

Exploring, Expanding and Evaluating Usable Security in Online Banking

Sven Kiljan

Copyright © 2017 Sven Kiljan

ISBN: 978-94-6233-648-3

NUR: 965

Typeset using L^AT_EX 2_ε (MikT_EX)

L^AT_EX template by Pim Vullers (https://github.com/pimvullers/phd_thesis)

Printed by Gildeprint (<https://www.gildeprint.nl>)



This research is part of the Dutch Research Program on Safety and Security of Online Banking, also known by its Dutch title: Kennisprogramma Veiligheid Digitaal Betalingsverkeer (KVDB). The PhD program it hosted was coordinated by Open University of the Netherlands. The author of this thesis was employed at Open University of the Netherlands from September, 2015 to August, 2016.



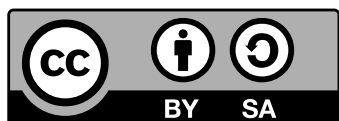
KVDB was funded by the Dutch banking sector (represented by the Dutch Banking Association) and the Dutch National Police (represented by the Police Academy of the Netherlands).



Radboud University



The work on which this thesis is based was performed at NHL University of Applied Sciences and Radboud University. The author of this thesis was employed at NHL University of Applied Sciences from September, 2012 until August, 2015.



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. To view a copy of this license, please visit <http://creativecommons.org/licenses/by-sa/4.0/>.

Exploring, Expanding and Evaluating Usable Security in Online Banking

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Open Universiteit
op gezag van de rector magnificus prof. mr. A. Oskamp
ten overstaan van een door het College voor promoties ingestelde commissie
in het openbaar te verdedigen

op vrijdag 9 juni 2017 te Heerlen
om 10:30 precies

door

Sven Zdenjek Kiljan
geboren op 2 oktober 1985 te Alkmaar

Promotoren

Prof. dr. M.C.J.D. (Marko) van Eekelen
Open Universiteit
Radboud Universiteit

Prof. dr. W.Ph. (Wouter) Stol
Open Universiteit

Co-promotor

Dr. ir. H.P.E. (Harald) Vranken
Open Universiteit
Radboud Universiteit

Overige leden beoordelingscommissie

Prof. dr. S. (Sjouke) Mauw
Universiteit van Luxemburg

Dr. K. (Karen) Renaud
Universiteit van Glasgow, Verenigd Koninkrijk

Prof. dr. G.C. (Gerrit) van der Veer
Open Universiteit
Vrije Universiteit Amsterdam

Prof. dr. T.E.J. (Tanja) Vos
Open Universiteit

Dexe scriptie is opgedragen aan mijn dierbare moeder.

Preface

Preface

This thesis symbolizes the closure of a four year experience. All other parts of it are dedicated to the products of the research that was conducted in this period. Despite my name as the author of this book, each part has been thoroughly examined, adjusted, reviewed, and rewritten with the support of numerous people. Due to the high level of specialization, the fieldwork often feels like a lonely endeavor, but I was never alone. To explain my feelings in the purest way possible, this preface is written solely by me. Consider it a letter dedicated to everyone who contributed directly or indirectly to my work and the final result that is this thesis.

First, let me thank all of you who I cannot refer to directly. Non-disclosure, limited space... Whatever the reason might be, know that I thought of you and that I appreciated the time and effort you spend to support my work.

Professional support for my research came from the Dutch Research Program on Safety and Security of Online Banking (in Dutch: Kennisprogramma Veiligheid Digitaal Betalingsverkeer, KVDB). While the program financed the work, I consider my colleagues as the most valuable resource this program provided. It is rare to meet a group that is both witty and silly, knowledgeable and practical, independent and cooperative, and stubborn and supportive. For some, the day that they meet such people would be the most important day of their lives. But for me, it was Tuesday. Many Tuesdays, actually. On this day of the week, I often traveled to the NHL University of Applied Sciences where my direct colleagues within the research program conducted their work. A single trip by car took two hours, and some would consider it silly to spend so much time on the road to meet others of which their work does not fully align with their own. I can honestly say that I have never regretted any second of the journey to see my fellow researchers and close friends. Sanne Boes, Jurjen Jansen and Rutger Leukfeldt, I thank you for the many laughs and cries that we shared in our office at NHL. Mariëtte van Kuik, while the period we shared in the program was short, our time is not any less appreciated. Since I am now writing about NHL, there are those who were not part of the program but who cannot go unmentioned. Joyce, Marja, Renske, Sander, and all the others of lectoraat Cybersafety, to you I wish the best of luck with the new Cyber Science Center.

Nicolien Kop, Evert Stamhuis, Wouter Stol, due to your shared efforts as initiators, I was able to join a research program in which I could extend my knowledge and experience to an unimaginable degree. For this I thank you. I want to give a special mention to Marko van Eekelen. It is true that you are an initiator and supervisor of the program. It is also true that you carry the formal role of being my coordinating

promotor. However, the most important truth is that you were the best daily supervisor a PhD could ask for. Many afternoons were spent in your office in which we talked, debated and even had heated arguments about the direction of the research, the methods to apply and the importance of the almost infinite aspects that we had to keep track of. I cannot say that we always fully agreed with each other, but do not consider that a negative point. The alternative views you gave me were a fuel source that recharged my motivation and inspiration. These are aspects which PhD candidates often struggle with over the years, but you gave me plenty whenever I was running on empty. I cannot thank you enough for this.

Although formally not part of KVDB, I do consider my co-promotor Harald Vranken an essential part of the program. Where Marko guided my research from a higher level, you supplemented the lower level with indispensable in-depth technical advice. Harald, do not consider your contribution and my gratitude any less compared to my other colleagues.

Two other direct contributors to my work are Hugo Jonker (Open University of the Netherlands) and Rolando Trujillo Rasua (University of Luxembourg). They gave me insight in formal protocol analysis, a field that is not my own and which I touched briefly on while writing this thesis. Your willingness to examine my suggestions and to give good advice on how to model it in Scyther gave me a needed push in the right direction.

Another contributor was Safet Acifovic, a former student of Radboud University. He assisted by formally verifying an earlier part of my work. Safet, thank you for your time spent with ProVerif to let Marko and me know what could be improved. I wish you good luck with your career.

In addition, I would like to thank Sung-Shik Jongmans. Your investment in time to evaluate all those authentication methods is highly appreciated.

There are also those who contributed indirectly to my work, but not any less significantly. To talk about them, I have to start from the beginning. The first day of my four year PhD course started on Monday 3 September 2012 at my daily workplace for my PhD program, which was in the Digital Security research group of Radboud University in Nijmegen. There I shared rooms and knowledge with some of the brightest minds in the area of technical and information security. Some of those worked directly for Radboud University while others were like me, guest researchers from Open University of the Netherlands. Wherever these fine folk came from, there are simply too many of them to thank by name. However, I cannot skip over a few important and honorable mentions. Let me start with Bernard van Gastel, with which I shared office rooms in different periods. The technical conversations we had will never be forgotten, and I am glad that I was able to introduce and partially convert you to Arch Linux. I hope it will serve you well.

For a while I also shared rooms with some of the members of the Privacy & Identity Lab. The nature of my research never focused much on privacy, but this was very much compensated by the talks we had. Merel Koning and Michael Colesky, thank you for tolerating my presence every time I performed the role of the devil's advocate. Time just flew by whenever we discussed something from the news or a potential new concept that popped in one of our minds, and it was time well spent. Never give up the good fight.

Two of the most influential persons I met at Radboud University are Anna

Krasnova and Markus Klinik. Your characters and humor lifted up my spirit whenever I was feeling down, and this extended well beyond the work floor into my personal life. For me, our friendship is as valuable as close family bonds. It is why I invited each of you to fulfill the role of paranymp for my thesis defense, and I appreciate it deeply that we can share this day together.

Whereas Anna and Markus touched my heart, someone else from Radboud University managed to capture it. Manxia Liu, it was at the start of the third year of my research program that you changed my life, but only in good ways. You supported me throughout the second half of my PhD course as my friend and partner. At the time of writing you are still working on your own PhD course. I will be there for you as you were there for me, and hope that we can continue our quest to create and have a comfortable and happy life together. Wǒ ài nǐ, xiǎo tùzǐ.

I also would like to thank some of the people who brought me to the point where I actually could start the PhD program through which I could do some amazing work and meet so many new people. Youri Wagenaar and Paul Pierlo, thank you for being for being such close fellow students and friends at various parts of my education. This thanks also spreads to your families. In Dutch we use the phrase ‘child at home’ for a friend who is treated as being part of the family whenever he or she visits the home of another friend, and I see myself like this due to the warm welcome I always received. During my PhD I did not have much time to spend with you, mostly because of my move to the other side of the country. However, I am sure we will manage to catch up.

Monique, you have been the best sister a brother could ask for. I really enjoyed our company the last couple of years, and hope we will still be able to do this for a long time. Your move to the south of the country brought many opportunities for us to visit each other again. You married Ramon, a guy who passed the Overprotective Brother’s Extremely Strict Background and Integrity Check with a perfect grade, and I also thank him for the hospitality at your place. My visits to you both always offered a brief distraction from my work whenever it was necessary.

Niels, we did not have much contact due to various reasons, but the little we had I really enjoyed. Stay safe, sane and healthy, brother.

As for my final words, I would like to show gratitude to my parents. Most parents ask themselves if they should have done something different while raising their child. This work is proof that you did the right thing. I was not an easy child. Despite this, you kept motivating me to become educated and to do the work I love most. I am happy that we managed to live together so well before I moved to Nijmegen.

Dad, thanks for the early introduction to modern computer technology and to the information security field, and of course for our many sparring sessions. Your energy is astounding, and I can only aspire to reach such dedication and enthusiasm.

Mom, you know better than no one else how much of a trouble maker I could be. It was you who always believed that your son could be more than what others gave him credit for, and who acted on this believe whenever it was necessary. Without your actions, I would not be where I am today. That is why I dedicate this thesis to you. It is a result of your determinism, your faith in me, and the love that you gave me throughout my life. I love you, mom.

*Sven Kiljan
Nijmegen, April 2017*

Contents

Preface	ix
List of acronyms	xvii
1 Introduction	1
1.1 Context	1
1.2 Problem statement	5
1.3 Research methodology	6
1.4 Research overview	6
1.4.1 Exploring security offered by banks	6
1.4.2 Expanding on security not offered by banks	7
1.4.3 Evaluating online banking authentication methods	8
1.5 Origin of chapters and contribution	10
1.6 Reading guide	11
I Exploring online banking security	13
2 A survey of authentication and communications security in online banking	17
2.1 Introduction	17
2.2 Online and mobile banking development	19
2.2.1 A short history of electronic banking	19
2.2.2 Development and acceptance of online and mobile banking	22
2.2.3 Standardization: the pressure to go multi-platform	24
2.2.4 Security implications	26
2.3 Customer to bank authentication	27
2.3.1 Single or multi-factor?	28
2.3.2 Knowledge: Passwords and PINs	32
2.3.3 Possession: Physical and digital	33
2.3.4 Biometrics and behavior anomaly detection	38
2.3.5 Comparing 2002, 2013 and 2015	40
2.4 Bank to customer authentication and communications security	41
2.4.1 Vulnerabilities	42
2.4.2 Additional TLS functions	45
2.4.3 Geographical spread of vulnerabilities and functions	48
2.4.4 SSL/TLS overall observations	49

2.5	Discussion, limitations and further research	49
2.6	Related work	51
2.7	Concluding remarks	51

II Expanding transaction authorization options 55

3 What You Enter Is What You Sign: input integrity in an online banking environment 59

3.1	Introduction	60
3.2	Transaction authentication	61
3.2.1	Traditional transaction authentication (TTA)	61
3.2.2	Customer verified transaction set authentication (CVTSA)	62
3.2.3	Entered single transaction authentication (ESTA)	63
3.3	Traditional transaction authentication (TTA)	64
3.4	Customer verified transaction set authentication (CVTSA)	66
3.5	Entered single transaction authentication (ESTA)	68
3.6	Formal verification	71
3.7	Related work	71
3.7.1	Devices that apply keyboard emulation	71
3.7.2	Interactive smart card terminals	72
3.7.3	Mobile devices used for out-of-band verification	73
3.7.4	Authentication solutions in trusted execution environments	73
3.8	Further research	74
3.9	Concluding remarks	75

4 User-friendly manual transfer of authenticated online banking transaction data 77

4.1	Introduction	78
4.2	Analysis of steps and methods to generate an MC	80
4.2.1	(De)compression algorithms	82
4.2.2	Securing and verifying authenticity and integrity	83
4.2.3	Encoding/Decoding methods	84
4.2.4	Code transfer	85
4.3	Online banking case study	85
4.3.1	Create the message	87
4.3.2	Create the Message Code	88
4.3.3	Transfer the Message Code	91
4.3.4	Verify the Message Code	91
4.3.5	Further processing	93
4.3.6	Attack analysis and mitigation	93
4.4	Formal verification	95
4.4.1	Assumptions and notation	95
4.4.2	Security requirements	96
4.4.3	Constraints	96
4.4.4	Formal verification example	96
4.5	Discussion, limitations and further research	99
4.6	Concluding remarks	100

III Evaluating authentication and authorization schemes 101

5	Theoretical evaluation to quantify qualitative characteristics	105
5.1	Introduction	106
5.2	Background and related work	107
5.2.1	Authentication evaluation mechanisms	107
5.2.2	Secure usability aspects	107
5.2.3	Proposed online banking transaction authentication methods	108
5.3	Choosing an evaluation method	108
5.4	Renaud's mechanism at a glance	111
5.5	Expanding Renaud's mechanism with the feasibility dimension . . .	112
5.5.1	Aspects of the feasibility dimension	114
5.5.2	Environmental factor: user correction	116
5.5.3	Adapted formulas	116
5.5.4	Relative scoring formulas	117
5.6	Multi-user evaluation	117
5.7	Evaluated authentication methods	119
5.8	Applying the mechanism	123
5.8.1	Proposals which lack entity authentication information	123
5.8.2	Applying feasibility aspects	124
5.8.3	Environmental factors	125
5.8.4	Performing a multi-user evaluation	125
5.9	Research data of the evaluated authentication methods	131
5.10	Resulting values	132
5.10.1	Effects of the feasibility dimension	132
5.10.2	Overall evaluation	133
5.10.3	Information scheme influence	136
5.10.4	Variation between the raters	137
5.11	Limitations, discussion and further research	140
5.12	Applying the mechanism on What You Enter Is What You Sign . . .	141
5.13	Concluding remarks	144
6	Practical evaluation to measure secure usability	145
6.1	Introduction	146
6.2	The need for a new evaluation framework for online banking security	146
6.3	Design of a virtual bank for secure usability research	147
6.4	A virtual bank for secure usability research: setup of experiments using a proof of concept	150
6.4.1	What You See Is What You Sign	153
6.4.2	What You Enter Is What You Sign	154
6.5	Validation of the use of the proof of concept virtual bank	154
6.6	Discussion and further research	159
6.6.1	Evaluation of the experiment and lessons learned	159
6.6.2	Building forth on the resulting data	159
6.7	Concluding remarks	160

Discussion and future work	163
Summary	169
Samenvatting	173
Bibliography	177
Curriculum Vitae	189

List of acronyms

ACM	Association for Computing Machinery
ASCII	American Standard Code for Information Interchange
ATM	Automated Teller Machine
BCBA	Blockwise Chosen-Boundary Attack
BEAST	Browser Exploit Against SSL Tool
CBC	Cipher Block Chaining
CPU	Central Processing Unit
CR	Challenge-Response
CRIME	Compression Ratio Info-leak Made Easy
CVTSA	Customer Verified Transaction Set Authentication
DOS	Disk Operating System
ENA	Enter Non-Aggregated
ESTA	Entered Single Transaction Authentication
EV	Extended Validation
HID	Human Interface Device
HMA	Hybrid Mobile Application
HPKP	HTTP Public Key Pinning
HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL
IBAN	International Bank Account Number
ISO	International Organization for Standardization
KVDB	Kennisprogramma Veiligheid Digitaal Betalingsverkeer
MAC	Message Authentication Code
MC	Message Code
MD	Mobile Device
MD5	Message Digest 5
NFC	Near-Field Communication
NHL	Noordelijke Hogeschool Leeuwarden
NL	Nederland
NN	None None
NIST	National Institute of Standards and Technology
OTP	One-Time Password
PC	Personal Computer
PS/2	Personal System/2

PIN	Personal Identification Number
POODLE	Padding Oracle On Downgraded Legacy Encryption
QR	Quick Response
SC	Smart Card
SHA	Secure Hash Algorithm
SMS	Short Message Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
TAC	Transaction-dependent Authorization Code
TAM	Technology Acceptance Model
TCP	Transmission Control Protocol
TEE	Trusted Execution Environment
TEP	Trusted Entry Pad
TLS	Transport Layer Security
TTA	Traditional Transaction Authentication
URL	Uniform Resource Locator
USB	Universal Serial Bus
VA	Verify Aggregated
VNA	Verify Non-aggregated
VPN	Virtual Private Network
WAP	Wireless Application Protocol
WTLS	Wireless Transport Layer Security
WWW	World Wide Web
WYseeIWYS	What You See Is What You Sign
WYenterIWYS	What You Enter Is What You Sign
ZTIC	Zone Trusted Information Channel

Chapter 1

Introduction

The main theme of the research in this thesis is the usable security of online banking. Security systems which banks have implemented today are examined, suggestions for improvements are made and new mechanisms are thought of to evaluate existing and proposed online banking security systems that interact with the user. This introduction serves as a primer for the rest of the thesis by describing the research and organizational context, and by giving an overview of the completed research. The research is based on five papers, and it is noted how these papers are integrated into the thesis. A reading guide at the end of the introduction provides recommendations on how the thesis can be read by different audiences.

1.1 Context

Money, in its many forms and shapes throughout history, is a very universal way to gain, possess and trade wealth [Men92]. It is the most commonly accepted ‘good’ to trade with, and the value of anything is often expressed in money to give it a context that everyone can relate to.

There are those that are not afraid to gain money through ways that are legally unacceptable. Through risk assessment, criminals determine whether such actions are worth their trouble [NW97]. Money is an attractive target for criminals who are out for self-enrichment, since it is easy to spend on anything due to its universal nature.

Robbing banks physically is high risk, high reward work for bank robbers. Physical exposure greatly increases the risk. When caught, in a best case scenario the robbers have to undergo processing by the legal system, which can end in a conviction.¹ In the worst case, the robbers are killed.²

With the introduction of electronic money and payments through Internet-enabled consumer-owned computers, both banks and their customers are able to handle

¹Bank Robber Sentenced to 96 Months in Federal Prison (2015): <https://www.fbi.gov/contact-us/field-offices/seattle/news/press-releases/bank-robber-sentenced-to-96-months-in-federal-prison>

²A Heist Gone Bad (2015): <https://www.policefoundation.org/publication/a-heist-gone-bad/>

money much more flexibly. No longer are bank account management and the transfer of money restrained physically or by time.

Extending traditional banking functions to the digital domain also extends the security domain. Users are not physically required to visit a bank anymore, and neither are criminals. Instead of robbing a bank directly, criminals have the option to interfere in the communication between user and bank. The user can be tricked to disclose information which a criminal can use to impersonate the user when communicating to the bank [DTH06]. Another attack vector is the user's computer, which cannot be trusted by the bank due to the lack of a trusted computing base [RP98]. The user could also be tricked by malicious software (malware) to give valuable information to criminals. Malware can also be used to silently inject transactions into a user's existing session with the bank.^{3,4} The user is unaware of these transactions, but the bank will receive them as sent by the user.

Banks sometimes have security expectations of regular users that are too high [MVO08]. In addition, banks should not ask too much attention from the user concerning security since the user's time is a valuable commodity [Her09]. The research in this thesis is focused on improving the secure use of online banking by examining existing implemented and proposed security methods, and by proposing new methods.

Organizational context

This thesis is a product of the Research Program on Safety and Security of Online Banking (Kennisprogramma Veiligheid Digitaal Betalingsverkeer). The program is a joint effort between universities, several banks and the Dutch National Police. The universities are Open University of the Netherlands, NHL University of Applied Sciences and Radboud University. The Dutch banking sector and the Dutch National Police provided funding and information, and are represented in the program respectively by the Dutch Banking Association and the Police Academy of the Netherlands.

The goal of the program is to make online banking more safe and secure by improving defenses and the cooperation between involved parties, and by destabilizing online criminal networks. It does so through four inter-disciplinary studies. These are as follows:

- *The criminological study* by Rutger Leukfeldt, which focuses on mapping and destabilizing online organized crime [Leu14b, Leu14a, LJ15, Leu15b, Leu15a, JL15, LKS16c, LKS16b, LKS16a, JL16, BL16]. An important part of the research was a study of criminal investigations in the Netherlands, Germany, United Kingdom and United States of America. Important conclusions from the research are that social ties play an important role in the origin and growth of cybercriminal networks, and that such networks often consist of static core members, dynamic facilitators and money mules.

³ A Case Study of EuroGrabber (2012): https://microwire.info/wp-content/uploads/2012/12/120712_chkp_eurograbber_wp.pdf

⁴ Automating Online Banking Fraud (2012): http://www.trendmicro.com/cloud-content/apac/pdfs/security-intelligence/white-papers/wp_automating_online_banking_fraud.pdf

- *The legal study* by Sanne Boes, which examines the involvement of banks in the fight against online banking fraud [BL16]. Legal boundaries of private contribution to criminal justice are unclear, and so is the allocation of tasks and responsibilities. The aim of this study is to outline the legal framework and to make a proposal to improve it, so that results of private investigation can be used in criminal proceedings in a legally acceptable way, while the public-private arrangement of tasks and responsibilities remains optimally effective. Data has been gathered by interviewing stakeholders and performing legal desk research, including laws, case law and legal doctrines. As is common in legal studies, a thesis is written first before it is followed by publications, which explains the single publication from this study at the time of writing.
- *The socio-psychological study* by Jurjen Jansen is dedicated to improving the resilience of online banking customers [Jan15, LJ15, JVZS16, JL16, JvS16]. Trust that customers have in online banking and banks is also an important subject in this research. Part of the research consisted of interviews with online banking fraud victims and a survey among online banking customers. A provisional conclusion from the research is that there are indications of a trust paradox: when people trust banks less, they tend to perform online banking more securely. Another provisional conclusion is that every online banking customer carries some risk of victimization.
- *The technological study* by Sven Kiljan, in which the usable security of online banking is explored, expanded and evaluated [KVvE14, KvEV16, KVvE16a, KVvE16b, KSC⁺16]. This thesis is based on this study.

While each study mainly focused on its own products, multi-disciplinary co-operation was sought for areas that touched or overlapped. Organizationally, the research team worked together for one day of almost every week at the NHL University of Applied Sciences in Leeuwarden. This pleasant cooperation consisted of meetings and regular brain-storm sessions. Tangible results include several joint papers [LJ15, JL15, JL16, BL16] as well as joint presentation programs. Examples of the latter include the Dutch Society of Criminology congresses in 2013 and 2015, and the 16th Annual conference of the European Society of Criminology in 2016. Intangible results include advice for and reviews of each other's research methods and results from the perspectives of different disciplines. The work in this thesis is a result of the technological study. Due to its both technological and human-centric nature, this work would not have been possible without the valuable input from the other research program members.

Research context

Security and usability can be considered opposites. If the user is required to perform security actions in addition to functional actions, it inherently decreases the usability of the system since the user has to perform more actions than what is strictly necessary to fulfill the user's job or goal. *Usable security* [SF05] is more than just a concatenation of the distinct terms usability and security. It has to be, since otherwise the term would be little more than an oxymoron. For this thesis, the term

refers to the usability of security actions. This excludes other actions related to the functional (non-security) use of a system.

User identification is a required capability of a multi-user system. Through user identification, the system determines which functions and data are available to whom. Providing a username (or some other form of identification that is associated with the user, such as a bank account number) would be a functional action, since it is required to let the system know who is currently using it. By only asking for a username, the system is fully balanced towards usability. A security action can be introduced to ensure that users do not get access to functionality that is not assigned to them. A classical example is the requirement to provide a password. From a functional perspective, the user is required to perform additional actions and spend more effort for the same outcome as when no password would be required. Password sharing and/or storing is often discouraged, so aside from entering a password it is also expected from the user that the password is remembered. The user even has to do this between moments that the system is not used. If the user forgets the password, the system becomes unusable and some kind of recovery procedure must be initiated.

Introducing the security action sacrificed some usability for security since the user is required to perform more work in order to use the system. Any security action would do this to some degree if it involves the user. Security actions are required to prevent illegitimate use. Systems which require security actions should be designed in a way that prevents the user from being the ‘weakest link’ due to a lack of usability [SF05]. Security must be high enough to make attacks unattractive to potential adversaries. Usability of the security actions must be high enough to make the proper execution of those actions attractive to users. The term *usable security* is used to define how a system’s usability and offered security level influence each other. The negative impact that security actions have on overall usability of the system can be limited by improving the usability of these actions. Performing security actions correctly also improves security, which is why the usability of a security system should cater to this. In the context of usable security, usability and security are intertwined and not opposites.

A simple reason for why banks started offering online banking through user-owned computers and connections is that it is cheap. A bank does not have to invest in a computer and a connection from the bank to each customer. Another reason is that online banking users will most likely not want to carry around a separate computer just for their online banking needs. Unfortunately, user-owned computers are considered untrustworthy. Social engineering attacks (such as phishing) allow an adversary to assume a certain role as perceived by the user, such as his or her bank. Adversaries can use this to get authentication credentials from users to steal money from their bank accounts. Malicious software (malware) attacks can also be used for authentication credential stealing. Regular users are not capable to protect themselves against such online banking attacks [MVO08].

This does not make online banking different from other services. Email, instant messaging and online video games are all examples that use the user’s computer and network connection. These services require user identification and authentication, and all could be compromised through credential stealing. What makes online banking different is that it presents a direct target for fraudulent financial transac-

tions. Email, instant messaging and online video game accounts can present some monetary value, but this will not be true for every account at every type of service provider. For example, a private email address which is only used to informally communicate with direct relatives and friends would not have much monetary value, whereas a corporate account could. Successful attacks on accounts of financial institutions are more directly profitable in most cases since most accounts have direct financial value. Also, in many developed countries online banking is quite ingrained in the population.^{5,6,7} The potential victim pool is therefore large enough to present many opportunities for criminals.

Multi-factor authentication is implemented by many banks, but it does not offer enough protection [Sch05]. An adversary acting as a man-in-the-middle can still trick a user to hand out one-time passwords or to give a response to challenge-response authentication. A more subtle attack that can be executed through malware is the silent injection of additional transactions in an existing ‘secure’ session.⁸ Based solely on the received set of transactions, a bank would not know the difference between transactions added by the user and those added by an adversary. The malware also makes sure that users do not see the additional transactions on their computers, and can also ‘correct’ the presented account balance to an amount that the user expects after the attack has been completed.

Banks are not only interested in verifying the user’s identity through user authentication, but also in which transactions are authorized by the user. This presents a problem when both banks and users cannot fully trust what the other party is saying due to the untrusted element that represents the user’s computer [RP98]. This is where usable security has an important role. Banks can implement additional security systems to mitigate the lack of trust in users’ computers. This will change the work flow of the user, possibly in a manner that negatively influences the (perceived) usability and therefore possibly the security as well, despite that the purpose of such systems is to have a positive influence on security.

1.2 Problem statement

Establishing security in a digital world has many challenges. Risks can be high, varying from loss of information and privacy to loss of important assets. In particular, in online banking the stakes are high. Money transactions are an attractive target for adversaries. This asks for rigid security measures to be put in place to secure such transactions with the highest possible guarantees.

On the other hand, customers of banks want online banking to be not only safe but also easy to use. In a commercially competitive environment, every transaction security measure is deemed to be a trade-off between ease of use and ultimate security. Research is needed to solve the many challenges involved.

⁵Home and mobile banking use in China in 2013: http://www.iresearchchina.com/content/details7_18315.html

⁶Home and mobile banking use in the United States in 2011-2014: <http://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201503.pdf>

⁷Online banking use in Europe in 2016: <http://ec.europa.eu/eurostat/tgm/table.do?tab=table&plugin=1&language=en&pcode=tin00099>

⁸See footnotes 3 and 4.

This thesis aims to contribute to this research area by exploring, expanding and evaluating usable security in online banking.

1.3 Research methodology

In academic research, the security of online banking is still a relatively new subject. Some exploratory research was previously conducted [CDDC⁺02], but that was more than ten years before the work started that resulted in this thesis. The first step of the research as described in this thesis was the exploration of both the research field and actual online banking. This started with a literature study and a study of used technologies by online banks.

Based on the knowledge gained through the exploration stage, the next step was an attempt to expand the currently available security options. The goal was not to design an authentication method that would offer maximum security. Instead, user-centered design was used as a cornerstone, without sacrificing the availability of adequate security against the man-in-the-middle threat.

From the exploration stage it was also learned that there are no mechanisms to evaluate online banking security methods and their usability. There are several frameworks to evaluate general web authentication methods, but these fit online banking poorly since the latter concerns itself not only with authentication of the user, but also authorization of data coming from the user. To accommodate the evaluation of methods found in the exploratory research and the method that was designed after that, two evaluation mechanisms were developed and tested.

1.4 Research overview

This section discusses the different parts of the research and this thesis, related to (usable) security in online banking. These parts are exploration, expansion and evaluation.

1.4.1 Exploring security offered by banks

The exploration part of the research consisted of examining which security methods were used to protect online banking worldwide. Authentication methods and implemented communications security of 80 banks were examined in 2013 and again in 2015. The result was the paper titled *A survey of authentication and communications security in online banking* [KSC⁺16]. Since the focus of the research was usable security, the scope was set to security methods that involve the user and/or the user's computer.

Different banks use different methods to verify the identity of their users. These can range from only using a single password or PIN to multi-factor authentication that involves some external component outside of the interaction between the user's computer and bank. Popular examples include physical cards that have one-time passwords printed as text, electronic devices that can generate one-time passwords and responses for challenge-response authentication, and the user's own phone to receive one-time passwords on through an out-of-band channel such as SMS text

messages. To counter illegitimate transactions that are added to a user's session through malware, several banks have implemented transaction authorization methods based on the What You See Is What You Sign information scheme. In this scheme, a bank sends received transaction information back to the user through a secure channel for verification. The user has to verify whether the received transaction information is correct.

Banks use SSL/TLS for communications security. When implemented correctly, this protocol suite provides confidentiality and integrity of the data flow between bank and client computer in addition to the authentication of the bank's identity. At the end of the observation period of the survey, all banks had SSL/TLS implementations that were good enough for daily use. Some implementations have vulnerabilities that could be exploitable when users use older browsers, but this can be considered an overall minor vulnerability. Man-in-the-middle attacks against SSL/TLS do not scale well due to the need for real-time intervention at the network connection between user and bank. Attacks such as these could be performed on a small scale, but large scale attacks would be unlikely since an attacker would have to interfere in real-time between connections of customers and banks. In addition, an updated browser would prevent these attacks.

1.4.2 Expanding on security not offered by banks

As discussed earlier, banks are introducing transaction authorization methods based on the What You See Is What You Sign (WYseeIWYS) transaction authorization scheme. This scheme gives users the option to securely verify transaction requests that banks receive from untrusted user-owned computers. A significant disadvantage in terms of usable security is that with WYseeIWYS, users can (accidentally or on purpose) process the verification incorrectly, or even skip it. That is because banks cannot force users to verify whether transactions were actually entered correctly.

Introducing What You Enter Is What You Sign

Part of the research was the proposal for What You Enter Is What You Sign (WYenterIWYS). WYenterIWYS is an alternative proposed transaction authorization information scheme that allows banks to securely verify the integrity and authenticity of requests to perform transactions that originate from the user. It aims to expand usable security compared to WYseeIWYS by removing the dependency on the user to verify transaction data after it is initially received by the bank. This is achieved by securing the authenticity and integrity of critical transaction data. The user enters the data in a secure environment, which adds a signature that provides authenticity and integrity as the data passes through untrusted user-owned devices and the Internet before it reaches the bank.

The WYenterIWYS information scheme was proposed in the paper titled *What You Enter Is What You Sign: Input Integrity in an Online Banking Environment* [KVvE14]. It introduced the information scheme by describing an information protocol between bank and user.

Refining What You Enter Is What You Sign

The idea to implement WYenterIWYS was only touched on briefly in the first paper. A suggestion was made to use keyboard emulation to transfer data between a secure device and the user's computer. First, the user would enter the data in an authentication device (representing the trusted environment). The device would then connect with the user's computer (the untrusted environment) and act as a keyboard to send both the data the user entered and the signature that warrants the data's authenticity and integrity. A problem with this, from functional and usability perspectives, is that there is not a single uniform keyboard interface supported by all user-owned computers which can be used for online banking. In addition, a connection between the secure device and the user's computer could open security vulnerabilities by exposing the trusted environment to attacks from the untrusted environment.

These issues and a proposal to solve them were discussed in a paper with the title *User-Friendly Manual Transfer Of Authenticated Online Banking Transaction Data* [KVvE16b]. Instead of relying on the user to make a connection between two devices, the proposal suggests that the user facilitates the transfer of data. The user enters the data in a separate authentication device as a first step. After this, the device generates a text string referred to as a Message Code (MC), which contains both the entered data and a signature to validate the authenticity and warranty. The user enters the MC in the computer used for online banking, which forwards it to the bank. Possessing the data and the signature, the bank is able to verify the message's integrity and authenticity without requiring assistance from the user.

1.4.3 Evaluating online banking authentication methods

Based on the exploration research that examined which security methods are offered by banks worldwide, the conclusion was made that a wide variation of authentication methods was implemented. In addition, the academic world also creates proposals for new authentication and transaction authorization methods [SFG09, AAJ10, WH11, LSH⁺12]. Another example is provided by the earlier mentioned proposal for the What You Enter Is What You Sign transaction authorization information scheme.

It can be useful to compare implemented and proposed methods with each other, and to test new proposals on their effectiveness and efficiency. For this thesis, two approaches were used to evaluate usable security in online banking. These approaches allow comparisons of implemented and proposed user authentication methods and transaction authorization methods on different levels. A theoretical approach quantifies qualitative characteristics and a practical approach focuses on user testing.

Theoretical evaluation

While most authentication methods are based on a handful of different information schemes (passwords/PINs, one-time passwords, challenge-response authentication, and What You See Is What You Sign), there is variation in implementations and proposals. For example, usable security variables that can make the simplicity of PINs more complex include whether they are initially chosen by users or randomly by the bank, whether users are offered the option to change their PINs, and whether

the bank forces users to change PINs after a certain amount of time. Each of these variables does something to the balance of security and usability when using PINs. Similar variables might not be applicable to other kinds of authentication schemes (such as one-time passwords). Therefore, direct comparison based on the offered usable security is difficult.

Renaud introduced an evaluation framework in 2004 that allowed the quantification of qualitative characteristics of web authentication methods [Ren04]. The characteristics are categorized in four dimensions that represent usability (the accessibility and memorability dimensions) and security (the security and vulnerability dimensions). Her framework adopts well to rate authentication methods on usability and security with the sole purpose of verifying the user's identity. However, it only takes usable security partially into account and does not examine some of the characteristics that are critical for processing transaction information in online banking.

Renaud's framework was extended with a 'feasibility' dimension, which examines aspects related to usable security. These aspects focus on the additional cognitive load that is put on the user due to the authentication method, and whether the user is able to circumvent its security willingly or unconsciously. The resulting paper was titled *Evaluation of transaction authentication methods for online banking* [KVvE16a].

Practical evaluation

(Secure) usability testing in online banking is often performed in an environment that does not represent the real world. This relates both to physical and digital environments. Physically, participants can be invited to visit a test center where the test will take place. Digitally, test environments are often solely tailored towards hosting the test. There are many factors in daily online banking. It is mostly an activity that overall is performed outside of banks and academic classrooms. Therefore, there are many distractions that can influence the participant's views and behavior, and therefore the usability and security of any tested method.

Digital testing environments are often made ad hoc in the academic world: they serve the purpose of the test, and are afterwards discarded. A researcher with a new idea often has to start from scratch by programming a brand new digital environment. This does not only include programming the explicit parts that need to be tested, but an entire web environment to support the tests. It is a waste of time when every researcher programs a similar environment that is discarded afterwards. This modus operandi does not support future enhancements by other researchers, since they too start from the beginning.

A web-based framework was envisioned which offers a virtual bank site that takes a modular approach to support different user authentication and transaction authorization methods. Such a framework could be adopted to easily test one's own ideas for transaction authentication. Measurements could both be objective (about what the candidate does) and subjective (what the candidate thinks), the former through the user's performance and actions, and the latter by offering surveys. It would have the advantage that it can be used over the Internet, allowing candidates to work with it at home or wherever else they practice online banking. This negates

the need for a physical test center, which adds several advantages to the legitimacy of the tests. The number of candidates is not limited to a geographical area or to a number of seats. Time is less of a constraint due to the absence of physical requirements, since candidates will have the option to perform the tests at any moment and at any location of their liking, similar to real online banking. Researchers will also be able to test aspects repeatedly and over a longer period of time.

To examine whether such a framework would be able to record and measure user data from which useful conclusions can be derived about the usable security of what is being tested, a proof of concept was developed to examine the accuracy of the What You See Is What You Sign transaction authorization scheme. Candidates were asked to perform transactions with the proof of concept and were either presented with a What You See Is What You Sign-based method, or they were part of a control group which did not have to apply this method. Some of the transactions were ‘attacked’ in order to see whether candidates were able to detect these attacks, which allowed measurements of the effectiveness of What You See Is What You Sign. Furthermore, time measurements were used to see how much time a candidate took to complete specific security actions.

The paper which proposed the framework and published the results of the proof of concept was titled *Towards A Virtual Bank For Evaluating Security Aspects With Focus On User Behavior* [KvEV16].

1.5 Origin of chapters and contribution

The main content of this thesis consists of three parts, based on themes related to the research of usable security in online banking: exploration, expansion and evaluation. Every part has one or more chapters. Each of these chapters is based on a paper that was written in the four year period of my research. I was the main author of every paper, and was responsible for performing the research and writing the papers. Harald Vranken and Marko van Eekelen are co-authors of all my papers. They provided valuable advice on how to best shape and present the research through written text, tables and figures in both papers and presentations.

Note that minor adjustments and corrections were made to the papers as they are represented in the chapters to provide a better integration with this thesis. Any major additions to a chapter are noted in the introduction of the part of the thesis that contains the relevant chapter.

A brief overview follows of the chapters and the papers they are based on.

Part I, Chapter 2 Sven Kiljan, Koen Simoens, Danny De Cock, Marko van Eekelen, and Harald Vranken. A survey of authentication and communications security in online banking. Published in ACM Computing Surveys, pages 61:1-61:35, February 2017. [KSC⁺16]

For the paper, Koen Simoens and Danny De Cock took up the role of co-authors. They gave useful input on which information to examine in an examination of 80 banks worldwide, and on how to present the resulting data. The paper is available in the ACM Digital Library.

Part II, Chapter 3 Sven Kiljan, Harald Vranken, and Marko van Eekelen. What You Enter Is What You Sign: Input Integrity in an Online Banking Environment. Published in Proceedings of the 4th International Workshop on Socio-Technical Aspects in Security and Trust (STAST), pages 40-47, July 2014. [KVvE14]
The work was presented by me at the 4th Workshop on Socio-Technical Aspects in Security and Trust (STAST 2014) in Vienna. The paper is available in the IEEE Xplore Digital Library.

Part II, Chapter 4 Sven Kiljan, Harald Vranken, and Marko van Eekelen. User-Friendly Manual Transfer of Authenticated Online Banking Transaction Data. Published in Proceedings of the 13th International Joint Conference on e-Business and Telecommunications, pages 259-270, July 2016. [KVvE16b]
The work was presented by me at the 13th International Conference on Security and Cryptography (SECRYPT 2016) in Lisbon, which was co-hosted with the 13th International Joint Conference on e-Business and Telecommunications. The paper is available in the SCITEPRESS Digital Library.

Part III, Chapter 5 Sven Kiljan, Harald Vranken, and Marko van Eekelen. Evaluation of transaction authentication methods for online banking. Accepted for publication by Elsevier Future Generation Computer Systems, 18 pages, 2016. [KVvE16a]
Volume, issue and page numbers have yet to be assigned as of the publication of this thesis. The paper is available in Elsevier ScienceDirect.

Part III, Chapter 6 Sven Kiljan, Harald Vranken, and Marko van Eekelen. Towards a virtual bank for evaluating security aspects with focus on user behavior. Published in Proceedings of the SAI Computing Conference, pages 1068-1075, July 2016. [KvEV16]
The work was presented by me at the SAI Computing Conference 2016 in London. The paper is available in the IEEE Xplore Digital Library.

1.6 Reading guide

The main content of the thesis is divided in parts that are identified by roman numerals. These parts represent themes of the research. Part I focuses on examination of usable security in online banking, Part II focuses on expansion and Part III focuses on evaluation. Each part has a number of chapters that are identified by decimal numbers, and each chapter is based on one published paper. This makes it easy to identify the papers on which the chapters are based. Note that the chapter titles and the contents of the chapters do not always match the contents of the papers exactly. Small changes were made to provide a more coherent flow between chapters and parts. Some larger changes were also made. These are noted in the introducing text of the corresponding part of the thesis.

All parts and their chapters can be read consecutively from the beginning, as depicted in Figure 1.1. Readers might find this comfortable if they want to read the full thesis since there is a natural flow from the beginning to the end. It starts with describing the state of security in online banking, continues with proposing a new

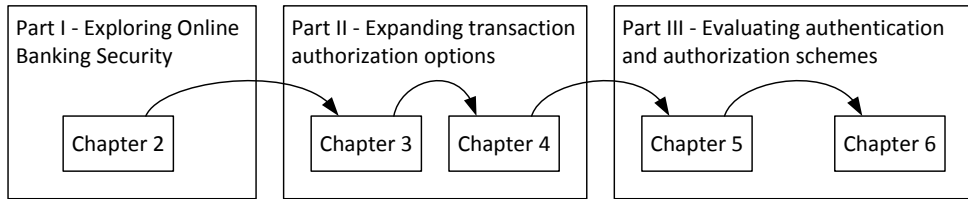


Figure 1.1: Conventional reading order. The thesis lends itself quite well to reading from front to back.

transaction authorization method and ends with evaluations of the proposed and other methods.

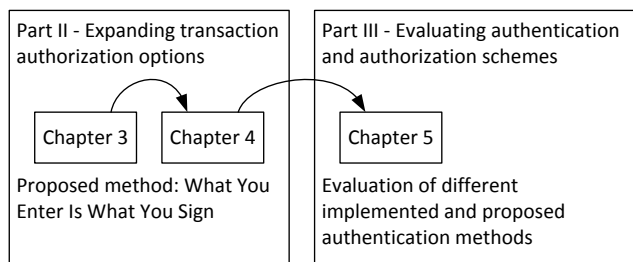


Figure 1.2: Reading order for those with a particular interest for proposed and implemented online banking authentication methods.

An alternative reading strategy based on interest is also possible, due to that each part and chapter can be read by itself. Readers interested in new proposals that attempt to make the interaction between users and banks more secure can follow the recommended reading order depicted in Figure 1.2.

For a more general picture of the security of online banking, Part I can be read. It sums up several years of development in this area and provides a good primer that can be read independently from other parts of the thesis.

If there is more of an interest in usability testing in general or specifically for usable security, Part III, Chapter 6 describes a proof of concept of online banking user testing through a site instead of through a testing center.

Part I

Exploring online banking security

Part I - Exploring online banking security

To make an addition to a field, that field must be explored first. Strengths and weaknesses determine where potential improvements can have the most impact. Before my research started, an exploration of the field was already made one decade earlier in 2002. Thirty banks worldwide were examined with a focus on communications security and client authentication [CDDC⁺02].

Sven Kiljan, Koen Simoens, Danny De Cock, Marko van Eekelen, and Harald Vranken. A survey of authentication and communications security in online banking. Published in ACM Computing Surveys, pages 61:1-61:35, February 2017. [KSC⁺16]

Ten years can be considered an eternity in the ever evolving information technology landscape, which is why a similar examination was done as the start of the research program. Assistance was received from Danny De Cock, one of the researchers which examined the thirty banks in 2002, and his colleague Koen Simoens. This resulted in a technical report in 2013 [KSDC⁺14], which contained information about communications security and client authentication methods of 81 banks worldwide. We made another survey over the same banks in 2015, to examine the evolution of online banking security between two points in time. Chapter 2 is based on the paper that came from this newer survey, and also compares the new results to the data from the technical report from 2013. Note that number of banks was reduced to 80 since two banks from the survey that was done in 2013 had merged, which explains the difference in the number of banks.

Chapter 2

A survey of authentication and communications security in online banking

Abstract

A survey was conducted to provide a state of the art of online banking authentication and communications security implementations. Between global regions the applied (single or multi-factor) authentication schemes differ greatly, as well as the security of SSL/TLS implementations. Three phases for online banking development are identified. It is predicted that mobile banking will enter a third phase, characterized by the use of standard web technologies to develop mobile banking applications for different platforms. This has the potential to make mobile banking a target for attacks in a similar manner that home banking currently is.

2.1 Introduction

An overview of the worldwide current state of online banking security was made in 2002 [CDDC⁺02]. It provided a state of the art which gave many researchers a base for their work. Since then, the adoption of online banking and the different ways to conduct it has changed quite a bit. We provide a new state of the art, based on a longer observation period between 2013 and 2015, and with a larger number of banks from different parts of the world. The scope of our work is authentication and communications security between banks and customers. These aspects were picked since they form a first line of defense against online banking fraud. Used information sources were the websites of banks and publicly available documentation.

In 2013, we examined 81 banks in on SSL/TLS use and offered customer authentication methods. At the time this was distributed as a technical report, not published as a paper [KSDC⁺14]. 80 of the same banks were examined again in the first half of 2015 (the number of banks has been reduced by one since two European

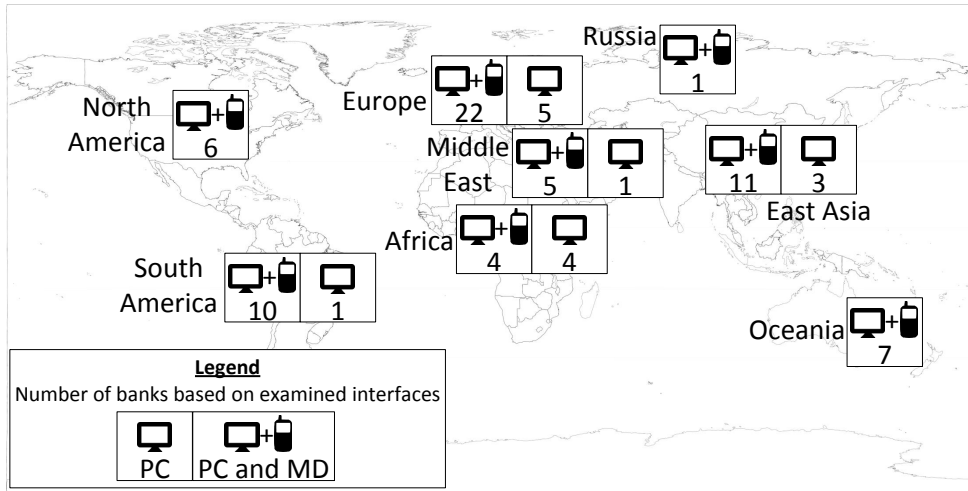


Figure 2.1: Overview of worldwide surveyed banks.

banks merged between 2013 and 2015). Search criteria for choosing the banks in 2013 were based on global representation and type of bank. The new data and the comparison with data from 2013 are included in this chapter. See Figure 2.1 for an overview of the global distribution of the 80 surveyed banks between North and South America, Europe, the Middle East, Africa, Russia, East Asia and Oceania.

The focus was on banks that provide account and payment services to consumers and small businesses (also referred to as ‘retail banks’). The assumption was made that retail banks with the most assets have the most customers. Having the most customers makes them represent how most people in their countries use online banking, and it also makes them a larger target for attacks conducted through customers. Random banks were picked for countries where their amounts of assets were more close to each other or unclear. Language barriers were overcome by colleagues who translated critical information and by automated translation tools. We only examined publicly available documentation and software.

Due to technical, security and usability differences between the different types of devices owned by banking customers, a distinction is made throughout this chapter between the use of personal computers (PCs) and mobile devices (MDs) for respectively home banking and mobile banking. All banks were examined on how they facilitate home banking, while mobile banking services were examined at 66 banks.⁹ Figure 2.1 makes a distinction between banks that were examined for the services they provide for both home and mobile banking, and for banks that were only examined for home banking.

The remainder of this chapter starts with Section 2.2, which opens with a short history of electronic banking and follows with an identified trend in the development of both home and mobile banking, based on both technological and sociological development.

The results of the conducted survey are split in two sections, based on the dir-

⁹We could not find information about any offered mobile banking services for the other 14 banks.

ection of the authentication process. In Section 2.3 (page 27) the wide range of methods to authenticate users to banks are discussed for both home and mobile banking. Banks use the SSL/TLS protocol suite to authenticate to their users in the opposite direction and to provide communications security. Findings about how well banks implement SSL/TLS can be found in Section 2.4 (page 41).

Section 2.5 (page 49) reflects on the differences in uniformity between communications security and user authentication, and the limitations of our survey. It also provides some pointers for possible further research, based on the current development of online banking and the knowledge gained from the survey. Related work is mentioned in Section 2.6 (page 51) and we give our concluding remarks in Section 2.7 (page 51).

The first major contribution of this chapter consists of a mapping of online banking development phases based on technological and sociological developments, as well as an insight which describes the future adoption of web standard-based Hybrid Mobile Applications and how this development has the potential to make mobile banking a future target for exploitation by adversaries. This can be found in Section 2.2. Our second major contribution is the survey data and analyses of data from different time points in Section 2.3 for user authentication methods in home and mobile banking, and Section 2.4 for communications security in home banking. Data is compared between 2013 and 2015. Additionally, for user authentication methods a comparison is made with data from 2002.

2.2 Online and mobile banking development

This section notes the history of electronic banking and an identified trend in the development of online and mobile banking platforms. Based on these observations, we note a number of security implications in the further development of mobile banking.

2.2.1 A short history of electronic banking

The electronic transfer of money was already offered in 1871 as a service.¹⁰ However, transmitting money orders through a telegraph relied on operators to translate between human language and Morse code. It would take almost 110 years before banking customers could do something similar themselves from the comfort of their homes. Starting from the beginning of the 1980s, it became possible for customers to simply call the bank and talk to an employee through a telephone instead of visiting a branch office.¹¹ Customers were capable of managing their account remotely. However, employing call operators to assist customers is expensive. Banks looked at ways to remove the bank employee from the process. The most obvious approach was to let customers interact with a bank computer.

Introduced in 1983, Pronto was the first electronic banking system which did not rely on bank employees to be used by customers. Access to the system was

¹⁰History of Western Union (2012), <https://www.westernunionbank.com/en/history/>

¹¹A history of innovation in payments (2012), <http://www.marketingweek.com/2012/11/28/a-history-of-innovation-in-payments/>

possible with various types of home computers and a modem.¹² Similar systems soon followed. An example is Citibank's Direct Access, which could be accessed through a Commodore 64 or a phone with an embedded terminal.¹³ These pioneers share a technical characteristic, which is that they all relied on proprietary terminal-based software. If a home computer was used, it was little more than a gateway to the user interface provided by the bank's computer.

This changed slowly at the end of the 80s and during the 90s. Having a continuous connection with a bank through a phone line to conduct banking business was expensive. To reduce costs (and with that, make it more accessible), so-called 'home banking software' was developed. Utilizing the increase in processing power, memory and storage space, bank customers could now conduct their banking business offline for the most part. Only a brief modem connection with the bank was necessary to receive up-to-date account information, such as an account's transaction history and balance, and to transmit money transactions.

The Internet became more accessible at the end of the 90s. Some banks updated their home banking software to support the use of the Internet to connect a customer's computer with the bank, instead of directly through a telephone line. Other banks saw potential in the World Wide Web (WWW), which offered a standardized way to present information to users and receive information from users through the Internet. A customer does not need any client-side software aside from a standard web browser when accessing a bank's website. When banks offer websites to conduct online banking, it saves them the effort of developing, maintaining and distributing platform-specific client-side software. Banks either had a website for online banking, or soon provided one after the turn of the millennium. Home banking software was slowly being discontinued, and by 2013 most banks only provide a website for online banking on home computers [KSDC⁺14].

Mobile banking is online banking through a mobile device in a way that is more location independent compared to home banking. It started with the Wireless Application Protocol (WAP) in the period in which home banking was becoming more mature.¹⁴ WAP can be described as a 'light' version of the World Wide Web and its underlying technologies. After it was introduced in 1997, banks started to offer mobile online banking services.¹⁵ The use of WAP can be compared to the use of terminal-based electronic banking which was done on home computers at the beginning of the 80s: it was revolutionary yet not user friendly, quite expensive and accessible only by a limited user base.

The mobile operating systems Android and iOS became popular to develop online banking applications for after 2010. Developing and publishing applications for both platforms is relatively easy, and most banks provide applications for these mobile operating systems [KSDC⁺14]. Mobile banking is also widely being embraced by customers since that time. The number of mobile banking users has been growing

¹²Pronto: Bank on Your Atari (1983), <http://www.atarimagazines.com/v1n6/pronto.html>

¹³Computer History - Citibank Direct Access and The Enhanced Telephone (2012), <http://www.kmoser.com/computerhistory/?id=citibank>

¹⁴We do note that SMS was used for mobile banking earlier and is still offered today by some banks, but elect to concentrate on online (Internet-based) banking instead.

¹⁵Verdens f  rste WAP-bank fra Norge (1999), <http://www.itavisen.no/nyheter/verdens-f%C3%B8rste-wap-bank-fra-norge-41812>

substantially in Belgium since 2013 and in the United States since 2011.^{16,17} The number of mobile banking logins surpassed the number of site logins in the United Kingdom in 2015.¹⁸ In volume of transactions, most are performed through mobile banking for the majority of banks worldwide, and an exponential increase in the number of users is expected for the period 2020-2025. The highest adoption rates are in developing countries (such as China and India).¹⁹ So far, the use of 'traditional' home banking (using personal computers) is not declining. Mobile banking seems to be used in addition to home banking, not as a replacement.^{20, 21}

Overall, online banking is very popular in different parts of the world. For 2014 in the United States, it was estimated that 74% of consumers with a bank account interacted with it through home banking while 35% did the same through mobile banking.²² For the same year in the Europe, the estimation is that 44% of individuals aged 16 to 74 used home and mobile banking.²³ This varies for individual countries in Europe. Bulgaria and Romania had relatively low numbers of persons using online banking (4-5%), while Iceland, Norway and Finland had relatively high numbers (all above 85%). Separate home and mobile banking statistics are not available for the whole of Europe, but some individual countries release such numbers. For Ireland it was reported that in 2014 home banking was used by 2.4 million users and mobile banking by 1.0 million users.²⁴ Based on the last population count of 2011²⁵, Ireland's population of individuals aged 15 to 74 was 3.37 million. Therefore, the relative population in Ireland that used home and mobile banking can be estimated to be respectively 71% and 29%. We can also make an estimation for China for 2013. Home banking was applied by 77.1% of the population that used the Internet, and the same value for mobile banking would be 44.6%.²⁶ The total number of Internet users in China at the end of 2013 was 618 million.²⁷ For the same year, the estimated population size of people aged 15 to 64 was 74% of the total population (estimated to be 1,357 million), so around 1,004 million.²⁸ When assumed that only people aged 15 to 64 would use the Internet, it can be estimated that 47% of the

¹⁶Cijfers - Succes internetbankieren (2015), <https://www.safeinternetbanking.be/nl/cijfers-internetbankieren>

¹⁷Consumers and Mobile Financial Services 2015 (2015): <http://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201503.pdf>

¹⁸The Way We Bank Now: World of change (2015), <https://www.bba.org.uk/publication/bba-reports/world-of-change-2/>

¹⁹KPMG Mobile Banking 2015 (2015), <http://www.kpmg.com/UK/en/IssuesAndInsights/ArticlesPublications/Documents/PDF/mobile-banking-report-2015.pdf>

²⁰See footnote 17.

²¹Online banking vs. mobile banking (2015), <http://www.bankrate.com/financing/mobile-finance/online-banking-vs-mobile-banking/>

²²See footnote 17.

²³Individuals using the internet for internet banking (2016), <http://ec.europa.eu/eurostat/tgm/table.do?tab=table&plugin=1&language=en&pcode=tin00099>

²⁴Online and Mobile Banking Report - Full Year 2014 and Q4 2014 (2015), <http://www.bpfi.ie/wp-content/uploads/2015/05/BPFI-Online-and-Mobile-Banking-Q4-2014-FINAL.pdf>

²⁵Population and Migration Estimates - April 2012 (2012), http://www.cso.ie/en/media/csoie/releasespublications/documents/population/2012/popmig_2012.pdf

²⁶Mobile Finance Becomes The Trend of Future Banking (2014), http://www.iresearchchina.com/content/details7_18315.html

²⁷Statistical Report on Internet Development in China (January 2014) (2014), <http://www1.cnnic.cn/IDR/ReportDownloads/201404/U020140417607531610855.pdf>

²⁸Data related to China (2016), <http://data.worldbank.org/country/china>

Chinese population aged 15 to 64 used home banking in 2013. For the same year and the same population, the estimation would be 27% for mobile banking.

2.2.2 Development and acceptance of online and mobile banking

Figure 2.2 illustrates when what kind of online services were offered by several banks in the United States and the Netherlands. This figure is used to show the similarities between the development of home banking and mobile banking. We chose these two countries because banks in the United States were among the early adopters to offer home banking services, while the same is true for the banks in the Netherlands for mobile banking.

Three technological phases for the development and use of home banking are identified: early adoption, expansion and exploitation. Some early adopters (both banks and customers in the United States) started with electronic banking using a terminal-based modem connection through a phone line. This evolved into intelligent client-side software which allowed connections with the bank either directly through a phone line or (later) through the Internet. It was this second phase that most banks in the United States and the Netherlands started to offer online banking

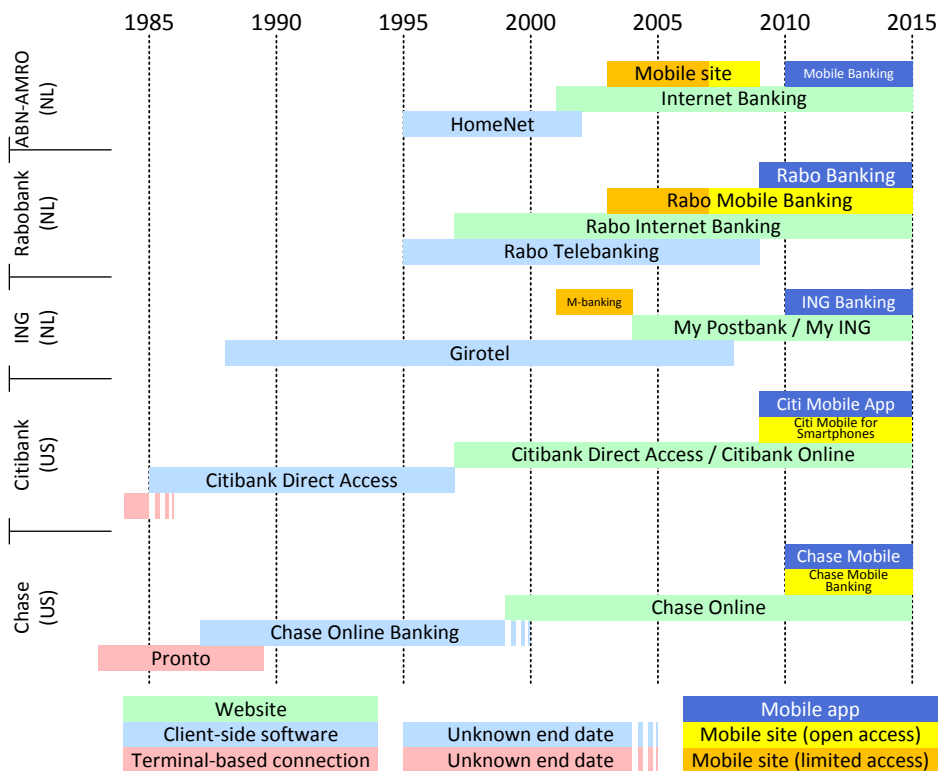


Figure 2.2: Some examples of the development of online banking from the Netherlands (NL) and the United States (US).

services that were picked up by the masses, which is why we named it expansion. All banks continued with the exploitation phase and broke off the second phase around the turn of the century. Sites were preferred above client-side code, and eventually were the only way to conduct home banking.

The technology acceptance model (TAM) can be applied to examine what motivates users to accept (intent to use) new technologies [DBW89]. TAM makes inferences for the conscious and unconscious acceptance and use of new technology based on perceived usefulness and perceived ease-of-use.

It is claimed that TAM can be applied to online banking [LL05a], and it has been applied with several extensions to the original model that include perceived credibility (trust) as a possible motivator [PPKP04, COLT10, KB12]. The overlapping conclusions from this research on home banking indicate that the most significant motivators are perceived usefulness and perceived credibility. Perceived ease of use has no direct significant effect on the intention to use online banking, although it can do so indirectly by affecting perceived usefulness.

For home banking in the early adoption phase, initially not many people had computers at home and hardware to make dial-up connections. The intent to use might have been there (if home banking would be perceived as useful and trustworthy at this time), but the ability to actually do so was simply missing. This changed in the expansion phase, in which more bank customers had access to computers at home and more banks started offering home banking. The idea that one could manage their bank affairs from the comfort of his or her home using a personal computer they already owned was considered useful and there was enough trust for users to intent to use it, which they did. Banks switched technologies in the exploitation phase from proprietary client-side code to open web standards by offering web sites accessible through popular browsers. Adoption rates did not stall since the use of a browser instead of a bank's own application did not hamper the perceived usefulness and perceived credibility by bank customers.

The development of the technological phases of mobile banking and motivators to use it are similar. Expectations were high²⁹, but the early adoption phase in the Netherlands was not successful.^{30,31} Early mobile sites used standardized technology for web page distribution (I-mode, WAP or HTTP), but these sites could only be reached if the mobile provider allowed access to the site and if the mobile phone supported the necessary security features to access the site. Therefore, as with the early adoption phase of home banking, not all potential bank customers were able to use mobile banking. The expansion phase began in 2007, when providers started offering more open Internet access and the introduction of affordable data subscriptions.^{32,33} All mobile sites were now easier to reach, including those of banks. It was also around this time that the mobile operating systems Apple iOS and Google

²⁹Het mobieltje van de Postbank (2001), <http://www.mt.nl/1/1727/home/het-mobieltje-van-postbank.html>

³⁰Postbank: 40 procent bankiert met mobiel toestel (2002), <http://www.emerce.nl/nieuws/postbank-40-procent-bankiert-met-mobiel-toestel>

³¹Rabobank ontevreden over gebruik mobiel bankieren (2006), <http://www.emerce.nl/nieuws/rabobank-ontevreden-over-gebruik-mobiel-bankieren>

³²Grote merken willen beter mobiel internet (2006), <http://www.emerce.nl/nieuws/grote-merken-willen-beter-mobiel-internet>

³³Mobiel internet groeide 30 procent in 2007 (2008), <http://www.emerce.nl/nieuws/mobiel-internet-groeide-30-procent-in-2007>

Android were introduced to the market, which both offer a very developer- and user-friendly eco-system. This made most banks release mobile applications around 2010, which were quickly accepted by their customers. Mobile banking is now offered by a large number of banks and its adoption by customers is steadily climbing.^{34,35,36}

The TAM with a perceived credibility extension has also been used to examine which perceptions contribute most to the acceptance of mobile banking. As with home banking, perceived usefulness and perceived credibility are large motivators to accept (use) mobile banking. However, perceived ease of use also has a direct influence on the intent to use mobile banking, unlike with home banking [LL05b, GLS09]. One explanation for this is that most mobile banking users use it in addition to home banking, instead of replacing the latter for the former.³⁷ The main (perceived) reasons to use mobile banking are the ability to access one's bank account from anywhere, that it saves time and that it can be used without either using a home computer or visiting a bank. This is in contrast to home banking, where the main reasons also include managing household finances and financial tasks without visiting a bank.³⁸ As mobile banking offers partly redundant functionality, perceived ease of use is also considered important. If it would not be (perceived to be) easy to use, users would be inclined to exclusively use home banking.

For mobile banking in its early adoption phase, the intent to use it was there for some part (as with home banking, due to the existing perceptions of usefulness and trustworthiness), but it was not perceived as being easy to use. Aesthetics play an important role in the adoption of mobile commerce, such as mobile banking [CHI06]. The displays of older phones did not have the capability to show an aesthetically pleasing user-interface due to low resolutions and (in very old phones) the absence of color, which likely influenced the intent to use mobile banking negatively. The expansion phase began at the moment smartphones were introduced. Touch controls likely influenced the ease of use perception positively and the increase in technical capabilities provided banks the opportunity to improve aesthetics.

2.2.3 Standardization: the pressure to go multi-platform

When looking at the development of home banking, the transfer from the early adoption phase to the expansion phase was driven by the need for more users to use home banking. This is different from the transfer of the expansion phase to the exploitation phase, which seems to be driven more by banks to reduce costs and increase client-side interoperability. The perceptions that have the most influence on embracing home banking were positive in the expansion phase, and stayed positive after the transfer to the exploitation phase.

For mobile banking, the changes between phases were quite similar. The changes made during the transfer from the early adoption phase to the expansion phase were also due to the need for an increase in user acceptance. Unlike home banking, it

³⁴Of the worldwide 80 banks examined in the survey, at least 66 offer mobile services.

³⁵Mobile Banking Deployment Widespread. Next Challenge: Adoption (2014), http://www.americanbanker.com/issues/179_209/1070929-1.html

³⁶See footnote 17.

³⁷See footnote 17.

³⁸Online banking vs. mobile banking (2015), <http://www.bankrate.com/financing/mobile-finance/online-banking-vs-mobile-banking/>

seems that an exploitation phase (characterized by the use of open (web) technologies) has not been reached yet for mobile banking. Browser-independent mobile banking sites actually exist, but they are not offered by as many banks as mobile client-side applications. While we do not know the exact reasons used to rationalize the choice to offer mobile banking applications over mobile sites, we understand that there are enough possible arguments from many different perspectives for why applications are preferred. A technical reason for this might be that applications have a better integration with the underlying operating system and hardware, through which banks can gain more information (such as data from sensors, biometrics, etc.) compared to mobile browsers. As noted earlier, aesthetics are an important factor in the acceptance of mobile banking. A functional reason might be that an application integrates visually better with the operating system, creating a more consistent user experience. A usable security reason might be that it is not required to provide a client-side application with information (e.g. a URL) to reach the bank, which is both user friendly and which reduces the risk of visiting a wrong (and possibly fraudulent) website. Of course, the risk still exists that a user accidentally installs a malicious application instead of code which legitimately comes from a bank, but that is a small risk that only applies the first time an online banking application is installed and used.

The current situation of mobile banking can be compared to the end of the expansion phase of home banking: there are only a limited number of software platforms used for online banking (home banking in the 90s: mostly DOS and Windows, mobile banking in 2015: mostly Android and iOS). At the time, banks slowly replaced their client-side applications with websites for online banking services, preferring the use of standardized technologies over custom client-side code. Ignoring the less often offered mobile banking sites, the current situation for mobile banking can be compared to the same time period for home banking. Mobile banking applications are currently written specifically for the most popular platforms of the moment (Android and iOS). There are several reasons for why this might change in the future.

The mobile landscape is still evolving. Standardization of technologies and device use seems to be a rising trend. Frameworks are available which allow developers to create Hybrid Mobile Applications (HMAs). An HMA is a mobile application of which the underlying code is largely written using web technologies, wrapped inside a native application which facilitates access to the mobile device's hardware, data sources and native looks.³⁹ Use of HMAs can become attractive for banks in the future for reasons related to cost. First of all, using HMAs can reduce the amount of client-specific code to maintain between the two current mobile platforms. Also, HMAs make it easier to support new mobile operating systems, due to that most code is platform-independent. Different parties are currently developing platforms which seamlessly integrate desktop and mobile work environments.^{40,41} Another promising aspect of HMAs is that they can reduce the overhead of developing graphical user interfaces for different platforms and screen sizes.

³⁹What is a Hybrid Mobile App? (2015), <http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>

⁴⁰Get the FAQs about Ubuntu on smartphones (2013), <https://insights.ubuntu.com/2013/02/15/get-the-faqs-about-ubuntu-on-smartphones/>

⁴¹Microsoft to bring back Start menu, windowed apps to Windows (2014), <http://www.zdnet.com/article/microsoft-to-bring-back-start-menu-windowed-apps-to-windows/>

Similar to the slow but steady move from client-side applications to websites for home banking in the exploitation phase, it can be expected that mobile banking will make a move from native client-side applications to a mix of native and web-based code, which is easier to maintain and more platform independent.

2.2.4 Security implications

The future use of standardized web technologies in mobile banking will likely be similar to that of home banking using browsers and websites, but not the same. The similarities and differences allow us to distill some implications.

What kept mobile banking relatively safe so far is a number of factors:

1. Mobile banking is not as popular as home banking.^{42,43,44} It is logical that malware is written for the most popular platform for online banking, since more users equals more possible fraud victims.
2. Home and mobile banking have an overlap in supported functions. If these functions are security critical (such as when transferring money), the mobile banking implementation is sometimes more limited compared to the home banking implementation by the same bank. Some banks in our survey only allow money transfers to previously used account numbers as destinations through mobile banking. Some others do allow first-time transfers to new accounts, but only with an extra authentication step or with a limit on the amount of money (which is sometimes adjustable by the user in the home banking environment).
3. Malware aimed at home banking can be written once and customized for each targeted bank site to allow browser injection and hijacking, a modus operandi known as Man-in-the-Browser [Eis10, CD12]. Malware kits are developed as an open platform to be customized by an adversary for a specific target audience [Oll08, AVW⁺12]. An example of such a malware kit is Zeus, which allows (silent) injection of data in a browser session on a Windows machine.⁴⁵ Such easy customization is currently not possible in the ecosystem of mobile platforms, since banks tend to write their own platform-specific code for each supported mobile operating system. Individual mobile banking application can be written in an insecure manner [FHM⁺12, GIJ⁺12, RSB⁺15], but these applications still have an inherent security advantage due to that the custom code base makes large-scale attacks on multiple banks difficult.

These factors slowly start to change in the evolving mobile landscape. It is claimed that the global number of mobile internet users surpassed the number of

⁴²Mobile Finance Becomes The Trend of Future Banking (2014), http://www.iresearchchina.com/content/details7_18315.html

⁴³See footnote 17.

⁴⁴Online and Mobile Banking Report - Full Year 2014 and Q4 2014 (2015), <http://www.bpfi.ie/wp-content/uploads/2015/05/BPFI-Online-and-Mobile-Banking-Q4-2014-FINAL.pdf>

⁴⁵Reversal and Analysis of Zeus and SpyEye Banking Trojans (2012), <http://www.ioactive.com/pdfs/ZeusSpyEyeBankingTrojanAnalysis.pdf>

traditional desktop Internet users in 2014,⁴⁶ the popularity of mobile banking is increasing⁴⁷ and its growth is expected to continue.⁴⁸ Banks have stated that customer loyalty is seen as critical in their mobile strategy. This might be seen as more important than security when it is considered that the latter has been neglected by a large number of banks during the development of their mobile applications.⁴⁹ The increase in popularity and the existence of vulnerabilities provide new fertile ground for adversaries, taking away the advantage which (1) provides.

Banks have been pushing their mobile services quite hard. When mobile banking replaces home banking further, banks could relax the restrictions placed on certain mobile banking functions. This could reduce the advantage that (2) brings. India provides an example where banks gained the freedom to do this. The government had a policy which stated that transactions initiated through mobile banking had an upper limit. This policy was removed to allow individual banks to set limits based on their own risk perception [Sri13]. In a market where mobile banking is becoming increasingly more popular,⁵⁰ banks could be urged to relax their limits.

As noted earlier in Section 2.2.3, banks will likely be motivated to move towards a largely shared code base, based on standard web technologies. Factor (3) will therefore change, since it will be possible for malware manufacturers to create malware kits which are easy to adjust to the mobile banking software of different banks which implement HMAs.

That personal computers by consumers are inherently insecure for home banking was noted in the expansion phase of home banking [RP98, SH04]. At the time, the notion that adoption was more important than security was accepted. Bank customers adopted a system that relies on an untrusted machine-in-the-middle, through which attacks are conducted to this day. Home banking communications rely on standard web technologies that make the alteration of the same attack to target different banks easy. Mobile banking is developing similarly to home banking. Adoption rates are high and a logical next step would be the reduction of operation costs, such as through shared code-bases offered by HMAs. This presents a new opportunity for attacks on mobile banking that scale well to large numbers of banks.

2.3 Customer to bank authentication

Customers must perform authentication to prove their identity to a bank before a session is initiated in which bank account(s) can be managed. This is referred to as entity authentication. Furthermore, it is possible that an extra authentication step is required to authorize the transfer of money. This is defined as transaction

⁴⁶Statistics on mobile usage and adoption to inform your mobile marketing strategy (2015), <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>

⁴⁷Consumers and Mobile Financial Services 2014 (2014), <http://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201403.pdf>

⁴⁸Online banking vs. mobile banking (2015), <http://www.bankrate.com/financing/mobile-finance/online-banking-vs-mobile-banking/>

⁴⁹Personal banking apps leak info through phone (2014), <http://blog.ioactive.com/2014/01/personal-banking-apps-leak-info-through.html>

⁵⁰Mobile banking zooms as India gets smarter (2014), http://www.business-standard.com/article/finance/mobile-banking-zooms-as-india-gets-smarter-114081100826_1.html

authentication. Entity authentication is mandatory while transaction authentication is optional to implement [CDDC⁺02].

Several factors can be used in user authentication. These are knowledge (something the user knows), possession (something the user physically has) and biometrics (something the user physically is or does). The terms two- or multi-factor authentication are used when at least two different factors need to be fulfilled to establish an authenticated session. Knowledge is mostly represented, followed by possession. Biometrics based on physical characteristics is rarely used, and was only observed in mobile banking.

We examined 80 home banking sites on the use of authentication methods for personal computers. The same was done for 60 mobile banking applications and 25 mobile banking sites. Not every bank which offers home banking offers mobile banking, which is why the numbers of the different types of examined online banking systems differ. Mobile banking applications seem to be far more popular compared to mobile banking sites, despite that the latter is more independent of the used platform. Also, for mobile banking we could not determine the used authentication methods for 2 applications and 1 site because we could not get the necessary information from the offered user interface or documentation. These are excluded from the 58 applications and 24 sites for which we could collect this information.

We also compare our findings with our research data from 2013. At the time, we examined 81 home banking sites, 45 mobile applications and 19 mobile sites. For one home banking site, it was in 2013 not possible to determine what kind of authentication method they used since a customer number had to be entered first before any information about authentication options were given [KSDC⁺14]. Therefore for user authentication, only 80 banks home banking sites are considered for 2013, the same number of home banking sites as which were examined in 2015.

First, we will present our findings concerning the combinations of factors (knowledge, possession, biometrics). After that, each factor will be discussed in more detail. We close with a comparison of data from 2002, 2013 and 2015.

2.3.1 Single or multi-factor?

Factors concern the difference in amount of resources required by users and by adversaries to use a system. Resources can be in the form of secret knowledge possessed by the user which nobody else is supposed to know, something the user has in his or her possession which is hard to duplicate, and something which only the user is or does and which can be measured using biometrics. The three factors (knowledge, possession and biometrics) can be combined to increase the amount of effort required by an adversary to commit successful identity fraud.

While multiple factors do provide protection against long-term credential stealing attacks, they are not the holy grail of information security. Multi-factor authentication does not protect against social (e.g. phishing) or various technical attacks (e.g. session hijacking/injection attacks) [Sch05]. There are also various technological, economical and usability limitations which delayed sector-wide acceptance of multi-factor authentication [HVOP09].

We will take a look at how factors are used and combined in home and mobile banking. More detailed information about individual authentication methods is

given in Section 2.3.2 for knowledge, Section 2.3.3 for possession and Section 2.3.4 for biometrics.

Home banking

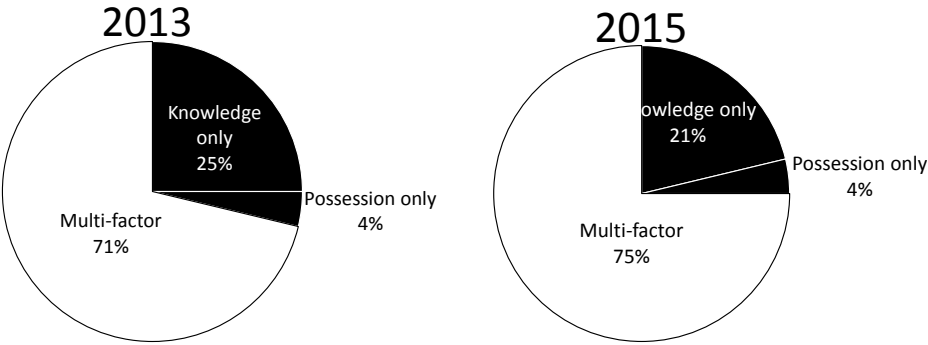


Figure 2.3: The use of multi-factor in home banking, 2013 and 2015 compared.

Figure 2.3 shows a comparison of the use of multi-factor authentication in 2013 and 2015. Not a lot has changed. Banks which applied multi-factor authentication still do so, and most banks which opted to use only knowledge have not gone back on this decision. Only a small number of banks which previously only relied on a password and/or PIN have changed their authentication methods to multi-factor.

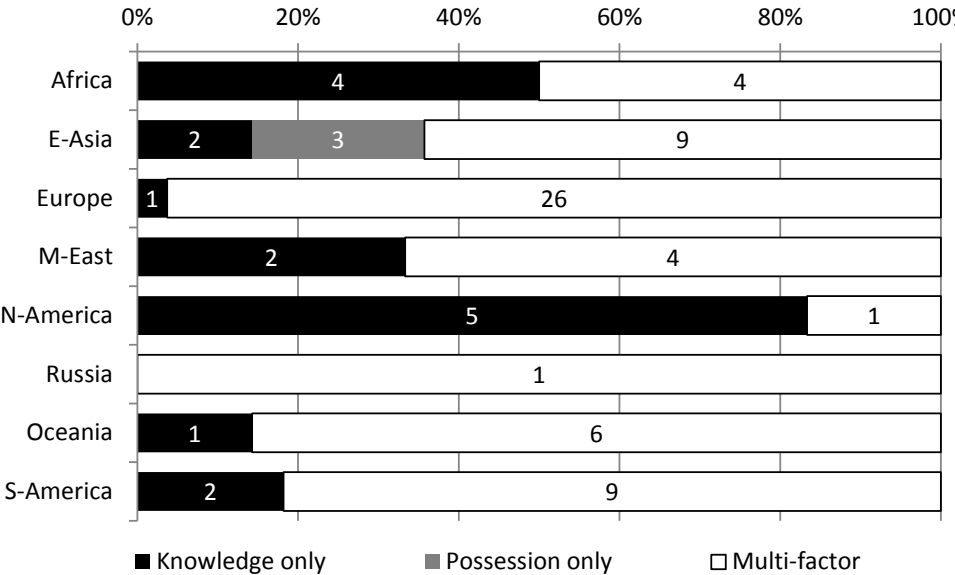


Figure 2.4: Authentication factors used in online banking on PCs in different regions in 2015.

A separation of authentication factors based on region is shown in Figure 2.4. Most banks in Europe, South America and Oceania require the use of multiple factors, while most other regions seem to be more divided.

Banks which offer multi-factor authentication are rare in North America compared to other parts of the world. An explanation for this could be that people in this region are reluctant to embrace the use of multiple factors to protect financial assets. For example, the United States has only recently started to widely implement smart card-based payment cards which require a PIN.⁵¹ Before that, shoppers were often able to electronically pay with a credit card using its magnetic stripe without using a PIN.

East Asia is quite exceptional, since some banks there allow users to login using only the possession factor. This is achieved by making users log in using digital certificates without any PIN or password validation. More information about the use of certificates can be found in Section 2.3.3.

Mobile banking

Figure 2.5 shows the use of knowledge and possession authentication factors by mobile banking applications and sites in 2013 and 2015.

We made categories for the possession factor since they vary greatly. These categories are as follows:

- Software indicates that supporting information for a possession factor is stored in the main memory of the mobile device which runs the banking application. Examples include a piece of information which binds the mobile device to a specific user's bank account(s) and the use of another mobile application which generates one-time passwords based on a secret key stored on the device itself.
- Online requires some kind of mobile connection to retrieve authentication credentials such as OTPs. The end-point of the connection on the receiving device used by the user acts as a possession factor, since it is assumed that adversaries do not have access to transferred information and that they do not have the ability to change the end-point of the connection.

Examples include the use of SMS text messages and of other mobile banking applications which receive an OTP from the bank using an internet connection. The use of an online connection can be almost the same as software if the mobile device used to receive the one-time password is also the same device used for mobile banking, such as with a smartphone. However, it can also be that a second mobile device is used to receive OTPs, such as with a tablet (to use the mobile banking application on) and a mobile phone to receive SMS text messages. Both options are supported and it is up to the user which is chosen.

- External means that the possession factor is a separate item issued by the bank to the user. Examples include physical paper or plastic cards from which OTPs

⁵¹Preparing for Chip-and-PIN Cards in the United States (2014), <http://bits.blogs.nytimes.com/2014/12/02/preparing-for-chip-and-pin-cards-in-the-united-states/>

can be read or derived, and electronic tokens (either stand-alone or using the user's bank card) which can generate OTPs.

More details about authentication methods from each category are given in Section 2.3.3.

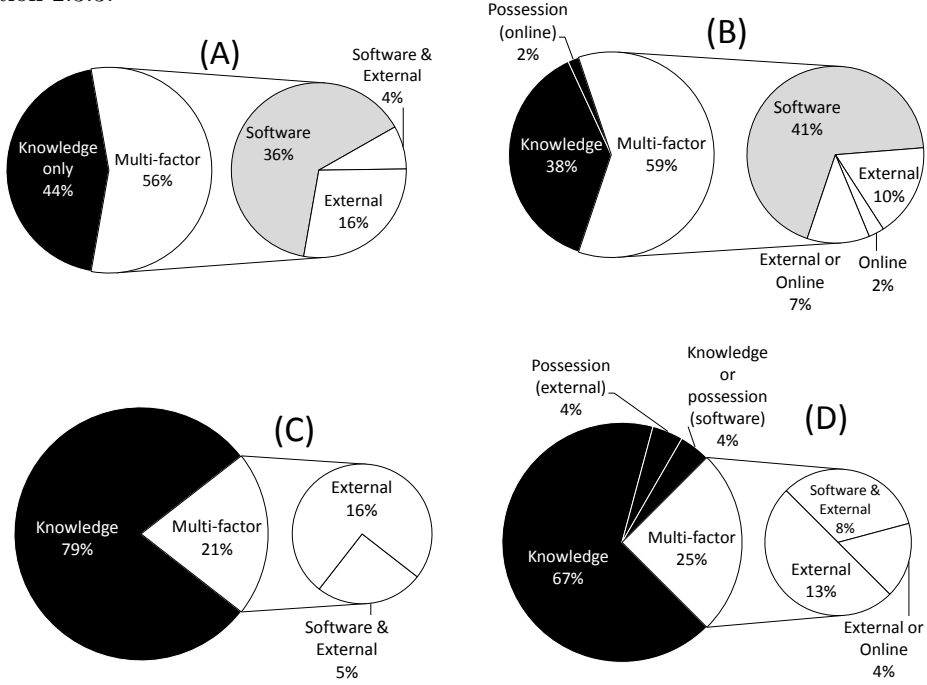


Figure 2.5: Knowledge and possession factor use in mobile banking. (A) Applications in 2013 (based on 56 banks examined banks), (B) applications in 2015 (58 banks), (C) sites in 2013 (22 banks) and (D) sites in 2015 (25 banks).

As Figure 2.5 shows, there is not much change in the overall use of knowledge and possession factors in both mobile applications and sites between 2013 and 2015. The only unusual sight is the introduction of possession only authentication in 2015 by a handful of banks. This concerns a single bank for mobile applications (using OTPs received by SMS) and a single bank for mobile sites (using OTPs from an external physical card).

For multi-factor authentication, the number of banks which apply an external possession factor stayed almost the same between 2013 and 2015. However, several banks started to offer an online possession factor as an alternative in 2015. A possible reason for this could be the inconvenience of carrying an additional device around, since the mobile device used for banking can also act as the possession factor. The online possession factor was not encountered in 2013, but was used compulsory or offered as a choice in 2015.

The use of a software possession factor stayed more or less the same. Similar to the online possession factor, the used mobile device for banking can also provide the possession factor if it is kept in software.

We noticed in 2013 a few mobile banking applications of which their use required both registration of the bank account to the user’s phone and an additional authentication device (accounting for ‘Software & External’ in the graph). This combination of different possession factors is something we did not encounter anymore in 2015 for applications. However, it was used by a few mobile banking sites in 2013 and 2015.

It is interesting to note that most banks which offer a mobile site and which apply multiple factors require the use of an external possession factor. This implies that banks trust the user’s device less for their mobile sites compared to their mobile applications. A possible explanation for this is that mobile sites are also usable on PCs with only a browser, which are less trusted by banks due to the higher risk of malware attacks.

Aside from knowledge and possession, there is also the biometrics factor (not shown in Figure 2.5. Its use was not observed in 2013, but in our survey for 2015 a few banks offer the use of a mobile device’s fingerprint scanner as an alternative authentication scheme. More detailed information about biometrics is given in Section 2.3.4.

2.3.2 Knowledge: Passwords and PINs

Text-based passwords and PINs were the only encountered implementations of knowledge-based factors. Other knowledge-based schemes have been proposed in the past, such as cognitive and graphical passwords [ZH90, SZO05], but none of them are used by the banks examined in the survey.

Using knowledge as a single factor for authentication is quite unsafe. Passwords and PINs are often kept static for longer periods of time to keep them memorizable. As long-term secrets, passwords and PINs entered as plain text on a user’s computer can be collected by software-based keyloggers, to be used instantly in subsequent attacks [MVO07]. Despite this vulnerability and as shown in Table 2.1, a relatively large number of banks still uses passwords or PINs exclusively in home and mobile banking (20% for home banking, 35% and 60% for respectively mobile banking applications and sites).

The ‘Password and/or PIN’ knowledge factor in Table 2.1 requires some explana-

Knowledge factor ↓	Possession factor					
	Regular sites (80)		Mobile apps (60)		Mobile sites (25)	
	None	Present	None	Present	None	Present
Password	14	32	16	14	13	4
PIN	2	17	5	20	2	3
Password and PIN	1	6	1	0	1	0
Password or PIN	0	3	0	1	0	0
Password and/or PIN	0	2	0	0	0	0
None	N/A	3	N/A	1	N/A	1
Unknown	0		2		1	

Table 2.1: Knowledge authentication factor use in online banking in 2015.

tion. It relates to banks which offer different combinations of authentication options that support either passwords, passwords and PINs or PINs only. This is because they offer different physical or electronic authentication devices, each with its own set of knowledge factors. For example, a bank can require a password to login, and either physical paper or an electronic device to derive one-time passwords from for transaction authorization. The electronic device requires a PIN to be accessible while a PIN is not necessary for the physical paper.

Passwords are popular in both situations where knowledge is used as a single factor and when a possession factor is used. PINs are only popular in combination with a possession factor. If the only factor is knowledge it is logical that passwords are preferred above PINs, since passwords offer more security due to their higher complexity, making them harder to guess. An explanation for why PINs are still quite popular in multi-factor scenarios is that they are often an intrinsic part of the authentication method. For instance, some OTP generators require the use of some knowledge to unlock their functionality. If the knowledge has to be provided on the (often relatively small) device itself, a PIN would be the most practical way since its entry requires less buttons and button presses compared to a password.

Some banks provide additional proprietary software to detect or protect against passive password or PIN sniffing attacks against home banking. The use of this software is mandatory at 8 banks and optional at 1 bank. We did not study this software in-depth on how passwords are protected, but it is implied by documentation that some possible offered features include a scanner for malware-based sniffers and an overlay for password and PIN fields which offers a randomized keyboard to be used with a pointer device, such as a mouse.

Another security enhancing feature is the use of an on-screen keyboard with randomly placed buttons, offered for home banking on a bank site and not through software. Passive sniffing of keyboard and mouse data will not gain passwords or PINs in an attack if this feature is used. 2 banks offer this through their sites for password entry and 1 bank offers this for PIN entry.

In addition to using a password, one bank implemented a system which relies on questions answerable by the user. Upon registration, the user creates three pairs of questions and answers. Whenever the user wants to log in, the bank asks for the password and one of the user-chosen questions. The user has to enter specific letters (chosen by the bank) of the answer and not the entire answer. This ensures that all secret knowledge cannot be gained in a single password sniffing attack. However, it does not protect against long-term repeated passive attacks, nor against active and social engineering attacks.

2.3.3 Possession: Physical and digital

The possession factor is often used in multi-factor authentication but rarely as a single-factor. Exact numbers of banks which apply possession as a single factor are shown in Table 2.1 (under the knowledge factor label ‘None’ and possession factor label ‘Present’).

There are many different types of possession factors banks can accept. We note the different types based on the earlier made separation between software, online and external possession factors.

Software

Software as a possession factor uses information to present some kind of proof that the user is in possession of said information. The information is stored and processed on a device owned by the user.

There are two authentication methods which use this in home banking:

- Software certificates

If the private key of a certificate is not stored in a secure hardware device, it must be stored and processed in software. 5 banks (all in the East Asian territory) apply this. Of these, 4 use proprietary software in the form of browser plugins for key management and signature handling. A single bank relies on the browser's own certificate management system. Potentially, users could also use a hardware device for this single bank if it is supported by the browser. Of the 4 banks, one also optionally supports proprietary hardware devices for key storage through the provided software.

The popularity of this software-based possession factor is slowly decreasing. In 2013, 7 banks in East Asia applied software certificates, of which 4 also offered hardware certificates as an option.

- File-based

One observed method is where a bank provides a file to be stored on a user's home computer. Whenever the user initiates an action that requires extra security (such as the transfer of money to a third party), the site requests the file. If the file is provided by the user, the operation is permitted and a new file is provided for the next action. The file can be stored on the user's computer itself or on removable media. A single bank used this in 2013 and still does so in 2015.

Software as a possession factor is not popular in home banking. However, the use of data stored in software as a possession factor is the most applied type of possession factor in mobile banking applications, as can be seen in Figure 2.5 (page 31). 24 of the 58 mobile banking applications (41.4%) for which we were able to document authentication factors in 2015 use software as a possession factor in a multi-factor authentication scheme. To be able to use mobile banking with one of these applications, a user must register his/her bank account through the application and bind its use to the mobile device. Once registered, the application only works for that specific user's account at the bank. The registered mobile device represents the possession factor. We did not analyze the internal workings of the applications, but it can be assumed that registration of a user's bank account on a mobile device results in a possession factor based on one or more identifiers from that device.

Software possession factors are stored and processed on user-owned devices. We marked the exclusive use of software factors gray in Figure 2.5 since using these introduces a security risk. The possession factor is represented by the mobile device itself in an untrusted digital environment, which makes it possible for an adversary to copy the possession factor in a malware attack. Combined with retrieval of the knowledge factor, every used authentication factor can be retrieved from the same mobile device in a single attack.

Online

Online is a type of possession factor where the end-point of a network connection represents the possession factor. Every method which applies this relies on the ability to authenticate or receive messages through an online connection.

For home banking on PCs, the most popular online possession factor is the use of SMS text messages to send OTPs to the user, used by 21 out of 80 banks (26.3%) in 2013 and 25 of the same 80 banks (31.3%) in 2015. In this case, SMS uses an out-of-band channel and the user's mobile phone represents the possession factor. The use of SMS to provide a user with OTPs is also proposed in literature [AZE09, HPN10, WH11]. A less popular variant is the use of a mobile application to receive OTPs from the bank through the Internet. We did not encounter this in 2013, but 11 of the 80 banks (13.8%) offered this as an alternative part of their authentication methods in 2015. We also encountered a rare variant which uses email instead of SMS to send transaction data and an OTP to the user, only observed in 2015 and used by 2 out of 80 banks (2.5%) for home banking authentication.

Home banking can alternatively use other connected devices as a possession factor to authenticate to the bank. Two types of such devices can be distinguished:

- **Hardware certificates**

Similar to software certificates, this can be used for signature-based authentication. The only difference is that secret key material is stored in an external device, protecting it against authentication credential stealing through malware attacks. With hardware certificates, the device which has the secret keys represents the possession factor. 7 out of 80 banks (8.8%) use hardware certificates to let their users authenticate in 2015, of which 4 require the use of proprietary software to support the hardware. 6 out of 80 banks (7.5%) supported hardware certificates in 2013. We did not encounter the use of hardware certificates in mobile banking in 2013 or 2015.

- **Connected hardware tokens**

A significant difference between hardware certificates and connected hardware tokens is the amount of user interaction with the device. Hardware certificates are only connected by the user to the computer used for home banking. Connected hardware tokens expect more user interaction with the device itself, such as PIN entry and information verification. 3 of 80 banks (3.8%) from our 2015 survey offer connected hardware tokens for home banking (all connected through USB). Of these, 2 banks use them to make the user verify transaction data and protect these devices with a PIN, while the last one only lets a user verify the entered name and account number of new beneficiaries, without the necessity to enter a PIN on the device itself.

A rare example of a connected hardware token used in mobile banking has been observed in our 2015 survey. A small token can use a two-way connection using near-field communication (NFC) to communicate with a mobile banking application on the phone. Critical transaction details can be verified and accepted or rejected by the user on the token.

A few other possession factors which are classified as online have also been observed in our 2015 survey for mobile banking. 4 out of 58 mobile bank applications

(6.9%) use OTPs received by SMS on either the same or another mobile device. A single bank does something similar, but relies on a mobile application to receive the OTP. Another bank uses challenge-response authentication using QR-like codes, which are shown on the first mobile device (used for mobile banking) to be scanned by a second mobile device. The entered transactions and a response code are shown on the screen of the second mobile device. The user is expected to verify the transactions and enter the response code for confirmation.⁵²

Depending on the kind of authentication method used, an online possession factor can be as insecure as a software possession factor or as secure as an external possession factor. It is insecure if the factor is effectively represented by the user's mobile device, since its untrusted environment is vulnerable to malware. It is also insecure if the transportation channel of information cannot be trusted, such as with SMS (which is vulnerable to SIM swap scams.⁵³ However, as a connected external device, an authentication method can be secure if it offers a trusted environment separate from the untrusted environment of the user's mobile device, and if it does not rely on the security of the communication channel between an untrusted device and the bank.

External

Some trusted devices have earlier been described under 'Online'. These concern devices which rely on a network connection (hence the name). An 'External' possession factor relies on a bank-issued authentication device which does not use electronic connections with other devices. There are a few variations which are either low-tech or high-tech.

- OTPs on paper/plastic

This is the simplest form of a possession factor. It consists of an indexed list of OTPs on paper or an indexed grid of characters on a small plastic card. A user derives an OTP from one of these when the bank requests it. The bank specifies which OTP it wants by referring to one or more index numbers. The physical paper or plastic represents the possession factor. Advantages are that it is easy to use and that it is protected against malware-based attacks, like all external possession factors. A disadvantage is that a physical medium with written text is easy to copy. A picture of the page or card made by a camera already represents a copy of the possession factor which is usable by an adversary.

We observed that 16 out of 80 banks (20%) let users authenticate with OTPs from paper or plastic in 2015. Of these 16 banks, 5 (31.3%) do not give the user an alternative choice for the possession factor. Most of the 16 banks are located in Europe and South-America, where this method seems to be more popular compared to other regions. Paper and plastic OTPs have become more popular since 2013, when only 13 banks (16.3%) applied it for home banking. At that time, 8 of the 13 banks (61.5%) required the use of a physical page

⁵²We did not examine these secondary mobile banking application (one which generates an OTP, one which scans QR-like codes) on a technical level and assume that they require an online connection to receive OTPs or response codes.

⁵³SIM swap scam (2011), *SIMswapscam*

or card to get OTPs from and an alternative was not available. This implies that this representation of the possession factor became more popular as an alternative authentication scheme instead of as the only (mandatory) option.

OTPs from a physical medium are used for authentication in 6 of 58 examined mobile applications (10.3%), and in 4 of 24 mobile sites (16.7%) in 2015. The same numbers for 2013 were respectively 3 out of 45 (6.7%) and 0 out of 19 (0%).

- Offline electronic tokens

We added the ‘offline’ keyword to the description of these kind of tokens to distinguish them from online hardware tokens. These tokens do not have electronic connections with any other device, but rely on their own battery as a power source and non-electronic methods for information transfer. There are different types of tokens, ranging in functionality and offered user interface.

The simplest token consists of a single button and a small display. When the button is pushed, the display shows a single OTP. 8 out of 80 (10%) observed home banking sites applied such a token in 2015 and 7 out of the same number of observed banks (8.8%) did so for home banking in 2013.

A slightly more complex token consists of a display, a number of function buttons and possibly a keypad. These tokens work stand-alone or rely on an inserted bank card to provide cryptographic credentials. The functions of some of these tokens are only usable after it is unlocked by a PIN (associated either with the device itself or with a smart card). There are several functions which can be supported by different kinds of tokens:

- Generate OTPs. Like the one-button tokens, OTPs can be generated after entering a PIN. Offering the OTP to the bank proves that the user is in possession of the device used to create the OTP and (indirectly) of the PIN required to operate the device.
- Generate responses for challenge-response (CR) authentication. After entering the PIN, the user must enter a challenge (given by the bank), after which the token will generate a response for the user to enter in the online banking site. Receiving the expected response to the sent challenge is an indication for the bank that the user is in possession of what is needed to generate the response (a specific token or bank card and a PIN).
- Show critical transaction information and confirmation codes. The information is received through a non-electronic one-way connection between the token and the user’s device. We only observed this new authentication method in our 2015 survey. The one-way information transfer is facilitated by an optical sensor, which scans QR-like codes from the monitor of a user’s device.

Table 2.2 provides an overview of the types and numbers of offline electronic tokens we encountered in our 2015 survey.

Device(s) and optional knowledge factor	Authentication method			
	OTP	CR	OTP & CR	WYseeIWYS
Stand-alone token	8 (1)	0	0	2
Stand-alone token with PIN	10	0	3	0
Smart card, token and PIN	1	2	5 (1)	1

Table 2.2: Offline tokens used for home banking at 31 out of 80 banks (38.8%). One bank implemented two different kinds of devices, resulting in a total value of 32. Numbers in parenthesis represent banks from the same group which also use tokens for mobile banking.

Region	Home banking	Mobile banking	
		App	Site
Africa	(8) Offline electronic tokens (OTP)	(3) None	(1) None
Asia	(14) Hardware & Software Certificates	(8) Software	(6) None, physical tokens (OTP)
Europe	(27) Offl. elect. tokens (OTP, CR, WYseeIWYS)	(22) Software	(7) Mixed OTP
M-East	(6) SMS (OTP)	(5) None	(0) N/A
N-America	(6) None	(5) None	(6) None
Oceania	(7) Offl. electr. tokens (OTP)	(7) Software	(3) Offline electronic token (OTP)
Russia	(1) Mixed (OTP)	(1) Software	(0) N/A
S-America	(11) Physical tokens (OTP)	(9) Software	(2) None

Table 2.3: Most applied possession factor in different regions and online banking environments.

Most often applied possession factors by region

There are many different possession factors employed in the online banking world. Table 2.3 shows for each region which possession factors are most often applied according to the survey data. A value of ‘none’ indicates that most banks do not prefer to use a possession factor at all. The numbers of observed banks are included for reference.

2.3.4 Biometrics and behavior anomaly detection

Biometrics is also known as the inherence factor in user authentication. Unlike the other factors, it does not concern something that the user should know or have. Instead, this factor focuses on what the user is or does to ensure the user’s identity. Physical biometrics measure the presence of physical characteristics of the user. Absence of such physical characteristics can be considered suspicious and a reason for the system to ask for alternative authentication credentials. Behavior anomaly detection concerns the use of user behavior data to, after a user action has been taken, detect deviations from a previously established baseline. Therefore, afterwards it can be said whether a user behaved as expected or not. Abnormal user behavior can be an indication of identity fraud, where an adversary has (direct or indirect) access to all authentication credentials.

Physical biometrics

Biometrics based on physical characteristics can be used as an additional or alternative authentication factor for user authentication. An advantage it has is that it generally is quite usable. Disadvantages include the unwillingness of some people to

use them due to social stigmas and the limited number of non-replaceable characteristics (which can be zero for users with disabilities). These disadvantages limit biometrics for user authentication to users who want and can use them, which is why an alternative authentication method (based on the other two factors) has to be available.

Physical characteristics were not used for user authentication by any of the surveyed banks in 2013. Registering these characteristics requires specialized sensors. These sensors were not widely integrated in user equipment at the time. While banks could opt to distribute needed equipment, it would be very expensive to support an authentication method for which an alternative must always be available.

The use of physical characteristics has been observed in our 2015 survey for a limited number of mobile banking applications. Two banks support Apple TouchID, which consists of a fingerprint sensor, operating system support and an application programming interface. The fingerprint is used as an alternative to providing a PIN, while the applied possession factor remains the same. Enrollment is performed by the operating system and not by the mobile banking application.

More banks have indicated that they will support Apple TouchID in the future, despite that it is possible to spoof fingerprints.^{54,55}

Detection of behavior anomalies

Behavior anomaly detection monitors the user's behavior to detect whether a transaction is possibly made by a fraudulent party. The identity of the user can only be ascertained after the user's behavioral patterns have been registered and compared to a past baseline. Since the user is required to do something before there is some certainty of the user's identity, anomaly detection based on behavior is unsuitable for user authentication (at the beginning of a session, before a user has performed any actions), but can be used to provide some certainty about the user's identity and the validity of a user's action afterwards. The origin of the data can come from user actions (such as at what time of the day a user performs an action and the user's data entry speed) and from the environment in which the user's device operates (such as the geographical location and local temperature). The used methods can be compared to those of data leakage detection systems used to spot anomalies in data transactions [CdP⁺14]. Several examples of implemented fraud detection techniques are given in literature [KLSH04, QS07].

Since user behavior anomalies are registered by the back-end technical infrastructure of banks, it cannot be said with full certainty how many of the banks in our 2013 and 2015 surveys apply this and to what extent. However, some banks state that they do use monitoring services for financial transactions.^{56,57,58} It has also been

⁵⁴Banks to allow account access using fingerprint tech (2015), <http://www.bbc.com/news/technology-31508932>

⁵⁵Chaos Computer Club breaks Apple TouchID (2013), <http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid>

⁵⁶How We Protect You (2010), <http://www.ally.com/security/>

⁵⁷Online Banking Security from Bank of America (2013), <https://www.bankofamerica.com/privacy/online-mobile-banking-privacy/online-banking-security.go>

⁵⁸What we're doing to protect your account (2014), <http://www.barclays.co.uk/Helpsupport/Whatweredoingtoprotectyou/P1242560037946#Fraudmonitoring>

claimed that some banks profile low-level user actions through mobile applications.⁵⁹

2.3.5 Comparing 2002, 2013 and 2015

Table 2.4 gives an overview of observed basic authentication methods used for home banking in 2002 by Claessens et al. in 2002 [CDDC⁺02] and by us in 2013 and 2015. Check marks indicate that a method was observed and percentages (if the required information is available to produce them) indicate the relative number of banks that apply the methods in the survey for a specific year. Note that all methods except ‘Password/PIN-only’ are not exclusive. For example, a bank that applies OTP authentication in any way can still require an additional password or PIN from the user. A bank that applies OTP authentication using a bank-issued device for entity authentication can also apply challenge-response authentication for transaction authentication.

Method	2002	2013	2015
Password/PIN-only	✓ (66.7%)	✓ (23.8% + 13.8%)	✓ (21.3% + 15%)
OTP (paper/plastic)	✓ (6.7%)	✓ (16.3%)	✓ (20%)
OTP (offline electronic tokens)	✓	✓ (13.8%)	✓ (33.8%)
OTP (SMS)		✓ (26.3%)	✓ (31.2%)
Challenge-response (offline electronic tokens)	✓	✓ (13.8%)	✓ (12.5%)
Certificate-based	✓ (14.3%)	✓ (3.8% + 2.5% + 5%)	✓ (3.8% + 6.3% + 2.5%)

Table 2.4: Overview of similarities and differences in basic authentication methods for online banking using a PC as observed in 2002, 2013 and 2015. Percentages represent the relative amount of observed banks that apply a method in a specific year. Observations can be compared between years (columns) but not between methods (rows) because some banks in our survey are counted multiple times when they offer multiple methods (which is why the percentages can exceed 100% when summed).

For Password/PIN-only, two percentages are given for our work. These stand respectively for the percentage of banks that offer Password/PIN-only based authentication without and with a multi-factor based alternative. For certificate-based authentication, our percentages relate respectively to the relative number of banks that offer this type of authentication with the private key stored in software only, in hardware only and for the banks which give this as a choice to the user (by supporting both hard- and software storage of the private key).

We excluded the exceptional methods found in our survey to keep the table easy to read. While the often used basic authentication mechanisms show little difference, the ratios in which they are implemented differ significantly between 2002 and 2013 and moderately between 2013 and 2015. Claessens et al. (2002) [CDDC⁺02] conclude that in their survey passwords and PINs as a single factor were most widely used to authenticate users. Today, most banks offer multi-factor authentication. Since 2002 a large number of online banks have migrated to methods which are safer but also have been available for quite some time. The only method

⁵⁹Don’t Be Afraid of Mobile Banking Apps (2012), <http://www.banktech.com/channels/dont-be-afraid-of-mobile-banking-apps/a/d-id/1295727>

that is popular now that was not discussed by Claessens et al. in 2002 is the use of text messages to send OTPs to users.

2.4 Bank to customer authentication and communications security

This section describes the observations from our surveys concerning authentication by the bank to the user. Data from 2013 and 2015 is compared.

In 2002, the standard solutions for communications security with online banking was Secure Sockets Layer/Transport Layer Security (SSL/TLS) for PCs and Wireless Transport Layer Security (WTLS) for MDs. Claessens et al. [2002] describe the various versions of SSL/TLS up to SSL 3.0 and TLS 1.0.^{60,61} Since then, several weaknesses in both SSL/TLS standards and implementations have been discovered, and TLS 1.1 and 1.2 have been developed.⁶² (2006), <http://tools.ietf.org/html/rfc4346>.⁶³ An update was also released for all TLS versions that breaks backwards compatibility of TLS with SSL 2.0.⁶⁴ WTLS has not seen a newer version since 2001⁶⁵ which is most likely caused by the decline of WAP in favor of SSL/TLS on MDs.

The use of SSL/TLS by home banking sites was examined in the survey. Due to the large number of banks, the analysis was narrowed down to the use of SSL/TLS to secure communication between home banking sites and web browsers (HTTPS). The use of SSL/TLS in mobile banking was not examined due to it requiring a more specialized approach that takes more time. Examples of research in SSL/TLS use by mobile applications, including bank applications, are given in Section 2.6. We did not examine SSL/TLS for mobile banking sites since most sites are hosted by the same server or SSL/TLS front-end, which would provide the same results as the examined regular sites. Also, SSL/TLS as possibly used by other services hosted by the bank (such as email and VPN for their employers) was not examined since these services are not meant for customer-bank interaction. From a technical perspective, the information required to connect with such services are time-consuming to find and not all banks will offer such services uniformly, making useful comparisons harder.

The authors who worked on the paper on which this chapter is based did not have accounts at most banks at the time the research was conducted, which is why SSL/TLS-usage was examined using login pages. All 80 surveyed online banking sites rely on SSL/TLS for both server authentication and secure communication. Used cryptographic algorithms, vulnerabilities and optional TLS functions were examined.

In an earlier technical report [KSDC⁺14], bank sites were examined using Qualys

⁶⁰RFC 6101 - The Secure Sockets Layer (SSL) Protocol Version 3.0 (2011), <http://tools.ietf.org/html/rfc6101>

⁶¹RFC 2246 - The TLS Protocol Version 1.0 (1999), <http://tools.ietf.org/html/rfc2246>

⁶²RFC 4346 - The Transport Layer Security (TLS) Protocol Version 1.1

⁶³RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 (2008), <http://tools.ietf.org/html/rfc5246>

⁶⁴RFC 6176 - Prohibiting Secure Sockets Layer (SSL) Version 2.0 (2011), <http://tools.ietf.org/html/rfc6176>

⁶⁵Wireless Transport Layer Security - Version 06-Apr-2001 (2001), <http://technical.openmobilealliance.org/tech/affiliates/wap/wap-261-wtls-20010406-a.pdf>

SSL Labs SSL Server Test.⁶⁶ The test by Qualys was chosen since it was the most expanded test available online. Alternative approaches were considered, such as using self-hosted security and vulnerability scanners. An example is Nmap⁶⁷, which has the potency to provide more information about scanned sites, but a disadvantage is that its scans can be quite intrusive. Whereas Qualys uses standard site requests (like a browser would when setting up a connection) and analyzes the responses, other scanners have capabilities to scan deeper by sending non-standard requests that could be interpreted as malicious, which in some countries could result in legal issues. The intrusive scans might be disabled, but then the retrieved relevant information would be the same or less as what Qualys collects. It was also considered to forego scanning entirely and instead use data collected by the Internet-Wide Scan Data Repository to scan all possible IPv4 hosts, which uses ZMap.^{68,69} Unfortunately, the data provided by this repository is wide but shallow. Data is only collected based on the connection the ZMap scanner negotiates. Unlike Qualys, it does not make an attempt to find out which weaker versions of SSL/TLS are supported, whether an vulnerabilities are present and whether the site supports additional TLS functions that can improve security. Therefore, Qualys' scanner was used instead.

2.4.1 Vulnerabilities

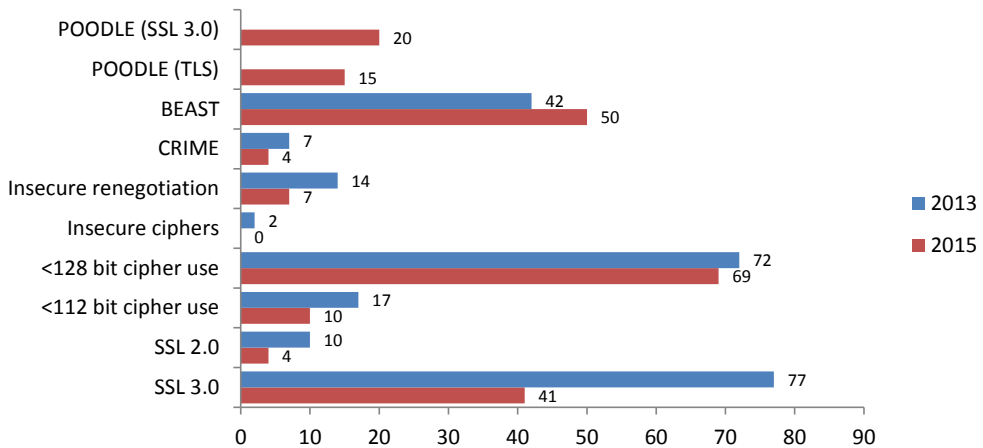


Figure 2.6: An overview of the encountered SSL/TLS vulnerabilities.

Figure 2.6 shows an overview of the vulnerabilities and how often we encountered them among the 80 surveyed banks. Each vulnerability is discussed briefly.

At the end of 2014, a successful attack was made against SSL 3.0 and TLS 1.0 when block ciphers are used. An adversary manipulates a user's browser to send requests to a site using SSL/TLS where the user is logged in. Important informa-

⁶⁶Qualys SSL Labs SSL Server Test: <https://www.ssllabs.com/ssltest/>

⁶⁷Nmap: <https://nmap.org/>

⁶⁸Internet-Wide Scan Data Repository: <https://scans.io/>

⁶⁹ZMap - The Internet Scanner: <https://zmap.io/>

tion can be derived by observing the cipher text, such as session cookies that can be used to hijack sessions. This attack was named POODLE.^{70,71} Vulnerabilities to POODLE are only noted for 2015 since the attack was not yet known in 2013. For SSL 3.0, the only way to protect against POODLE is by disabling cipher suites that use block ciphers. POODLE also works with some web servers which implement padding in TLS 1.0 incorrectly, which updates to the web server software might be able to solve. Figure 2.6 shows the number of banks that are vulnerable to POODLE with either SSL 3.0 or TLS. 5 banks overlap, and are vulnerable to POODLE attacks with both protocol versions. Therefore, 30 out of 80 banks in our survey are vulnerable to POODLE attacks.

Within the SSL/TLS protocol suite (up to versions 3.0/1.0, respectively), one method to encrypt data is with block ciphers used in cipher-block chaining mode (CBC). The SSL/TLS standard mandates that chained initialization vectors (IVs) are used with CBC mode encryption. With chained initialization vectors, the last block of the previous ciphertext is used as an IV for the next message. This presents a vulnerability that can be exploited using a blockwise chosen-boundary attack (BCBA).⁷² A BCBA applied on a HTTPS session is known as a BEAST attack.⁷³ BEAST can be mitigated by letting servers only allow connections exclusively using TLS 1.1 or 1.2. Figure 2.6 shows that between 2013 and 2015 the number of banks that are vulnerable to BEAST attacks has increased. An explanation for this is that banks that became vulnerable at one point in time stopped supporting RC4, the only streaming cipher supported by SSL/TLS, since it is vulnerable to attacks [ABP⁺13]. The only alternative without disabling support for the older SSL 3.0 and TLS 1.0 protocol versions were cipher suites that applied CBC, and by implementing those the relevant banks became vulnerable to BEAST. This is likely seen as the preferable alternative, since the BEAST attack can be mitigated by browsers by implementing 1/n-1 record splitting as a workaround.^{74,75}

If an attacker can observe network traffic and manipulate a victim's browser to submit requests to a target site, it is possible to retrieve data from the TLS stream when DEFLATE compression is used. An attacker can steal session cookies with CRIME, which makes it possible to hijack a session.⁷⁶ While this attack is easier to execute compared to BEAST, it is also easier to defend against by disabling TLS compression. This can be done server- or client-side. The vulnerability is only exploitable when both server and client support and use TLS compression when a session is established. In 2013 only 7 banks supported TLS compression, which after two years was reduced to 4 banks.

⁷⁰This POODLE Bites: Exploiting The SSL 3.0 Fallback (2014), <https://www.openssl.org/~bodo/ssl-poodle.pdf>

⁷¹The POODLE bites again (2014), <https://www.imperialviolet.org/2014/12/08/poodleagain.html>

⁷²Here Come The \oplus Ninjas (2011), <http://www.hit.bme.hu/~buttyan/courses/EIT-SEC/abib/04-TLS/BEAST.pdf>

⁷³Vulnerability Summary for CVE-2011-3389 (2011), <http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2011-3389>

⁷⁴BEAST followup (2012), <https://www.imperialviolet.org/2012/01/15/beastfollowup.html>

⁷⁵Is BEAST Still a Threat? (2013), <https://community.qualys.com/blogs/securitylabs/2013/09/10/is-beast-still-a-threat>

⁷⁶CRIME: Information Leakage Attack against SSL/TLS (2012), <https://community.qualys.com/blogs/securitylabs/2012/09/14/crime-information-leakage-attack-against-ssl/tls>

SSL/TLS renegotiation makes it possible to use the same data session over multiple connections. Originally, the SSL and TLS protocols did not consider that different parties can use the same data session due to renegotiation, where one party has control of the connection before renegotiation, and the other afterwards. This allows a man-in-the-middle to inject plain text in an established session with a web server before the web server is reconnected with the user's browser through renegotiation, while others implemented an extension which fixed the problem.⁷⁷ As a response, some sites disabled the renegotiation feature.^{78,79} Renegotiating is cryptographically protected when both the server and the browser support the extension, thereby preventing the same data session to be shared between an end-user and a man-in-the-middle. Half of the banks that were vulnerable in 2013 fixed the issue in the following two years.

SSL/TLS supports a number of cipher suites with different key sizes to support the confidentiality and integrity of an established session. Some of these cipher suites are merely meant for testing and unlike regular cipher suites, they do not offer either authenticity of the server's identity (such as with anonymous (Elliptic curve) Diffie-Hellman) or encryption, due to the lack of required algorithms in the suite. To prevent this, insecure cipher suites should be disabled. Only two sites from our survey supported insecure ciphers in 2013, which have since then fixed this issue.

The SSL Server Test from Qualys designates all cipher suites that are less than 112 bits as 'weak'. If the assumption is made that data has to stay confidential and its integrity safeguarded against eavesdroppers for the period '2031 and Beyond', a minimum of 128 bits conforms with recommendations by NIST.⁸⁰ This is why any applied cipher suite with a symmetrical key length of less than 128 bits is considered vulnerable. However, there are a large number of sites that deploy cipher suites of which the shortest key length is 112 bits. These sites are noted separately in Figure 2.6 to distinguish sites which slightly deviate from NIST recommendations (less than 128 bits but at least 112 bits) and those which deviate significantly (less than 112 bits, i.e. 40 or 56 bits). Not much has changed since 2013. The most significant change was a moderate reduction in banks that supported very weak ciphers (from 17 to 10 banks). These banks disabled the weaker cipher suites in their web server configurations, forcing clients to use the stronger alternatives.

Support for SSL 2.0 (with cipher suites enabled) or SSL 3.0 is considered a vulnerability. SSL 2.0 has a number of flaws which were already acknowledged by Claessens et al. [2002]. These are the use of the same cryptographic keys for message authentication and for encryption (which makes the security of Message Authentication Codes (MACs) unnecessary weak when encryption key size is limited due to export restrictions), the sole dependence on MD5 as a vulnerable hash function to

⁷⁷SSL and TLS Authentication Gap vulnerability discovered (2009), <https://community.qualys.com/blogs/securitylabs/2009/11/05/ssl-and-tls-authentication-gap-vulnerability-discovered>

⁷⁸Disabling SSL renegotiation is a crutch, not a fix (2010), <https://community.qualys.com/blogs/securitylabs/2010/10/06/disabling-ssl-renegotiation-is-a-crutch-not-a-fix>

⁷⁹RFC 5746 - Transport Layer Security (TLS) Renegotiation Indication Extension (2010), <https://tools.ietf.org/html/rfc5746>

⁸⁰Transitions: Recommendation for key management—part 1: General (revision 3) (2012), http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

construct MACs, the lack of handshake protection, and the possibility to truncate a connection due to relying on the closure of the TCP connection. SSL 3.0 is also considered insecure since all available cipher suites for that protocol version are vulnerable. Cipher suites using CBC are vulnerable to the earlier discussed POODLE attack due to an inherent flaw in the SSL 3.0 protocol itself while the only supported streaming cipher is the RC4 algorithm, which is prone to attacks [ABP⁺13]. To mitigate these vulnerabilities, it is enough to simply disable SSL 2.0 and 3.0 support on the server. While support for SSL 2.0 and 3.0 has dropped by almost half of the banks that supported it in 2013, there still are a surprising number of banks that support these older versions.

It must be noted that modern browsers can mitigate some of the vulnerabilities as discussed in this section. SSL 2.0 support has been disabled in modern browsers for quite a while. Examples include Microsoft Internet Explorer, Mozilla Firefox and Opera.^{81,82,83} Support for SSL 3.0 in these browsers was disabled more recently.^{84,85,86} When such browsers are used to connect to a site, they are not vulnerable to the protocol vulnerabilities of SSL 2.0 or 3.0 since a higher protocol version must be negotiated with the server. If the server does not support a higher protocol version, the connection will simply fail. Users with older browsers are still vulnerable.

2.4.2 Additional TLS functions

There are several optional functions in TLS that can be used to increase security. These have to be implemented server-side. Figure 2.7 shows the support of several of these functions between 2013 and 2015 by the surveyed banking sites.

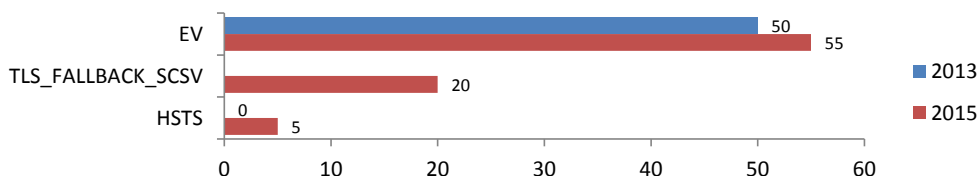


Figure 2.7: An overview of additional SSL/TLS functions supported by bank sites.

The functions found in the survey and some missing functions are each described briefly.

⁸¹Upcoming HTTPS Improvements in Internet Explorer 7 Beta 2 (2005), <http://blogs.msdn.com/b/ie/archive/2005/10/22/483795.aspx>

⁸²Bug 236933 - Disable SSL2 and other weak ciphers (2006), https://bugzilla.mozilla.org/show_bug.cgi?id=236933

⁸³Opera 9.5 for Windows Changelog (2008), [Opera9.5forWindowsChangelog](http://www.opera.com/changelog)

⁸⁴Security changes in Opera 25; the poodle attacks (2014), <http://www.opera.com/blogs/security/2014/10/security-changes-opera-25-poodle-attacks/>

⁸⁵Firefox - Notes (34.0) (2014), <https://www.mozilla.org/en-US/firefox/34.0/releasenotes/>

⁸⁶Security Bulletin MS15-032 - Cumulative Security Update for Internet Explorer (3038314) (2015), <https://technet.microsoft.com/en-us/library/security/MS15-032>

Extended Validation (EV)

We tested the availability of the EV attribute in certificates offered by bank sites, which notifies the user in various ways (depending on the used browser) that a more thorough identification process was followed before the certificate was issued. The expected procedures are noted in guidelines as published by the CA/Browser forum.⁸⁷ EV depends on the capabilities and willingness of users to recognize the difference between basic certificates and EV certificates. Whether EV provides any benefit is disputed. Without training or guidance, a considerable number of users do not notice the differences between offered basic and EV certificates in web browsers [JSTB07, SBVOP08, BVOP⁺09]. As shown in Figure 2.7, EV was already quite popular in 2013, but its popularity only raised marginally between then and 2015.

TLS Fallback Signaling Cipher Suite Value (TLS_FALLBACK_SCSV)

When a browser and server negotiate which SSL/TLS versions and cipher suites will be used, a fallback mechanism exists in case the handshake fails. If a connection on a higher protocol version fails, the default policy is to try one lower protocol version since it is assumed that the other party does not support the higher version. This fallback mechanism sometimes is used incorrectly in a situation where both parties actually do support a higher version. For example, a browser will try to reconnect with a server using a lower protocol version even though both browser and server support a higher version. Reasons for failure can simply be a network disruption the first time a browser attempts to connect, but an adversary can also use this flaw to force a downgrade of the protocol (also known as a downgrade attack) to an exploitable version by influencing the availability of a connection between browser and server. TLS Fallback Signaling Cipher Suite Value (also known by its TLS cipher suite value: TLS_FALLBACK_SCSV) is an extension for TLS which prevents the use of a lower version protocol in scenarios where the initial handshake for protocols to use between browser and server fails.⁸⁸ The extension is added to any re-connection attempt by the browser, so the server knows that a downgrade was performed. If the downgrade was unjustified (both the browser and server support a higher protocol version), the server refuses the connection. Support for TLS_FALLBACK_SCSV requires up-to-date web server software and SSL/TLS libraries. This extension is relatively new since it was proposed in 2014, yet a quarter of the banks that we examined already implemented it one year later.

HTTP Strict Transport Security (HSTS)

When a user enters a website name without specifying the protocol the insecure ‘http’ protocol will be used by default, even if SSL/TLS (through the ‘https’ protocol identifier) is available. A man-in-the-middle who has control of the connection between the user’s computer and the bank can prevent a user from ever connect-

⁸⁷The latest version of the Guidelines For The Issuance And Management Of Extended Validation Certificates can be obtained from: <https://cabforum.org/documents/>

⁸⁸RFC 7507 - TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks (2015), <https://tools.ietf.org/html/rfc7507>

ing to the secure site by manipulating all replies from the bank.⁸⁹ HTTP Strict Transport Security (HSTS) provides protection against man-in-the-middle attacks which exploit this initial insecure connection by implementing an additional HTTP response header.⁹⁰ This header instructs browsers that for future visits within a specific time frame only secure connections through SSL/TLS ('https') should be allowed. To also protect the first visit, browser updates include a list of sites that should only be visited securely. (Retro)fitting web servers with HSTS support is quite simple, since only a HTTP response header has to be added to its existing configuration. An example which states that only secure connections should be allowed for a year would be: **Strict-Transport-Security: max-age=31536000**. Note that this yearly counter is updated every time the user visits the site, making it unlikely that it would ever expire if the user visits the site regularly. Despite its simplicity, HSTS is only implemented by a few banks in our survey.

HTTP Public Key Pinning (HPKP)

A similar useful HTTP response header is HTTP Public Key Pinning (HPKP), which allows browsers to detect fraudulently issued certificates from trusted certificate authorities.⁹¹ On the first visit to a site that supports HPKP, the site tells the browser that at least one certificate in the trust chain should contain a specific public key for subsequent visits in a certain time frame. If within this time frame the site is revisited and a valid certificate chain is offered that does not contain one of the earlier registered public keys, the browser refuses to connect. This protects against trusted but fraudulent certificate authorities who issue valid certificates of sites for adversaries. If in a subsequent visit the certificate chain has been changed in such a way that the HPKP policy is violated, it indicates that a wrongfully issued certificate is being offered, possibly as part of a man-in-the-middle attack. HPKP requires that two public keys are specified. If the primary public key is compromised (such as when an adversary obtains the paired private key) and revoked, the backup public key can be used to replace the lost part of the certificate chain. This avoids the situation where the security measures of a browser prevent access to the site with a new legitimate certificate chain. It is recommended that a backup private key and backup certificate are kept on an offline medium for safekeeping, since they can be used for undetectable man-in-the-middle attacks if compromised. An example of an HPKP HTTP response header that would pin two public keys for two months on each visit of the site and any of its subdomains would be: **Public-Key-Pins: pin-sha256="ABCxyz123+(...)"; pin-sha256="XYZabc987+(...)"; max-age=5184000; includeSubDomains**. Note that for the example the PINs (encoded in Base64 SHA-256 hash values of public keys) have been shortened for readability.

This extension was not used by any of the banks at the time the survey was conducted, which is why it is absent in the graph shown in Figure 2.7.

⁸⁹New Tricks For Defeating SSL In Practice (2009), <https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf>

⁹⁰RFC 6797 - HTTP Strict Transport Security (HSTS) (2012), <https://tools.ietf.org/html/rfc6797>

⁹¹RFC 7469 - Public Key Pinning Extension for HTTP (2015), <https://tools.ietf.org/html/rfc7469>

Other non-SSL/TLS related response headers

There are several HTTP response headers that increase security, but which do not relate to SSL/TLS. Examples focus on preventing cross-site interaction that can be insecure, such as through frames (the `X-Frame-Options` header⁹²) and scripting (the `X-WebKit-CSP` header⁹³). These response headers were excluded from the survey both due to limited time, and due to that they have not (yet) been accepted as standards (`X-WebKit-CSP`) or have been obsoleted without a replacement being available (`X-Frame-Options`).

2.4.3 Geographical spread of vulnerabilities and functions

The data in Table 2.5 shows the global distribution of important SSL/TLS functions and vulnerabilities according to our survey data.

Region # banks	Africa	Asia	Europe	M-East	N-America	Russia	Oceania	S-America
Extended validation	3	12	19	4	4	1	7	5
TLS Fallback Sign.	0	0	14	1	4	0	1	0
HSTS	0	0	5	0	0	0	0	0
POODLE (SSL 3.0)	6	4	7	0	1	0	1	1
POODLE (TLS 1.0)	1	5	3	1	0	0	2	3
BEAST	7	6	19	4	4	1	5	4
CRIME	2	1	1	0	0	0	0	0
Insecure reneg.	1	0	1	1	1	0	0	3
<112 bit ciphers	3	2	1	1	1	0	0	2
SSL 2.0	2	0	0	0	0	0	0	2
SSL 3.0	7	11	13	1	1	0	3	5
Function support	13%	29%	47%	28%	44%	33%	38%	15%
Vulnerability	45%	26%	21%	17%	17%	13%	20%	23%

Table 2.5: Geographical distribution of important SSL/TLS functions and vulnerabilities in 2015.

As noted in Section 6.1, we believe that our set of banks is representative for banking worldwide. However, there is room for variance due to the limited number of observed banks in some regions. Comparisons using the final percentages should therefore be made with care. The most obvious case is Russia, for which only a single bank was examined which neither supports any of the listed functions and which has none of the listed vulnerabilities. We therefore only make conclusions based on the other regions.

Europe and North America seem to be quite active in supporting new SSL/TLS functions, followed by Oceania, East-Asia and the Middle East. Other regions do not seem to give any priority to the support of additional secure functions of SSL/TLS. SSL/TLS vulnerabilities in banking sites were observed mostly in Africa, while the presence of such vulnerabilities seems to be less in other regions.

⁹²RFC 7034 - HTTP Header Field X-Frame-Options (2015), <https://tools.ietf.org/html/rfc7034>

⁹³Content Security Policy Level 2 (2015), <https://www.w3.org/TR/CSP2/>

2.4.4 SSL/TLS overall observations

It is a positive development that most banks see SSL/TLS as something that should not be configured once and then be left alone. Examining Figure 2.6 shows that the number of occurrences for most vulnerabilities dropped between 2013 and 2015 (ignoring BEAST, which can be considered a non-issue if an updated browser is used). Unfortunately, there still are a large number of banks (10 out of 80, or 12.5%) that support insecure cipher suites with 40 or 56 bit key sizes.

Optional security-enhancing SSL/TLS functions became slightly more popular, as shown by Figure 2.7. Extended Validation is already widely implemented, and saw a slight increase. The TLS Fallback Signaling Cipher Suite Value was implemented by 25% of the examined bank sites between its introduction and the 2015 survey. HSTS is a bit of an exception. Of all the optional functions that we examined, it has the lowest technical threshold to implement. Considering that HSTS was already available before the 2013 survey, it is not known why there are only a few bank sites which implemented it.

Mobile banking and payment applications are prone to implementing communications security wrong, resulting in large security gaps [GIJ⁺12, FHM⁺12, RSB⁺15]. Online banking sites have the advantage that they rely on browsers to implement SSL/TLS correctly client-side, which reduces the development area of banks in which mistakes can be made. Therefore the work for banks is to keep their server-side implementations as secure as possible. From the data in the survey, it can be concluded that most banks do it well, but there are some sites that are still vulnerable in ways that were considered old a decade ago.

2.5 Discussion, limitations and further research

Of 81 banks in 2013 and 80 banks in 2015, the user authentication methods were examined in Section 2.3 and the communications security implementations for home banking were examined in Section 2.4. When the conclusions are compared, a difference in uniformity is quite clear. All banks rely on SSL/TLS for communications security in home banking. There is some variety in how well SSL/TLS is implemented, but all banks chose to use parts from a single, standardized protocol suite. This is in sharp contrast to the methods applied for user authentication, which vary greatly. Most likely SSL/TLS provides ‘good enough’ communications security, while there are several factors for why banks cannot agree on a single user authentication method. Such factors can include demographic differences in which methods are accepted by bank customers. For example, in the United States bank card or credit card payments are often conducted without requiring a PIN ⁹⁴. Instead, a physical signature (easy to forge, and easy to forget to check) is requested from the person who uses the card. One reason why issuers hesitate to introduce PINs to cards is that they do not want to have a card in the user’s wallet that is more difficult to use compared to cards from competitors ⁹⁵. Such differences exist as well in online bank-

⁹⁴Preparing for Chip-and-PIN Cards in the United States (2014), <http://bits.blogs.nytimes.com/2014/12/02/preparing-for-chip-and-pin-cards-in-the-united-states/>

⁹⁵Chip & PIN vs. Chip & Signature (2014), <http://krebsonsecurity.com/2014/10/chip-pin-vs-chip-signature/>

ing user authentication methods in different regions, as shown by the survey data in Section 2.3. Further research that examines online banking should also focus on whether communications security is correctly implemented. However, the area with the most work for further research seems to be user authentication and transaction authorization, for which the industry does not have a unified answer.

A limitation of the survey related to communications security is that it was examined for home banking, but not for mobile banking. Whenever a browser is not used, banks must implement SSL/TLS in their mobile applications themselves. These implementations are not always secure [FHM⁺12, GIJ⁺12, ODC15, RSB⁺15]. Aside from client-side, there can be server-side issues. The survey in this chapter has shown that for home banking, servers can have vulnerabilities that potentially weaken communications security. The same could be true for servers used by mobile banking applications. More research in both can provide a more complete overview on how well SSL/TLS is used client- and server-side for mobile banking.

Another limitation is in that authentication methods were only examined from an external perspective, using login pages and documentation. For example, for the discussed password and PIN implementations in Section 2.3.2 we were unable to get additional details concerning password policies that could influence security and usability. Letting the user choose a password, having relaxed rules about the length and complexity of the password, and not requiring the user to renew passwords on a regular base all increase usability but potentially decrease security, and vice versa. Most banks do not publish their password policies, which is why this was excluded from the survey. Examining such policies would give more insight in the overall security concerning the often applied knowledge factor (either by itself, or in combination with a possession factor) but would require accounts at most banks, since it is quite rare that password policies are public. Gaining such detailed information could be investigated in further research. One approach would be by letting researchers around the world cooperate by sharing information about security systems from the banks where they are customers at. The same further research could also expand on the 80 chosen banks and provide a better geographical distribution of examined banks. Examining home and mobile banking qualitatively and quantifying the results takes a lot of effort. This is especially true for the collection of information concerning authentication methods, since it requires the examination of documentation login pages and mobile banking software, all in many different languages. 80 banks worldwide is a small number, but it takes a large amount of time to examine them and it could be that some of the finer points of their authentication methods were missed.

In Section 2.4.3 some observations were made about the geographical distribution of SSL/TLS vulnerabilities and optional functions. Due to the somewhat limited sample size these observations are solely based on the impressions that the survey gives and not on a statistical foundation. The same is true for other sections in which different regions are compared.

A question not answered by our survey is what influences the observed regional differences. The results suggest that there are differences in how regions implement user authentication methods and SSL/TLS, as shown by Figure 2.4, Table 2.3 and Table 2.5. Answering this question could give more insight in which motivators (e.g. regulations) drive banks and users to adopt more secure implementations and

methods.

2.6 Related work

The primary focus of this work is on the development of online banking in general and the security in online banking in particular. Security in online banking has been an active research subject for many years. This section notes related work.

Several references are made in Section 2.2.1 to work which examines what makes users accept online banking, based on the technology acceptance model. Two other models that are also used to examine the acceptance of technology are the theory of reasoned action and the theory of planned behavior. All three models have been examined in an online banking context. The technology acceptance model has the best fit to determine what makes online banking acceptable [YFP10].

Data was collected for the survey about methods used by banks to authenticate customers, which are discussed in Section 2.3. Many of these methods have also previously been examined and proposed in the academic field. AlZomai et al. investigated the effectiveness of an information scheme which makes the customer verify transactions securely and also proposed a method which implemented such a scheme [AAJM08, AAJ10]. This scheme is known as What You See Is What You Sign (WYseeIWYS). Weigold and Hiltgen proposed several methods which use WYseeIWYS [WH11]. An alternative to WYseeIWYS was proposed by several authors of the paper on which this chapter is based, under the name What You Enter Is What You Sign [KVvE14]. More information about What You Enter Is What You Sign can be found in Part II of this thesis.

Section 2.4 is dedicated to the use of SSL/TLS by bank sites and browsers, which authenticates the bank to the customer and provides confidentiality and integrity. When this protocol is used by a browser to conduct online banking, it relies on the perception of the customer to see if it is used and whether it is offered in a secure manner, based on several visual security indicators. The availability and effectiveness of these browser security indicators have been examined to a great extent [DTH06, Ogh09, ATVO12].

SSL/TLS is used for more than just browser-server traffic. Several authors have examined SSL/TLS implementations used for (among others) mobile banking applications [GIJ⁺12, FHM⁺12, RSB⁺15].

Of course, security in online banking is more than authentication and communications security. One example is the detection of fraudulent transactions by banks, based on characteristics of the transaction itself and on customer behavior (also briefly discussed in Section 2.3.4). Academic proposals for such systems have also been made [Agg06, WLC⁺13].

2.7 Concluding remarks

We identified a pattern in the development of online banking which seems to rely on three phases, each relating to both technological and adoption trends. In the early adoption phase, banks offer a technologically crude way to conduct online banking that is expensive and not available to everyone. Availability and popularity of online

banking rise in the following expansion phase, in which users start to accept online banking due to that critical aspects are perceived as being satisfactory. Finally, the exploitation phase relies on standardized technologies to make online banking available to almost anyone. The three phases are identified in the development of home banking (using a ‘desktop’ computer), and the first two phases can also be identified in the development of mobile banking (using a mobile device anywhere an internet connection is available). Based on the identified trend we predict that mobile banking has yet to enter the exploitation phase. In this predicted third phase, Hybrid Mobile Applications that are based mostly on standard web technologies will likely be introduced to reduce the costs of supporting multiple platforms and form factors. For mobile banking, this opens opportunities for new kinds of scalable malware attacks that are similar to attacks made against home banking.

Security is an important aspect in online banking. For home banking, we examined 80 banks worldwide on how they authenticate their customers and how they implemented communications security. We also examined the implemented authentication methods for mobile banking at 66 banks.

For user to bank authentication, 75% of the banks offer an authentication method which relies on multiple factors (what the user knows and possesses) for home banking. The possible use of multiple factors was found in 59% of mobile applications and 25% of mobile sites. The adoption of multi-factor authentication in both home and mobile banking increased slightly in a two year period, and seems to be most absent in North America. While there is not much diversity in the used knowledge factor (either password or PIN), different regions have different preferences for the possession factor. Noteworthy are the wide embrace of offline electronic devices used to generate login credentials in Africa, Europe and Oceania, and the popularity of one-time password distributed on paper or plastic in South America. Different possession factors are also used in mobile banking. Use of the mobile device itself as the possession factor is overall most favored. A recent development in mobile banking is that fingerprint-based biometrics are slowly starting to be offered in alternative authentication schemes, despite that it is trivial to spoof fingerprint sensors embedded in user devices.

Whereas authentication from customer to bank is quite varied, the opposite is true for bank to customer authentication. The SSL/TLS protocol suite is used for communications security in home banking by all examined banks. All banks apply ciphers which provide confidentiality and integrity, but 12.5% of the banks support ciphers which provide an amount of protection that is far below NIST recommendations. The server-side implementation of SSL/TLS can also present several vulnerabilities which endanger the communication between bank and customer to eavesdropping and man-in-the-middle attacks. We found most of these vulnerabilities at banks in Africa. Support for optional SSL/TLS functions which increase security is mostly found in Europe and North America. Most banks have an implementation that is adequate to protect against man-in-the-middle attacks, but there are some sites which still present vulnerabilities that could have been solved more than a decade ago.

Our work gives quite a good overview of security technologies used in online customer-bank interaction. The research area with low-hanging fruit seems to be user authentication (and transaction authorization). Unlike communications secur-

ity, a de facto standard that provides ‘good enough’ security for user authentication is missing.

The survey took quite some time. Communications security was easy to examine because only a single protocol is used for security. Its multiple versions, features and flaws can be examined using standard tooling since most of the Internet relies on it. However, this is different for user authentication methods in online banking. These vary greatly, and are not easy to examine from an outsider’s perspective. Future research could focus on improving the examination of user authentication methods in online banking. One approach might be the assistance of security researchers worldwide, who can be customers at one or more banks and therefore provide more detailed information about applied authentication methods. There are a few banks in the survey which use exceptional methods. It would be a shame not to examine these exceptions, since they could provide new insights that could improve yet to be designed authentication methods.

It is important to note that the survey does not look at all security aspects of online banking. For example, banks can implement behavior anomaly detection, used to detect financial transactions that are made under suspicious circumstances. While there are indications that some banks implement such in-house systems, they are hard to examine from an outside perspective. Examination of and improving these systems would require cooperation between academic institutes and banks in an open-minded setting.

Part II

Expanding transaction authorization options

Part II - Expanding transaction authorization options

What You Enter Is What You Sign (WYenterIWYS) is an alternative transaction authorization scheme that aims to expand the existing range of options to secure transactions. Its goal is to secure the authenticity and integrity of transactions initiated by the user. This is also the goal of the What You See Is What You Sign (WYseeIWYS) scheme as currently used by banks, but the key difference is that the user will be cognitively less challenged by WYenterIWYS. In addition, the user is not in a position to actively refuse participation or take shortcuts in the secure use of the authorization scheme. WYenterIWYS does so by securing data as soon as it is entered by the user, rather than after it is received by the bank (as is the case with WYseeIWYS).

Sven Kiljan, Harald Vranken, and Marko van Eekelen. What You Enter Is What You Sign: Input Integrity in an Online Banking Environment. Published in Proceedings of the 4th International Workshop on Socio-Technical Aspects in Security and Trust (STAST), pages 40-47, July 2014. [KVvE14]

Chapter 3 is based on the paper that originally introduced WYenterIWYS. Its main goal was to introduce the concept as a transaction authorization information scheme by describing a possible information flow between user and bank using both secure and insecure elements. A possible implementation was given in a follow-up paper, which was the base for Chapter 4. What makes this implementation suggestion practical is its independence from the user-owned device. If a user is able to perform online banking with a device, the same device can be used with WYenterIWYS-based authentication.

Sven Kiljan, Harald Vranken, and Marko van Eekelen. User-Friendly Manual Transfer of Authenticated Online Banking Transaction Data. Published in Proceedings of the 13th International Joint Conference on e-Business and Telecommunications, pages 259-270, July 2016. [KVvE16b]

What I wanted to show with WYenterIWYS is that it is possible to design a transaction authorization scheme that is not based on an existing scheme, with a little bit of effort and out-of-the box thinking. As an alternative to WYseeIWYS, it taxes the user cognitively less since the user is not required to compare values. With the suggested implementation in Chapter 4, the only action required of the user would be to transcribe a code, an action which gives users little room for making mistakes or circumventing security. If users are able to perform user authentication using one-time passwords or challenge-response authentication, they should be able to use WYenterIWYS to authorize transactions.

For this thesis, Chapters 3 and 4 have been extended respectively with Section 3.6 and Section 4.4, which focus on formal protocol verification of WYenterIWYS as described in each chapter.

Chapter 3

What You Enter Is What You Sign: input integrity in an online banking environment

Abstract

One problem with most currently used transaction authentication methods is that they depend on the customer's computer for integrity of the information flow between customer and bank. This allows man-in-the-middle attacks to be conducted using malware for financial fraud. Some banks are implementing new authentication methods that allow customers to verify transactions received by a bank without depending on the customer's computer to provide information integrity. These new methods are more complex compared to traditional authentication methods and need the customer's attention to be effective, since it is up to the customer to verify the information that was received by his or her bank. By examining the intrinsic problems of traditional and new transaction authentication methods as used by banks, we designed an alternative authentication method named 'Entered Single Transaction Authentication'. Our method ensures that the bank receives information as the customer entered it without requiring further verification by the customer. We introduce the concept 'What You Enter Is What You Sign', which ensures the digital integrity of information as soon as it is entered. Our proposal is theoretical and high-level, but opens the way for secure transaction authentication methods that rely to a lesser extent on the authenticating party to provide correct information, thereby reducing errors and improving user friendliness.

3.1 Introduction

The use of online banking continues to grow in many countries. For instance, in the European Union the use of online banking by individuals aged 16 to 74 increased from 25% in 2007 to 42% in 2013.⁹⁶ Examples of growth in individual countries where Internet banking is being used by a large part of the population include the Netherlands (65% in 2007 and 82% in 2013) and Denmark (57% in 2007 and 82% in 2013). Opposite examples of countries where Internet banking is slowly gaining acceptance include Greece and Turkey (each 4% in 2007 and 11% in 2013). The trend is that the use of online banking continues to grow in most countries.

With this relatively new type of banking comes a new type of fraud. Instead of interacting directly with a bank (i.e. by talking to an employee at a bank's local office), more and more banking customers rely on electronic devices to effect wire transfers. Criminals follow suit. Instead of robbing a bank directly (e.g. by threatening an employee or by breaking into a vault), criminals that commit online banking fraud often focus on deceiving customers instead.

Types of attacks that involve the customer can be distinguished by the actions of an adversary. There are impersonation attacks, with which an adversary obtains authentication information (such as user names, passwords and PIN codes) to create malicious transactions. Impersonation attacks are characterized by an adversary creating a new session with the bank in name of (and thereby impersonating) the customer. Another type is a man-in-the-middle attack, in which the adversary injects information in an existing session between customer and bank. With a successful man-in-the-middle attack, neither the bank nor the customer notice any discrepancies when the adversary makes sure that both parties in the session see what they expect to see.

Man-in-the-middle attacks are often executed through the computers of banking customers [KSDC⁺14]. Redhead and Povey's (1998) work states that in the development of online banking applications at the time, general attention was too strongly focused on the issues of network security and not enough on the security of the customer's computer [RP98]. Their prediction was that the developers of malware would use their skills for financial gain by targeting online banking, which they did.

Several banks are applying Customer Verified Transaction Set Authentication (CVTSA) to prevent man-in-the-middle attacks. With CVTSA, a bank receives transaction information in an insecure way (either from the customer or an adversary) and applies a secure way to make a customer validate the information it received. What You See Is What You Sign (WYseeIWYS) is used with CVTSA since the customer has to sign information presented by the bank in a secure way. Former empirical research concluded that 21% of online banking customers do not spot significant changes when comparing critical transaction values [AAJM08]. CVTSA leaves room for improvement through the mitigation of insecure user behavior.

Our contribution is a new transaction authentication method with the name Entered Single Transaction Authentication (ESTA) which, like CVTSA, aims to prevent man-in-the-middle attacks. With ESTA, a bank can make a distinction between actions of a customer and those of an adversary. What distinguishes ESTA

⁹⁶Eurostat - Individuals using the Internet for Internet banking: <http://epp.eurostat.ec.europa.eu/tgm/refreshTableAction.do?pcode=tin00099&language=en>

from CVTSA is a new concept which ESTA applies, named What You Enter Is What You Sign (WYenterIWYS). WYenterIWYS adds data integrity to digital information as soon as it is created (entered by a human). When data integrity is added as early as possible, customers do not need to verify information that the bank received to detect man-in-the-middle attacks. We introduce ESTA using our research platform, known as the Trusted Entry Pad (TEP).

In Section 3.2, we give a high level view of three different types of transaction authentication: the type that is currently in widespread use (Traditional Transaction Authentication, or TTA), a new type which is being introduced by several banks (CVTSA) and our proposal (ESTA). We note an important limitation of TTA and how CVTSA and ESTA solve this. Finally, we note the difference between CVTSA and ESTA.

The following three sections contain use scenarios of each transaction authentication type. Section 3.3 gives an example of how a man-in-the-middle attack can be used to work around the security offered by TTA. In Section 3.4, we explain how CVTSA protects against man-in-the-middle attacks and note how this requires additional attention from the customer compared to TTA. How ESTA is used is noted in Section 3.5. This section also mentions how ESTA offers the same protection as CVTSA without requiring additional attention or actions from the customer during transaction authentication.

Safet Acifovic formally verified ESTA during his time as a student at Radboud University [Aci15]. His work is briefly discussed in Section 3.6.

Related work to ESTA is noted in Section 3.7. This includes comparisons with TTA and CVTSA-based authentication devices. We also look at existing technology and concepts which a potential ESTA implementation can use. Section 3.8 follows with possible directions for further research based on our work. Finally, Section 3.9 contains our concluding remarks.

3.2 Transaction authentication

Entity and transaction authentication in online banking each apply to different actions initiated by a customer. Entity authentication concerns the customer proving his or her identity to the bank to initiate a new session. Transaction authentication concerns the customer proving the authenticity of transaction requests when the customer asks the bank to approve the requests and create transactions based on them. We distinguish three different types of authentication methods that relate to transaction authentication.

3.2.1 Traditional transaction authentication (TTA)

TTA effectively re-applies entity authentication since it misses or has a limited relation with transaction requests. When a bank asks a customer to authenticate a transaction request or a set of transaction requests, the necessary information concerning the transactions is presented to the customer on his or her computer. The computer owned by the customer is a potential man-in-the-middle when it is compromised by malware. When this happens, it cannot protect the information flow integrity from a customer to a bank and vice versa by itself. An adversary can hide

newly created illegitimate transaction requests or change characteristics of requests made by the customer before they are sent to the bank. When the bank asks for authentication, the adversary only has to show the original transaction requests to the customer. By hiding the new or modified transaction requests, the customer has no reason for suspicion and continues to authenticate the transaction requests that he or she did not create using TTA.

More information on this flaw in TTA is given in Section 3.3.

3.2.2 Customer verified transaction set authentication (CVTSA)

Several banks implement new authentication methods that allow customers to validate transaction requests received by banks without relying on the customer's computer for integrity of information presented to the customer. We refer to these methods as CVTSA. CVTSA applies the concept of What You See Is What You Sign (WYseeIWYS) when customers verify (sign) information that can be interpreted in a single semantic context. The information flow of CVTSA is shown in Figure 3.1.

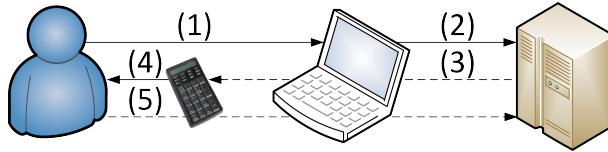


Figure 3.1: The transaction information flow in CVTSA methods. A secure device, provided by the bank, is used to present transaction requests received by the bank to the customer for authentication. Dashed lines represent information flows which are optionally forwarded by the traversed devices, depending on the type of authentication device.

The customer enters one or more transaction requests in the client computer (step 1), which are forwarded to the bank (step 2). The bank sends information concerning the transaction requests cryptographically secured to the authentication device in step 3, either through an out-of-band channel or through the customer's computer. The device verifies the authenticity of the received information (i.e. whether it was sent by the bank) and presents transaction request information to the customer in step 4. The customer must confirm that the set of transaction requests entered in step 1 is equal to the set received in step 4. Accepting the transaction requests must only be done if there are no discrepancies. Either the customer does nothing and the transactions are rejected, or the acceptance or rejection is communicated to the bank in step 5. This final step varies in the use of the authentication device and the customer's computer. One example is the use of a verification code (sent earlier in step 4) to be entered in the customer's computer. Another example is the customer expressing acceptance or rejection on the authentication device, which forwards the customer's decision to the bank either directly or through the customer's computer.

One problem of CVTSA is that it requires conscious attention to be paid by the customer. Not only must the customer check whether information is entered correctly in his or her computer. The customer must also check the information which the bank received. If the customer does not perform the check thoroughly, fraud is

still possible. A study of AlZomai et al. (2008) shows that of the test participants, 79% successfully noticed account numbers of which five out of eight digits were replaced in a simulated attack on CVTSA [AAJM08]. While this percentage is quite high, it must be noted that the use of the same authentication method over a longer time span was not tested. The risk exists that customers will trade security for usability. They can do this by only looking for the validation code that is required for authentication while ignoring the information concerning transaction requests.

An example of the use of CVTSA is given in Section 3.4.

3.2.3 Entered single transaction authentication (ESTA)

We propose a minimalistic approach for transaction authentication which we name ESTA. 'Minimalistic' refers to a minimum of complexity in both usability and technology. Instead of applying WYseeIWYS, we apply a new concept that we name What You Enter Is What You Sign (WYenterIWYS) for ESTA. This concept is less complex in usability compared to WYseeIWYS as applied by CVTSA since the customer does not have to verify data received by the bank. Technical complexity is kept to a minimum by using standard technologies in a simple design.

The information flow of ESTA is shown in Figure 3.2.

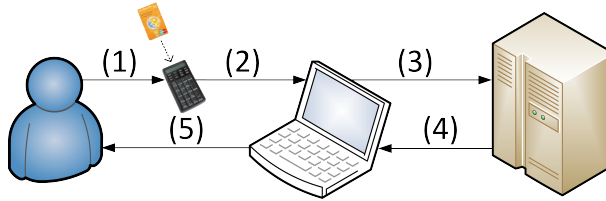


Figure 3.2: The transaction information flow of our proposed WYenterIWYS-based authentication method for single transaction requests.

ESTA was conceived using our research platform named 'Trusted Entry Pad' (TEP), which represents a device with a display, keypad, smart card slot and a connection to a customer's computer. A smart card is used for storing and applying cryptographic resources. The customer enters one critical value of a single transaction request into the TEP (step 1). After the customer confirms his or her entry, a digital signature of the value is made. Both the value and the digital signature are sent to the customer's computer (step 2), which forwards it to the bank (step 3). The bank checks whether the received value and signature match and sends a confirmation back to the customer's computer (step 4), which shows it to the customer (step 5). The process is repeated for each critical value necessary to complete the transaction request. Examples of critical values include the destination account number (i.e. where the money will go to) and the amount (i.e. how much money will be removed from the customer's account to be sent to the destination account number).

An example of how ESTA can be applied to prevent the manipulation of critical information is given in Section 3.5.

3.3 Traditional transaction authentication (TTA)

In this section, we clarify why the dependence on an untrusted device to provide information integrity is a shortcoming of TTA. We also point out why aggregated data should not be used as verification information using the sum of a transaction set as an example. First, we will give a scenario in which TTA is used successfully. After that, we demonstrate an attack using the same scenario in which a legitimate transaction is turned into a fraudulent transaction.

Note that the scenario we give for TTA is not fictional. Several banks apply challenge-response authentication for transaction authentication [KSDC⁺14].

In the scenario, entity authentication already took place and the customer is logged in the secure banking environment.

The customer is Alice (A). She wants to send money to both Bob and Charlie, and creates two transaction requests in the bank's (B) online environment using her computer (C). The critical values of the transaction requests are the destination account number and the amount (respectively D_1 and S_1 for Bob, and D_2 and S_2 for Charlie). After entry, the bank returns an overview of all prepared transaction requests.

$$D_1 = 123456789, S_1 = 500$$

$$D_2 = 987654321, S_2 = 100$$

$$S_a = \sum S_{1,2} = 600$$

- | | |
|--|---|
| (1) $A \rightarrow C : D_1, S_1$ | (7) $B \rightarrow C : N_b, S_a$ |
| (2) $C \rightarrow B : D_1, S_1$ | (8) $C \rightarrow A : N_b, S_a$ |
| (3) $A \rightarrow C : D_2, S_2$ | (9) $A \rightarrow T : PIN, N_b, S_a$ |
| (4) $C \rightarrow B : D_2, S_2$ | (10) $T \rightarrow A : E_K\{N_t, N_b, S_a\}$ |
| (5) $B \rightarrow C : D_1, S_1, D_2, S_2$ | (11) $A \rightarrow C : E_K\{N_t, N_b, S_a\}$ |
| (6) $C \rightarrow A : D_1, S_1, D_2, S_2$ | (12) $C \rightarrow B : E_K\{N_t, N_b, S_a\}$ |

Alice enters the critical values of each transaction request in her computer, which sends it to the bank (steps 1 to 4). The bank returns an overview of all entered transactions for Alice to verify on her computer (steps 5 and 6).

Before the wire transfers are created, the bank authenticates Alice's transaction requests using challenge-response authentication. Alice receives a random nonce generated by the bank (N_b) and the rounded down total amount of all prepared transaction requests (S_a) through her computer (steps 7 and 8). These two values form the challenge. S_a is first used by Alice to verify that the total sum of all transactions as received by the bank is the same total sum of the transaction requests she entered in steps 1 and 3. After the verification, Alice unlocks the functionality of an electronic token (T) using a Personal Identification Code (PIN) and enters the challenge (step 9).

The token creates a response by encrypting the current time stamp from its local clock (N_t) and the challenge with a symmetric key (K) that is only known to the token and the bank. Alice enters the result in her computer, which in turn sends the encrypted message to the bank (steps 10 to 12).

To verify the correctness of the transaction requests, the bank must first decrypt the received response using K . Of the decrypted values, N_t and the current time must both be in an accepted time frame to prevent replay attacks. N_b and S_a must be equal to the challenge that was sent. If verification is successful, wire transfers are created for the transaction requests that Alice made in steps 1 and 3.

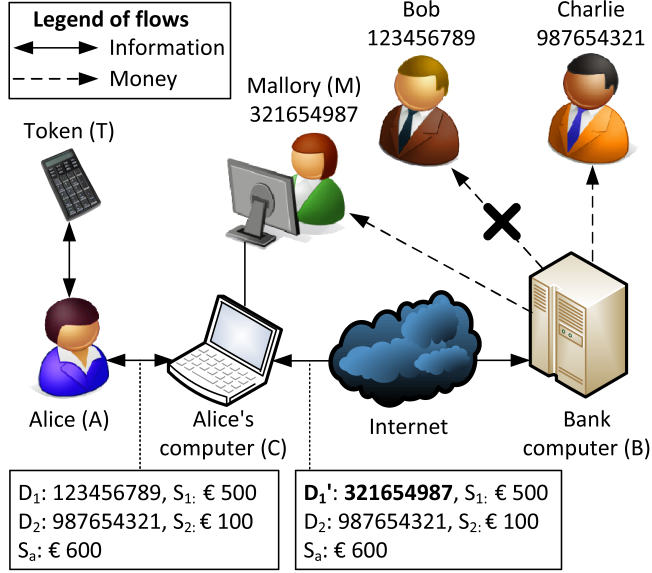


Figure 3.3: Alice fails to transfer money to Bob.

Mallory (M) is a malicious adversary and has control over Alice's computer, as shown in Figure 3.3. Her goal is to gain the money that is meant for Bob.

$$D'_1 = 321654987$$

- | | |
|-----------------------------------|---|
| (1) $A \rightarrow C : D_1, S_1$ | (4) $C \rightarrow B : D_2, S_2$ |
| $C \rightarrow M : D_1, S_1$ | (5) $B \rightarrow C : D'_1, S_1, D_2, S_2$ |
| $M \rightarrow C : D'_1$ | $C \rightarrow M : D'_1, S_1, D_2, S_2$ |
| (2) $C \rightarrow B : D'_1, S_1$ | $M \rightarrow C : D_1$ |
| (3) $A \rightarrow C : D_2, S_2$ | (6) $C \rightarrow A : D_1, S_1, D_2, S_2$ |

Before the destination account number and amount of the first transaction request are sent to the bank, Mallory replaces the original destination account number D_1 with account number D'_1 , which is under her control (between steps 1 and 2). Mallory intervenes again when the bank sends the transaction requests back to Alice's computer for review. She makes sure that Alice sees the transactions as she entered them in her computer. The modified destination account number that was sent to the bank is kept hidden from Alice due to Mallory's second intervention (between steps 5 and 6).

Alice proceeds with the challenge-response scheme, in which Mallory does not have to intervene (steps 7 to 12). Alice does not get suspicious when she checks S_a

(the total sum of all transaction requests) as part of the challenge in step 8 since it has not changed. S_1 and S_2 were not modified, and therefore S_a stays the same. The attack is a success.

Mallory could also change both S_1 and S_2 for her benefit in a more advanced attack. She only needs to make sure that S_a stays the same to avoid suspicion by Alice (e.g. S'_1 as € 599 and S'_2 as € 1), since S_a is part of the challenge. If S_a would be changed, Alice would enter the wrong challenge in T (the bank's token), which creates an invalid response that will be noticed by the bank.

The flaw in the design of the used TTA method is that the integrity of critical information provided by Alice was not safeguarded, which allowed Mallory's attack to succeed. Alice's computer acts as a man-in-the-middle, which allows it to change unverified information between Alice and the bank.

The bank offers its customers the possibility to verify the total sum that is received by the bank. Because this concerns aggregated information (S_a in steps 7 and 8), the semantic content of the message (the set of transaction requests from steps 1 to 6) can be changed.

The example applied challenge-response authentication, but an adversary also has the same opportunity if other multi-factor authentication methods are used in which the integrity of the customer's input is not protected. This includes one-time passwords and digital signatures over information which a customer can only verify on the display of his or her computer.

We have explained why TTA does not protect the information integrity of transaction requests between customer and bank with this scenario. An opening for man-in-the-middle attacks is present when a bank depends on a customer's computer for information verification by a customer.

3.4 Customer verified transaction set authentication (CVTSA)

In the previous section we have shown an example of a legitimate banking session being turned into an illegitimate session by a man-in-the-middle, represented by the banking customer's computer. In this section, we show an example of how CVTSA methods that are currently being introduced by banks cope with this. As noted in Section 3.2, the concept of WYseeIWYS is applied by signing a set of transaction requests. See Figure 3.1 on page 62 for an information flow overview.

The start of the scenario is similar to the scenario given in the previous section. We assume that some form of entity authentication already took place. Alice (A) is ready to use her computer (C) to transfer money from her account at her bank (B) to both Bob and Charlie. The difference in this scenario is that Alice now possesses a different authentication device provided by the bank with the name 'Reader' (R). Reader is an electronic device with a relatively large display, a keypad and a camera. It can be used to capture barcodes with information sent by the bank through the display of Alice's computer. The information includes critical values of transaction requests and the verification code to be read by Alice from the display of Reader. If Alice deems the transaction requests received by the bank valid, she can enter the verification code in her computer to be forwarded to the bank.

- (1) $A \rightarrow C : D_1, S_1$
- (2) $C \rightarrow B : D_1, S_1$
- (3) $A \rightarrow C : D_2, S_2$
- (4) $C \rightarrow B : D_2, S_2$
- (5) $B \rightarrow C : D_1, D_2, S_1, S_2, E_K\{ H(D_1, D_2, S_1, S_2, N), N, V \}$
- (6) $C \rightarrow R : D_1, D_2, S_1, S_2, E_K\{ H(D_1, D_2, S_1, S_2, N), N, V \}$
- (7) $R \rightarrow A : D_1, D_2, S_1, S_2, V$
- (8) $A \rightarrow C : V$
- (9) $C \rightarrow B : V$

Alice starts by entering the required transaction request information in her computer, which forwards this to the bank (steps 1 to 4). The bank returns a barcode, which contains a message. The barcode is projected on the display of Alice's computer (step 5). The message contains both plain and cipher text. The plain text part consists of destination account numbers (D_1 and D_2) and amounts (S_1 and S_2) of the transaction requests. The cipher text $E_K\{ H(D_1, D_2, S_1, S_2, N), N, V \}$, encrypted with shared secret key K ⁹⁷, contains a digest (hash) of the transaction requests' critical values concatenated with a nonce ($H(D_1, D_2, S_1, S_2, N)$), the nonce itself (N) and a verification code (V).

Alice uses her Reader to scan the code (step 6). The Reader decrypts the cipher text information (digest, nonce and verification code) and creates a digest using the received plain text information and the nonce. A check is made whether the plain text was modified in-transit using the created and received digests. If both are equal, the plain-text values were not modified. In this example, the values are equal and authentication can therefore continue. If this would not be the case, the Reader would show an error and not allow authentication to proceed.

The Reader shows the transaction requests as received by the bank with a verification code generated by the bank to Alice (step 7). Alice checks whether the transaction requests are as she entered them. If they are, she reads the verification code from the Reader and enters it in her computer (step 8). The verification code is then sent to the bank (step 9).

Where this scenario differs with TTA as applied in Section 3.3 is that the bank relies on its own infrastructure to provide integrity of information that must be verified by Alice. The message in steps 5 and 6 only relies on Alice's computer to provide availability. While part of the message is plain text, its integrity is protected by a mandatory check by the Reader using the digest and nonce from the cipher text.

Mallory is again intervening. Her attack is similar to the used approach in

⁹⁷For this scenario, we assume that symmetric encryption is used and that secret key K is known by both B and R . An alternative would be the use of asymmetric encryption in which B uses a public key and R a private key from the same keypair.

Section 3.3.

- (1) $A \rightarrow C : D_1, S_1$
 $C \rightarrow M : D_1, S_1$
 $M \rightarrow C : D'_1$
- (2) $C \rightarrow B : D'_1, S_1$
- (3) $A \rightarrow C : D_2, S_2$
- (4) $C \rightarrow B : D_2, S_2$
- (5) $B \rightarrow C : D'_1, D_2, S_1, S_2, E_K\{ H(D'_1, D_2, S_1, S_2, N), N, V \}$
- (6) $C \rightarrow R : D'_1, D_2, S_1, S_2, E_K\{ H(D'_1, D_2, S_1, S_2, N), N, V \}$
- (7) $R \rightarrow A : D'_1, D_2, S_1, S_2, V$

Mallory changes the destination account number of the first transaction from D_1 to D'_1 between steps 1 and 2. It is not possible for Mallory to change D'_1 back to D_1 between steps 5 and 6 without the Reader reporting an error, since she cannot change the cipher text containing the information used to verify the plain text values. Alice can see that D'_1 is received by the bank, which allows her to abort the authentication and transaction requests by not entering the verification code in her computer.

This demonstrates both the strength and the weakness of CVTSA. The customer has the opportunity to check whether the bank received the correct information, unlike with TTA. Unfortunately, nothing stops the customer from skipping this check. The customer can just read the verification code from the display of the Reader and enter it in his or her computer without taking a look at the transaction request information.

Humans cannot be treated as machines. They take actions that may seem irrational, although they are perfectly justifiable from cognitive and social perspectives. With CVTSA, a customer can use validation information without paying any attention to transaction request information. This nullifies the added security of CVTSA and therefore seems irrational, but from cognitive and social perspectives it can make sense because skipping the validation is the shortest and easiest route to what the customer needs to accomplish (conduct payments).

3.5 Entered single transaction authentication (ESTA)

We noted how TTA methods which depend on the customer's computer for information integrity are flawed in Section 3.3. We also noted in Section 3.4 how CVTSA mitigates this flaw by relying on the attention span of the customer. In this section, we demonstrate how the use of What You Enter is What You Sign (WYenterIWYS) protects against session modifying attacks in a similar way to CVTSA without requiring additional effort from the customer.

See Figure 3.2 on page 63 for an overview. Alice (A) again wants to use her computer (C) to transfer money from her account at her Bank (B) to Bob and Charlie. The bank provided her with a Trusted Entry Pad (TEP) and a smart card (SC). We also assume for this scenario that Alice is already logged in the secure

banking environment by previously applying entity authentication. Therefore, the bank already knows that Alice authenticated the session with her smart card.

$$(1) \quad A \rightarrow TEP : PIN$$

$$(2) \quad TEP \rightarrow SC : PIN$$

Alice starts the first transaction request. Because this is the first use of the 'Pay' function, her smart card must be unlocked. She inserts her smart card in the TEP, chooses the function 'Pay' and enters her PIN on the device (step 1). The TEP forwards the unlock request with the PIN to the smart card (step 2). The PIN is valid and therefore the required functionality is unlocked.

$$(3) \quad A \rightarrow TEP : D_1$$

$$(4) \quad TEP \rightarrow SC : D_1$$

$$(5) \quad SC \rightarrow TEP : E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \}$$

$$(6) \quad TEP \rightarrow C : D_1, E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \}$$

$$(7) \quad C \rightarrow B : D_1, E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \}$$

The first transaction request Alice wants to enter is for Bob. The TEP asks Alice to enter a destination account number for the transaction request (D_1). Alice enters this using the TEP's keypad and reads her entry on the TEP's display while she is typing. Any typographical mistakes can be corrected using the keypad. Alice confirms her entry with a push on the OK button on the device (step 3). After Alice's confirmation, the TEP sends D_1 to the smart card (step 4).

A nonce (N_a) is generated by the smart card. The smart card concatenates its own identifier SC , D_1 (received earlier from the TEP) and N_a . A digest (hash) is computed over the concatenated values, represented by $H(SC, D_1, N_a)$. The smart card encrypts the digest and nonce with its private key $PrK(SC)$ and returns encrypted message $E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \}$ to the TEP (step 5).⁹⁸ The TEP sends both values (the destination account number in plain text and the encrypted message) to Alice's computer (step 6), which forwards both to the bank (step 7).

$$(8) \quad A \rightarrow TEP : S_1$$

$$(9) \quad TEP \rightarrow SC : S_1$$

$$(10) \quad SC \rightarrow TEP : E_{PrK(SC)}\{ H(SC, D_1, S_1, N_b), N_b \}$$

$$(11) \quad TEP \rightarrow C : S_1, E_{PrK(SC)}\{ H(SC, D_1, S_1, N_b), N_b \}$$

$$(12) \quad C \rightarrow B : S_1, E_{PrK(SC)}\{ H(SC, D_1, S_1, N_b), N_b \}$$

The TEP asks for the next value, which is the amount of money associated with the transaction (S_1). Use of the TEP by Alice and the communication between TEP and Alice's computer in steps 8 to 12 is similar to steps 3 to 7. Note that the digest of the message is calculated over SC , D_1 , S_1 and the new nonce N_b .

⁹⁸We apply asymmetric encryption in our example for information integrity and non-repudiation. For confidentiality, the data can in addition be encrypted with a public key from B or with a symmetric key K shared by B and SC . Alternatively, only symmetric encryption could be used to get confidentiality and integrity, but non-repudiation would be lost.

The transaction request for Bob is now received by the bank. For Charlie, Alice repeats steps 3 to 12 using Charlie's account number and the amount of money she wants to send to Charlie.

The bank performs an integrity check to determine whether received messages are valid. This is done after each critical transaction request value is received (after steps 7 and 12). Before the integrity check is started, the bank decrypts the signature using public key $PuK(SC)$. The bank knows which public key is appropriate based on earlier performed entity authentication, which identified the smart card.

For the integrity check, the bank starts by computing a digest. When a destination account number is received, the digest is based on the known identifier SC , the received D_1 and nonce N_a from step 7. When an amount is received, the computed digest is based on the known identifier SC , the previously received (step 7) D_1 and the received (step 12) plain text S_1 and nonce N_b . This binds the amount of the transaction to the destination account number and ensures that the messages from steps 7 and 12 cannot be used independently. The message is valid if the digest of the received message is equal to the digest that the bank computed.

Mallory (M) is again an attacking party and has full control over Alice's computer. She can see that Alice is transferring money to bank accounts of Bob and Charlie since D_1 , S_1 , D_2 and S_2 are forwarded as plain text by Alice's computer from TEP to bank.

$$\begin{aligned}
(6) \quad & TEP \rightarrow C : D_1, E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \} \\
& C \rightarrow M : D_1, E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \} \\
& M \rightarrow C : D'_1 \\
(7) \quad & C \rightarrow B : D'_1, E_{PrK(SC)}\{ H(SC, D_1, N_a), N_a \}
\end{aligned}$$

It is possible for Mallory to change each plain text value before it is sent to the bank (in this example, D_1 to D'_1 between steps 6 and 7). This attack fails when the bank decrypts the encrypted message and notices that the digest does not match the received input. Alice is not allowed to continue.

$$\begin{aligned}
(11) \quad & TEP \rightarrow C : S_1, E_{PrK(SC)}\{ H(SC, D_1, S_1, N_b), N_b \} \\
& C \rightarrow M : S_1, E_{PrK(SC)}\{ H(SC, D_1, S_1, N_b), N_b \} \\
& M \rightarrow C : S'_1 \\
(12) \quad & C \rightarrow B : S'_1, E_{PrK(SC)}\{ H(SC, D_1, S_1, N_b), N_b \}
\end{aligned}$$

If Mallory would only change the amount instead (e.g. she works together with Bob to get more money than Alice intends to give), then the intervention would look as shown between steps 11 and 12. Similar to the previous attack, the bank notices that the signature does not match the received value S'_1 and will not allow the transaction to continue.

To summarize, applying ESTA protects against man-in-the-middle attacks that modify critical transaction request information in a customer's session. This is similar to the added information integrity of CVTSA when compared to TTA, but differs in that the use of WYenterIWYS does not introduce a dependency on the customer to perform the required validation, unlike WYseeIWYS as applied by CVTSA.

3.6 Formal verification

Due to time constraints it was not possible for the authors of the paper on which this chapter is based to formally verify the steps as described in Section 3.5. Fortunately, a student from Radboud University was able to perform this step. Safet Acifovic used the protocol verification tool known as ProVerif⁹⁹ and formally defined and verified the steps as a protocol with it [Aci15].

Based on the model that he made, it could not be fully proven that all relevant security properties are assured for the transfer of the data as depicted in steps 6, 7, 11 and 12 of Section 3.5). Authenticity and integrity of the transaction data is assured, but the protocol is vulnerable to replay attacks. An adversary that has control of the customer's computer has the option to capture an intercepted set of messages and replay them within the same session, which the bank (when fully complying to the protocol) would interpret as multiple transactions with the same destination account number and amount of money. This could be mitigated by a server-side check by the bank, but this would be outside of the protocol definition. Furthermore, Safet concluded that the protocol is susceptible to denial of service attacks since a client computer can drop messages from and to the bank.

Safet did not have time left to formally verify non-repudiation as a security property. He expects that non-repudiation of the origin (the TEP) is easy to prove due to the use of a signature. Non-repudiation of receipt (by the bank) would be more difficult to prove. He notes that this might be added to the protocol, which could make ESTA capable of achieving full non-repudiation.

3.7 Related work

Several characteristics of the TEP are already represented in existing authentication methods. We note several examples and how they relate to the TEP.

3.7.1 Devices that apply keyboard emulation

Keyboard emulation can be used to transfer information from one electronic device to another by applying a hardware interface and a protocol used by keyboards (e.g. PS/2 or USB HID). The receiving device does not require changes to hardware or device drivers to facilitate the communication. An example of an existing entity authentication device which utilizes this is Yubico's YubiKey [KS13].

We do not specify the interface between TEP and customer computer (see Figure 3.2 on page 63, step 2). Since communication is modeled as unidirectional from TEP to the customer's computer, keyboard emulation can provide a software independent bridge between these devices. In this case, the implementation of client-side device drivers is unnecessary, which is beneficial for both banks (lower implementation costs) and customers (less installation time).

3.7.2 Interactive smart card terminals

This category includes devices which apply their own user interface (keypad and display) while connected to a computer and that depend on a smart card for cryptographic functions. An overview of the discussed devices, their similarities and their differences is given in Table 3.1. We note two banks from our former work that apply this to authenticate their customers [KSDC⁺14]. Nordea Bank (Nordic countries) allows its corporate customers to connect their card reader to a client computer using USB.¹⁰⁰ The client can use the card reader after software is installed (provided by the bank). Before messages can be signed by the smart card, its functionality must be unlocked by entering a PIN on the reader. Information for the customer to sign is shown on his or her computer. ABN-AMRO (the Netherlands) is a bank that uses a similar card reader, named the E.Dentifier2. This device differs from Nordea’s card reader by showing information to sign on the device itself instead of on the customer’s computer. An older version of the E.Dentifier2 authentication device had a notable security flaw, which was fixed in a later version [BdKGP⁺12].

The FINREAD Card Reader is a bank independent example [HKW06]. It also features a smart card reader and a connection to a customer’s computer. A difference with the previous examples is that the reader itself also hosts cryptographic credentials and functions together with user installable applications from different providers. Communication between reader and provider is secured in terms of confidentiality and integrity. One recognized weakness of FINREAD Card Reader is the high cost required to produce the device [SH04, HPN10]. The Radboud Reader is another interactive smart card terminal [PdR13], which by itself is less complex compared to FINREAD. Complex functionality and control is instead moved to the smart card.

Rabobank (the Netherlands) announced a new authentication device with the name Rabo Scanner.¹⁰¹ It allows one-way communication without the installation of additional software by displaying a color code on the customer’s computer, which is scanned using a camera on the Rabo Scanner. The color code contains the in-

⁹⁹ProVerif: <http://prosecco.gforge.inria.fr/personal/bblanche/proverif/>

¹⁰⁰About Nordea’s card reader: <http://www.nordea.com/Our+services/International+products+and+services/Corporate+Netbank/Nordea+card+reader/1079602.html>

¹⁰¹Rabobank introduces the Rabo Scanner (Dutch): http://www.rabobank.nl/particulieren/servicemenu/nieuws/rabobank_nieuws/rabobank_introduceert_de_rabo_scanner

	Transaction authentication				Requires client-side software
	Type	Secure entry	Verification by user	Data flow	
Nordea	TTA	×	×	↔	✓
ABN-AMRO	CVTSA	×	✓	↔	✓
FINREAD	CVTSA	×	✓ ^b	↔	✓
Radboud	CVTSA	×	✓	↔	✓
Rabobank	CVTSA	×	✓	←	×
TEP	ESTA	✓	×	→	×

Table 3.1: Comparison of interactive smart card terminals.

^{a b} Based on default use scenarios.

formation to be verified, which is shown on the display of the Rabo Scanner after it is scanned. For confirmation, the customer reads a verification code from the device and enters it in the bank's site on his or her computer.

The TEP has a very minimalistic approach compared to the other discussed authentication methods while still providing the ability to verify and sign transaction requests separately from the customer's computer. It is not required for customers to install client-side software. Unlike the CVTSA examples, it is also not required for the customer to verify information received by the bank. Instead, the process of protecting the integrity of information is initiated by the customer and completed by the bank, which does not require a round trip of the same information for verification.

3.7.3 Mobile devices used for out-of-band verification

A customer owned smartphone and its connection to the Internet can be used as an out-of-band channel to allow a customer to verify information from a banking session with a desktop computer.¹⁰² This can apply WYseeIWYS if information to sign can be interpreted in a single semantic context. Drawbacks include the previously mentioned limitation of WYseeIWYS (customers must perform additional actions adequately for effective security), but also that the customer's smartphone is an untrusted device that is vulnerable to malware [Law08, FFC⁺11].

ESTA can potentially be implemented on a smartphone, with the caveat that the platform is not as trustworthy as a bank provided device. While mobile applications can be hardened against malware and other software-based threats¹⁰³, malware threats cannot be prevented in an untrusted environment.

3.7.4 Authentication solutions in trusted execution environments

A trusted execution environment (TEE) is a collection of resources and controls of those resources that are physically or logically separated from other resources on the same device [VOZ⁺12]. Such resources can include volatile memory space, persistent storage space, CPU cycles, security functions and different types of in- and output interfaces. Other resources outside the TEE cannot interact with any of the resources that compose the TEE unless explicitly permitted by the TEE. A TEE can attest its identity and allows authorized remote parties to interact with applications within the secure environment.

The principles of TEEs potentially allow the integration of the TEP's functions into a customer's computer if it has a TEE. Requirements of a TEE to host a TEP are that the user knows whether he or she works within the normal or the trusted environment and that in- and output interfaces are secure against injection attacks.

¹⁰²An example of a software product which provides this is Entersekt's online banking authentication: <http://www.entersekt.com/>

¹⁰³An example of a framework which allows mobile application hardening is Versafe's MobileSafe: <http://www.versafe-login.com/?q=mobilesafe>

3.8 Further research

We give a high-level description of ESTA Section 3.5 to showcase its principles. While ESTA offers information integrity of transaction requests between customer and bank by applying WYenterIWYS, it does not reduce the effectiveness of social engineering attacks on the customer. Further research can answer what would be required to protect against social engineering (e.g. phishing attacks) in addition to the use of ESTA. It might be possible that unambiguous labels for functions (e.g. 'Login' and 'Pay') and requested information (e.g. 'Destination account number' and 'Amount of money') increases the awareness of customers. This is something that can be tested in addition to the behavior of customers when warnings are (repeatedly) observed (e.g. 'The use of this function WILL cost you money').

The TEP is introduced as a technical concept to reduce the effectiveness of malware attacks on banking customers' computers. Injection attacks which add or replace financial transactions can be detected by protecting the integrity of the information flow between customer and bank. While we kept the question about whether banking customers can use this in the back of our minds, the technical concept has not been tested for usability. It is possible that changes have to be made to make WYenterIWYS acceptable in everyday use. The use of two input devices (a TEP for the entry of simple but critical values and a regular keyboard for the entry of non-critical and possibly more complex values) might confuse customers.

There are different possible approaches to implement the unidirectional communication between TEP and the customer's computer. Comparing approaches and their (dis)advantages would have to take the possible interfaces of customer devices and their prerequisites for use into account.

Information to be entered in the TEP by the customer itself can also present a challenge if the information next to digits and a decimal separation character also contains letters. A full destination account number can contain letters if it is an International Bank Account Number (IBAN).¹⁰⁴ A common keypad might prove too cumbersome to enter letters. A full keyboard can also present challenges regarding usability on a small form factor. User input methods can be examined for the best fit between user entry and critical information to enter. Also, the possibility of an increase in insecure user behavior through typographical errors on an external device (ESTA) can be compared against the possibility of insecure user behavior when the user is required to compare values (CVTSA).

Another point for further research is the improvement and new formal verification of the steps as described in Section 3.5 as a communication protocol. As noted in Section 3.6, Safet Acifovic verified different security properties and there is definitely room for improvement, mostly related to replay attacks as an attack vector and to adding (and verifying) full non-repudiation as a security property.

¹⁰⁴European Committee for banking Standards - IBAN: Standard Implementation Guidelines (SIG203 V4): http://www.pruefziffernberechnung.de/Originaldokumente/IBAN/SIG203V3FV_181200.pdf

3.9 Concluding remarks

In this chapter we introduced Entered Single Transaction Authentication using the Trusted Entry Pad, which applies the new concept of What You Enter Is What You Sign to verify customer entered data without the need for an extra verification step by the customer. It has a smaller margin for customer errors compared to What You See Is What You Sign-based approaches while still being independent from the customer's computer for information integrity, unlike traditional transaction authentication methods.

Chapter 4

User-friendly manual transfer of authenticated online banking transaction data

Abstract

Online banking relies on user-owned home computers and mobile devices, all vulnerable to man-in-the-middle attacks which are used to steal money from bank accounts. Banks mitigate this by letting users verify information that originates from these untrusted devices. This is not very usable since the user has to process the same information twice. In addition, it makes the user an unnecessary critical factor and risk in the security process. This chapter concerns a case study of an information scheme which allows the user to enter critical information in a trusted device, which adds data necessary for the recipient to verify its integrity and authenticity. The output of the device is a code that contains the information and the additional verification data, which the user enters in the computer used for online banking so it can be forwarded to the bank. With this, the bank receives the information in a secure manner without requiring an additional check by the user, since the data is protected from the moment the user entered it in the trusted device. This proposal shows that mundane tasks for the user in online banking can be offloaded to computers, which improves both security and usability.

4.1 Introduction

User-friendly is a term associated with systems that offer a high usability level without the requirement for technical knowledge to disclose a system’s functionality. Security and usability are often seen as opposites. It is easy to sacrifice one in order to improve the other, but hard to improve one without affecting the other negatively. The work in this chapter has the main goal to, in a very specific area, improve usability without reducing security. This area is the secure creation of financial transactions by users in online banking.

Our contribution is an analysis and a case study of a process that allows users to transcribe information and data required to verify the integrity and authenticity of the information, using a code. The case study uses the information from the analysis to create an alternative online banking transaction authorization scheme which, when compared to currently used schemes, relies less on the user to perform critical actions. While the main focus of the case study is on improving usability, it inherently also improves security since users have less room to make mistakes when they have to make less security decisions.

Home computers are vulnerable to Man-in-the-Browser (MitB) attacks [CD12] while mobile devices, such as smartphones and tablets, are not exempt from malware attacks either [FFC⁺11]. These user-owned devices are seen as untrusted environments by banks, since adversaries can intervene in the communication flow between user and bank when they are used for online banking. Banks often rely on small devices given to their customers to have a trusted environment at the user’s side [PdR13, KSDC⁺14]. These devices are used for authentication to the bank itself (using one-time passwords or challenge-response authentication), and for the verification of transaction data. This work focuses on improving the latter.

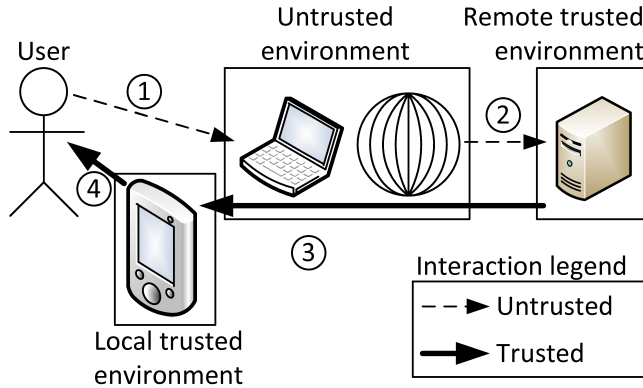


Figure 4.1: The information flow between environments with the What You See Is What You Sign authentication information scheme.

Figure 4.1 gives an overview of the What You See Is What You Sign (WYseeI-WYS) information scheme, currently being used by banks to process transaction data. In step 1, the user enters transaction data in the computer used for online banking, which sends it to the bank using the Internet in step 2. Note that from the perspective of the bank the origin of the transaction is an untrusted environment,

and therefore all the steps up so far are also untrusted: the bank cannot know for certain whether step 1 was actually performed by the user and whether step 2 concerns information from the user. To verify whether it was actually the user and not an adversary which offered one or more transactions, in step 3 the most important transaction information and a one-time password are returned over a secure channel to the bank authentication device in possession of the user. The user states whether he or she did or did not enter these transactions earlier in step 1. A confirmation is given by repeating steps 1 and 2 with the one-time password. Note that only the user can know the one-time password and can authorize the transactions or not. Returning the one-time password through an untrusted environment is therefore not insecure. An aspect of WYseeIWYS that is neither secure nor user-friendly is that the user is required to verify the same information twice: once upon entry (step 1) and once when it is returned (step 4). Users are quite unreliable in comparing numbers [AAJM08], and mistakes (genuine due to simply not seeing changed numbers, or abusive due to laziness) can be expected.

The authors of the paper on which this chapter is based proposed an alternative information scheme named What You Enter Is What You Sign (WYenterIWYS) [KVvE14], as discussed in Part II, Chapter 3. With WYenterIWYS, the transaction data is entered immediately in a trusted environment before it is forwarded to the bank in a secure manner, which makes the second verification that WYseeIWYS has unnecessary. Their proposal misses an implementation which describes how the critical transaction data entered by the user is transferred from a trusted environment to an untrusted environment, which will forward it to the bank. They suggest the use of a connection between the authentication device and the computer used for online banking. Requiring a connection between the trusted authentication device and the untrusted environment is troublesome. For security, it widens the attack surface of the local trusted environment, since it will be more exposed to the untrusted environment. In addition, online banking can be done with a plethora of different devices. The authentication device might not be capable of creating a connection to the computer used for online banking due to the lack of a compatible interface.

The challenge therefore is to apply WYenterIWYS in a user-friendly way. We

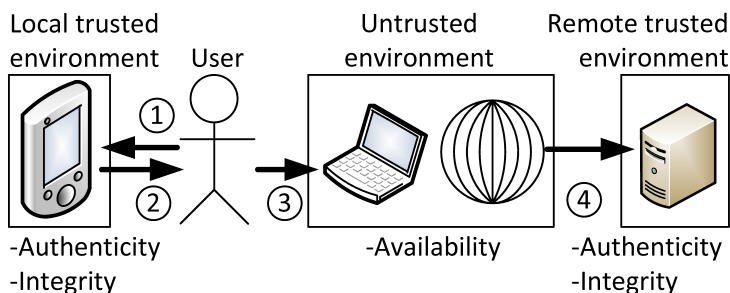


Figure 4.2: An overview of the path the transaction data follows (trusted from the very first step), and the information security principles for which each environment is responsible.

propose the use of a code to let the user manually transfer the information. See Figure 4.2 for an overview. In step 1, the user starts with entering the critical transaction data in a trusted authentication device. Our contribution is the analysis and a case study of a method that allows users to manually transfer authenticated data using a code from a trusted authentication device to an untrusted environment (steps 2 and 3), which forwards the data to a remote trusted environment (step 4). Integrity and authenticity of the data is protected as it passes through the untrusted environment.

We refer to the code as a Message Code (MC), since it contains both the message (the critical transaction data) and data to validate its integrity and authenticity. Using a code that is manually transferred by the user has several security and usability advantages. The one-way information flow of reading a code from a display and writing it on another device physically and logically separates the local trusted environment from the untrusted environment, reducing the attack surface of the former. Since a connection between devices is not necessary, compatibility issues are non-existent as any untrusted user-owned device can be used.

The critical aspect of the MC is in its construction, which determines whether the resulting code is user-friendly to transfer from one device to another. If the MC would be too long or consist of characters that are hard to read or enter, authorizing a transaction would be quite a hassle for the user. While it is a security and usability advantage that the user does not have to make a second critical decision anymore as with WYseeIWYS, the actions that the user has to perform to apply WYenterIWYS using an MC should not introduce new frustrations.

In Section 4.2, the steps and methods that can be used to generate an MC in a user-friendly and secure manner are analyzed. The case study for an implementation of an MC to secure financial transaction online banking is described Section 4.3. Section 4.4 was not part of the original article on which this chapter is based. It is an addition that is dedicated to the formal verification of the steps described in the section before it as a communication protocol. Furthermore, in Section 4.5 we reflect back on the case study, limitations of our work and possible further research. Section 4.6 closes with our concluding remarks.

4.2 Analysis of steps and methods to generate an MC

There are scenarios in which it is necessary for a service provider to know whether a specific user provides a certain piece of information, and that the information was not altered along the way (either accidentally or by an adversary). The amount of resources spent on this depends on the value of the information and how often the information flow occurs. For example, if an individual needs to register for a service that needs certainty about the user's identity, it is not enough to just assume that the named individual is actually the one he or she claims to be. If such a registration is only done once, it would be worthwhile to have the user visit a branch office of the service provider and provide identification documents, such as a passport. In this scenario, the branch office and the person assisting the user (both under control of the service provider) and the passport (provided by the government) ascertain the

identity of the individual.

However, such physical interaction is unwanted if it concerns valuable information that the user sends more often to the service provider. An example is given by online banking. Users expect that they can make transactions anywhere at any-time. A trusted (from the perspective of the bank) environment is required that lets users provide information securely. There are two distinct trusted environments: the local trusted environment available to the user, and the infrastructure of the service provider. Between these environments is a large untrusted environment, where Man-in-the-Middle (MitM) attacks occur. These attacks do not only occur on the Internet (against which SSL/TLS can provide adequate protection), but also on users' computers. An example are Man-in-the-Browser (MitB) attacks, through which an adversary retrieves authentication credentials or silently changes information between user and service provider [CD12]. If users' devices cannot be trusted, the most obvious approach would be to provide each user a device that hosts a local trusted environment. Users should be able to use this device to provide required information to the service provider in a secure manner.

Weigold and Hiltgen (2011) proposed several online banking transaction authentication methods [WH11]. For one of their proposals, the user enters the same transaction details in an untrusted computer used for online banking and in a trusted device, provided by the bank and in possession of the user. The trusted device generates a 'Transaction-dependent Authentication Code' (TAC), which the user enters on the untrusted computer used for online banking. The TAC from their proposal is created based on the information entered on the trusted device, and protects the information entered on the untrusted device. The information, entered twice by the user, should be equal to correspond with a valid TAC.

What we propose and analyze further in this section is the use of a Message Code (MC). An MC not only contains the information required to verify the integrity and authenticity of a message, but also the message itself.

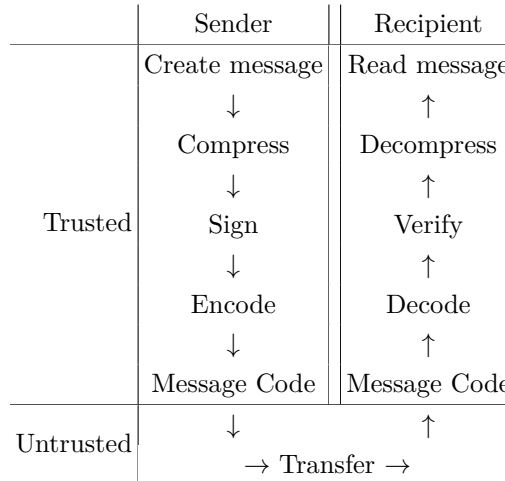


Figure 4.3: An overview of the process required to create a Message Code.

Figure 4.3 shows two trusted environments, and the different actions required for

the sender to send a message securely and for the recipient to receive the information securely using an MC. It is assumed that each party is the only one which can control his or her own trusted environment. The paired intermediate steps related to the processing of the message by both parties are explained in more detail.

4.2.1 (De)compression algorithms

If applied correctly, compression reduces the size of messages. A message will be part of the MC. Reducing its size would imply a reduction in the MC's size, which is beneficial when the user has to transfer the information later.

For compressing a message, an existing general-purpose algorithm can be used. An advantage of these algorithms is that they take a lot of work out of hand, since it is not required to design a new algorithm. General-purpose algorithms can work with any kind of content. This is also a disadvantage, since a general-purpose algorithm might compress data less or not at all, compared to an algorithm that is tailored towards the to-be compressed content. Worst case, a general-purpose algorithm does not compress the content at all since there is nothing to compress, and its output size is increased instead of decreased due to the overhead of the algorithm.

Number of characters	Size (bits)	Compressed size (bits)	Number of characters	Size (bits)	Compressed size (bits)
5	40	40-56	18	144	128-160
6	48	40-64	19	152	136-168
7	56	48-72	20	160	136-176
8	64	48-80	21	168	144-184
9	72	64-88	22	176	160-192
10	80	72-96	23	184	160-200
11	88	72-104	24	192	176-208
12	96	80-112	25	200	184-216
13	104	96-120	26	208	192-224
14	112	104-128	27	216	200-232
15	120	104-136	28	224	200-240
16	128	112-144	29	232	216-248
17	136	112-152	30	240	224-256

Table 4.1: Possible output sizes of the Deflate compression algorithm, based on 10,000,000 randomly generated text values with a character range of [A-Za-z0-9], encoded in 8 bit ASCII.

Table 4.1 shows an overview of the effect that the Deflate general-purpose compression algorithm has on random text of different sizes. For such small pieces of data a general-purpose compression algorithm can have either a positive or a negative effect, and are therefore unreliable. In the most optimal circumstance (in very specific cases of having 8 characters of data), Deflate¹⁰⁵ compresses data by 25%. Depending on the number of characters, the output can actually have an increased

¹⁰⁵RFC 1951 - DEFLATE Compressed Data Format Specification version 1.3: <https://tools.ietf.org/html/rfc1951>

size of between 6.3% (30 characters) and 28.6% (5 characters). This behavior can be explained by that general-purpose compression algorithms such as Deflate build a dictionary of data to compress based on earlier compressed data. Therefore, small pieces of data do not compress really well (or in some cases, at all) with such algorithms.

An algorithm that specializes more on the compression of very short pieces of text could be used instead, and can be created by using a static dictionary of commonly used text parts.¹⁰⁶ However, the amount of compression it provides can be unreliable, depending on how well the message ‘fits’ the pre-established dictionary.

A completely custom algorithm can be designed instead, tailored to a specific message structure. A message could have redundant information. When removed, this information can be inferred from either the syntax or the semantics of the remaining message. Removal would be compression, while recovery and reconstruction would be decompression. It is important that both the sender and the recipient respectively possess compression and decompression implementations of the same algorithm. For the algorithm to be durable, the messages’ structure should not change.

4.2.2 Securing and verifying authenticity and integrity

The recipient of an MC needs to be able to establish the authenticity of the source of the message, and to verify that the message was not changed after it was sent. For this, the recipient must make some inferences on the data based on previous agreements with the sender. Therefore, the sender has to prepare the data in such a way that the recipient will be able to perform the necessary checks. There are several ways in which the sender can make these preparations, and for the receiver to verify the message.

Digital signatures are an option. Using public key cryptography, the integrity and authenticity of a message can be established by adding a signature to the message. In addition, digital signatures provide non-repudiation: the sender cannot claim that he or she did not send a message that the recipient has received and validated. However, a drawback of digital signatures is that key management is quite complex.

Message authentication codes (MAC) are another option. A MAC is data added to a message from which inferences can be made about its integrity and authenticity, similar to a digital signature. Creating a MAC requires the data of which the integrity and authenticity should be protected (the message), and some secret data which provides the authenticity (a secret key). Verifying a MAC requires the payload to be verified and a secret key known to the sender and the recipient. A difference with digital signatures is that a single key is used. This simplifies key management somewhat since all involved parties (sender and recipient(s)) use the same key, but it sacrifices non-repudiation since it cannot be proven which party sends which message.

With MACs, it is initially required that the trusted environment that generates the key sends it to the other trusted environment, without an intermediate untrusted environment. Otherwise a man-in-the-middle could intercept the key. A public key infrastructure with trusted third-parties that issue certificates to create digital

¹⁰⁶An example is SMAZ, compression for very small strings: <https://github.com/antirez/smaz>

signatures might be a better choice if it is not viable to securely transfer a secret key from one trusted environment to the other at the beginning of the authentication device's lifespan.

Whether a digital signature or MAC is used, it is important to provide protection against replay attacks. A cryptographic nonce (number used once) can provide protection against such attacks if it is used to generate the digital signature or MAC. Note that a nonce does not have to add data to the message itself as long as the remote trusted environment is able to reconstruct the nonce when verifying the message. Examples of such nonces include time stamps and counters.

4.2.3 Encoding/Decoding methods

Compressed data, digital signatures and MACs often do not consist exclusively of human readable data. An encoding can convert data to human readable and writable text. There are several approaches. Wiseman et al. (2016) performed a comparison of three distinct encoding schemes for one-time passwords used in device pairing [WMC⁺16]. These will be referred to as Wiseman's word encoding, Wiseman's alphanumerical encoding and Wiseman's numerical encoding. Each encoding was tested on its efficiency and perceived usability when transcribed by users on home computers and mobile devices. Wiseman et al. set a lower limit on the length of the codes of 500 million possible combinations for each tested encoding scheme. In addition, codes were padded when required to always give a fixed length.

For Wiseman's word encoding, the fixed length is 3 words of 3 letters. An index of 800 words was used. Each of the three words in a code represents approximately 9.644 bits since $\log_2(800) \approx 9.644$.

For Wiseman's alphanumerical encoding the fixed length is 5 alphanumeric characters using a set of 56 different characters from the range [a-zA-Z0-9], excluding 'i', 'o', '1', 'l', 'O' and 'L'. Each character in the code represents approximately 5.807 bits per character ($\log_2(56) \approx 5.807$). There are also other alphanumerical encoding schemes. Base64 is probably the most famous encoding scheme that outputs 'real' characters exclusively (that is: characters that most humans can see and interpret, assuming that they are familiar with the English alphabet), and each of its 64 characters represents a 6 bit value.¹⁰⁷ Base32 is a variation which uses 32 characters instead and avoids the use of special and mixed case characters by only using upper case letters and some digits. Each of Base32's characters represents 5 bits. The same binary data encoded in Base32 will be represented by more characters compared to when it is encoded in Base64. A variation of Base32 is Z-Base 32.¹⁰⁸ The most important difference is the choice of characters, which is lower case instead of Base32's use of upper case characters, and characters are chosen in a specific order that makes them easier to distinguish.

Finally, there is Wiseman's numerical encoding, which presents all data as a base 10 number with a fixed length of 9 digits. Each digit represents approximately 3.322 bits [Buc59].

¹⁰⁷RFC 4648 - The Base16, Base32 and Base64 Data Encodings: <https://tools.ietf.org/html/rfc4648>

¹⁰⁸Human oriented base-32 encoding - O'Whielacronx (2009): <http://philzimmermann.com/docs/human-oriented-base-32-encoding.txt>

Wiseman et al. concluded that words are the easiest to transcribe. It must be noted that they were only interested in using an encoding scheme to create a one-time password. This can be compared to the security information required to validate the integrity and authenticity of data, as discussed in the previous section. However, an MC will contain both the data itself and this security information, implying that its length will be longer compared that of a one-time password in any encoding scheme. This is something that should be kept in mind when making a decision on what kind of encoding should be used.

The output of the encoding phase is an MC. Its characters can be grouped to make them easier to transcribe. For example, words can be separated by spaces and characters of alphanumerical codes can be separated by dashes.

4.2.4 Code transfer

After an MC is made, it needs to be transferred from the trusted environment. The MC is meant to reach another (remote) trusted environment. Different pathways can be taken, and most will use an untrusted environment (as shown in Figure 4.3). If that is the case, it is expected that the untrusted environment forwards the code to another trusted environment. The untrusted environment provides availability while the MC provides integrity and authenticity.

The MC can be read by the user from one (trusted) device and entered in another (possibly untrusted). To make this process user-friendly the MC should be as short as possible and consist of characters that are easy to read and write. The compression phase mostly dealt with the MC's length, while the encoding phase focused on the data that is actually to be transferred by the user.

Typographical mistakes can be detected by the recipient due to that the code contains both the message and additional data to verify its integrity and authenticity. When a mistake is made, the message will not correspond with the additional data when it is verified. The recipient refuses to process the message further and notifies the user to correct his entry. This further improves user-friendliness, since the user can make and correct mistakes without repercussions.

This chapter focuses mostly on a human-transferable MC since it is the most universal method to transfer information from a trusted device to any untrusted device that allows user input, independent from used software that the latter runs. However, user-friendliness can be improved without sacrificing security in specific scenarios, depending on the used untrusted device. Most smartphones have a rear-facing camera which can be used to scan QR codes. An MC could be converted to a QR code that can be scanned by an application that forwards the data to the trusted environment). A QR code would aid in cases where user input is less user-friendly, such as with touch keyboards on smartphones [GB15].

4.3 Online banking case study

This section notes a case study for the feasibility of using a Message Code (MC) for transaction authentication in online banking using the methods described in Section 4.2. It first explains why the use of an MC can be beneficial for security

and usability in online banking before describing each step of its generation and verification.

A well known information scheme in online banking is What You See Is What You Sign (WYseeIWYS), of which an overview is shown by Figure 4.1. As discussed in the introduction, WYseeIWYS lets users verify transaction information in a trusted environment that a bank received previously from an untrusted environment. Therefore, the user has to verify transaction information twice: once when entered (step 1) and once when received (step 4).

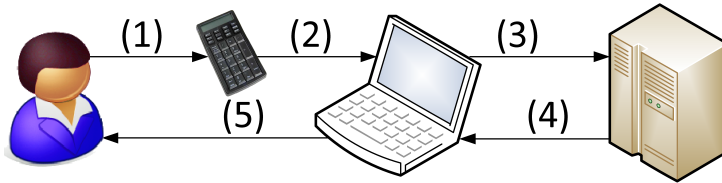


Figure 4.4: An overview of the What You Enter Is What You Sign information scheme.

The user's time should be used sparingly for security actions [Her09], so it might be beneficial if the user would not have to process transaction details twice. The authors of the paper on which this chapter is based proposed an alternative to WYseeIWYS with the name What You Enter Is What You Sign (WYenterIWYS) [KVvE14] (on which Part II, Chapter 3 is based). An overview of the original proposal is given by Figure 4.4. The user enters critical transaction data (the destination account number and the amount) in a trusted device in step 1. The trusted device adds a digital signature and enters all data in the user's computer in step 2. The user's computer forwards the data to the bank in step 3. After verifying the signature, the bank reports the validity of the entered values back to the user in steps 4 and 5. The idea behind WYenterIWYS is that a man-in-the-middle will not be able to change the data between steps 2 and 3 without the bank noticing it, since the digital signature ensures the integrity and authenticity of the entered critical transaction data. As shown in Figure 4.4, the secure information flow is one way only and the user is not expected to perform any checks afterwards.

The proposal does not specify which technology should be used in step 2. Keyboard emulation was suggested as one possible method, but this does require a connection and assumes that the user's computer (the untrusted device) actually supports keyboards. Connecting devices (wired or wirelessly) can also be quite cumbersome.

An MC could be used for transferring critical transaction data in a way that reduces compatibility issues. Figure 4.5 shows the information flow if the user would facilitate the transfer of the MC between both devices. In step 1, the user enters critical transaction information in the trusted device, which creates an MC in step 2 that is shown to the user in step 3. The user enters the MC into his home computer or mobile device in step 4. A browser or application receives the MC in step 5 and forwards it to the bank in step 6. The bank processes the MC in step 7 and returns the resulting data back in step 8 if the MC is valid. The data is processed by the user's computer in step 9 and shown to the user in step 10. If the MC is invalid, an

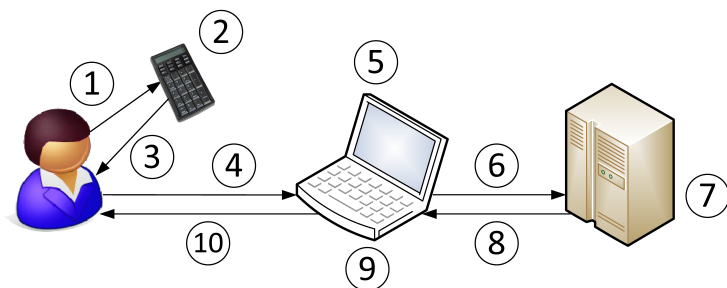


Figure 4.5: Overview of the information flow in the WYenterIWYS manual method, with a short description of each action.

error would be returned through steps 8, 9, and 10, after which the user can correct any typographical mistakes made in step 4.

This section continues with the considerations of constructing an MC in step 2 and the deconstruction in step 7. Looking at Figure 4.3, the authentication device in possession of the user would be the trusted environment available to the sender, the trusted environment of the recipient would be the bank's infrastructure, and the untrusted environment between them would be represented by the user's computer and the Internet. Therefore, the authentication device has to be used to create the message and compress, sign and encode it to create an MC, while decoding, verifying the signature and decompression would have to be done by the bank.

4.3.1 Create the message

We consider the destination account number and amount of money to be critical transaction information. Both are entered by the user in step 1 in the authentication device.

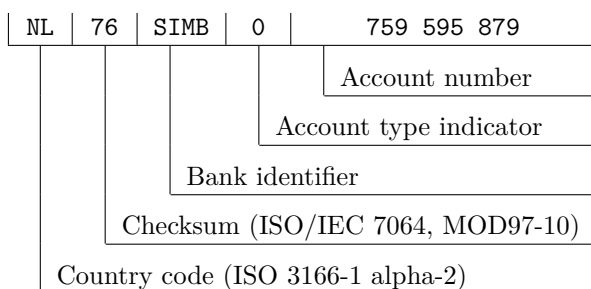


Figure 4.6: The structure of a Dutch IBAN account. The example IBAN account number is at the fictional **SIM**ulation **B**ank and has a valid checksum.

For the example, we assume that a Dutch IBAN is used as a destination account number. The structure of a Dutch IBAN is shown with an example in Figure 4.6. The country code is an ISO 3166-1 alpha-2 code and always has the same location and length for all IBANs. The checksum also always has the same location and

length. For Dutch account numbers, the bank identifier consists of four uppercase letters, the account type indicator is 0 for payment accounts and the account number is always nine digits.

The amount of money is a value with two decimal digits. For our proposal, we assume that a transaction has a maximum value of less than one million euro (so a maximum of € 999,999.99). We also assume that this method will only be used for domestic transactions (from a Dutch account to another Dutch account), so the user does not have to specify an alternative currency. An example value the user could enter is:

123456.78

To enter all values fully, the authentication device would require a keyboard which allows the user to enter [A-Z0-9] and possibly a decimal separator. Typographical mistakes in the IBAN can be detected by the device by verifying the checksum. Such a check is not possible on the amount of money, but the user can notice typographical mistakes when shown on the screen of the authentication device, and later (after step 10) on the display of his or her computer.

4.3.2 Create the Message Code

For step 2, we define several phases to generate an MC from the data entered in step 1.

- A compression phase, to initially compress the data the user entered. This allows the MC to be shorter.
- A signature phase, to generate data to protect the authenticity and integrity of the message when transferred from authentication device to bank.
- An encoding phase, which creates a string value from the data which a user can process.

Compress

There are several approaches for compression noted in Section 4.2.1. Data in online banking transactions is quite structured and well defined. Therefore, an example is given for a custom algorithm, tailored towards the data to be transferred. The functions described in Section 4.2.1 will be used: removal and reconstruction of redundant information, and restructuring data types for more efficient transportation.

To start, we use the IBAN from the example in Figure 4.6 and an amount of € 123456.78:

NL76SIMB0759595879 12345678

We removed the decimal separator from the amount to make it an integer value, which reduces complexity for further processing.

We already stated that we will only use this method for domestic transfers, so the country code can be removed. The checksum in the next two digits of the IBAN is only meant to protect against typographical mistakes. It has already been verified

	Message												MAC			
	Destinat. account: 759595879 (30 bits)						Amount: 12345678 (27 bits)						0x7764ce (23 bits)			
Bits	10110101000110100000110110011100010111000110000101001110111011101100110011001110															
Decimal	22	20	13	0	27	7	2	30	6	2	19	23	14	25	6	14
Z-Base 32	s	w	p	y	5	8	n	6	g	n	u	z	q	3	g	q

Table 4.2: Creating a single human readable string from the message and the MAC using Z-Base 32.

during entry in step 1, which is why we can also remove it. As discussed earlier, a Dutch IBAN has a bank identifier consisting of four upper case letters. Banks in the Netherlands can reconstruct the destination bank based on solely the account number, which is why we can also remove it. Finally, the leading 0 in front of the bank account number specifies the type of account. Any other number would be a savings account. Savings accounts do not support direct withdrawals or deposits from payment accounts owned by other account holders, which is why we can also remove this digit. This leaves us with:

~~NL 76 SIMB 0~~ 759595879 12345678

The striked through data has been removed, and needs to be reconstructed in step 5.

Dutch account numbers have a maximum value of 999,999,999, which can be represented as an unsigned integer by 30 bits.¹⁰⁹ The amount (as an integer) has a maximum of 99,999,999, and can be represented as an unsigned integer by 27 bits. In binary, account number *an* and amount *am* are as follows (most significant bit first):

$$\begin{aligned}
 an &= 101101010001101000001101100111 \\
 am &= 000101111000110000101001110
 \end{aligned}$$

Since the length of bits for each value is fixed, the values can be concatenated to create a single value that can be split again in step 7 by the bank. This concatenated value is the raw message which must be processed further.

Sign

For our example, there are only two parties involved who need to access key material: the authentication device (to create data for verifying the integrity and authenticity of the message) and the bank (to perform the verification). Based on the various methods noted in Section 4.2.2, a fitting approach would be the use of a MAC. It can be assumed that the bank can embed a shared secret key in the authentication device in a secure environment before it is given to a customer, and there are no other parties involved which would warrant the use of a public key infrastructure.

To prevent replay attacks, the MAC should be based on both the message and a nonce. The nonce itself is not secret, but should be unique and fresh for each MC. We assume a nonce is used that does not have to be part of the message (to keep it as short as possible). This could be a nonce based on synchronized clocks between

¹⁰⁹Actually, the maximum value would be 999999990 due to the inclusion of a checksum known as the ‘elfproof’, but for the sake of simplicity, we assume the maximum is the maximum value provided by 9 digits.

authentication device and the bank, or an incrementing counter of which the bank keeps track.

To let the authentication device calculate MAC mac_a over message m (an and am concatenated, from the previous step), we use a randomly generated key k of 2048 bit, a 32-bit unsigned incrementing counter as nonce n_a , and a hash function H , we could use:

$$mac_a = H(k \mid n_a \mid m)$$

Care must be taken with this approach, since it is susceptible to length extension attacks if H is based on the Merkle-Damgård construction (such as MD5, SHA-1 and SHA-2) [SG12]. A mitigation would be to use an alternative hash algorithm. SHA-3 is based on the sponge construction, which does not have this inherit limitation.¹¹⁰ Therefore, SHA-3 will be used for H for this example.

The strength of the MAC is based on its length. A larger MAC is more secure against brute-force attacks, which are further discussed in Section 4.3.6. However, since the MAC is part of the MC, a longer MAC would result in a longer MC that needs to be transcribed by the user. The length further depends on the chosen encoding in the next step. A dynamic MAC size can prevent padding, which ensures that the MC can have a fixed length and that bits are not wasted in the encoding progress. We assume that a range of one million values would be enough to construct the MAC, which can be represented by 20 bits ($2^{20} = 1048576$). Our motivation for choosing an encoding scheme will be explained in the next section, but it is important that we now keep the choice in mind to align the number of bits with the number of characters required in the MC to represent both the message and the MAC. We choose Z-Base 32 as our encoding scheme. The message itself is 57 bits, as discussed in the previous section. With 20 bits added for the minimum MAC, the number of bits for the MC would be 77. Since Z-Base 32 uses 5 bits for each character, this number is rounded up to 80 bits. Therefore, the MAC length is 23 bits because $57 + 23 = 80$.

All values and the output of the formula:

$$\begin{aligned} k &= 0x00\ 0x01\ \dots\ 0xFF \\ n_a &= 0x00\ 0x00\ 0x00\ 0x01 \\ m &= 0x01\ 0x6A\ 0x34\ 0x1B\ 0x38\ 0xBC\ 0x61\ 0x4E \\ mac_a &= H\{ k \mid n_a \mid m \} = 0x77\ 0x64\ 0xCE \end{aligned}$$

Encode

The compression phase was used to compress the data and turn it into a bit stream. Furthermore, we gave an example of how a MAC can be calculated in the signature phase and computed a MAC of 23 bits. In the encoding phase, we combine the values from both phases to create an MC, a human-readable text string that the authentication device will show on its display.

The output of the compression phase is 80 bits. Assuming that we want a code with a fixed length, based on the encoding methods noted in Section 4.2.3 we can

¹¹⁰The Keccak sponge function family: <http://keccak.noekeon.org/>

tell how long the MC will be by taking the number of bits and calculating how much words or characters would be needed to encode it, rounding up to ensure a fixed length. If the 800 word encoding from Wiseman et al. (2016) would be used [WMC⁺16], the length would be 9 words or 27 characters (excluding spaces between the words). Base64 would require 14 characters, and Base32 and Z-Base 32 would require 16 characters (all excluding separators). Finally, 25 characters would be required if only digits were used.

To improve readability, separators would be required to create groups of characters. For words, that would be a total of 35 characters (based on 8 spaces). Based on groups of 4 characters each, three separators would have to be added to Base64, Base32 and Z-Base 32, bringing their number of characters respectively to 17, 19 and 19. When only digits would be used and 4 separators would be used to create groups of 5 digits, 30 characters would be required. These are the number of characters that the authentication device's display would be required to show.

Wiseman et al. (2016) tested their word encoding with 3 words, which was the preferred encoding by their test candidates. However, 9 words would be quite long to show on an authentication device and for users to enter on other devices. For this case study we will use Z-Base 32 instead, which provides a good balance between character length and recognition of the used characters.

See Table 4.2 for an overview of the encoding process using Z-Base 32. The result is an MC, which could be displayed as:

`swpy-58n6-gnuz-q3gq`

4.3.3 Transfer the Message Code

As described in Section 4.2.4 and as shown by steps 3 and 4 in Figure 4.5, the user reads the MC from the trusted authentication device and enters it in the untrusted computer used for online banking. To make the transfer user-friendly, the MC's length was reduced by compressing the message in Section 4.3.2 while it is structured by the alphanumerical encoding chosen in Section 4.3.2.

An alternative to the manual transfer by the user of the MC from the authentication device to a smartphone or tablet running a mobile banking application might be the use of a QR code if the device has a rear-facing camera. In that case, the authentication device would show the QR code to be scanned and the mobile banking application would scan the QR code. Compression and encoding would be less of an issue of user-friendliness in these cases, and more of an issue in keeping the QR code as small as possible to make it easier to scan it.

4.3.4 Verify the Message Code

Step 7 concerns the verification of the MC, and is mostly step 2 in reverse order. First, the data is decoded back into a bit stream and the message and MAC are separated. After that, the bank needs to verify if the included MAC corresponds with the payload to verify integrity and authenticity. Only if both are verified will the message be processed further.

Decode

Decoding is the reverse of encoding as done at the end of Step 2. Table 4.2 can be read starting from the bottom row to get an idea of the decoding process. The result is the top row's bitstream.

Verify

The bank calculates its own MAC, and with that effectively performs the same steps as the authentication device did in step 2. The assumption is that the bank has access to the same k and is able to deduce n_a . A counter was used for n_a , which increases by one for every generated MAC. The bank has to store the nonce of the last received valid message to detect replay attacks in future messages, which we will refer to as n_b .

n_a as a 32-bit value is not included in the message to keep the MC as short as possible. That is why the bank has to deduce it. It is possible that the user generates MACs which the bank never receives. This can happen if the user is testing the workings of the authentication device or when an online banking session is disconnected, after generating the MAC but before the bank receives it. Since n_a is a counter that is only incremented by the authentication device, all the bank has to do to compensate for discrepancies between the last stored value in the authentication device and at the bank would be to attempt to verify the MAC multiple times with increasing values for the nonce, starting at n_b and increasing for an acceptable number of messages which the bank did possibly not receive. It might be likely that the user created a message or two that were missed, but it is unlikely that the user generated 50 messages. In this exceptional scenario, the user could be requested to contact the bank.

For our example, we assume that no previous MCs were missed by the bank. Let k be the shared key between authentication device and bank, n_b the earlier mentioned stored nonce value at the bank (increased to the value of $n_a + 1$ with each valid received message), m the message, and mac_b the MAC that the bank calculates.

$$\begin{aligned}k &= 0x00\ 0x01 \dots 0xFF \\n_b &= 0x00\ 0x00\ 0x00\ 0x01 \\m &= 0x01\ 0x6A\ 0x34\ 0x1B\ 0x38\ 0xBC\ 0x61\ 0x4E \\mac_b &= H\{ k \mid n_b \mid m \} = 0x77\ 0x64\ 0xCE\end{aligned}$$

Now all the bank has to do is verify if mac_a equals mac_b . If they are equal, the message is authentic and its integrity is protected, and should therefore be processed further.

Decompress

Decompression of m is the opposite operation of compression as done in step 2. First, the two values are separated, based on their lengths and positions. Let an

again be the destination account number and am the amount of money.

$$\begin{aligned}an &= 101101010001101000001101100111 \\am &= 000101111000110000101001110\end{aligned}$$

an is converted to an integer which results in a value of 759,595,879. The IBAN is reconstructed by supplementing the integer with known values. In the end, the bank wants to have the same number as shown in Figure 4.6.

The country code is easy since the transfers were limited to Dutch domestic accounts. Therefore, the bank knows the country code is NL. As noted earlier, banks in the Netherlands can identify which account number belongs to which bank, allowing them to match the account number to the bank with the bank code SIMB. Finally, the account type indicator is always 0 for payment accounts. The value the bank now has is:

NL ?? SIMB 0 759 595 879

The IBAN checksum is still missing, which can be recalculated.¹¹¹ The fully reconstructed IBAN:

NL 76 SIMB 0 759 595 879

Reconstructing the amount of am is fairly easy. All that is needed is a conversion to an unsigned 32-bit integer and a division by 100 to create the original decimal value. With domestic transfers it is not required for the user to specify the currency if there is only one. Our example concerns the Netherlands, which uses the euro. Therefore, the bank knows that the value to transfer is € 123456.78.

4.3.5 Further processing

The bank returns the IBAN and the amount back to the user's browser (step 8 of Figure 4.5), which receives (step 9) and shows (step 10) them. The user has the opportunity to fill in the rest of the values in the form and submit the transaction. An overview screen could be shown before the user gives his final approval.

4.3.6 Attack analysis and mitigation

The security of the system is based upon the generation and verification of the MAC, which provides integrity ('Was the data changed in any way?') and authenticity ('Does the data come from the expected source?'). An adversary not having access to the key should be unable to generate valid MACs at will.

The random adversary

Step 5 is the point where malware can modify transaction data before it is sent to the bank, and step 9 is the point where malware can modify what the bank returns to the user. Imagine that an adversary wants to change the destination account number

¹¹¹European Committee for Banking Standards (August 2013) - IBAN: http://www.europeanpaymentscouncil.eu/documents/ECBS%20IBAN%20standard%20EBS204_V3.2.pdf

silently at the beginning of step 5. An account under control of the adversary is NL 38 VIRB 0 307 633 357 (at the fictional **VIR**tual **B**ank). The attacker modifies the data as follows (presented in their most clear data types for clarity) in step 5:

	Account	Amount	MAC
Original:	759595879	123456.78	0x77 0x64 0xCE
Modified:	307633357	123456.78	0x77 0x64 0xCE

When only the account number is changed in step 5, the bank would notice upon processing the data in step 7 that the generated MAC is not based on the received account and amount data. If the MAC does not fit the message (account and amount), the transaction is discarded. For steps 8 and onward, the bank might notify the user to contact the bank for clarification.

The known adversary

A ‘known’ adversary is an adversary to which the user (for a legitimate reason) transfers money to, either at the moment of an attack or in the past. The known adversary, having full control of the user’s computer, can attempt to use an older valid transaction code of a transaction to the adversary to create a new transaction in a replay attack.

Due to the inclusion of a nonce in the MAC, it is possible for the bank to detect a replay attack. In our example, we used a counter that increases by one for each generated MAC. A bank could detect a replay attack in several ways:

- The bank stores each received (legit) MC. If a received MC (message and MAC) is equal to a previously stored MC, it is part of a replay attack and should be refused.
- The bank only stores n_b , the nonce of the last received legit message. Attempts to verify the MAC of the replayed message would fail since n_a would be lower than n_b , and the bank only checks the current and higher n_b values.

Therefore, previously created legitimate transactions do not aid an adversary in creating illegitimate transactions.

Another potential attack vector exists with a known adversary. If the user prepares a legitimate transaction to the adversary, the adversary could change the confirmation that is given in step 10 to the user. Through control of the user’s computer, the adversary could change the verification information in step 9 and make the user think that the transaction failed, which could make the user re-authenticate the transaction and send it again. The re-attempt would concern an illegitimate transaction. Banks could provide protection against this by monitoring repeated transactions to the same destination account number with the same amount within a limited time frame.

The determined adversary

The used MAC is relatively short (23 bits in the example), to keep the transaction code as short as possible and make it align with the 5 bit boundary of Z-Base 32.

A determined adversary might attempt to brute force a valid MAC, since for every payload it can be expected that one valid MAC exists in the range of 2^{23} .

What limits an adversary is that the MAC can only be verified by the bank and not by an adversary (due to missing the secret key required for both generating and verifying the MAC). The bank can register users that consecutively offer wrong MACs, and not process further attempts from them for a specified amount of time.

While an adversary might try to brute force a transaction to a random destination address (of which the adversary has no control), this would not help him or her in gaining money. In the unlikely event that a valid MAC would be generated for one payload, it would not disclose the key required to generate MACs for another payload with a different account number. Therefore, the destination address provided by an adversary would most likely be an address under control of the adversary. Further transactions to a specific destination address can be delayed by the bank if multiple wrong MACs are generated for it, either by a single or several users.

4.4 Formal verification

Formal protocol verification concerns the proof of (in)correctness of protocols. It can verify or dismiss whether a protocol would perform as it is designed in terms of security properties. We formally verify several security properties of the steps as described in Section 4.3 as a protocol.

4.4.1 Assumptions and notation

Several assumptions (or limitations) have to be noted to set boundaries for the verification. WYenterIWYS is a transaction authorization information scheme. It does not deal with user authentication or the establishment of a secure session to work in. Only a single action is performed by preparing and sending a single Message Code mc from one party to another.

For readability, C will represent both a bank customer and a trusted device that is personalized to the customer, and B will represent his or her bank. A represents an adversary.

Before a transaction is started, authenticity of the identity C has already been established by B in some secure manner (entity authentication). Therefore, both C and B know which key material to use. The key material consists of:

- A secret key k , known only by C and B .
- A randomly initialized clock t (an unsigned 32 bit value). The value (epoch) with which the random clock of C was initialized is secret. C only has its current clock value, which increases by one every second. B has the value with which the clock was initialized stored together with the time the clock was initialized. This allows B to calculate t at any moment, effectively giving it also access to t .

mc consists of transaction data m , and MAC h generated using k , t and m .

A can intercept any mc and prevent mc from being received by the bank, and change/inject their own mc . It is not possible for A to send messages to C , since C

does not have the capability to receive messages from B . In reality, A could attempt social engineering attempts through the user's browser or using some other method, but we exclude this possibility to focus solely on the technical implementation of the protocol.

It is assumed that C works as a single, trustworthy entity. Adversaries do not have access to or influence on whatever the user enters on the device for the purpose of this verification.

Also, it is assumed that clock skew will not occur. The length of seconds for both C and B is exactly the same and they are synchronized at all times. In reality, clock skew likely has to be compensated by B , but this is avoided for formal verification for the sake of simplicity.

Finally, t will never go as far as reaching its own starting value again. Once initialized, it would take 4 294 967 296 seconds (including a reset of the binary 32 bit value to 0 once it reaches 2^{32}) before it would reach its initialized value again. This is approximately 136 years. It is assumed that the device its lifespan will be less than this, making the 'reset' of the clock a non-issue.

4.4.2 Security requirements

For the protocol to work, several requirements have to be met.

- k has to remain secret, since it provides the secret key values used to generate and verify the MAC, and ensures that the message was generated by C .
- t has to remain secret. For lack of an exchanged random nonce, it is the only part of the protocol that provides freshness to each message.
- C and B have to agree with the contents of the message. In other words, when C sends a message, B has to agree with the received data.
- mc should not be usable in replay attacks.

The values m and h are not secret.

4.4.3 Constraints

The constraints of execution are that each second either a complete mc is sent by C , or nothing is sent. The rate at which C is able to generate mc has a maximum of one per second. This ensures that h of each message always has new freshness through t .

4.4.4 Formal verification example

For formal verification an automatic tool called Scyther was used.¹¹² Scyther uses a simple language to define a protocol. WYenterIWYS, as defined in this chapter, is modeled as follows in Scyther's `spdl` format:

¹¹²Scyther tool: <https://www.cs.ox.ac.uk/people/cas.cremers/scyther/>

```

1  usertype Timestamp;
2  hashfunction H;
3
4  macro h = H(k, t, m); # MAC
5  macro mc = m, h; # Message Code
6
7  protocol WYenterIWYS(C,B)
8  {
9      role C # Customer
10     {
11         fresh m: Ticket; # Message (to send)
12         fresh t: Timestamp; # Random shared clock
13         send_1(C, B, mc);
14
15         claim_C1(C, Secret, k);
16         claim_C2(C, Secret, t);
17         claim_C3(C, Secret, m);
18         claim_C4(C, Alive);
19         claim_C5(C, Weakagree);
20         claim_C6(C, Niagree);
21         claim_C7(C, Nisynch);
22     }
23
24     role B # Bank
25     {
26         var m: Ticket; # Message (to receive)
27         fresh t: Timestamp; # Random shared clock
28         recv_1(C, B, mc);
29
30         claim_B1(B, Secret, k);
31         claim_B2(B, Secret, t);
32         claim_B3(B, Secret, m);
33         claim_B4(B, Alive);
34         claim_B5(B, Weakagree);
35         claim_B6(B, Niagree);
36         claim_B7(B, Nisynch);
37     }
38 }

```

One of the aspects that is not often modeled in Scyther is the use of **fresh** for the same value by two communicating parties. Normally, **fresh** is used by one role to indicate that role creates the value, while the other role uses **var** to indicate that that role should receive the value. An example of this normal behavior is shown in the code above, where *C* creates and sends *m* (lines 11 and 13), which is received by *B* (lines 26 and 28). However, this is not the case for *t*. As discussed earlier, *t* is already known by both parties before the protocol starts. To model this, *fresh t* is used by both roles (lines 12 and 27).

The result of Scyther's runs without a maximum on the number of runs is shown in Figure 4.7. The claims will be discussed based on [Low97] and [CM12]. Note that claims made for B are ignored. There are no attacks within bounds because B never sends a message. We will only discuss claims for C , which sends mc to B .

As shown, no attacks were found within the bounded statespace against the secret state of k (claim C1) and t (claim C2). This is as expected since both k and t are not transferred. h is a product of k and t (and m), but as an irreversible hash function it cannot be used to retrieve k or t . As shown, m is not a secret (claim C3) since it is transferred as plain text. This is as expected.

Claims C4 and C5 are related to 'Alive' and 'Weakagree'. *Aliveness* indicates to an initiating party (C) that the other party (B) has run their part of the protocol after completing it. *Weak agreement* indicates to an initiating party (C) that the other party (B) has run their part of the protocol after completing it *explicitly and exclusively with B*. Both of these claims fail because there is no communication from

Scyther results : verify							
Claim				Status	Comments	Patterns	
WYenterIWYS	C	WYenterIWYS,C1	Secret k	Ok	No attacks within bounds.		
		WYenterIWYS,C2	Secret t	Ok	No attacks within bounds.		
		WYenterIWYS,C3	Secret m	Fail	Falsified	At least 1 attack.	1 attack
		WYenterIWYS,C4	Alive	Fail	Falsified	Exactly 1 attack.	1 attack
		WYenterIWYS,C5	Weakagree	Fail	Falsified	Exactly 1 attack.	1 attack
		WYenterIWYS,C6	Niagree	Ok	Verified	No attacks.	
		WYenterIWYS,C7	Nisynch	Ok	Verified	No attacks.	
B		WYenterIWYS,B1	Secret k	Ok	No attacks within bounds.		
		WYenterIWYS,B2	Secret t	Ok	No attacks within bounds.		
		WYenterIWYS,B3	Secret m	Ok	No attacks within bounds.		
		WYenterIWYS,B4	Alive	Ok	No attacks within bounds.		
		WYenterIWYS,B5	Weakagree	Ok	No attacks within bounds.		
		WYenterIWYS,B6	Niagree	Ok	No attacks within bounds.		
		WYenterIWYS,B7	Nisynch	Ok	No attacks within bounds.		
Done.							

Figure 4.7: Scyther verification.

B to C . There is no way for C to know whether B ran the protocol with C or with anyone else.

This is a weakness in that it is possible for A to perform a replay attack by impersonating the bank and tricking the bank customer (outside of the model, through the computer that the customer is using) that B did not receive the transaction. The customer would attempt to authorize the transaction again. C (representing the customer and the trusted device used by the customer) would send a second transaction with the same parameters as the first transaction to B , not knowing that the first transaction was already accepted. It is important to note that this attack would only benefit A financially if the first transaction was already destined for A directly or indirectly. This restraint limits the usefulness of the attack unless it is combined with a social engineering attack.

Claims C6 and C7 relate to ‘Niagree’ and ‘Nisynch’. ‘Niagree’ is a shorthand for *non-injective agreement*, which requires that the content of received messages correspond with those of sent messages as defined by the protocol. ‘Nisynch’ represents *non-injective synchronization*. Being similar to ‘Niagree’, it requires in addition that the communication order is respected (messages have to be sent before they can be received, as defined by the protocol). Both claims are valid with the given definition, indicating that the protocol as it is defined fulfills these criteria.

The automated formal verification tells us that the protocol as defined is working as intended. It was expected that m would not be secret, and that there is no way for C to verify that B performed their part of the protocol due to the constraint of one-way communication. An undetectable denial of service attack is quite easy to perform. With such an attack, A can prevent a C from sending money to an intended party through B . This is nothing new. Even after a bank customer successfully authenticates with WYseeIWYS, an adversary can block the verification code from ever reaching the bank while still displaying a ‘transaction is completed’ message to the user on an untrusted device. In addition, an adversary will not gain any money through such an attack. The possibility of such an attack is a weakness in the protocol, but motivation to perform such an attack will be low since an adversary does not gain money by performing it.

Claims for non-injective agreement and non-injective synchronization pass verification. This means that an adversary is not able to change any messages without breaking the protocol as it is defined in and analyzed by Scyther. In addition, send and receive events are executed in the order as defined and have the same contents.

Scyther does not verify all security properties of a protocol. In particular, it does not verify injective synchronization (replay attacks) and injective agreement. This is a subject left for future research.

4.5 Discussion, limitations and further research

In Section 4.2 we analyzed methods that can be used to create an MC. The case study in Section 4.3 shows that it is possible to use a selection of these methods to create and verify an MC for transaction authentication in online banking in a way that is device-independent and does not rely on the user to make redundant security decisions. Note that this is a case study which only addresses the challenges faced in a single scenario. Other scenarios could present different obstacles and different

methods might be more appropriate to overcome these using an MC, or the use of an MC might not be appropriate at all. For example, the case study applied the WYenterIWYS information scheme, which is only usable if the user provides transaction information and not a third-party (such as an online e-commerce system). In the case of the latter, WYseeIWYS might still be the best method to authorize such transactions.

Further research could test the (perceived) usability when transferring MCs from one device to another using different manual methods (alphanumeric, numerical, words-based), and possibly expand on the idea of using automated methods instead, whenever the technology allows it (such as QR codes).

An automated formal protocol verification tool was used to verify different security properties of the steps described in Section 4.4. The tool Scyther identified a potential attack vector that by itself has limited use, but still has to be considered a weakness. It also indicated that despite this weakness, the messages themselves cannot be tampered with in the protocol as it is defined. A combination of research in both further formal verification and improvements to the proposed WYenterIWYS implementation could enhance the protocol in a way that ensures all security properties are met. This will require some feedback over a secure channel from the bank back to the trusted device used by the user. It might be possible to combine some aspects of different WYseeIWYS implementations to return a signal to the user's device to indicate that the transaction was successful or whether the transaction failed. In addition, other forms of formal verification can support or dispute claims about other security properties.

4.6 Concluding remarks

We proposed a method that allows humans to transfer both a message and data to secure the integrity and authenticity of the message by transcribing a single code. Different methods were examined to construct such a code in a way that makes the transfer user-friendly. In addition, a case study was performed in which such a code was used to secure online banking transactions for which the user provides critical information.

The case study shows that for online banking, the use of a Message Code can remove the necessity for a user to verify entered information twice, as is currently done. By taking away a critical decision from the user, usability is improved since the user has less critical choices to make, which also improves security since authenticity and integrity of the data rely less on user activities.

Part III

Evaluating authentication and authorization schemes

Part III - Evaluating authentication and authorization schemes

After creating the first version of the proposal for What You Enter Is What You Sign as an alternative transaction authorization scheme to What You See Is What You Sign (on which Part II, Chapter 3 is based), the question rose about how this and other online banking authentication methods can be evaluated and compared. Ideally, comparisons based on characteristics can be done ‘at a glance’, while their effectiveness and efficiency could be measured by user testing.

Sven Kiljan, Harald Vranken, and Marko van Eekelen. Evaluation of transaction authentication methods for online banking. Accepted for publication by Elsevier Future Generation Computer Systems, 18 pages, 2016. [KVvE16a]

Methods exist to compare user authentication method implementations. However, these methods often focus on usability and security as separate aspects. They do not consider usable security, which is an important aspect in the additional interaction a user has with online bank security systems for transaction authorization. An evaluation mechanism that quantifies qualitative characteristics in web authentication methods was extended to include different usable security aspects. These aspects are related to the cognitive load that is put on the user, and to whether users are able to circumvent security willingly or accidentally. Chapter 5 is based on the resulting paper. Not included in the original paper was Section 5.12. This is an expansion for this thesis in which the evaluation mechanism is applied on the What You Enter Is What You Sign-based transaction authorization method that was proposed in Chapter 4. In addition, the original (Dutch) questions asked to the raters who tested the evaluation mechanism were added to Section 5.8.4.

Sven Kiljan, Harald Vranken, and Marko van Eekelen. Towards a virtual bank for evaluating security aspects with focus on user behavior. Published in Proceedings of the SAI Computing Conference, pages 1068-1075, July 2016. [KvEV16]

A standardized platform to perform online banking usability (and usable security) tests does not exist. Technically capable academical researchers sometimes create their own ad hoc testing environments, but these are often developed with a very limited set of functions in mind and discarded after testing. Such environments include whatever needs to be tested and anything that is needed to support the tests indirectly. For example, a new authentication method does not only require a login page to be tested, but also a functional online banking environment when the method also caters to transaction authorization. A framework was envisioned which researchers could use to test self-developed security modules. This would not only make it easy to share test data, but also the modules used for testing. Researchers would be able to further enhance and re-test what was shared by others. Development began of the framework, but the question rose about whether it is actually possible to collect interesting usage data using only web technologies. A proof of concept was presented in a paper on which Chapter 6 is based.

Wary readers will notice that the titles of these chapters are not the same as the titles of the papers that they are based on. The names were changed to make a more clear distinction between the respectively presented theoretical and practical evaluations.

Chapter 5

Theoretical evaluation to quantify qualitative characteristics

Abstract

Authentication is a major research topic in the information security field. Much has been written about assessing entity (user) authentication methods, but there is a lack of literature concerning the evaluation of authenticating financial transactions in online banking. Entity authentication methods have been systematized by quantifying their qualitative aspects, but there is no evaluation mechanism which also places the additional characteristics of transaction authentication in a user-centric context. Based on an existing mechanism which quantifies accessibility, memorability, security and vulnerability characteristics in entity authentication methods, we propose feasibility as an additional dimension which quantifies aspects related to the secure usability of transaction authentication methods. We also propose the use of the evaluation mechanism by multiple raters to reduce personal bias. Four implemented and eight proposed authentication methods for online banking were evaluated by seven experts. The results indicate that the mechanism can be widely used, since it is able to evaluate authentication methods with different information schemes. However, care must be taken that evaluations are performed by multiple experts, due to the amount of subjectivity inherent in the mechanism and in the different opinions of the raters.

5.1 Introduction

Two forms of authentication can be used in online banking to authorize financial transactions [KSDC⁺14]. Entity authentication is concerned with proving the identity of an online banking user, similar to authentication for other online services (email, instant messaging, etc.). Transaction authentication concerns the certainty that financial transactions (the destination account number, the amount of money, etc.) are deliberately authorized by the user. Current evaluation mechanisms of entity authentication methods do not evaluate the specifics of online banking environments. A mechanism which also evaluates and compares aspects specific to transaction authentication is missing. Such a mechanism should take into account that transaction authentication methods can rely on an active role of the user to provide the security the method needs. Banks slowly start to introduce new user-centric transaction authentication methods which require users to verify information received by the bank on bank-issued trusted devices and on user-owned mobile devices. The possible reliance on the user's actions and the trustworthiness of what the user observes should also be considered when comparing authentication methods.

Our goal was to evaluate different implemented and proposed online banking authentication methods to identify points for improvement. Our contribution includes an examination of different proposed evaluation mechanisms and our own proposal. We extended an existing mechanism with aspects related to the feasibility of using an authentication method securely. The new aspects cover the taxation of the user's cognitive capacity through expansion of the user's work flow, the ability for security to be (willingly or unwillingly by the user) circumvented and the lack of function and information clarity through the user interface and in- and output channels. The mechanism we propose can be used to evaluate online banking authentication methods in a way which takes the active role of the authenticating user into consideration. Seven raters performed an evaluation of 4 implemented authentication methods and 8 proposals.

The rest of this chapter is structured as follows. Section 5.2 (page 107) starts with an overview of the background material our work builds on. This includes sources for the evaluation mechanisms we examined, papers about secure usability aspects in information security, and proposals for transaction authentication methods. Different proposed evaluation mechanisms are compared and our choice for Renaud's mechanism is explained in Section 5.3 (page 108). We give an overview of Renaud's mechanism in Section 5.4 (page 111). The new feasibility dimension is introduced in Section 5.5 (page 112), which accounts for the secure usability of the authentication method by the user. In Section 5.6 (page 117) it is noted how Renaud's mechanism and our expansion can be used by multiple raters to come to a single answer with less personal bias. We apply the original mechanism and the new dimension on four implemented and eight proposed online banking authentication methods, which are briefly described in Section 5.7 (page 119). Considerations for the evaluations are noted in Section 5.8 (page 123) and the result from the evaluations is given in Section 5.9. These results are analyzed in Section 5.10 (page 132). We wrap up with limitations, discussion and further research in Section 5.11 (page 140), and our concluding remarks in Section 5.13 (page 144).

5.2 Background and related work

In this section, we note the most influential past work on which we base our contribution.

5.2.1 Authentication evaluation mechanisms

Renaud introduced a mechanism which quantified the qualitative characteristics of user authentication systems [Ren04]. Aspects related to security and usability are given values based on qualitative characteristics to calculate a deficiency value over the aspects' respective dimensions. This approach allows comparisons of authentication methods by comparing weighted values without losing sight of important details. Values can be compared on three levels: aspect, dimension and overall. Since the environment in which an authentication method is used can have a positive or negative effect on its security and usability, Renaud also introduced environmental factors. These are modifiers that represent the influence an environment has over the dimensions to which the environmental factors are assigned, and allow comparisons of authentication methods in their respective environments.

Mihajlov et al. present a conceptual framework, which uses Renaud's quality criteria and their own predefined quantification approach [MBJ11, MJB11]. Differences with Renaud include an alternative mathematical model, and a reduction in the number of evaluated dimensions.

Another framework with a similar goal to Renaud's mechanism was proposed by Bonneau et al. [BHVOS12]. Aside from security and usability, their framework also took deployability aspects into account. This framework only evaluates aspects on a single level and does not assign numerical values.

All noted evaluation mechanisms and frameworks are further discussed in Section 5.3.

5.2.2 Secure usability aspects

Yee provides a list of design principles for a secure usable design of systems [Yee02]. A criteria of each principle was that it is fairly obvious that a violation of the principle equals to the introduction of a security vulnerability. They are proposed as guidelines for system designers to keep in mind.

Herley promotes the idea that users are economical instead of lazy in their decision to follow security instructions [Her09]. The cost of direct damage is often seen as a risk when security advice is ignored, but the far greater costs of indirect damage due to actually following the security advice to the letter is often not considered. It is these larger costs that makes users reject security advice, since the trade-off (in terms of the (perceived) reduced risk versus the (perceived) increase in user effort) is not considered worth the additional effort. In a follow-up, Herley explains how valuable a user's time is and how security has to compete for this time in today's information overloaded society [Her14]. He gives valuable advice to increase the acceptability of security instructions. The advice that relates most to our research results is that users should only be given instructions of which it can be expected that they will be followed.

5.2.3 Proposed online banking transaction authentication methods

Many authors have proposed conceptual improvements for transaction authentication in online banking. The proposals of Starnberger et al., AlZomai et al., Weigold et al. and Li et al. present different approaches to protect against attacks in which transaction data created by the user is modified before it reaches the bank for further processing [SFG09, AAJ10, WH11, LSH⁺12]. While the approaches are conceptual, they are clearly enough defined to analyze qualitatively.

5.3 Choosing an evaluation method

For our survey, we wanted to compare different authentication methods implemented by banks and from academical proposals on both security and usability related aspects. We chose a qualitative approach, in which the availability or abundance of specific characteristics would be observed. An advantage of this approach is that it produces comparable results. It also scales quite well when comparing more authentication methods, since only qualitative data is collected and analyzed. Measuring quantitative characteristics takes more effort for each evaluated authentication method and has a risk that the higher level of detail will not provide added value for comparisons. A disadvantage of examining qualitative characteristics is that results may not be reproducible since the observation is never completely objective. However, variance between observers can be reduced by stating the characteristics clearly.

We initially looked at rubrics as a base for the evaluation method. Rubrics are structured scoring guides which consist of specific pre-established performance criteria used to evaluate the quality of student work [Mer01]. A holistic rubric provides a score based on the overall quality, proficiency or understanding of the specific content and skills. This rubric type evaluates student work on a single level. There are also analytic rubrics, which give scores for specific aspects of student work and a summed total score, representing assessment on two levels. In general, holistic rubrics take less time to use while analytic rubrics provide specific performance feedback, giving insight in a student’s strengths and weaknesses. An overview of both is shown in Figure 5.1.

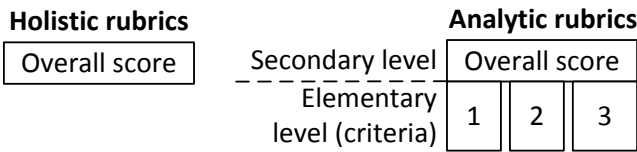


Figure 5.1: The levels and outputs of different rubrics types.

As noted, we wanted to evaluate methods based on both security and usability. Only an overall score for each authentication method would not tell us whether something is either secure or usable. Therefore, the analytic approach seemed more suitable. Instead of starting from scratch, we evaluated different proposals for eval-

uating authentication methods qualitatively to see what we could use as a base for our work.

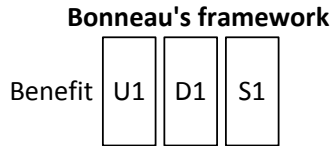


Figure 5.2: The single level outputs of Bonneau's framework.

Bonneau et al. introduced a framework (hereafter referred to as Bonneau's framework) for comparative evaluation of web authentication authentication methods with a specific focus on user authentication on the web through uncontrolled client computers [BHVOS12]. The 25 criteria in the dimensions usability, deployability and security represent what could be the characteristics provided by an ideal authentication method, and are therefore referred to as benefits. The deployability dimension is a combination of usability criteria (e.g. accessible to users with disabilities, independence of the installed browser, etc.) and economical criteria (e.g. negligible cost per user, and whether the authentication method is non-proprietary), and would definitely be useful when considering authentication methods which need to be deployed to a large number of users (such as with large banks).

One disadvantage of Bonneau's framework is that the output is only on a very detailed level, and lacks a 'total' score which allows easier overall comparisons between evaluated authentication methods. Each criteria gets an 'offers the benefit', 'almost offers the benefit' or 'does not offer the benefit' value, which for some criteria is quite ambiguous and can therefore be interpreted in multiple ways by different raters. As noted earlier, it is possible to reduce the variance in observations, but only if the criteria are very narrowly defined. Furthermore, the authors recognize that weights of criteria can change based on specific goals for which authentication methods are compared, and see this as a reason not to assign weights to the individual criteria at all.

We also looked at a mechanism introduced by Renaud (hereafter referred to as Renaud's mechanism), which is used to compare the quality of web authentication methods [Ren04]. Renaud's mechanism can be used for feature analysis of authentication methods and provides quantified scores on overall, dimension and aspect levels. Four equally weighted dimensions are recognized: accessibility, memorability, security and usability. Each dimension has three equally weighted aspects, each represented by a value that is constructed from either multiple criteria or from a single criteria which can have one of three or four specifically defined values.

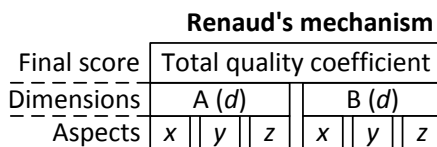


Figure 5.3: The levels and outputs of Renaud's mechanism.

Renaud’s mechanism is closer to the idea of analytic rubrics compared to Bonneau’s framework, which can be seen by comparing Figure 5.1 and Figure 5.3. Like analytic rubrics, Renaud’s mechanism applies pre-established and specifically defined performance criteria to qualitatively rate different aspects of some work while also providing an overall score. Aside from the most detailed ‘aspects’ (represented by x , y and z in each dimension), it also provides intermediate deficiency (d) values which can be used for comparisons between authentication methods based on specific dimensions. As noted by Mertler who cites Trice [Mer01, Tri00], the process of converting rubric scores to student grades and descriptive feedback involves more logic than math. In the case of Renaud’s mechanism the resulting quantified values have no mathematical context, nor is the source arbitrary. The values are only used as weights for easy comparisons and they are established using specific and detailed rules.

Another concept of Renaud’s mechanism is the environmental factor. Each dimension has one or two environmental factors which act as modifiers for the dimension’s deficiency value. These factors allow aspects from a dimension to weigh heavier or lighter depending on how well the environment supports the dimension. Environmental factors make it possible to compare authentication methods, where the environments’ influence is included in the comparison.

We were therefore inclined to use Renaud’s mechanism as a base for our work, since it closely matches analytic rubrics. It is vital that the to-be observed criteria are described in as much detail as possible, and Renaud’s mechanism gives a more detailed description on more levels compared to Bonneau’s framework. The output of the mechanism allows comparisons of authentication methods on different levels, which makes it easier to spot where a method is strong and where it could be improved. While environmental factors are not directly relevant for our research, they can be used by other researchers to compare our results with evaluated authentication methods from other fields which are not online banking. This does not imply that Bonneau’s framework is completely inappropriate. The economical aspects of the deployability dimension are something that Renaud’s mechanism does not have. A bank would most certainly be interested in comparing the economic feasibility of authentication methods.

We also considered the work of Mihajlov et al. (2011), who presented a conceptual framework (hereafter referred to as Mihajlov’s framework) partly based on the qualitative characteristics provided by Renaud for usability [MBJ11, MJB11]. In this framework, the number of dimensions are reduced to two: security and usability. One other difference is that the conceptual framework allows raters to more explicitly define how several criteria apply to an authentication method.

Mihajlov’s framework has a heavy focus on the values of its dimensions and, derived from these two values, the total quality value as an end result of an evaluation. This is similar to Renaud’s mechanism. However, the reduction in number of dimensions reduces the amount of output the framework offers. Renaud’s mechanism provides separate output values for quality criteria related to usability (through the total values of the accessibility and memorability dimensions) and security (through similar values for the vulnerability and security dimensions), while Mihajlov’s framework only provides overall values for usability and security. This makes Renaud’s mechanism more transparent on the second (dimension) level. Furthermore, while

Dimension		Value modifiers
←	Environmental factor / aspect	
Accessibility	Environmental factor: Control of Environment	Controlled (=1.00), uncontrolled (=1.50)
	Aspect: Special Requirements	Hardware configuration (+0.33), software configuration (+0.33), technical expertise (+0.33)
	Aspect: Convenience	Enrollment time (+0.25), key replacement time (+0.25), authentication time (+0.50)
	Aspect: Inclusivity	Cognitive excluded (+0.33), mobility excluded (+0.33), sensory excluded (+0.33)
Memorability	Environmental factor: Frequency of use	Daily (=0.50), weekly (=1.00), monthly or less (=1.50)
	Environmental factor: Forced Renewal	Not enforced (=1.00), enforced (=1.50)
	Aspect: Retrieval Strategy	Fully recognition-based (=0.00), recall-based with cues support (=0.50), recall-based (=1.00)
	Aspect: Meaningfulness	Self-assigned & deducible through special scheme (=0.00), self-assigned & meaningful to user (=0.33), self-assigned but not necessarily meaningful or deducible (=0.67), arbitrarily assigned (=1.00)
Security	Aspect: Depth of Processing	No effort (=0.00), particular level (=0.33), visual mechanism (=0.67), rehearsal-based (=1.00)
	Environmental factor: Risk	No damage when compromised (=0.50), damage to user (=1.00), damage multiple users (=1.50)
	Environmental factor: Security Motivation	Sanctions can be applied to irresponsible users (=1.00), sanctions cannot be enforced (=1.50)
	Aspect: Predictability	Authentication key is unpredictable (=0.00), only by friends/family (=0.50), widely predictable (=1.00)
Vulnerability	Aspect: Abundance	Range of keys is $\geq 2^{64}$ (=0.00), $\geq 2^{40}$ and $< 2^{64}$ (=0.50), $< 2^{40}$ or unique and irreplaceable (=1.00)
	Aspect: Disclosure	Impossible to disclose (=0.00), possible by shoulder surfing (=0.50), easily by user/attacker (=1.00)
	Environmental factor: Auditing	System applies auditing (=1.00), does not apply auditing (=1.50)
	Confidentiality	Key is not revealed or cannot be reused (=0.00), key is partly revealed (=0.50), full key is revealed (=1.00)
Vulnerability	Privacy	No personal details required (=0.00), allowed to use (=0.50), required to use (=1.00)
	Break-/Crackability	Does not apply (=0.00), vulnerable to research-based attacks (=0.33), dictionary/brute-force attacks (=0.67), keylogging (=1.00)

Table 5.1: Modifiers of environmental factors and aspects in Renaud’s mechanism. Environmental factors (marked in gray for clarity) influence all aspect values within their respective dimensions.

Mihajlov’s framework allows raters to more precisely define the applicability of some of its criteria, this makes it more complex for raters to evaluate the system while it is unclear what the added value is of such precision on the end result.

In the end, we chose Renaud’s framework since its use is more clearly defined and its output is more transparent compared to the frameworks provided by Bonneau et al. and Mihajlov et al.

5.4 Renaud’s mechanism at a glance

An overview of the dimensions’ aspects in Renaud’s mechanism, their criteria and environmental factors is given in Table 5.1. We give a short description of the formulas used for aggregating the values of aspects to dimension deficiencies and from dimension deficiencies to the total quality coefficient. The same is done for applying the environmental factors.

Each dimension has three aspects (x , y and z). Each aspect has a minimum value of 0 (representing that the authentication method provides the highest quality or best fit for a particular aspect) and a maximum value of 1 (representing the lowest quality or worst fit). Each aspect is seen as equally important and therefore has an equal weight. The same is true for the different modifiers which define each aspect’s value, with a single exception. For the Convenience aspect in the Accessibility dimension, the authentication time is seen as more important since users often authenticate, while both initial enrollment in the system and the replacement of lost security credentials happen less often.

The aspect values are used to calculate deficiency value d for each dimension using:

$$d = \sqrt{x^2 + y^2 + z^2}$$

d can be used to see the quality an authentication method has in a specific dimension, where a lower value is a higher quality. The minimum and maximum

values of the aspects represent respectively the highest and the lowest quality an authentication method can offer in each dimension:

$$\min(d) = \sqrt{0^2 + 0^2 + 0^2} = 0$$

$$\max(d) = \sqrt{1^2 + 1^2 + 1^2} = 1.732$$

In formulas, *ad* represents the deficiency value for the accessibility dimension, *md* does the same for the memorability dimension, etc. The total quality coefficient represents how well an authentication method fits all dimensions, and can be calculated by:

$$\overline{eq} = \max(\overline{eq}) - (ad + md + sd + vd)$$

A higher total quality coefficient value represents a higher overall quality. The maximum total quality coefficient is based on the summed maximum deficiency values of the four dimensions:

$$\max(\overline{eq}) = \max(d) * 4 = 6.93$$

Each dimension also has one or two environmental factors, representing the influence characteristics of the environment over the aspects within their respective dimensions. Environmental factors are represented in formulas by their shortened names. Whereas the total quality coefficient value is used to determine the overall quality an authentication method has on its own, the environmental quality coefficient represents the same but with the embedded influence of the environmental factors. To calculate the environmental quality coefficient value, first the total environmental deficiency has to be calculated:

$$\overline{d_{env}} = ad * control + md * freq * renewal + sd * risk * motivation + vd * auditing$$

Then, the environmental quality coefficient can be calculated using:

$$\overline{eq_{env}} = \max(\overline{d_{env}}) - \overline{d_{env}}$$

Similar to the total quality coefficient, a higher value of the environmental quality coefficient represents a higher overall quality. The maximum environmental quality coefficient is:

$$\max(\overline{d_{env}}) = \max(ad) * \max(control) + \dots = 12.98$$

5.5 Expanding Renaud's mechanism with the feasibility dimension

Renaud notes that users are required to authenticate themselves to use computer systems and web sites securely [Ren04]. Her evaluation mechanism targets user authentication methods in web environments, which correspond with entity authentication in online banking. Unfortunately, the mechanism misses some aspects which

Environmental factor and aspects	
	Modifiers
Environmental factor: User Correction	Users can correct mistakes within a reasonable time frame without repercussions. (=1.00) Users are not allowed to correct errors without repercussions. (=1.50)
Aspect: Work Flow Expansion	User does not have to perform additional actions. (=0.00) Some existing user actions are repeated as part of the authentication procedure. (=0.50) New user actions are introduced to the user's work flow to support authentication. (=1.00)
Aspect: Circumvention	The system's default state is insecure. (+0.33) The user interface does not support secure user behavior. (+0.33) User could subvert security due to inconvenience. (+0.33)
Aspect: Clarity	Interface gives a false impression of an ability or lacks the right information to ascertain its limits. (+0.33) Information necessary to make a good decision before an action is taken is inaccurate or missing. (+0.33) Input and output channels can be spoofed or are corruptible. (+0.33)

Table 5.2: The feasibility dimension's environmental factor (in gray), aspects and their modifiers.

are vital to the secure use of an authentication method, especially transaction authentication methods. The four dimensions focus on aspects concerning usability (accessibility and memorability) and technical security (security and vulnerability). While the dimension memorability concerns usable security, it is limited to information in the authentication method which the user has to remember. There are other usable security aspects which are not part of Renaud's mechanism, but which are relevant to transaction authentication.

We introduce the new feasibility dimension. Its three aspects and environmental factor concern the feasibility of secure use of an evaluated authentication method. 'Secure use' is not simply a combined phrase to keep security and usability in mind as two aspects. Instead, it relates to the challenge of having a security system which is feasible for users to use in a secure way. Herley notes that a user's capacity for effort (basically a combination of time and energy) is one of the most valuable and scarce resources available in the information security field. If a user is expected to spend his or her resources inefficiently or ineffectively on security, it can only be expected in return that security instructions will be ignored or circumvented [Her14].

The qualitative characteristics which are quantified for the aspects related to feasibility can be found in Table 5.2. As noted in Section 5.3, Renaud's mechanism does not have a deployability dimension. While we do recognize the added value of such a dimension, we decided that deployability does not fit the user-centric context of our scope. The cost of deployment does not necessarily improve the security or usability of authentication methods.

Note that the exact deficiency and coefficient values are not interesting. It is the relative weight of each dimension which allows comparisons to improve authentication methods. The focus should also not solely be on each authentication method's overall percentage, which we only use for comparison within the context of the evaluation mechanism and to demonstrate the influence of the feasibility dimension. We can learn much by comparing the fulfillment of dimensions by each authentication method and observing where the low hanging fruit of improvements can be found, and which dimensions provide challenges.

As shown in Table 5.2, the different aspect levels of the work flow expansion dimension increase the aspect value linearly, while the criteria of the circumvention and clarity dimensions are proportionally equal in their dimensions. We consider the criteria for the circumvention and clarity aspects equal, which is why they have equal values. Similarly, the values of the work flow expansion aspect's criteria levels are based on the order in which each level taxes the aspect. Use of equal proportions re-

duces possible bias when applying the evaluation mechanism. Someone who applies the mechanism could decide that of an aspect one characteristic is more important than another and adjust the values accordingly. However, results of different evaluations are only comparable if the used mechanism to evaluate each authentication method is the same.

We describe the new aspects and environmental factor before we note the effects of the new dimension on the formulas of Renaud's mechanism.

5.5.1 Aspects of the feasibility dimension

As with the dimension's in Renaud's mechanism, the new feasibility dimension has three aspects.

Work flow expansion

Authentication can hardly be described as a desirable or enjoyable task for the user. It is a mandatory procedure which distracts from other tasks the user needs or wants to conduct. Therefore, there is not much incentive for users to spend more time and cognitive capacity than strictly required to fulfill the task. The user's time should only be spent if the benefits outweigh the costs. If not, security advice is rejected [Her09, Her14], resulting in insecure use of authentication methods.

The cost in time for enrollment, recovery and authentication is represented in Renaud's mechanism by the accessibility dimension's convenience aspect. The question whether a user is cognitively capable of using an authentication method is also covered by the same dimension's inclusivity aspect. Work flow expansion focuses instead on the question if and how the user's normal work is expanded solely for the purpose of authentication. An action is defined as either something the user is expected to do physically or cognitively. We recognize three distinct levels:

- The user is not required to perform any actions aside from possibly remembering and entering something. Memorability is excluded since it is already rated in its own dimension. The installation and configuration of hard- and software is also excluded since these are quantified by the accessibility dimension's special requirements aspect.
- The user must perform some actions redundantly. No additional actions are introduced, but the user must perform some of the existing actions twice (or more). An action has to be executed multiple times, at least once as part of a regular work flow and at least once as part of the authentication procedure. An example would be the entry of the same transaction data in both the user's computer and in an authentication device provided by the bank.
- Additional actions are introduced in the authentication procedure which require cognitive processing, such as when a user has to verify transactions by comparing transaction data as entered and as received by the bank (and shown to the user on a secure device) on equality.

Circumvention

Another feasibility aspect is the user's (dis)ability to circumvent the security system. Yee describes ten principles for user interaction design in secure systems [Yee02].

- The principle of the path of least resistance notes that the natural way to use a system should be the secure way. Since users use their physical and mental effort sparingly, the path of least resistance is the natural path for a user to follow and should therefore be the secure way to use the system. The ultimate path of least resistance is for the user to do nothing. Not doing anything is classified as an action which a system should also securely handle. The system must be secure against attacks while it is not being used.
- User errors should not be accepted as a source of security problems [Zur05]. It should not be possible for the user to subvert security unintentionally due to that the user interface does not support secure user behavior. An example is given by Yee, in which an icon of a lock can be clicked for security information [Yee02]. The associated action (examining security information) could be overlooked by the user if the icon does not look like it can be interacted with.
- It should not be possible for the user to subvert security intentionally (e.g. due to the amount of required effort). Inconveniences for the user increase the probability that the system will be used insecurely. If we use the lock icon again as an example and make it a button (so it is clear that it can be interacted with), a user can still opt-out of examining security information by simply not clicking the button. It is a security risk if the user is expected to perform an unenforceable action (e.g. verifying information on correctness).

Clarity

The final feasibility aspect is the clarity of the system towards the user. This concerns clarity in both information and offered functionality.

- Yee notes the principle of clarity [Yee02], which states that information should be accurate and available before a user action is taken, and also that the user interface does not present misleading, ambiguous or incomplete information. While the principle of clarity is defined from the perspective of a user who is granting security authorities, the same principle also applies when the act of authentication equals an act of authorization. Reliable information is required to make a good decision. The integrity of the information and its presentation should be protected. If they are not, the information (as interpreted by the user) is unreliable and unfit to make secure decisions on. An example of unreliable information would be aggregated transaction data, such as the number and sum of a set of transactions offered to the user on a secure device for verification. In this scenario, an attacker could change the destination account numbers of the transactions without the user being able to verify it. If non-aggregated information would be used, the user could check all the critical values (such as of each transaction the destination account number and amount of money).

- As remarked by Yee’s principles of identifiability and expected ability, the user interface itself should be identifiable and unambiguous regarding its abilities. If it is not, false user expectations can lead to wrong decisions with serious consequences. For example, the use of ambiguous terms for functions and labels can obfuscate what the authentication method can and cannot be used for. Likewise, if functions are unidentifiable, the user is vulnerable to error through inadvertent collision or intentional masquerading, on which social engineering attacks thrive.
- Finally, Yee’s principle of the trusted path describes that input and output channels should be secure against spoofing or corruption. An example of an insecure channel in authentication would be the use of SMS text messages to send critical decision information to a user, which can be spoofed [ETMLP05]. Also, smartphones used for receiving text messages are vulnerable to malware, which compromises both the integrity and availability of any received text messages since they can be spoofed, changed and forwarded to another phone while being kept hidden from the user [FFC⁺11].

5.5.2 Environmental factor: user correction

Yee notes the principle of revocability [Yee02]. Facilitating revocation is needed to accommodate users’ ability to correct slip-ups and errors. If a correction can be made without repercussions within the system’s environment, the available space for damage coming from user errors is reduced. The stakes for additional authentication actions a user has to perform as part of an expanded work flow are much higher when mistakes cannot be corrected. In this case, a larger burden is placed on the user since there is less room for error. Likewise, consequences of circumventing the system are more serious if the user has no way to make amends. The same is true for unclear information, tasks, and in- and output channels, for which the lack of clarity will have a bigger impact if slip-ups are not correctable.

The ability of users to make corrections is noted as an environmental factor that influences all feasibility aspects. If users can make corrections, the environmental factor does not have any influence. When the ability is absent, all aspects of the feasibility dimension are weighted heavier.

5.5.3 Adapted formulas

Based on the formula used for other dimensions, the deficiency of the feasibility dimension (fd) can be calculated by:

$$fd = \sqrt{x^2 + y^2 + z^2}$$

Where x , y and z are the values of the dimension’s aspects. Similar to the original four dimensions, the individual values can be used to compare authentication methods on specific aspects while the deficiency can be used to measure the quality an authentication method in a specific dimension.

The total quality coefficient for all dimensions, including the feasibility dimension, is now calculated by:

$$\overline{eq} = \max(d) - (ad + md + sd + vd + fd)$$

Note that $\max(d)$ is 8.66 due to the inclusion of fd . The new total quality coefficient can be used to measure an authentication method's overall quality.

The new formula for the total environmental deficiency is:

$$\overline{d_{env}} = \begin{aligned} &ad * control + md * freq * renewal \\ &sd * risk * motive + vd * audit + fd * correction \end{aligned}$$

Where *correction* is the value for the user correction environmental factor from the new dimension. The environmental quality coefficient is still calculated by:

$$\overline{eq_{env}} = \max(\overline{eq_{env}}) - \overline{d_{env}}$$

However, the new $\max(\overline{eq_{env}})$ is 15.58 due to the added maximum values of the feasibility dimension's aspects and environmental factor.

5.5.4 Relative scoring formulas

We chose to improve the readability of the deficiency and coefficient values by converting them to relative values. This also makes it easier to read the effect the additional dimension has on the total quality coefficient. To compare the deficiency of each dimension against its minimum and maximum values in a way that makes a higher value represent a better fit, we calculate a percentage (*adp* for the accessibility dimension, *mdp* for the memorability dimension, etc.) using:

$$dp = (1 - \frac{d}{\max(d)}) * 100\%$$

We also calculate an overall percentage for the total quality coefficient using:

$$\overline{dp} = \frac{\overline{eq}}{\max(\overline{eq})} * 100\%.$$

Calculating a percentage-based score for the total environmental quality coefficient would not have any added value. The resulting percentages would be the same as \overline{dp} due to the use of the same fractions.

5.6 Multi-user evaluation

Based on Renaud's mechanism as described in Section 5.4, we propose an expansion in Section 5.5. To test whether both can give useful results, we applied the evaluation mechanism on 4 implemented and 8 proposed transaction authentication methods, which are described up ahead in Section 5.7.

With qualification mechanisms there always is some subjectivity involved. For example, Renaud's mechanism asks the rater whether technical expertise is required to apply the authentication method (as part of the Accessibility Dimension's Special Requirements aspect). Technical expertise is quite an ambiguous term. Does installation of software on a home computer require technical expertise? Or the installation of an application on a smartphone? Another example would be the question of whether the method is time-consuming. A case can be made for that

enrollment and replacement takes a large amount of time, since the user at least has to visit a bank's office or needs to wait until a new/replacement device or code arrives in the mail. However, for authentication it is up to the rater to decide whether something is time-consuming or not.

To compensate for this subjectivity, it is possible to apply the same evaluation mechanism on the same authentication methods multiple times by different raters. Renaud's mechanism (with and without our expansion) has the advantage that it is simple to translate its characteristics that define the aspect values to survey questions which can be answered with either 'yes', 'no' or 'I do not know'. It is not needed for raters to be familiar with Renaud's mechanism when they are provided with a description of an authentication method and a list of questions to answer. The average of an answer can be fed back into Renaud's mechanism to fill in the characteristics that define the aspect values. This is repeated for every authentication method to be evaluated. Of course, it would be recommended to choose experts to be raters to come to a meaningful answer.

Table 5.3 gives an example of how four questions about an authentication method are answered by five raters.¹¹³ For question 1 and 2, it is simply the majority that defines what the average answer is. Although one rater did not know the answer for question 3, the answer would not have mattered since a majority had been reached by three other raters. Question 4 shows an uncomfortable situation, in which a majority could not be reached. This can happen when one or more raters do not know an answer (as in the example) or when an even amount of raters would provide equally distributed answers to the question. Since in this case the answer would be neither 'yes' or 'no', half of the relevant modifier's value (as shown in Tables 5.1 and 5.2) would be assigned. That would be 0.17 (or 1/6) for a value that's worth 0.33 (or 1/3) of an aspect's full value, 0.25 (or 1/4) for a value that's worth 0.50 (or 1/2) of an aspect's full value, etc.

In Section 5.8.4 starting at page 125 we describe how we let different raters apply Renaud's mechanism by itself and including our expansion. The results of the multi-user aspect of our experiment can be found in Section 5.10.4 starting at page 137.

¹¹³The example uses five raters for clarity. The evaluations are performed by seven raters.

Question →	1	2	3	4
Rater 1	yes	yes	yes	yes
Rater 2	yes	yes	yes	yes
Rater 3	yes	no	yes	unknown
Rater 4	yes	no	unknown	no
Rater 5	yes	no	no	no
Majority	yes	no	yes	unknown

Table 5.3: Example answers to questions related to a single authentication method. The majority defines the answer that will be used to evaluate an authentication method in Renaud's mechanism.

5.7 Evaluated authentication methods

We applied Renaud’s mechanism and our expansion on several used and proposed authentication methods. This section briefly describes the methods.

Each transaction authentication method applies an information scheme. We recognize three schemes [KVvE14]:

- Traditional transaction authentication (TTA). The method used for entity authentication is (re-)applied to authenticate transactions. User-recognizable transaction information is not used in this scheme.
- Customer verified transaction set authentication (CVTSA). A bank sends transaction information back to the authenticating user for verification.
- Entered single transaction authentication (ESTA). The integrity of transaction information is secured as soon as the information is created by the user.

The chosen identifiers used to refer to the authentication methods in the rest of this chapter are based on the following format:

<issuer> <characteristic> <user action and information type>

<issuer> is the unique identifier of either a bank or a proposal’s first author’s last name.

<characteristic> is a short description (possibly abbreviated) of the method’s main characteristic(s). These values can be:

- Entry. Applies to devices which require the user to enter transaction data on the device.
- hPIN/hTAN. A specific name for a proposal by Li et al. [LSH⁺12].
- Scan. Applies to devices which uses an optical sensor to scan data from a customer’s computer display.
- SMS (Short Message Service). Applies to methods which use SMS for transferring authentication information.
- USB (Universal Serial Bus). Applies to devices with which users interact and which are connected to a customer’s computer through USB.
- USB CR (Universal Serial Bus Card Reader). Applies to card readers without a user interface, connected through USB to the customer’s computer.
- ZTIC. A specific name for a proposal by Weigold et al. [WH11].

<user action and information type> is an abbreviation that specifies which kind of action (**None**, **Verify** or **Enter**) the user performs or is expected to perform for what kind of information (**None**, **Aggregated** or **Non-Aggregated**) when using the method. With **None**, no additional action is necessary. When **Verify** is specified, the user is expected to verify transaction information that the bank received and that was sent back to an authentication device in possession of the user. With **Enter**, the user has to enter critical transaction information on an authentication

device. **Verify** and **Enter** relate to either **Aggregated** (such as the number of transactions and the total amount of money of a set of transactions) or **Non-Aggregated** transaction information (such as the destination account number and amount of each transaction).

The following combinations of abbreviations are used:

Abbreviation	User action on information	Transaction information processed by authentication device
NN	None	None
VA	Verify	Aggregated
VNA	Verify	Non-aggregated
ENA	Enter	Non-aggregated

What follows are identifiers and brief descriptions of the evaluated authentication methods. The first four are based on methods used by banks, each at least used by half a million customers on a regular base. The other eight are proposals by different authors.

Bank USB CR NN (Bank Universal Serial Bus Card Reader None None)

This method consists of a bank-issued USB smart card reader connected to the user’s computer and supporting software. An example of such a reader is shown in Figure 5.4. This device is used in combination with a user’s bank card to login to the bank site and sign transactions shown on the user’s computer. The bank card requires a PIN to unlock its functionality, which is entered on the user’s computer. The user does nothing with any kind of transaction information in the authentication process (explaining the ‘None None’ or NN). Therefore, this method applies the TTA information scheme.

Bank SMS VA (Bank Short Message Service Verify Aggregated)

An SMS text message is sent to the user’s mobile phone during transaction authentication when a set of transactions is ready to be authenticated. The message contains aggregated information (the total amount of money and the number of transactions) and a one-time password. The one-time password must only be used if the total transaction amount in the text message corresponds with the value shown on the user’s computer. This method applies the CVTSA information scheme since users are expected to verify aggregated data (VA) of a set of transactions.

Bank USB VA (Bank USB Verify Aggregated)

Users are issued a device by their bank, of which Figure 5.5 gives an example. The device is similar to Bank USB CR NN in that it features a card reader and a USB connection, but it also has a display and buttons for user interaction. This authentication method allows the user to verify aggregated transaction information of a transaction set (the number of transactions and the total amount of money) on the device during transaction authentication. Confidentiality and integrity of information between the bank and the device is protected. A browser plugin on the computer translates the USB commands to network commands to be sent to



Figure 5.4: A USB smart card reader.

the bank site and vice versa. This method applies the CVTSA information scheme since users are expected to verify aggregated data (VA) of a set of transactions.

Bank Scan VNA (Bank Scan Verify Non-Aggregated)

This is another method which uses a bank-issued authentication device. The device is not connected to the user's computer. Interaction relies on a keypad, display, camera and smart card slot. In combination with a bank card and a PIN, the device is used to verify and sign transactions. During transaction authentication, non-aggregated information concerning individual transactions (destination account number and the amount of money) is projected on the display of a user's computer in a structured image and registered by the camera. The user enters a verification code shown by the device's display in his or her computer when confirming transactions. This method applies the CVTSA information scheme since users are expected to verify non-aggregated data (VNA) of a set of transactions.



Figure 5.5: A USB smart card reader with its own display and keypad.

Starnberger Scan VNA (Starnberger Scan Verify Non-Aggregated)

Starnberger et al. propose a transaction authentication method using an application on a user-owned mobile device [SFG09]. The camera of the device is used to scan a QR code from a personal computer, which contains (confidentiality and integrity protected) non-aggregated transaction information and a verification code. The user can enter the code on his or her PC to verify the transactions shown on the device. This method applies the CVTSA information scheme since users are expected to verify non-aggregated data (VNA) of a set of transactions.

AlZomai Scan+SMS VNA (AlZomai Scan+SMS Verify Non-Aggregated)

AlZomai et al. propose something similar to Starnberger et al. Instead of scanning a QR code, they suggest to scan plain-text transaction details from a computer screen using the device's camera [AAJ10]. The scanned data is verified against SMS text messages received from the bank. If the data matches, a verification code is shown on the mobile device to enter on the user's computer. Users are still expected to verify that non-aggregated data (VNA) on their computer screen is correct, which is why this method also applies the CVTSA information scheme.

(Li hPIN/hTAN Verify Non-Aggregated)

A bank-supplied device is proposed by Li et al. [LSH⁺12]. The hPIN/hTAN consists of a USB connector, display and a single 'OK' button. A prototype of the device is shown in Figure 5.6. Software on the user's computer is used to forward secure messages between the device and the bank. For entity authentication using hPIN, the bank sends a random digit (0-9) substitution table to the device for each new session, to be shown to the user. The user enters the required PIN in his or her computer using substituted digits. Only the bank and the device have access to the substitution table, which prevents the user's computer from eavesdropping the PIN. With hTAN for transaction authentication, users enter critical transaction details on their keyboards, which is simultaneously sent to the authentication device's trusted display.

Li hPIN/hTAN VNA During entry, the user verifies that the information is securely entered using the trusted display of the device. One press on the ‘OK’ button sends the information securely to the bank when it is deemed correct. Due to the verification of non-aggregated data (VNA), the device applies a CVTSA information scheme, although it must be noted that each transaction in a set submitted to the bank is processed individually by the user.



Figure 5.6: Prototype of the hPIN/hTAN.

Weigold Entry ENA (Weigold Entry Enter Non-Aggregated)

Several solutions are proposed by Weigold et al. Weigold Entry ENA consists of a disconnected, bank-supplied device on which the user enters critical transaction information [WH11]. A transaction-dependent authorization code (TAC) is created by the device, based on the entered transaction information. The same information is entered by the user again in his or her computer, together with the earlier created TAC. The bank receives the information and checks whether it matches the TAC. If valid, the message is accepted. Due to that the user has to enter non-aggregated transaction information (ENA), this proposal applies an ESTA information scheme.

Weigold SMS VNA (Weigold SMS Verify Non-Aggregated)

Another proposal by Weigold, et al. suggests the use of SMS text messages to send critical transaction information received by the bank back to the user for verification [WH11]. A verification code is also part of the message, which the user can enter on his or her computer to notify the bank that the received data is correct. This proposal also applies a CVTSA information scheme and is quite similar to the use of SMS text messages by Bank SMS VA, with the difference that this proposal presents non-aggregated transaction information (VNA) to the user.

Weigold Scan VNA (Weigold Scan Verify Non-Aggregated)

This is a variation of Weigold Entry ENA. A bank-issued device is used to verify entered transaction details [WH11]. Data are not entered by the user, but scanned by an optical sensor through a flickering image on the user’s computer. A verification code, shown on the display of the bank-issued device together with the critical transaction information, is entered by the user in his or her computer to indicate that information earlier received by the bank is correct. This proposal is similar to Bank Scan VNA and also applies a CVTSA information scheme to make the user verify non-aggregated data (VNA).

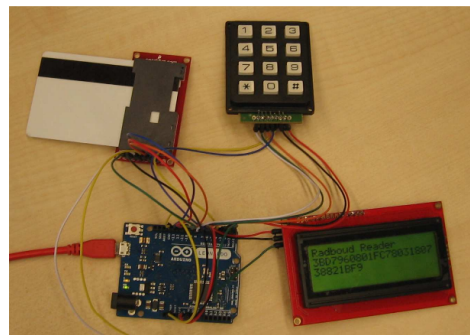


Figure 5.7: A prototype USB smart card reader with a display and keypad.

Weigold USB VNA (Weigold USB Verify Non-Aggregated)

This proposal is similar to Bank USB VA. It describes a device equipped with a display, keypad and smart card slot, connected with USB to a PC [WH11]. The device is used during transaction authentication to verify transaction data, so it also applies the CVTSA information scheme to make the user verify non-aggregated data (VNA). This solution also relies on a browser plugin to translate data from the bank server to USB commands. What makes this proposal different from Bank USB VA is that the former lets the user verify non-aggregated data (VNA) of each transaction instead of the total number of transactions and the total amount. An implementation has also been proposed by other authors [PdR13], of which a prototype can be seen in Figure 5.7.

Weigold ZTIC VNA (Weigold ZTIC Verify Non-Aggregated)

Finally, Weigold et al. mention the use of Zone Trusted Information Channel (ZTIC), as depicted in Figure 5.8. Most banks use SSL/TLS for communication through a secure channel between a user's computer and a bank [KSDC⁺14]. ZTIC uses a bank-issued device to put the client-side creation of the SSL/TLS channel outside of the untrusted domain of the user's computer [WH11]. The device provides a USB connection, a display, two buttons and a smart card slot. A smart card is used for cryptographic functions and storage. Its display and buttons are used by the user to confirm or reject login and transaction requests based on non-aggregated information (VNA), which is why this method also applies a CVTSA information scheme.



Figure 5.8: IBM's ZTIC used to verify a transaction.

5.8 Applying the mechanism

In this section, we describe how Renaud's mechanism and the feasibility dimension were used to evaluate online banking authentication methods. We note an assumption concerning entity authentication we had to make for several proposals which only specify how transaction authentication is performed. After that, we describe how the new dimension's aspects apply to the evaluated transaction authentication methods. Finally, we provide environmental factor values to represent online banking to aid fellow researchers when comparing our results with their own.

5.8.1 Proposals which lack entity authentication information

We evaluate several proposals from literature. Some proposals focus exclusively on transaction authentication and not on entity authentication. The former complements the latter, which is why both should be evaluated. We make some assumptions about the use of entity authentication for some proposals.

For proposals which only focus on transaction authentication and which do not rely on a bank-issued trusted device, we assume that a password is required for entity authentication and that it is entered in the user's computer. The initial password is chosen by the user. Methods for which we make this assumption are Starnberger Scan VNA, AlZomai Scan+SMS VNA and Weigold SMS VNA.

Weigold Entry ENA, Weigold Scan VNA and Weigold USB VNA rely on a bank-issued device. It is assumed that a user enters a PIN to unlock the device's functionality. The initial PIN is random and can be changed by the user.

Weigold ZTIC VNA and Li hPIN/hTAN VNA both describe entity and transaction authentication methods. ZTIC relies on PIN entry on the user's computer and not on the device itself. Li hPIN/hTAN VNA also relies on PIN entry on the user's computer, but the entered digits are manually substituted by the user using a table provided by the device. We assume for both methods that the initial PIN is randomly chosen and that a user can change it afterwards.

5.8.2 Applying feasibility aspects

In this section we note how we, the authors of the paper on which this chapter is based (counted as a single rater), apply the feasibility aspect on the evaluated authentication methods. The questions we asked the other raters are based on this. Note that the examples we give in this section are those we give from our own perspective. Other raters did not have to agree with these examples when answering the questions.

For each tested authentication method, the work flow expansion aspect is given a value based on whether a user needs to apply additional effort for transaction authentication. If the actions for authentication fit in a normal work flow, a value of 0.00 is given since no additional effort is required from the user aside from what is required for entity authentication. If the actions fit in the user's existing work flow but are redundant (e.g. the user has to perform a specific action twice instead of once), a value of 0.50 is given. Finally, if a work flow is expanded with one or more new kinds of additional actions, a value of 1.00 is given. Additional actions are those which are exclusive to transaction authentication and which are not considered by the other four dimensions. Examples of qualifying actions include comparing and substituting data values. Examples of actions which do not qualify are remembering and entering passwords or PINs, which are already covered by the memorability dimension and by the accessibility dimension's inclusivity aspect.

The circumvention aspect has three characteristics which can increase its value. If users are required to perform security actions which they can skip as part of their work flow, 0.33 is added to the aspect's value. The second characteristic concerns itself with whether the user interface supports secure user behavior. Banks have some control over the user interface with most transaction authentication methods, be it through a web interface, a mobile application interface or a separate user interface on a provided device. An exception is the use of text messages through a mobile phone, which relies completely on an existing user interface which is not tailored to support secure user behavior. Transaction authentication methods which rely on user interfaces which banks do not control get an additional 0.33. Finally, whenever a system's default state is insecure, another 0.33 is added. This last characteristic manifests itself if the user does nothing, yet an adversary can still launch an attack without (further) user action. An example is an adversary which has (remote) access to the user's password and to the user's smartphone. Even if a user does not initiate payments, an adversary can create a session with the bank (with the user's password) and verify transactions (through the user's smartphone).

For the clarity aspect, a minimum value of 0.33 is given to each transaction authentication method because the communication channel between the user's browser and the bank server is corruptible by malware. Some authentication methods rely on other corruptible communication channels between user and bank. The user cannot make an informed decision if all channels can provide inaccurate information. In this case, another 0.33 is added. A user interface which can present misleading, ambiguous or incomplete information is another characteristic which adds 0.33. For example, a browser can have a secure connection with a bank site and show this, whereas a mobile phone's interface for text messages does not.

5.8.3 Environmental factors

We do not apply environmental factors in our evaluation because we assume that all implemented and proposed authentication methods which we evaluate are in the same environment. Therefore, the values of the environmental factors are the same for each authentication method, and the relative score is the same for both \overline{eq} and \overline{eq}_{env} . However, we do give the factor values for the online banking environment. This allows researchers to compare authentication solutions in the online banking environment with those in other environments with different environmental factors.

The control of environment factor in the accessibility dimension is deemed 'uncontrolled' with a value of 1.50. Online banking relies on the Internet, which cannot be exclusively managed by banks.

The frequency of use and forced renewal environmental factors of the memorability dimension concern how easy it is made for the user to remember required knowledge for authentication. We use a value of 1.00 for both. It is assumed that there will be periods in which online banking is used once a week or less (e.g. during a holiday), and that users are not required to renew their passwords or PINs.

There is financial damage in a successful attack. Who is affected depends on several factors. Banks can give reimbursements, but do not always have to do so. Since a single party is affected (the bank or the user), the risk environment factor of the security dimension gets a value of 1.00. While banks in some cases hold users liable for damage, it is not their role to give sanctions as a deterrent against insecure behavior. It can be assumed that sanctions will not be enforced to keep the public image of banks positive, which gives a value of 1.50 to the security motivation environmental factor of the security dimension.

Banks can apply pattern-based recognition of malicious transactions, giving a value of 1.00 to the vulnerability dimension's auditing environment factor.

Transactions are usually non-reversible by the end-user. The feasibility dimension's user error tolerance environment factor gets a value of 1.50.

5.8.4 Performing a multi-user evaluation

As we described in Section 5.6, one way to decrease the amount of subjectivity when evaluating something is to evaluate the same subject with the same method by multiple raters. We do this with Renaud's mechanism itself (described in Section 5.4) and with our expansion (proposed in Section 5.5).

Based on the modifiers as shown in Table 5.1 on page 111, it can be expected that some parts of Renaud’s mechanism will be more sensitive to subjectivity compared to others. To reduce the amount of time required to perform the evaluations we only prepared questions for the most subjective parts of Renaud’s mechanism, and all dimensions and aspects of our expansion. Most aspects of Renaud’s mechanism are quite objective. For example, whether extra hardware or software is required (measured by the accessibility dimension’s special requirements aspect) is not based on opinion but on clearly stated specifications of the authentication method. The subjective parts of Renaud’s mechanism that we presented to the raters are the requirement for technical expertise (from the accessibility dimension’s special requirements aspect) and whether much time is required to perform authentication (from the accessibility dimension’s convenience aspect). We consider that the other required values for Renaud’s mechanism can clearly be derived from the specifications we have of the authentication methods. This allows us to focus most of the raters’ attention to our expansion, where we will let them rate each aspect in full. Focusing on the subjective parts of Renaud’s mechanism and on our added dimension allows us to get an insight in how sensitive the mechanism (original and with our expansion) is to subjectivity.

#	Question	Answers	Relates to	
			Dimension	Aspect
1	Does the user require technical expertise to prepare or use the authentication method?	yes/no	Accessibility	Special requirements
2	Does the user require much or little time to authorize a transaction?	much/little	Accessibility	Convenience
3	When replacing password authentication, does the user now have to perform redundant actions?	yes/no	Feasibility	User effort cost
4	When replacing password authentication, is the user now required to perform new actions?	yes/no	Feasibility	User effort cost
5	Are one or more of the devices used for authorization protected against remote attacks?	yes/no	Feasibility	Circumvention
6	Can the user know or check that he or she is using the authorization system of the bank?	yes/no	Feasibility	Circumvention
7	Is it possible for the user to skip steps in the authorization process?	yes/no	Feasibility	Circumvention
8	Are all communication channels between user and bank secure against adversaries?	yes/no	Feasibility	Clarity
9	Of any information the user is required to verify, is the information complete?	yes/no	Feasibility	Clarity
10	Of any information the user is required to verify, is the information accurate?	yes/no	Feasibility	Clarity
11	Is the primary user interface capable of showing, misleading, ambiguous or incomplete information?	yes/no	Feasibility	Clarity

Table 5.4: Summary of the questions asked to the raters for each authentication method.

For each of the 12 authentication methods we asked the same questions in the same order. The order of the authentication methods to evaluate was also the same for all raters due to restraints in the system we used to perform the survey. This order is the same as the order of the authentication methods in Section 5.9, from left to right.

The questions are listed in Table 5.4 in a condensed form in English, and com-

pletely in Dutch (the language of the survey) at the end of this section. In the survey, the questions were a bit more extensive and each question had some background information that could help raters if they did not understand the context. In addition to the possible answers shown in Table 5.4 and as discussed in Section 5.6, each question could also be skipped by answering ‘I do not know’, and raters were encouraged to pick this option if they could not think of an answer.

The raters were personally asked to participate in the survey and given a personal URL for participation, which they could do at their workplace or at home. Due to the length of the survey, they could pause and continue it at anytime they wanted at any location they wanted, so there was no time pressure. Before, during and after the survey the raters were given the continuous opportunity to ask questions. The raters did not communicate with each other while performing the survey.

For the experiment, we informed ourselves of the rules stated by the ethical review board of Open University of the Netherlands (known as Commissie Ethische Toetsing Onderzoek) whether a review would be required. The review board’s main focus is medical examination, and the experiment did not require a review or approval. We were careful in our judgment on whether the experiment was safely performed, and also asked the raters (each of them a researcher and familiar with research ethics) whether they saw any ethical problems before the survey was conducted. The raters were not pressured to participate in the survey, and they were told several times explicitly that they can pause or cancel the survey at any time without stating a reason and without any repercussion. No personal information was asked or collected in the survey. A link between answered questions and personal information of the raters (name and email address) was only used for administrating the survey, and no further personal information was collected or processed. Any questions that we asked about the survey after it was finished were done in person. We reduced risks of personal damage as much as we could by only making raters evaluate the authentication methods from a theoretical perspective. For our experiment we were only interested in the opinions of the raters, not in how they perform actions themselves. Therefore, we did not request the raters to test or use any of the authentication methods (e.g. with their own bank accounts), neither did we make any implication that such an action would be necessary to partake in the survey, nor did we register any such actions.

7 experts rated the authentication methods defined in Section 5.7 using the questions we prepared. For our experiment, we considered an expert as someone whose research field and work (indirectly) relates to transaction authentication in online banking. 4 raters have a technical background and their research relates to technology that is used in, among others, online banking. The other 3 raters have backgrounds in social sciences, and their research focuses on combating online banking fraud from an organizational perspective (of law enforcement, banks and criminals), and improving the self-defense of users against external threats.

The raters were provided a summary of the workings of all authentication methods. Also, for the proposed methods they were given copies of the work which propose these [SFG09, AAJ10, WH11, LSH⁺12].

Section 5.10.4 continues with the results of the evaluation as performed by multiple raters.

For reference, this section closes with a list of the original questions as they

were asked in Dutch. This list includes the questions themselves, case descriptions and help texts. See Table 5.4 to map questions to dimensions and aspects, and the possible answers.

Question 1

Question:

Heeft de gebruiker technische expertise nodig om de authenticatiemethode in gebruik te nemen of te gebruiken?

Help text:

Let op dat onder ‘in gebruik nemen’ de initiële configuratie kan vallen (installatie hard/software, etc.). Iemand met een achtergrond in IT kan de autorisatiemethode ongetwijfeld gebruiken zonder een handleiding te lezen. Het gaat bij deze vraag juist om niet-IT’ers. Kunnen die de methode alleen gebruiken met (zelf)scholing?

Question 2

Question:

Denk je dat er veel of weinig tijd nodig is om een transactie te autoriseren met deze methode?

Help text:

Let op dat deze vraag alleen de acties betreft die de gebruiker elke keer moet uitvoeren bij het opzetten van financiële transacties. Het aansluiten van hardware en installeren van software vallen hier bijvoorbeeld niet onder, omdat dit vaak maar eenmalige acties betreffen.

Questions 3 and 4

Case description:

De bank wil de omschreven autorisatiemethode invoeren, maar heeft dit nog niet gedaan. De gebruiker hoeft in de oude situatie alleen maar in te loggen met een wachtwoord, en geeft opnieuw zijn wachtwoord om transacties te autoriseren. Toch moet de gebruiker in de toekomst gebruik gaan maken van de nieuwe autorisatiemethode. Of en hoe wordt de workflow van de gebruiker door de introductie van het nieuwe autorisatiemethode beïnvloed?

Help text:

Door het invoeren en verplicht gebruik van de methode kan het zijn dat de gebruiker plots één of meer acties meerdere malen moet uitvoeren. Het kan bijvoorbeeld gaan om een actie die de gebruiker bij het invullen van transactiegegevens eenmaal uitvoert, en dit nogmaals moet herhalen tijdens het autoriseren van de transactie. Het gaat bij de tweede vraag expliciet om nieuwe acties en niet om redundante acties. Als de gebruiker iets nieuws moet doen dat hij vroeger (voor het in gebruik nemen van het systeem) niet deed, dan is het antwoord ja. Let op dat de initiële installatie van de nieuwe autorisatiemethode niet valt onder deze vraag. Het gaat erom dat de gebruiker een nieuwe actie moet uitvoeren, elke keer als die een transactie aanmaakt.

Question 3:

Moet de gebruiker bestaande acties nu redundant uitvoeren?

Question 4:

Moet de gebruiker nieuwe acties uitvoeren?

Question 5

Question:

Zijn één of meer apparaten van de autorisatiemethode beveiligd tegen misbruik op afstand door een aanvaller?

Help text:

Let op dat het gaat om beveiliging tegen de situatie waarbij de aanvaller zelf alle benodigde beveiligingselementen kan gebruiken zonder tussenkomst van de gebruiker. Voor initiële toegang kan de gebruiker een element zijn (bijvoorbeeld via social engineering, om de deur open te zetten voor de aanvaller). Bij verdere toegang van het systeem (en mogelijk herhaaldelijke aanvallen) kan de aanvaller direct een verbinding maken en het systeem misbruiken zonder dat de gebruiker nog iets hoeft te doen om de aanval te faciliteren. Let op dat een beveiliging ook intrinsiek kan zijn. Bijvoorbeeld: een door de bank uitgegeven apparaat dat geen dataoverdracht heeft met een computer en niet is aangesloten op een computernetwerk is intrinsiek veilig tegen aanvallen op afstand, omdat het niet mogelijk is om het apparaat op afstand te benaderen.

Question 6

Question:

Kan de gebruiker weten of controleren dat daadwerkelijk van het autorisatiesysteem van de bank gebruik gemaakt wordt tijdens het gebruik?

Help text:

De vraag betreft of de gebruiker ondersteund wordt in het veilig gebruik van het autorisatiesysteem. Het gebruik van een apparaat uitgegeven door de bank geeft bijvoorbeeld zekerheid dat de gebruiker het autorisatiesysteem van de bank gebruikt. Het gaat immers om een ander apparaat waarvan de gebruiker weet (of tot in redelijkheid kan weten) dat het van de bank afkomt.

Question 7

Question:

De gebruiker vindt het autorisatiesysteem maar omslachtig. Is het mogelijk voor de gebruiker om stappen in het autorisatieproces over te slaan?

Help text:

Het gaat om de mogelijkheid voor de gebruiker om het proces korter te maken. De

uitkomst moet vanuit het perspectief van de gebruiker hetzelfde zijn (een verzonden transactie).

Question 8

Question:

Zijn alle communicatiekanalen tussen de gebruiker en bank veilig tegen aanvallen door derden bij het gebruik van deze autorisatiemethode?

Help text:

Een aanval kan gedefinieerd worden als de mogelijkheid van een externe partij om ongemerkt informatie te beïnvloeden terwijl het onderweg is van de klant naar de bank, of vice versa.

Questions 9 and 10

Case description:

Een gebruiker maakt een aantal transacties aan en gebruikt deze methode om ze te autoriseren. Heeft de gebruiker de beschikking over de juiste informatie om de gevraagde acties voor het autoriseren van transacties veilig uit te voeren?

Help text:

Volledig: bedenk dat de gebruiker mogelijk iets moet weten per transactie. Is die informatie beschikbaar? Accuraat: is het mogelijk voor derden om de betreffende informatie te wijzigen zonder dat dit opgemerkt wordt?

Question 9:

Is de informatie volledig?

Question 10:

Is de informatie accuraat?

Question 11

Question:

Is de primaire gebruikersinterface van het autorisatiemiddel in staat om misleidende, dubbelzinnige of incomplete informatie te tonen?

Question 11 help text:

Deze vraag heeft betrekking op informatie die de gebruiker moet verwerken voor het autoriseren van een transactie. Misleidend: de informatie hoeft niet te zijn wat de gebruiker denkt dat het is, bijvoorbeeld omdat een derde partij iets stuurt of wijzigt aan informatie verstuurd door de bank. Dubbelzinnig of incompleet: ondanks dat de informatie legitiem kan zijn, is de informatie niet voldoende voor de gebruiker om veilig transacties uit te voeren.

5.9 Research data of the evaluated authentication methods

Dimension(s)		Authentication method reference →	Bank USB CR	Bank SMS VA	Bank USB VA	Bank Scan VNA	Starnberger Scan VNA	AlZomai Scan+SMS VNA	Li hPIN/hTAN VNA	Entry ENA	SMS VNA	Scan VNA	USB VNA	ZTIC VNA
↓	Description	Formula	NN	VA	VA	VNA	VNA	VNA	VNA	ENA	VNA	VNA	VNA	VNA
Accessibility	Special req.	x	0.83	0.33	0.83	0	0.67	1	1	0	0.33	0	0.67	0.67
	Convenience	y	0.50	0.50	1	1	1	1	1	1	1	0.50	0.50	0.50
	Inclusivity	z	0.67	0.33	0.67	0.67	0.33	0.33	0.67	0.67	0.33	0.67	0.67	0.67
	Deficiency	$ad = \sqrt{x^2 + y^2 + z^2}$	1.18	0.68	1.46	1.20	1.25	1.45	1.56	1.20	1.10	0.84	1.07	1.07
	Percentage	$adp = (1 - \frac{ad}{\max(d)}) * 100\%$	32%	61%	16%	31%	28%	16%	10%	31%	36%	52%	38%	38%
Memorability	Retrieval str.	x	1	1	1	1	1	1	1	1	1	1	1	1
	Meaningfulness	y	0.67	0.33	0.67	1	0.33	0.33	0.67	0.67	0.33	0.67	0.67	0.67
	Depth of proc.	z	1	1	1	1	1	1	1	1	1	1	1	1
	Deficiency	$md = \sqrt{x^2 + y^2 + z^2}$	1.56	1.45	1.56	1.73	1.45	1.45	1.56	1.56	1.45	1.56	1.56	1.56
	Percentage	$mdp = (1 - \frac{md}{\max(d)}) * 100\%$	10%	16%	10%	0%	16%	16%	10%	10%	16%	10%	10%	10%
Security	Predictability	x	1	1	1	0	1	1	1	1	1	1	1	1
	Abundance	y	0	0	0	0	0	0	0	0	0	0	0	0
	Disclosure	z	1	1	0.5	0.5	1	1	0.5	0.5	1	0.5	0.5	1
	Deficiency	$sd = \sqrt{x^2 + y^2 + z^2}$	1.41	1.41	1.12	0.50	1.41	1.41	1.12	1.12	1.41	1.12	1.12	1.41
	Percentage	$sdp = (1 - \frac{sd}{\max(d)}) * 100\%$	18%	18%	35%	71%	18%	18%	35%	35%	18%	35%	35%	18%
Vulnerability	Confidentiality	x	1	1	1	1	1	1	1	1	1	1	1	1
	Privacy	y	0	0	0	0	0	0	0	0	0	0	0	0
	Breakability	z	1	1	0.33	0.33	1	1	0.33	0.33	1	0.33	0.33	1
	Deficiency	$vd = \sqrt{x^2 + y^2 + z^2}$	1.41	1.41	1.05	1.05	1.41	1.41	1.05	1.05	1.41	1.05	1.05	1.41
	Percentage	$vdp = (1 - \frac{vd}{\max(d)}) * 100\%$	18%	18%	39%	39%	18%	18%	39%	39%	18%	39%	39%	18%
1 - 4	Total quality coeff.	$\overline{eq} = \max(d) - d$	1.36	1.97	1.73	2.44	1.40	1.20	1.63	1.99	1.55	2.36	2.12	1.47
	Overall percentage	$\overline{dp} = \frac{\overline{eq}}{\max(\overline{eq})} * 100\%$	19.6%	28.4%	25.0%	35.2%	20.2%	17.3%	23.5%	28.7%	22.4%	34.1%	30.6%	21.2%
	Ranking		11	5	6	1	10	12	7	4	8	2	3	9
Feasibility	Work flow expansion	x	0.5	1	1	1	1	1	1	1	1	1	1	1
	Circumvention	y	0.33	0.67	0	0.33	1	0.67	0.67	0	1	0.33	0.33	0.33
	Clarity	z	0.67	0.67	1	0.33	0.67	0.67	0.33	0.33	0.67	0.33	0.33	0.33
	Deficiency	$ed = \sqrt{x^2 + y^2 + z^2}$	0.90	1.38	1.41	1.10	1.56	1.38	1.25	1.05	1.56	1.10	1.10	1.10
	Percentage	$edp = (1 - \frac{ed}{\max(d)}) * 100\%$	48%	20%	18%	36%	10%	20%	28%	39%	10%	36%	36%	36%
1 - 5	Total quality coeff.	$\overline{eq} = \max(d) - d$	2.19	2.32	2.05	3.07	1.57	1.55	2.11	2.67	1.71	2.98	2.75	2.09
	Overall percentage	$\overline{dp} = \frac{\overline{eq}}{\max(\overline{eq})} * 100\%$	25.3%	26.8%	23.7%	35.5%	18.1%	17.9%	24.4%	30.8%	19.7%	34.4%	31.8%	24.1%
	Ranking		6	5	9	1	11	12	7	4	10	2	3	8
Relative difference between dimensions 1-4 and 1-5			5.7%	-1.6%	-1.3%	0.2%	-2.1%	0.6%	0.8%	2.1%	-2.6%	0.4%	1.2%	2.9%
Ranking difference			5	0	-3	0	-1	0	0	0	-2	0	0	1

Table 5.5: Our research data. Each authentication method was first quantified using the original four dimensions of Renaud’s mechanism. The results for the original four dimensions are noted on the rows for dimensions ’1-4’, and the results for the original four dimensions plus the feasibility dimension on the rows for dimensions ’1-5’. Total quality coefficient is the resulting value of the mechanism and represents the fit of an authentication method within the specified dimensions. A higher value implies a better fit. Its maximum value can be calculated by $\max(\overline{eq}) = n * \max(d)$, where n is the number of dimensions and $\max(d)$ is the maximum value for a dimension’s deficiency ($\max(d) = 1.73$). Overall percentages (given in **bold**) can be used to at a glance compare total quality coefficients for the original four dimensions with the same values for five dimensions.

5.10 Resulting values

This section notes the results of our evaluation of implemented and proposed transaction authentication methods. The content of this section is based on our research data, which can be found in Section 5.9.

5.10.1 Effects of the feasibility dimension

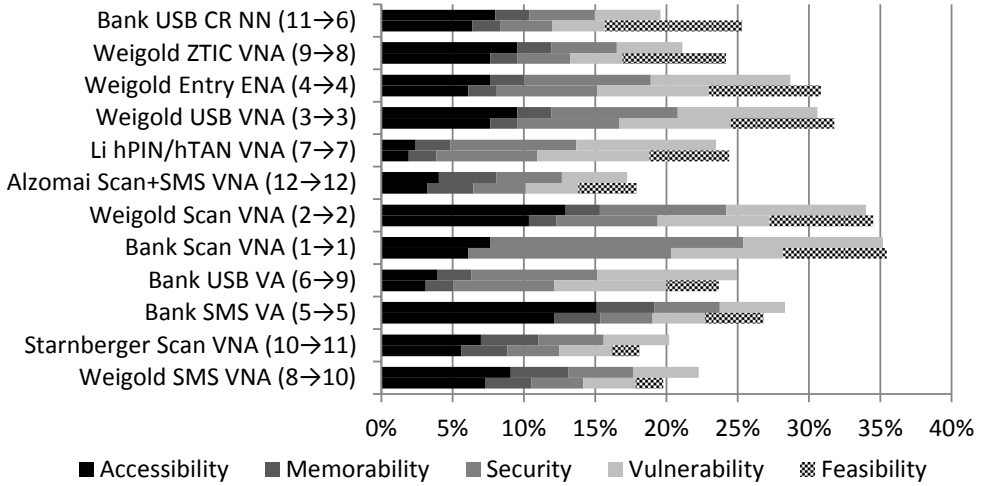


Figure 5.9: Relative total quality coefficient values of the evaluated authentication methods of the original four dimensions, and the influence of the feasibility dimension. The list is sorted by the amount of influence the feasibility dimension has (from positive, to neutral, to negative). The maximum relative value of 100% represents the best fit in the original four dimensions of Renaud’s mechanism (when the feasibility dimension is ignored) or all five dimensions (when the feasibility dimension is included).

Figure 5.9 visualizes the influence the feasibility dimension has on the overall percentages. Every authentication method has two bars. Each top bar illustrates the overall quality of the authentication method within Renaud’s mechanism. Each bottom bar does the same, but also includes the feasibility dimension. A value of 100% for any bar represents the maximum value of the total quality coefficient $\bar{e}q$ based on the number of dimensions as noted in Sections 5.3 and 5.5 (6.93 for the original four dimensions, 8.66 for all five dimensions). The colors inside each bar show how each dimension’s deficiency value d contributes to the overall quality. The first four colors in each top bar are also present in the corresponding bottom bar in a compressed form. This is because the addition of the feasibility dimension does not change the absolute d values of the original four dimensions. All it does is influence $\bar{e}q$ and its maximum possible value. When the feasibility dimension is added, the original dimension’s retain their absolute d values but will relatively make up less of $\bar{e}q$, which is why their colors on the bottom bar take up less space. An effect of this

is that the feasibility dimension can have a positive or negative effect on the relative $\bar{e}q$ value.

The feasibility dimension has the strongest effect on Bank USB CR NN. This is because it is the only authentication method that does not get the maximum penalty for work flow expansion while still scoring averagely for the circumvention and clarity aspects. For circumvention, the system is in a secure state by default and it is not possible for the user to circumvent security in any way. As for clarity, the only redeeming quality the card reader has is that its limited interface does not give the user a false impression of its functionality.

Weigold ZTIC VNA also shows a significant better quality due to the addition of the feasibility dimension. Although it shares the maximum work flow expansion with most other authentication methods because new user actions are introduced, it is quite favorably for the circumvention and clarity aspects. The system's default state is secure, and unlike Bank USB CR NN the user interface does support secure user behavior. The only negative modifier that applies to the circumvention aspect is that the user can subvert security due to inconvenience. As for clarity, the only penalty Weigold ZTIC VNA gets is that at least one in- and output channel (the communication channel between browser and bank) is corruptible.

Weigold Entry ENA, Weigold USB VNA, Li hPIN/hTAN VNA, AlZomai Scan+SMS VNA, Weigold Scan VNA and Bank Scan VNA receive a minor increase in overall percentage but are barely affected due to the feasibility dimension's mediocre deficiency value.

The final four evaluated authentication methods are negatively affected by the feasibility dimension. Most notable is Weigold SMS VNA, which is fully penalized for both the work flow expansion and circumvention aspects. The only redeeming quality it has in this dimension is for the clarity aspect, for which the raters have stated that all information necessary to make a good decision is available during transaction authentication.

The added dimension changed 5 out of 12 ranks of the transaction authentication methods. The method with the highest overall percentage represents the best fit within the context of all dimensions, and therefore has the highest rank. With the original four dimensions, Bank USB CR NN had quite a bad overall quality, mostly due to its poor fit in the memorability dimension and also fitting quite poorly in the other dimensions. The feasibility dimension brings some of its commendable characteristics to the surface, boosting its overall percentage and giving it a 6 rank increase. Weigold ZTIC VNA received a smaller increase, but enough to make it rise one rank. As for rank decreases, Weigold SMS VNA loses a rank due to the earlier discussed bad fit in the feasibility dimension. The same is true for Starnberger Scan VNA, who drops a rank because it has the same poor fit. Bank USB VA drops three ranks, but this has less to do with its mediocre fit in the feasibility dimension. Instead, it drops three ranks due to the good fit Bank USB CR NN, Weigold ZTIC VNA and Li hPIN/hTAN VNA have.

5.10.2 Overall evaluation

We note the fit of the evaluated authentication methods within the dimensions of Renaud's mechanism and our added dimension. These results can only be used to

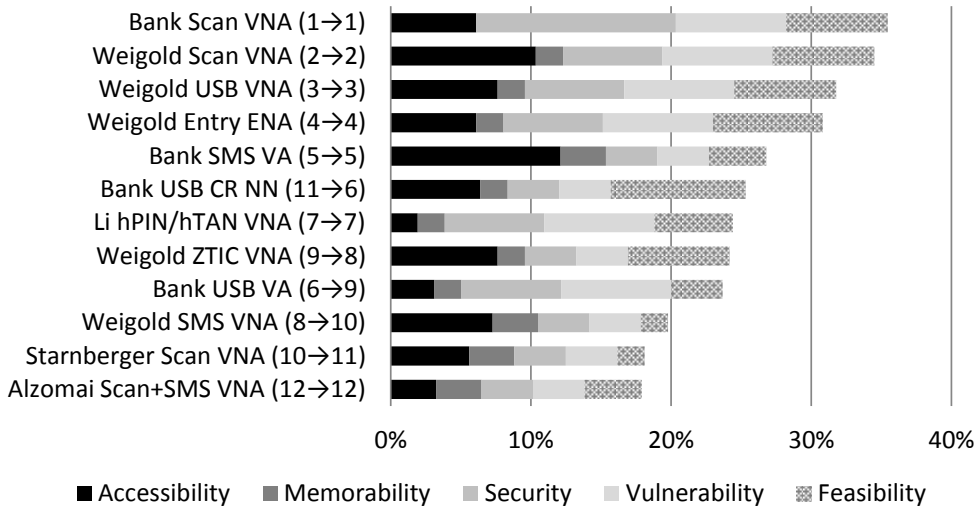


Figure 5.10: Relative total quality coefficient values of the evaluated authentication methods in all five dimensions. The list is sorted by the relative total quality coefficient. The maximum value of 100% represents the best fit in all dimensions.

rate the authentication methods on the criteria of the applied evaluation mechanism.

An overview based on the fulfillment of each dimension and based on the overall percentage is given in Figure 5.10. The data represented in this Figure is also depicted by the bottom bars in Figure 5.9, but Figure 5.10 sorts the authentication methods by the relative total quality coefficient to make it easier to compare the qualitative fit of the methods with each other.

With the feasibility dimension included, Bank Scan VNA and Weigold Scan VNA have the highest overall percentage. They therefore have the best fit within the context of all five dimensions. This does not state that they have the best fit in each dimension. For example, while Bank Scan VNA has an exceptionally good fit in the security dimension, it has the worst fit in the memorability dimension. The latter is because the used authentication device relies on a bank card with a PIN code which cannot be changed (a bank-specific policy), which gives the most negative value to the memorability dimension's meaningless aspect. The other implemented methods allow the user to change PINs or passwords, and we assumed that this was also true for the proposed methods.

The evaluated methods are grouped by implementations and proposals for further comparisons. The proposal group has been split into two groups to compare the five proposals from Weigold and to improve the readability of the graphs. Radar charts are used to provide an overview of the different dimensions' fits. The center of each radar chart represents 0%, while each line from inside to outside represents an additional 20%, making the total range 0% to 80%. Note that we do not rely on absolute numbers, but instead use the relative weights of the qualitative aspects to observe the fit within the dimensions of different authentication methods.

As shown in Figure 5.11, it is quite easy to see where methods score favorably. Bank Scan VNA has a good fit in the security dimension because all secret key

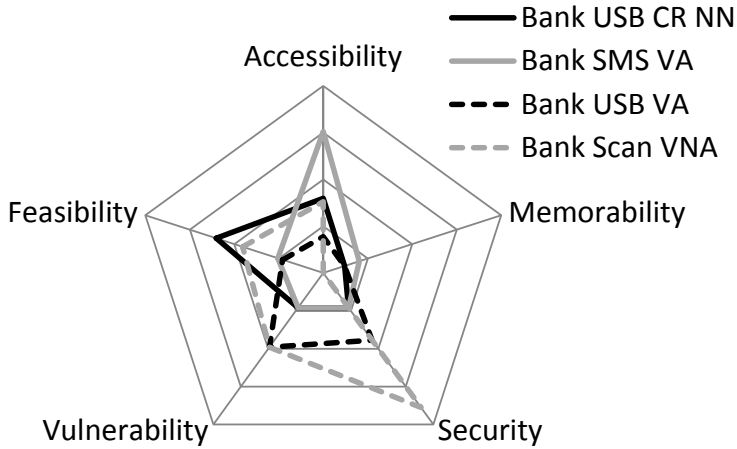


Figure 5.11: Dimension fulfillment for implemented bank methods.

material used by this method (including the user's PIN) is distributed randomly and cannot be chosen by the user. This is different from Bank USB CR NN, Bank SMS VA and Bank USB VA, which do allow users to change their secret knowledge. The good fit of Bank SMS VA in the accessibility dimension can be explained that it does not require software or technical expertise to use, authentication does not take a lot of time and users with mobility or sensory disabilities are not excluded from using the authentication method.

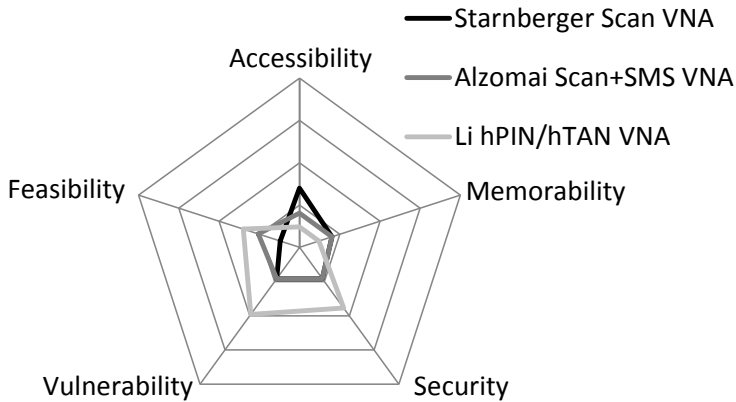


Figure 5.12: Dimension fulfillment for various proposals.

The proposals made by Starnberger et al., AlZomai et al. and Li et al. do not fit particularly well, as depicted in Figure 5.12. The line of Starnberger Scan VNA is hidden by the line of AlZomai Scan+SMS VNA in the memorability, security and vulnerability dimensions, which can be explained by that both methods are quite similar. Their differences are in the accessibility dimension (where it is thought that for Starnberger Scan VNA no technical expertise is required) and in the feasibility dimension (where it is thought that with AlZomai Scan+SMS VNA the user is unable

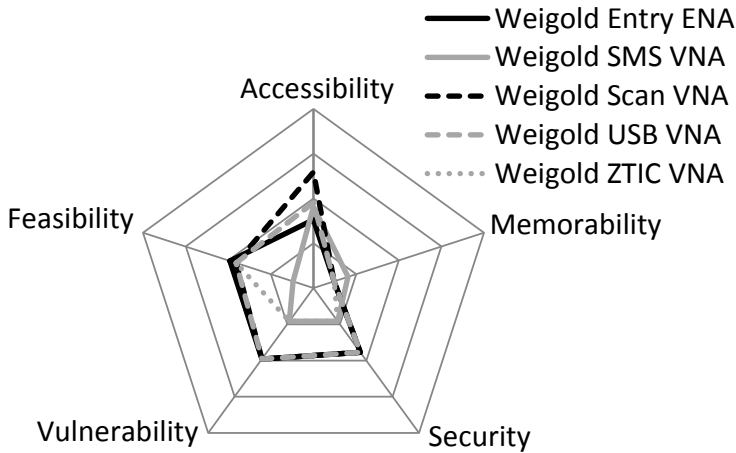


Figure 5.13: Dimension fulfillment for Weigold's proposals.

to subvert security due to inconvenience). Li hPIN/hTAN VNA has a slightly better fit in the security and vulnerability dimensions since the required PIN to login to the bank's site is only entered in the user's computer with substitution digits, which the user received in a secure manner. As for the feasibility dimension, it has a better fit compared to the others because it does not rely on a smartphone as an authentication device, and therefore less susceptible to remote intrusion by an adversary.

As shown in Figure 5.13, all methods proposed by Weigold et al. fit quite poorly in the memorability dimension. However, the earlier discussed authentication methods have this poor fit as well. Syntactical passwords tax the memorability dimension heavily.

Weigold Scan VNA does neither require technical expertise to use, nor does the user require a lot of time to use it. This makes it have the best fit in the accessibility dimension. Weigold SMS VNA fits the worst in the feasibility dimension, because it requires a lot of effort from the user to use (new actions are introduced in the user's work flow), it can be circumvented in every way that is evaluated by the mechanism and it relies on a smartphone, which is seen as an unreliable authentication platform.

5.10.3 Information scheme influence

We mentioned at the start of Section 5.7 three information schemes for transaction authentication methods. Bank USB CR NN uses TTA, Weigold Entry ENA uses ESTA and all other implemented methods and proposals apply CVTSA. Before we started our evaluation, we expected that TTA would rank low since it does not offer the user the option to securely verify transactions (like CVTSA) or the requirement to enter transactions in a secure device for automated verification (like ESTA), and therefore does not offer protection against malware attacks which change transaction information. We also expected that TTA and CVTSA would rank lower compared to ESTA, since a user can with the former (intentionally or not) perform the verification process incorrectly or skip it entirely.

Our first expectation was incorrect. Bank USB CR NN settles on a still admirable

sixth position. It fits decently in the accessibility dimension due to that users do not need a lot of time to use it. The authentication method also has the highest fit in the feasibility dimension, which it mostly owes to that users are only required to perform redundant actions and not new actions during transaction authentication.

Our second expectation is also incorrect. Weigold Entry ENA (ESTA) has a high position, but is surpassed by Bank Scan VNA, Weigold Scan VNA and Weigold USB VNA (all CVTSA). Although Weigold Entry ENA scores high in the feasibility dimension due to that it cannot be circumvented in any way that the evaluation mechanism considers, it does not score exceptionally high in the other dimensions.

The evaluation we performed does not rule out any information scheme, which suggests that the evaluation mechanism can be used to compare authentication methods with different underlying schemes.

5.10.4 Variation between the raters

As we noted in Section 5.8.4, we did not perform the evaluation alone. For all aspects in the feasibility dimension and the most subjective questions in Renaud’s original mechanism we used the average answer of seven raters.

It is unlikely that seven raters would always have the same opinion, especially since we expect that some aspects of Renaud’s mechanism and possibly our dimension are sensitive to subjectivity (as earlier discussed in Section 5.8.4). Indeed, only 32.6% of the questions were answered unanimously. For 64.4% of the questions an answer was provided by the majority. No majority was reached for the remaining 3.0% since for each of these questions three raters answered yes, three raters answered no and the final rater did not know the answer. One of seven raters did not know the answer for 8.33% of all questions.

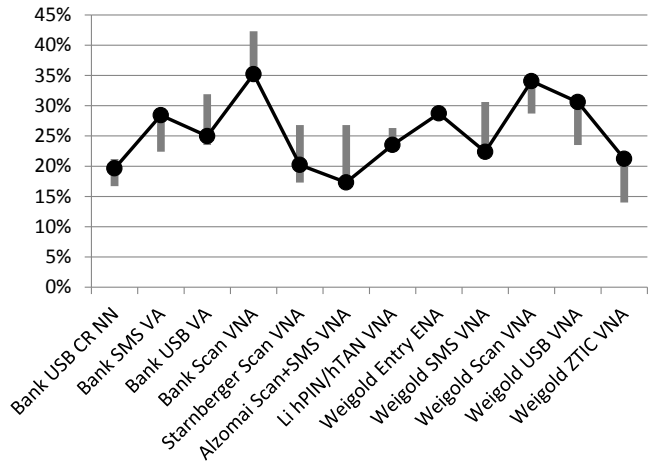


Figure 5.14: Inter-rater variation of total quality coefficients with the dimensions of Renaud’s mechanism. The black dots represent the total quality coefficients of the average of the raters’ answers (as noted in Table 5.5), while the top and bottom of each gray bar represent respectively the highest and lowest total quality coefficients coming from the individual sets of answers.

Figure 5.14 gives an overview of the total quality coefficient of each authentication method based on the four dimensions of Renaud's mechanism. It also shows the amount of variation there is between the sets of answers provided by the raters, giving an indication of how certain the raters are of the total fit within the mechanism.

Weigold Entry ENA has the similar total quality coefficient between all sets of answers, followed by Li hPIN/hTAN VNA and Bank USB CR NN. All other individual sets have more variation that either is more positive or negative compared to the average answer set. The one which stands out most is Bank Scan VNA, of which even its minimal value (corresponding with the opinion of most raters) is higher compared to the others. At the opposite spectrum of the average answers is AlZomai Scan+SMS VNA, which fit the lowest. Still, there were some answer sets which could provide a higher outcome if the evaluation would solely rely on them.

The amount of variation can be explained by that the two tested aspect modifiers in the accessibility dimension (the need for technical expertise for the special requirements aspect, and a long authentication time for the convenience aspect) are quite ambiguous.

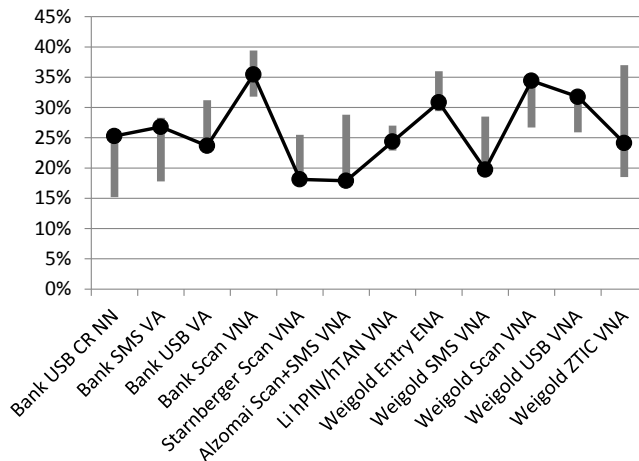


Figure 5.15: Inter-rater variation of total quality coefficients with the dimensions of Renaud's mechanism and the proposed feasibility dimension.

Figure 5.15 shows the same graph for all five dimensions. The earlier discussed increase of the total quality coefficient for Bank USB CR NN is clearly visible, but it also shows that individual raters do not always come to such a final value.

Bank SMS VA, Bank USB VA, Li hPIN/hTAN VNA, Weigold Scan VNA and Weigold USB VNA stay more or less the same, both in the combined answer set and in variation of the individual answer sets. A significant increase in the amount of variation of Weigold ZTIC VNA can be seen, while the average value only climbs marginally. This can be explained by that two raters had the exact opposite values for the accessibility dimension. In addition, the rater with the more favorable values also gave the most favorable values to all aspects in the feasibility dimension, while the other rater was more critical.

As noted in Section 5.8.4, we also performed the role of a single rater. We expect-

ted that implemented and proposed authentication methods which used user-owned mobile devices would score lower compared to those which use a more secure environment offered by a bank-provided device. This is true for the proposed AlZomai Scan+SMS VNA, Starnberger Scan VNA and Weigold SMS VNA, but not for the implemented Bank SMS VA. The only effective difference between Bank SMS VA and Weigold SMS VNA is that the former uses text messages to make the user verify aggregated transaction data, whereas the latter proposes to do the same using non-aggregated transaction data. Therefore, it would be expected that Bank SMS VA would rank at least worse compared to Weigold SMS VNA, and not that it would rank so high as it does now. We asked one rater about this difference, who told us that he actually uses Bank SMS VA in daily life. The rater thinks that this method does not take much of his time during authentication, explaining that he skips the verification check over the provided transaction data. He believes that the verification is not worth his time due to a perceived low security risk.

The difference between the proposed and implemented text message-based methods combined with what the rater told us implies that there is a certain bias for methods which raters are familiar with. This presents a subjectivity that comes forth from the perspective in which raters answer questions. Combined with the earlier discussed ambiguous terms, it is suggested that the subjectivity came from two sources:

- Inherent subjectivity. This comes from the evaluation mechanism itself and the questions we asked based on it.
- Hidden subjectivity, which comes from the raters their personal experience with the matter.

The former can possibly be reduced by providing more clear information. For the evaluation, we offered the original proposals for applicable authentication methods, which might have been too much information. The latter might be reduced by more explicitly asking questions. For example, we asked whether users would require much time to use the method. Instead, we could have asked whether the average user or most users would require much time to use the method. Different raters would still have different ideas about what an average or common user would be, but they might be more inclined to not think about the question from their own perspective.

As noted in Section 5.6, the order in which raters were required to rate the authentication methods was the same. This can be considered a bad practice due to that it can induce order bias. For example, a rater might be more positive about the first authentication method since he or she starts with a fresh look. By the time the rater is rating the last few authentication methods, tiredness might make them rate more negatively. We tried to reduce the impact of any such bias this by telling the raters that they can pause and resume the survey at their leisure. The order in which the methods were rated is the same as shown in Figures 5.14 and 5.15 (from left to right). As the Figures show, there does not seem to be any order bias. Methods were not rated overly negative or positive at the beginning or the end of the survey. It can be reasoned that the large variance in answers given for Weigold ZTIC VNA (the last rated authentication method) exists due to that this method differs quite a bit from the other methods. For example, this method is the only one

using a device directly positioned in the communication flow between browser and bank. This increase in complexity might have confused the raters.

To conclude, Figures 5.14 and 5.15 do indicate that it is useful to have multiple raters perform evaluations, either with Renaud’s mechanism solely or with our expansion. There is quite a bit of subjectivity involved when answering the questions that need to be asked to use the mechanism, which is why it is unlikely that a single rater can state what is true and what is not.

5.11 Limitations, discussion and further research

The first step we performed was the examination of different evaluation mechanisms and frameworks. As noted in Section 5.3, we also examined Bonneau’s framework [BHVOS12]. For the scope of our work, we considered this framework wide in aspects and dimensions, but too shallow in its output since it does not quantify qualitative characteristics, nor does it provide indications of quality on multiple levels. For the work of others, Bonneau’s framework could be modified to provide both quantified results on multiple levels while being less ambiguous on the values assigned to aspects. Considering the number of aspects this would mean a lot of work, but it has the potential to provide more detailed comparisons between authentication methods compared to our extension of Renaud. This partly comes from the inclusion of a deployability dimension, which also includes the potential cost to implement and maintain such a system.

Seven experts evaluated the subjective parts of Renaud’s mechanism. We chose these parts since it would save the raters time which could be spend on rating our dimension instead. This presents a potential limitation in the multi-user evaluation since the choice of what is objective and what is subjective in Renaud’s framework might be a subjective choice by itself. What we consider objective might be considered as subjective by others, in which case the relevant aspects should also be given a multi-user evaluation.

One of the raters told us that with an evaluation method that he personally uses in daily life that he skips some of the instructions for secure use due to that he perceives the risk as low. This complies with Herley’s vision [Her09, Her14]: the user’s time is valuable and they will not spend it on security if not strictly required and if they perceive the need to do so. It would be useful to examine how users objectively work with and subjectively experience different authentication methods for online banking over a longer period of time, and to find out what triggers them to use the system in a more secure manner. This could be tested in a simulated online banking environment which needs to be flexible enough to support existing and new authentication methods.

We considered the differences in Renaud’s mechanism when a new dimension is added. Further research can examine how the mechanism and dimensions can be changed or enhanced to take more aspects into account. Instead of modifying Bonneau’s framework, a deployability dimension could also be considered as an extension for Renaud’s and our work.

We discovered that the memorability dimension’s meaningfulness aspect and the security dimension’s predictability aspect of Renaud’s mechanism are linked with each other whenever a knowledge factor is present. The possible values depend on

Aspect ↓ / Source →	Random	User
Meaningfulness	1	0-0.67
Predictability	0	0.5-1
Effective range	6.2% - 96.5%	

Table 5.6: Linked aspects.

whether the knowledge is randomly or user chosen. This limitation of the model reduces the effective total coefficient range for methods which rely on knowledge. Table 5.6 shows the linked values and the effective relative range of the total quality coefficient. Further research could redefine the aspects in such a way that this and similar constraints are reduced or entirely removed from the mechanism.

5.12 Applying the mechanism on What You Enter Is What You Sign

What You Enter Is What You Sign (WYenterIWYS) as a transaction information scheme and method was proposed in the chapters of Part II. For the expansion of Renaud’s mechanism, we did not examine What You Enter Is What You Sign when we wrote the paper on which this chapter is based. Instead, only implemented and proposed authentication methods by others were examined to avoid the notion of author bias in our published paper. In this section, WYenterIWYS is examined using Renaud’s mechanism and our expansion to determine how it compares against the results of other authentication methods that can be found in Table 5.5 on page 131. Note that this evaluation is not performed by multiple parties and therefore it should not be seen as an extension to the original paper on which this chapter is based. Instead, it should be seen as an extension of the thesis.

Table 5.7 shows the quantification of qualitative characteristics of WYenterIWYS as based on the proposal of Part II, Chapter 4 using the evaluation mechanism described in this chapter. The data in Table 5.7 can be compared with the data of the methods that were evaluated by multiple raters shown in Table 5.5 on page 131. What follows is a brief description of the authentication method, a description of how the values were assigned, and a discussion of the strong and weak points of WYenterIWYS.

As with some proposed methods discussed earlier in this chapter, the proposal for WYenterIWYS does not describe entity authentication. For WYenterIWYS, the

Table 5.7: What You Enter Is What You Sign quantified.

Dimension(s)		WYenterIWYS manual code
↓	Description	
Accessibility	Special req.	0.33
	Convenience	0.50
	Inclusivity	0.67
	Deficiency	0.90
	Percentage	48%
Memorability	Retrieval str.	1
	Meaningfulness	0.33
	Depth of proc.	1
	Deficiency	1.45
	Percentage	16%
Security	Predictability	1
	Abundance	0
	Disclosure	0.5
	Deficiency	1.12
	Percentage	35%
Vulnerability	Confidentiality	1
	Privacy	0
	Breakability	0.33
	Deficiency	1.05
	Percentage	39%
1 - 4	Total quality coeff.	2.41
	Overall percentage	34.8%
Feasibility	Work flow expansion	0
	Circumvention	0.33
	Clarity	0.67
	Deficiency	0.75
	Percentage	57%
1 - 5	Total quality coeff.	3.39
	Overall percentage	39.1%
Difference between dimensions 1-4 and 1-5		4.4%

same assumption is made for bank-issued devices as described in Section 5.8.1. Before any functionality of the authentication device can be used, the user is expected to enter a PIN. The initial PIN is randomly chosen by the bank and can be changed by the user. One-time passwords are used for user authentication (the initial login to establish a user session). WYenterIWYS is used to authorize transactions. We propose the use of the implementation as described in Part II, Chapter 4: the user enters critical transaction information in a separate authentication device, which generates a code containing this critical transaction information and a digital signature. The user enters the code in the computer used for online banking, after which the bank receives the information securely.

For the accessibility dimension, the only special requirement is hardware since for the implementation a separate, bank-issued device will be used. It is not required to install software, nor is technical knowledge required to use the authentication device. The only action asked from the user is the ability to read and enter data, which relies on skills that users require to perform online banking with or without security. The hardware requirement assigns a value of 0.33 to the special requirements aspect.

The convenience aspect is only weighed down by initial enrollment and replacement of the user's secret keys, which are physically located in the authentication device. The time to authenticate is quite short due to that the user is not required to compare values, warranting a value of 0.50.

We assume that persons with cognitive or sensory disabilities will be unable to use the method. Users with cognitive disabilities could have problems with the requirement to remember a PIN, while sensory disabilities can hinder human-computer interaction when a device is not tailored to the specific needs of the user. Mobility disabilities should not be a problem in the assumption that a user is capable of using common human-computer interfaces (keyboards, mice, touchscreens, etc.). The exclusion of people with cognitive and sensory disabilities gives the inclusivity aspect a value of 0.67.

The values in the memorability dimension are based on that an implementation will rely on a PIN that is initially distributed randomly and which can be changed by the user. Because of this, the PIN is hard to retrieve, it is possibly meaningful to the user, and the user requires full depth of processing to remember it, warranting for the retrieval strategy, meaningfulness and depth of processing aspects the respective values of 1, 0.33 and 1.

For the security dimension, the predictability aspect gets a 1 since users are able to pick their own PIN, which could make it predictable. The number of possible keys used for cryptography can be very high, which is why abundance stays at 0. It is not possible to steal all authentication credentials easily, since that would require the physical theft of one's authentication device after observing the entry of the correct PIN. Since the entry of the PIN can be observed, the disclosure aspect is rated at 0.5.

In the vulnerability dimension, confidentiality is fully taxed with a 1 since the user has to supply the full knowledge authentication factor (PIN) on each attempt. The user is not required to use any personal information in the authentication process, leaving privacy unaffected at 0. A research-based attack could circumvent the security of the system if an adversary learns where the personalized authentication device (the possession factor) is stored and what the chosen PIN is (the knowledge

factor). The device is not vulnerable to brute force or dictionary attacks due to a lockout after three failed attempts to enter the PIN, and the absence of the ability for the user to install software or attach additional hardware prevents the installation of a keylogger. Since the device is only vulnerable to research-based attacks, the breakability aspect is assigned the value of 0.33.

Finally, in the new feasibility dimension the work flow expansion aspect is fully favorable since the user is not required to perform any new actions to perform transaction authorization. Some existing actions do change (for instance, the user is required to enter critical transaction information on a separate device), but there are no new actions introduced that cognitively challenges the user. The user is not required to compare numbers or to perform actions redundantly, which is why the work flow expansion aspect receives the most favorable value. When the system is not used it is secure as long as the user does not write down his or her PIN anywhere, since the authentication device would lock its functionality once three incorrect PIN attempts are made. This makes it unlikely that an adversary can use the system if the user does not actually use it. Also, the user cannot circumvent the security of the system on purpose by skipping steps in the authentication process. Unfortunately, to authorize transactions with WYenterIWYS there is a reliance on information that is returned to the bank on the user's computer, even if it is only for a message that informs the user that a transaction has succeeded. Therefore, the user interface does not necessarily support secure user behavior since part of the user interface is the user's untrusted computer (to which the bank sends status information).

Clarity should be penalized at two points. First, the information channel from the bank to the user is corruptible, since it ends at the user's computer. Any responses from the bank can be changed by an adversary. For example, when a legitimate transaction is completed the 'Transaction complete' message could be changed to 'Transaction failed, please try again'. If the user would attempt to authorize he same transaction again, a second illegitimate transaction (with the same critical values as the first legitimate transaction) would be made and approved. This attack vector taxes the clarity aspect for two-third, since in it information the user requires to make a correct action is missing and due to the corruptible bank to user information channel. The only redeeming quality is that the interface of the device itself could be designed in a way that it could give the user information to ascertain its abilities and limits.

Before Table 5.7 is analyzed, it is again important to note that the evaluation mechanism has some subjective aspects (as was discussed in Section 5.10.4). In addition, the evaluation for WYenterIWYS was done by a single rater. Therefore, the analysis of Table 5.7 should only be done to provide an estimate on how a future alternative authentication device could perform based on only the discussed characteristics.

The data in Table 5.7 can be compared with the data from Table 5.5. As shown, the overall percentage of dimensions 1 to 5 of the WYenterIWYS evaluation is a bit higher compared to the highest rated method in Table 5.5, Bank Scan VNA. This high rating is mostly owed due to the fairly high scores in the original four dimensions, and due to the high score in the feasibility dimension. For the latter, the work flow expansion aspect was set to 0 since it is believed that, unlike all other methods, users do not have to perform any additional redundant or new actions to

perform transaction authorization using WYenterIWYS, which makes this score so high.

Due to the subjectivity involved in applying the mechanism it cannot be said that WYenterIWYS is better than everything else. However, its high qualitative fit gives a strong indication that it is an alternative transaction authorization method with a lot of potential, which should be explored further. This is an additional point for further research on top of those already mentioned in Chapter 5.11.

5.13 Concluding remarks

We expanded Renaud's quantifying mechanism to accommodate aspects related to transaction authentication in online banking in a user-centric context. Several used and proposed transaction authentication methods for online banking were evaluated using the original four dimensions and our expansion by seven raters. The inclusion of an additional dimension changed the ranks of 5 out of the 12 evaluated authentication methods.

There is a large amount of subjectivity involved when applying Renaud's mechanism and our expansion. In a bit less than a third of the asked questions did the (independent) raters come to an unanimous answer. This does not make the mechanism worthless, but it is advised that evaluations are performed by multiple raters, since it would be unwise to consider the opinion of a single expert as the truth.

The methods which have a good overall fit in both the original and the expanded mechanism include Bank Scan VNA, Weigold Scan VNA and Weigold USB VNA, closely followed by Weigold Entry ENA. The first three concern an implemented and three proposed authentication methods which use a Customer Verified Transaction Set Authentication information scheme, while the fourth uses Entered Single Transaction Authentication. This suggests that either information scheme can be applied to design an authentication method which can satisfy many aspects.

Trusted bank devices have a very good overall fit within the dimensions of the mechanism. User-owned mobile devices have a worse fit for online banking authentication purposes, except for the implemented Bank SMS VA. That this authentication method ranks so high is likely due to personal bias among the raters who actually use this method in daily life, considering that the proposed Weigold SMS VNA is mostly the same but ranks much lower. When this outlier is ignored, it can be said that authentication methods which rely on user-owned devices tend to have an overall worse fit compared to those which rely on bank-issued devices.

The mechanism was also separately applied to What You Enter Is What You Sign, a transaction authorization method proposed by the authors, to give an indication about its qualitative fit within the evaluation mechanism. Resulting data indicates that this alternative method has much potential. Although not applied yet in online banking in practice, it should be considered due to its good fit within the user's workflow.

Chapter 6

Practical evaluation to measure secure usability

Abstract

Security and usability improvements in online banking are often made in academic proposals. Testing of these proposals could provide vital information for designing new systems and for proposing further improvements. A modular evaluation framework, presented as a virtual bank, could provide a common ground for testing and reduces the overhead of setting up experiments. We propose such a framework for testing secure usability in online banking, since it does not exist to our knowledge. To validate that the envisioned framework would provide useful information, we created a first proof of concept to measure secure usability user behavior with two different authentication methods in an experiment. The results confirm that online bank users pay more attention to security actions after they noticed an attack. We were also able to conclude that of two tested authentication methods, one is significantly faster in use compared to the other. These results validate that the envisioned framework will be able to provide useful information. What we learned from the proof of concept will be used in the development of a modular evaluation framework, which we will release in the near future as open source for others to experiment with.

6.1 Introduction

Traditionally, banks existed to keep their customers' valuables safe against theft. A customer would visit the bank, give some proof of identification to a person behind the counter and either make a deposit or withdrawal to an owned account, or give the order for a money transfer to someone else's. It was clear who was responsible for keeping the customer's money safe. Today, customers use online banking through their own home and mobile computers. Security now relies on behavior of both bank and users to prevent successful attacks such as malware and phishing. To support this behavior, it is vital that a bank's system is designed to support secure usability.

In Section 6.2, we describe why there is a need for an evaluation framework, represented as a virtual bank, to give researchers the opportunity to easily examine and compare existing and proposed authentication methods. We are currently developing a modular evaluation framework, of which an overview and our goals are noted in Section 6.3. The contributions of this chapter are (1) a proposal for an evaluation framework of online banking security aspects on the user behavior level, (2) a first proof of concept to examine whether the envisioned framework will be able to collect the necessary data, and (3) a small experiment using the proof of concept.

The main research contribution of this chapter is the retrieval and analysis of data from a proof of concept. The data was gained from 20 test participants who used two different authentication methods, of which one was also attacked, to measure participant behavior. A description of the proof of concept is given in Section 6.4, and more information about the tests themselves and the analysis of the resulting data is given in Section 6.5. Finally, our view on the validation of the framework and on the resulting data together with several opportunities for further research are given in Section 6.6, before we close with our concluding remarks in Section 6.7.

6.2 The need for a new evaluation framework for online banking security

Online banking relies on both usability and security. Usability concerns that users can use it, while security is about preventing adversaries from doing the same thing illegitimately. These two characteristics have the potential to conflict with each other. Rigorous security can make it almost impossible for an adversary to perform a successful attack, but users might not want to use the system anymore because of inconveniences the high level of security brings. On the other hand, nobody is waiting for a system that is very user friendly to anyone, including parties that should not be able to use it.

There are also designs in which user actions are expected to add security to a system. In these cases, incorrect user actions might compromise security. What makes this complicated is that users often choose the path of least resistance [Yee02], and that they are unmotivated to give security their full attention [Wes08]. This is why the concept of secure usability is important. Security and usability should not be seen as two separate goals. Instead, systems destined for user interaction should be designed in such a way that motivates to use them securely.

Tests for secure usability in a safe environment are not uncommon. Pilots are trained to affirm that they are up to the task of controlling an airplane and following the procedures in case of calamities.¹¹⁴ Medical personnel can safely practice procedures before they ever risk human life.¹¹⁵ These examples use simulators which can be used for a plethora of different scenarios. Training is already mentioned, but they also have the potency to improve real world systems through safe experiments without dire consequences if something goes wrong.

For personal banking, researchers sometimes create their own tools to evaluate usability and security aspects, such as for a study on the effect of menu structures in ATM machines [TPC10]. Examples of online bank simulators also exist, such as for examining the number of typographical errors made in destination account numbers [Ols08] and the mistakes users make when comparing such numbers on equality [AAJM08].

Simulators created for pilots and medical personnel can be used in many different scenarios, but the banking examples have in common that the created tools are only developed for specific tests that align with the authors' own research. Especially for online banking and the large variation in authentication systems [KSDC⁺14], a simulator would enable researchers to test existing scenarios and proposed changes. There are many proposals for improvements in online banking authentication and transaction authorization [SFG09, HPN10, AAJ10, WH11, LSH⁺12, KVvE14]. Most of these proposals have not been tested on their usability, or results from such tests are not public. A virtual bank with a modular design would allow comparisons between methods and experiments to improve methods without re-inventing the wheel every time a new research question is formed.

We did not find such a tool in our search. Virtual bank environments might be used by banks to perform tests based on what they (consider to) offer but results are not publicly shared. We decided to start the development of an evaluation framework to support a virtual bank environment which can be used by us and other researchers to test security and usability concepts in online banking.

6.3 Design of a virtual bank for secure usability research

We are developing an evaluation framework to test existing and proposed security and usability concepts in online banking. Our framework measures behavior by recording user interaction with a virtual bank's graphical user interface through standard input devices. The effects of differences in user interfaces or instructions can be registered to audit general usability, but also the secure usability of security mechanisms (such as authentication methods). The idea is that test participants are given an account at an online virtual bank, which they can visit to make transactions (money transfers).

Our major goals are:

¹¹⁴Jonathan Gabbai - The art of flight simulation: <http://gabbai.com/academic/the-art-of-flight-simulation>

¹¹⁵Center for Medical Simulation about High Realism Usability Testing: <https://harvardmedsim.org/usability-testing.php>

- A modular design. This allows authors of new proposals to create proof of concepts and other researchers to compare existing concepts, without developing an entire supporting environment. For example, modules can concern participant login and the authorization of transactions, also sometimes referred to as entity and transaction authentication [CDDC⁺02].
- Integration with survey software. The evaluation framework will allow the registration of participant behavior, while surveys (given before, during, after or separate from an experiment) can record participant perception. For example, this can be used to measure the amount of trust a user has as a whole or for specific parts of a tested system, and how that trust changes over time.

We want to create an environment in which:

- Participants can login and create transactions.
- The exact input actions of participants are registered. This data can be used to measure user behavior.
- Simulated attacks can be conducted to measure whether participants use the system in a secure manner.
- The number of participants is not arbitrarily limited.
- The location of participants is not physically limited.
- Participants can participate over a period of time.

The last three points serve two purposes. Firstly, taking away physical limitations ensures that more participants can join. There are countries in which more than 80% of individuals aged 16 to 74 use the internet for internet banking.¹¹⁶ By offering a virtual bank through an online web environment, the only technical requirements for individuals to become participants are an Internet connection and a suitable device to participate with. These are the same requirements for online banking. Because of this, a population sample does not have to be limited to a specific geographical area or to the number of seats in a local test center.

Secondly, it allows participation in similar conditions as online banking. This includes the time and place where online banking is conducted, which might change over time. Providing the same kind of service as online banks makes resulting data represent the real world as much as possible.

Figure 6.1 illustrates the design and structure of the framework, based on three levels. The front-end consists of web pages which a participant navigates through (log in, the main page, pages related to creating new financial transactions, and survey pages). Between the web-based frontend and the data backend is a layer of different module types, which are abstractly represented in Figure 6.1 to leave room for new module types. Participants will usually interact with the system through the web-based frontend. However, modules can be used to interact with the user through alternative communication channels if the browser-based channel is inadequate.

¹¹⁶Eurostat on internet banking use: <http://ec.europa.eu/eurostat/tgm/table.do?tab=table&plugin=1&language=en&pcode=tin00099>

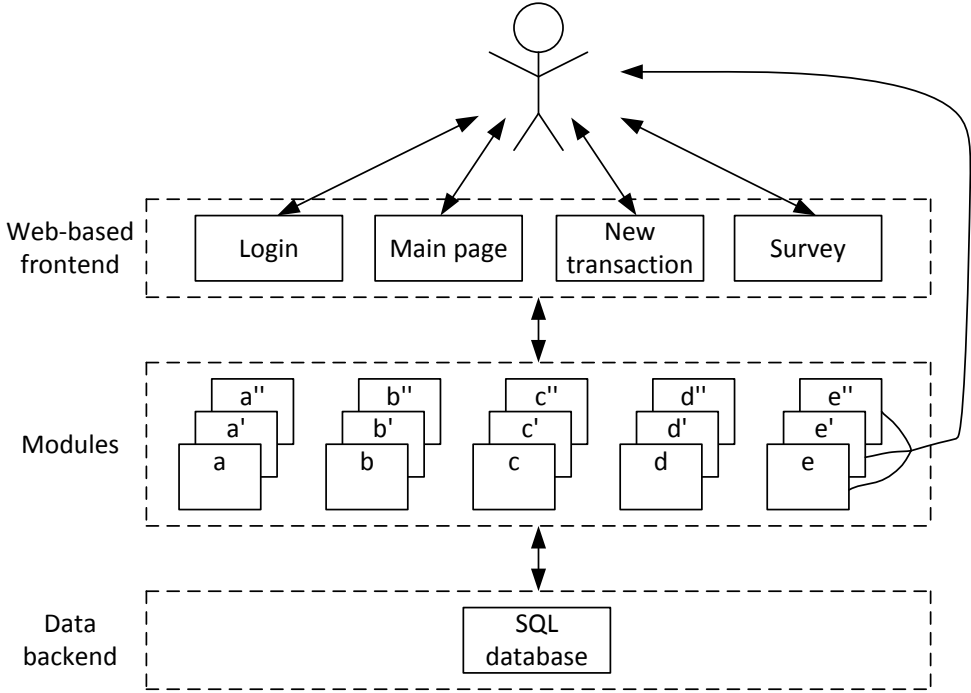


Figure 6.1: An overview of the envisioned framework’s structure.

We give a brief description of the module types that we are working on. Initial login and authorization of financial transactions by the participant are supported by one or more authentication modules. These currently consist of a password module which allows users to login, and two modules related to transaction authorization for which more information is given in Section 6.4. Attack modules can influence user behavior by simulating attacks. Examples include malware attacks which silently create or change transactions [Sch05], phishing attacks with which users are tricked to give valuable information on fraudulent websites that look and act like the websites they expect to visit [DTH06], and attacks which combine malware and phishing [FFC⁺11].¹¹⁷ External communication modules can be used to exchange information with the user through alternative channels, such as email and SMS. Modules can interact with each other. For example, an authentication module would be able to use an external communication module to send a one-time password, which the participant has to enter when authorizing transactions.

Finally, the database backend is used by the framework to persistently store and retrieve information, and can be used by other software to extract collected information for analysis.

We developed a proof of concept based on our work in progress. The goal of the proof of concept is to examine whether we can collect useful data from actions that users perform when creating transactions using different authentication methods,

¹¹⁷An example of such an attack targeting a Dutch bank (2015-06-12): <https://www.dearbytes.com/blog/phishing-via-mobiele-malware/>

in an environment conforming to the earlier noted six points. Since the framework is still a work in progress and not yet suitable to make those measurements, we reused some of the existing code to make a standalone, non-modular environment to conduct two experiments. We used code from modules that have already been developed. The structure of the proof of concept is shown in Figure 6.2, including the interaction between different components.

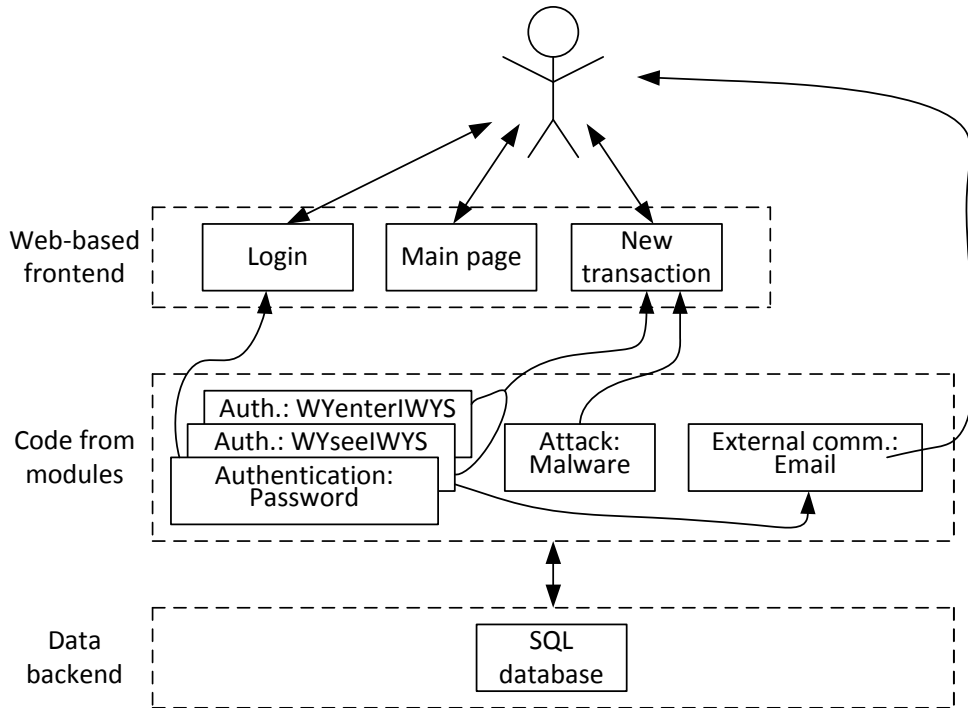


Figure 6.2: An overview of the proof of concept's structure.

More information about the proof of concept is given in Section 6.4. The transaction authentication methods What You See Is What You Sign (WYseeIWYS) and What You Enter Is What You Sign (WYenterIWYS) are also described in this section. More information about the experiments in which the transaction authentication methods are used and the results can be found in Section 6.5.

6.4 A virtual bank for secure usability research: setup of experiments using a proof of concept

It is considered a good practice to create a proof of concept to give an indication of the feasibility and to steer further development. Based on what we wanted from the environment of an evaluation framework as mentioned in Section 6.3, we developed a proof of concept, which is based on a work in progress version of our framework, but distinct in several ways. The proof of concept differs in that it is only meant to

examine the possibility of collecting relevant data. Unlike the framework we have yet to complete, it is not modular, nor does it allow the recording of user experience through surveys.

The primary objective is to retrieve data from a group of participants in a way which is not constrained by physical space or time such as traditional test environments are. Analysis of the resulting data (see Section 6.5) is not specifically aimed to criticize or reaffirm results from other researchers. Rather, the analysis is used to test whether the data that we retrieved from the proof of concept makes sense. In other words, it answers the question whether we can measure that what we want to be measured. It is not needed for the proof of concept to be modular or to interface with survey software to fulfill its goal.

The proof of concept consists of a website. After login (using a username and password), each participant can make transactions. The life cycle of a transaction is depicted in Figure 6.3, and a description of each step follows.

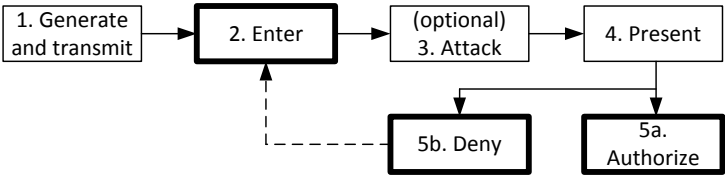


Figure 6.3: The actions within a transaction life cycle. Thicker lines indicate actions by the participant.

1. The server generates a request for a transaction (consisting of an account number and an amount of money) and sends it to the participant using email. This request will be used by the participant in subsequent steps to create a transaction in the bank’s web interface.

User information

Username: skiljan
 Name: Sven Kiljan
 Account number: NL64 SIMB 0933 0778 07

Transaction history

Date/time	Destination account number	Amount
2015-10-04 06:02	NL05SIMB0349204926	€ 50.00
2015-10-04 06:06	NL64SIMB0972789316	€ 40.00

Log out

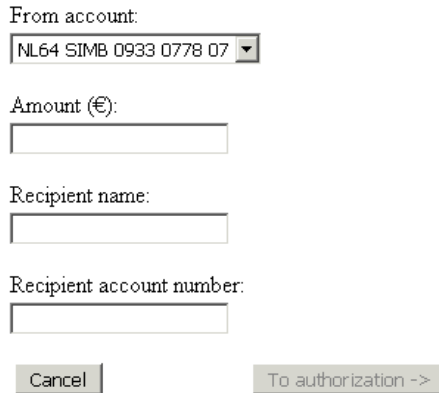
New transaction ->

Figure 6.4: Main menu of the virtual bank. The transaction history shows two previously authorized transactions.

2. The participant visits the bank website and is presented with the main menu as shown in Figure 6.4. He chooses to create a new transaction and is redirected to the transaction entry form (depicted in Figure 6.5). There, the participant

enters an amount of money to be transferred, a name of the recipient and a destination account (IBAN) number. Earlier in step 1, the participant received the relevant amount and account number in an email. Participants are free to choose a name of their liking. Clipboard (cut/copy/paste) functionality is disabled so that users have to enter the data provided in step 1 manually. This allows the registration of typographical mistakes which could occur in real world scenarios (for example, a bank user needs to enter data from a physical form digitally).

Note: you cannot use cut, copy or paste text functions.



The form consists of the following elements:

- A label "From account:" followed by a dropdown menu showing "NL64 SIMB 0933 0778 07".
- A label "Amount (€):" followed by an empty text input field.
- A label "Recipient name:" followed by an empty text input field.
- A label "Recipient account number:" followed by an empty text input field.
- At the bottom, two buttons: "Cancel" on the left and "To authorization ->" on the right.

Figure 6.5: Transaction entry screen. The 'From account' menu had no purpose other than making users feel that they were transferring money from their account (of which the account number is as shown in the main menu).

3. Optionally, a simulated attack is conducted on the transaction. Certain data that the user entered is modified after the user has entered the data and confirmed the entry.
4. The information entered in step 2 is presented to the participant for authorization through one or more information channels. Modifications from step 3 are either visible or hidden, depending on which channel is used.
5. The participant authorizes or denies the transaction in the authorization form (shown in Figure 6.6). Required actions to authorize a transaction differ between authentication methods. If a transaction is denied, the participant can optionally try it again, starting at step 2.

We performed two experiments using two transaction authentication methods. In the first experiment, the behavior of participants when confronted with silent malware attacks is registered to show if and when participants pay attention when comparing transaction details. This is solely done with the What You See Is What You Sign method.

The second experiment aims at making conclusions about the differences and similarities between the participants which use two different authentication methods.

Prepared transaction

From account: NL64 SIMB 0933 0778 07
Amount: € 12.34
Recipient name: Recipient
Recipient account: NL23SIMB0111882559

An email with information about the transaction and an authorization code has been sent to your email address.
Only enter the authorization code in the field below if the information on this page matches the information in the email.

Authorization code:

Figure 6.6: Authorization screen of WYseeIWYS. An authorization code (received through email) is required to authorize the transaction.

This is done by examining the time required for participants to perform certain actions. What You See Is What You Sign is one method used by one group, while a control group does not have to perform any action when authorizing transactions. Effectively, participants in the control group would perform the same actions as when What You Enter Is What You Sign would be implemented, with the difference that data is not actually signed when entered by our participants. Results from the data of the second experiment indicate whether there is a significant difference in processing time between both authentication methods. For data entry it is expected that the processing times would be the same for both groups, while for transaction authorization it is expected that the control group would require less time.

More information about the authentication methods follows.

6.4.1 What You See Is What You Sign

When a bank receives a transaction request, critical information from the request is relayed back to the user over a secure channel. This allows the user to verify whether the information received by the bank in an earlier stage is correct. By securely ‘signing’ and sending the signature back to the bank, the user indicates that the data is correct and that the bank should proceed with fulfilling the transaction.

We relate this method and how we implemented it to Figure 6.3. The data of a transaction is entered in step 2. In step 3, there is a 50% chance that an attack will occur. If an attack occurs, a single digit of the destination account number is changed. This simulates a malware attack in which transactions are silently changed on the user’s computer, and is similar to the ‘Stealthy Attack’ described and performed by AlZomai et al. (2008) [AAJM08]. A difference with their attack is that ours is performed on an IBAN account number, which is structured differently from the plain numbers that they used. The attack is not realistic since an adversary would need access to a bank account with a number that is very similar to the victim’s, and IBANs have a validation mechanism that can be used to detect

entry errors. As noted earlier, the objective of the proof of concept is to see whether we can measure the user's behavior. By using the same attack as AlZomai et al. (2008) used, we can compare our results to theirs.

For step 4, the participant is confronted with transaction data through two different channels. The web browser is the insecure channel. For the experiment, email messages are used as a simulated secure channel. Email is not secure since (by itself) it does not provide confidentiality, integrity or authenticity. However, for our experiment it is the most economical and user-friendly way to simulate the secure channel since all participants have an email address.

The web browser shows transaction data from step 2. An email will contain transaction data from step 3 (if an attack occurred) or from step 2 (in the absence of an attack). The email also contains an authorization code. It is the participant's task to compare data (account number and amount) shown in the web browser and received by email. If the data sets are equal, no attack occurred and the transaction should be authorized by entering the authorization code in the web interface. If data differs, the transaction is attacked and should be denied. The simulated attack is similar to the possible effects of a man-in-the-browser attack, where an adversary controls what is shown in a web browser and which data the bank receives [CD12].

6.4.2 What You Enter Is What You Sign

Our control group does not have to perform data comparison in step 4, because they are never attacked in step 3. Instead, candidates simply have to indicate whether they accept or reject the transaction. This effectively implements What You Enter Is What You Sign (WYenterIWYS) [KVvE14] from a user's perspective, which concerns a proposal that offers an alternative transaction authorization method. WYenterIWYS relies only on a secure information channel from user to bank. The integrity of the user's input is secured from the moment it is entered.

This is in contrast to the earlier mentioned WYseeIWYS method, which relies on an insecure information channel from user to bank to transmit user input, and on a secure channel from bank to user to send critical information back for verification. The difference between the two methods is that with WYseeIWYS the user is expected to verify transaction information, which is not necessary with WYenterIWYS.

An actual implementation of WYenterIWYS would require some kind of trusted environment which cannot be influenced by third-parties, which could be given shape by a separate hardware device or by an environment inside a user's computer system that is separated from outside influences. It was not feasible for our experiment to design and distribute hardware, and there is no secure environment available to us on consumer hardware used by participants.

6.5 Validation of the use of the proof of concept virtual bank

Our test took six days. We asked 26 Integral Safety students to participate in our test. The request to participate was sent over email and was the same for all students. One authentication method was assigned to each participant, and the participants

were equally distributed over the two authentication methods. Each participant received two additional messages, specifically send to the participant's mail address and greeted by first name in the message. The first message contained account information and the web address to visit the virtual bank. The second message contained three transaction requests as part of step 1 of the transaction life cycle (see Figure 6.3) in addition to instructions on how to conduct the transactions based on the assigned authentication method. For WYseeIWYS, participants were instructed to carefully compare what their web browser showed and what they received through email (steps 4 and 5 of the transaction life cycle). It was stated that if there is a difference, that users should deny the transaction for safety concerns. We did not state explicitly that an attack could occur (step 3 of the transaction life cycle).

After three days, three additional transaction requests for each participant were sent in personalized email messages.

20 of the 26 participants conducted transactions in six days (10 for each authentication method). We registered data based on attempts made by the participants to authorize (complete) a transaction. An attempt is defined as one cycle of actions between steps 2 and 5 in Figure 6.3. One transaction can have multiple attempts if it is denied in earlier attempts.

Method →	WYseeIWYS	WYenterIWYS
Participants		
Asked for participation	13	13
Participated	10	10
Transactions		
Prepared	78	78
Executed	49	57
Attempts		
Total attempts to authorize transactions	68	57
Not attacked and authorized	35	57
Attacked and authorized	10	N/A
Attacked and denied	23	N/A
Not attacked and denied	0	0

Table 6.1: An overview of the transactions and attempts.

Table 6.1 shows summarized data of the participants, transactions and attempts to complete them. As explained in Section 6.4, attacks were only conducted during attempts to complete transactions by participants who were assigned the WYseeIWYS authentication method. This is why the values related to attacks are not available for WYenterIWYS.

The first experiment was noted in Section 6.4. In it, the behavior of participants is registered when confronted with silent malware attacks. We want to examine if and when participants pay attention when comparing transaction details with WYseeIWYS-based authentication methods. For the WYseeIWYS authentication method, 33 attempts to complete a transaction were attacked. 23 of these attacks failed (the participant denied the transaction) and 10 or 30.3% succeeded (the participant authorized the transaction while the destination account number was modified due to the attack). These results are bit more positive compared to those of AlZomai et al. (2008), who registered in their experiment that 61% of such attacks succeeded.

6 participants were successfully attacked in 10 attempts. The other 4 participants

were each attacked at least once and denied the transaction each time an attack occurred.

We assume that the participants paid careful attention and spotted the changed digit correctly in the 23 attempts in which an attack was successfully deflected. This assumption is supported by the number of attempts that were not attacked and denied, which is 0. Therefore, it is unlikely that participants accidentally denied transactions when they were attacked.

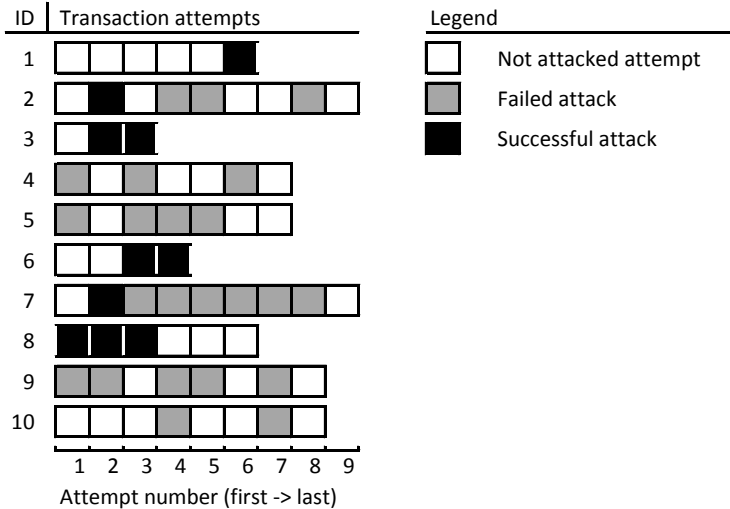


Figure 6.7: The order of WYseeIWYS participants’ attempts, whether they were attacked and if an attack was successful.

Figure 6.7 shows an overview of the 10 attacked WYseeIWYS participants and the order in which their attempts occurred. Of the 6 who were successfully attacked (having black blocks somewhere in the attempts bar), 5 were not attacked on the first attempt. Of the 4 participants who recognized all attacks, 3 were confronted with an attack on their first attempts. How participants reacted to attacks suggests that WYseeIWYS users are more careful the first time they apply the authentication method, and that they pay more attention once they recognize an attack. AlZomai et al. 2008 [AAJM08] noted that their participants tended to avoid attacks more often once the user has had some experience with the simulated online banking interface. This is not exactly supported by our data. It seems that it is not experience with the online bank itself that strengthens participants’ vigilance in later attempts. Instead, it seems to come from experience with previously noticed attacks.

Of course, our research data is limited. AlZomai’s experiment had an average of 7.4 fully processed transactions per participant (for 92 participants), while ours had an average of 4.9 (for 10 participants). This difference in scale could explain the different conclusions. It could be that participants who repeatedly recognized attacks after an earlier recognized attack still are successfully attacked in later attempts.

When examining the time required to perform step 5, we can separate these attempts in two categories: those where the participant did not pay attention and those where the participant could be paying attention but simply missed the changed

number. In 23 attempts an attack was recognized by the user and the transaction denied. The minimum time for recognizing the discrepancy between the entered account number and the account number in the verification message was 51 seconds. In the 10 attempts where the attack was not recognized, there are two attempts in which the authorization time is drastically low (14 and 22 seconds). All other attempts start at 46 seconds and increase from there. It is implied that the participants in the two attempts with the least amount of time simply did not pay attention, whereas the other participants likely failed to see the changed digit.

As noted in Section 6.4, the second experiment aims at making conclusions about the differences and similarities between the participants which use two different authentication methods. We could not attack WYenterIWYS in the same way as WYseeIWYS, simply because participants were not asked to verify transactions. The assumption was made that a WYenterIWYS participant's entries would be secure as soon as they were entered. However, step 2 was the same for both authentication methods. A comparison of the data in this step can indicate whether the behavior of both groups differs. Figure 6.8 shows a histogram of how much time the entry for each first attempt of a transaction took. For WYseeIWYS we only used the entry of the first attempt for each transaction, and we removed a single outlier (of 825 seconds) from an initial entry using the same authentication method.

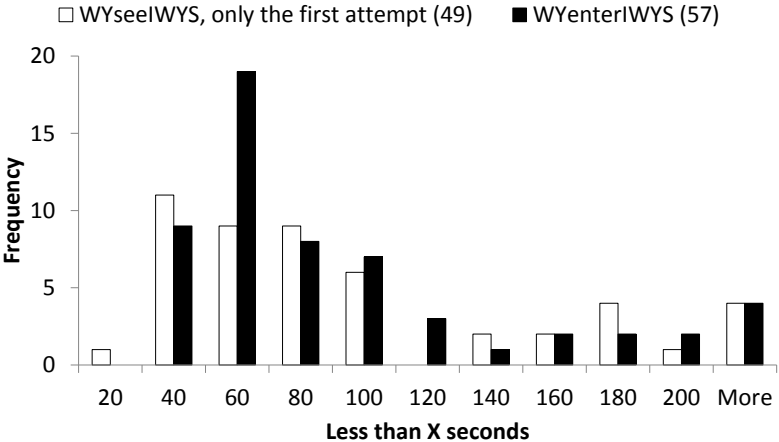


Figure 6.8: Distribution of entry times in non-attacked WYseeIWYS and all WYenterIWYS attempts.

The participants were not asked or forced to perform each attempt within a specific time frame. As with real online banking, participants could be multi-tasking or simply walk away from their computers. An extreme case was the earlier mentioned and excluded outlier, where a participant required more than 13 minutes.

Table 6.2 gives an overview of several t-tests we performed to determine whether the mean values between the processing time for specific aspects of WYseeIWYS and WYenterIWYS attempts are equal. As shown for the overall time used for the initial entry of all values, the standard deviations (s) for each set of entry times are almost the same and the sample sizes (n) are large enough to make the t distribution approach a normal distribution. Therefore, we used an independent two-sample t-

test to determine the probability that the means are different. We performed the t-test with equal variance since the standard deviations are quite close, and two-sided since we test whether the mean values are different in any way. The resulting probability value p is nowhere near enough to state that they would be. This implies that both groups of participants are homologous in this regard.

		WYseeIWYS	WYenterIWYS	T-test	p
Step 2 - Entry (overall)	n	49	57	2-sided	0.7098
	s	67.2	64.3	Eq. var.	
Step 2 - Entry (amount)	n	47	57	2-sided	0.3535
	s	35.9	29.4	Eq. var.	
Step 2 - Entry (account nr.)	n	47	57	2-sided	0.9589
	s	19,3	22,6	Eq. var.	
Step 5 - Authorization (not attacked only)	n	35	57	1-sided	<0.0001
	s	91,1	11,4	Uneq. v.	

Table 6.2: Probability values for claims about the mean values of transaction entry times (in seconds).

This conclusion is supported when we zoom further into entry times of specific entered values. Three data values had to be entered: amount, name of the recipient (which the participant was free to choose) and destination account number. The amount and destination account number are also included in Table 6.2. Their probability values are far too high to indicate a significant difference between the mean times. Note that for WYseeIWYS n is lowered by two since we were unable to measure entry delays for individual fields during two attempts. These missing values would barely influence the outcome, based on the similar s and the earlier overall analysis of all entries.

Where the entry stage (step 2) was the same for both WYseeIWYS and WYenterIWYS participants, the authorization stage (step 5) was different due to the use of different authentication methods. It would be expected that the WYenterIWYS group would perform this step quicker compared to the WYseeIWYS group since the former does not have to perform any checks, while the latter has to check whether the transaction data was correctly received by the bank. This is also suggested by the histogram in Figure 6.9.

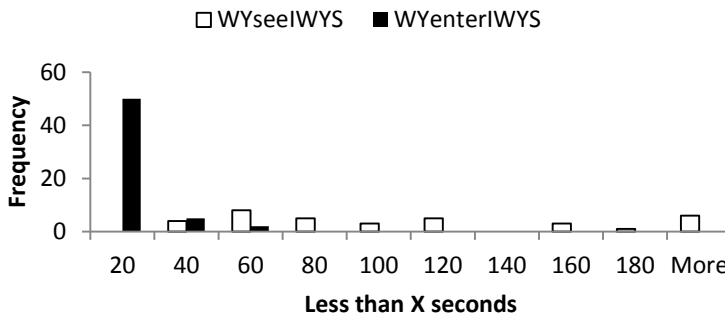


Figure 6.9: Distribution of authorization times for non-attacked WYseeIWYS and all WYenterIWYS attempts.

The expectation that WYenterIWYS authorizations require less time than WYseeI-

WYS authorizations is why we opted to perform the t-test one-sided. Also, as Table 6.2 shows, the standard deviations differ quite a bit. We applied Welch's t-test since it is more reliable when variances are unequal [Rux06].

The result is as expected. p indicates a very significant probability that WYseeI-WYS authorization times are larger.

6.6 Discussion and further research

This section first notes our thoughts about the experiment with the proof of concept. After that, it will continue with what we want to research in the future based on the retrieved data.

6.6.1 Evaluation of the experiment and lessons learned

We learned a lot from the experiment with the proof of concept. As shown in the previous section, different aspects of participant behavior were successfully registered by monitoring the exact actions participants took. The conclusions we made were obvious, but the goal of the experiment was not to reject a hypothesis. Instead, our goal was to see whether we could collect and analyze data to warrant the validity of the framework we are developing. The conclusions warrant this.

The proof of concept also showed several limitations of our current approach. Participants were not urged to perform the experiments as quickly as reasonably possible, which can add high processing times to the results when participants do something else during the experiment. This makes analysis of the data more difficult. To counter this, a (visible) time limit might be added when a participant initiates a transaction. Alternatively, only asking participants that they perform the experiments in a timely manner might be enough.

Another limitation was in how we register the input of individual fields in the transaction entry screen, which is mostly done server-side and a single client-side event. In the proof of concept, we send values as soon as a field loses focus. This is not optimal for various reasons. Sometimes field focus is lost if a participant opens a different application or browser tab (for example, to read an email message related to the experiment), which registered as a complete input. Also, it requires participants to explicitly select another element after the last value has been entered for data validation. To improve this, more client-side intelligence could be used to determine when a user is truly done with entering input values, such as whenever another web element is explicitly selected by the user instead of the currently selected element losing focus.

6.6.2 Building forth on the resulting data

Our data in Section 6.5 indicates that WYseeIWYS users seem to be more careful the first time they apply the authentication method, and that they pay more attention once they recognize an attack. In further research, we would like to do a more thorough experiment in which we want to measure how long the period of higher awareness lasts. With this information, it can be determined when online banking users need a mental reminder to keep their awareness high. Such a reminder could for

example be implemented by banks as a short online training course, in which users are confronted with what an attack could look like. Aside from possibly improving security, it could also affect trust in online banking when users recognize the risks and learn to defend themselves.

As for authorization times, WYenterIWYS uses less time compared to WYseeIWYS, also as indicated in Section 6.5. For WYenterIWYS we only tested the most ideal situation by assuming that the user's input was secure. There is the research opportunity for testing the usability of an actual implementation. Since this will probably introduce a new element to the work flow, not only should the participant's behavior be measured but also the participant's perception and experience.

6.7 Concluding remarks

We proposed an evaluation framework that implements a virtual bank to measure user behavior when performing financial and security related tasks. To validate the usefulness of such a framework we first created and used a proof of concept. While the to be developed framework will be modular and allow registration of user experiences through surveys, the proof of concept has a much narrower scope: to validate that the future framework will be able to collect useful data. We used the proof of concept in a small experiment to collect data on user behavior using two different authentication methods.

Analysis of the resulting data shows that user behavior similarities and differences between the two distinct groups were measurable. Our first conclusion was that users who were expected to compare transaction information as part of the authorization process paid more attention when they were aware that such an attack took place against them. The second conclusion is that transaction authorization using What You See Is What You Sign-based authentication methods takes more time compared to What You Enter Is What You Sign.

That we were able to make these conclusions based on the proof of concept's collected data does support the validity of an evaluation framework that measures secure usability behavior by presenting a virtual bank to testers. The envisioned modular framework can be used to improve existing and design new online banking authentication methods on both technical and use levels. Both levels are important, since effective security in online banking relies on good technically grounded design elements that are usable in a secure way.

In the near future, we are going to examine the length of the period in which online users pay more attention after they spotted a single attack. Knowing this length could aid in establishing an interval frequency between actions which raise user awareness when making important decisions related to security in online banking. After that, we intend to release the virtual bank as open source so that it can serve as a common framework for researchers to perform secure usability research in online banking.

Discussion and future work

Discussion and future work

This part of the thesis notes the potential interaction between the research results of the different paths (exploring, expanding and evaluating) for further research. It can be used as a guide to steer efforts meant to further improve security and usability in online customer-bank interaction.

Exploring and Evaluating

After examining worldwide online banks, it was concluded that there is variation in implemented authentication methods. However, they are all based on a small set of information schemes. This is particularly true for the only implemented transaction authorization scheme that actually processes transaction information, known as What You See Is What You Sign. Still there are some new developments, such as the adoption of physical biometrics by using the fingerprint sensor of specific smartphones. It would be a good idea to examine such new developments on a regular base. When new methods are also examined in-depth, the resulting information could be used to slowly steer online banking in a direction with more usable security. This could be done by an independent institute, possibly financed by and cooperating with the banking sector. The discussion, limitations and further research section of Chapter 2 suggested that security researchers around the world could work together by sharing information about the authentication methods that their banks use. It would take security researchers a trivial amount of time to examine the authentication methods and other technologies they themselves use to manage their financial affairs. The suggested research institute could coordinate such a worldwide effort, and the retrieved information can be used for further research. In addition, the effectiveness of existing implemented methods and proposed new methods could be examined by such an institute. The proposed evaluation framework in Part III, Chapter 5 could provide a useful tool in comparing the subtleties of different authentication methods. The proof of concept of an online, web-based test site as discussed in Part III, Chapter 6 can be worked out to a full modular framework. This framework could then also be used to test and share technical implementations of existing authentication methods and new proposals, ideally creating a cycle of continuous improvement.

Expanding

What You Enter Is What You Sign is a new transaction authorization information scheme suggested in this thesis. Its applicability is limited to transactions of which the information is provided by the user, but in this area it has great potential in providing more usable security compared to What You See Is What You Sign. As a concept it shows that alternative methods, not based on existing information schemes, are a worthwhile research subject that should be explored further. The concept in Part II, Chapter 3 is meant to introduce the idea of What You Enter Is What You Sign. This is contrary to Part II, Chapter 4, of which the described potential implementation is far more realistic since it does not concern connected devices. Further research should focus on security, functional and usability aspects of the information scheme, and practical protocols which implement this.

For security, more formal verification of the protocol between device and bank is required to reduce the risk of flaws in its design. The concept protocol as proposed in Part II, Chapter 3 was partly tested by Safet Acifovic, a student of Radboud University [Aci15]. He formally proved the integrity of the data as well as the authenticity of the transaction data and the initiating user. However, due to time constraints he was unable to formally proof non-repudiation. While providing some formal proof, a weakness was found in the concept protocol that would allow in-session replay attacks. This was not unexpected but is still something that should be solved. The given advice is to focus future research and extensive peer reviewing on solving this weakness in addition to further formal verification. The same advice can also be given based on the formal verification in Part II, Chapter 4. In the proposed implementation of What You Enter Is What You Sign in this chapter a similar weakness was found. If one would have to choose between the implementations of Chapter 3 or Chapter 4 for further research, the preference would likely be for the latter. It does not require a connection between the trusted device and the user's untrusted computer, making it inherently more secure.

Functional, the suggested What You Enter Is What You Sign implementation in Chapter 4 is currently limited to Dutch bank accounts. To be more useful, ideally any bank account number should be supported, and there should be support for specifying the currency if international transactions have to be supported and if the exact amount of money to be transferred has to be part of transaction authorization.

Last but not least, usability can be examined and improved further. In Chapter 4 it was assumed that shorter Message Codes are more usable for various reasons, but it was also suggested that alternatives (longer numerical codes or a words-based system) could be perceived as being more usable. This could be examined with user testing, possibly with the earlier discussed framework. Also, more research could be spend on the idea that a manually transferred Message Code is not necessary at all. For example, a QR code containing the exact same information as the Message Code could be projected by a trusted device and scanned by a user's smartphone.

Security, functional and usability are good aspects to research when improving a security product. However, the practical aspect of deployability should not be forgotten. While not part of the research in this thesis, real world implementations rely on aspects such as costs and distribution. Having a dedicated device implies that such a device has to be manufactured and distributed to a customer. Multiply this by

the number of devices necessary (in terms of customers and additional/replacement devices for customers who need this), and the potential costs could be very high. Banks might also be reluctant to give their customers an authentication device if they did not do so in the past, since it is yet another device a user has to carry around.

The concept of What You Enter Is What You Sign could also be implemented cheaper. Instead of relying on a trusted environment, a bank could rely on two untrusted environments: that of a user's home computer and that of a smartphone. The smartphone could simulate the trusted environment for home banking. The authentication scheme would only be usable for home banking, but it would be cheaper to implement due to a lack of physical devices to ship to users. Security would be reduced since only untrusted environments are concerned, but the use of two untrusted environments has been deemed acceptable by banks that use SMS to transfer one-time passwords. Usability would also improve slightly, since a user would not have an additional device to keep track of.

Banks often use the same authentication methods for multiple years, as discussed in Part I, Chapter 2. To adopt What You Enter Is What You Sign at a bank for transactions initiated by the user, it needs to be worked out into a viable option that can compete with other methods in terms of security, usability, functionality and deployability. Any other motivators for banks to choose one method over another also have to be examined, as well as the ways in which What You Enter Is What You Sign can be adopted to accommodate these motivators.

Summary

Summary

This doctoral thesis concerns the exploration, expansion and evaluation of usable security in online banking. Each of these three parts provides its own contribution.

The exploration part of the research consisted of examining the development of home and mobile banking, and the observation of 80 banks spread across the globe over a two year period. If mobile banking continues to develop as home banking did, it can be expected that more mobile banking applications will be written using standard web technologies. The security implication of this is that mobile banking will become a larger attack target since the use of standard web technologies will make attacks scale better across different banks, as it has done to home banking sites. For the 80 banks, the used authentication methods and their ways to secure communications were examined. Communications security is mostly uniform and ‘good enough’ for daily use. Implementations of authentication methods are quite varied, yet for transaction authorization only the What You See Is What You Sign information scheme is used. This scheme allows users to securely verify transactions that were sent to the bank over an insecure channel before they are executed.

The expansion part focused on introducing a new transaction authorization information scheme known as What You Enter Is What You Sign. This scheme adds integrity and authenticity to critical transaction information entered by the user before it is transmitted through an insecure environment. The user is not required to verify transaction information after it has been sent. Compared to What You See Is What You Sign, this improves usability since the user has to perform less actions while also improving security since the user has less opportunities to make mistakes. The two discussed potential implementations were each as a protocol formally verified on several security properties. No unexpected attacks or weaknesses were found.

The evaluation part was dedicated to mechanisms to compare and evaluate online banking user authentication and transaction authorization methods. An existing mechanism that quantifies the qualities of user authentication methods on three levels was expanded with aspects related to online banking. This mechanism was used by seven raters to evaluate four implemented and eight proposed sets of user authentication and transaction authorization methods. An evaluation was also made of the What You Enter Is What You Sign transaction authorization scheme, which indicates that it could provide a good alternative to What You See Is What You Sign. Another evaluation mechanism was designed that focuses on user testing. It concerns a modular web framework that allows the testing of new ideas for authentication and authorization methods, without developing from scratch and without using a physical test center. A proof of concept was developed and used by test candidates.

Usage data was retrieved from which conclusions could be made, which warrants the usefulness of such a framework.

Samenvatting

Samenvatting

Dit proefschrift heeft als titel en als onderwerp het Verkennen, Uitbreiden en Evalueren van Bruikbare Beveiliging van online bankieren. Elk van de drie onderdelen van het onderwerp draagt bij aan het eindresultaat.

Het verkennende deel beslaat een studie naar de ontwikkeling van thuisbankieren en mobiel bankieren, samen de observatie van 80 wereldwijd verspreide banken gedurende een periode van twee jaar. Als mobiel bankieren zich blijft ontwikkelen zoals thuisbankieren dat ooit deed, dan kan verwacht worden dat meer mobiele bankapplicaties geschreven zullen worden met standaard webtechnieken. Hiervan is de implicatie voor beveiliging dat mobiel bankieren een groter doelwit zal worden omdat standaard webtechnieken de reikwijdte van aanvallen ten opzichte van het aantal banken versterken. Dit gebeurde al eerder voor thuisbankieren op een soortgelijke manier. Van de 80 banken werden de gebruikte authenticatiemethoden onderzocht, samen met manier waarop zij communicatie beveiligen. De gebruikte communicatiebeveiliging is veelal uniform en ‘goed genoeg’ voor dagelijks gebruik. De implementaties van authenticatiemethoden variëren sterk, maar voor transactie-authenticatie wordt enkel het What You See Is What You Sign informatieschema gebruikt. Dit schema stelt gebruikers in staat om op een veilige manier transacties te verifiëren die eerder op een onveilige manier naar de bank verstuurd zijn, voordat de transacties worden uitgevoerd.

Het uitbreidende deel richt zich op een nieuw informatieschema om transacties veilig te autoriseren, genaamd What You Enter Is What You Sign. Dit schema voegt integriteit en authenticiteit toe aan kritieke transactiegegevens bij de invoer door de gebruiker, voordat ze verstuurd worden door een onveilige omgeving. Het is niet nodig dat de gebruiker achteraf de transacties nogmaals verifieert. Vergelijken met What You See Is What You Sign verbetert de bruikbaarheid omdat de gebruiker minder acties hoeft uit te voeren. Tegelijkertijd verbetert ook de beveiliging, omdat de gebruiker minder ruimte heeft om fouten te maken. Verschillende beveiligingseigenschappen van het protocol van elk van de twee besproken potentiële implementaties zijn formeel geverifieerd. Bij de verificatie zijn geen onverwachte aanvallen of zwakheden aangetroffen.

Het evaluerende deel staat in het teken van mechanismes voor het vergelijken en evalueren van methoden om voor online bankieren gebruikers te authenticeren en transacties te autoriseren. Een bestaand mechanisme dat de kwaliteiten van authenticatiemethoden voor gebruikers kwantificeerde op drie niveaus is uitgebreid met aspecten gerelateerd aan online bankieren. Dit mechanisme werd door zeven beoordelaars gebruikt om vier geïmplementeerde en acht voorgestelde sets van

gebruikersauthenticatie- en transactieautorisatiemethoden te evalueren. Ook is er een aparte evaluatie gemaakt van het What You Enter Is What You Sign transactieautorisatieschema, waarvan het resultaat de indruk geeft dat het gebruikt kan worden als een goed alternatief voor What You See Is What You Sign. Een ander evaluatiemechanisme is ontworpen dat zich richt op gebruikerstesten. Het betreft een webapplicatie die is ingericht als een modulair raamwerk waarmee nieuwe ideeën voor authenticatie- en autorisatiemethoden getest kunnen worden, zonder dat men het wiel opnieuw moet uitvinden bij het ontwikkelen. Ook is het niet nodig om een fysiek testcentrum te hebben. Een testversie van het raamwerk is ontwikkeld en gebruikt door testkandidaten. Gebruiksgegevens werden verzameld die het nut van een dergelijk raamwerk aantonen.

Bibliography

Bibliography

- [AAJ10] Mohammed AlZomai, Bander AlFayyadh, and Audun Jøsang. Display security for online transactions: SMS-based authentication scheme. In *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, pages 1–7, Nov 2010. (Cited on pages 8, 51, 108, 121, 127, and 147)
- [AAJM08] Mohammed AlZomai, Bander AlFayyadh, Audun Jøsang, and Adrian McCullagh. An experimental investigation of the usability of transaction authorization in online bank security systems. In *Proceedings of the sixth Australasian conference on Information security-Volume 81*, pages 65–73. Australian Computer Society, Inc., 2008. (Cited on pages 51, 60, 63, 79, 147, 153, and 156)
- [ABP⁺13] Nadhem J. AlFardan, Daniel J. Bernstein, Kenneth G. Paterson, Bertram Poettering, and Jacob C. N. Schuldt. On the Security of RC4 in TLS. In *Proceedings of the 22Nd USENIX Conference on Security, SEC'13*, pages 305–320, Berkeley, CA, USA, 2013. USENIX Association. (Cited on pages 43 and 45)
- [Aci15] Safet Acifovic. (Technical paper at Radboud University) Formal model verification of a new online banking authentication, January 2015. (Cited on pages 61, 71, and 164)
- [Agg06] V. Aggelis. Offline Internet banking fraud detection. In *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, pages 2 pp.–, April 2006. (Cited on page 51)
- [ATVO12] Chaitrali Amrutkar, Patrick Traynor, and Paul C. Van Oorschot. Measuring SSL Indicators on Mobile Browsers: Extended Life, or End of the Road? In Dieter Gollmann and Felix C. Freiling, editors, *Information Security*, volume 7483 of *Lecture Notes in Computer Science*, pages 86–103. Springer Berlin Heidelberg, 2012. (Cited on page 51)
- [AVW⁺12] Mamoun Alazab, Sitalakshmi Venkatraman, Paul Watters, Moutaz Alazab, and Ammar Alazab. Cybercrime: The Case of Obfuscated Malware. In Christos K. Georgiadis, Hamid Jahankhani, Elias Pimenidis, Rabih Bashroush, and Ameer Al-Nemrat, editors, *Global Se-*

curity, Safety and Sustainability & e-Democracy, volume 99 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 204–211. Springer Berlin Heidelberg, 2012. (Cited on page 26)

- [AZEHO9] F. Aloul, S. Zahidi, and W. El-Hajj. Two factor authentication using mobile phones. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pages 641–644, May 2009. (Cited on page 35)
- [BdKGP⁺12] A. Blom, G. de Koning Gans, E. Poll, J. de Ruiter, and R. Verdult. Designed to fail: A USB-connected reader for online banking. In *17th Nordic Conference on Secure IT Systems (NordSec 2012)*, volume 7617, 2012. (Cited on page 72)
- [BHVOS12] Joseph Bonneau, Cormac Herley, Paul C. Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567. IEEE, 2012. (Cited on pages 107, 109, and 140)
- [BL16] Sanne Boes and Eric Rutger Leukfeldt. *Cyber-Physical Security at the State, Provincial and Local Level: Protecting Critical Infrastructure*. Springer Science, New York, 2016. (Cited on pages 2 and 3)
- [Buc59] W. Buchholz. Fingers or Fists? (The Choice of Decimal or Binary Representation). *Communications of the ACM*, 2(12):3–11, December 1959. (Cited on page 84)
- [BVOP⁺09] Robert Biddle, Paul C. Van Oorschot, Andrew Patrick, Jennifer Sobey, and Tara Whalen. Browser interfaces and extended validation SSL certificates: an empirical study. In *Proceedings of the 2009 ACM workshop on cloud computing security*, pages 19–30. ACM, 2009. (Cited on page 46)
- [CD12] Kevin Curran and Timothy Dougan. Man in the Browser Attacks. *Int. J. Ambient Comput. Intell.*, 4(1):29–39, January 2012. (Cited on pages 26, 78, 81, and 154)
- [CDDC⁺02] Joris Claessens, Valentin Dem, Danny De Cock, Bart Preneel, and Joos Vandewalle. On the Security of Today’s Online Electronic Banking Systems. *Computers and Security*, 21(3):253–265, June 2002. (Cited on pages 6, 15, 17, 28, 40, and 148)
- [CdP⁺14] E. Constante, J. I. den Hartog, M. Petkovic, S. Etalle, and M. Pechenizkiy. Hunting the Unknown. In *28th Annual IFIP WG 11.3 Working Conference Data and Applications Security and Privacy (DBSec), Vienna, Austria*, volume 8566 of *Lecture Notes in Computer Science*, pages 243–259, Berlin, 2014. Springer. (Cited on page 39)

- [CHI06] Dianne Cyr, Milena Head, and Alex Ivanov. Design aesthetics leading to m-loyalty in mobile commerce. *Information & Management*, 43(8):950 – 963, 2006. (Cited on page 24)
- [CM12] Cas Cremers and Sjouke Mauw. *Operational Semantics and Verification of Security Protocols*. Information Security and Cryptography. Springer, 1 edition, 2012. (Cited on page 98)
- [COLT10] Alain Yee-Loong Chong, Keng-Boon Ooi, Binshan Lin, and Boon-In Tan. Online banking adoption: an empirical analysis. *International Journal of Bank Marketing*, 28(4):267–287, 2010. (Cited on page 23)
- [DBW89] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Manage. Sci.*, 35(8):982–1003, August 1989. (Cited on page 23)
- [DTH06] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why Phishing Works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’06, pages 581–590, New York, NY, USA, 2006. ACM. (Cited on pages 2, 51, and 149)
- [Eis10] Ori Eisen. Catching the fraudulent Man-in-the-Middle and Man-in-the-Browser. *Network Security*, 2010(4):11 – 12, 2010. (Cited on page 26)
- [ETMLP05] William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-capable Cellular Networks. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, CCS ’05, pages 393–404, New York, NY, USA, 2005. ACM. (Cited on page 116)
- [FFC⁺11] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. A Survey of Mobile Malware in the Wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM ’11, pages 3–14, New York, NY, USA, 2011. ACM. (Cited on pages 73, 78, 116, and 149)
- [FHM⁺12] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. Why Eve and Mallory Love Android: An Analysis of Android SSL (in)Security. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS ’12, pages 50–61, New York, NY, USA, 2012. ACM. (Cited on pages 26, 49, 50, and 51)
- [GB15] Melissa A. Gallagher and Michael D. Byrne. Modeling Password Entry on a Mobile Device. In *Proceedings of the International Conference on Cognitive Modeling*, 2015. (Cited on page 85)
- [GIJ⁺12] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. The Most Dangerous Code in

- the World: Validating SSL Certificates in Non-browser Software. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12*, pages 38–49, New York, NY, USA, 2012. ACM. (Cited on pages 26, 49, 50, and 51)
- [GLS09] Ja-Chul Gu, Sang-Chul Lee, and Yung-Ho Suh. Determinants of behavioral intention to mobile banking. *Expert Systems with Applications*, 36(9):11605 – 11616, 2009. (Cited on page 24)
- [Her09] Cormac Herley. So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users. In *Proceedings of the 2009 Workshop on New Security Paradigms Workshop, NSPW '09*, pages 133–144, New York, NY, USA, 2009. ACM. (Cited on pages 2, 86, 107, 114, and 140)
- [Her14] Cormac Herley. More Is Not the Answer. *IEEE Security Privacy*, 12(1):14–19, Jan 2014. (Cited on pages 107, 113, 114, and 140)
- [HKW06] Alain Hiltgen, Thorsten Kramp, and Thomas Weigold. Secure internet banking authentication. *Security & Privacy, IEEE*, 4(2):21–29, 2006. (Cited on page 72)
- [HPN10] A. Hisamatsu, D. Pishva, and G.G.D. Nishantha. Online banking and modern approaches toward its enhanced security. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 2, pages 1459–1463, Feb 2010. (Cited on pages 35, 72, and 147)
- [HVOP09] Cormac Herley, Paul C. Van Oorschot, and Andrew S. Patrick. Passwords: If We’re So Smart, Why Are We Still Using Them? In Roger Dingledine and Philippe Golle, editors, *Financial Cryptography and Data Security*, volume 5628 of *Lecture Notes in Computer Science*, pages 230–237. Springer Berlin Heidelberg, 2009. (Cited on page 28)
- [Jan15] Jurjen Jansen. Studying safe online banking behaviour: A protection motivation theory approach. In *Proceedings of the Ninth International Symposium on Human Aspects of Information Security & Assurance (HAISA)*, pages 120–130, 2015. (Cited on page 3)
- [JL15] Jurjen Jansen and Eric Rutger Leukfeldt. How people help fraudsters steal their money: An analysis of 600 online banking fraud cases. In *Proceedings of the 2015 Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, pages 24–31, 2015. (Cited on pages 2 and 3)
- [JL16] Jurjen Jansen and Eric Rutger Leukfeldt. Phishing and malware attacks on online banking customers in the Netherlands: A qualitative analysis of factors leading to victimization. *International Journal of Cyber Criminology*, 10(1):79–91, 2016. (Cited on pages 2 and 3)

- [JSTB07] Collin Jackson, Daniel R. Simon, Desney Tan, and Adam Barth. An evaluation of extended validation and picture-in-picture phishing attacks. In *Financial Cryptography and Data Security*, pages 281–293. Springer, 2007. (Cited on page 46)
- [JvS16] Jurjen Jansen and Paul van Schaik. Understanding precautionary online behavioural intentions: A comparison of three models. In *Proceedings of the Tenth International Symposium on Human Aspects of Information Security & Assurance (HAISA)*, pages 1–11, 2016. (Cited on page 3)
- [JVZS16] Jurjen Jansen, Sander Veenstra, Renske Zuurveen, and Wouter Philip Stol. Guarding against online threats: Why entrepreneurs take protective measures. *Behaviour & Information Technology*, 35(5):368–379, 2016. (Cited on page 3)
- [KB12] Ankit Kesharwani and Shailendra Singh Bisht. The impact of trust and perceived risk on internet banking adoption in India: An extension of technology acceptance model. *International Journal of Bank Marketing*, 30(4):303–322, 2012. (Cited on page 23)
- [KLSH04] Yufeng Kou, Chang-Tien Lu, S. Sirwongwattana, and Yo-Ping Huang. Survey of fraud detection techniques. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 2, pages 749–754 Vol.2, 2004. (Cited on page 39)
- [KS13] Robert Künnemann and Graham Steel. YubiSecure? Formal security analysis results for the YubiKey and YubiHSM. In *Security and Trust Management*, pages 257–272. Springer, 2013. (Cited on page 71)
- [KSC⁺16] Sven Kiljan, Koen Simoens, Danny De Cock, Marko van Eekelen, and Harald Vranken. A Survey of Authentication and Communications Security in Online Banking. *ACM Computing Surveys*, 49(4):61:1–61:35, December 2016. (Cited on pages 3, 6, 10, and 15)
- [KSDC⁺14] Sven Kiljan, Koen Simoens, Danny De Cock, Marko van Eekelen, and Harald Vranken. Security of Online Banking Systems. Technical Report TR-OU-INF-2014-01, Open University of the Netherlands, February 2014. (Cited on pages 15, 17, 20, 28, 41, 60, 64, 72, 78, 106, 123, and 147)
- [KvEV16] Sven Kiljan, Marko van Eekelen, and Harald Vranken. Towards a virtual bank for evaluating security aspects with focus on user behavior. In *2016 SAI Computing Conference (SAI)*, pages 1068–1075, July 2016. (Cited on pages 3, 10, 11, and 103)
- [KVvE14] Sven Kiljan, Harald Vranken, and Marko. van Eekelen. What You Enter Is What You Sign: Input Integrity in an Online Banking Environment. In *Proceedings of the 2014 Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, pages 40–47, July 2014. (Cited on pages 3, 7, 11, 51, 57, 79, 86, 119, 147, and 154)

- [KVvE16a] Sven Kiljan, Harald Vranken, and Marko van Eekelen. Evaluation of transaction authentication methods for online banking (In Press, Corrected Proof, Available Online). *Future Generation Computer Systems*, 2016. (Cited on pages 3, 9, 11, and 103)
- [KVvE16b] Sven Kiljan, Harald Vranken, and Marko van Eekelen. User-friendly Manual Transfer of Authenticated Online Banking Transaction Data - A Case Study that Applies the What You Enter Is What You Sign Transaction Authorization Information Scheme. In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications - Volume 4: SECRIPT, (ICETE 2016)*, pages 259–270, July 2016. (Cited on pages 3, 8, 11, and 58)
- [Law08] G. Lawton. Is It Finally Time to Worry about Mobile Malware? *Computer*, 41(5):12–14, May 2008. (Cited on page 73)
- [Leu14a] Eric Rutger Leukfeldt. Cybercrime and social ties: Phishing in Amsterdam. *Trends in Organized Crime*, 17(4):231–249, 2014. (Cited on page 2)
- [Leu14b] Eric Rutger Leukfeldt. Phishing for suitable targets in the Netherlands: Routine activity theory and phishing victimization. *Cyberpsychology, Behavior and Social Networking*, 17(8):551–555, 2014. (Cited on page 2)
- [Leu15a] Eric Rutger Leukfeldt. Comparing victims of phishing and malware attacks: Unraveling risk factors and possibilities for situational crime prevention. *International Journal of Advanced Studies in Computer Science and Engineering*, 4(5):26–32, 2015. (Cited on page 2)
- [Leu15b] Eric Rutger Leukfeldt. Organised cybercrime and social opportunity structures: A proposal for future research directions. *European Review of Organised Crime*, 2(2):91–103, 2015. (Cited on page 2)
- [LJ15] Eric Rutger Leukfeldt and Jurjen Jansen. Cyber criminal networks and money mules: An analysis of low-tech and high-tech fraud attacks in the Netherlands. *International Journal of Cyber Criminology*, 9(2):173–184, 2015. (Cited on pages 2 and 3)
- [LKS16a] Eric Rutger Leukfeldt, Edward R. Kleemans, and Wouter Philip Stol. Cybercriminal networks, social ties and online forums: Social ties versus digital ties with phishing and malware networks. *Crime, Law and Social Change*, 2016. (Cited on page 2)
- [LKS16b] Eric Rutger Leukfeldt, Edward R. Kleemans, and Wouter Philip Stol. (in press) From low tech locals to high tech specialists: A typology of phishing networks. *Crime, Law and Social Change*, 2016. (Cited on page 2)
- [LKS16c] Eric Rutger Leukfeldt, Edward R. Kleemans, and Wouter Philip Stol. (in press) Origin, growth and criminal capabilities of cybercriminal

- networks: An international empirical analysis. *Crime, Law and Social Change*, 2016. (Cited on page 2)
- [LL05a] Vincent S. Lai and Honglei Li. Technology Acceptance Model for Internet Banking: An Invariance Analysis. *Inf. Manage.*, 42(2):373–386, January 2005. (Cited on page 23)
- [LL05b] Pin Luarn and Hsin-Hui Lin. Toward an understanding of the behavioral intention to use mobile banking. *Computers in Human Behavior*, 21(6):873 – 891, 2005. (Cited on page 24)
- [Low97] G. Lowe. A hierarchy of authentication specifications. In *Proceedings 10th Computer Security Foundations Workshop*, pages 31–43, Jun 1997. (Cited on page 98)
- [LSH⁺12] Shujun Li, Ahmad-Reza Sadeghi, Sören Heisrath, Roland Schmitz, and Junaid Jameel Ahmad. hPIN/hTAN: A Lightweight and Low-Cost E-Banking Solution against Untrusted Computers. In George Danezis, editor, *Financial Cryptography and Data Security*, volume 7035 of *Lecture Notes in Computer Science*, pages 235–249. Springer Berlin Heidelberg, 2012. (Cited on pages 8, 108, 119, 121, 127, and 147)
- [MBJ11] M. Mihajlov, B.J. Blazic, and S. Josimovski. Quantifying Usability and Security in Authentication. In *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*, pages 626–629, July 2011. (Cited on pages 107 and 110)
- [Men92] Karl Menger. On the Origin of Money. *The Economic Journal*, 2(6):239–255, 1892. (Cited on page 1)
- [Mer01] Craig A Mertler. Designing scoring rubrics for your classroom. *Practical Assessment, Research & Evaluation*, 7(25):1–10, 2001. (Cited on pages 108 and 110)
- [MJB11] M. Mihajlov, B. Jerman-Blazic, and S. Josimovski. A conceptual framework for evaluating usable security in authentication mechanisms - usability perspectives. In *Network and System Security (NSS), 2011 5th International Conference on*, pages 332–336, Sept 2011. (Cited on pages 107 and 110)
- [MVO07] Mohammad Mannan and Paul C. Van Oorschot. Using a Personal Device to Strengthen Password Authentication from an Untrusted Computer. In Sven Dietrich and Rachna Dhamija, editors, *Financial Cryptography and Data Security*, volume 4886 of *Lecture Notes in Computer Science*, pages 88–103. Springer Berlin Heidelberg, 2007. (Cited on page 32)
- [MVO08] Mohammad Mannan and Paul C. Van Oorschot. Security and Usability: The Gap in Real-world Online Banking. In *Proceedings of the 2007 Workshop on New Security Paradigms*, NSPW '07, pages 1–14, New York, NY, USA, 2008. ACM. (Cited on pages 2 and 4)

- [NW97] William S. Neilson and Harold Winter. On criminals' risk attitudes. *Economics Letters*, 55(1):97–102, August 1997. (Cited on page 1)
- [ODC15] Lucky Onwuzurike and Emiliano De Cristofaro. Danger is My Middle Name: Experimenting with SSL Vulnerabilities in Android Apps. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, pages 15:1–15:6, New York, NY, USA, 2015. ACM. (Cited on page 50)
- [Ogh09] Egwali Annie Oghenerukeybe. Customers perception of security indicators in online banking sites in Nigeria. *Journal of Internet Banking and Commerce*, 14(1):1–15, 2009. (Cited on page 51)
- [Oll08] Gunter Ollmann. The evolution of commercial malware development kits and colour-by-numbers custom malware. *Computer Fraud & Security*, 2008(9):4 – 7, 2008. (Cited on page 26)
- [Ols08] Kai A. Olsen. The \$100,000 Keying Error. *Computer*, 41(4):108–107, April 2008. (Cited on page 147)
- [PdR13] Erik Poll and Joeri de Ruiter. The Radboud Reader: A Minimal Trusted Smartcard Reader for Securing Online Transactions. In *Policies and Research in Identity Management - Third IFIP WG 11.6 Working Conference, IDMAN 2013, London, UK, April 8-9, 2013. Proceedings*, pages 107–120, 2013. (Cited on pages 72, 78, and 123)
- [PPKP04] Tero Pikkarainen, Kari Pikkarainen, Heikki Karjaluo, and Seppo Pahnla. Consumer acceptance of online banking: an extension of the technology acceptance model. *Internet Research*, 14(3):224–235, 2004. (Cited on page 23)
- [QS07] J.T.S. Quah and M. Sriganesh. Real Time Credit Card Fraud Detection using Computational Intelligence. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 863–868, Aug 2007. (Cited on page 39)
- [Ren04] Karen Renaud. Quantifying the Quality of Web Authentication Mechanisms: A Usability Perspective. *J. Web Eng.*, 3(2):95–123, October 2004. (Cited on pages 9, 107, 109, and 112)
- [RP98] Tim Redhead and Dean Povey. The Problems With Secure On-line Banking. *Proceedings of the XVIIth annual South East Asia Regional Conference (SEARCC 98)*, 1998. (Cited on pages 2, 5, 27, and 60)
- [RSB⁺15] Bradley Reaves, Nolen Scaife, Adam Bates, Patrick Traynor, and Kevin R. B. Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. In *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC'15, pages 17–32, Berkeley, CA, USA, 2015. USENIX Association. (Cited on pages 26, 49, 50, and 51)

- [Rux06] Graeme D. Ruxton. The unequal variance t-test is an underused alternative to Student's t-test and the Mann-Whitney U test. *Behavioral Ecology*, 17(4):688–690, 2006. (Cited on page 159)
- [SBVOP08] Jennifer Sobey, Robert Biddle, Paul C. Van Oorschot, and Andrew Patrick. Exploring User Reactions to New Browser Cues for Extended Validation Certificates. In *Proceedings of the 13th European Symposium on Research in Computer Security: Computer Security*, ESORICS '08, pages 411–427, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 46)
- [Sch05] Bruce Schneier. Two-factor Authentication: Too Little, Too Late. *Communications of the ACM*, 48(4):136–136, April 2005. (Cited on pages 5, 28, and 149)
- [SF05] M. Angela Sasse and Ivan Flechais. *Usable Security: Why Do We Need It? How Do We Get It?*, pages 13–30. O'Reilly, Sebastopol, United States, 2005. (Cited on pages 3 and 4)
- [SFG09] G. Starnberger, L. Frohofer, and K.M. Goeschka. QR-TAN: Secure Mobile Transaction Authentication. In *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, pages 578–583, March 2009. (Cited on pages 8, 108, 121, 127, and 147)
- [SG12] R. Sobti and G. Geetha. Cryptographic Hash Functions: A Review. *International Journal of Computer Science Issues*, 9(2):461–479, March 2012. (Cited on page 90)
- [SH04] Ton Slewe and Mark Hoogenboom. Who Will Rob You on the Digital Highway? *Commun. ACM*, 47(5):56–60, May 2004. (Cited on pages 27 and 72)
- [Sri13] Atul Srivastava. Mobile Banking and Sustainable Growth. *American Journal of Economics and Business Administration*, 5(3):89–94, 2013. (Cited on page 27)
- [SZO05] Xiaoyuan Suo, Ying Zhu, and G.S. Owen. Graphical passwords: a survey. In *Computer Security Applications Conference, 21st Annual*, pages 10 pp.–472, Dec 2005. (Cited on page 32)
- [TPC10] K. Taohai, S. Phimoltares, and N. Cooharojananone. Usability Comparisons of Seven Main Functions for Automated Teller Machine (ATM) Banking Service of Five Banks in Thailand. In *Computational Science and Its Applications (ICCSA), 2010 International Conference on*, pages 176–182, March 2010. (Cited on page 147)
- [Tri00] Ashton D Trice. *A handbook of classroom assessment*. Longman, 2000. (Cited on page 110)
- [VOZ⁺12] Amit Vasudevan, Emmanuel Owusu, Zongwei Zhou, James Newsome, and Jonathan M. McCune. Trustworthy Execution on Mobile Devices:

- What security properties can my mobile platform give me? In *Trust and Trustworthy Computing*, pages 159–178. Springer, 2012. (Cited on page 73)
- [Wes08] Ryan West. The Psychology of Security. *Communications of the ACM*, 51(4):34–40, April 2008. (Cited on page 146)
- [WH11] T. Weigold and A. Hiltgen. Secure confirmation of sensitive transaction data in modern Internet banking services. In *Internet Security (WorldCIS), 2011 World Congress on*, pages 125–132, Feb 2011. (Cited on pages 8, 35, 51, 81, 108, 119, 122, 123, 127, and 147)
- [WLC⁺13] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4):449–475, 2013. (Cited on page 51)
- [WMC⁺16] Sarah Wiseman, Gustavo Soto Mino, Anna L. Cox, Sandy J.J. Gould, Joanne Moore, and Chris Needham. Use Your Words: Designing One-time Pairing Codes to Improve User Experience (to be published). In *Proceedings of the 34rd Annual ACM Conference on Human Factors in Computing Systems*. ACM Publications, 2016. (Cited on pages 84 and 91)
- [Yee02] Ka-Ping Yee. User Interaction Design for Secure Systems. In Robert Deng, Feng Bao, Jianying Zhou, and Sihan Qing, editors, *Information and Communications Security*, volume 2513 of *Lecture Notes in Computer Science*, pages 278–290. Springer Berlin Heidelberg, 2002. (Cited on pages 107, 115, 116, and 146)
- [YFP10] Shumaila Y. Yousafzai, Gordon R. Foxall, and John G. Pallister. Explaining Internet Banking Behavior: Theory of Reasoned Action, Theory of Planned Behavior, or Technology Acceptance Model? *Journal of Applied Social Psychology*, 40(5):1172–1202, 2010. (Cited on page 51)
- [ZH90] Moshe Zviran and William J. Haga. Cognitive passwords: The key to easy access control. *Computers & Security*, 9(8):723 – 736, 1990. (Cited on page 32)
- [Zur05] M.E. Zurko. User-centered security: stepping up to the grand challenge. In *Computer Security Applications Conference, 21st Annual*, pages 14 pp.–202, Dec 2005. (Cited on page 115)

Curriculum Vitae

Curriculum Vitae

Sven Kiljan

October 02, 1985 Born in Alkmaar, the Netherlands

September 2003 - August 2007 Pre-university secondary education
Middle-level applied education, level 4 - ICT Management
Horizon College, Hoorn, the Netherlands

September 2003 - August 2007 (different periods in time span)
Several internships for system, information and network management
PricewaterhouseCoopers, Utrecht, the Netherlands

September 2007 - August 2010 Bachelor of Business Informatics (cum laude)
Technique, Design and Informatics cluster
Hogeschool Inholland Alkmaar, Alkmaar, the Netherlands

February 2010 - July 2010
Bachelor thesis, research internship on license management
PricewaterhouseCoopers, Utrecht, the Netherlands

September 2010 - August 2012 Master of Science
Faculty of Sciences, Department of Computer Sciences
VU University Amsterdam, Amsterdam, the Netherlands

January 2012 - July 2012
Master thesis, research internship on quantifying software development
Rabobank, Utrecht, the Netherlands

September 2012 - June 2017 Doctor of Philosophy
Management, Science and Technology
Open University of the Netherlands

October 2016 -
Security Expert, SIEM management
CCV Nederland, Arnhem, the Netherlands