta,\Pi\rangle}].$$

Therefore, the protocol satisfies cpriv. □

2. There exists ρ, δ such that $\mathsf{cpriv}_\delta \not\Longrightarrow \mathsf{cepriv}_{\rho,\delta}$.

   We prove the statement by showing an example in which a protocol satisfies cpriv_δ but not $\mathsf{cepriv}_{\rho,\delta}$ for some ρ, δ. As shown in Section 6.5, vote-privacy is an instance of cpriv where the defending third party is the counter-balancing voter, and the coalition is the counter-balancing voter replaces his vote to counter balance to target voter's vote, and receipt-freeness is an instance of cepriv with the same defending third party and coalition. The protocol FOO92 [FOO92] is shown to satisfy vote-privacy but not receipt-freeness [DKR09].

(4) $\forall\theta$, $\mathsf{cipriv}_{\theta,\delta} \Longrightarrow \mathsf{cpriv}_\delta$

We prove the statement in the following two directions: 1. $\forall\theta$, $\mathsf{cipriv}_{\theta,\delta} \Longrightarrow \mathsf{cpriv}_\delta$ 2. $\exists\theta,\delta$, $\mathsf{cpriv}_\delta \not\Longrightarrow \mathsf{cipriv}_{\theta,\delta}$

1. $\forall\theta = (R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a protocol satisfies $\mathsf{ipriv}_{\theta,\delta}$ for some δ, we prove that the protocol also satisfies cpriv_δ with the existence of R_T.

   For a collaboration of third parties $\theta = (R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a well-formed protocol P_w satisfies cipriv w.r.t. τ, $(R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$ and $(R_D, \langle\Theta, \Delta, \Pi\rangle)$ the following equivalence holds.
   eqci1:

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\end{aligned}$$

   Since for all context of the adversary which supplies information needed by the collaborating third parties the protocol satisfies $\mathsf{cipriv}_{\theta,\delta}$, thus, for the following context which provides information for collaborating third parties, $\mathcal{C}_t[\_] := \nu\mathsf{c}^t_{out}.\nu\mathsf{c}^t_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}_t[\_]) = \emptyset$ and
   eqci2:

$$\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell R_T^{\langle\Psi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle},$$

   the protocol satisfies $\mathsf{cipriv}_{\theta,\delta}$.
   1) By applying context $\mathcal{C}_t[\_]$ on both sides of (`eqci1`), we have
   eqci3:

$$\begin{aligned}&\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]\\ \approx_\ell\, &\mathcal{C}_t[\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]].\end{aligned}$$

2) By applying the evaluation context $\mathcal{C}_h^t[\_] := \nu \mathsf{c}^t_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}^t_{out}, x))$ (x is a fresh variable), on both sides of (eqci3), we have
eqci4:

$$\begin{aligned}&\mathcal{C}_h^t[\mathcal{C}_t[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_h^t[\mathcal{C}_t[\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle}) \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

3) According to Lemma 4 and Lemma 5, we move the position of contexts $\mathcal{C}_h^t[\_]$ and $\mathcal{C}_t[\_]$ and have
eqci5:

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle}) \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

4) By applying context $\mathcal{C}_h^t[\_]$ on both sides of (eqci2), we have

$$\mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]] \approx_\ell \mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]$$

Because of Lemma 3, we have

$$\mathcal{C}_h^t[R_T^{\langle \Psi^t, \emptyset, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}] \approx_\ell R_T$$

Thus, by transitivity, combining the above two equivalences, we have

$$\mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]] \approx_\ell R_T.$$

Thus, by applying context $\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid \_]$ and context $\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle}) \mid \_]$ on both sides of the above equivalence, because of transitivity via (eqci5), we have
eqci6:

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle}) \mid R_T].\end{aligned}$$

5) Since R_T can be absorbed by the context $\mathcal{C}_{P_w}[\_]$, we have
eqci7:

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle})].$$

Thus, the protocol satisfies cpriv. □

2. There exists θ such that $\mathsf{cpriv}_\delta \not\Longrightarrow \mathsf{cipriv}_{\theta,\delta}$.

   We prove the statement by showing an example in which a protocol satisfies cpriv_δ for some δ but not $\mathsf{epriv}_{\theta,\delta}$ for some θ. For instance, Dreier et al. proved that the protocol by Lee et al. [LBD+03] satisfies vote-privacy – an instance of cpriv where coalition is the counter-balancing voter votes differently from the target voter, but not vote-independence – an instance of cipriv where the coalition is the same as in cpriv and the attacking third party is the third voter [DLL11].

(2) $\forall\rho$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \Longrightarrow \mathsf{cipriv}_{\theta,\delta}$

We prove the statement in the following two directions: 1. $\forall\rho$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \Longrightarrow \mathsf{cipriv}_{\theta,\delta}$ 2. $\exists \rho,\theta,\delta$, $\mathsf{cipriv}_{\theta,\delta} \not\Longrightarrow \mathsf{ciepriv}_{\rho,\theta,\delta}$

1. $\forall\rho$, when a protocol satisfies $\mathsf{ciepriv}_{\rho,\theta,\delta}$ for some θ,δ, we prove that the protocol also satisfies $\mathsf{cipriv}_{\theta,\delta}$.

   For a collaboration $\rho = \langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, when a well-formed protocol P_w satisfies $\mathsf{ciepriv}$ w.r.t. τ, $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$ and R_D, $\langle \Theta, \Delta, \Pi\rangle$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqiei1:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_T \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_T \mid R_D],\end{aligned}$$
   we have
   eqiei2:
   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_2/\tau\}|R_D)^{\langle\Theta,\Delta,\Pi\rangle},$$
   eqiei3:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   1) Similar as in proving $\forall\rho$, $\mathsf{iepriv}_{\rho,\theta} \implies \mathsf{ipriv}_\theta$, we can prove that if a context $\mathcal{C}'[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q')$ satisfies $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqie4:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_D \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}],\end{aligned}$$
   then this context satisfies the following equivalence (replacing $\mathcal{C}[\_]$ with $\mathcal{C}'[\_]$ in (`eqiei1`)) when R_T exists.
   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_T \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_T \mid R_D].\end{aligned}$$

   2) Thus, for $\mathcal{C}'[\_]$, the following equivalence holds (replacing $\mathcal{C}[\_]$ with $\mathcal{C}'[\_]$ in (`eqiei2`) and (`eqiei3`)).
   eqiei5:
   $$\nu\Omega.(\nu\eta.(\mathcal{C}'[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_2/\tau\}|R_D)^{\langle\Theta,\Delta,\Pi\rangle},$$
   eqiei6:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}'[\hat{R}_i\{\mathsf{id}/id_i, t/\tau\}^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}] \mid R_D \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}'[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\end{aligned}$$

   3) Combining (`eqiei4`) and (`eqiei6`), we have
   eqiei7:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}'[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\}^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle} \mid R_D \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\end{aligned}$$

4) By applying evaluation context $\mathcal{C}_h[\_] := \nu \mathsf{c}'_{out}.(\_ \mid !\mathsf{in}(\mathsf{c}'_{out}, x))$ (x is a fresh variable) on both sides of (`eqiei7`), we have
eqiei8:

$$\begin{aligned}&\mathcal{C}_h[\mathcal{C}_{P_w}[\nu \Omega.(\nu \eta.(\mathcal{C}'[P_f] \mid P_\gamma) \mid R_D^{\langle \Theta, \Delta, \Pi \rangle}) \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]] \\ \approx_\ell\, &\mathcal{C}_h[\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon \rangle} \mid R_D \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]]\end{aligned}$$

5) By Lemma 4 and Lemma 5, we move the position of context $\mathcal{C}_h[\_]$ and have
eqiei9:

$$\begin{aligned}&\mathcal{C}_{P_w}[\nu \Omega.(\nu \eta.(\mathcal{C}_h[\mathcal{C}'[P_f]] \mid P_\gamma) \mid R_D^{\langle \Theta, \Delta, \Pi \rangle}) \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}_h[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon \rangle}] \mid R_D \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]]\end{aligned}$$

6) Because of Lemma 3, we have

$$\mathcal{C}_h[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}^{\langle \Psi, \emptyset, \mathsf{c}'_{out}, \epsilon \rangle}] \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\},$$

thus, we have that the right side of (`eqiei9`) is equivalent to

$$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}].$$

7) By applying context $\mathcal{C}_{P_w}[\_ \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]$ on both sides of (`eqiei2`), we have

$$\begin{aligned}&\mathcal{C}_{P_w}[\nu \Omega.(\nu \eta.(\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out}, \cdot)} \mid P_\gamma) \mid R_D^{\langle \Theta, \Delta, \Pi \rangle}) \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu \Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} | R_D)^{\langle \Theta, \Delta, \Pi \rangle} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]\end{aligned}$$

That is, the left side of (`eqiei9`) is equivalent to

$$\mathcal{C}_{P_w}[\nu \Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} | R_D)^{\langle \Theta, \Delta, \Pi \rangle} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}].$$

Therefore, by transitivity, we have

$$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu \Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} | R_D)^{\langle \Theta, \Delta, \Pi \rangle} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in} \rangle}]\end{aligned}$$

Therefore, the protocol satisfies $\mathsf{cipriv}_{\theta,\delta}$. □

2. There exists ρ, θ, δ such that $\mathsf{cipriv}_{\theta,\delta} \not\Longrightarrow \mathsf{ciepriv}_{\rho,\theta,\delta}$.

   We prove the statement by showing an example in which a protocol satisfies $\mathsf{cipriv}_{\theta,\delta}$ but not $\mathsf{ciepriv}_{\rho,\theta,\delta}$ for some ρ, θ, δ. For instance, Dreier et al. prove that the voting protocol FOO92 [FOO92] satisfies vote-independence – an instance of cipriv where the coalition is the counter-balancing voter votes differently from the target voter and the attacking third party is the third voter, but not vote-independence with passive collaboration – an instance of ciepriv where the coalition and attacking third party are the same as in cipriv and the collaboration is forwarding private information to the adversary [DLL11].

(1) $\forall\theta$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \implies \mathsf{cepriv}_{\rho,\delta}$

We prove the statement in the following two directions: 1. $\forall\theta$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \implies \mathsf{cepriv}_{\rho,\delta}$ 2. $\exists\rho,\theta,\delta$, $\mathsf{cepriv}_{\rho,\delta} \;\not\Longrightarrow\; \mathsf{ciepriv}_{\rho,\theta,\delta}$

1. $\forall\theta = (R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a protocol satisfies $\mathsf{ciepriv}_{\rho,\theta,\delta}$ for some ρ,δ, we prove that the protocol also satisfies $\mathsf{cepriv}_{\rho,\delta}$ with the existence of R_T.

   For a collaboration of third parties $\theta = (R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a well-formed protocol P_w satisfies $\mathsf{ciepriv}$ w.r.t. τ, $\langle\Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, $(R_D, \langle\Theta, \Delta, \Pi\rangle)$ and $(R_T, \langle\Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqciee1:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathsf{t}_1/\tau\} \mid R_T \mid R_D],\end{aligned}$$

   we have
   eqciee2:

   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathsf{t}_2/\tau\}|R_D)^{\langle\Theta,\Delta,\Pi\rangle},$$

   eqciee3:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   1) Since for any context of the adversary which provides information for the collaborating third parties, the equivalence (`eqciee3`) holds. Thus, for the following context $\mathcal{C}_t[\_]$ of the adversary which provides information for the collaborating third parties, the equivalence (`eqciee3`) still holds. $\mathcal{C}_t[\_] := \nu\mathsf{c}^t_{out}.\nu\mathsf{c}^t_{in}.(\_\mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}_t[\_]) = \emptyset$ and
   eqciee4:
   $$\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}] \approx_\ell R_T^{\langle\Psi^t,\emptyset,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}.$$
   Therefore, by applying the context $\mathcal{C}_t[\_]$ on both sides of (`eqciee3`), we have,
   eqciee5:

   $$\begin{aligned}&\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]\\ \approx_\ell\, &\mathcal{C}_t[\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]\end{aligned}$$

   2) By applying the evaluation context $\mathcal{C}^t_h[\_] := \nu\mathsf{c}^t_{out}.(\_ \mid \mathsf{in}(\mathsf{c}^t_{out}, x))$ (x is a fresh variable), on both sides of (`eqciee5`), we have
   eqciee6:

   $$\begin{aligned}&\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}^t_h[\mathcal{C}_t[\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\end{aligned}$$

   3) By Lemma 4 and Lemma 5, we move the position of context $\mathcal{C}^t_h[\_]$ and $\mathcal{C}_t[\_]$ in (`eqciee6`), and have
   eqciee7:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid \mathcal{C}^t_h[\mathcal{C}_t[R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]]].\end{aligned}$$

4) By applying context $\mathcal{C}_h^t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid \_]]$ on both sides of (`eqciee4`), we have
eqciee8:

$$\begin{aligned}&\mathcal{C}_h^t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid \mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]]] \\ \approx_\ell\, &\mathcal{C}_h^t[\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T^{\langle \Psi^t,\emptyset,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]].\end{aligned}$$

5) By Lemma 5, we move the position of context $\mathcal{C}_h^t[\_]$ in (`eqciee8`) and have
eqciee9:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]]] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid \mathcal{C}_h^t[R_T^{\langle \Psi^t,\emptyset,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]].\end{aligned}$$

6) By Lemma 1, combining (`eqciee7`) and (`eqciee9`),we have
eqciee10:

$$\begin{aligned}&\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle}) \mid \mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]]] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid \mathcal{C}_h^t[R_T^{\langle \Psi^t,\emptyset,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]].\end{aligned}$$

7) According to Lemma 3, we have

$$\mathcal{C}_h^t[R_T^{\langle \Psi^t,\emptyset,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}] \approx_\ell R_T.$$

8) By Lemma 1, combining the above equivalence and (`eqiee4`), we have

$$\mathcal{C}_h^t[\mathcal{C}_t[R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}_{out}^t,\mathsf{c}_{in}^t\rangle}]] \approx_\ell R_T.$$

9) Thus, the left side of (`eqciee10`) is bisimilar to

$$\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle}) \mid R_T]$$

and the right side of (`eqiee10`) is bisimilar to

$$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T].$$

Thus,
eqciee11:

$$\begin{aligned}&\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle}) \mid R_T] \\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D \mid R_T].\end{aligned}$$

10) Because of rule

$$!P \equiv P \mid !P,$$

R_T can be absorbed by the context. $\mathcal{C}_{P_w}[\_ \mid R_T]$ is a type of context where there requires R_T to be present. We define $\mathcal{C}'_{P_w}[\_] := \mathcal{C}_{P_w}[\_ \mid R_T]$, where R_T has to be present in the context, Thus, we have
eqciee12:

$$\begin{aligned}&\mathcal{C}'_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle \Theta,\Delta,\Pi\rangle})] \\ \approx_\ell\, &\mathcal{C}'_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D].\end{aligned}$$

From (`eqciee1`), we can obtain
eqciee13:

$$\mathcal{C}'_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D] \approx_\ell \mathcal{C}'_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t_1}/\tau\} \mid R_D],$$

Therefore, for any context $\mathcal{C}[\_]$ satisfying (`eqciee13`), (`eqciee2`) and (`eqciee12`) hold. Thus, the protocol satisfies cepriv_ρ. □

2. There exists θ, ρ, δ such that $\mathsf{cepriv}_{\rho,\delta} \not\Longrightarrow \mathsf{ciepriv}_{\rho,\theta,\delta}$.

   We prove the statement by showing an example in which a protocol satisfies $\mathsf{cepriv}_{\rho,\delta}$ but not $\mathsf{ciepriv}_{\rho,\theta,\delta}$ for some ρ, θ, δ. For instance, Dreier et al. proved that the voting protocol by Lee et al. [LBD$^+$03] satisfies receipt-freeness – an instance of cepriv where the coalition is is the counter-balancing voter votes differently from the target voter and the collaboration is forwarding private information to the adversary, but not vote-independence with passive collaboration – an instance of ciepriv where the coalition and collaboration are the same as in cepriv and the defending third party is the third voter [DLL11].

B.4 Theorem. 6.22

(4) $\mathsf{priv} \Longrightarrow \exists\delta$, cpriv_δ

We prove the statement in the following two directions: 1. $\mathsf{priv} \Longrightarrow \exists\delta$, cpriv_δ 2. $\exists\delta$, $\mathsf{cpriv}_\delta \not\Longrightarrow \mathsf{priv}$

1. When a protocol satisfies priv, then there exists a coalition δ such that the protocol satisfies cpriv_δ.

   When a well-formed protocol P_w satisfies priv w.r.t. τ we have
   eqcc1:
   $$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\}].$$
   The context $\mathcal{C}_{P_w}[\_]$ has the following form
   $$\mathcal{C}_{P_w}[\_] := \nu\tilde{\mathsf{c}}.(genkey \mid !R_1 \mid \cdots \mid !R_p \mid \_).$$
   Because of rule
   $$!P \equiv P \mid !P,$$
   we have (for a set of defending third parties R_D)
   eqcc2:
   $$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D].$$
   Let $\delta = (R_D, \langle\emptyset, \emptyset, \emptyset\rangle)$ be a coalition,
   eqcc3:
   $$\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle} \hat{=} \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$$
   Thus,
   eqcc4:
   $$\mathcal{C}_{P_w}[\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle}] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D]$$
   Because of (`eqcc2`), we have
   eqcc5:
   $$\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle}]$$
   Thus, the protocol satisfies cpriv_δ. □

2. There exists δ such that $\mathsf{cpriv}_\delta \;\not\Longrightarrow\; \mathsf{priv}$.

   We prove the statement by showing an example in which a protocol satisfies cpriv_δ but not priv. For instance, FOO92 [FOO92] is shown that it does not satisfy priv w.r.t. vote *vote*, but satisfies vote-privacy – an instance of cpriv where the coalition is the counter-balancing votes differently from the target voter [KR05].

(3) $\mathsf{ipriv}_\theta \Longrightarrow \exists\delta$, $\mathsf{cipriv}_{\theta,\delta}$

We prove the statement in the following two directions: 1. $\mathsf{ipriv}_\theta \Longrightarrow \exists\delta$, $\mathsf{cipriv}_{\theta,\delta}$ 2. $\exists\theta,\delta$, $\mathsf{cipriv}_{\theta,\delta} \;\not\Longrightarrow\; \mathsf{ipriv}_\theta$

1. When a protocol satisfies ipriv_θ for some θ, then there exists a coalition δ such that the protocol satisfies $\mathsf{cipriv}_{\theta,\delta}$.

   For a collaboration of third parties $\theta = (R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a well-formed protocol P_w satisfies epriv w.r.t. τ and $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, the following equivalence holds.
   eqcci1:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   Thus, we have (for a set of defending third parties R_D)
   eqcci2:
   $$\begin{aligned}&\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle} \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle} \mid R_D].\end{aligned}$$
   Let $\delta = (R_D, \langle \emptyset, \emptyset, \emptyset\rangle)$ be a coalition, then
   eqcci3:
   $$\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle}) \,\hat{=}\, \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$$

   Thus, we have
   $$\begin{aligned}&\mathcal{C}_{P_w}[\nu\Omega.((\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle \Theta, \Delta, \Pi\rangle}) \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D \mid R_T^{\langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   Therefore, the protocol satisfies $\mathsf{cepriv}_{\theta,\delta}$. □

2. There exists θ, ρ such that $\mathsf{cipriv}_{\theta,\delta} \;\not\Longrightarrow\; \mathsf{ipriv}_\theta$.

   We prove the statement by showing an example in which a protocol satisfies $\mathsf{cipriv}_{\theta,\delta}$ but not ipriv_θ. For instance, voting protocols FOO92 are shown does not satisfies priv w.r.t. vote *vote* [KR05], thus does not satisfy ipriv, but satisfies vote-independence – an instance of cipriv where the coalition is the counter-balancing voter votes differently from the target voter and the attacking third party is the third voter [DLL11].

(2) $\mathsf{epriv}_\rho \Longrightarrow \exists\delta$,$\mathsf{cepriv}_{\rho,\delta}$

We prove the statement in the following two directions: 1. $\mathsf{epriv}_\rho \Longrightarrow \exists\delta$,$\mathsf{cepriv}_{\rho,\delta}$ 2. $\exists\rho,\delta$, $\mathsf{cepriv}_{\rho,\delta} \;\not\Longrightarrow\; \mathsf{epriv}_\rho$

1. When a protocol satisfies epriv_ρ for some ρ, then there exists a coalition δ such that the protocol satisfies $\mathsf{cepriv}_{\rho,\delta}$.

   When a well-formed protocol P_w satisfies epriv w.r.t. τ and ρ where $\rho = \langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu \mathsf{c}_{out}.\nu \mathsf{c}_{in}.(\_ \mid Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqcce1:
   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\}],$$
   we have
   eqcce2:
   $$\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\},$$
   eqcce3:
   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}]] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f]].$$

   From ($\mathtt{eqcce3}$), we have
   eqcce4:
   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_D].$$
   Let $\delta = (R_D, \langle\emptyset,\emptyset,\emptyset\rangle)$ be a coalition, then
   eqcce5:
   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mathrel{\hat{=}} \mathcal{C}[P_f] \mid R_D$$
   By applying context $\mathcal{C}_{P_w}[\_]$ on both sides of ($\mathtt{eqcce5}$) we have
   eqcce6:
   $$\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle})] \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_D]$$
   Combining ($\mathtt{eqcce4}$) and ($\mathtt{eqcce6}$), by Lemma 1, we have
   eqcce7:
   $$\begin{gathered}\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle})] \\ \approx_\ell \mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D]\end{gathered}$$
   Since $\delta = (R_D, \langle\emptyset,\emptyset,\emptyset\rangle)$, we have
   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mathrel{\hat{=}} \mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid R_D$$
   Because of ($\mathtt{eqcce2}$), we have
   eqcce8:
   $$\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid R_D \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D.$$
   Since $\delta = (R_D, \langle\emptyset,\emptyset,\emptyset\rangle)$, we have
   $$\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle} := \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$$

   Thus,
   eqcce9:
   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle}$$

   Because of ($\mathtt{eqcce1}$), we have
   eqcce10:
   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_D] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_D],$$

   Therefore, for any context $\mathcal{C}[\_]$ satisfying ($\mathtt{eqcce10}$), the protocol satisfies ($\mathtt{eqcce7}$) and ($\mathtt{eqcce9}$), thus, the protocol satisfies $\mathsf{cepriv}_{\rho,\delta}$. □

2. There exists δ, ρ such that $\mathsf{cepriv}_{\rho,\delta} \centernot\implies \mathsf{epriv}_{\rho}$.

   We prove the statement by showing an example in which a protocol satisfies $\mathsf{cepriv}_{\rho,\delta}$ but not epriv_{ρ}. For instance, voting protocol by Okamoto [Oka96] does not satisfy priv w.r.t. vote *vote* [KR05] in the case of unanimous result, thus does not satisfy epriv where ρ is forwarding private information to the adversary, but satisfies receipt-freeness – an instance of cepriv where the coalition is the counter-balancing votes differently from the target voter and the collaboration is forwarding private information to the adversary [DKR09].

(1) $\mathsf{iepriv}_{\rho,\theta} \implies \exists\delta$, $\mathsf{ciepriv}_{\rho,\theta,\delta}$

We prove the statement in the following two directions: 1. $\mathsf{iepriv}_{\rho,\theta} \implies \exists\delta$, $\mathsf{ciepriv}_{\rho,\theta,\delta}$ 2. $\exists\rho,\theta,\delta$, $\mathsf{ciepriv}_{\rho,\theta,\delta} \centernot\implies \mathsf{iepriv}_{\rho,\theta}$

1. When a protocol satisfies $\mathsf{iepriv}_{\rho,\theta}$ for some ρ, θ, then there exists a coalition δ such that the protocol satisfies $\mathsf{ciepriv}_{\rho,\theta,\delta}$.

   For a collaboration of the target user $\rho = \langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and a collaboration of third parties $\theta = (R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, when a well-formed protocol P_w satisfies iepriv w.r.t. τ, $\langle \Psi, \Phi, \mathsf{c}_{out}, \mathsf{c}_{in}\rangle$ and $(R_T, \langle \Psi^t, \Phi^t, \mathsf{c}^t_{out}, \mathsf{c}^t_{in}\rangle)$, there exists a closed plain process P_f, such that for any context $\mathcal{C}[\_] := \nu\mathsf{c}_{out}.\nu\mathsf{c}_{in}.(\_|Q)$ satisfying $\mathsf{bn}(P_w) \cap \mathsf{fn}(\mathcal{C}[\_]) = \emptyset$ and
   eqccee1:

   $$\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T] \approx_\ell \mathcal{C}_{P_w}[\hat{R}_i^{\langle \Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T],$$

   we have
   eqccee2:

   $$\mathcal{C}[P_f]^{\backslash \mathsf{out}(\mathsf{c}'_{out},\cdot)} \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\},$$

   eqccee3:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}].\end{aligned}$$

   Because of (eqccee3), we have
   eqccee4:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle \Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle} \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle} \mid R_D].\end{aligned}$$

   Let $\delta = (R_D, \langle\emptyset,\emptyset,\emptyset\rangle)$ be a coalition, then
   eqccee5:

   $$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \hat{=} \mathcal{C}[P_f] \mid R_D$$

   By applying context $\mathcal{C}_{P_w}[\_ \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]$ on both sides of (eqccee5), we have
   eqccee6:

   $$\begin{aligned}&\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\mathcal{C}[P_f] \mid R_D \mid R_T^{\langle \Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\end{aligned}$$

By Lemma 1, combining (`eqccee4`) and (`eqccee6`), we have
eqccee7:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle} \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\nu\Omega.(\nu\eta.(\mathcal{C}[P_f] \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \mid R_T^{\langle\Psi^t,\Phi^t,\mathsf{c}^t_{out},\mathsf{c}^t_{in}\rangle}]\end{aligned}$$

Since $\delta = (R_D, \langle\emptyset,\emptyset,\emptyset\rangle)$, we have

$$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \hat{=} \mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid R_D.$$

Because of (`eqccee2`), we have
eqccee8:

$$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$$

Since $\delta = (R_D, \langle\emptyset,\emptyset,\emptyset\rangle)$, we also have

$$\nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle} \hat{=} \hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D$$

Thus, we have
eqccee9:

$$\nu\Omega.(\nu\eta.(\mathcal{C}[P_f]^{\backslash\mathsf{out}(\mathsf{c}'_{out},\cdot)} \mid P_\gamma) \mid R_D^{\langle\Theta,\Delta,\Pi\rangle}) \approx_\ell \nu\Omega.(\hat{R}_i\{\mathsf{id}/id_i, \mathtt{t}_2/\tau\} \mid R_D)^{\langle\Theta,\Delta,\Pi\rangle}$$

From (`eqccee1`), we have
eqccee10:

$$\begin{aligned}&\mathcal{C}_{P_w}[\mathcal{C}[\hat{R}_i^{\langle\Psi,\Phi,\mathsf{c}_{out},\mathsf{c}_{in}\rangle}\{\mathsf{id}/id_i, t/\tau\}] \mid R_T \mid R_D]\\ \approx_\ell\, &\mathcal{C}_{P_w}[\hat{R}_i^{\langle\Psi,\emptyset,\mathsf{c}'_{out},\epsilon\rangle}\{\mathsf{id}/id_i, \mathtt{t}_1/\tau\} \mid R_T \mid R_D]\end{aligned}$$

Therefore, for any context $\mathcal{C}[\_]$ satisfying (`eqccee10`), (`eqccee7`) and (`eqccee9`) are satisfies. Thus, the protocol satisfies $\mathsf{ciepriv}_{\rho,\theta,\delta}$. □

2. There exists θ, ρ, δ such that $\mathsf{ciepriv}_{\rho,\theta,\delta} \not\Longrightarrow \mathsf{iepriv}_{\rho,\theta}$.

   We prove the statement by showing an example in which a protocol satisfies $\mathsf{ciepriv}_{\rho,\theta,\delta}$ but not $\mathsf{iepriv}_{\rho,\theta}$. For instance, voting protocol by Okamoto [Oka96] does not satisfy priv w.r.t. vote *vote* when all votes are unanimous. Thus, the protocol does not satisfy iepriv w.r.t. vote *vote*, ρ and θ, where ρ is the target voter forwarding information to the adversary, θ is the collaborating third voter communicating with the adversary. However, the protocol satisfies vote-independence with passive collaboration – an instance of ciepriv w.r.t. vote *vote*, ρ, θ and δ where ρ and θ are the same as in ieprivand δ is the counter-balancing voter voting differently from the target voter [DLL11].

Publications

[FAST10] N. Dong, H. L. Jonker, and J. Pang, *Analysis of a receipt-free auction protocol in the applied pi calculus*, Proc. 7th Workshop Formal Aspects in Security and Trust – FAST'10, LNCS, vol. 6561, Springer, 2011, pp. 223–238.

[FHIES11] N. Dong, H. L. Jonker, and J. Pang, *Challenges in ehealth: From enabling to enforcing privacy*, Proc. 1st Symposium on Foundations of Health Informatics Engineering and Systems– FHIES'11, LNCS, vol. 7151, Springer, 2011, pp. 195–206.

[ESORICS12] N. Dong, H. L. Jonker, and J. Pang, *Formal analysis of privacy in an ehealth protocol*, Proc. 17th European Symposium on Research in Computer Security – ESORICS'12, LNCS, vol. 7459, Springer, 2012, pp. 325–342.

[ESORICS13] N. Dong, H. L. Jonker, and J. Pang, *Enforcing privacy in the presence of others: Notions, formalisations and relations*, Proc. 18th European Symposium on Research in Computer Security – ESORICS'13, LNCS, vol. 8134, Springer, 2013, pp. 499–516.

Bibliography

[AB05] M. Abadi and B. Blanchet. Computer-Assisted Verification of a Protocol for Certified Email. *Science of Computer Programming*, 58(1–2):3–27, 2005. Special issue SAS'03.

[ABF07] M. Abadi, B. Blanchet, and C. Fournet. Just fast keying in the pi calculus. *ACM Transactions on Information and System Security*, 10(3):1–59, 2007.

[ACdD03] G. Ateniese, R. Curtmola, B. de Medeiros, and D. Davis. Medical information privacy assurance: cryptographic and system aspects. In *Proc. 3rd Conference on Security in Communication Networks*, volume 2576 of *LNCS*, pages 199–218. Springer, 2003.

[ACRR10] M. Arapinis, T. Chothia, E. Ritter, and M. D. Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *Proc. 23rd IEEE Computer Security Foundations Symposium*, pages 107–121. IEEE CS, 2010.

[Ad02] G. Ateniese and B. de Medeiros. Anonymous e-prescriptions. In *Proc. ACM Workshop on Privacy in the Electronic Society*, pages 19–31. ACM Press, 2002.

[AF01] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proc. 28th Symposium on Principles of Programming Languages*, pages 104–115. ACM Press, 2001.

[AG97] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *ACM Conference on Computer and Communications Security*, pages 36–47, 1997.

[And96] R. Anderson. A security policy model for clinical information systems. In *Proc. 17th IEEE Symposium on Security and Privacy*, pages 30–43. IEEE CS, 1996.

[AS02] M. Abe and K. Suzuki. Receipt-free sealed-bid auction. In *Proc. 5th Conference on Information Security*, volume 2433 of *LNCS*, pages 191–199. Springer, 2002.

[Avo05] G. Avoine. Adversarial model for radio frequency identification. *IACR Cryptology ePrint Archive*, 2005:49, 2005.

[BAF08] B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.

[BB96] Joachim Biskup and Gerrit Bleumer. Cryptographic protection of health information: cost and benefit. *International Journal of Bio-Medical Computing*, 43(1):61–67, 1996.

[BC08] B. Blanchet and A. Chaudhuri. Automated formal analysis of a protocol for secure file sharing on untrusted storage. In *Proc. IEEE Symposium on Security and Privacy*, pages 417–431. IEEE, 2008.

[BHM08] M. Backes, C. Hriţcu, and M. Maffei. Automated verification of remote electronic voting protocols in the applied pi-calculus. In *Proc. 21st IEEE Computer Security Foundations Symposium*, pages 195–209. IEEE CS, 2008.

[Bla01] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proc. 14th IEEE Computer Security Foundations Workshop*, pages 82–96. IEEE CS, 2001.

[Bla02] B. Blanchet. From secrecy to authenticity in security protocols. In *Proc. 9th International Symposium on Static Analysis*, volume 2477 of *LNCS*, pages 342–359. Springer, 2002.

[Bla04] B. Blanchet. Automatic proof of strong secrecy for security protocols. In *Proc. 25th IEEE Symposium on Security and Privacy*, pages 86–100. IEEE CS, 2004.

[Bla09] B. Blanchet. Automatic verification of correspondences for security protocols. *Journal of Computer Security*, 17(4):363–434, 2009.

[BMP11] M. Backes, M. Maffei, and K. Pecina. A security API for distributed social networks, 2011.

[BMU08] M. Backes, M. Maffei, and D. Unruh. Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol. In *Proc. IEEE Symposium on Security and Privacy*, pages 202–215. IEEE CS, 2008.

[BMW03] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.

[BP11] J. Bohli and A. Pashalidis. Relations among privacy notions. *ACM Transactions on Information and System Security*, 14(1):4:1–4:24, 2011.

[BPS00] O. Berthold, A. Pfitzmann, and R. Standtke. The disadvantages of free mix routes and how to overcome them. In *Proc. Workshop on Design Issues in Anonymity and Unobservability*, pages 30–45, 2000.

[Bra00] S. A. Brands. *Rethinking public key infrastructures and digital certificates: building in privacy.* MIT Press, 2000.

[BT94] J. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proc. 26th ACM Symposium on Theory of Computing – SOTC'94*, pages 544–553. ACM Press, 1994.

[Cer01] I. Cervesato. The dolev-yao intruder is the most powerful attacker. In *Proceedings of the Sixteenth Annual Symposium on Logic in Computer Science — LICS'01*, pages 16–19. IEEE Computer Society Press. Short, 2001.

[Cha85] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[Cha88] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[CHCK07] D. K. W. Chiu, P. C. K. Hung, V. S. Y. Cheng, and E. Kafeza. Protecting the exchange of medical images in healthcare process integration with web services. In *Proc. 40th Hawaii Conference on Systems Science*, pages 131–140. IEEE CS, 2007.

[CKS04] R. Chadha, S. Kremer, and A. Scedrov. Formal analysis of multi-party contract signing. In *Proc. 17th IEEE Computer Security Foundations Workshop*, pages 266–279. IEEE CS, 2004.

[CL01] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.

[CLK03] X. Chen, B. Lee, and K. Kim. Receipt-free electronic auction schemes using homomorphic encryption. In *Proc. 6th Conference on Information Security and Cryptology*, volume 2971 of *LNCS*, pages 259–273. Springer, 2003.

[COPD06] T. Chothia, S. Orzan, J. Pang, and M. T. Dashti. A framework for automatically checking anonymity with *mu*crl. In *Proc. 2nd Symposium on Trustworthy Global Computing, – TGC'06*, pages 301–318, 2006.

[dDLVV08] B. de Decker, M. Layouni, H. Vangheluwe, and K. Verslype. A privacy-preserving eHealth protocol compliant with the Belgian healthcare system. In *Proc. 5th European Workshop on Public Key Infrastructures, Services and Application*, volume 5057 of *LNCS*, pages 118–133. Springer, 2008.

[DDS11] M. Dahl, S. Delaune, and G. Steel. Formal analysis of privacy for anonymous location based services. In *Proc. Joint Workshop on Theory of Security and Applications*, volume 6993 of *LNCS*, pages 98–112. Springer, 2011.

[DKR09] S. Delaune, S. Kremer, and M. D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.

[DLL11] J. Dreier, P. Lafourcade, and Y. Lakhnech. Vote-independence: A powerful privacy notion for voting protocols. In *Proc. 4th Workshop on Foundations & Practice of Security*, volume 6888 of *LNCS*, pages 164–180. Springer, 2011.

[DLL12] J. Dreier, P. Lafourcade, and Y. Lakhnech. Defining privacy for weighted votes, single and multi-voter coercion. In *Proc. 17th European Symposium on Research in Computer Security*, volume 7459 of *LNCS*, pages 451–468. Springer, 2012.

[DS81] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):533–536, 1981.

[DY83] D. Dolev and A. C.-C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *Advances in Cryptology – AUSCRYPT'92*, pages 244–251, 1992.

[HGP09] J. Howlader, A. Ghosh, and T. D. Pal. Secure receipt-free sealed-bid electronic auction. In *Proc. Contemporary Computing – IC3*, volume 40 of *Communications in Computer and Information Science*, pages 228–239. Springer, 2009.

[HM04] D. Henrici and P. Müller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *PerCom Workshops*, pages 149–153, 2004.

[HM08] A. Hevia and D. Micciancio. An indistinguishability-based characterization of anonymous channels. In *Privacy Enhancing Technologies*, volume 5134 of *LNCS*, pages 24–43. Springer, 2008.

[HMZH08] J. Ha, S.-J. Moon, J. Zhou, and J. Ha. A new formal proof model for rfid location privacy. In *Proc. 13th European Symposium on Research in Computer Security –ESORICS*, volume 5283 of *LNCS*, pages 267–281. Springer, 2008.

[HO05] J. Y. Halpern and K. R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–512, 2005.

[Hoa78] C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.

[Hor51] A. Horn. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, 16(1):14–21, 1951.

[JP06] H. L. Jonker and W. Pieters. Receipt-freeness as a special case of anonymity in epistemic logic, June 2006. IAVoSS Workshop On Trustworthy Elections (WOTE'06), Cambridge.

[JPM09] H. L. Jonker, J. Pang, and S. Mauw. A formal framework for quantifying voter-controlled privacy. *Journal of Algorithms in Cognition, Informatics and Logic*, 64(2-3):89–105, 2009.

[KAB09] D. Kotz, S. Avancha, and A. Baxi. A privacy framework for mobile health and home-care systems. In *Proc. Workshop on Security and Privacy in Medical and Home-Care Systems*, pages 1–12. ACM Press, 2009.

[KBM$^+$03] A. Kalam, S. Benferhat, A. Miège, R. Baida, F. Cuppens, C. Saurel, P. Balbiani, Y. Deswarte, and G. Trouessin. Organization based access control. In *Proc. 4th IEEE Workshop on Policies for Distributed Systems and Networks*, pages 120–131. IEEE CS, 2003.

[KLS$^+$10] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh. Wireless sensor networks for healthcare. *Proceedings of IEEE*, 98(11):1947–1960, 2010.

[KR05] S. Kremer and M. D. Ryan. Analysis of an electronic voting protocol in the applied pi calculus. In *Proc. 14th European Symposium on Programming*, volume 3444 of *LNCS*, pages 186–200. Springer, 2005.

[KR11] D. Khader and P. Y. A. Ryan. Receipt freeness of prêt à voter provably secure. *IACR Cryptology ePrint Archive*, 2011:594, 2011.

[KT09] R. Küsters and T. Truderung. An epistemic approach to coercion-resistance for electronic voting protocols. In *Proc. 30th IEEE Symposium on Security and Privacy*, pages 251–266. IEEE CS, 2009.

[KTV10] R. Küsters, T. Truderung, and A. Vogt. A game-based definition of coercion-resistance and its applications. In *Proc. 23rd IEEE Computer Security Foundations Symposium*, pages 122–136. IEEE CS, 2010.

[LBD$^+$03] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *Proc. Information Security and Cryptology – ICISC*, pages 245–258, 2003.

[Liu11] J. Liu. A proof of coincidence of labeled bisimilarity and observational equivalence in applied pi calculus, 2011. Available at `http://lcs.ios.ac.cn/~jliu/`.

[Lou98] K. Louwerse. The electronic patient record; the management of access – case study: Leiden university hospital. *International Journal of Medical Informatics*, 49(1):39–44, 1998.

[Low96] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. 2nd Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, volume 1055 of *LNCS*, pages 147–166. Springer, 1996.

[LVS$^+$09] M. Layouni, K. Verslype, M. T. Sandikkaya, B. de Decker, and H. Vangheluwe. Privacy-preserving telemonitoring for eHealth. In *Proc. 23rd Annual IFIP Working Conference on Data and Applications Security*, volume 5645 of *LNCS*, pages 95–110. Springer, 2009.

[Mat98] V. Matyáš. Protecting doctors' identity in drug prescription analysis, 1998.

[MKDH09] I. Maglogiannis, L. Kazatzopoulos, C. Delakouridis, and S. Hadjiefthymiades. Enabling location privacy and medical data encryption in patient telemonitoring systems. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):946–954, 2009.

[MRS06] M. Meingast, T. Roosta, and S. S. Sastry. Security and privacy issues with health care information technology. In *Proc. 28th Annual Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5453–5458. IEEE CS, 2006.

[NS78] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.

[OC02] S. Older and S. Chin. Formal methods for assuring security of protocols. *Computer Journal*, 45(1):46–54, 2002.

[Oka96] T. Okamoto. An electronic voting scheme. In *Proc. IFIP World Conference on IT Tools*, pages 21–30, 1996.

[Oka97] T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols Workshop*, pages 25–35, 1997.

[PK00] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Proc. Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 1–9. Springer, 2000.

[RCHS03] J. Reid, I. Cheong, M. Henricksen, and J. Smith. A novel use of rBAC to protect privacy in distributed health care information systems. In *Proc. 8th Australian Conference on Information Security and Privacy*, volume 2727 of *LNCS*, pages 403–415. Springer, 2003.

[RR98] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[RS10] M. D. Ryan and B. Smyth. Applied pi calculus. Technical report, School of Computer Science, University of Birmingham, 2010. http://www.cs.bham.ac.uk/ mdr/research/papers/pdf/11-applied-pi.extended.pdf.

[Sch96] S. Schneider. Security properties and CSP. In *Proc. IEEE Symposium on Security and Privacy*, pages 174–187. IEEE Computer Society, 1996.

[SD02] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proc. Privacy Enhancing Technologies*, volume 2482 of *LNCS*, pages 41–53. Springer, 2002.

[SM00] K. Sakurai and S. Miyazaki. An anonymous electronic bidding protocol based on a new convertible group signature scheme. In *Proc. Information Security and Privacy – ACISP*, volume 1841 of *LNCS*, pages 385–399. Springer, 2000.

[SS96] S. Schneider and A. Sidiropoulos. CSP and anonymity. In *Proc. 4th European Symposium on Research in Computer Security*, volume 1146 of *LNCS*, pages 198–218. Springer, 1996.

[SS99] P. F. Syverson and S. G. Stubblebine. Group principals and the formalization of anonymity. In *World Congress on Formal Methods*, volume 1708 of *LNCS*, pages 814–833. Springer, 1999.

[SV09] S. Sneha and U. Varshney. Enabling ubiquitous patient monitoring: Model, decision protocols, opportunities and challenges. *Decision Support Systems*, 46(3):606–619, 2009.

[TGC09] J. M. Tien and P. Goldschmidt-Clermont. Healthcare: A complex service system. *Journal of Systems Science and Systems Engineering*, 18(3):257–282, 2009.

[Tre05] J. Trevathan. Security, anonymity and trust in electronic auctions. *ACM Crossroads*, 11(3):2, 2005.

[Tre07] J. Trevathan. *Privacy and security in online auctions.* Ph.D. dissertation, James Cook University, 2007.

[vMR08] T. van Deursen, S. Mauw, and S. Radomirović. Untraceability of RFID protocols. In *Proc. 2nd Workshop on Information Security Theory and Practices. Smart Devices, Convergence and Next Generation*, volume 5019 of *LNCS*, pages 1–15. Springer, 2008.

[vWB$^+$03] M. van der Haak, A. C. Wolff, R. Brandner, P. Drings, M. Wannenmacher, and T. Wetter. Data security and protection in cross-institutional electronic patient records. *International Journal of Medical Informatics*, 70(2-3):117–130, 2003.

Website References

[1] Proverif code for verifying the AS02 e-auction protocol and DLV08 e-health protocol. `http://satoss.uni.lu/members/naipeng/thesis.php`.

[2] Michael Daly. You Thought You Had Privacy Before the NSA Leak? What About Facebook?, June 2013.
`http://www.thedailybeast.com/articles/2013/06/13/you-thought-you-had-privacy-before-the-nsa-leak-what-about-facebook.html`
last visited at 1 October 2013.

[3] Saul Hansell. Google's Chef Speaks, but Not Its Finance Officer, February 2005.
`http://www.nytimes.com/2005/02/10/technology/10google.html?ex=1266123600&en=60d19019bb842d20&ei=5088&partner=rssnyt&_r=0`
last visited at 1 October 2013.

[4] Jeremy Kirk. Facebook quickly fixes privacy leak in new timeline, group says – The leak allowed 'friends of friends' to see events another person attended, March 2013.
`http://www.computerworld.com/s/article/9237672/Facebook_quickly_fixes_privacy_leak_in_new_timeline_group_says`
last visited at 1 October 2013.

[5] Privacy Rights Clearinghouse. Fact Sheet 2b: Privacy in the Age of the Smartphone, June 2013.
`https://www.privacyrights.org/fs/fs2b-cellprivacy.htm#service provider`
last visited at 1 October 2013.

[6] Daniel Schäfer and Andrew Edgecliffe-Johnson. Bloomberg users messages leaked online, May 2013. `http://archive.is/6nxES`
last visited at 1 October 2013.

[7] T-Mobile. Privacy Policy: Highlights, November 2012.
`http://www.t-mobile.com/company/website/privacypolicy.aspx#what typeinfo`
last visited at 1 October 2013.

Index of subjects

active substitution, 20, 21
anonymity, 3, 113
anonymous authentication, 57, 58, 64, 66, 68, 78, 80, 85
authentication, 24, 25, 56, 61, 74, 78

base type, 20, 22, 27, 28
bi-process, 25, 79–82, 84
bidding booth, 33, 46
bidding-price-secrecy, 30
bit-commitment, 50, 56, 58, 63, 65, 82
bound name, 91, 93, 94
bound variable, 21, 91, 93, 94

chameleon bit-commitments, 33, 35, 36, 39, 43, 45, 46
choice operation, 25, 79
closed process, 21, 23, 27, 28, 32, 53, 55, 91, 92, 99, 100, 102, 103
coercion-resistance in e-voting, 3, 5, 6, 26, 28, 98, 105, 107, 113
constant, 19, 36, 37, 93
context, 21, 28, 31, 42, 44, 93, 95, 99, 103, 105, 106
correspondence property, 24, 25
cryptographic primitive
  anonymous authentication, 57, 58, 64, 66, 68, 78, 80, 85
  bit-commitment, 50, 56, 58, 63, 65, 82
  chameleon bit-commitments, 33, 35, 36, 39, 43, 45, 46
  digital credential, 57, 64
  signed proof of knowledge, 57–59, 65, 66, 68
  verifiable encryption, 57, 58, 61, 65, 68, 71
  zero knowledge proof, 2
  zero-knowledge proof, 57, 62–66, 77, 80, 113

digital credential, 57, 64
doctor anonymity, 51, 55, 74, 79, 80
doctor untraceability, 51, 55, 74, 80, 81, 85
Dolev-Yao adversary, 3, 11, 77
domain, 21

e-auctions, 3, 5–8, 29
e-health, 5, 47–56
e-voting, 3, 5, 6, 8, 16, 17
enforced prescribing-privacy, 51, 53, 55, 61, 82, 85
enforced privacy property
  coercion-resistance, 15
  coercion-resistance in e-voting, 3, 5, 6, 26, 28, 98, 105, 107, 113
  enforced prescribing-privacy, 51, 53, 55, 61, 82, 85
  independency of enforced prescribing-privacy, 51, 54, 88, 112, 114
  independency of prescribing-privacy, 50, 51, 54, 61, 88, 112
  receipt-freeness in e-auctions, 3, 8, 30, 32, 34, 41, 43, 46, 112, 113
  receipt-freeness in e-voting, 3, 5–7, 15, 17, 26, 27, 53, 88, 98, 105, 107, 113
equational theory, 20, 36, 45, 63, 64, 71
evaluation context, 21, 22, 105, 106
extended process, 20, 22, 27

frame, 21–23, 44
free name, 21, 91
free variable, 21, 91
function symbol, 19, 20, 36, 44, 63–66

ground term, 21, 91

HII, 56

independency of enforced prescribing-privacy, 51, 54, 88, 112, 114
independency of prescribing-privacy, 50, 51, 54, 61, 88, 112

internal reduction, 22

labelled bisimilarity, 22, 23, 25
labelled reduction, 22

MPA, 56

name, 19, 20, 22, 27, 91, 93

Open-bid auctions, 29

physical setting of protocols
  bidding booth, 33, 46
  untappable channel, 33, 36, 38, 39, 43, 46, 62, 85, 86, 113
plain process, 20, 21, 23, 26–28, 32, 53, 55, 91, 92, 94, 97, 99, 100, 102, 103
prescribing-privacy, 51, 52, 55, 61, 82, 84, 85
privacy property
  anonymity, 3, 12, 13, 113
  bidding-price-secrecy, 30
  doctor anonymity, 51, 55, 74, 79, 80, 85
  doctor untraceability, 51, 55, 74, 80, 81, 85
  prescribing-privacy, 51, 52, 55, 61, 82, 84, 85
  strong bidding-price-secrecy, 31
  strong doctor anonymity, 51, 55, 80
  strong doctor untraceability, 51, 56, 81
  unlinkability, 3, 12, 13
  untraceability, 3, 12, 113
process
  active substitution, 20, 21
  closed process, 21, 23, 27, 28, 32, 53, 55, 91, 92, 99, 100, 102, 103
  context, 21, 28, 31, 42, 44, 93, 95, 99, 103, 105, 106
  domain, 21
  evaluation context, 21, 22, 105, 106
  extended process, 20, 22, 23, 27
  frame, 21–23, 44
  plain process, 20, 21, 23, 26–28, 32, 53, 55, 91, 92, 94, 97, 99, 100, 102, 103
process relation
  internal reduction, 22
  labelled bisimilarity, 22, 23, 25
  labelled reduction, 22
  static equivalence, 22, 23
  structural equivalence, 22
ProVerif
  bi-process, 25, 79–82, 84
  choice operation, 25, 79

receipt-freeness in e-auctions, 8, 30, 32, 34, 41, 43, 46, 112, 113
receipt-freeness in e-voting, 3, 5–7, 17, 26, 27, 53, 88, 98, 105, 107, 113
role in e-health
  HII, 56
  MPA, 56

scope of terms
  bound name, 91, 93, 94
  bound variable, 21, 91, 93, 94
  free name, 21, 91
  free variable, 21, 91
sealed-bid auctions, 29
secrecy, 24, 31, 35, 42, 56, 61, 74, 76, 77
security property
  authentication, 24, 25, 56, 61, 74, 78
  correspondence property, 24, 25
  secrecy, 4, 24, 25, 31, 35, 42, 56, 61, 74, 76, 77
signed proof of knowledge, 57–59, 65, 66, 68
static equivalence, 22, 23
strong bidding-price-secrecy, 31
strong doctor anonymity, 51, 55, 80
strong doctor untraceability, 51, 56, 81
structural equivalence, 22
system
  e-auctions, 3, 5–8, 29–32
  e-health, 5, 47–56
  e-voting, 1, 3, 5, 6, 8, 12, 15–17, 26–28
  Open-bid auctions, 29
  sealed-bid auctions, 29

term, 19–25, 79, 94, 96, 98, 107, 108
  base type, 20, 22, 27, 28
  constant, 19, 36, 37, 93
  equational theory, 20, 36, 45, 63, 64, 71

function symbol, 19, 20, 36, 44, 63–66
ground term, 21, 91
metavariable, 20
name, 19, 20, 22, 27, 91, 93
scope of terms
bound name, 91, 93, 94
bound variable, 21, 91, 93, 94
free name, 21, 91, 93
free variable, 21, 91
variable, 19–22, 27, 28, 37, 43, 52, 91–93, 96–98, 101
the applied pi calculus, 5–9, 19–23, 25, 31, 35, 51, 62, 91, 111
the DKR framework, 6, 7, 16, 17, 19, 26–28

unlinkability, 3
untappable channel, 33, 36, 38, 39, 43, 46, 62, 85, 86, 113
untraceability, 113

variable, 19–22, 27, 28, 37, 43, 52, 91–93, 96–98, 101
verifiable encryption, 57, 58, 61, 65, 68, 71

zero knowledge proof, 2
zero-knowledge proof, 57, 62–66, 77, 80, 113

List of Symbols

$\mathtt{a}$	General name	19
$\mathtt{b}$	General name	19
$\mathtt{c}$	General channel name	19
$\mathtt{c}_{out}$	Channel for sending messages to the adversary in a collaboration	100
$\mathtt{c}_{in}$	Channel for reading messages from the adversary in a collaboration	100
$\mathtt{ch}$	General public channel	25
$\mathtt{chc}$	Channel for sending information to the adversary	26
$\mathtt{m}$	General name	19
$\mathtt{n}$	General name	19
$\tilde{\mathtt{n}}$	List of names	21
f	General function	13
v	Meta variable (either a name or a variable)	20
x	General variable	19
y	General variable	19
z	General variable	19
A	General extended process	20
B	General extended process	20
C	General extended process	20
E	Equational theory	20
M	General term	19
N	General term	19
T	General term	19
P	General plain process	20
Q	General plain process	20
R	General plain process	20
$\hat{R}$	Process for a role with two variables: identity and target data	98
R_T	Process for attacking third parties	105
R_D	Process for defending third parties	105
$\mathcal{R}$	General relation	23
$\mathtt{C}_f$	Size of equivalence classes induced by f	14
$\mathtt{P}_f$	Partitions of invocations induced by f	13
$\mathtt{Q}_f$	Invocations per participant induced by f	13
$\mathtt{U}_f$	Participant set induced by f	13
$\|\mathtt{U}_f\|$	Size of participant set that is induced by f	14
$\{M/x\}$	Substitution: replace x with M	20
f	General function symbol	19
g	General function symbol	23
k	Indication of containing a key	36
α	Transition label	22

ϵ	The channel is not necessary	100
ω	General type of terms	20
σ	General substitutions	23
τ	Target data	99
ξ	Associations between channels and variables	103
η	A set of internal channels in the definition of coalition behaviour	103
$\varPsi$	Terms sent to the adversary in a collaboration	100
$\varPhi$	Terms replaced by the adversary in a collaboration	100
$\varTheta$	Communication specified in a coalition	102
$\varDelta$	Substitutions specified in a coalition	102
$\varPi$	Assignments specified in a coalition	102
$\varGamma$	Sending communication of a user in a coalition	103
$\varOmega$	Internal channels between users in a coalition	104
$\emptyset$	Empty set	23
$\not\in$	Negation of set membership	22
$\cup$	Set union	23
$\cap$	Set intersection	23
$\subseteq$	Set containment	23
$\forall$	For all quantification	97
$\mathit{0}$	Null process	20
$P \mid Q$	Parallel composition	20
$!P$	Replication	20
$\nu\mathtt{n}.P$	Generate name $\mathtt{n}$ and bound it in process P	20
$\mathsf{in}(v, x).P$	Read message on channel v and bound the message to x in P	20
$\mathsf{out}(v, M).P$	Send message M on channel v, then run P	20
$\mathcal{C}[\_]$	General context	21
$\neq$	Negation of equality	23
$::=$	Definition of terms and processes	19
$=_E$	The equivalence relation induced by E	20
$\neq_E$	Not equivalent by E	22
$\equiv$	Structural equivalence	22
$\rightarrow$	Internal reduction	22
$\xrightarrow{\alpha}$	Labelled reduction	22
$\approx_s$	Static equivalence	23
$\approx_\ell$	Labelled bisimilarity	23
$\widehat{=}$	Definition of collaboration, hide on channels, and coalition	26
$\leadsto$	Correspondence to	24
$\leadsto_{inj}$	Injectively correspondence to	24
$f\langle M\rangle$	Event $f\langle M\rangle$ with parameter M	24
$g\langle M\rangle$	Event $g\langle M\rangle$ with parameter M	24
$*$	Zero or more	23
?	Any candidate	28