

# Attaining Basically Everything in Attribute-Based Encryption

Simplifying the Design of Practical Schemes via Pair Encodings

Marloes Venema

Radboud University, Nijmegen, the Netherlands

OUrsi, December 13th 2022

# Motivation

- ▶ Attribute-based encryption (ABE) is an advanced type of public-key encryption in which the keys are associated with attributes
- ▶ Allows for enforcement of access control on a cryptographic level
- ▶ Various use cases, e.g., cloud-based, IoT settings, email encryption

# Motivation

- ▶ Attribute-based encryption (ABE) is an advanced type of public-key encryption in which the keys are associated with attributes
- ▶ Allows for enforcement of access control on a cryptographic level
- ▶ Various use cases, e.g., cloud-based, IoT settings, email encryption
- ▶ Many ABE schemes exist, with various different properties
- ▶ Some of these properties are desirable for practice

# Motivation

- ▶ Attribute-based encryption (ABE) is an advanced type of public-key encryption in which the keys are associated with attributes
- ▶ Allows for enforcement of access control on a cryptographic level
- ▶ Various use cases, e.g., cloud-based, IoT settings, email encryption
- ▶ Many ABE schemes exist, with various different properties
- ▶ Some of these properties are desirable for practice
- ▶ Not really clear which schemes are efficient (enough)
- ▶ **Goal of my research:** analyzing existing schemes, and eventually, achieving all necessary properties as efficiently as possible

# High-level overview

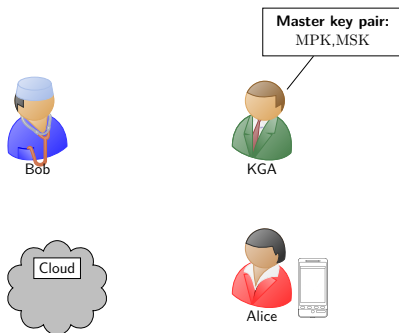
- 1 Introduction to ABE
- 2 The pair encodings framework
- 3 ABE Squared
- 4 New schemes
- 5 Conclusions

# High-level overview

- 1 Introduction to ABE
- 2 The pair encodings framework
- 3 ABE Squared
- 4 New schemes
- 5 Conclusions

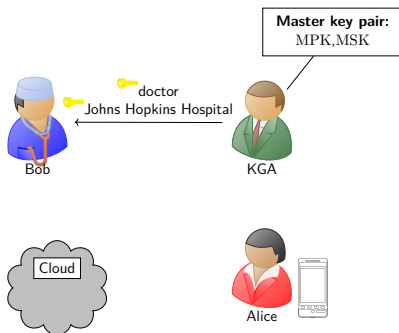
# Ciphertext-policy attribute-based encryption (CP-ABE)

## Setup:



# Ciphertext-policy attribute-based encryption (CP-ABE)

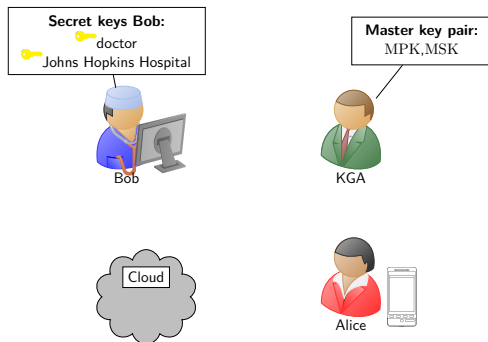
## Key generation:





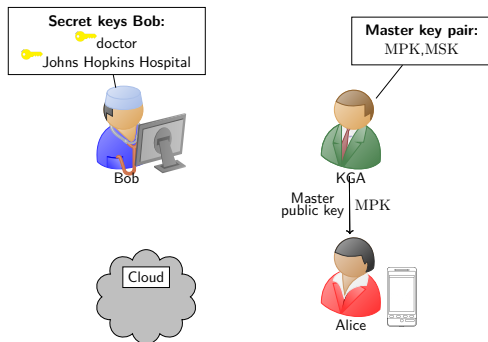
# Ciphertext-policy attribute-based encryption (CP-ABE)

## Key generation:



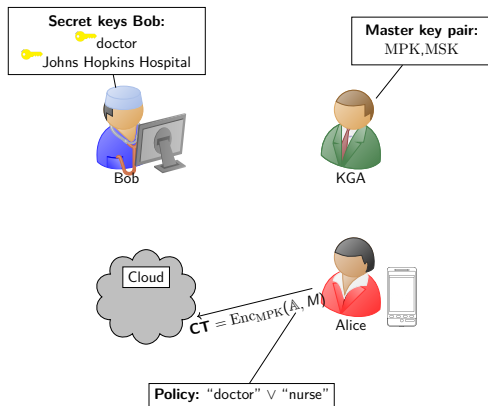
# Ciphertext-policy attribute-based encryption (CP-ABE)

## Encryption:

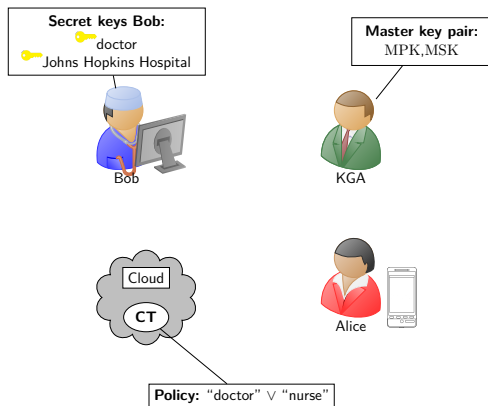


# Ciphertext-policy attribute-based encryption (CP-ABE)

## Encryption:

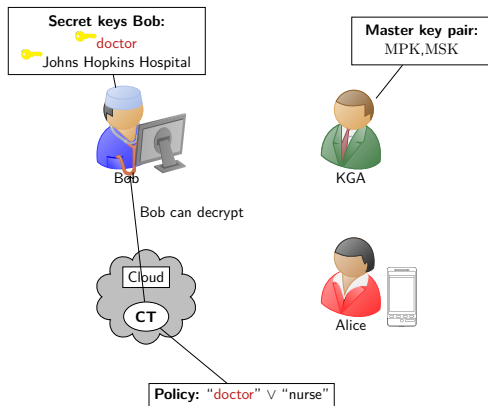


# Ciphertext-policy attribute-based encryption (CP-ABE)

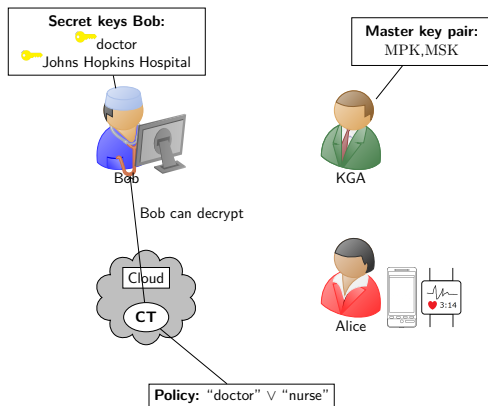


# Ciphertext-policy attribute-based encryption (CP-ABE)

## Decryption:




# Ciphertext-policy attribute-based encryption (CP-ABE)






# Use cases

Many use cases, e.g.,

- ▶ European Telecommunications Standards Institute (ETSI): 
- ▶ Cloud
- ▶ Internet of Things (IoT)
- ▶ WLAN
- ▶ Mobile services

# Use cases

Many use cases, e.g.,

- ▶ European Telecommunications Standards Institute (ETSI): 
- ▶ Cloud
- ▶ Internet of Things (IoT)
- ▶ WLAN
- ▶ Mobile services
- ▶ Cloudflare: Geo Key Manager 
- ▶ Radboud University's iHub: PostGuard 



# Requirements for ABE

These use cases share many common requirements for ABE:

- ▶ **Expressive policies:** policies should support Boolean formulas consisting of AND and OR operators
- ▶ **Large universes:** attribute could be any arbitrary string, e.g., names, roles, MAC addresses
- ▶ **Unbounded:** no bounds on any parameters, such as the length of the policies or attribute sets

# Requirements for ABE

These use cases share many common requirements for ABE:

- ▶ **Expressive policies:** policies should support Boolean formulas consisting of AND and OR operators
- ▶ **Large universes:** attribute could be any arbitrary string, e.g., names, roles, MAC addresses
- ▶ **Unbounded:** no bounds on any parameters, such as the length of the policies or attribute sets

Some use cases also require **non-monotonicity**, i.e., the support of NOT operators in the policies.

# Requirements for ABE

These use cases share many common requirements for ABE:

- ▶ **Expressive policies:** policies should support Boolean formulas consisting of AND and OR operators
- ▶ **Large universes:** attribute could be any arbitrary string, e.g., names, roles, MAC addresses
- ▶ **Unbounded:** no bounds on any parameters, such as the length of the policies or attribute sets

Some use cases also require **non-monotonicity**, i.e., the support of NOT operators in the policies.

**Storage and computational efficiency** requirements may vary per use case.

# Requirements for storage and computational efficiency

## Examples:

- ▶ WLAN and cloud settings: fast decryption
- ▶ Internet of Things: small ciphertexts, fast encryption
- ▶ PostGuard (email encryption): fast key generation

# Pairing-based ABE

- ▶ Currently, pairing-based ABE is the most popular candidate
- ▶ Can satisfy most properties
- ▶ Good security guarantees
- ▶ Efficient enough for most settings

# Pairing-based ABE

- ▶ Currently, pairing-based ABE is the most popular candidate
- ▶ Can satisfy most properties
- ▶ Good security guarantees
- ▶ Efficient enough for most settings
- ▶ Unfortunately, not post-quantum secure
- ▶ Post-quantum secure schemes exist, but still heavily under development

# Taxonomy

We analyzed many ABE schemes with respect to these properties [VAH22]:

Scheme	KP/CP	Expr.	Negations	LU	Unbounded
[SW05] I	KP	X	X	X	✓
[SW05] II	KP	X	X	✓	X
[GPSW06] I	KP	✓	X	X	✓
[GPSW06] II	KP	✓	X	✓	X
[Cha07] I	KP	X	X	X	✓
[Cha07] II	KP	X	X	✓	X
[BSW07]	CP	✓	X	✓	✓
[CN07]	KP*	X	X	X	✓
[OSW07]	KP	✓	✓	✓	X
[NYO08] I	KP*	X	X	X	✓
[NYO08] II	CP	X	X	X	✓
[CC09]	KP	X	X	X	✓
[YWRL10]	KP*	X	X	X	✓
[HLR10]	CP	X	X	X	✓
[LOS <sup>+</sup> 10]	CP	✓	X	X	X
[OT10]	KP,CP	✓	✓	✓	X
[Wat11] I	CP	✓	X	X	✓
[Wat11] II	CP	✓	X	X	X
[Wat11] III	CP	✓	X	X	X
[LHC <sup>+</sup> 11]	KP*	X	X	X	✓
[ALdP11]	KP	✓	✓	✓	X
[LW11a] I	CP	✓	X	X	X
[LW11a] II	CP	✓	X	X	X
[LW11b]	KP	✓	X	✓	✓
[GHW11] I	CP	✓	X	✓	✓
[GHW11] II	KP	✓	X	✓	✓
[LCH <sup>+</sup> 11]	CP	✓	X	X	✓

Scheme	KP/CP	Expr.	Negations	LU	Unbounded
[LW12]	CP	✓	X	X	✓
[SSW12]	KP,CP	✓	X	X	X
[OT12]	KP,CP	✓	✓	✓	✓
[HW13]	KP	✓	X	X	✓
[CCL <sup>+</sup> 13]	KP	X	X	✓	X
[OT13]	KP,CP	✓	✓	✓	X
[LCL <sup>+</sup> 13]	KP	X	X	X	✓
[RW13]	KP,CP	✓	X	✓	✓
[LCW13]	CP	✓	X	X	✓
[HW14]	KP,CP	✓	X	✓	✓
[KL15]	KP	✓	X	X	✓
[RW15]	CP	✓	X	✓	✓
[CGW15]	KP,CP	✓	X	X	X
[LW15]	CP	✓	X	✓	✓
[ZGT <sup>+</sup> 16] I	KP	✓	X	X	✓
[ZGT <sup>+</sup> 16] II	KP	✓	X	✓	X
[AHM <sup>+</sup> 16]	KP	✓	X	✓	✓
[CDLQ16]	CP	✓	X	✓	✓
[ABGW17]	KP,CP	✓	X	✓	✓
[AC17a]	KP,CP	✓	X	✓	X
[CGKW18]	KP,CP	✓	X	✓	X
[LYZL18]	CP	✓	X	X	X
[MJ18]	CP	X	X	X	✓
[KW19] I,II	KP,CP	✓	X	X	✓
[KW19] III	KP	✓	X	✓	✓
[TKN20]	KP,CP	✓	✓	✓	✓

# Research directions

- ▶ Many properties can be achieved, even with good security guarantees
- ▶ How efficient are these schemes?



# Research directions

- ▶ Many properties can be achieved, even with good security guarantees
- ▶ How efficient are these schemes?
- ▶ Multiple problems in this area:
  - ▶ Comparing the efficiency of ABE is difficult
  - ▶ Schemes with many desirable properties are typically considered less efficient than schemes with fewer properties

# Research directions

- ▶ Many properties can be achieved, even with good security guarantees
- ▶ How efficient are these schemes?
- ▶ Multiple problems in this area:
  - ▶ Comparing the efficiency of ABE is difficult
  - ▶ Schemes with many desirable properties are typically considered less efficient than schemes with fewer properties

## Goal of my research: Simplifying the

- ▶ accurate benchmarking of efficiency
- ▶ design of schemes that
  - ▶ can satisfy all necessary properties
  - ▶ with the desired efficiency requirements

# High-level overview

- 1 Introduction to ABE
- 2 The pair encodings framework**
- 3 ABE Squared
- 4 New schemes
- 5 Conclusions

# Pairings

A **pairing** is a function  $e: \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$  over three groups  $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$  of prime order  $p$  such that

- ▶ for all  $a, b \in \mathbb{Z}_p$  and  $g \in \mathbb{G}, h \in \mathbb{H}$  we have  $e(g^a, h^b) = e(g, h)^{ab}$ ;
- ▶  $e(g, h) \neq 1$ ;
- ▶  $e$  is efficient.

# Pairings

A **pairing** is a function  $e: \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$  over three groups  $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$  of prime order  $p$  such that

- ▶ for all  $a, b \in \mathbb{Z}_p$  and  $g \in \mathbb{G}, h \in \mathbb{H}$  we have  $e(g^a, h^b) = e(g, h)^{ab}$ ;
- ▶  $e(g, h) \neq 1$ ;
- ▶  $e$  is efficient.

Essentially allows you to exponentiate with “hidden” exponent, e.g., consider pairing-based version of ElGamal:

$$\begin{array}{ccc}
 \text{PK} = g^\alpha, & \text{SK} = \alpha, & (\text{CT}_1, \text{CT}_2) = (M \cdot \text{PK}^\beta, g^\beta) \\
 \downarrow & \downarrow & \downarrow \\
 \text{PK} = e(g, h)^\alpha, & \text{SK} = h^\alpha, & (\text{CT}_1, \text{CT}_2) = (M \cdot \text{PK}^\beta, g^\beta)
 \end{array}$$

Decrypt by computing  $\text{CT}_1 / e(\text{CT}_2, \text{SK})$

# Structure of pairing-based ABE

Most schemes have the following structure:

- ▶ Setup:  $\text{MPK} = (g, h, A = e(g, h)^\alpha, g^b, \dots)$
- ▶ KeyGen:  $\text{SK}_S = (h^{\alpha+rb}, h^r, \dots)$
- ▶ Encrypt:  $\text{CT}_A = (M \cdot A^s, g^s, \dots)$
- ▶ Decrypt: pairing elements in  $\text{SK}_S$  and  $\text{CT}_A$  to obtain  $A^s = e(g, h)^{\alpha s}$

# Structure of pairing-based ABE

Most schemes have the following structure:

- ▶ Setup:  $\text{MPK} = (g, h, A = e(g, h)^\alpha, g^b, \dots)$
- ▶ KeyGen:  $\text{SK}_S = (h^{\alpha+rb}, h^r, \dots)$
- ▶ Encrypt:  $\text{CT}_A = (M \cdot A^s, g^s, \dots)$
- ▶ Decrypt: pairing elements in  $\text{SK}_S$  and  $\text{CT}_A$  to obtain  $A^s = e(g, h)^{\alpha s}$

Most concrete differences in keys and ciphertexts:

$$\text{SK}_S = h^{\mathbf{k}(\alpha, \mathbf{r}, \mathbf{b}, S)} = (h^{k_1}, h^{k_2}, \dots),$$

$$\text{CT}_A = (M \cdot e(g, h)^{\alpha s}, g^{\mathbf{c}(\mathbf{s}, \mathbf{b}, A)} = (g^{c_1}, g^{c_2}, \dots)),$$

where  $\mathbf{k} = (k_1, \dots)$  and  $\mathbf{c} = (c_1, \dots)$  are vectors, and  $k_i$  and  $c_i$  are polynomials.

# Pair encodings

- ▶ **Pair encodings:** common abstraction of pairing-based ABE [Att14]
- ▶ Considers “what happens in the exponent”
- ▶ More concretely: considers vectors  $\mathbf{k}$  and  $\mathbf{c}$



# Pair encodings

- ▶ **Pair encodings:** common abstraction of pairing-based ABE [Att14]
- ▶ Considers “what happens in the exponent”
- ▶ More concretely: considers vectors  $\mathbf{k}$  and  $\mathbf{c}$
- ▶ Pairing a key element with a ciphertext element corresponds to multiplying entry in  $\mathbf{k}$  with entry in  $\mathbf{c}$
- ▶ Much more compact notation:

*Setup( $\mathcal{G}$ )*. The setup algorithm takes as input the number of attributes in the system,  $k$ , then chooses a group  $\mathcal{G}$  of prime order  $p$ , a generator  $g$ , and  $\ell$  random group elements  $h_1, \dots, h_\ell \in \mathcal{G}$  that are associated with the  $\ell$  attributes in the system. In addition, it chooses random exponents  $\alpha, r \in \mathbb{Z}_p$ .

The public key is published as

$$PK := (g, h_1 g^\alpha, g^r, h_1, \dots, h_\ell).$$

The authority sets  $MSK := g^\alpha$  as the master secret key.

*Encrypt( $M, \mathcal{A}, M$ )*. The encryption algorithm takes as input the public parameters  $PK$  and a message  $M$  to encrypt. In addition, it takes as input an LSSS access structure  $(M, \rho)$ . The function  $\rho$  associates rows of  $M$  to attributes.

Let  $M$  be an  $n \times k$  matrix. The algorithm first chooses a random vector  $s = (s_1, \dots, s_k) \in \mathbb{Z}_p^k$ . These values will be used to choose the encryption exponent  $\alpha$ .

For  $i = 1$  to  $k$ , it chooses  $b_i = \alpha \cdot h_i$ , where  $h_i$  is the vector corresponding to the  $i$ th row of  $M$ . In addition, the algorithm chooses random  $s'_1, \dots, s'_\ell \in \mathbb{Z}_p$ .

The ciphertext is published as CT =

$$C = M \cdot (s_1 g^{\alpha s'_1}, \dots, s_k g^{\alpha s'_k})$$

$$(s_1, \dots, s_k g^{\alpha s'_1}, s_2, \dots, s_k) \cdot (s'_1, \dots, s'_\ell) = s'_1 b_1 g^{\alpha s'_1}, \dots, s_k g^{\alpha s'_k}$$

along with a description of  $(M, \rho)$ .

*KeyGen( $MSK, \mathcal{S}$ )*. The key generation algorithm takes as input the master secret key and a set  $\mathcal{S}$  of attributes. The algorithm first chooses a random  $r \in \mathbb{Z}_p$ . It outputs the private key as

$$K = (s'_1 g^{\alpha r}, \dots, s'_\ell g^{\alpha r}, \{s_i \cdot r\}_{i \in \mathcal{S}}, K_0 = G).$$

*Decrypt( $CT, K, M$ )*. The decryption algorithm takes as input a ciphertext  $CT$  for some message  $M$ , and a private key for  $\mathcal{S}$ . Suppose that  $\mathcal{S}$  satisfies the access structure and let  $I = \{1, 2, \dots, k\}$  be defined as  $I = \{i : \rho(i) \in \mathcal{S}\}$ . Then, let  $\mathbf{h}_I = (h_{i_1}, \dots, h_{i_\ell})$  be the set of random elements  $h_i$  for  $i \in I$ . Let  $\mathbf{c}_I$  be the vector corresponding to  $M$  then  $\sum_{i \in I} c_i h_i = \alpha \cdot G$ . (Note there could potentially be different ways of choosing  $\mathbf{h}_I$  or  $\mathbf{c}_I$  to satisfy this.)

The decryption algorithm first computes

$$\begin{aligned} & \langle \mathbf{h}_I, K \rangle / \langle \mathbf{h}_I, (s'_1 g^{\alpha r}, \dots, s'_\ell g^{\alpha r}, K_0) \rangle \\ & = \langle s'_1 h_{i_1} g^{\alpha r}, \dots, s'_\ell h_{i_\ell} g^{\alpha r}, \mathbf{h}_I \rangle / \langle s'_1 h_{i_1} g^{\alpha r}, \dots, s'_\ell h_{i_\ell} g^{\alpha r}, \mathbf{h}_I \rangle = \langle \mathbf{h}_I, \alpha \cdot G \rangle / \langle \mathbf{h}_I, \alpha \cdot G \rangle \end{aligned}$$

The decryption algorithm can then divide out this value from  $CT$  and obtain the message  $M$ .

 $\Rightarrow$ 

$$\begin{aligned} \mathbf{k}(\alpha, r, \mathbf{b}, \mathcal{S}) &= (\alpha + rb, r, \{rb_i\}_{i \in \mathcal{S}}) \\ \mathbf{c}(s, s'_i, \mathbf{b}, \mathbb{A}) &= (s, \{\lambda_i b + s'_i b_i, s'_i\}_{i \in \mathbb{A}}) \end{aligned}$$

# Pair encodings simplify design and analysis

Analysis:

- ▶ **Security proofs:** either information-theoretic [Att14, Wee14, CGW15] or even algebraic [AC17b]
- ▶ **Cryptanalysis:** automatically [ABGW17] or manually [VA21]
- ▶ **Fairer efficiency comparison:** ABE Squared [dIPVA]

# Pair encodings simplify design and analysis

Analysis:

- ▶ **Security proofs:** either information-theoretic [Att14, Wee14, CGW15] or even **algebraic** [AC17b]
- ▶ **Cryptanalysis:** automatically [ABGW17] or manually [VA21]
- ▶ **Fairer efficiency comparison:** ABE Squared [dIPVA]

Especially the **algebraic** notion of security simplifies the design of schemes:

- ▶ **New schemes:** e.g., designing more efficient schemes [AC17b, VA22a, VA22b]
- ▶ **Composing existing schemes:** building larger systems without complicating the proofs
- ▶ **Transforming existing schemes:** e.g., to achieve properties that are otherwise difficult to achieve, e.g., support of NOT operators [Att19, Amb21] or CCA-security [VB22]

# Schemes of interest

Many existing schemes can be captured in the pair encodings framework, but we mainly considered

# Schemes of interest

Many existing schemes can be captured in the pair encodings framework, but we mainly considered

- ▶ Wat11-IV [Wat11, Wat08]
- ▶ RW13 [RW13]
- ▶ AC17-LU [AC17b]

## Schemes of interest

Many existing schemes can be captured in the pair encodings framework, but we mainly considered

- ▶ Wat11-IV [Wat11, Wat08]
- ▶ RW13 [RW13]
- ▶ AC17-LU [AC17b]

These all satisfy the “basic requirements”, i.e., support expressive policies and large universes, and are unbounded in all parameters.

# Schemes of interest

Many existing schemes can be captured in the pair encodings framework, but we mainly considered

- ▶ Wat11-IV [Wat11, Wat08]
- ▶ RW13 [RW13]
- ▶ AC17-LU [AC17b]

These all satisfy the “basic requirements”, i.e., support expressive policies and large universes, and are unbounded in all parameters. However, only RW13 can be transformed to support NOT operators in the policies.

# Schemes of interest

Many existing schemes can be captured in the pair encodings framework, but we mainly considered

- ▶ Wat11-IV [Wat11, Wat08]
- ▶ RW13 [RW13]
- ▶ AC17-LU [AC17b]

These all satisfy the “basic requirements”, i.e., support expressive policies and large universes, and are unbounded in all parameters. However, only RW13 can be transformed to support NOT operators in the policies.

How efficient are these schemes?



# High-level overview

- 1 Introduction to ABE
- 2 The pair encodings framework
- 3 ABE Squared**
- 4 New schemes
- 5 Conclusions

# Accurately Benchmarking Efficiency

- ▶ Common practice: comparing efficiency of implementations
- ▶ Unfortunately: few implementations of ABE schemes exist
- ▶ More unfortunate: existing implementations may not be fairly comparable

# Accurately Benchmarking Efficiency

- ▶ Common practice: comparing efficiency of implementations
- ▶ Unfortunately: few implementations of ABE schemes exist
- ▶ More unfortunate: existing implementations may not be fairly comparable
- ▶ Problem: they use very outdated groups  $\mathbb{G}$ ,  $\mathbb{H}$ ,  $\mathbb{G}_T$
- ▶ Another problem: many layers of optimization
- ▶ Some layers really depend on what is trying to be optimized, e.g., decryption for the cloud setting or encryption for IoT

# Accurately Benchmarking Efficiency

- ▶ Common practice: comparing efficiency of implementations
- ▶ Unfortunately: few implementations of ABE schemes exist
- ▶ More unfortunate: existing implementations may not be fairly comparable
- ▶ Problem: they use very outdated groups  $\mathbb{G}$ ,  $\mathbb{H}$ ,  $\mathbb{G}_T$
- ▶ Another problem: many layers of optimization
- ▶ Some layers really depend on what is trying to be optimized, e.g., decryption for the cloud setting or encryption for IoT
- ▶ **Our solution:** ABE Squared

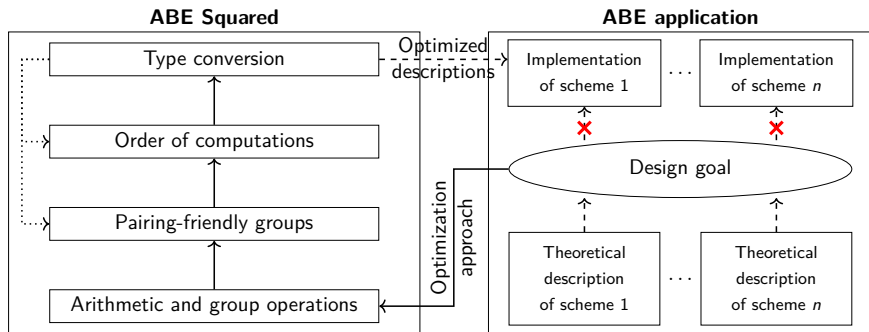
# ABE Squared

- ▶ ABE Squared is a framework for accurately benchmarking efficiency of ABE
- ▶ Four layers of optimization

# ABE Squared

- ▶ ABE Squared is a framework for accurately benchmarking efficiency of ABE
- ▶ Four layers of optimization
- ▶ Take as input a **design goal**
- ▶ Design goal specifies the computational needs of the application in which the scheme will be used
- ▶ For example, cloud settings may require an optimized decryption efficiency, while IoT requires an optimized encryption efficiency

# Overview of ABE Squared



The arrows have the following meaning:

$a \longrightarrow b = \text{"a influences b"}$

$a \cdots \rightarrow b = \text{"a may require adjustment in b"}$

$a \dashrightarrow b = \text{"a is input to b" / "b is output of a"}$

# Illustrating the framework

- ▶ To illustrate the framework, we have implemented<sup>1</sup> Wat11-IV, RW13 and AC17-LU
- ▶ Three design goals:
  - ▶ Optimized encryption
  - ▶ Optimized decryption
  - ▶ Optimized key generation

---

<sup>1</sup>see [https://github.com/abecryptools/abe\\_squared](https://github.com/abecryptools/abe_squared)



# Illustrating the framework

- ▶ To illustrate the framework, we have implemented<sup>1</sup> Wat11-IV, RW13 and AC17-LU
- ▶ Three design goals:
  - ▶ Optimized encryption
  - ▶ Optimized decryption
  - ▶ Optimized key generation
- ▶ Common thought: RW13 is very inefficient compared to Wat11-IV (and AC17-LU) in all algorithms
- ▶ Presumably, the reason why Wat11-IV is preferred over RW13 despite not being able to support NOT operators

---

<sup>1</sup>see [https://github.com/abecryptools/abe\\_squared](https://github.com/abecryptools/abe_squared)

# Benchmarks

Implementation of Wat11-IV, RW13 and AC17-LU, based on their optimization approaches<sup>1</sup> (OA). The costs are for 100 attributes, expressed in  $10^3$  clock cycles<sup>2</sup>.

OA	Scheme	Curve	Key generation		Encryption		Decryption	
			Costs	Increase %	Costs	Increase %	Costs	Increase %
OE	Wat11-IV	BLS12-381	42275	0.2%	77641	48.8%	58290	543.4%
	RW13	BLS12-381	51401	21.8%	54491	4.4%	112072	1137.1%
	AC17-LU	BLS12-381	<b>42196</b>	-	<b>52176</b>	-	<b>9060</b>	-
OK	Wat11-IV	BLS12-381	42135	94.6%	77898	48.9%	58441	543.9%
	RW13	BLS12-381	<b>21657</b>	-	128221	145.0%	118998	1211.2%
	AC17-LU	BLS12-381	41913	93.5%	<b>52326</b>	-	<b>9076</b>	-
OD	Wat11-IV	BLS12-381	<b>42275</b>	-	77641	42.5%	58290	1336.5%
	RW13	BLS12-381	51401	21.6%	<b>54491</b>	-	112072	2661.9%
	AC17-LU	BN382	45093	6.7%	59276	8.8%	<b>4058</b>	-

<sup>1</sup> OE/OD/OK = optimized encryption/decryption/key generation.

<sup>2</sup> AMD Ryzen 7 PRO 4750 processor, one single core at 4.1 GHz.

# Takeaways

- ▶ All algorithms take at most 29 milliseconds on the target device

# Takeaways

- ▶ All algorithms take at most 29 milliseconds on the target device
- ▶ For an optimized encryption or decryption, use AC17-LU
- ▶ For an optimized key generation, use RW13

# Takeaways

- ▶ All algorithms take at most 29 milliseconds on the target device
- ▶ For an optimized encryption or decryption, use AC17-LU
- ▶ For an optimized key generation, use RW13
- ▶ **Surprising result:** RW13 outperforms Wat11-IV in the key generation and encryption algorithms

# Takeaways

- ▶ All algorithms take at most 29 milliseconds on the target device
- ▶ For an optimized encryption or decryption, use AC17-LU
- ▶ For an optimized key generation, use RW13
- ▶ **Surprising result:** RW13 outperforms Wat11-IV in the key generation and encryption algorithms
- ▶ RW13 only scheme that can be extended to support NOT operators
- ▶ Supporting NOT = inherently inefficient decryption?

# Takeaways

- ▶ All algorithms take at most 29 milliseconds on the target device
- ▶ For an optimized encryption or decryption, use AC17-LU
- ▶ For an optimized key generation, use RW13
- ▶ **Surprising result:** RW13 outperforms Wat11-IV in the key generation and encryption algorithms
- ▶ RW13 only scheme that can be extended to support NOT operators
- ▶ Supporting NOT = inherently inefficient decryption?
- ▶ Furthermore, these numbers are acceptable for fast devices (computer, smartphone)
- ▶ Not for IoT: seconds instead of milliseconds

# High-level overview

- 1 Introduction to ABE
- 2 The pair encodings framework
- 3 ABE Squared
- 4 New schemes**
- 5 Conclusions



# New schemes

- ▶ Two new schemes:
  - ▶ GLUE [VA22b]
  - ▶ TinyABE [VA22a]
- ▶ Both schemes are generalizations of RW13

# New schemes

- ▶ Two new schemes:
  - ▶ GLUE [VA22b]
  - ▶ TinyABE [VA22a]
- ▶ Both schemes are generalizations of RW13
- ▶ Both schemes have a flexible efficiency trade-off
- ▶ Can be configured during the setup, taking into account the computational devices

# New schemes

- ▶ Two new schemes:
  - ▶ GLUE [VA22b]
  - ▶ TinyABE [VA22a]
- ▶ Both schemes are generalizations of RW13
- ▶ Both schemes have a flexible efficiency trade-off
- ▶ Can be configured during the setup, taking into account the computational devices
- ▶ GLUE's trade-off is between the encryption and decryption costs
- ▶ TinyABE's trade-off is between the master public key and ciphertext sizes → designed for IoT

# New schemes

- ▶ Two new schemes:
  - ▶ GLUE [VA22b]
  - ▶ TinyABE [VA22a]
- ▶ Both schemes are generalizations of RW13
- ▶ Both schemes have a flexible efficiency trade-off
- ▶ Can be configured during the setup, taking into account the computational devices
- ▶ GLUE's trade-off is between the encryption and decryption costs
- ▶ TinyABE's trade-off is between the master public key and ciphertext sizes → designed for IoT
- ▶ I will discuss GLUE in more detail

# GLUE

GLUE addresses the need for a scheme that

- ▶ supports NOT operators
- ▶ has an efficient decryption

# GLUE

GLUE addresses the need for a scheme that

- ▶ supports NOT operators
- ▶ has an efficient decryption

GLUE generalizes the 1-degree polynomial of RW13 to an  $n$ -degree polynomial, where  $n = n_k + n_c - 1$ .

# GLUE

GLUE addresses the need for a scheme that

- ▶ supports NOT operators
- ▶ has an efficient decryption

GLUE generalizes the 1-degree polynomial of RW13 to an  $n$ -degree polynomial, where  $n = n_k + n_c - 1$ .

→ partition the sets and policies in smaller subsets of maximum size  $n_k$  and  $n_c$ , respectively.

# GLUE

GLUE addresses the need for a scheme that

- ▶ supports NOT operators
- ▶ has an efficient decryption

GLUE generalizes the 1-degree polynomial of RW13 to an  $n$ -degree polynomial, where  $n = n_k + n_c - 1$ .

→ partition the sets and policies in smaller subsets of maximum size  $n_k$  and  $n_c$ , respectively.

→ number of pairings needed during decryption can be reduced by a factor in  $n_k$  and  $n_c$ , e.g., a factor of  $n_k$  if  $n_k = n_c$ .



# GLUE

GLUE addresses the need for a scheme that

- ▶ supports NOT operators
- ▶ has an efficient decryption

GLUE generalizes the 1-degree polynomial of RW13 to an  $n$ -degree polynomial, where  $n = n_k + n_c - 1$ .

→ partition the sets and policies in smaller subsets of maximum size  $n_k$  and  $n_c$ , respectively.

→ number of pairings needed during decryption can be reduced by a factor in  $n_k$  and  $n_c$ , e.g., a factor of  $n_k$  if  $n_k = n_c$ .

The higher  $n_k$  and  $n_c$ , the more efficient decryption is.

## Performance estimates

Rough estimates<sup>2</sup> of the storage costs of the secret keys and the ciphertexts in kilobytes (KB), where 1 KB = 1024 bytes, and the computational costs incurred by the key generation, encryption and decryption algorithms of  $\text{GLUE}_{(n_k, n_c)}$  and RW13, expressed in milliseconds (ms), for 10 and 100 attributes.

Scheme	Storage costs					Computational costs					
	MPK	SK		CT		KeyGen		Encrypt		Decrypt	
		10	100	10	100	10	100	10	100	10	100
RW13	1.42	4.86	44.58	4.05	33.58	26.0	238.7	32.9	305.9	46.2	375.2
$\text{GLUE}_{(3,3)}$	2.08	3.53	30.02	3.39	26.36	18.9	160.7	59.8	571.4	24.3	133.9
$\text{GLUE}_{(5,5)}$	2.74	3.09	26.93	3.17	24.83	16.5	144.2	82.3	800.4	17.0	82.8
$\text{GLUE}_{(10,5)}$	3.28	2.87	24.72	3.17	24.83	15.4	132.3	102.1	998.4	15.1	64.5

<sup>2</sup>On a 1.6 GHz Intel i5-8250U processor for the BLS12-446 curve

# Variants supporting NOT operators

- ▶ We prove security in the pair encodings framework
- ▶ Specifically, the algebraic notion [AC17b]

# Variants supporting NOT operators

- ▶ We prove security in the pair encodings framework
- ▶ Specifically, the algebraic notion [AC17b]
- ▶ RW13 and GLUE can be transformed to support NOT operators with [Att19, Amb21]

# Variants supporting NOT operators

- ▶ We prove security in the pair encodings framework
- ▶ Specifically, the algebraic notion [AC17b]
- ▶ RW13 and GLUE can be transformed to support NOT operators with [Att19, Amb21]
- ▶ Variant of RW13 with NOT operators: Att19-I-CP

## Performance estimates for variants with NOTs

Rough estimates<sup>3</sup> of the storage costs of the secret keys and the ciphertexts in kilobytes (KB), where 1 KB = 1024 bytes, and the computational costs incurred by the key generation, encryption and decryption algorithms of  $\text{GLUE-N}_{(n_k, n_c)}$  and  $\text{Att19-I-CP}$ , expressed in milliseconds (ms), for 10 and 100 attributes.

Scheme	Storage costs					Computational costs					
	MPK	SK		CT		KeyGen		Encrypt		Decrypt	
		10	100	10	100	10	100	10	100	10	100
Att19-I-CP	1.4	10.9	100.0	6.4	55.8	59.0	541.8	66.6	637.5	51.7-216.2	367.9-1779.0
$\text{GLUE-N}_{(3,3)}$	2.1	7.6	63.7	5.0	41.2	44.8	385.9	90.0	865.1	29.8-109.5	139.4-745.5
$\text{GLUE-N}_{(5,5)}$	2.7	6.5	56.0	4.6	38.1	40.1	352.8	111.4	1086.0	22.4-55.3	88.3-382.5
$\text{GLUE-N}_{(10,5)}$	3.3	5.9	50.4	4.6	38.1	37.7	329.2	131.2	1284.0	20.6-78.5	70.0-614.1

<sup>3</sup>On a 1.6 GHz Intel i5-8250U processor for the BLS12-446 curve

# Optimization via online/offline versions

- ▶ Main disadvantage of GLUE: encryption costs increase significantly

# Optimization via online/offline versions

- ▶ Main disadvantage of GLUE: encryption costs increase significantly
- ▶ Online/offline version of RW13 [HW14]
- ▶ Also applies to generalizations of RW13, e.g., GLUE



# Optimization via online/offline versions

- ▶ Main disadvantage of GLUE: encryption costs increase significantly
- ▶ Online/offline version of RW13 [HW14]
- ▶ Also applies to generalizations of RW13, e.g., GLUE
- ▶ High-level idea: precompute the ciphertexts
- ▶ Online execution time minimal: only some simple additions and multiplications

# Optimization via online/offline versions

- ▶ Main disadvantage of GLUE: encryption costs increase significantly
- ▶ Online/offline version of RW13 [HW14]
- ▶ Also applies to generalizations of RW13, e.g., GLUE
- ▶ High-level idea: precompute the ciphertexts
- ▶ Online execution time minimal: only some simple additions and multiplications
- ▶ Trade-off: larger ciphertexts, i.e., factor of 2-3
- ▶ Acceptable for many settings in which decryption needs to be fast, e.g., cloud

# High-level overview

- 1 Introduction to ABE
- 2 The pair encodings framework
- 3 ABE Squared
- 4 New schemes
- 5 Conclusions**

# Conclusions

- ▶ ABE implements access control on a cryptographic level
- ▶ Interesting for various use cases, e.g., cloud, IoT

# Conclusions

- ▶ ABE implements access control on a cryptographic level
- ▶ Interesting for various use cases, e.g., cloud, IoT
- ▶ Pairing-based ABE can support many desirable properties
- ▶ Still much to do in terms of efficiency

# Conclusions

- ▶ ABE implements access control on a cryptographic level
- ▶ Interesting for various use cases, e.g., cloud, IoT
- ▶ Pairing-based ABE can support many desirable properties
- ▶ Still much to do in terms of efficiency
- ▶ To benchmark and compare more fairly: ABE Squared
- ▶ Scheme supporting NOT operators with efficient decryption: GLUE

Thank you for your attention!

# References I

- [ABGW17] M. Ambrona, G. Barthe, R. Gay, and H. Wee.  
Attribute-based encryption in the generic group model: Automated proofs and new constructions.  
In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *CCS*, pages 647–664. ACM, 2017.
- [AC17a] S. Agrawal and M. Chase.  
FAME: fast attribute-based message encryption.  
In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *CCS*, pages 665–682. ACM, 2017.
- [AC17b] S. Agrawal and M. Chase.  
Simplifying design and analysis of complex predicate encryption schemes.  
In J.-S. Coron and J. B. Nielsen, editors, *EUROCRYPT*, volume 10210 of *LNCS*, pages 627–656. Springer, 2017.
- [AHM<sup>+</sup>16] N. Attrapadung, G. Hanaoka, T. Matsumoto, T. Teruya, and S. Yamada.  
Attribute based encryption with direct efficiency tradeoff.  
In M. Manulis, A.-R. Sadeghi, and S. A. Schneider, editors, *ACNS*, volume 9696 of *LNCS*, pages 249–266. Springer, 2016.
- [ALdP11] N. Attrapadung, B. Libert, and E. de Panafieu.  
Expressive key-policy attribute-based encryption with constant-size ciphertexts.  
In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC*, volume 6571 of *LNCS*, pages 90–108. Springer, 2011.
- [Amb21] M. Ambrona.  
Generic negation of pair encodings.  
In J. A. Garay, editor, *PKC*, volume 12711 of *LNCS*, pages 120–146. Springer, 2021.
- [Att14] N. Attrapadung.  
Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more.  
In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT*, volume 8441 of *LNCS*, pages 557–577. Springer, 2014.



# References II

- [Att19] N. Attrapadung.  
Unbounded dynamic predicate compositions in attribute-based encryption.  
In Y. Ishai and V. Rijmen, editors, *EUROCRYPT*, volume 11476 of *LNCS*, pages 34–67. Springer, 2019.
- [BSW07] J. Bethencourt, A. Sahai, and B. Waters.  
Ciphertext-policy attribute-based encryption.  
In *S&P*, pages 321–334. IEEE, 2007.
- [CC09] M. Chase and S. S. M. Chow.  
Improving privacy and security in multi-authority attribute-based encryption.  
In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *CCS*, pages 121–130. ACM, 2009.
- [CCL<sup>+</sup>13] C. Chen, J. Chen, H. W. Lim, Z. Zhang, D. Feng, S. Ling, and H. Wang.  
Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures.  
In *CT-RSA*, volume 7779 of *LNCS*, pages 50–67. Springer, 2013.
- [CDLQ16] H. Cui, R. H. Deng, Y. Li, and B. Qin.  
Server-aided revocable attribute-based encryption.  
In I. G. Askoxylakis, S. Ioannidis, S. K. Katsikas, and C. A. Meadows, editors, *ESORICS*, volume 9879 of *LNCS*, pages 570–587. Springer, 2016.
- [CGKW18] J. Chen, J. Gong, L. Kowalczyk, and H. Wee.  
Unbounded ABE via bilinear entropy expansion, revisited.  
In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT*, volume 10820 of *LNCS*, pages 503–534. Springer, 2018.
- [CGW15] J. Chen, R. Gay, and H. Wee.  
Improved dual system ABE in prime-order groups via predicate encodings.  
In E. Oswald and M. Fischlin, editors, *EUROCRYPT*, volume 9057 of *LNCS*, pages 595–624. Springer, 2015.

# References III

- [Cha07] M. Chase.  
Multi-authority attribute-based encryption.  
In S. P. Vadhan, editor, *TCC*, volume 4392 of *LNCS*, pages 515–534. Springer, 2007.
- [CN07] L. Cheung and C. C. Newport.  
Provably secure ciphertext policy ABE.  
In P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, editors, *CCS*, pages 456–465. ACM, 2007.
- [dIPVA22] A. de la Piedra, M. Venema, and G. Alpár.  
ABE squared: Accurately benchmarking efficiency of attribute-based encryption.  
*TCHES*, 2022(2):192–239, 2022.
- [GHW11] M. Green, S. Hohenberger, and B. Waters.  
Outsourcing the decryption of ABE ciphertexts.  
In *USENIX Security Symposium*. USENIX Association, 2011.
- [GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters.  
Attribute-based encryption for fine-grained access control of encrypted data.  
In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *CCS*. ACM, 2006.
- [HLR10] J. Herranz, F. Laguillaumie, and C. Ràfols.  
Constant size ciphertexts in threshold attribute-based encryption.  
In P. Q. Nguyen and D. Pointcheval, editors, *PKC*, volume 6056 of *LNCS*, pages 19–34. Springer, 2010.
- [HW13] S. Hohenberger and B. Waters.  
Attribute-based encryption with fast decryption.  
In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC*, volume 7778 of *LNCS*, pages 162–179. Springer, 2013.

## References IV

- [HW14] S. Hohenberger and B. Waters.  
Online/offline attribute-based encryption.  
In Hugo Krawczyk, editor, *PKC*, volume 8383 of *LNCS*, pages 293–310. Springer, 2014.
- [KL15] L. Kowalczyk and A. Lewko.  
Bilinear entropy expansion from the decisional linear assumption.  
In R. Gennaro and M. Robshaw, editors, *CRYPTO*, volume 9216 of *LNCS*, pages 524–541. Springer, 2015.
- [KW19] L. Kowalczyk and H. Wee.  
Compact adaptively secure ABE for NC1 from k-lin.  
In Y. Ishai and V. Rijmen, editors, *EUROCRYPT*, volume 11476 of *LNCS*, pages 3–33. Springer, 2019.
- [LCH<sup>+</sup>11] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen.  
Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles.  
In V. Atluri and C. Díaz, editors, *ESORICS*, volume 6879 of *LNCS*, pages 278–297. Springer, 2011.
- [LCL<sup>+</sup>13] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou.  
Fine-grained access control system based on outsourced attribute-based encryption.  
In J. Crampton, S. Jajodia, and K. Mayes, editors, *ESORICS*, volume 8134 of *LNCS*, pages 592–609. Springer, 2013.
- [LCW13] Z. Liu, Z. Cao, and D. S. Wong.  
Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay.  
In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *CCS*, pages 475–486. ACM, 2013.
- [LHC<sup>+</sup>11] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie.  
Multi-authority ciphertext-policy attribute-based encryption with accountability.  
In B. S. N. Cheung, L. Chi Kwong Hui, R. S. Sandhu, and D. S. Wong, editors, *ASIACCS*, pages 386–390. ACM, 2011.

# References V

- [LOS<sup>+</sup>10] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters.  
Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption.  
In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *LNCS*, pages 62–91. Springer, 2010.
- [LW11a] A. Lewko and B. Waters.  
Decentralizing attribute-based encryption.  
In *EUROCRYPT*, pages 568–588. Springer, 2011.
- [LW11b] A. B. Lewko and B. Waters.  
Unbounded HIBE and attribute-based encryption.  
In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *LNCS*, pages 547–567. Springer, 2011.
- [LW12] A. B. Lewko and B. Waters.  
New proof methods for attribute-based encryption: Achieving full security through selective techniques.  
In *CRYPTO*, pages 180–198. Springer, 2012.
- [LW15] Z. Liu and D. S. Wong.  
Practical ciphertext-policy attribute-based encryption: Traitor tracing, revocation, and large universe.  
In T. Malkin, V. Kolesnikov, A. Lewko, and M. Polychronakis, editors, *ACNS*, volume 9092 of *LNCS*, pages 127–146. Springer, 2015.
- [LYZL18] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang.  
Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list.  
In B. Preneel and F. Vercauteren, editors, *ACNS*, volume 10892 of *LNCS*, pages 516–534. Springer, 2018.
- [MJ18] Y. Michalevsky and M. Joye.  
Decentralized policy-hiding ABE with receiver privacy.  
In J. López, J. Zhou, and M. Soriano, editors, *ESORICS*, volume 11099 of *LNCS*, pages 548–567. Springer, 2018.

# References VI

- [NYO08] T. Nishide, K. Yoneyama, and K. Ohta.  
Attribute-based encryption with partially hidden encryptor-specified access structures.  
In S. M. Bellare, R. Gennaro, A. D. Keromytis, and M. Yung, editors, *ACNS*, volume 5037 of *LNCS*, pages 111–129, 2008.
- [OSW07] R. Ostrovsky, A. Sahai, and B. Waters.  
Attribute-based encryption with non-monotonic access structures.  
In *CCS*, pages 195–203. ACM, 2007.
- [OT10] T. Okamoto and K. Takashima.  
Fully secure functional encryption with general relations from the decisional linear assumption.  
In T. Rabin, editor, *CRYPTO*, volume 6223 of *LNCS*, pages 191–208. Springer, 2010.
- [OT12] T. Okamoto and K. Takashima.  
Fully secure unbounded inner-product and attribute-based encryption.  
In *ASIACRYPT*, pages 349–366. Springer, 2012.
- [OT13] T. Okamoto and K. Takashima.  
Decentralized attribute-based signatures.  
In K. Kurosawa and G. Hanaoka, editors, *PKC*, volume 7778 of *LNCS*, pages 125–142. Springer, 2013.
- [RW13] Y. Rouselakis and B. Waters.  
Practical constructions and new proof methods for large universe attribute-based encryption.  
In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *CCS*, pages 463–474. ACM, 2013.
- [RW15] Y. Rouselakis and B. Waters.  
Efficient statically-secure large-universe multi-authority attribute-based encryption.  
In R. Böhme and T. Okamoto, editors, *FC*, volume 8975 of *LNCS*, pages 315–332. Springer, 2015.

# References VII

- [SSW12] A. Sahai, H. Seyalioglu, and B. Waters.  
Dynamic credentials and ciphertext delegation for attribute-based encryption.  
In *CRYPTO*, pages 199–217. Springer, 2012.
- [SW05] A. Sahai and B. Waters.  
Fuzzy identity-based encryption.  
In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.
- [TKN20] J. Tomida, Y. Kawahara, and R. Nishimaki.  
Fast, compact, and expressive attribute-based encryption.  
In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *PKC*, volume 12110 of *LNCS*, pages 3–33. Springer, 2020.
- [VA21] M. Venema and G. Alpár.  
A bunch of broken schemes: A simple yet powerful linear approach to analyzing security of attribute-based encryption.  
In K. G. Paterson, editor, *CT-RSA*, volume 12704 of *LNCS*, pages 100–125. Springer, 2021.
- [VA22a] M. Venema and G. Alpár.  
TinyABE: Unrestricted ciphertext-policy attribute-based encryption for embedded devices and low-quality networks.  
In L. Batina and J. Daemen, editors, *AFRICACRYPT*, volume 13503 of *LNCS*, pages 103–129. Springer, 2022.
- [VA22b] M. Venema and G. Alpár.  
GLUE: Generalizing unbounded attribute-based encryption for flexible efficiency trade-offs.  
*Cryptology ePrint Archive*, Paper 2022/613, 2022.
- [VAH22] M. Venema, G. Alpár, and J.-H. Hoepman.  
Systematizing core properties of pairing-based attribute-based encryption to uncover remaining challenges in enforcing access control in practice.  
In *To appear at Des. Codes Cryptogr.*, 2022.

# References VIII

- [VB22] M. Venema and L. Botros.  
Efficient and generic transformations for chosen-ciphertext secure predicate encryption.  
*Cryptology ePrint Archive, Paper 2022/1436*, 2022.
- [Wat08] B. Waters.  
Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.  
*Cryptology ePrint Archive, Report 2008/290*, 2008.
- [Wat11] B. Waters.  
Ciphertext-policy attribute-based encryption - an expressive, efficient, and provably secure realization.  
In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC*, volume 6571 of *LNCS*, pages 53–70. Springer, 2011.
- [Wee14] H. Wee.  
Dual system encryption via predicate encodings.  
In Y. Lindell, editor, *TCC*, volume 8349 of *LNCS*, pages 616–637. Springer, 2014.
- [YWRL10] S. Yu, C. Wang, K. Ren, and W. Lou.  
Attribute based data sharing with attribute revocation.  
In D. Feng, D. A. Basin, and P. Liu, editors, *ASIACCS*, pages 261–270. ACM, 2010.
- [ZGT<sup>+</sup>16] K. Zhang, J. Gong, S. Tang, J. Chen, X. Li, H. Qian, and Z. Cao.  
Practical and efficient attribute-based encryption with constant-size ciphertexts in outsourced verifiable computation.  
In X. Chen, X. Wang, and X. Huang, editors, *ASIACCS*, pages 269–279. ACM, 2016.