

Practical Identity-Based Encryption for Online Social Networks

Stijn Meul

Thesis submitted for the degree of
Master of Science in
Electrical Engineering, option
Embedded Systems and Multimedia

Thesis supervisors:

Prof. dr. ir. Bart Preneel
Prof. dr. ir. Vincent Rijmen

Assessors:

Prof. dr. ir. Claudia Diaz
Prof. dr. ir. Frank Piessens

Mentor:

Filipe Beato

© Copyright KU Leuven

Without written permission of the thesis supervisors and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to Departement Elektrotechniek, Kasteelpark Arenberg 10 postbus 2440, B-3001 Heverlee, +32-16-321130 or by email info@esat.kuleuven.be.

A written permission of the thesis supervisors is also required to use the methods, products, schematics and programs described in this work for industrial or commercial use, and for submitting this publication in scientific contests.



Preface

I would like to thank everybody who kept me busy the last year, especially my promotor and my assistants. I would also like to thank the jury for reading the text. My sincere gratitude also goes to my wife and the rest of my family.

Stijn Meul

Contents

Preface	i
Abstract	iv
List of Figures and Tables	v
List of Abbreviations	vi
List of Symbols	vii
1 Introduction	1
1.1 Problem Statement	1
1.2 Previous Work	1
1.3 Goals of this Thesis	1
1.4 Structure of this Thesis	1
2 Background	3
2.1 Negligible Function	3
2.2 Abstract Algebra	4
2.3 Number Theoretic Assumptions	7
2.4 Bilinear Maps	9
2.5 Hash Functions	10
2.6 Summary	12
3 Literature Review	13
3.1 Asymmetric Cryptography	13
3.2 Identity-Based Encryption	15
3.3 Broadcast Encryption	22
3.4 Secret Sharing	27
3.5 Distributed Key Generation	29
4 Outsider Anonymous Identity-Based Broadcast Encryption	33
4.1 Security Model	33
4.2 Proposed Scheme	33
4.3 Conclusion	33
5 Implementation	35

5.1 Existing Solutions	35
5.2 Anonymous Identity-Based Broadcasting Implementation	35
5.3 Distributed Key Generation Implementation	35
5.4 Evaluation	35
5.5 Performance Analysis	35
5.6 Conclusion	35
6 Conclusion	37
A Installing and Executing the Code	41
A.1 Setting up the DKG	41
A.2 Setting up Scramble	41
B The Last Appendix	43
B.1 Lorem 20-24	43
B.2 Lorem 25-27	43
Bibliography	45



Abstract

The **abstract** environment contains a more extensive overview of the work. But it should be limited to one page.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



List of Figures and Tables

List of Figures

3.1 Generic identity-based encryption scheme. The orange arrow denotes an insecure channel that can be eavesdropped. 16

List of Tables



List of Abbreviations

IBE	Identity-Based Encryption
PKG	Public Key Generator
DKG	Distributed Key Generator
IND-CPA	Indistinguishability under Chosen Plaintext Attack
IND-CCA	Indistinguishability under Chosen Ciphertext Attack
ANO-IBE	Anonymous IBE
ANO-IND-CPA	Anonymity preserving IBE scheme that is indistinguishable under chosen plaintext attacks
ANO-IND-CCA	Anonymity preserving IBE scheme that is indistinguishable under chosen ciphertext attacks
OSN	Online Social Network

List of Symbols

λ	Security parameter denoting the level of security
l	The number of bits required to realise security level λ
s	A secret
sk_i	The secret key corresponding to the public key pk_i
pk_i	A public key with corresponding secret key sk_i
m	A bit sequence representing a message
c	A bit sequence representing a ciphertext
v, w	Small letters represent binary bit sequences
id_{Alice}	A bit sequence uniquely identifying Alice
$s\text{id}_{\text{Alice}}$	The corresponding IBE secret key to the identifier id_{Alice} serving as an IBE public key
k	A symmetric session key
$E_k(m)$	Symmetric encryption of the message m under session key k
$D_k(c)$	Symmetric decryption of the ciphertext c under session key k
G	A group $(G, *)$
$S_A(m)$	Signature of entity A on message m
$e : G_1 \times G_2 \rightarrow G_T$	Admissible bilinear map
U, P, Q	Points on an elliptic curve
$e(P, Q)$	Admissible bilinear map $P \in G_1, Q \in G_2, e(P, Q) \in G_T$
$\mathcal{A}(a, b)$	An algorithm \mathcal{A} with parameters a and b
$\langle a, b, c \rangle \leftarrow \mathcal{A}(d, e)$	Algorithm \mathcal{A} with parameters d and e returns the collection of values a, b, c

Introduction

The newest internet trend at the dawn of the 21st century certainly is the Online Social Network (OSN). Words like tweeting, sharing, liking, trending and tagging have found common acceptance in the vocabulary of today's internet savvy users while services like Facebook, Google+, LinkedIn and Twitter have become part of everyday life.

The far reaching influence of today's most popular OSNs is best illustrated with the help of some statistics. In May 2013, 72% of all internet users were active on a social network [53]. At the time of writing, Facebook has 1.23 Billion monthly active users which corresponds to 17% of the global population [23, 81]. Furthermore, the average Facebook user spends 15 hours and 33 minutes online per month [76]. These numbers show that social networks no longer represent the latest craze of an internet bubble but are conversely deeply rooted in our daily habits.

1.1 Problem Statement

1.2 Previous Work

1.3 Goals of this Thesis

1.4 Structure of this Thesis

Background

This chapter briefly covers the mathematical knowledge required to understand cryptographic algorithms presented later in this text. Although understanding all mathematical details of this chapter can be quite a struggle, the math serves as a fundament of a challenging world containing exciting cryptographic concepts like identity-based encryption.

First, the notion of negligible functions will be introduced followed by an overview of algebraic structures and their properties. Then, a number of theoretic assumptions fundamental for cryptographic security are presented. By exploring these variants of the Diffie-Hellman assumption, the introduction of gap groups and bilinear maps follows naturally. Finally, hash functions are defined as well as their relation to the random oracle assumption.

Note that this chapter only scratches the surface of cryptographic fundamentals required to understand the remainder of the thesis. Definitions and theorems are always provided without proof. For a more in depth discussion about algebraic topics in this chapter, the reader is referred to [65] and [12]. More information on elliptic curves, Diffie-Hellman assumptions and pairing based cryptography can be found in [1].

If the reader feels he has sufficient background of the concepts covered in this chapter, the chapter can be skipped without loss of comprehension.

2.1 Negligible Function

In practice no modern cryptographic algorithm achieves perfect secrecy¹, i.e. with unbounded computational power all practical cryptographic algorithms can be broken. Therefore a more pragmatic definition of security is always considered, namely security against adversaries that are computationally bound to their finite resources. In this pragmatic view of security an algorithm is considered secure only if the probability of success is smaller than the reciprocal of any polynomial function. The negligible function can be used to exactly describe this notion in a formal way.

¹Note that the one-time pad is not taken into account. Although it is the only proven information secure cryptographic algorithm, it is seldom used in practical cryptographic systems.

Definition 2.1. A **negligible function** in k is a function $\mu(k) : \mathbb{N} \rightarrow \mathbb{R}$ if for every polynomial $p(\cdot)$ there exists an N such that for all $k > N$ [45]

$$\mu(k) < \frac{1}{p(k)}$$

The negligible function will be used later on in this chapter to formally describe computationally infeasible problems. In such a context k often represents the security parameter. The larger k will be chosen, the smaller $\mu(k)$ will be.

2.2 Abstract Algebra

Abstract algebra is a field of mathematics that studies algebraic structures such as groups, rings and vector spaces. These algebraic structures define a collection of requirements on mathematical sets such as e.g., the natural numbers \mathbb{N} or matrices of dimension 2×2 $\mathbb{R}^{2 \times 2}$. If these requirements hold, abstract properties can be derived. Once a mathematical set is then categorised as the correct algebraic structure, properties derived for the algebraic structure will hold for the set as a whole.

In the light of our further discussion, especially additive and multiplicative groups prove to be essential concepts. However, algebraic groups come with a specific vocabulary such as binary operation, group order and cyclic group that are defined in this section as well.

Definition 2.2 (Binary operation). A *binary operation* $*$ on a set S is a mapping $S \times S \rightarrow S$. That is, a binary operation is a rule which assigns to each ordered pair of elements a and b from S a uniquely defined third element $c = a * b$ in the same set S . [65, 12]

Definition 2.3 (Group). A *group* $(G, *)$ consists of a set G with a binary operation $*$ on G satisfying the following three axioms:

1. *Associativity* $\forall a, b, c \in G : a * (b * c) = (a * b) * c$
2. *Identity element* $\forall a \in G, \exists e \in G : a * e = e * a = a$ where e denotes the *identity element* of G
3. *Inverse element* $\forall a \in G, \exists a^{-1} : a * a^{-1} = a^{-1} * a = 1$ where a^{-1} denotes the *inverse element* of a

Definition 2.4 (Commutative group). A group $(G, *)$ is called a *commutative group* or an *abelian group* if in addition to the properties in Definition 2.3, also commutativity holds.

4. **Commutativity** $\forall a, b \in G : a * b = b * a$

Depending on the group operation $*$, $(G, *)$ is called either a *multiplicative group* or an *additive group*. In Definition 2.3 the multiplicative notation is used. For an additive group the inverse of a is often denoted $-a$ [65].

A group $(G, *)$ is often denoted by the more concise symbol G although groups are always defined with respect to a binary group operation $*$. Despite of a more concise notation, any group G still obeys all axioms from Definition 2.3 with respect to an implicitly known group operation $*$.

A perfect example of a commutative group is the set of integers with the addition operation $(\mathbb{Z}, +)$ since the addition is both associative and commutative in \mathbb{Z} . Furthermore, the identity element $e = 0$ and the inverse element $\forall a \in \mathbb{Z}$ is $-a \in \mathbb{Z}$. Note that the set of natural numbers with the addition operation $(\mathbb{N}, +)$ is not a commutative group as not every element of \mathbb{N} has an inverse element.

Definition 2.5 (Cyclic group). A group G is *cyclic* if and only if $\forall b \in G, \exists g \in G, \exists n \in \mathbb{Z} : g^n = b$. Such an element g is called a **generator** of G .

Definition 2.5 implies that in a cyclic group every element can be written as a power of one of the group's generators.

Definition 2.6 (Finite group). A group G is *finite* if the number of elements in G denoted $|G|$ is finite. The number of elements $|G|$ in a finite group is called the *group order*.

The set \mathbb{Z}_n denotes the set of integers modulo n . The set \mathbb{Z}_5 with the addition operation is a cyclic finite group of order 5. The set $\mathbb{Z}_5 \setminus \{0\}$ with the multiplication operation, often denoted \mathbb{Z}_5^* , is a cyclic finite group of order 4 where the neutral element $e = 1$. Two is an example of a generator in \mathbb{Z}_5^* since every element in \mathbb{Z}_5^* can be written as $\{2^n | n \in \mathbb{Z}\}$.

Definition 2.7 (Order of an element). Let G be a group. The *order of an element* $a \in G$ is defined as the least positive integer t such that $a^t = e$. If there exists no such t , t is defined as ∞ .

Theorem 2.8. *If the order of a group G equals a prime p , the group is cyclic and commutative.*

Definition 2.9 (Subgroup). Given a group $(G, *)$, any H that is a non-empty subset $H \subseteq G$ and satisfies the axioms of a group with respect to the group operation $*$ in H , is a *subgroup* of G .

Definition 2.10 (Ring). A *ring* $(R, +, *)$ consists of a set R with two binary operations $+$ and $*$ on R satisfying the following axioms:

1. $(R, +)$ is an abelian group with identity denoted e
2. *Associativity* $\forall a, b, c \in R : a * (b * c) = (a * b) * c$
3. *Multiplicative identity element* $\forall a \in R, \exists 1 \in R : a * 1 = 1 * a = a$ where 1 denotes the *multiplicative identity element* of R
4. *Left distributivity* $\forall a, b, c \in R : a * (b + c) = (a * b) + (a * c)$

5. *Right distributivity* $\forall a, b, c \in R : (b + c) * a = (b * a) + (c * a)$

Definition 2.11 (Commutative ring). A ring $(R, +, *)$ is called a *commutative ring* or an *abelian ring* if in addition to the properties in Definition 2.10, also commutativity holds.

6. **Commutativity** $\forall a, b \in R : a * b = b * a$

Definition 2.12 (Field). A commutative ring $(R, +, *)$ is called a *field* if in addition to the properties in Definition 2.11 and Definition 2.10 all elements of R have a multiplicative inverse.

7. *Multiplicative inverse* $\forall a \in R, \exists a^{-1} : a * a^{-1} = a^{-1} * a = 1$ where a^{-1} denotes the *inverse element* of a

Definition 2.13 (Finite field). A *finite field* or a *Galois Field* is a field F with a finite number of elements. The number of elements $|F|$ of a finite field F is called its *order*.

Definition 2.14 (Ring homomorphism). Given rings R and S , a *ring homomorphism* is a function $f : R \rightarrow S$ such that the following axioms hold:

1. $\forall a, b \in R : f(a + b) = f(a) + f(b)$
2. $\forall a, b \in R : f(ab) = f(a)f(b)$
3. $f(e_R) = f(e_S)$ where e_S and e_R denote the identity element of respectively S and R

Definition 2.15 (Bijective function). Any function $f : R \rightarrow S$ is bijective if it satisfies the following axioms

1. *Injective* Each element in S is the image of at most one element in R . Hence, $\forall a_1, a_2 \in R$ if $f(a_1) = f(a_2)$ then $a_1 = a_2$ follows naturally.
2. *Surjective* Each $s \in S$ is the image of at least one $r \in R$.

Definition 2.16 (Ring isomorphism). A ring isomorphism is a bijective homomorphism.

Informally speaking, a ring isomorphism $f : R \rightarrow S$ is a mapping between rings that are structurally the same such that any element of R has exactly one image in S .

Note that $(\mathbb{Z}_n, +, \cdot)$ is a finite field if and only if n is a prime number. Furthermore, if F is a finite field, then F contains p^m elements for some prime p and integer $m \geq 1$. For every prime power order p^m , there is a unique finite field of order p^m . This field is denoted by \mathbb{F}_{p^m} or $GF(p^m)$. The finite field \mathbb{F}_{p^m} is unique up to an isomorphism.

2.3 Number Theoretic Assumptions

This section presents a collection of number theoretic assumptions. The security of our future constructions falls or stands on these assumptions. If one of these assumptions would prove to be invalid, not only this thesis would be superfluous, society would no longer be protected by widely adopted cryptographic protocols like RSA or ElGamal encryption [15, 65].

In the definitions that follow $\langle G, n, g \rangle \leftarrow \mathcal{G}(1^k)$ is defined as the setup algorithm that generates a group G of order n and a generator $g \in G$ on input of the security parameter k .

Definition 2.17 (DL). The *discrete logarithm problem* is defined as follows. Given a finite cyclic group G of order n , a generator $g \in G$ and an element $a \in G$, find the integer $x, 0 \leq x \leq n-1$ such that $g^x = a$.

The *discrete logarithm assumption* holds if for any algorithm $\mathcal{A}(g, g^x)$ trying to solve the DL problem there exists a negligible function $\mu(k)$ such that

$$\Pr \left[\mathcal{A}(g, g^x) = a \mid \langle G, n, g \rangle \leftarrow \mathcal{G}(1^k) \right] \leq \mu(k)$$

where the probability is over the random choice of n, g in G according to the distribution induced by $\mathcal{G}(1^k)$, the random choice of a in G and the random bits of the algorithm \mathcal{A} .

Definition 2.18 (CDH). The *Computational Diffie-Hellman problem* is defined as follows. Given a finite cyclic group G of order n , a generator $g \in G$ and g^a, g^b with uniformly chosen random independent elements $a, b \in \{1, \dots, |G|\}$, find the value g^{ab} .

The *Computational Diffie-Hellman assumption* holds if for any algorithm $\mathcal{A}(g, g^a, g^b)$ trying to solve the CDH problem there exists a negligible function $\mu(k)$ such that

$$\Pr \left[\mathcal{A}(g, g^a, g^b) = g^{ab} \mid \langle G, n, g \rangle \leftarrow \mathcal{G}(1^k) \right] \leq \mu(k)$$

where the probability is over the random choice of n, g in G according to the distribution induced by $\mathcal{G}(1^k)$, the random choice of a, b in $\{1, \dots, |G|\}$ and the random bits of the algorithm \mathcal{A} .

Definition 2.19 (DDH). The *Decisional Diffie-Hellman problem* is defined as follows. Given a finite cyclic group G of order n , a generator $g \in G$ and g^a, g^b, g^{ab}, g^c with uniformly chosen random independent elements $a, b, c \in \{1, \dots, |G|\}$, distinguish $\langle g, g^a, g^b, g^{ab} \rangle$ from $\langle g, g^a, g^b, g^c \rangle$.

Define $\mathcal{A}(x)$ as an algorithm returning **true** if $x = \langle g, g^a, g^b, g^{ab} \rangle$ and **false** if $x = \langle g, g^a, g^b, g^c \rangle$ for $c \neq ab$. The *Decisional Diffie-Hellman assumption* holds if for any such algorithm $\mathcal{A}(x)$ there exists a negligible function $\mu(k)$ such that

$$|\Pr \left[\mathcal{A}(\langle g, g^a, g^b, g^{ab} \rangle) = \text{true} \right] - \Pr \left[\mathcal{A}(\langle g, g^a, g^b, g^c \rangle) = \text{true} \right]| \leq \mu(k)$$

where the probability is over the random choice of n, g in G according to the distribution induced by $\mathcal{G}(1^k)$, the random choice of a, b, c in $\{1, \dots, |G|\}$ and the random bits of the algorithm \mathcal{A} .

Definition 2.19 states that $\langle g, g^a, g^b, g^{ab} \rangle$ and $\langle g, g^a, g^b, g^c \rangle$ are *computationally indistinguishable*. It means that no efficient algorithm exists that can distinguish both arguments with non-negligible probability. The concept of computational indistinguishable arguments bears close resemblance to statistically indistinguishable ensembles. The reader is referred to [46] and [47] for a more in depth discussion of the topic. The intuitive interpretation of Definition 2.19 is that g^{ab} looks like any other random element in G .

Someone with the ability to calculate discrete logarithms could trivially solve the CDH problem. That is, if a and b can be derived only from $\langle g^a, g^b \rangle$, it becomes easy to calculate g^{ab} . Therefore, a group structure where the CDH assumption holds, immediately implies a group where the DL assumption is valid as well. There is no mathematical proof that supports the inverse relation. Thus, a group where the DL problem is hard not necessarily implies the CDH problem. For specific group structures [63] and [64] show that CDH immediately follows from the DL assumption, however, their proof can not be generalised to just any group.

There exists a similar relation between the CDH and the DDH problem. If a powerful algorithm could solve CDH, i.e. derive g^{ab} from $\langle g, g^a, g^b \rangle$ alone, it would become trivial to distinguish $\langle g, g^a, g^b, g^{ab} \rangle$ from $\langle g, g^a, g^b, g^c \rangle$. Again, an inverse relation can not be proven. As a matter of fact, concrete examples of groups exist where CDH is hard although DDH is not.

Therefore, the relation between DL, CDH and DDH is often written as follows

$$DDH \Rightarrow CDH \Rightarrow DL$$

The \Rightarrow notation is then translated into "immediately implies". In a group where DDH is hard both CDH and DL will be hard. On the contrary, there exist group structures where the CDH and the DL assumption hold while DDH can be found easily. Such groups are called *Gap Diffie-Hellman Groups*.

Definition 2.20 (GDH). The *Gap Diffie-Hellman problem* is defined as follows. Solve the CDH problem with the help of a DDH oracle. Given a finite cyclic group G of order n , a generator $g \in G$ and g^a, g^b with uniformly chosen random independent elements $a, b \in \{1, \dots, |G|\}$, find the value g^{ab} with the help of a DDH oracle $\mathcal{DDH}(g, g^a, g^b, z)$. Where the DDH oracle $\mathcal{DDH}(g, g^a, g^b, z)$ is defined to return **true** if $z = g^{ab}$ and **false** if $z \neq g^{ab}$.

The *Gap Diffie-Hellman assumption* holds if for any algorithm $\mathcal{A}(g, g^a, g^b)$ trying to solve the CDH problem with the help of a DDH oracle $\mathcal{DDH}(g, g^a, g^b, z)$ there exists a negligible function $\mu(k)$ such that

$$\Pr \left[\mathcal{A}(g, g^a, g^b) = g^{ab} \mid \langle G, n, g \rangle \leftarrow \mathcal{G}(1^k) \right] \leq \mu(k)$$

where the probability is over the random choice of n, g in G according to the distribution induced by $\mathcal{G}(1^k)$, the random choice of a, b in $\{1, \dots, |G|\}$ and the random bits of the algorithm \mathcal{A} .

As discussed in the next Section 2.4 bilinear pairings are an example of a practical usable DDH oracle [55].

2.4 Bilinear Maps

2.4.1 Definition

Definition 2.21 (Admissible bilinear map). Let G_1, G_2 and G_T be three groups of order q for some large prime q . An *admissible bilinear map* $e : G_1 \times G_2 \rightarrow G_T$ is defined as a map from the gap groups G_1 and G_2 to the target group G_T that satisfies the following properties:

1. *Bilinearity* $\forall a, b \in \mathbb{Z}, \forall P \in G_1, \forall Q \in G_2 : e(aP, bQ) = e(P, Q)^{ab}$
2. *Non-degeneracy* If P is a generator of G_1 and Q is a generator of G_2 , $e(P, Q)$ is a generator of G_T
3. *Computability* There is an efficient algorithm to compute $e(P, Q)$ for all $P \in G_1$ and $Q \in G_2$

In literature, authors distinguish two types of admissible bilinear maps. A *symmetric bilinear map* is an admissible bilinear map where the gap groups are the same, i.e. $G_1 = G_2$. Definition 2.21 describes the more general *asymmetric bilinear map* where $G_1 \neq G_2$. Schemes relying on symmetric bilinear maps are easier to construct information theoretic security proofs although asymmetric bilinear maps are more efficient and suitable for implementation thanks to their flexible embedding degree [19, 83].

In practice, bilinear maps are constructed using pairings. The most popular pairings implementing admissible bilinear maps are the Weil pairing [19] and the Tate pairing [41]. Both the Tate and the Weil pairing rely on abelian varieties for their implementation. G_1 is mostly an additive elliptic curve group, G_2 a multiplicative elliptic curve group while G_T is a finite field. For instance, the asymmetric Weil pairing is often implemented with a cyclic subgroup of $E(\mathbb{F}_p)$ of order q for G_2 and a different cyclic subgroup of $E(\mathbb{F}_{p^6})$ of the same order q for G_1 where $E(\mathbb{F}_{p^6})$ denotes the group of points on an elliptic curve E over the finite field \mathbb{F}_{p^6} . The interested reader is referred to [1] for more information concerning elliptic curves and their use in pairing based cryptography. Details on Elliptic Curve Cryptography fall out of the scope of this thesis as it suffices to make abstraction of these concepts for the remainder of the text.

Research [7, 54, 4] has recently shown that the discrete logarithm problem is easier in the symmetric setting because symmetric pairings rely on more structured supersingular (hyper)elliptic curves. Therefore, care should be taken when using symmetric pairings [83].

2.4.2 Bilinear Diffie-Hellman Assumption

It is not a coincidence that G_1 and G_2 are called gap groups. A bilinear map allows to solve the Decisional Diffie-Hellman problem in G_1 and G_2 . The DDH problem in G_1 consists of distinguishing $\langle P, aP, bP, abP \rangle$ from $\langle P, aP, bP, cP \rangle$ where $P \in G_1$, P is a generator of G_1 and a, b, c randomly chosen in $\{1, \dots, |G_1|\}$. Given a symmetric bilinear map $e : G_1 \times G_1 \rightarrow G_T$ a solution to this problem can be found by relying on the bilinearity of the pairing as follows:

$$e(aP, bP) = e(P, P)^{ab} \stackrel{?}{=} e(P, cP) = e(P, P)^c$$

Such that the second equality will hold only if $ab = c$. A similar statement can be made concerning G_2 with the help of the map $e : G_2 \times G_2 \rightarrow G_T$. Consequently G_1 and G_2 are both GDH groups. From Section 2.3 it follows that CDH can still be hard in GDH groups because DDH is a stronger assumption [19].

Since DDH in the Gap groups G_1 and G_2 is easy, DDH can not serve as a basis for crypto systems in these groups. Therefore, an alternative to the CDH problem is defined called the Bilinear Diffie-Hellman problem.

In the definition that follows $\mathcal{G}(1^k)$ is defined to be a BDH parameter generator as in [19], i.e. \mathcal{G} takes as input a security parameter k , \mathcal{G} runs in polynomial time in k and \mathcal{G} outputs a prime number q , the description of two groups G_1, G_2 of order q and the description of an admissible bilinear map $e : G_1 \times G_2 \rightarrow G_T$.

Definition 2.22 (BDH). The *Bilinear Diffie-Hellman problem* is defined as follows. Given any admissible bilinear pairing $e : G_1 \times G_2 \rightarrow G_T$ with random $P, aP, bP \in G_1$ and random $Q, aQ, bQ \in G_2$ with uniformly chosen random independent elements $a, b, c \in \{1, \dots, |G|\}$, find $e(P, Q)^{abc}$

The *Bilinear Diffie-Hellman assumption* holds if for any algorithm $\mathcal{A}(P, aP, bP, Q, aQ, bQ)$ trying to solve the BDH problem there exists a negligible function $\mu(k)$ such that

$$\Pr \left[\mathcal{A}(P, aP, bP, Q, aQ, bQ) = e(P, Q)^{abc} \mid \langle q, G_1, G_2, e \rangle \leftarrow \mathcal{G}(1^k) \right] \leq \mu(k)$$

where the probability is over the random choice of q, G_1, G_2, e according to the distribution induced by $\mathcal{G}(1^k)$, the random choice of a, b in $\{1, \dots, |G|\}$ and the random bits of the algorithm \mathcal{A} .

2.5 Hash Functions

2.5.1 Definition

A *hash function* is a computationally efficient deterministic function mapping binary strings of arbitrary length to binary strings of some fixed length, called *hash-values*.

Cryptographic hash functions have the following desirable properties:

- *Computability:* Given a binary string m , the hash value h can be calculated efficiently $h = \text{hash}(m)$

- *Pre-image resistance:* Given a hash value h , it is infeasible to calculate a corresponding binary string m such that $h = \text{hash}(m)$
- *Second pre-image resistance:* Given a binary string m_1 , it is hard to find a different binary string m_2 such that $\text{hash}(m_1) = \text{hash}(m_2)$
- *Strong collision resistance:* Given a **hash** function $\text{hash}(\cdot)$, it is hard to find two different binary strings m_1 and m_2 such that $\text{hash}(m_1) = \text{hash}(m_2)$

Hash functions are useful in a wide plethora of practical applications. Hash functions serve as one way functions in password databases to relax sensitivity of the stored content. Furthermore hash functions are a valuable tool for data authentication and integrity checking. Another use of hash functions is in protocols involving a priori commitments. If the reader is new to the concept of hash functions, he is referred to [65] for an in depth discussion on the topic.

2.5.2 Random Oracles

A *random oracle* is a theoretical black box that returns for each unique query a uniformly random chosen result from its output domain. A random oracle is deterministic, i.e. given a particular input it will always produce the same output.

In a perfect world hash functions can be considered random oracles. That is, if hash functions were perfect, they would behave as random oracles. Therefore, hash functions are often considered random oracles in security proofs. Such security proofs are said to be *proven secure in the random oracle model*. Proofs in the random oracle model first show that an algorithm is secure if a theoretical random oracle would be used. A next step of these security proofs is replacing the random oracle accesses by the computation of an appropriately chosen (hash) function h [10]. Algorithms that do not require such a construction in their security proof are said to be *proven secure in the standard model*.

Although theoretical definitions of random oracles and hash functions are quite similar, some practical implementations of hash functions do not behave like random oracles at all. Canetti et al. show that there exist signature and encryption schemes that are secure in the Random Oracle Model, although any implementation of the random oracle results in insecure schemes [25]. Coron et al. counter these findings with indistinguishability, i.e. if a hash function is indistinguishable from a random oracle the random oracle can be replaced by the hash function while maintaining a valid security proof [31]. Although research results from Coron et al. are debated in [39] and [69], it is a common belief that proofs in the random oracle model provide some evidence that a system is secure. As a matter of fact, indistinguishability from random oracles certainly contributed to the victory of Keccak in the NIST hash function competition for a new SHA-3 hashing standard as all final round hashing algorithms supported this property [9].

2.6 Summary

The first part of this chapter introduced the concepts of a negligible function as well as algebraic structures such as groups and finite fields. These basic notions were used further on to define number theoretic hard problems that serve as a basis for security. Starting from the discrete logarithm assumption, several variants of the Diffie-Hellman problem were introduced eventually leading to the Gap Diffie-Hellman assumption. Notion of the Gap Diffie-Hellman assumption allowed to uncover gap groups and their use in admissible bilinear maps. The Bilinear Diffie-Hellman assumption was defined as a computationally infeasible problem for the construction of cryptographic protocols relying on bilinear maps. Finally, this chapter concluded with differences between security under random oracle assumptions and security in the standard model.

Now the reader has knowledge of the mathematic fundamentals, more advanced cryptographic constructions like identity-based encryption, broadcast encryption and distributed key generation are revealed in Chapter 3.

Literature Review

This chapter briefly highlights the most useful cryptographic building blocks when trying to construct an encryption mechanism for online social networks. A profound background of the existing literature, will make it easier to design the perfect encryption tool.

For starters, an introduction is given to asymmetric cryptography and its drawbacks. In a next section, identity-based encryption (IBE) is proposed as a possible alternative to the existing asymmetric approaches. The reader is introduced to the basic concept of identity-based encryption, drawbacks and advantages, different security definitions and the evolution of IBE in literature. The section about IBE is concluded with a concrete IBE scheme from Boneh and Franklin. This is followed by an elaborate discussion on broadcast encryption (BE) and secret sharing. Finally, distributed key generation is proposed as a possible solution to the inherent key escrow problem of IBE.

3.1 Asymmetric Cryptography

The idea behind asymmetric cryptography is to use a different key for encryption than decryption thereby making secure communication easier between parties who have never met before. A background on the basic concepts of traditional asymmetric cryptography will allow to put the conclusions of this chapter into context.

3.1.1 Definition

In a public key system each entity A has a key pair $\langle pk_A, sk_A \rangle$ where pk_A and sk_A respectively denote the public and the private key of entity A . The public key is published publicly while the secret key often remains a secret only known by A or a third party trusted by A . The domain of cryptography describing the concept of these key pairs is often called *asymmetric cryptography*.

Key pair generation is a two step process. In the first step, the secret key sk is chosen uniformly random. In the next step the public key pk is derived by applying a one-way function to sk . The one-way property of the applied function implies that it is computationally hard to derive the public key from the secret key, e.g. in the

ElGamal encryption scheme the public key is calculated as $pk = g^{sk}$ in a group \mathbb{Z}_p for some large prime p . As long as the discrete logarithm assumption from Section 2.3 holds it is infeasible to derive sk from pk .

The idea of two different keys enables a wide plethora of useful applications. Once all public keys of a PKI system are known, two entities who never met before can immediately setup secure communication by encrypting under the correct public keys. Asymmetric crypto also enables these entities to authenticate their messages by relying on signature schemes. Authenticity of signatures then immediately follows from the privacy of sk .

However, public key systems only shift the problem from trusting the users to trusting their keys. For example, if Eve would succeed in authenticating her own public key pk_{Eve} as the public key of Alice pk_{Alice} , Eve would be able to read all confidential information that is intended to Alice as she obviously has the secret key sk_{Eve} corresponding to pk_{Eve} . Therefore, it is important that public key system rely on an architecture that authenticates whether keypairs belong to the claimed owner. In practice this is mostly achieved with the help of Public Key Infrastructures (PKIs) such as certification authorities or a web of trust.

3.1.2 Certification Authorities

In a traditional PKI system, all entities in the system trust a central party called the *Certification Authority* (CA). It is the CA that guarantees public keys belong to the claimed owner.

Suppose Alice wants to start using a key pair $\langle pk_A, sk_A \rangle$. She has to authenticate herself with the CA by correctly following a protocol that confirms Alice's identity. Once Alice is authenticated with the CA, Alice sends the public key pk_A to the CA along with a proof showing that Alice also owns the corresponding secret key sk_A . This "proof of correct possession" often takes the form of a signature $S_{sk_A}(pk_A)$ generated by the secret key sk_A on the public key pk_A .

Once the CA is convinced of the authenticity of Alice's public key, it distributes a certificate approving that pk_A effectively belongs to Alice. To avoid forged certificates, the CA signs Alice's certificate with its private key sk_{CA} . Anyone doubting the authenticity of the public key pk_A can get convinced pk_A effectively belongs to Alice by checking the signature of the CA with the CA's public key pk_{CA} .

In practice, CAs often approve each others trustworthiness by issuing certificates on their signing keys. In this way, often highly complex hierarchical architectures are achieved that boil down to the trust in one signing key of the highest authority. This puts heavy requirements on the CA's infrastructure as a compromised CA signing key can break the system completely. Indeed, a compromised signing key would allow to sign certificates of unauthenticated public keys or even certificates of public keys that belong to malicious entities.

If an entity's secret key would get compromised, it can be revoked by the CA. CAs achieve this by periodic publication of *revocation lists*. These revocation lists contain all public keys of entities who lost or even revealed their secret keys. Users relying on a PKI should therefore always check these continuously growing lists

before trusting a keypair. Therefore, revocation lists not only make the system less transparent, they also put high demands on the infrastructure of entities relying on the PKI.

To partially get around the issue of revocation lists, certificates contain an expiration date. After expiry of a certificate it should no longer be trusted. However, this requires keypair owners to contact CAs more frequently to sign new certificates each time the previous one has expired. Clearly, this puts a higher computational demand on the authentication procedure of the CAs as well.

3.1.3 OpenPGP and Webs of Trust

An alternative to the traditional PKI setting relying on CAs is a *web of trust*. In a web of trust any entity can rate the trustworthiness of a public key. For example, if Bob receives Alice's public key personally during a date, the public key can be considered more trustworthy than when Bob receives Alice's key via e-mail. Web of trust systems allow users to vet for the authenticity other users' keys in the system. A standardised web of trust system is OpenPGP [24].

The major advantage of a web of trust is that there no longer needs to be a CA with highly secure infrastructure as the publication of certificates now becomes a shared responsibility.

The system also has its drawbacks. Usability studies already have shown that non tech-savvy users have problems using PGP systems [79]. Furthermore, users are now required to judge for themselves whether they can trust a public key or not. This gives more responsibility to users than most of them can handle without proper knowledge of the consequences to their actions.

3.2 Identity-Based Encryption

Although architectures relying on CAs or webs of trust are common practice, they seem to have their drawbacks. However, recent research has uncovered a new paradigm with promising features called identity-based encryption.

Shamir [73] already proposed a first concept of identity-based cryptography in 1984. In identity-based cryptography any string can be a valid public key for encryption or signature schemes thereby eliminating the need for digital certificates. Identity-based cryptography proves to be particularly elegant if the public key is related to an attribute that uniquely identifies the identity of the user like an e-mail address, an IP address or a telephone number. Consequently, identity-based cryptography reduces system complexity and the cost for establishing and managing the Public Key Infrastructure (PKI) [5].

3.2.1 Definition

A generic Identity-Based Encryption (IBE) scheme is composed of four probabilistic polynomial time algorithms [19]:

3. LITERATURE REVIEW

IBE.Setup(1^λ) On input of a security parameter λ , outputs a master secret s_k and public parameters $params$.

IBE.Extract($params, s_k, id$): Takes public parameters $params$, the master secret s_k , and an id as input and returns the private key s_{id} corresponding to the identity id .

IBE.Encrypt($params, id, m$): Returns the encryption c of the message m on the input of the public parameters $params$, the id , and the arbitrary length message m .

IBE.Decrypt(s_{id}, c): Decrypts the ciphertext $c = \text{IBE.Encrypt}(params, id, m)$ back to the message m on input of the secret key s_{id} corresponding to the receiving identity id .

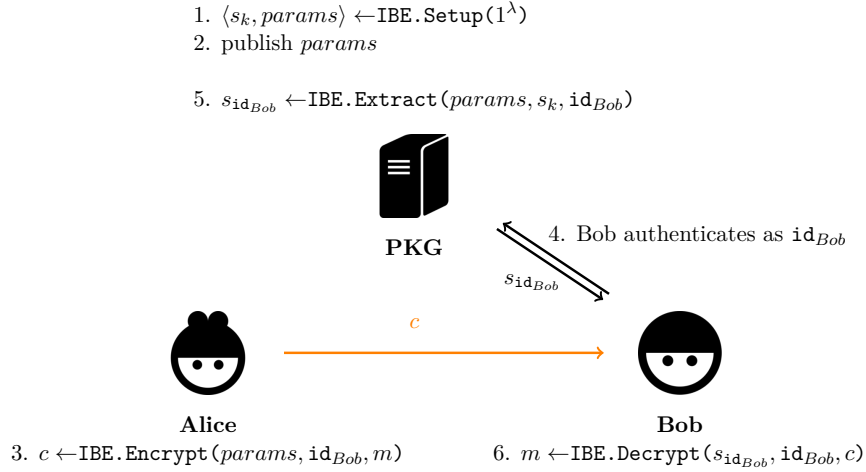


FIGURE 3.1: Generic identity-based encryption scheme. The orange arrow denotes an insecure channel that can be eavesdropped.

These generic algorithms are best explained using Figure 3.1. A trusted Public Key Generator (PKG) generates a master secret key s_k and public parameters $params$ on input of the security parameter k . Next, the PKG publishes the public parameters $params$ while storing s_k preferably in encrypted fashion on a local disk. When Alice wants to send a message m to Bob, it suffices for her to know the public parameters $params$ and his id_{Bob} uniquely identifying Bob. Alice encrypts the message to a ciphertext c that is sent over to Bob. On receipt of the ciphertext, Bob authenticates to the PKG over a secure channel to request his secret key $s_{id_{Bob}}$. Subsequently, the PKG generates the secret key $s_{id_{Bob}}$ corresponding to Bob's identity id_{Bob} on input of the master secret key s_k , Bob's id_{Bob} and public parameters $params$. Subsequently, the PKG sends $s_{id_{Bob}}$ back again over a secure channel. Finally, Bob has all the required information to decrypt the ciphertext c to its original plaintext message m .

3.2.2 Advantages and Drawbacks of IBE

Note that it is important that the PKG can be fully trusted as it will generate all the secret keys s_{id} in the system. Referring to Figure 3.1, a malicious PKG server could use this information to start eavesdropping on the insecure channel between Alice and Bob while decrypting all ciphertexts that are being sent over. The undesired property that private keys have to be shared with a trusted third party is often called *key escrow* in literature.

Another drawback from the scheme in Figure 3.1 is that keys can not be revoked in the system. However, Bob's secret key $s_{id_{Bob}}$ can still get compromised if he is careless with the storage of his secret key. In fact, a lot of research has been done in literature on the revocation of IBE keys [14, 18, 52, 61]. Key revocation often requires additional infrastructure that complicates the elegance of the currently proposed IBE scheme. As a matter of fact, the major drawback of revoking Bobs key is that Bob can no longer receive encrypted messages because his public key is part of his identity. Therefore, a pragmatic solution to this issue could be to append expiration dates to the public keys. Consequently, public keys will only be valid for a limited amount of time thereby restricting the damage that could be done with a compromised secret key [19].

The scheme from Figure 3.1 has some desired properties as well. For starters, only one PKG suffices to realise the system, which relaxes expensive requirements on the public key infrastructure. Furthermore, once the PKG has successfully delivered all the secret keys in the system, it can go offline as the scheme does not require any future interactions between the PKG and the users in the system.

Another useful property of an IBE scheme is that Bob does not need to subscribe to a hierarchy of Certification Authorities neither a chain of trust before Alice can start sending him messages. In this way, the possibility to send encrypted messages becomes inherently part of any system in which the users are assigned unique identifiers. This is particularly useful in systems where the majority of the users has no knowledge about cryptographic primitives. Users do no longer need to generate a key pair neither subscribe to a third party infrastructure. It suffices to recall how their connections can be uniquely identified in the system to know their public keys.

3.2.3 Security of IBE

A cryptographic scheme should meet two properties in order to be of practical use. First, the scheme should be consistent, i.e. decryption reverses encryption. Second, the scheme should be secure. The definition of security is more subtle as different levels of security can be defined. In IBE *indistinguishability under chosen plaintext attack* (IND-CPA) and *indistinguishability under chosen ciphertext attack* (IND-CCA) are often considered. Anonymity of the encryption scheme is an additional property of the scheme that is often desired [2].

Note that both the notion of IND-CPA, IND-CCA and anonymity are only introduced in an informal way in this section to give a basic understanding of these

concepts to the reader. If a more formal description of IND-CPA and IND-CCA is required, the reader is referred to [19]. For a more formal description of ciphertext anonymity the reader is referred to [3].

Indistinguishability Under Chosen Plaintext Attack

Indistinguishability under chosen plaintext attack (IND-CPA) means that an adversary has no advantage in trying to determine which of both given plaintext messages m_0 and m_1 generated a ciphertext c . It captures the notion of *semantic security*, i.e. that any ciphertext c should not give more information about the original plaintext m than any other random binary string of the same length.

IND-CPA is best defined with the help of a game that challenges the adversary. If the adversary has negligible advantage trying to win the IND-CPA game in Algorithm 1, the IBE system is said to be IND-CPA secure.

Algorithm 1 Generic IBE-IND-CPA Game [2]

Goal: An adversary is challenged by a game to check the IND-CPA security of an IBE scheme.

Result: This IBE-IND-CPA Game helps to define the concept of IND-CPA security for IBE schemes.

1. The challenger runs $\langle s_k, params \rangle \leftarrow \text{IBE.Setup}(1^\lambda)$ and returns $params$ to the adversary.
 2. The adversary can start querying an oracle $O_{\text{Extract}}(\text{id}_i)$ that returns a secret key $s_{\text{id}_i} \leftarrow \text{IBE.Extract}(params, s_k, \text{id})$ corresponding to an adversary defined identity id_i .
 3. The adversary picks two equal length plaintext messages m_0 and m_1 and an identity $\text{id}_{\text{encrypt}}$. The adversary honestly passes $\langle m_0, m_1, \text{id}_{\text{encrypt}} \rangle$ to the challenger.
 4. The challenger picks a random bit b and executes $c \leftarrow \text{IBE.Encrypt}(params, \text{id}_{\text{encrypt}}, m_b)$. The challenger gives c to the adversary.
 5. The adversary continues querying the oracle $O_{\text{Extract}}(\text{id}_i)$ adaptively.
 6. The adversary outputs a bit b' based on the ciphertext c . If $b = b'$ the adversary wins the game. If $b \neq b'$ or if the adversary queried the oracle $O_{\text{Extract}}(\text{id}_i)$ with $\text{id}_i = \text{id}_{\text{encrypt}}$ during step 2 or step 5, the adversary loses the game.
-

Indistinguishability Under Chosen Ciphertext Attack

Indistinguishability under chosen ciphertext (IND-CCA) is a more demanding level of security. Therefore, an algorithm that is IND-CCA secure is considered more secure than an IND-CPA secure algorithm. IND-CCA security means that an adversary has no advantage in trying to determine which of both given plaintext messages m_0 and m_1 generated a ciphertext c even if the adversary has access to a list of (plaintext, ciphertext)-tuples.

IND-CCA is easiest to define with the help of a game that challenges an adversary similar to the IND-CPA game. The IND-CCA game contains two additional steps compared to the IND-CPA game in which the adversary gets access to another oracle. If the adversary has negligible advantage trying to win the IND-CCA game from Algorithm 2, the IBE system is said to be IND-CCA secure.

In literature a distinction is often made between a *non-adaptive* case (IND-CCA1) and an *adaptive* case (IND-CCA2) of IND-CCA. In the non-adaptive case, step 6 of Algorithm 2 is not allowed. More precisely, an IBE scheme that satisfies Algorithm 2 is said to be IND-CCA2 secure.

Anonymous Identity-Based Encryption

An IBE scheme is called anonymous (ANO-IBE) when the ciphertext does not leak the identity of the recipient. In the case of Figure 3.1, this would mean that no eavesdropper on the insecure channel between Alice and Bob could derive that Bob is the recipient based on the information in the ciphertext c alone [21].

ANO-IBE is easiest to define with the help of a game that challenges an adversary similar to the IND-CPA game. If the adversary has negligible advantage trying to win the ANO-IBE game in Algorithm 3, the IBE system is said to be anonymous.

Gentry [43] is the first to combine the notions of IND-CPA and IND-CCA with ANO-IBE. A system is then said to be IND-ANO-CPA secure or IND-ANO-CCA secure if it satisfies a modified version of the game in Algorithm 3. For a more detailed discussion on the topic the reader is referred to the original paper [43].

3.2.4 Evolution of IBE

Although Shamir easily constructed an identity-based signature scheme based on RSA in 1984, the use case of IBE remained an open problem until the introduction of bilinear maps. In [19] Boneh and Franklin propose the first practically usable IBE scheme based on the Weil pairing. However, the security proof in [19] still relies on the random oracle assumption. Sakai and Kasahara [68] propose a different IBE scheme independently from Boneh and Franklin. The scheme from Sakai and Kasahara initially received less attention though, this is partially because the original paper is written in Japanese and partially because it is proposed without security proof. A later reference [71] from the same authors proposes a more efficient version of their original scheme that is later proven IND-CCA secure in the random oracle model by Chen et al [29].

Algorithm 2 Generic IBE-IND-CCA Game [2]

Goal: An adversary is challenged by a game to check the IND-CCA security of an IBE scheme.

Result: This IBE-IND-CCA Game helps to define the concept of IND-CPA security for IBE schemes.

1. The challenger runs $\langle s_k, params \rangle \leftarrow \text{IBE.Setup}(1^\lambda)$ and returns $params$ to the adversary.
 2. The adversary can start querying an oracle $O_{Extract}(\text{id}_i)$ that returns a secret key $s_{\text{id}_i} \leftarrow \text{IBE.Extract}(params, s_k, \text{id})$ corresponding to an adversary defined identity id_i .
 3. The adversary can start querying another oracle $O_{Decrypt}(s_{\text{id}_i}, c_j)$ that returns a plaintext $m_j \leftarrow \text{IBE.Decrypt}(s_{\text{id}_i}, c_j)$ corresponding to an adversary defined ciphertext c_j and identity id_i .
 4. The adversary picks two equal length plaintext messages m_0 and m_1 and an identity $\text{id}_{encrypt}$. The adversary honestly passes $\langle m_0, m_1, \text{id}_{encrypt} \rangle$ to the challenger.
 5. The challenger picks a random bit b and executes $c \leftarrow \text{IBE.Encrypt}(params, \text{id}, m_b)$. The challenger gives c to the adversary.
 6. The adversary continues querying the oracle $O_{Extract}(\text{id}_i)$ adaptively.
 7. The adversary continues querying the oracle $O_{Decrypt}(s_{\text{id}_i}, c_j)$ adaptively.
 8. The adversary outputs a bit b' based on the ciphertext c . If $b = b'$ the adversary wins the game. If $b \neq b'$, if the adversary queried the oracle $O_{Extract}(\text{id}_i)$ with $\text{id}_i = \text{id}_{encrypt}$ during step 2 or step 6 or if the adversary queried the oracle $O_{Decrypt}(s_{\text{id}_i}, c_j)$ with $c_j = c$ during step 3 or step 7, the adversary loses the game.
-

Algorithm 3 Generic ANO-IBE Game [2]

Goal: An adversary is challenged by a game to check the ANO-IBE security of an IBE scheme.

Result: This ANO-IBE Game helps to define the concept of ANO-IBE security for IBE schemes.

1. The challenger runs $\langle s_k, params \rangle \leftarrow \text{IBE.Setup}(1^\lambda)$ and returns $params$ to the adversary.
 2. The adversary can start querying an oracle $O_{Extract}(\text{id}_i)$ that returns a secret key $s_{\text{id}_i} \leftarrow \text{IBE.Extract}(params, s_k, \text{id}_i)$ corresponding to an adversary defined identity id_i .
 3. The adversary picks a plaintext message m and an identity $\text{id}_{encrypt}$. The adversary honestly passes $\langle m, \text{id}_{encrypt} \rangle$ to the challenger.
 4. The challenger picks a random bit b and computes $c \leftarrow \text{IBE.Encrypt}(params, \text{id}_{encrypt}, m)$ if $b = 0$. If $b = 1$, the challenger computes $c \leftarrow \text{IBE.Encrypt}(params, \text{id}_{encrypt}, r)$ where r is a random bit sequence with the same length as the message m . The challenger gives c to the adversary.
 5. The adversary continues querying the oracle $O_{Extract}(\text{id}_i)$ adaptively.
 6. The adversary outputs a bit b' based on the ciphertext c . If $b = b'$ the adversary wins the game. If $b \neq b'$ or if the adversary queried the oracle $O_{Extract}(\text{id}_i)$ with $\text{id}_i = \text{id}_{encrypt}$ during step 2 or step 5, the adversary loses the game.
-

Canetti et al. [26] are the first to succeed in proposing a secure IBE scheme without having to rely on the random oracle model. Nevertheless, the attacker model in [26] requires the adversary to declare which identity id that will be targeted during step 5 of the CCA Game (Algorithm 2) and step 4 of the CPA Game, therefore the scheme in [19] is considered more secure as attackers can adaptively choose the targeted identity. Boneh and Boyen [16] present a variant to [26] which also realises only selective ID security.

Waters [77] is the first to present a scheme that is IND-CCA secure in the standard model. Drawback of the scheme from Waters [77] is that it requires large public parameters. Gentry [43] proposes a more efficient alternative to this scheme in the standard model while achieving shorter public parameters. However, the scheme from Gentry relies on a complicated hardness assumption called q-BDHE. It is only after the introduction of the Dual System paradigm by Waters [78] in 2009 that IND-CCA security can be achieved in the standard model based on reasonable assumptions. De Caro et al. [28] are the first to define an IND-ANO-CCA secure IBE scheme on the Dual System construction of Waters [78].

Although all these contributions were a step forward in the evolution of IBE, not

all of these schemes are ANO-IBE. Most IBE systems in the random oracle model can be proven anonymous. Therefore, the IBE scheme from Boneh and Franklin [19] is IND-ANO-CCA secure. In the standard model, it appeared to be harder to construct ANO-IBE schemes at first sight, e.g. it can be proven that the scheme from Boneh and Boyen [16] is not anonymous in its original form. The scheme from Gentry [43] was the first anonymous IBE scheme in the standard model. Boyen and Waters [21] published almost synchronously another IBE scheme in the standard model that is also IND-ANO-CCA secure. In 2010, Ducas [35] showed that even schemes that were first considered not anonymous like the one from Boneh and Boyen [16] but also [17, 77] can be proven anonymous when relying on asymmetric pairings thereby making anonymity a more common property in IBE schemes.

3.2.5 Most Attractive IBE Schemes

From all schemes discussed in Section 3.2.4 the ones initially developed by Boneh and Franklin [19] and Sakai and Kasahara [71] are the most attractive ones in the random oracle model because of their anonymity and non-selective security. Consequently, it is not a coincidence that both schemes have found description in an informational RFC document. Sakai and Kasahara IBE is described in RFC 6508 [50] and RFC 6509 [49]. Boneh and Franklin IBE can be found in RFC 5409 [62].

In the standard model mainly the anonymous IBE constructions from Gentry [43] and De Caro et al. [28] have the most satisfying properties. However, IBE constructions in the standard model often come at the cost of higher computational requirements [20]. Certainly the scheme from De Caro demands a higher amount of computational resources since it relies on composite order groups. Although methods [40, 58] have been developed to convert IBE schemes from composite order groups to single order prime groups, these methods do not apply to the scheme from De Caro et al [57].

3.2.6 Boneh and Franklin IBE

Boneh and Franklin [19] propose two IBE algorithms called **BasicIndent** and **FullIndent** that will prove particularly useful in the remainder of this text. **BasicIndent** can be proven IND-ANO-CPA secure while **FullIndent** is proven IND-ANO-CCA secure. **BasicIndent** can be found in Algorithm 4 while **FullIndent** is explained in Algorithm 5. As already mentioned in Section 3.2.4, the scheme from Boneh and Franklin relies on the random oracle assumption and is therefore more efficient than similar schemes in the standard model.

3.3 Broadcast Encryption

Another relevant aspect of encryption in OSNs is how one encrypted message can be securely broadcasted to multiple users. Broadcast encryption (BE) was introduced by Fiat and Naor [38], as a public-key generalisation to a multi user setting. A BE scheme allows a user to encrypt a message m to a subset \mathcal{S} of users in a public key

Algorithm 4 BasicIndent [19]

Goal: Alice wants to send an IBE encrypted message to Bob.

Result: Alice sends an IBE encrypted ciphertext c that is successfully decrypted by Bob.

1. **Setup**(1^λ): Let λ be the security parameter for a security level of l bits.

a) Execute setup algorithm $\langle q, G_1, G_2, e : G_1 \times G_2 \rightarrow G_T, P \in G_1 \rangle \leftarrow \mathcal{G}(1^\lambda)$ to generate the parameters

- i. A large prime q
- ii. Gap groups G_1 and G_2 of order q
- iii. An admissible bilinear map $e : G_1 \times G_2 \rightarrow G_T$
- iv. A random generator $P \in G_1$

b) Choose a uniformly random $s_k \in \mathbb{Z}_q^*$ and calculate

$$P_{pub} = s_k P$$

c) Choose cryptographic hash functions

- i. $H_1 : (0, 1)^* \rightarrow G_1$
- ii. $H_2 : G_2 \rightarrow (0, 1)^l$

2. **Extract**($params, s_k, id$):

- a) Compute $Q_{id} = H_1(id) \in G_1$
- b) Set the private key of id to $s_{id} = s_k Q_{id}$

3. **Encrypt**($params, id, m$):

- a) Compute $Q_{id} = H_1(id)$
- b) Choose a random $r \in \mathbb{Z}_q$
- c) Encrypt the plaintext message m to the ciphertext c as

$$c = \langle rP, m \oplus H_2(g_{id}^r) \rangle = \langle U, v \rangle \quad \text{with } g_{id} = e(Q_{id}, P_{pub}) \in G_T$$

4. **Decrypt**(s_{id}, c): Decrypt the ciphertext c back to the plaintext message m as

$$m = v \oplus H_2(e(s_{id}, U))$$

Algorithm 5 FullIndent [19]

Goal: Alice wants to send an IBE encrypted message to Bob.

Result: Alice sends an IBE encrypted ciphertext c that is successfully decrypted by Bob.

1. **Setup**(1^λ):

- a) As in the BasicIndent scheme
- b) As in the BasicIndent scheme
- c) Choose cryptographic hash functions
 - i. $H_1 : (0, 1)^* \rightarrow G_1$
 - ii. $H_2 : G_2 \rightarrow (0, 1)^l$
 - iii. $H_3 : (0, 1)^l \rightarrow (0, 1)^l$

2. **Extract**($params, s_k, id$): As in the BasicIndent scheme

3. **Encrypt**($params, id, m$):

- a) Compute $Q_{id} = H_1(id)$
- b) Choose a random $sigma \in (0, 1)^l$
- c) Compute $r = H_3(sigma, m)$
- d) Encrypt the plaintext message m to the ciphertext c as

$$c = \langle rP, sigma \oplus H_2(g_{id}^r), m \oplus H_3(sigma) \rangle = \langle U, v, w \rangle$$

with $g_{id} = e(Q_{id}, P_{pub}) \in G_T$

4. **Decrypt**(s_{id}, c): Decrypt the ciphertext c back to the plaintext message m as follows

- a) Compute $sigma = v \oplus H_2(e(s_{id}, U))$
 - b) Compute $m = w \oplus H_3(sigma)$
 - c) Set $r = H_3(sigma, m)$. Test that $U = rP$. If not, reject the ciphertext.
 - d) Output m as the decryption of c
-

system, such that, only users in the set \mathcal{S} are able to decrypt the message. The computational overhead of BE is generally bound to the ciphertext and the number of recipients.

3.3.1 Definition

A generic Broadcast Encryption (BE) scheme is composed of four probabilistic polynomial time algorithms:

BE.Setup(1^λ) : On input of a security parameter λ , generates the public parameters *params* of the system.

BE.KeyGen(*params*) : Returns the public and private key (pk_i, sk_i) for each user i while taking the public parameters *params* into account.

BE.Encrypt : Takes a set of public key values $\mathcal{S} = \{pk_i \dots pk_{|\mathcal{S}|}\}$ corresponding to users i in the system along with a plaintext message m to generate a corresponding ciphertext c .

BE.Decrypt: Reconstructs m from c using the private key sk_i if the corresponding public key $pk_i \in \mathcal{S}$. Otherwise, return \perp .

Note that this definition is stated generically enough to allow all kinds of public keys to be used. Therefore, not only traditional PKIs can benefit from BE schemes, but also IBE schemes in which a public identifier id_i serves as a public key pk_i .

3.3.2 Evolution of Broadcast Encryption

The problem of BE has been widely studied in literature since its first introduction by Fiat and Naor [38]. This section highlights the most important evolutions of BE in literature. The summary that follows is far from complete as it only considers publications that are relevant to our final goal: achieving user-friendly broadcast encryption for OSNs.

Broadcast Encryption

The implementation from Fiat and Naor [38] requires a ciphertext of size $O(t \log^2 t \log n)$ to be secure against t colluding users. The first fully collusion resistant scheme was proposed in [66] by Naor et al. thereby making the ciphertext size independent of the number of colluding users. A collusion resistant BE scheme refers to a broadcast encryption scheme that is secure even if all users that are not in the recipient set \mathcal{S} would collaborate. Halevy and Shamir further reduce the required ciphertext length for collusion resistant schemes in [51]. It is the first paper in a series of many [34, 48, 59] that achieves ciphertext sizes only dependent on the number of revoked users $O(r)$. Boneh, Gentry and Waters [17] are the first to consider utilisation of bilinear maps to realise constant size ciphertexts and $O(n)$ public keys.

Identity-Based Broadcast Encryption

Sakai and Furukawa are the first to define a collusion resistant identity based broadcast encryption (IBBE) scheme in [70]. Independently from [70] Delerablée realises a similar IBBE scheme and claims to be the first as well in [33]. The size of the public key in both [70] and [33] is proportional to the maximum size of the intended set of recipients while realising short ciphertexts and private keys.

Baek et al. [5] define an IBBE scheme that requires only one pairing computation. The scheme in [5] is proven secure under the random oracle assumption where the attacker ties himself to a selective-ID attack. Gentry and Waters achieve identity based broadcast encryption with sublinear ciphertexts in [44]. Their scheme is proven secure against a stronger notion of adaptive security where the attacker can adaptively alter its queries depending on earlier received information. Barbosa and Farshim [6] proposed an identity-based key encapsulation scheme for multiple parties which is an extension of $mKEM$ as considered by Smart [75] to the identity-based setting. An $mKEM$ is a Key Encapsulation Mechanism which takes multiple public keys as input. An encrypted message under $mKEM$ consists of an encapsulated session key k and a symmetric encryption $E_k(m)$ of the plaintext message m under k . However, the scheme from Smart [75] is only proven secure under the random oracle assumption.

Anonymous Broadcast Encryption

All earlier mentioned references describing BE require the intended set of recipients to be published to realise higher efficiency. Barth, Boneh and Waters [8] are the first to design a BE scheme that takes the anonymity of the recipient into account. The proposed anonymous broadcast encryption (ANOBE) scheme imposes a linear dependency of the ciphertext on the number of recipients and can only be proven secure in the random oracle model. In [60] Libert et al., propose an alternative ANOBE scheme that is proven secure in the standard model. Both [8] and [60] propose a tag based system that allows efficient decryption at the cost of making the public master key linear dependent on the total number of users. Krzywicki et al. [56] propose a scheme that is proportional to the number of revoked users, although the security proof is rather informal. In [82], Yu et al. design an architecture that even hides the number of users in the recipient set using Attribute Based Encryption (ABE) [36].

However, ABE requires that all users are assigned attributes such that all users who have sufficient attributes in common can decrypt the message. In networks where the total number of users is large it can be a work intensive task to label each user with the correct attributes.

Outsider-Anonymous Broadcast Encryption

Fazio and Perera introduce the notion of outsider anonymous broadcast encryption in [36]. The scheme relies on IBE to encode where a recipient is positioned in a publicly published tree to achieve sublinear ciphertexts. It is remarkable that

sublinear ciphertexts are achieved while attaining recipient anonymity to all users that are outside the intended set of receivers. However, the scheme has the drawback of immediately fixing the total number of users that are allowed in the system. Furthermore, an additional architecture is required to maintain the tree of subscribed users. Finally, although IBE is used, the scheme does not allow to represent public keys of users by their public identifiers because the public key needs to be the position of a user in the tree structure of the external architecture. In this way, most of the desirable properties of IBE cancel out.

Although the scheme from Fazio and Perera does not fit the requirements for user-friendly broadcast-encryption in OSNs, it is useful to remember their definition of outsider-anonymity.

Definition 3.1 (Outsider Anonymity). A BE scheme is called *outsider anonymous* if the identities of the recipients are known to the other identities in the recipient set \mathcal{S} while remaining secret to other parties of the BE scheme.

3.3.3 Most Attractive BE Schemes

From all schemes discussed in Section 3.2.4 mainly the scheme from Libert et al. [61] has the most attractive properties as it is proven secure in the standard model at almost no reduced computational efficiency. The scheme supports anonymity in both identity-based BE as well as traditional asymmetric cryptosystems.

If anonymity is not an issue, different BE schemes have to be considered depending on the goals of the target application. The scheme from Libert et al. [60] will certainly not have the most desirable properties in non-anonymous BE environments since it can not benefit from higher efficiency due to the recipient being publicly known.

3.4 Secret Sharing

In earlier paragraphs of this chapter, identity-based encryption was already explored as a possible alternative to traditional public key infrastructures. IBE mainly profits from a less complex architecture and increased ease of usability. Nevertheless, the major drawback of IBE seems to be the inherent key escrow property. In order to get around this issue, distributed key generation seems a promising solution. However, before diving into distributed key generation protocols, some knowledge on secret sharing is appropriate.

3.4.1 Definition

Definition 3.2 (Secret Sharing Scheme). A *Secret Sharing Scheme* is a cryptographic scheme that divides a secret S into n pieces of data S_1, \dots, S_n called *shares*. Shares are distributed over n different parties called *shareholders* such that only specific subsets of the distributed shares allow reconstruction of the original secret S .

Definition 3.3 (Threshold scheme). A (t, n) *threshold scheme* ($t \leq n$) is a secret sharing scheme by which a trusted party securely distributes n different shares S_i to

n different parties P_i for $1 \leq i \leq n$ such that any subset of t or more different shares S_i easily allows to reconstruct the original secret S . Knowledge of $t - 1$ or less shares is insufficient to reconstruct the original secret S .

Definition 3.4 (Perfect threshold scheme). A (t, n) threshold scheme is said to be *perfect* if no subset of fewer than t shareholders can derive any partial information in the information theoretic sense about the original secret S even with infinite computational resources.

3.4.2 Shamir Secret Sharing

In 1979, both Shamir [72] and Blakley [13] independently found an algorithm achieving perfect threshold secret sharing. Shamir's solution was based on polynomial interpolation while Blakley's algorithm relied on finite geometries. Blakley secret sharing uses more bits than necessary as it describes multidimensional planes. In contrast, Shamir secret sharing requires as many bits for each share as the length of the original secret. Therefore Shamir secret sharing has gained more popularity in both research communities and in practical implementations.

Algorithm 6 Shamir's (t, n) threshold scheme [65]

Goal: A trusted party T distributes shares of a secret S to n parties.

Result: If a subset of at least t out of n shareholders collaborates, they can reconstruct the original secret S .

1. *Setup* The trusted party T begins with a secret integer $S \geq 0$ it wishes to distribute among n parties
 - a) T chooses a prime $p > \max(S, n)$ and defines $a_0 = S$
 - b) T selects $t - 1$ random, independent coefficients $a_1, \dots, a_{t-1}, 0 \leq a_j \leq p - 1$ defining the random polynomial over \mathbb{Z}_p , $f(x) = \sum_{j=0}^{t-1} a_j x^j$
 - c) T computes $S_i = f(i) \bmod p, 1 \leq i \leq n$ and securely transfers the share S_i to shareholder P_i , along with a public index i .
2. *Reconstruction* Any group of t or more shareholders pool their shares. Their shares provide t distinct points $(x, y) = (i, S_i)$ allowing computation of the coefficients $a_j, 1 \leq j \leq t - 1$ of $f(x)$ by Lagrange interpolation. The secret is recovered by calculating

$$f(0) = \sum_{i=1}^t y_i \prod_{1 \leq j \leq t, j \neq i} \frac{x_j}{x_j - x_i} = S$$

The idea behind Shamir secret sharing is elegant in its simplicity. Any polynomial $f(x)$ of degree $t - 1$ is uniquely defined by t points lying on the polynomial. For example, it is possible to draw only one straight line between 2 different coordinates,

a quadratic is fully defined by 3 different coordinates and so on. If the trusted party randomly generates a polynomial of degree $t - 1$ it suffices to securely distribute one of n different coordinates on the curve to each party $P_i, 0 \leq i \leq n$. A subset of at least t different shareholders has to collaborate in order to reconstruct the original polynomial by interpolation. For security reasons the polynomial $f(x)$ is calculated in a finite field modulo a large prime number p . The complete mechanism of Shamir's threshold scheme can be found in Algorithm 6. The mechanism behind reconstruction in Algorithm 6 is explained because the coefficients of an unknown polynomial $f(x)$ of degree less than t , defined by points $(x_i, y_i), 1 \leq i \leq t$ are given by the Lagrange interpolation formula

$$f(x) = \sum_{i=1}^t y_i \prod_{1 \leq j \leq t, j \neq i} \frac{x - x_j}{x_i - x_j}$$

A proof of this formula is omitted but can be found in [80].

3.4.3 Verifiable Secret Sharing

Verifiable secret sharing [30] tries to ensure the participating parties that their received shares are consistent by providing a verification mechanism. This verification mechanism can either detect an unfair dealer during setup or participants submitting incorrect shares during the reconstruction phase. The first verifiable secret sharing schemes were *interactive*, i.e. interaction between shareholders and the trusted party was required to verify their shares. In *non-interactive verifiable secret sharing* only the trusted party is allowed to send messages to the future shareholders. Shareholders can not communicate with each other neither can they send messages back to the trusted party. Non-interactive verifiable secret sharing is preferred over interactive alternatives as there is no chance of shareholders accidentally leaking too much information.

Popular verifiable secret sharing schemes are Feldman's scheme [37] and Benaloh's scheme [11]. No further details are given as a basic notion of verifiable secret sharing suffices for the remainder of this text.

3.5 Distributed Key Generation

Distributed key generation is inspired on secret sharing. The idea behind distributed key generation is that a secret s can be shared among n shareholders without the requirement for a trusted party. In this way a secret can be negotiated between all shareholders without any of the shareholders explicitly computing the secret. The major advantage of such a scheme is that no party in the scheme requires a higher level of trust like the trusted party T in the Shamir secret sharing scheme. Similarly to the Shamir secret sharing scheme a group of t or more shareholders will need to pool their shares in order to reconstruct the secret s .

3.5.1 Definition

Definition 3.5 (Distributed key generation scheme). A *distributed key generation scheme* is a (t, n) perfect threshold scheme ($t \leq n$) that requires no trusted party. That is, a distributed key generation scheme is a cryptographic scheme that negotiates a secret s with n different parties P_1, \dots, P_n by letting each party P_i distribute shares s_{ij} of its own private secret s_i with all other parties P_i where $1 \leq i \leq n, 1 \leq j \leq n$. At least t out of n parties will need to collude in order to compute the original secret s explicitly.

3.5.2 Commitment Schemes

To fully appreciate most distributed key generation schemes the reader should be familiar with commitment schemes. A commitment scheme is a cryptographic scheme that will allow to commit to a certain value while keeping the exact value private. The value can be revealed later giving anyone who received the commitment the possibility to check whether the value was changed between the commitment phase and the revealing of the value.

More formally, a generic commitment scheme is composed of three probabilistic polynomial time algorithms:

CS.Setup(1^λ): On input of a security parameter λ , generates the public parameters *params* of the system

CS.Commit(*params*, m , r): Returns a commitment $c_{m,r}$ to a message m and a random binary sequence r .

CS.Open(*params*, $c_{m,r}$, m' , r'): On input of public parameters *params*, a commitment $c_{m,r}$, a message m and a random binary sequence r it returns **true** if $c_{m,r} \leftarrow \text{CS.Commit}(\text{params}, m, r)$ with $m = m'$ and $r = r'$ and **false** otherwise.

For a more elaborate discussion on commitment schemes the reader is referred to the original paper formalising commitment schemes from Brassard et al. [22].

3.5.3 Pedersen Distributed Key Generation

The first usable distributed key generation protocol was defined by Pedersen [67]. A later publication from Gennaro et al. [42] proves the Pedersen scheme to be insecure in its original form in the presence of malicious Distributed Key Generators (DKGs).

Although the Pedersen scheme [67] is proven insecure, it is most instructive to describe the protocol in its original form as later schemes like the one from Gennaro [42] extensively rely on the same concepts. Therefore, the original Pedersen protocol is shown in Algorithm 7. **TODO for Algorithm 8: Complete it after fully understanding it**

Algorithm 7 Pedersen's distributed key generation [67]

Goal: A secret s is negotiated with n uniquely numbered parties $\{P_1, \dots, P_n\}$ without any of the parties explicitly computing the secret s .

Result: If a subset of at least t out of n parties colludes, they can reconstruct the original secret s .

1. *Setup* At initialisation, a setup algorithm $\langle p, g \rangle \leftarrow \mathcal{G}(1^\lambda)$ is executed that returns a large prime number p and a generator g of \mathbb{Z}_p on input of a security parameter λ . After execution of $\mathcal{G}(1^\lambda)$ each party $P_i, 1 \leq i \leq n$ should do the following:
 - a) P_i generates a random secret key $s_i \in \mathbb{Z}_p$ and publishes the corresponding public key $pk_i = g^{s_i}$
 - b) P_i chooses $t-1$ random independent coefficients $a_{i,1}, \dots, a_{i,t-1}, 0 \leq a_{i,j} \leq p-1$ defining a random polynomial $f_i(x)$ over \mathbb{Z}_p , $f_i(x) = \sum_{b=0}^{t-1} a_{i,j} x^b$.
 - c) P_i commits to the coefficients $a_{i,1}, \dots, a_{i,t-1}, 0 \leq a_{i,j} \leq p-1$ by broadcasting $A_{ib} = g^{a_{i,b}} \bmod p$ for $b = 1, \dots, t$ to all other parties.
 - d) P_i computes the share $s_{ij} = f_i(j) \bmod p$ and securely transfers the share s_{ij} to party P_j along with a signature $S_{P_i}(s_{ij})$ authenticating the share. P_i keeps s_{ii} to itself.
 - e) P_i verifies for each share s_{ji} received from P_j whether it is consistent by verifying that

$$g^{s_{ji}} = \prod_{b=0}^{n-1} (A_{jb})^{i^b} \bmod p$$

If the check fails for an index j , P_i broadcasts a complaint against P_j along with the received share s_{ij} and its signature $S_{P_j}(s_{ij})$. If a party receives t complaints, he is excluded from the set of participating parties \mathcal{Q} .

2. *Reconstruction* Any group of t or more shareholders pool their shares. Their shares provide t distinct points $(x, y) = (i, S_i)$ allowing computation of the coefficients $a_j, 1 \leq j \leq t-1$ of $f(x)$ by Lagrange interpolation. The secret is recovered by calculating

$$f(0) = \sum_{i=1}^t y_i \prod_{1 \leq j \leq t, j \neq i} \frac{x_j}{x_j - x_i} = S$$

Outsider Anonymous Identity-Based Broadcast Encryption

4.1 Security Model

4.1.1 Threat Model

4.1.2 Goals

4.2 Proposed Scheme

4.2.1 Scheme

4.2.2 Security Proof

4.2.3 Evaluation

4.3 Conclusion

Implementation

5.1 Existing Solutions

5.2 Anonymous Identity-Based Broadcasting Implementation

5.2.1 Implemented Scheme

5.2.2 Data Structures

5.3 Distributed Key Generation Implementation

5.3.1 Implemented Scheme

5.3.2 Data Structures

5.4 Evaluation

5.5 Performance Analysis

5.6 Conclusion

Conclusion

The final chapter contains the overall conclusion. It also contains suggestions for future work and industrial applications.

Appendices

A

Installing and Executing the Code

Appendices hold useful data which is not essential to understand the work done in the master thesis. An example is a (program) source. An appendix can also have sections as well as figures and references[?].

A.1 Setting up the DKG

A.2 Setting up Scramble

B

The Last Appendix

Appendices are numbered with letters, but the sections and subsections use arabic numerals, as can be seen below.

B.1 Lorem 20-24

B.2 Lorem 25-27



Bibliography

- [1] Pairing based cryptography. Master's thesis, Technische Universiteit Eindhoven, 2004.
- [2] Public-key encryption with oblivious keyword search. priced oblivious transfer. Master's thesis, KU Leuven, 2008.
- [3] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In Shoup [74], pages 205–222.
- [4] G. Adj, A. Menezes, T. Oliveira, and F. Rodríguez-Henríquez. Weakness of $x_{25}^{36 \cdot 509}$ for discrete logarithm cryptography. In Cao and Zhang [27], pages 20–44.
- [5] J. Baek, J. Newmarch, R. Safavi-naini, and W. Susilo. A survey of identity-based cryptography. In *Proc. of Australian Unix Users Group Annual Conference*, pages 95–102, 2004.
- [6] M. Barbosa and P. Farshim. Efficient identity-based key encapsulation to multiple parties. In N. P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 428–441. Springer, 2005.
- [7] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.
- [8] A. Barth, D. Boneh, and B. Waters. Privacy in encrypted content distribution using private broadcast encryption. In G. D. Crescenzo and A. D. Rubin, editors, *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 52–64. Springer, 2006.
- [9] G. Barthe, B. Grégoire, S. Héraud, F. Olmedo, and S. Z. Béguelin. Verified indifferentiable hashing into elliptic curves. *Journal of Computer Security*, 21(6):881–917, 2013.

- [10] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, editors, *ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.
- [11] J. C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret sharing. In A. M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 1986.
- [12] G. Birkhoff and S. MacLane. *A Survey of Modern Algebra*. The Macmillan Comp., 1965.
- [13] G. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, Monval, NJ, USA, 1979. AFIPS Press.
- [14] A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. *IACR Cryptology ePrint Archive*, 2012:52, 2012.
- [15] D. Boneh. The decision diffie-hellman problem. In J. Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998.
- [16] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.
- [17] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In Cramer [32], pages 440–456.
- [18] D. Boneh, X. Ding, G. Tsudik, and C.-M. Wong. A method for fast revocation of public key certificates and security capabilities. In D. S. Wallach, editor, *USENIX Security Symposium*. USENIX, 2001.
- [19] D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *IACR Cryptology ePrint Archive*, 2001:90, 2001.
- [20] X. Boyen. A tapestry of identity-based encryption: practical frameworks compared. *IJACT*, 1(1):3–21, 2008.
- [21] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.
- [22] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

- [23] J. Bullas. 22 social media facts and statistics you should know in 2014. URL: <http://www.jeffbullas.com/2014/01/17/20-social-media-facts-and-statistics-you-should-know-in-2014/>, last checked on 2014-05-08.
- [24] J. Callas, L. Donnerhackle, H. Finney, D. Shaw, and R. Thayer. OpenPGP Message Format. RFC 4880 (Proposed Standard), Nov. 2007. Updated by RFC 5581.
- [25] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
- [26] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. *IACR Cryptology ePrint Archive*, 2003:83, 2003.
- [27] Z. Cao and F. Zhang, editors. *Pairing-Based Cryptography - Pairing 2013 - 6th International Conference, Beijing, China, November 22-24, 2013, Revised Selected Papers*, volume 8365 of *Lecture Notes in Computer Science*. Springer, 2014.
- [28] A. D. Caro, V. Iovino, and G. Persiano. Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts. *IACR Cryptology ePrint Archive*, 2010:197, 2010.
- [29] L. Chen and Z. Cheng. Security proof of sakai-kasahara’s identity-based encryption scheme. *IACR Cryptology ePrint Archive*, 2005:226, 2005.
- [30] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *FOCS*, pages 383–395, 1985.
- [31] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-damgård revisited: How to construct a hash function. In Shoup [74], pages 430–448.
- [32] R. Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.
- [33] C. Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In K. Kurosawa, editor, *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2007.
- [34] Y. Dodis and N. Fazio. Public key broadcast encryption for stateless receivers. In J. Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002.
- [35] L. Ducas. Anonymity from asymmetry: New constructions for anonymous hibe. In J. Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 2010.

- [36] N. Fazio and I. M. Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. *IACR Cryptology ePrint Archive*, 2012:129, 2012.
- [37] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *FOCS*, pages 427–437. IEEE Computer Society, 1987.
- [38] A. Fiat and M. Naor. Broadcast encryption. In D. R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.
- [39] E. Fleischmann, M. Gorski, and S. Lucks. Some observations on indifferentiability. *IACR Cryptology ePrint Archive*, 2010:222, 2010.
- [40] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61. Springer, 2010.
- [41] G. Frey, M. Müller, and H.-G. Rück. The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1719, 1999.
- [42] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *J. Cryptology*, 20(1):51–83, 2007.
- [43] C. Gentry. Practical identity-based encryption without random oracles. In S. Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.
- [44] C. Gentry and B. Waters. Adaptive security in broadcast encryption systems. *IACR Cryptology ePrint Archive*, 2008:268, 2008.
- [45] O. Goldreich. On the foundations of modern cryptography. In B. S. K. Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 46–74. Springer, 1997.
- [46] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.
- [47] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.
- [48] M. T. Goodrich, J. Z. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In M. K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 511–527. Springer, 2004.
- [49] M. Groves. MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY). RFC 6509 (Informational), Feb. 2012.

-
- [50] M. Groves. Sakai-Kasahara Key Encryption (SAKKE). RFC 6508 (Informational), Feb. 2012.
 - [51] D. Halevy and A. Shamir. The lsd broadcast encryption scheme. In M. Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2002.
 - [52] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai. Identity-based hierarchical strongly key-insulated encryption and its application. In B. K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 495–514. Springer, 2005.
 - [53] K. Jones. The growth of social media v2.0. URL: <http://www.searchenginejournal.com/growth-social-media-2-0-infographic/77055/>, last checked on 2014-05-08.
 - [54] A. Joux. A new index calculus algorithm with complexity $l(1/4+o(1))$ in very small characteristic. *IACR Cryptology ePrint Archive*, 2013:95, 2013.
 - [55] A. Joux and K. Nguyen. Separating decision diffie-hellman from computational diffie-hellman in cryptographic groups. *J. Cryptology*, 16(4):239–247, 2003.
 - [56] L. Krzywiecki, P. Kubiak, and M. Kutylowski. A revocation scheme preserving privacy. In H. Lipmaa, M. Yung, and D. Lin, editors, *Inscrypt*, volume 4318 of *Lecture Notes in Computer Science*, pages 130–143. Springer, 2006.
 - [57] K. Lee and D. H. Lee. New techniques for anonymous hibe with short ciphertexts in prime order groups. *TIIS*, 4(5):968–988, 2010.
 - [58] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.
 - [59] A. B. Lewko, A. Sahai, and B. Waters. Revocation systems with very small private keys. *IACR Cryptology ePrint Archive*, 2008:309, 2008.
 - [60] B. Libert, K. G. Paterson, and E. A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 206–224. Springer, 2012.
 - [61] B. Libert and J.-J. Quisquater. Efficient revocation and threshold pairing based cryptosystems. In E. Borowsky and S. Rajsbaum, editors, *PODC*, pages 163–171. ACM, 2003.
 - [62] L. Martin and M. Schertler. Using the Boneh-Franklin and Boneh-Boyen Identity-Based Encryption Algorithms with the Cryptographic Message Syntax (CMS). RFC 5409 (Informational), Jan. 2009.

- [63] U. M. Maurer and S. Wolf. Lower bounds on generic algorithms in groups. In K. Nyberg, editor, *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 72–84. Springer, 1998.
- [64] U. M. Maurer and S. Wolf. The relationship between breaking the diffie-hellman protocol and computing discrete logarithms. *SIAM J. Comput.*, 28(5):1689–1721, 1999.
- [65] A. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
- [66] D. Naor, M. Naor, and J. B. Lotspiech. Revocation and tracing schemes for stateless receivers. *IACR Cryptology ePrint Archive*, 2001:59, 2001.
- [67] T. P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In D. W. Davies, editor, *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 522–526. Springer, 1991.
- [68] M. K. R Sakai, K Ohgishi. Cryptosystem based on pairing over elliptic curve (in Japanese). In *The 2001 Symposium on Cryptography and Information Security, Oiso, Japan, January, 2001*.
- [69] T. Ristenpart, H. Shacham, and T. Shrimpton. Careful with composition: Limitations of indifferentiability and universal composability. *IACR Cryptology ePrint Archive*, 2011:339, 2011.
- [70] R. Sakai and J. Furukawa. Identity-based broadcast encryption. *IACR Cryptology ePrint Archive*, 2007:217, 2007.
- [71] R. Sakai and M. Kasahara. Id based cryptosystems with pairing on elliptic curve. *IACR Cryptology ePrint Archive*, 2003:54, 2003.
- [72] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
- [73] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
- [74] V. Shoup, editor. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.
- [75] N. P. Smart. Efficient key encapsulation to multiple parties. In C. Blundo and S. Cimato, editors, *SCN*, volume 3352 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 2004.
- [76] StatisticBrain. Social networking statistics. URL: <http://www.statisticbrain.com/social-networking-statistics/>, last checked on 2014-05-08.

- [77] B. Waters. Efficient identity-based encryption without random oracles. In Cramer [32], pages 114–127.
- [78] B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.
- [79] A. Whitten and J. D. Tygar. Why johnny can’t encrypt: A usability evaluation of pgp 5.0. In G. W. Treese, editor, *USENIX Security*. USENIX Association, 1999.
- [80] P. Wiki. Lagrange interpolation formula. URL: http://www.proofwiki.org/wiki/Lagrange_Interpolation_Formula.
- [81] Worldometers. Worldometers real time world statistics. URL: <http://www.worldometers.info/>, last checked on 2014-05-08.
- [82] S. Yu, K. Ren, and W. Lou. Attribute-based on-demand multicast group setup with membership anonymity. *Computer Networks*, 54(3):377–386, 2010.
- [83] X. Zhang and K. Wang. Fast symmetric pairing revisited. In Cao and Zhang [27], pages 131–148.

Master thesis filing card

Student: Stijn Meul

Title: Practical Identity-Based Encryption for Online Social Networks

UDC: 621.3

Abstract:

Here comes a very short abstract, containing no more than 500 words. \LaTeX commands can be used here. Blank lines (or the command `\par`) are not allowed!

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Thesis submitted for the degree of Master of Science in Electrical Engineering,
option Embedded Systems and Multimedia

Thesis supervisors: Prof. dr. ir. Bart Preneel
Prof. dr. ir. Vincent Rijmen

Assessors: Prof. dr. ir. Claudia Diaz
Prof. dr. ir. Frank Piessens

Mentor: Filipe Beato