UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2009.1

SPACE-EFFICIENT IDENTITY-BASED ENCRYPTION

FINAL YEAR PROJECT

Student Patrícia Lustosa Ventura Ribeiro (plvr@cin.ufpe.br)
Advisor Ruy J. Guerra B. de Queiroz (ruy@cin.ufpe.br)

Recife, June 2009

Universidade Federal de Pernambuco
Centro de Informática

Patrícia Lustosa Ventura Ribeiro

SPACE-EFFICIENT IDENTITY BASED ENCRYPTION

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Prof. Dr. Ruy José Guerra Barretto de Queiroz*

Recife

2009

To my mother: Zélia Cristina

AKNOWLEDGMENTS

Not everything that is counted counts, and not everything that counts can be counted.
ALBERT EINSTEIN

Firstly, I would like to thank God for everything He gave me and for always being by my side, giving me strength whenever I needed it.

I would like to sincerely thank my family for all their support. My mother, to whom I dedicate this work, for teaching me since I was very young the importance of studying to have a better life. My father that always work hard to give us a good life. My brother and sisters, specially Juliana, just for existing in my life. My grandmother Zélia that always tried to help me the way she could. My godmother Duca, for always treating me like a daughter and for all weekends that she lost to be with me while I was studying.

I would like to thank my boyfriend Nitai for the incentive he gave me while I was doing this work. Also for his help, reading what I write and giving opinions. I would also like to thank him for his disposal in understanding my work, or at least a part of it.

I would like to thank my friends Daniel Dias, Milena Loureiro, Bruna Machado, Juliana Amaral, Edilson Quintela, for all the help, comprehension and friendship, since I was preparing myself for entering the university.

I would like to thank my coworkers from vEye Team for everything I learn from them and with them and for my professional growth during this year that we are working together.

Finally, I would like to thank my advisor Ruy de Queiroz for the support and for everything I learned from him, in his classes, that wasn't few. I would also like to thank all other professors that participate of my academic life, specially Paulo Borba and Sérgio Soares.

Far better it is to dare mighty things, to win glorious triumphs, even though checkered by failure, than to rank with those poor spirits who neither enjoy much nor suffer much, because they live in that grey twilight that knows neither victory nor defeat.
—THEODORE ROOSEVELT

É muito melhor arriscar coisas grandiosas, alcançar triunfo e glória, mesmo expondo-se a derrota, do que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito, porque vivem nessa penumbra cinzenta que não conhecem a vitória nem a derrota.
—THEODORE ROOSEVELT

ABSTRACT

In 1984, Shamir [1] proposed a public-key encryption scheme such that the public key could be an arbitrary string, in particular, some form of unique identity of the user. This kind of scheme is known as Identity-Based Encryption (IBE). Shamir's original motivation for constructing IBE was to simplify key management in email systems.

There are two basic approaches to the construction of IBE system. The first one, upon which Boneh-Franklin [2] scheme in based, builds IBE systems using bilinear maps [3,4,5]. The resulting systems are efficient both in performance and ciphertext length.

The second approach, due to Cocks [6], builds an elegant IBE system based on the quadratic residuosity problem modulo an RSA composite N. The ciphertext in this system contains two elements of $\mathbb{Z}/N\mathbb{Z}$ to each bit of the plaintext. Hence, the encryption of an l-bit message yields a ciphertext of size $2l \cdot log_2 N$ bits. For example, encrypting a 128-bit message using a 1024 bits modulo, the resulting ciphertext is of size 32678 bytes. For comparison, pairing based methods produce a 36 byte ciphertext.

An open problem since Cocks scheme was the construction of a space efficient IBE scheme without pairings, namely a system with short ciphertexts. In 2007, Boneh, Gentry and Hamburg [7] proposed such a system. In their scheme, the ciphertext size is about $l + log_2 N$. Encrypting a 128-bit message produces a ciphertext of size 145 bytes. The security of the system is based on the quadratic residuosity problem.

Encryption time in this system is quartic on the security parameter, while in the most part of practical public-key system the encryption is cubic on the security parameter.

The objective of this work is to study and spell out the Boneh-Gentry-Hamburg scheme.

RESUMO

Em 1984, Shamir [1] propôs um esquema de encriptação de chave pública tal que a chave pública pode ser uma string arbitrária, em particular, alguma forma de identificação do usuário. Esse tipo de esquema é chamado Identity-Based Enscryption (IBE). A motivação original de Shamir para IBE era simplificar o gerenciamento de certificados em sistemas de email.

Existem duas abordagens para a construção de sistemas IBE. A primeira delas, na qual o esquema Boneh-Franklin [2] é baseada, constrói sistemas IBE usando pareamentos bilineares [3,4,5]. Os sistemas resultantes são eficientes tanto em performance quanto em tamanho do cifrotexto.

A segunda abordagem, utilizada por Cocks [6], constrói um sistema IBE elegante baseado no problema padrão da residuosidade quadrática módulo um RSA composto N. Os cifrotextos nesse sistema contém dois elementos de $\mathbb{Z}/N\mathbb{Z}$ para cada bit do puro-texto. Assim, a encriptação de uma mensagem de l bits possui tamanho $2l \cdot log_2 N$. Por exemplo, encriptando uma mensagem de 128 bits usando um módulo de 1024 bits, o cifrotexto gerado possui 32678 bytes de tamanho. Em comparação, métodos baseados em pareamentos produzem um cifrotexto de 36 bytes para o mesmo nível de segurança.

Um problema em aberto desde o sistema de Cocks era a construção de um sistema IBE eficiente em espaço sem recorrer a pareamentos, ou seja, um sistema com cifrotexto curto. Em 2007, Boneh, Gentry e Hamburg [7] construíram um sistema com essas condições. O cifrotexto possui tamanho $l + log_2 N$. Encriptando uma mensagem de 128 bits, o resultado é um cifrotexto de tamanho 145 bytes. A segurança do sistema é baseada no problema da residuosidade quadrática.

O tempo de encriptação nesse sistema é quártico no parâmetro do segurança, enquanto na maior parte dos sistemas práticos de chave pública a encriptação é cúbica no parâmetro de segurança.

O objetivo desse trabalho é estudar o sistema de Boneh-Gentry-Hamburg.

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Our globalized society evolves very fast and with this grows the reliance in the Internet as the main mean of communication. With this, came the need of exchanging information that only the involved parties could have access to. This security is needed to send confidential emails, access your web account through the web, buy things online and many other activities that are becoming a habit in many people lives.

The older way to exchange information securely is cryptography. The first known use of cryptography was in 1900 BC. The earlier ciphers were much simpler. The main classical cipher types are transposition ciphers, which simply consist on rearranging the order of the letters of a message, and substitution ciphers, which systematically replace a letter or a group of letters with other letters or group of letters. With the advance of computers, and specially the internet, cryptography became mostly based in mathematical manipulation of the information.

Public-key cryptography is widely used in the internet. It consists of a pair of keys, namely a public key and a private key, where the public key is used to encrypt information and everybody can have access to it, while the private key is used to decrypt information and only the owner of the key can have access to it. Public-key cryptography requires a complex infra-structure to manage the keys.

In 1984, Adi Shamir proposed a new kind of cryptography called identity-based cryptography [1]. The main purpose was to avoid the need of maintaining a complex infra-structure as in public-key systems and thus simplify the use of cryptography in email systems. Identity-based systems uses some information that identifies the user uniquely as his public key, while the private key is generated by some entity called Private-key generator, or PKG.

Since the problem was proposed, many identity-based schemes were proposed [8,9,10,11,12], but none of them was fully satisfactory. The first fully functional scheme was proposed by Dan Boneh and Mathew Franklin in 2001 [2]. Their scheme was based on pairings over elliptic curves. Also in 2001, Clifford Cocks [6] proposed another IBE scheme that was based in the quadratic residuosity problem. The problem with his approach is that it produces large ciphertexts.

Since the creation of Cocks and Boneh-Franklin IBE, it was an open-problem to find a space-efficient IBE-scheme that was not based on pairings in elliptic curves. This problem was solved in 2007 when Boneh, Gentry and Hamburg proposed such a scheme [7]. The purpose of this work is to study their proposal.

1.1 DOCUMENT STRUCTURE

This work is divided in 5 chapters. Chapter 2 presents some preliminary concepts that are needed to understand what comes next. It includes some cryptographic concepts, such as the types of attack, symmetric and public-key cryptography. It also includes some mathematical concepts, such as modular arithmetic, rings, fields, quadratic residues, Legendre's and Jacobi's numbers.

Chapter 3 is about identity-based encryption. It shows the problems that motivates the creation of this new kind of cryptography and the main ideas besides it. It also presents the history of the schemes proposed and the notion of security appropriate to it.

Chapter 4 presents the Boneh-Gentry-Hamburg scheme. It shows an abstract single-bit encryption system and then transforms this system in order to encrypt multi-bit messages. The proof of security is also presented. Finally, it shows a concrete instantiation of the schemes defined before and algorithm to help the implementation of them.

The last chapter presents the conclusion and future works.

Chapter 2

PRELIMINARY CONCEPTS

God invented the integers; all else is the work of man.
LEOPOLD KRONECKER

Mathematics is the queen of the sciences and number theory is the queen of mathematics.
CARL FRIEDRICH GAUSS

In this chapter we present the preliminary concepts that a reader needs to know in order to understand the remaining chapters. The first section shows cryptography concepts such as types of attack, symmetric cryptography and public-key cryptography. The second section explains mathematical concepts such as modular arithmetic, groups, quadratic residues and quadratic residuosity problem.

2.1 CRYPTOGRAPHY

In this section, we present the types of attack and notions of symmetric and assymetric cryptography. We define plaintext as being the message itself that one wants to send and ciphertext as the "scrambled" information that is actually transmitted from the sender to the receiver.

2.1.1 Types of Attack

According to Katz and Lindell [13], the basic types of attack against encryption schemes are, in order of severity:

- Ciphertext-only attack: This is the most basic type of attack and refers to the scenario where the adversary just observes a ciphertext (or multiple ciphertexts) and attempts to determine the underlying plaintext (or plaintexts).
- Known-plaintext attack: Here, the adversary learns one or more pair of plaintext/ciphertext encrypted under the same key. The aim of the adversary is then to determine the plaintext associated with some other ciphertext (for which it doesn't know the corresponding plaintext).

- Chosen-plaintext attack: In this attack, the adversary has the ability to obtain the encryption of plaintexts of its choice. It then attempts to determine the plaintext that was encrypted in some other ciphertext.
- Chosen-ciphertext attack: The final type of attack is one where the adversary is even given the capability to obtain the decryption of ciphertexts of his choice. The adversary's aim, once again, is to determine the plaintext that was encrypted in some other ciphertext (whose decryption the adversary is unable to obtain directly).

Ciphertext indistinguishability is an important security property of many cryptographic schemes. Intuitively, if a cryptosystem possesses the property of indistinguishability, then an adversary is not able to distinguish the origin of a ciphertext even if he knows it came from either one of two plaintexts. Indistinguishability under a chosen plaintext attack is equivalent to the property of semantic security.

Notice that the idea of ciphertext indistinguishability is similar to the idea of the Turing Test [14]. This refers to the game proposed by Turing as a way of dealing with the question whether machines can think. Suppose that we have a person, a machine, and an interrogator. The interrogator is in a room separated from the other person and the machine. The object of the game is for the interrogator to determine which of the other two is the person, and which is the machine. The objective of the machine is to try to cause the interrogator to mistakenly conclude that the machine is the other person; the object of the other person is to try to help the interrogator to correctly identify the machine.

2.1.2 Symmetric Cryptography

Symmetric cryptography refers to any form where the same key is used to encrypt and decrypt the message. It is also known as secret key cryptography. To be more precise, the encryption and decryption keys don't have to be exactly the same but they do have to be trivially related: this means that they may be identical or there is a simple transformation to go between the two keys [15].

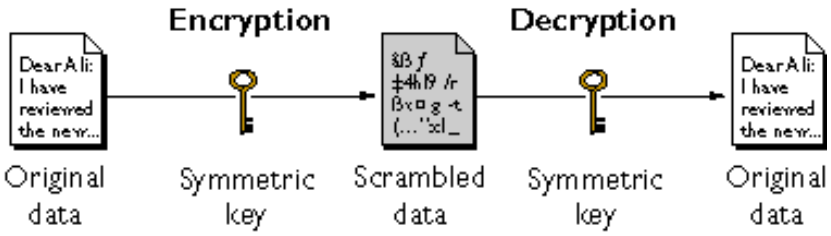

Figura 2-1 Symmetric Cryptography Scheme

Symmetric cryptography has been in use for thousands of years. One of the simplest forms is called Caesar cipher. It is a type of substation cipher in which each letter is replaced by a letter some fixed number of positions down the alphabet. This cipher dates to the first century before Christ.

An implicit assumption in any system using private-key cryptography is that the communication parties must have some way of initially sharing a key in a secret manner. Note that if one party sets the key and sends it to the other party over a public channel, an eavesdropper obtains the key too. If exists a secure channel which the parties could use to share the key, they didn't need to use cryptography: they could send the message through the secure channel. In military settings, this is not a severe problem because communication parties are able to physically meet in a secure location in order to agree upon a key. In many modern settings, however, parties cannot arrange any such physical meeting [13].

Note that the number of keys needed to be shared increases rapidly with the number of people wanting to communicate with each other. For example, assume that a company has 10 employees and they all need to communicate with each other. Each employee must have 9 keys, in order to communicate with the others. Once we have 10 employees, we should have to distribute 90 keys. As each key is distributed to 2 employees, we have 45 different keys to generate and distribute. If we double the number of employees, the number of keys became 190, more than four times bigger than before.

This is called the key management problem and it is the major problem with the use of symmetric cryptography in modern applications. This is a source of great concern and actually limits the applicability of cryptographic systems that rely solely on secret-keys methods.

2.1.3 Public-key cryptography

In order to solve the key management problem existent in symmetric cryptography, Whitfield Diffie and Martin Hellman [16] introduced the concept of public-key cryptography in 1976. Public-key cryptography is also known as asymmetric cryptography. In their system, each user has a pair of keys: one called the public key and the other called the private key. The public key is published while the private key is kept in secret.

The public key is used to encrypt data and the private key is used to decrypt ciphertexts. Hence, if Bob wants to send a secret message to Alice, he needs to know her public key, uses this key to encrypt the message and sends it off. When Alice receives the ciphertext encrypted by Bob, she will need her private key in order to decrypt it and read the original message. Anyone can send a confidential message by just using public information, but the message could only be decrypted with a private key, which is in the sole possession of the intended recipient.

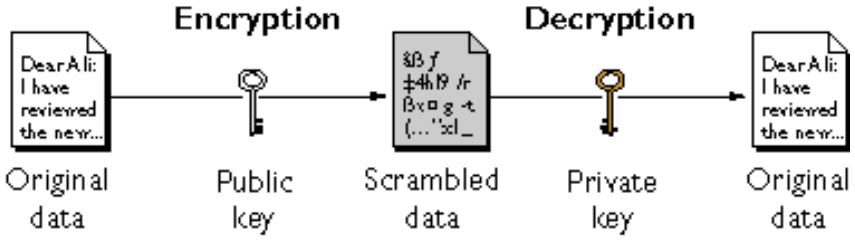

Figura 2-2 Public-key Cryptography Scheme

Note that the need for the sender and receiver to share secret information is eliminated. All the communication involves only the public key and the private key never need to be shared or transmitted. In this system, that is no longer necessary to trust the security of some means of communications.

In a public key system, the private key can always be mathematically derived from the public key. Hence, it is always possible to attack a public-key system by deriving the private key from the corresponding public key. Typically, the defense against this attack is to make the problem of deriving the private key as difficult as possible. For instance, some public-key systems require the attacker to factor a very large number, making the derivation of the key computationally infeasible [17].

Public-key systems are considerably slower than symmetric systems and are therefore not appropriate to large amounts of data. They also usually need a larger key to provide the

same level of security of symmetric systems. Because of this, it is often used a hybrid approach: a public-key algorithm is used to exchange the key and this key is used to transmit data using a symmetric algorithm.

Public keys should be associated with their users in a trusted manner. If it doesn't happen, an attacker can generate a public-private key pair and sends the public key to a user that wants to communicate with Alice as if it is Alice's public key. Once the attacker possesses the associated private key, he would be able to read the messages sent to Alice by that user.

The association of public keys to its owner is typically done by protocols implementing a public-key infrastructure. This allows the validity of the association to be formally verified by a trusted third party. This is usually done by a certificate authority and the association is called digital signatures.

In implementations of a traditional public-key system that uses digital certificates to manage public keys, a public-private key pair is generated randomly by the user. After it is created, the public key, along with the identity of the owner of the key, is digitally signed by a certificate authority to create a digital certificate that is then used to transport and manage the key [18].

The identity of a user is usually carefully verified before a digital certificate is issued to him, a process that is typically relatively expensive. The process of generating public-private key pairs can also be computationally expensive. Because generating keys and verifying user's identity can be expensive, digital certificates are often issued with fairly long validity periods, often between one and three years. Because of the relatively long validity period of the public keys managed by digital certificates, it is often necessary to check the key in a certificate for validity before using it.

The difficulty of maintaining the necessary public-key infrastructure motivated the creation of identity-based encryption, as we will see in the next chapter.

2.2 MATHEMATICS

In this section, we explore the mathematical concepts that are necessary to understand the following chapters. Most part of the content from this section was based on [13].

2.2.1 Basics Concepts from Number Theory

Number theory is the branch of pure mathematics concerned with the properties of numbers in general, and integers in particular, as well as the wider classes of problems that arise from their study.

The set of integers $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$ is denoted by the symbol $\mathbb{Z}$.

Definition 2.1 (Divisor): *If a and b are integers, then a divides b or a is a divisor of b if there exists an integer c such that $b = ac$. In this case we write $a|b$ and we say that a is a factor of b.*

Definition 2.2 (Prime): *An integer $p \geq 2$ is a prime if its only positive divisors are 1 and p.*

Definition 2.3 (Greatest Common Divisor): *A nonnegative integer d is the greatest common divisor of integers a and b if d is the largest positive integer that divides both a and b. This is denoted by $d = gcd(a, b)$.*

Definition 2.4 (Relatively Primes): *For integers a and b, if $gcd(a, b) = 1$ then we say that a and b are relatively primes.*

Definition 2.5 (Equivalence Relation): *An equivalence relation is, loosely, a binary relation on a set that specifies how to split up (i.e. partition) the set into subsets such that every element of the larger set is in exactly one of the subsets.*

Definition 2.6 (Equivalence Class): *Given a set X and an equivalence relation $\sim$ on X, the equivalence class of an element a in X is the subset of all elements in X which are equivalent to a: $[a] = \{x \in X | x \sim a\}$.*

2.2.2 Modular Arithmetic

Modular arithmetic is a system of arithmetic of integers, where numbers "wrap-around" after they reach a certain value – the modulus. Modular arithmetic can be handled in mathematics by introducing a congruence relation of the integers.

Definition 2.7 (Congruence): *Given three integers a, b and m, we say that 'a is congruent to b modulo m' and write $a \equiv b \bmod m$, if the difference $a - b$ is divisible by m. m is called the modulus of the congruence.*

Like any congruence relation, congruence modulo n is an equivalence relation, and the equivalence class of the integer a, denoted by $\bar{a}_n$, is the set $\{\dots, a-2n, a-n, a, a+n, a+2n, \dots\}$. This set, consisting of the integers congruent to a modulo n, is called the **congruence class** or **residue class** of a modulo n.

Definition 2.8 ($\mathbb{Z}/N\mathbb{Z}$): *The set of congruence classes modulo n is denoted as $\mathbb{Z}/N\mathbb{Z}$ (or, alternatively, $\mathbb{Z}/N$ or $\mathbb{Z}_N$) and defined by:*

$$\mathbb{Z}/N\mathbb{Z} = \{\bar{a}_n | a \in \mathbb{Z}\}$$

Definition 2.9 (Inverse): *If for a given integer b there exists an integer b^{-1} such that $bb^{-1} = 1 \bmod N$, we say that b^{-1} is a (multiplicative) inverse of b modulo N and call b invertible modulo N.*

Definition 2.10 (Division Modulo N): *When b is invertible modulo N we define division by b modulo N as a multiplication by $b^{-1} \bmod N$ (i.e.: $a/b \stackrel{\text{def}}{=} ab^{-1} \bmod N$).*

We stress that division by b is only defined when b is invertible. If $ab = cb \bmod N$ and b is invertible, then we may divide each side of the equation by b to obtain:

$$ab = cb \bmod N \Rightarrow (ab)b^{-1} = (cb)b^{-1} \bmod N \Rightarrow a = c \bmod N$$

Proposition 2.11: *Let a, N be integers, with $N > 1$. Then a is invertible modulo N if and only if $gcd(a, N) = 1$.*

2.2.3 Groups

Definition 2.12 (Group): *A group is a set* $\mathbb{G}$ *along with a binary operation* $+$ *for which the following conditions hold:*

- Closure*: For all* $g, h \in \mathbb{G}$, $g + h \in \mathbb{G}$*;*
- Existence of an Identity*: There exists an identity* $e \in \mathbb{G}$ *such that for all* $g \in \mathbb{G}$, $e + g = g = g + e$*;*
- Existence of Inverses*: For all* $g \in \mathbb{G}$ *there exists an element* $h \in \mathbb{G}$ *such that* $g + h = e = h + g$*. Such an* h *is called an inverse of* g*.*
- Associativity*: For all* $g_1, g_2, g_3 \in \mathbb{G}$, $(g1 + g2) + g3 = g1 + (g2 + g3)$.

Definition 2.13 (Abelian Group): *A group* $\mathbb{G}$ *with an operation* $+$ *is abelian if the following holds:*

- *Commutativity: For all* $g, h \in \mathbb{G}$, $g + h = h + g$.

Definition 2.14 (Cyclic Group): *A group* $\mathbb{G}$ *is called cyclic if there exists an element* g *in* $\mathbb{G}$ *such that* $\mathbb{G} =< g > \{g^n \mid n \text{ is an } integer\}$.

Definition 2.15 (Order of a Group): *The order of a group* $\mathbb{G}$ *is the number of elements of the set, must known as the cardinality.*

Definition 2.16 (Isomorphism): *Let* $\mathbb{G}$, $\mathbb{H}$ *be groups with respect to the operations* $\circ_{\mathbb{G}}$, $\circ_{\mathbb{H}}$, *respectively. A function* $f: \mathbb{G} \to \mathbb{H}$ *is an isomorphism from* $\mathbb{G}$ *to* $\mathbb{H}$ *if:*

- f *is a bijection*
- *For all* $g_1, g_2 \in \mathbb{G}$ *we have* $f(g_1 \circ_{\mathbb{G}} g_2) = f(g_1) \circ_{\mathbb{H}} f(g_2)$.

If there exists as isomorphism from $\mathbb{G}$ *to* $\mathbb{H}$ *then we say that these groups are isomorphic and write this as* $\mathbb{G} \simeq \mathbb{H}$.

2.2.4 Properties of $\mathbb{Z}/N\mathbb{Z}$

Proposition 2.17: *Let* $N \geq 2$ *be an integer. The set* $\mathbb{Z}/N\mathbb{Z}$ *with respect to addition modulo N is an abelian group.*

Proof: Closure is obvious. Asssociativity and commutativity follow from the fact that the integers satisfy the properties. The identity is 0. Since $a + (N - a) = 0 \, mod \, N$, it follows that the inverse of any element a is $[(N - a) \, mod \, N]$. □

In order to define a group structure over the set $\mathbb{Z}/N\mathbb{Z}$ using multiplication as the binary operation, we have to eliminate those elements in this set that are not invertible. As we saw in proposition 2.11, an element is invertible if $gcd(a, N) = 1$. We define $\mathbb{Z}_N^*$ as the set of elements of $\mathbb{Z}/N\mathbb{Z}$ that are invertible.

Definition 2.18 ($\mathbb{Z}_N^*$): $\mathbb{Z}_N^* \stackrel{\text{def}}{=} \{a \in \mathbb{Z}/N\mathbb{Z} \mid gcd(a, N) = 1\}$.

Proposition 2.19: *Let $N > 1$ be an integer. Then $\mathbb{Z}_N^*$ is an abelian group under multiplication modulo N.*

Theorem 2.20: *If p is a prime, then $\mathbb{Z}_p^*$ is cyclic.*

Proposition 2.21: The order of $\mathbb{Z}_p^*$ is $p - 1$.

Theorem 2.22 (Chinese Remainder Theorem): *Let $N = pq$ where p and q are relative prime. Then*

$$\mathbb{Z}_N \simeq \mathbb{Z}_p \times \mathbb{Z}_q \text{ and } \mathbb{Z}_N^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*.$$

Moreover, let f be a function mapping elements $x \in \{0, \dots, N - 1\}$ to pairs (x_p, x_q) with $x_p \in \{0, \dots, p - 1\}$ and $x_q \in \{0, \dots, q - 1\}$ defined by

$$f(x) \stackrel{\text{def}}{=} ([x \, mod \, p], [x \, mod \, q]).$$

Then f is an isomorphism from $\mathbb{Z}_N$ to $\mathbb{Z}_p \times \mathbb{Z}_q$ as well as an isomorphism from $\mathbb{Z}_N^*$ to $\mathbb{Z}_p^* \times \mathbb{Z}q*$.

2.2.5 Quadratic Residuosity

Definition 2.23 (Quadratic Residue): *Given a group $\mathbb{G}$, an element $y \in \mathbb{G}$ is a quadratic residue if there exists a $x \in \mathbb{G}$ with $x^2 = y$. We call x a square root of y. An element that is not a quadratic residue is called a quadratic non-residue.*

Proposition 2.24: *Let $p > 2$. Every quadratic residue in $\mathbb{Z}_p^*$ has exactly two square roots.*

Let $sqp: \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ be the function $sqp(x) \stackrel{\text{def}}{=} [x^2 \ mod \ p]$. The above proposition show that sqp is a two-to-one function when $p > 2$ is prime. This immediately implies that exactly half the elements of $\mathbb{Z}_p^*$ are quadratic residues. We denote the set of quadratic residues modulo p by $\mathcal{QR}_p$, and the set of quadratic non-residues by $\mathcal{QNR}_p$. We have just seen that for $p > 2$ prime

$$\left|\mathcal{QR}_p\right| = \left|\mathcal{QNR}_p\right| = \frac{|\mathbb{Z}_p^*|}{2} = \frac{p-1}{2}$$

We want to characterize the quadratic residues in $\mathbb{Z}_p^*$ for $p > 2$ prime. We begin with the fact that $\mathbb{Z}_p^*$ is a cyclic group of order $p - 1$, according to theorem 2.20 and proposition 2.21. Let g be a generator of $\mathbb{Z}_p^*$. This means that

$$\mathbb{Z}_p^* = \{g^0, g^1, \dots, g^{p-2}\}$$

(recall that p is odd, so $p - 1$ is even). Squaring each element in this list and reducing modulo $p - 1$ in the exponent yields a list of all quadratic residues in $\mathbb{Z}_p^*$:

$$\mathcal{QR}_p = \{g^0, g^2, \dots, g^{p-3}, g^0, g^2, \dots, g^{p-3}\}$$

Note that each quadratic residue appears twice in this list. We see that the quadratic residues in $\mathbb{Z}_p^*$: are exactly those elements that can be written as g^i with $i \in \{0,2, \dots, p-2\}$ an even integer.

2.2.6 Legendre and Jacobi Symbols

Definition 2.25 (Legendre Symbol): *Let $p > 2$ be prime and $x \in \mathbb{Z}_p^*$. The Legendre symbol of x modulo p, denoted by $\left(\frac{x}{p}\right)$, is defined by:*

$$\left(\frac{x}{p}\right) \stackrel{\text{def}}{=} \begin{cases} +1 \; if \; x \; is \; a \; quadratic \; residue \; modulo \; p \\ -1 \; if \; x \; is \; not \; a \; quadratic \; residue \; modulo \; p \end{cases}$$

Proposition 2.26: *Let $p > 2$ be a prime. Then $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \; mod \; p$.*

Proposition 2.27: *Let $p > 2$ be a prime and$x, y \in \mathbb{Z}_p^*$. Then $\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right)$.*

Proposition 2.28: *Let $N = pq$, with p and q distinct primes, and $y \in \mathbb{Z}_N^*$ with $y \leftrightarrow (y_p, y_q)$, where $y_p = y \; mod \; p$ and $y_q = y \; mod \; q$. Then y is a quadratic residue modulo N if and only if y_p is a quadratic residue modulo p and y_q is a quadratic residue modulo q.*

The above proposition characterizes the quadratic residues modulo N. A careful examination of the proof yields another important observation: each quadratic residue $y \in \mathbb{Z}_N^*$ has exactly four square roots. To see this, let $y \leftrightarrow (y_p, y_q)$ be a quadratic residue modulo N and let x_p and x_q be the square roots of y_p and y_q modulo p and q, respectively. Then the four square roots of y are given by the elements in $\mathbb{Z}_N^*$ corresponding to

$(x_p, x_q), (-x_p, x_q), (x_p, -x_q), (-x_p, -x_q)$.

Let $\mathcal{QR}_N$ denote the set of quadratic residues modulo N. Since squaring is a four-to-one function, we immediately see that exactly ¼ of the elements of $\mathbb{Z}_N^*$ are quadratic residues.

We define now a generalization of Legendre symbol called Jacobi symbol.

Definition 2.29 (Jacobi Symbol): *For any integer a and any positive odd integer n the Jacobi symbol is defined as the product of Legendre symbols corresponding to the prime factors of n:*

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1}\left(\frac{a}{p_2}\right)^{\alpha_2} \ldots \left(\frac{a}{p_k}\right)^{\alpha_k} \; where \; n = {p_1}^{\alpha_1}.{p_1}^{\alpha_2} \ldots {p_1}^{\alpha_k}.$$

In the special case where $N = pq$, with p and q odd primes, we have:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

We define $\mathcal{J}_N^{+1}$ as the set of elements in $\mathbb{Z}_N^*$ having Jacobi symbol $+1$ and define $\mathcal{J}_N^{-1}$ analogously.

We know from proposition 2.28 that if x is a quadratic residue modulo N, then $x \bmod p$ and $x \bmod q$ are quadratic residues modulo p and q, respectively. That is

$$\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = +1$$

So $\left(\frac{x}{N}\right) = +1$ and the next proposition becomes trivial.

Proposition 2.30: *If x is a quadratic residue modulo N, then $\left(\frac{x}{N}\right) = +1$.*

However, $\left(\frac{x}{N}\right) = +1$ can also occur when $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1$; that is, when both $x \bmod p$ and $x \bmod q$ are not quadratic residues modulo p and q. We introduce the notation $\mathcal{QNR}_N^{+1}$ for the set of elements of this type. That is

$$\mathcal{QNR}_N^{+1} \stackrel{\text{def}}{=} \{x \in \mathbb{Z}_N^* \mid x \text{ is not a quadratic residue modulo } N \text{ and } \left(\frac{x}{N}\right) = +1\}$$

Proposition 2.31: *Let $N = pq$ with p, q distinct, odd primes. Then*

1. *Exactly half the elements of $\mathbb{Z}_N^*$ are in $\mathcal{J}_N^{+1}$.*
2. *$\mathcal{QR}_N$ is contained in $\mathcal{J}_N^{+1}$.*
3. *Exactly half the elements of $\mathcal{J}_N^{+1}$ are in $\mathcal{QR}_N$ (the other half are in $\mathcal{QNR}_N^{+1}$).*

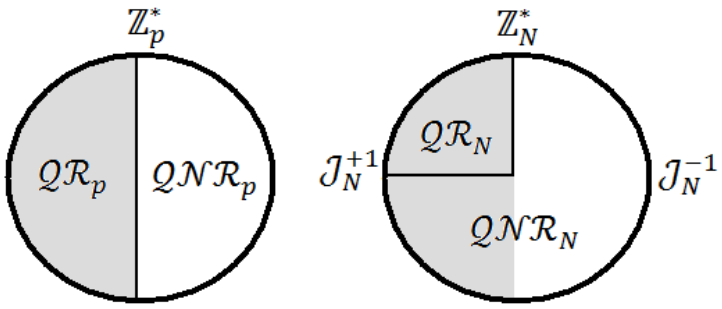

Figura 2-3 The structure of $\mathbb{Z}_p^*$ and $\mathbb{Z}_N^*$

Proposition 2.32: *Let $N = pq$ be a product of distinct, odd primes, and $x, y \in \mathbb{Z}_N^*$. Then* $\left(\frac{xy}{N}\right) = \left(\frac{x}{N}\right)\left(\frac{y}{N}\right)$.

2.2.7 Quadratic Residuosity Assumption

We showed in proposition 2.26 a simple method for deciding whether a given input x is a quadratic residue modulo a prime p. Can we adapt it to work modulo a composite number N? If the factorization of N is known and $N = pq$, we could simply compute $\left(\frac{x}{p}\right)$ and $\left(\frac{x}{q}\right)$ and if $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = +1$, x is a quadratic residue modulo N; if not, x is not a quadratic residue.

When the factorization of N is unknown, however, there is no know polynomial-time algorithm for deciding whether a given x is a quadratic residue modulo N or not. However, a polynomial-time algorithm is known for computing $\left(\frac{x}{N}\right)$ without the factorization of N. This leads to a partial test of quadratic residuosity: if, for a given input x, it holds that $\left(\frac{x}{N}\right) = -1$, then x is not a quadratic residue. This test says nothing in the case $\left(\frac{x}{N}\right) = +1$, and it is widely believed that there does not exist any polynomial-time algorithm for deciding quadratic residuosity in this case.

Definition 2.33 (Quadratic Residuosity): *We say deciding quadratic residuosity is hard relative to $RSAgen$ if for all probabilistic polynomial-time algorithms $\mathcal{A}$ there exists a negligible function $negl$ such that*

$$|\, Pr[\mathcal{A}(n, qr) = 1] - Pr[\mathcal{A}(n, qnr) = 1]| \leq negl(n).$$

where in each case the probabilities are taken over the experiment in which $RSAgen$ is run to give (N, p, q), qr is chosen at random from $\mathcal{QR}_N$ and qnr is chosen at random from $\mathcal{QNR}_N^{+1}$.

Chapter 3

IDENTITY BASED ENCRYPTION

If you have ambition, you might not achieve anything, but without ambition, you are almost certain not to achieve anything.
WHITFIELD DIFFIE

Identity-based encryption was first proposed by Adir Shamir in 1984[1]. The objective was to avoid the need to maintain the complex infra-structure of key management that exists in public-key systems and thus making encryption in e-mail systems easier. The scheme is based on public-key cryptography with an extra benefit: the user chooses some unique identifier as its public key instead of generating a random public/private key. This unique identifier can be his/her name, social number security, email address, or any other information that uniquely identifies him/her.

According to Martin[18], IBE systems are very similar to public-key systems in many aspects, but it also has significant differences. In public-key systems, a public-key certificate has all the information needed to encrypt the message. In IBE, a user needs to get a set of public key parameters from a trusted third party. Once he/she has these public parameters, he/she can use it to calculate public keys for any user he/she wants and uses it to encrypt the messages.

The recipient of an IBE-encrypted message needs his/her private key to decrypt the message. In order to obtain his/her private key, the recipient must authenticate himself/herself to a private key generator (PKG), a trusted third party that calculates the private keys. The PKG uses the identity of the user together with some secret information, called a master secret, to calculate the private key for that user. After that, the private key can be securely sent to the user.

In public-key cryptography, public-key certificates have a preset expiration date. This can be made in IBE-systems by setting the public key as the identifier concatenated with the current year. The user can only use his/her private key during that year. After that, he/she needs to obtain a new private key. Note that a user who wants to communicate with him/her does not have to obtain his/her public key every time his/her private key expires.

A problem with this approach is that during the validating period of some key, there is no way to revoke that key. To solve this problem, IBE systems typically use short-lived keys. This is not as precise as having the ability of revoking immediately a key but it makes key validation trivial [18].

An interesting use of identity-based encryption is to send messages in the future [19]. The public key can be defined as the email address concatenated with the date. Thus, one could send an email that the recipient could only read in the future, in the date specified by the sender. This can also be used in some companies to protected information that must be available to only few people during a period of time, but that after some date or hour can be made public.

In IBE schemes there are four algorithms that are responsible for creating and using a public/private key pair. They are called *Setup*, *KeyGen*, *Encryption* and *Decryption*.

The *Setup* algorithm initializes the system parameters (also known as public parameters or PP) and the master key. Intuitively, the system parameters will be publicly known, while the master key will be kept in secret by the PKG. The *KeyGen* algorithm takes as inputs the master secret and the identity and returns the private key associated with this identity.

The *Encryption* algorithm takes as input the public parameters, the identity of the receiver and the message and returns the corresponding ciphertext. The *Decryption* algorithm takes as input the private key and a ciphertext and returns the original message.

3.1 SECURITY NOTIONS

Chosen ciphertext security (IND-CCA), as defined in chapter 2, is the standard acceptable model of security for public key schemes. Hence, it is natural to require this notion of security to identity-based encryption schemes. However, it is not enough for an IBE scheme to be IND-CCA secure [2]. The reason is that, when an adversary attacks an identity, he might already have the corresponding private keys to other identities. The system should remain secure despite the adversary being able to obtain private keys for any identity of his/her choice (other than the identity being attacked). The adversary is also allowed to choose the identity being attacked.

An IBE scheme may also be required to be anonymous. It means that the ciphertext reveals nothing about the identity of the user used to create it. The following IBE security game [7] captures chosen ciphertext security, private key queries and anonymity:

Setup: The challenger runs $Setup(\lambda)$ and gives the adversary the resulting public parameters PP. It keeps the master-key (MSK) to itself. We set ID_0^*, $ID_1^* \leftarrow \perp$ and $C^* \leftarrow \perp$.

Queries: The adversary can issue adaptive queries of the following types:

- *Private key query ‹ID_i ›*: the challenger returns the resulting private key $d_i = KeyGen(MSK, ID_i)$ to the adversary. ID_i must be different from both ID_0^* and ID_1^*.
- *Decryption query (ID_i ,C_i)*: the challenger responds by running $KeyGen(MSK, ID_i)$ to obtain the private key d_i and *Decrypt*(d_i,C_i) to obtain the plaintext and then sends it to the adversary. (ID_i , C_i) must be different from both (ID_0^*, C^*) and (ID_1^*, C^*).
- *A single encryption query* $((ID_0 , m_0), (ID_1 , m_1))$: ID_0, ID_1 are distinct from all previous key queries and m_0, m_1 are two equal length plaintexts. The challenger picks a random bit $b \overset{R}{\leftarrow} \{0,1\}$ and sets

$$C^* \leftarrow Encrypt(PP, ID_b, m_b),\ ID_0^* \leftarrow ID_0 , ID_1^* \leftarrow ID_1$$

  It sends C^* to the adversary.

Guess: Eventually, the adversary outputs $b' \overset{R}{\leftarrow} \{0,1\}$. The adversary wins if $b = b'$.

Note that, initially, there is no value set to ID_0^*, ID_1^* and C^*. So the adversary can make arbitrary private key queries and decryption queries. When he/she does his single encryption query, the two identities he/she chose must be different from all previous identities he queried for private keys. After that, the private key queries and decryption queries have restrictions to avoid the adversary to obtain the private key for ID_0^* or ID_1^* and to decrypt the challenge ciphertext with one of these two identities.

We call the adversary $\mathcal{A}$ and define its advantage in attacking the scheme $\mathcal{E}$ as

$$\text{IBEAdv}_{\mathcal{A},\mathcal{E}(\lambda)} = \left|\Pr[b = b'] - \frac{1}{2}\right|$$

We subtract $\frac{1}{2}$ because the adversary already has this chance of guessing the right value of b by simply choosing a random bit. So, IBEAdv represents the chance that the adversary

has to win the game without taking into consideration the probability of he randomly guessing.

An adversary in this game is called an ANON-IND-ID-CCA adversary. ANON stands for anonymous, IND stands for indistinguishability, ID refers to the ability of the adversary to make private key queries and CCA stands for chosen ciphertext attack.

We will also consider three types of weaker adversaries:

- If $\mathcal{A}$ makes no decryption queries we say that $\mathcal{A}$ is an ANON-IND-ID-CPA adversary. This models an anonymous IBE under a chosen plaintext attack.
- If in the single encryption query the adversary uses $ID_0 = ID_1$, then we say that the adversary is an IND-ID-CCA adversary. This models a chosen ciphertext secure IBE that is not necessarily anonymous.
- If $\mathcal{A}$ makes no decryption queries and uses $ID_0 = ID_1$ we say that $\mathcal{A}$ is IND-ID-CPA adversary. This is the standard IBE security model under a chosen plaintext attack.

A single encryption query with different identities, as in the original game, guarantees that the system is anonymous. This happens because, if the adversary was able to extract some information about the ID from the ciphertext, he/she could use this information to find out the right value of b, increasing his chance of winning the game. On the other hand, if we change the game to use the same identity for both m_0 e m_1 in the encryption query, there is no guarantee that the ciphertext reveals no information about the identity used to create it. Hence, the second type of adversary is not anonymous.

Definition 3.1: *Let $\mathcal{S}$ be one of {IND-ID-CPA, IND-ID-CCA, ANON-IND-ID-CPA, ANON-IND-ID-CCA}. We say that an IBE system $\mathcal{E}$ is $\mathcal{S}$-secure if for all polynomial time $\mathcal{S}$ adversaries $\mathcal{A}$ we have that $IBEAdv_{\mathcal{A},\mathcal{E}(\lambda)}$ is a negligible function.*

3.2 HISTORY

In 1984, Adi Shamir [1] introduced a novel type of cryptography scheme, which enables any pair of users to communicate securely and to verify each other's signature without exchanging private or public keys, without keeping keys directory, and without using the services of a third party. This type of scheme became known as identity-based encryption and

signature schemes. In the same article, Shamir proposed a concrete implementation of identity-based signature scheme. However, he couldn't find an identity-based cryptosystem, though he conjectures that they exist.

Since the problem was posed by Shamir, there have been several proposals for identity-based encryption schemes [8,11,12]. S. Tsuji and T. Itoh proposed a scheme based in the discrete logarithm problem [10]. Hatsukazu Tanaka also proposed a system based in the same problem [9]. None of these schemes were fully satisfactory.

In 2001, the first fully functional identity-based encryption scheme was finally proposed by Dan Boneh and Matthew Franklin [2]. Their work is based on bilinear maps between groups and they prove its security based on the random oracle model. Boneh-Franklin IBE requires the calculation of a pairing, an expensive calculation that accounts for almost all the computation required for decryption and most of the computation required for encryption. Besides that, the assumptions about the hardness of problems in certain elliptic curves groups are relatively new compared to others cryptographic assumptions.

In the same year, Clifford Cocks [6] invented another IBE scheme. The security of Cocks IBE is based on both the computational difficulty of integer factorization and on the quadratic residuosity problem. Cocks IBE is efficient with respect to time but is not space efficient. Ciphertexts in this system contain two elements of $\mathbb{Z}/N\mathbb{Z}$ for each bit of the plaintext. Therefore the encryption of an l-bit message is $2l.log_2 N$ long.

In 2004, Dan Boneh and Xavier Boyen [3] proposed a scheme that is secure under the standard model, i.e. which does not rely on random oracles. Boneh-Boyen is also based in pairings on elliptic curves. Sakai-Kasahara IBE was proposed in [4] and has the advantage of being more efficient than the previous IBE schemes.

Because of the uncertainty about the cryptographic assumption about pairings in elliptic curves and the space inefficiency of the only practical IBE scheme that was not based on elliptic curves [6], it was an open problem to find a space efficient IBE scheme that wasn't based on pairings on elliptic curves. In 2007, the problem was solved by Dan Boneh, Craig Gentry and Michael Hamburg [7]. They proposed an IBE scheme that is based on the theory of ternary quadratic forms. The security of their system relies on the quadratic residuosity problem, as in Cocks system. The scheme is space efficient, though encryption is slower than in the Cocks system. The study and understanding of this scheme is the main purpose of this work. The description and security proof of it is given in the following chapters.

Chapter 4

BONEH, GENTRY, HAMBURG SCHEME

The best system is to use a simple, well understood algorithm which relies on the security of a key rather than the algorithm itself. This means if anybody steals a key, you could just roll another and they have to start all over.
ANDREW CAROL

In this chapter, we explore the Boneh-Gentry-Hamburg scheme [7]. As we saw in the last chapter, specialists were looking for a space efficient IBE scheme that was not based in pairings. Boneh, Gentry and Hamburg found such a scheme. In their system, the encryption on an l-bit message consists of a single element in $\mathbb{Z}/N\mathbb{Z}$ plus $(l+1)$ additional bits. Hence, ciphertext size is about $l + log_2 N$.

To construct the IBE system, we are going to start constructing a deterministic algorithm with the following properties.

Definition 4.1. *Let $\mathcal{Q}$ be a deterministic algorithm that takes as input (N, R, S) where $N \in \mathbb{Z}^+$ and $R, S \in \mathbb{Z}/N\mathbb{Z}$. The algorithm outputs two polynomials $f, g \in \mathbb{Z}/N\mathbb{Z}[x]$. We say that $\mathcal{Q}$ is IBE compatible if the following two conditions hold:*

- *Condition 1: If R and S are quadratic residues, then $f(r)g(s)$ is a quadratic residue for all square roots r of R and s of S.*
- *Condition 2: If R is a quadratic residue, then $f(r)f(-r)S$ is a quadratic residue for all square roots r of R.*

Condition 1 implies that the Legendre symbol $\left(\frac{f(r)}{N}\right)$ is equal to $\left(\frac{g(s)}{N}\right)$. This happens because the product being a quadratic form implies that the Jacobi symbol $\left(\frac{f(r)g(s)}{N}\right)$ is 1. Since $\left(\frac{f(r)g(s)}{N}\right) = \left(\frac{f(r)}{N}\right).\left(\frac{g(s)}{N}\right)$ and the possible values for $\left(\frac{f(r)}{N}\right)$ and $\left(\frac{g(s)}{N}\right)$ is 1 and -1, $\left(\frac{f(r)}{N}\right)$ must be equal to $\left(\frac{g(s)}{N}\right)$. This fact will be used during decryption. Condition 2 will be used to prove security.

We begin by describing a simple IBE for one bit messages as a warm up to the main IBE construction. Then we are going to see the multi-bit abstract IBE system. After that, we are going to show the security proof of the multi-bit abstract IBE. Finally, we are going to

make the abstract system concrete, by showing a concrete instantiation of the IBE compatible algorithm Q.

4.1 SINGLE BIT ENCRYPTION

We are going to describe the scheme by showing its four algorithms: Setup, KeyGen, Encrypt and Decrypt.

$\boldsymbol{Setup}(\boldsymbol{\lambda})$: generate $(p,q) \leftarrow RSAgen(\lambda)$, $N \leftarrow pq$ and a random $u \overset{R}{\leftarrow} \mathcal{J}_N^{+1} \setminus \mathcal{QR}_N$. Output public parameters $PP = (N, u, H)$ where H is a hash function $H : \mathcal{ID} \rightarrow \mathcal{J}_N^{+1}$. The master key MSK is the factorization of N.

$\boldsymbol{KeyGen}(\boldsymbol{MSK}, \boldsymbol{ID})$: generate a private key by first setting $R \leftarrow H(ID)$. If $R \in \mathcal{QR}_N$ set $r \leftarrow R^{1/2}$ and otherwise set $r \leftarrow (uR)^{1/2}$. Output r as the private key for ID.

$\boldsymbol{Encrypt}(\boldsymbol{PP}, \boldsymbol{ID}, \boldsymbol{m})$: to encrypt $m \in \{\pm 1\}$ with public key ID pick a random $s \in \mathbb{Z}/N\mathbb{Z}$ and compute $S \leftarrow s^2$. Let $R \leftarrow H(ID)$. Run $\mathcal{Q}$ twice:

$$(f, g) \leftarrow \mathcal{Q}(N, R, S) \text{ and } (\bar{f}, \bar{g}) \leftarrow \mathcal{Q}(N, uR, S)$$

and encrypt m using the two Jacobi symbols: $c \leftarrow m \cdot \left(\frac{g(s)}{N}\right)$ and $\bar{c} \leftarrow m \cdot \left(\frac{\bar{g}(s)}{N}\right)$. Output the ciphertext $C \leftarrow (S, c, \bar{c})$.

$\boldsymbol{Decrypt}(\boldsymbol{C}, \boldsymbol{r})$: decrypt $(S, c, \bar{c})$ using private key r. Let us first suppose that $R = H(ID)$ is in $\mathcal{QR}_N$ so that $r^2 = R$. The decryptor runs $\mathcal{Q}(N, R, S)$ to obtain (f, g). By condition (1) of Definition 4.1 we know that

$$\left(\frac{g(s)}{N}\right) = \left(\frac{f(r)}{N}\right)$$

Note that, since $\left(\frac{g(s)}{N}\right) = \left(\frac{f(r)}{N}\right)$ and the possible values for Jacobi symbols are 1 and -1, $c \cdot \left(\frac{g(s)}{N}\right) = m \cdot \left(\frac{g(s)}{N}\right) \cdot \left(\frac{f(r)}{N}\right) = m$. Hence the plaintext is obtained by setting $m \leftarrow c \cdot \left(\frac{g(s)}{N}\right)$. If R is a non-residue then uR is a quadratic residue and $r^2 = uR$. We decrypt by running $\mathcal{Q}(N, uR, S)$ and recovering m from $\bar{c}$. Since $\mathcal{Q}$ is deterministic, both sender and receiver always obtain the same pairs (f, g) and $(\bar{f}, \bar{g})$.

An adversary trying to break the system would have access to the ciphertext, that includes S, c and $\bar{c}$, and maybe would know the identity of the user used to create it. In order to recover the plaintext, he would have to know $\left(\frac{g(s)}{N}\right)$ or $\left(\frac{f(r)}{N}\right)$.

The information he have about s is that $s^2 = S$ and the value of S. If he knew the factorization of N, he could calculate s easily, as we saw in section 2.2.7. But he doesn't possess the factorization of N and, by the quadratic residuosity assumption, he is not able to differ a residue number from a non-residue. So, he can't find s, because if he could, he would be able to differ a residue number from a non-residue and the quadratic residuosity assumption would not be valid. In the same way, the adversary only knows R and can't find r.

4.2 MULTIBIT ENCRYPTION

We describe now the scheme that encrypts multi-bit messages, called BasicIBE.

$\boldsymbol{Setup}(\boldsymbol{\lambda})$: Generate $(p, q) \leftarrow RSAgen(\lambda)$, $N \leftarrow pq$ and a random $u \overset{R}{\leftarrow} \mathcal{J}_N^{+1} \setminus \mathcal{QR}_N$. Output public parameters $PP = (N, u, H)$ where H is a hash function $H : \mathcal{ID} \times [1, l] \rightarrow \mathcal{J}_N^{+1}$. The master key MSK is the factorization of N and a random key K for a pseudorandom function $F_K: \mathcal{ID} \times [1, l] \rightarrow \{0,1,2,3\}$.

$\boldsymbol{KeyGen}(\boldsymbol{MSK}, \boldsymbol{ID}, \boldsymbol{l})$: It generates a private key for encrypting l-bit messages. Takes as input the master secret key, the identity of the user and the length l. For $j = 1, \dots, l$ do:

$R_j \leftarrow H(ID, j) \in \mathcal{J}_N^{+1}$ and $w \leftarrow F_K(ID, j) \in \{0,1,2,3\}$

let $a \in \{0,1\}$ be such that $u^a R_j \in \mathcal{QR}_N$

let $\{z_0, z_1, z_2, z_3\}$ be the square roots of $u^a R_j$ in $\mathbb{Z}/N\mathbb{Z}$

Set $r_j \leftarrow z_w$

Output the decryption key $d_{ID} \leftarrow (PP, r_1, \dots, r_l)$. The PRF F guarantees that the same square roots is output for a given ID, but an adversary can't tell ahead of time which one will be output.

Notice that the master secret key (MSK) is used to find the four square roots of $u^a R_j$. Without the factorization, we wouldn't be able to determine whether $u^a R_j$ is a quadratic residue, which is easier than finding the square roots.

$\boldsymbol{Encrypt}(\boldsymbol{PP}, \boldsymbol{ID}, \boldsymbol{m})$: Takes as input the public parameters, the ID of the user and the message to be encrypted $m = m_1 \dots m_l \in \{\pm 1\}^l$. It picks a random $s \in \mathbb{Z}/N\mathbb{Z}$ and computes $S \leftarrow s^2$. For $j = 1, \dots, l$ do:

$$R_j \leftarrow H(ID, j), \quad (f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S) \quad \text{and} \quad (\bar{f}_j, \bar{g}_j) \leftarrow \mathcal{Q}(N, uR_j, S)$$

$$c_j \leftarrow m_j \cdot \left(\frac{g_j(s)}{N}\right) \quad \text{and} \quad \bar{c}_j \leftarrow m_j \cdot \left(\frac{\bar{g}_j(s)}{N}\right).$$

Set $c \leftarrow c_1 \dots c_l$ and $\bar{c} \leftarrow \bar{c}_1 \dots \bar{c}_l$ and output the ciphertext $C \leftarrow (S, c, \bar{c})$.

$\boldsymbol{Decrypt}(\boldsymbol{C}, \boldsymbol{d_{ID}})$: Takes as input the ciphertext C and the decryption key $d_{ID} = (PP, r_1, \dots, r_l)$. For $j = 1, \dots, l$ let $R_j \leftarrow H(ID, j)$ and do:

if $r_j^2 = R_j$ run $(f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S)$ and set $m_j \leftarrow c_j \cdot \left(\frac{f_j(r_j)}{N}\right)$

if $r_j^2 = uR_j$ run $(\bar{f}_j, \bar{g}_j) \leftarrow \mathcal{Q}(N, uR_j, S)$ and set $m_j \leftarrow \bar{c}_j \cdot \left(\frac{\bar{f}_j(r_j)}{N}\right)$

Output $m = m_1 \dots m_l$.

This completes the description of BasicIBE.

The same value of S can be used to encrypt the l bits of the message. To encrypt an l-bit message we hash ID multiple times by computing $R_i \leftarrow H(ID, i)$ for $i = 1, \dots, l$. Now each pair (S, R_i) can be used to encrypt one message bit. The length of the ciphertext is the size of S plus 2 bits for each message bit. Hence, when encrypting a l-bit message, the length of the ciphertext $(S, (c_1, c_1{}'), \dots, (c_l, c_l{}'))$ is $log_2 N + 2l$ bits.

Note that encryption and decryption in this system are similar to the ones in the single-bit scheme. The same arguments can be used to show that the message m is retrieved in the decryption by multiplying the ciphertext by $\left(\frac{f_j(r_j)}{N}\right)$ or $\left(\frac{\bar{f}_j(r_j)}{N}\right)$. Also in the same way, the adversary can´t find s and r, because he only knows S and R.

The hash function H, created in the setup step, outputs elements in $\mathcal{J}_N^{+1}$. We can easily implement this function using another hash function that outputs elements in $\mathbb{Z}/N\mathbb{Z}$. Choose an element $z \in \mathbb{Z}/N\mathbb{Z}$ such that $\left(\frac{z}{N}\right) = -1$. Let $x \leftarrow H'(ID, j)$. If $\left(\frac{x}{N}\right) = 1$, output $H(ID, j) = x$, otherwise output $H(ID, j) = xz$. In the second case, $\left(\frac{xz}{N}\right) = \left(\frac{x}{N}\right)\left(\frac{z}{N}\right) = (-1)(-1) = 1$. Either way, $\left(\frac{H(ID,j)}{N}\right) = 1$ and so $H(ID, j) \in \mathcal{J}_N^{+1}$ as required.

4.3 SECURITY

We now present the proof of security of the multi-bit abstract IBE presented. We start by proving a lemma that is needed in the security proof. Then we describe a public key system, called BasicPKE, which is semantically secure in the standard model under the quadratic residuosity assumption. After that, we deduce security of the IBE system. However, this requires the random oracle model.

Lemma 4.2: *Let $N = pq$ be as RSA modulus, $X \in \mathcal{QR}_N$ and $S \in \mathcal{J}_N^{+1}$. Let x be a random variable uniformly chosen from among the four square roots of X. Let f be a polynomial such that $f(x)f(-x)S$ is a quadratic residue for all four values of x. Then:*

- *when $S \notin \mathcal{QR}_N$ the Jacobi symbol $(f(x)/N)$ is uniformly distributes in $\{\pm 1\}$;*
- *when $S \in \mathcal{QR}_N$ then $(f(x)/N)$ is constant, namely the same for all four values of x.*

Proof: Let x_p be a square root of $X \bmod p$ and x_q a square root of $X \bmod q$. The four square roots of $X \bmod N$ are (x_p, x_q), $(-x_p, x_q)$, $(x_p, -x_q)$ and $(-x_p, -x_q)$. By the Chinese Remainder Theorem, we can find the elements of $\mathbb{Z}/N\mathbb{Z}$ corresponding to each of these pairs. Let $x = (x_p, x_q)$ and $x' = (x_p, -x_q)$. Then we have that the four square roots of X modulo N are $\{\pm x, \pm x'\}$. We know that $f(x)f(-x)S$ is a quadratic residue modulo N. It means that $f(x)f(-x)$S is a quadratic residue modulo p and modulo q. So, we have that $\left(\frac{f(x)f(-x)S}{\mathrm{p}}\right) = \left(\frac{f(x)}{\mathrm{p}}\right)\left(\frac{f(-x)}{\mathrm{p}}\right)\left(\frac{S}{\mathrm{p}}\right) = 1$ and the same on q. When $S \notin \mathcal{QR}_N$, $\left(\frac{S}{\mathrm{p}}\right) = -1$. This means that $\left(\frac{f(x)}{\mathrm{p}}\right) = -1.\left(\frac{f(-x)}{\mathrm{p}}\right)$ and the same on q. Because $x' = x \bmod p$ and $x' = -x \bmod p$, we have that $\left(\frac{f(x')}{\mathrm{p}}\right) = \left(\frac{f(x)}{\mathrm{p}}\right)$ and $\left(\frac{f(x')}{\mathrm{q}}\right) = -1.\left(\frac{f(x)}{\mathrm{q}}\right)$. So $\left(\frac{f(x')}{\mathrm{N}}\right) = -1.\left(\frac{f(x)}{\mathrm{N}}\right)$. So of the four values $f(x)$, $f(x')$, $f(-x)$, $f(-x')$, the first two must have different Jacobi symbols, as must the last two. Hence, among the four symbols, two are $+1$ and two are -1. When $S \in \mathcal{QR}_N$ all four symbols are equal. □

4.3.1 The public key system BasicPKE

The public key system (PKE) consists of four algorithms: a setup algorithm G to generate the common reference string C, a key generation algorithm K to generate public/private key pairs, an encryption algorithm E to encrypt messages by receiving the message and the public key and an decryption algorithm D to decrypt messages by receiving the ciphertext and the private key. This system allows multiple users to use the same modulus N, and N is treated as the common reference string C. If for some reason one doesn't want to share the same modulus to all users, then the pair of algorithms (G, K) can be viewed as a single key generation algorithm.

Algorithm $\boldsymbol{G}(\boldsymbol{\lambda})$: generate $(p, q) \leftarrow RSAgen(\lambda)$, $N \leftarrow pq$ and output N as the common reference string. The factorization of N is erased.

Algorithm $\boldsymbol{K}(\boldsymbol{N}, \boldsymbol{l})$: Takes as input the common reference string N and a message length l. For $j = 1, .., l$ pick a random $r_j \overset{R}{\leftarrow} \mathbb{Z}/N\mathbb{Z}$ and sets $R_j \leftarrow r_j^2$. It then outputs the public key $PK \leftarrow (R_1, \dots, R_l)$ and the private key $SK \leftarrow (N, r_1, \dots, r_l)$.

Algorithm $\boldsymbol{E}(\boldsymbol{N}, \boldsymbol{PK}, \boldsymbol{m})$: Takes as input the common reference string N, the public key $PK \leftarrow (R_1, \dots, R_l)$ and the message to be encrypted $m = m_1 \dots m_l \in \{\pm 1\}^l$. It picks a random $s \in \mathbb{Z}/N\mathbb{Z}$ and compute $S \leftarrow s^2$. For $j = 1, \dots, l$ do:

$$(f_j, g_j) \leftarrow Q(N, R_j, S) \quad \text{and } c_j \leftarrow m_j \cdot \left(\frac{g_j(s)}{N}\right).$$

Set $c \leftarrow c_1 \dots c_l$ and output the ciphertext $C \leftarrow (S, c)$.

Algorithm $\boldsymbol{D}(\boldsymbol{SK}, \boldsymbol{C})$: Takes as input the ciphertext $C \leftarrow (S, c_1, \dots, c_l)$ and the secret key $SK \leftarrow (N, r_1, \dots, r_l)$. For $j = 1, \dots, l$ do:

$$R_j \leftarrow r_j^2, \quad (f_j, g_j) \leftarrow Q(N, R_j, S) \quad \text{and} \quad m_j \leftarrow c_j \cdot \left(\frac{f_j(r_j)}{N}\right)$$

This completes the description of the PKE system BasicPKE. Note that the factorization of N can be erased in the generation algorithm, while in the IBE system the factorization of N is the MSK. This happens because, in the key generation algorithm of

BasicPKE, the factorization of N is not needed once the public/private key is generated at random.

We now prove that BasicPKE is semantically secure in the standard model. The standard security game starts by the challenger running the setup algorithm G and the key generation algorithm K and sending to the attacker $\mathcal{A}$ the common reference string C and the public key PK. The challenger than picks a random bit $b \overset{R}{\leftarrow} \{0,1\}$. Next, $\mathcal{A}$ gives the challenger two equal length messages m_0, m_1. The challenger runs the encryption algorithm for m_b and returns the challenge ciphertext $C^* \leftarrow E(C, PK, m_b)$. Finally, $\mathcal{A}$ outputs its guess b' for the bit b. $\mathcal{A}$ wins the game if $b = b'$. We refer to such adversary $\mathcal{A}$ as an IND-CPA adversary. We call $PKEAdv_{\mathcal{A},\mathcal{E}}$ the adversary's advantage in attacking the PKE scheme $\mathcal{E}$. We define $PKEAdv$ as

$$PKEAdv_{\mathcal{A},\mathcal{E}} = \left| Pr[b = b'] - \frac{1}{2} \right|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 4.3: *We say that a PKE system $\mathcal{E}$ is IND-CPA secure if for all polynomial time adversaries $\mathcal{A}$ we have that $PKEAdv_{\mathcal{E},\mathcal{A}}$ is a negligible function.*

The next lemma is the security proof of BasicPKE.

Lemma 4.4: *The PKE system $BasicPKE = (G, K, E, D)$ is IND-CPA secure in the standard model if the QR assumption holds for RSAgen. In particular, suppose $\mathcal{A}$ is a polynomial time IND-CPA adversary attacking BasicPKE. Then there exists an efficient QR algorithm $\mathcal{B}$ (whose running time is about the same as that of $\mathcal{A}$) such that*

$$PKEAdv_{\mathcal{A},BasicPKE\,(\lambda)} = QRAdv_{\mathcal{B},RSAgen\,(\lambda)}$$

Proof: This proof is by direct reduction to quadratic redisuosity assumption. Algorithm $\mathcal{B}$ is given a random tuple (N, V) where $N = p.q$, $(p, q) \leftarrow RSAgen(\lambda)$ and $V \in \mathcal{J}_N^{+1}$. It must determine whether V is a quadratic residue. In order to do that, algorithm $\mathcal{B}$ runs $\mathcal{A}$ and plays the role of the challenger to $\mathcal{A}$. Then, $\mathcal{B}$ sets N as the common reference string, runs $K(N, l)$ to obtain a public/private key pair (PK, SK) and sends (N, PK) to $\mathcal{A}$.

Adversary $\mathcal{A}$ chooses two messages m_0 and m_1, both with length l, and gives it to $\mathcal{B}$. $\mathcal{B}$ choose a $b \in \{0,1\}$ and creates the encryption of m_b using the given V as follows:

- Let $PK = (R_1, \dots R_l)$ and $SK = (r_1, \dots r_l)$
- Let $m_b = m_1 \dots m_l \in \{\pm 1\}^l$. For $u = 1, \dots l$, do:

$$(f_u, g_u) \leftarrow \mathcal{Q}(N, R_u, V) \text{ and } c_u \leftarrow m_u \left(\frac{f_u(R_u)}{N}\right)$$

Set $c \leftarrow c_1 \dots c_l$. The challenge ciphertext is $C^* \leftarrow (V, c)$. $\mathcal{B}$ sends C^* to $\mathcal{A}$.

Note that the creation of the ciphertext is done differently from the original scheme. In BasicPKE, $c_u \leftarrow m_u \left(\frac{g_u(v)}{N}\right)$, where $v^2 = V$. $\mathcal{B}$ doesn't know the square root of V, it doesn't even know if V has a square root. So, $\mathcal{B}$ sets $c_u \leftarrow m_u \left(\frac{f_u(R_u)}{N}\right)$.

$\mathcal{A}$ now outputs a guess b'. If $b = b'$, $\mathcal{B}$ outputs 1. Otherwise, $\mathcal{B}$ outputs 0. This completes the description of $\mathcal{B}$.

We argue that $\mathcal{B}$ breaks the QR assumption with the same advantage as $\mathcal{A}$ breaking the BasicPKE. This will follow from the following two claims:

Claim 1: When $(N, V) \overset{R}{\leftarrow} P_{QR}$ (i.e. V is uniformly distributed in $\mathcal{QR}_N$) then

$$\left|Pr[\mathcal{B}(N, V) = 1] - \frac{1}{2}\right| = \left|Pr[b = b'] - \frac{1}{2}\right| = PKEAdv_{\mathcal{A}, BasicPKE\,(\lambda)} \quad (1)$$

Proof: When V is uniformly distributed in $\mathcal{QR}_N$, $\mathcal{B}$ emulates perfectly the challenger. The common reference string N and public key PK are as in the real game. The challenger ciphertext is also equal to the one of the real game. $\mathcal{B}$ constructs the ciphertext by setting $c_u \leftarrow m_u \left(\frac{f_u(R_u)}{N}\right)$. Consider v as being a square root of V (v exists because $V \in \mathcal{QR}_N$. In the real game, the ciphertext is constructed by setting $c_u \leftarrow m_u \left(\frac{g_u(v)}{N}\right)$. Note that, by condition 1 of definition 4.1, $\left(\frac{g_u(v)}{N}\right) = \left(\frac{f_u(R_u)}{N}\right)$. So, for all $u = 1, \dots, l$, we have $c_u = m_u \left(\frac{g_u(v)}{N}\right) = m_u \left(\frac{f_u(R_u)}{N}\right)$. With this, we confirm that $\mathcal{B}$ emulates perfectly an IND-CPA challenger. $\left|Pr[\mathcal{B}(N, V) = 1] - \frac{1}{2}\right| = \left|Pr[b = b'] - \frac{1}{2}\right|$ because $\mathcal{B}$ outputs 1 only if $b = b'$.

$\left|Pr[b = b'] - \frac{1}{2}\right| = PKEAdv_{\mathcal{A}, BasicPKE\,(\lambda)}$ follows from the definition of $PKEAdv_{\mathcal{A}, BasicPKE\,(\lambda)}$, once $\mathcal{B}$ emulates perfectly the challenger. □

Claim 2: When $(N,V) \overset{R}{\leftarrow} P_{NQR}$ (i.e. V is uniformly distributed in $\mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$) then

$$Pr[\mathcal{B}(N,V) = 1] = Pr[b = b'] = \frac{1}{2} \quad (2)$$

Proof: When (N,V) are distributed in P_{NQR} we claim that the bit b is independent of $\mathcal{A}$'s view. In particular, we claim that the challenge ciphertext C^* is independent of b. To see why, consider the bit $c_u \leftarrow m_u \left(\frac{f_u(R_u)}{N}\right) \in \{\pm 1\}$ for all $u = 1, \ldots, l$. Recall that $PK = (R_1, \ldots R_l)$. The only information $\mathcal{A}$ has about r_u is the value of R_u. Hence, from $\mathcal{A}$'s view, r_u is uniformly distributed in the set of four square roots of R_u. Since $V \in \mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$, it follows from condition 2 of definition 4.1 and lemma 4.2 that the symbol $\left(\frac{f_u(R_u)}{N}\right)$ is uniformly distributed in $\{\pm 1\}$. Hence, c_u is independent of m_u. The same argument holds for all $u = 1, \ldots, l$ and therefore C^* is independent of the message being encrypted. Overall, when (N,V) are distributed in P_{NQR} we have $\Pr[\mathcal{B}(N,V) = 1] = \Pr[b = b'] = \frac{1}{2}$. □

By combining equations (1) and (2) we obtain $PKEAdv_{\mathcal{A},BasicPKE}(\lambda) = QRAdv_{\mathcal{B},RSAgen}(\lambda)$ as required. This completes the proof of lemma 4.4. □

4.3.2 Proof of Security

We now prove the security of BasicIBE in the random oracle model based on the QR assumption.

Theorem 4.5: *Suppose the QR assumption holds for RSAgen and F is a secure PRF. Then the system BasicIBE is IND-ID-CPA secure when H is modeled as a random oracle. In particular, suppose $\mathcal{A}$ is an efficient IND-ID-CPA adversary. Then there exist efficient algorithms $\mathcal{B}_1, \mathcal{B}_2$ (whose running time is about the same as that of $\mathcal{A}$) such that*

$$IBEAdv_{\mathcal{A},BasicIBE}(\lambda) \leq 2.QRAdv_{\mathcal{B}_1,RSAgen}(\lambda) + PRFAdv_{\mathcal{B}_2,F}(\lambda)$$

Proof: We present the proof as a sequence of games. We let W_i denote the event that the adversary $\mathcal{A}$ wins the game i.

Game 0: This game is identical to the one defined in section 3.1. Hence, we know that

$$\left|\Pr[W_0] - \frac{1}{2}\right| = IBEAdv_{\mathcal{A},BasicIBE}(\lambda)$$

The challenger chooses the random oracle $H : \mathcal{ID} \times [1,l] \rightarrow \mathcal{J}_N^{+1}$ at random from the set of all such functions.

Game 1: This game differs from the past game in the way it generates private keys. Instead of using a pseudorandom function F, the challenger uses a truly random function. If F is a secure pseudorandom function, the adversary won't notice the difference between the two games. In particular, there exists an algorithm $\mathcal{B}_2$ (whose running time is about the same as that of $\mathcal{A}$) such that

$$|\Pr[W_1] - \Pr[W_0]| = PRFAdv_{\mathcal{B}_2,F}(\lambda)$$

Game 2: In game 1 the public parameters given to $\mathcal{A}$ contain (N,u,H) where u is uniform in $\mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$, as in the original system. Moreover, the random oracle H is a random function $H : \mathcal{ID} \times [1,l] \rightarrow \mathcal{J}_N^{+1}$. In game 2, we change H in the following manner: H outputs $H(ID,j) = u^a v^2$ by choosing at random $a \overset{R}{\leftarrow} \{0,1\}$ and $v \overset{R}{\leftarrow} \mathbb{Z}/N\mathbb{Z}$. We know that $u \in \mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$ so $\left(\frac{u}{N}\right) = 1$. Since $v^2 \in \mathcal{QR}_N$, $\left(\frac{v^2}{N}\right) = 1$. If $a = 0$, $H(ID,j) = v^2 \in \mathcal{J}_N^{+1}$. If $a = 1$, $H(ID,j) = uv^2$ and we have $\left(\frac{uv^2}{N}\right) = \left(\frac{u}{N}\right).\left(\frac{v^2}{N}\right) = 1.1 = 1$. In both cases, we have $H(ID,j) \in \mathcal{J}_N^{+1}$. So, H implements a random function $H : \mathcal{ID} \times [1,l] \rightarrow \mathcal{J}_N^{+1}$.

Let $R_j \leftarrow H(ID,j)$ for some (ID,j). In game 1 the challenger responds to private key queries by outputting a random square root of R_j or uR_j for $j = 1, \ldots l$. In game 2 let $R_j \leftarrow H(ID,j) = u^a v^2$. The challenger responds to private key queries by outputting either ${R_j}^{1/2} = v$ (used if $a = 0$) or $\left(uR_j\right)^{1/2} = uv$ (used if $a = 1$) for $j = 1, \ldots, l$. Since v is uniform in $\mathbb{Z}/N\mathbb{Z}$, r_j is uniform in the set of the square roots of R_j or uR_j, just like in game 1. Because of this, from $\mathcal{A}$'s view, games 1 and 2 are identical and therefore

$$\Pr[W_2] = \Pr[W_1]$$

Note that in game 2 the challenger no longer needs the factorization of N to respond to $\mathcal{A}$'s queries. In game 1, the challenger needs it to calculate the square roots of R_j.

Game 3: We slightly modify game 2 by choosing a random u in $\mathcal{QR}_N$ instead of in $\mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$. The adversary won't notice any differences assuming QR assumption holds for

RSAgen, since it is the only change between the two games. In particular, there exists an efficient algorithm $\mathcal{B}_1$ such that

$$|\Pr[W_3] - \Pr[W_2]| = QRAdv_{\mathcal{B}_1,RSAgen}(\lambda)$$

We note that since $H(ID, j) = u^a v^2$ and $u \in \mathcal{QR}_N$, H will always output elements in $\mathcal{QR}_N$. Let u_0 be a square root of u.

Game 4: We slightly change the way that the challenger builds the ciphertext C^*. We pick C^* in a similar way to the one used in the proof of lemma 4.4. To respond to the encryption query (ID, m_0, m_1) from $\mathcal{A}$ the challenger chooses $b \overset{R}{\leftarrow} \{0,1\}$ and does:

$R_i \leftarrow H(ID, j) = u^{a_i}.v_i^2$ and $r_i \leftarrow u_0^{a_i}.v_i$ for $i = 1, \dots, l$

(then r_i is a root of R_i and $u_0 r_i$ is a root of uR_i)

$(*)$ $s \overset{R}{\leftarrow} \mathbb{Z}/N\mathbb{Z}$ and $S \leftarrow s^2$

write $m^{(b)} = m_1 \dots m_l \in \{\pm 1\}^l$

for $k = 1, \dots, l$ do:

$$(f_k, g_k) \leftarrow \mathcal{Q}(N, R_k, S) \quad \text{and} \quad (\bar{f}_k, \bar{g}_k) \leftarrow \mathcal{Q}(N, uR_k, S)$$

$(**)$ $$c_k \leftarrow m_k . \left(\frac{f_k(r_k)}{N}\right) \quad \text{and} \quad \bar{c}_k \leftarrow m_k . \left(\frac{\bar{f}_k(u_0 r_k)}{N}\right).$$

$c \leftarrow c_1 \dots c_l$ and $\bar{c} \leftarrow \bar{c}_1 \dots \bar{c}_l$. Send $\mathcal{A}$ the challenge ciphertext $C \leftarrow (S, c, \bar{c})$.

Since S, R_k, uR_k are all in $\mathcal{QR}_N$, we know by condition (1) of 4.1 that $\left(\frac{f_k(r_k)}{N}\right) = \left(\frac{g_k(s)}{N}\right)$ for all $k = 1, \dots, l$ and also $\left(\frac{\overline{f_k}(u_0 r_k)}{N}\right) = \left(\frac{\overline{g_k}(s)}{N}\right)$. Hence, the ciphertext C^* created in this way is identical to the challenge ciphertext created in game 3. Therefore,

$$Pr[W_3] = Pr[W_4]$$

It is important to note that s is not used in the creation of C^*.

Game 5: We slightly modify the challenger in game 4 by choosing S uniformly in $\mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$ instead of $\mathcal{QR}_N$. That is, we change the line marked with $(*)$ in Game 4 into

$(*)$ $S \overset{R}{\leftarrow} \mathcal{J}_N^{+1} \backslash \mathcal{QR}_N$

Since this is the only difference between the games, the adversary will not notice the difference, assuming the QR assumption holds for RSAgen. In particular, there exists an algorithm $\mathcal{B}_1$ such that

$$|\Pr[W_5] - \Pr[W_4]| = QRAdv_{\mathcal{B}_1,RSAgen}(\lambda)$$

Game 6: We can now change Game 5 and make the challenge ciphertext C^* be independent of the challenge bit b. We change the line marked $(**)$ in Game 4 as follows:

$$(**) \qquad z_k \overset{R}{\leftarrow} \{\pm 1\},\ c_k \leftarrow z_k \,.\left(\frac{f_k(r_k)}{N}\right) \quad \text{and} \quad \bar{c}_k \leftarrow z_k \,.\left(\frac{\bar{f}_k(u_0 r_k)}{N}\right).$$

As a result, the challenge ciphertext C^* is an encryption of a random message $z_1, \ldots, z_l$, independent of the bit b.

We argue that because S is a non-residue, Games 5 and 6 are indistinguishable due to Condition 2 of definition 4.1 and lemma 4.2. The argument is similar to the argument in the proof of lemma 4.4. The challenge ciphertext is created by using $2l$ elements $\{R_1, uR_1, \ldots, R_l, uR_l\}$ all in $\mathcal{QR}_N$. For each R the adversary does not know which of the four square roots of R is used in the creation of C^*. This is used to satisfy the condition of lemma 4.2 which says that r is a random variable chosen over the four square roots of R.

Consider now a specific $k \in \{1, \ldots, l\}$ and let $x = \left(\frac{f_k(r_k)}{N}\right)$ and $y = \left(\frac{\overline{f_k}(u_0 r_k)}{N}\right)$. Using condition (2) of definition 4.1 and lemma 4.2, we have that x is uniformly distributed in $\{\pm 1\}$ and the same happens with y. With this, we have that

$$Pr[(x,y) = (1,1)] = Pr[(x,y) = (-1,-1)] \text{ and}$$
$$Pr[(x,y) = (1,-1)] = Pr[(x,y) = (-1,1)]$$

It follows that the pair (x, y), which is an encryption of $+1$, is distributed identically as the pair $(-x, -y)$, which is an encryption of -1. Hence, in $\mathcal{A}$'s view, the bits $(ck, \overline{ck})$ are distributed identically whether the plaintext is $+1$ or -1. Since this holds for all $k = 1, \ldots, l$ it follows that C^* is distributed identically in Games 5 and 6. As a result, we have:

$$Pr[W_6] = Pr[W_5]$$

End. In game 6 we have the ciphertext as the encryption of a random message z. Because of this, we clearly have

$$Pr[W_6] = \frac{1}{2}$$

Combining the equations, we have that

$$IBEAdv_{\mathcal{A},BasicIBE}(\lambda) \leq 2.QRAdv_{\mathcal{B}_1,RSAgen}(\lambda) + PRFAdv_{\mathcal{B}_2,F}(\lambda)$$

Thus, the theorem is proved. □

4.4 CONCRETE INSTANTIATION

To make the IBE abstract system concrete, we need a concrete instantiation of the IBE compatible algorithm $\mathcal{Q}$. We present it now:

Algorithm $\mathcal{Q}(N, R, S)$:

1. Construct a solution $(x, y) \in (\mathbb{Z}/N\mathbb{Z})^2$ to the equation $Rx^2 + Sy^2 = 1$.
   In the next section, we describe algorithms for solving this equation, that is the main bottleneck in this system.
2. Output the polynomials $f(r) \leftarrow xr + 1$ and $g(s) \leftarrow 2ys + 2$.

We now show that $\mathcal{Q}$ is IBE compatible and satisfies the two conditions of definition 4.1. Let $R, S \in \mathbb{Z}/N\mathbb{Z}$. Let r be a square root of R and s a square root of S, if one exists. $\mathcal{Q}$ satisfies condition 1, since

$$\begin{aligned} f(r).g(s) &= (xr + 1)(2ys + 2) = 2xrys + 2xr + 2ys + 2 \\ &= 2xrys + 2xr + 2ys + 2 + (Rx^2 + Sy^2 - 1) \\ &= Rx^2 + Sy^2 + 2\text{xrys} + 2\text{xr} + 2\text{ys} + 2 - 1 \\ &= \text{r}^2x^2 + s^2y^2 + 2\text{xrys} + 2\text{xr} + 2\text{ys} + 1 \\ &= (\text{xr} + \text{ys} + 1)^2 (\text{mod N}) \end{aligned}$$

With this, we know that $f(r).g(s)$ is a quadratic residue and $\text{xr} + \text{ys} + 1$ is its square root.

Now we show that $\mathcal{Q}$ satisfies condition 2.

$$\begin{aligned} f(r)f(-r)S &= (xr + 1)(-xr + 1)S = (1 - x^2r^2)S \\ &= (1 - Rx^2)S = Sy^2S = S^2y^2 = (Sy)^2 \end{aligned}$$

With this, we know that $f(r)f(-r)S$ is a quadratic residue and Sy is its square root.

Hence we have a valid instantiation of $\mathcal{Q}$.

4.4.1 Algorithms

The instantiation of algorithm $\mathcal{Q}$ presented before requires the computation of to integers x and y that satisfies the equation

$$Rx^2 + Sy^2 = 1 \qquad (1)$$

where $R, S \in \mathbb{Z}/N\mathbb{Z}$. Recall that when encrypting a l-bit message using the abstract system, one must solve $2l$ of these equations. In particular, the encryptor must find solutions $(x_i, y_i), (\overline{x_i}, \overline{y_i}) \in (\mathbb{Z}/N\mathbb{Z})^2$ such that

$$R_i {x_i}^2 + S{y_i}^2 = 1 \quad \text{and} \quad (uR_i){x_i}^2 + S{y_i}^2 = 1 \tag{2}$$

For $i = 1, \dots, l$, the decryptor needs a solution to l of these equations, once he only needs to calculate $R_i {x_i}^2 + S{y_i}^2 = 1$ or $(uR_i){x_i}^2 + S{y_i}^2 = 1$ to each bit i.

4.4.1.1 A Product Formula

The encryptor needs solution to $2l$ equations in the form of (1), which are the $2l$ equations in (2). However, we show that solving $l + 1$ equations is sufficient. We do that by using a product formula that, given the solution of two equations, builds the solution to a third equation.

Lemma 4.6*: Suppose that (x_1, y_1) is a solution to $A_1x^2 + By^2 = 1$ and (x_2, y_2) is a solution to $A_2x^2 + By^2 = 1$. Then (x_3, y_3) is a solution to*

$$(A_1A_2)x^2 + By^2 = 1 \tag{3}$$

where $x_3 = \frac{(x_1x_2)}{By_1y_2+1}$ and $y_3 = \frac{(y_1+y_2)}{By_1y_2+1}$.

Proof: This proof is by direct substitution into (3):

$$(A_1A_2)x^2 + By^2 = 1$$

$$(A_1A_2)\left(\frac{(x_1x_2)}{By_1y_2 + 1}\right)^2 + B\left(\frac{(y_1 + y_2)}{By_1y_2 + 1}\right)^2 = 1$$

$$\frac{A_1A_2{x_1}^2{x_2}^2}{(By_1y_2)^2 + 2By_1y_2 + 1} + \frac{B({y_1}^2 + 2y_1y_2 + {y_2}^2)}{(By_1y_2)^2 + 2By_1y_2 + 1} = 1$$

$$A_1A_2{x_1}^2{x_2}^2 + B\left({y_1}^2 + 2y_1y_2 + {y_2}^2\right) = (By_1y_2)^2 + 2By_1y_2 + 1$$

$$A_1A_2{x_1}^2{x_2}^2 + B{y_1}^2 + 2By_1y_2 + B{y_2}^2 = (By_1y_2)^2 + 2By_1y_2 + 1$$

$$A_1A_2{x_1}^2{x_2}^2 + B{y_1}^2 + B{y_2}^2 = (By_1y_2)^2 + 1$$

$$A_1A_2{x_1}^2{x_2}^2 + (1 - A_1{x_1}^2) + (1 - A_2{x_2}^2) = (By_1y_2)^2 + 1$$

$$(A_1{x_1}^2 - 1)(A_2{x_2}^2 - 1) + 1 = (By_1y_2)^2 + 1$$

$$B{y_1}^2 . B{y_2}^2 = (By_1y_2)^2$$

$$(By_1y_2)^2 = (By_1y_2)^2$$ □

During encryption, one needs the solution to the following $l + 1$ equations:

$$ux^2 + Sy^2 = 1 \quad \text{and} \quad R_i{x_i}^2 + S{y_i}^2 = 1 \quad \text{for } i = 1, \dots, l \tag{4}$$

The encryptor uses lemma 4.6 to quickly find the solution to the remaining l equations in (2). Simply apply lemma 4.6 to the left equation in (4) and the ith right equation in (4) to obtain the solution to $(uR_i){x_i}^2 + S{y_i}^2 = 1$, as required.

Chapter 5

CONCLUSION

You're braver than you believe, stronger than you seem, and smarter than you think.
WINNIE THE POOH'S FRIEND CHRISTOPHER ROBIN

Dream big and dare to fail.
NORMAN VAUGHAN

We have described a space-efficient IBE scheme without pairings based on the quadratic residuosity assumption in the random oracle model. The abstract scheme makes use of an abstract algorithm $\mathcal{Q}$ that satisfies some properties. We have also described a concrete instantiation of the IBE compatible algorithm $\mathcal{Q}$. However, this concrete instantiation requires the generation of primes of order $\sqrt{N}$ during the encryption. It makes the encryption quartic in the security parameter, while the decryption is cubic in the security parameter. Most practical public-key systems, like RSA and the existing IBE schemes including Cocks' IBE, are cubic in the security parameter. Thus the encryption time in this scheme is not ideal. It is natural to look to other concrete instantiations of the IBE compatible algorithm $\mathcal{Q}$ that makes the scheme more time-efficient.

In [7], Boneh, Gentry and Hamburg described a modification of the original scheme, changing the abstract algorithm $\mathcal{Q}$ by adding new conditions that the algorithm must satisfy. The modified scheme is proved to be anonymous. They also present a concrete instantiation for this modified algorithm $\mathcal{Q}$.

As a future work, is proposed the implementation of the scheme described here. The study and implementation of the modified anonymous scheme is also a target. This would allow a major understanding of how they work and of the viability of applying them in the real world.

REFERENCES

[1] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO 1984*, volume 196 of LNCS, pages 47–53. Springer-Verlag, 1984.

[2] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. Extended abstract in Crypto ’01.

[3] Dan Boneh and Xavier Boyen. Efficient selective-ID identity based encryption without random oracles. In *Proceedings of Eurocrypt 2004*, LNCS, pages 223–238. Springer-Verlag, 2004.

[4] R. Sakai and M. Kasahara. ID based cryptosystems with pairing over elliptic curve. http://eprint.iacr.org/2003/054, 2003.

[5] B. Waters. Efficient identity-based encryption without random oracles. In *Proceedings of Eurocrypt 2005*, LNCS, 2005.

[6] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 26–8, 2001.

[7] D. Boneh, C. Gentry, and M. Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In *Proceedings of FOCS 2007*, pp. 647-657, 2007

[8] Y. Desmedt and J. Quisquater. Public-key systems based on the difficulty of tampering. in *Advances in Cryptology - Crypto '86*, Lecture Notes in Computer Science, Vol. 263. Springer-Verlag, pp. 111-117, 1986.

[9] H. Tanaka. A realization scheme for the identity-based cryptosystem. In *Advances in Cryptology* - Crypto '87, Lecture Notes in Computer Science, Vol. 293. Springer-Verlag, pp. 341-349,1987.

[10] S. Tsuji and T. Itoh. An ID-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communication*, vol. 7, no. 4, pp. 467-473. 1989.

[11] U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In *Advances in Cryptology - Crypto '91*, Lecture Notes in Computer Science, Vol. 547, Springer-Verlag, pp. 498-507,1991.

[12] D. Hühnlein, M. Jacobson, D. Weber, Towards Practical Non-interactive Public Key Cryptosystems Using Non-maximal Imaginary Quadratic Orders. In *Selected Areas in Cryptography*, Lecture Notes in Computer Science, Vol. 2012, Springer-Verlag, pp. 275-287, 2000.

[13] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2008.

[14] Stanford Encyclopedia of Philosophy. *The Turing Test*. Las modified in April 9th 2003. Available at <http://plato.stanford.edu/entries/turing-test/ >. Accessed in June 3rd 2009.

[15] Murdoch Mactaggart. *Introduction to Cryptography, Part 2: Symmetric cryptography*. Last modified in March 1st 2001. Available at <http://www.ibm.com/developerworks/library/s-crypt02.html>. Accessed in May 2nd 2009.

[16] Whitfield Diffie and Martin Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22, 6, pp.644-654. November 1976.

[17] RSA Laboratories. *What is public-key cryptography?*. Available at <http://www.rsa.com/rsalabs/node.asp?id=2165>. Accessed in 4 May 2009.

[18] Luther Martin. *Introduction to Identity-Based Encryption*. Artech House Publishers, 1st edition, 2008.

[19] Stanford Applied Crypto Group. *IBE Secure E-mail*. Last modified in March 8th 2002. Available at <http://crypto.stanford.edu/ibe/>. Accessed in May 15th 2008.