Studies on Public Key and Identity-Based Cryptographic Primitives

Thesis submitted to the Indian Statistical Institute in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy

by

Mahabir Prasad Jhanwar
C R Rao Advanced Institute of Mathematics, Statistics and Computer Science
University of Hyderabad Campus
Prof C R Rao Road, Gachibowli
Hyderabad 500046
INDIA
e-mail: mahavir.jhawar@gmail.com

under the supervision of

Prof Rana Barua
Theoretical Statistics and Mathematics Unit
Indian Statistical Institute
203, B.T. Road
Kolkata 700108
INDIA
e-mail: rana@isical.ac.in

Acknowledgments

First and foremost I offer my sincerest gratitude to my supervisor, Prof Rana Barua, who has supported me throughout my thesis with his patience and knowledge. I gratefully acknowledge him for his advice, supervision, and crucial contribution, which made him a backbone of this research and so to this thesis. I would like to express my deepest appreciation to Prof Barua for placing his faith on my research ideas and limited ability. I earnestly wish for his guidance in my future research endeavors.

I learned to be kind whenever possible, but after meeting Prof Bimal Roy, I realize it is "always" possible. He knows the art of encouraging people to find out for themselves how wonderful you are. I am much indebted to Prof Bimal Roy for all his help. To the role model for hard workers in the Crypto Research Group, Prof Palash Sarkar, I would like to express my gratitude to him for having greatly benefited from his teaching. I often see Prof Subhamoy Maitra as the best example of a "truly fast and efficient" algorithm. His sheer energy is exemplary. I will remain thankful for his advices. My utmost gratitude to my teachers at Stat-Math Unit, in particular to S C Bagchi, Mahuya Di, SMS, Amartya Da, Haimanti Di, Rudra Da and Gautam Da, for their kind concern and consideration during my early days as a research scholar. I cherished my extended discussions with Dr Kishan Chand Gupta as prized possessions. They were always full of encouragement and reassurance about my ability. There is no way but to be happy when you are with Kishan Da. Dr Sanjit Chatterjee's Thesis turned out to be a driving force for my own thesis writing. I gratefully acknowledge Professor S B Rao and Shri K Nageswara Rao for providing an excellent work atmosphere at C R Rao AIMSCS. I am much indebted to Venku Da for his valuable advice during many discussions, his patience in reading a part of this thesis and critical comments about it.

Collective and individual acknowledgments are also owed to my seniors and friends at CRG whose present remain helpful and memorable. Many thanks go in particular to Dalai Da, Avishek Da, Sushmita, Srimanta Da, Sumanta Da, Somitra Da, Prem Da, Ratna Di, Rishi. It is pleasure to mention Abhijit Pal, Prosenjit, Ashis Da, Abhijit Mandal, Debashis for being such a nice friends and always ready to lend a hand.

A man's growth is seen in the successive choirs of his friends. I am blessed to have found these true gems in form of Sumit, Kuldeep, Rajesh, Santanu, and Vikash. Each of them are too different to call them any way near to similar in their respective amazing talent. They are such a great band of good souls. Thanks friends, I owe you a great deal.

Words fail me to express my appreciation to Ina for all her support. I owe her for being there at the time of my crises.

This research work would not have been possible without the support of my family. Their love and persistent confidence in me has taken the load off my shoulder. My parents deserve special mention for their inseparable support and prayers. I was extraordinarily fortunate to have blessed with the care and support of Pawan Bhai, Bhabi, my sister Seema, Debu Da and the kids Khusboo, Neha and Abhijit.

I gratefully acknowledge the National Board of Higher Mathematics (NBHM) in India for my research fellowship. I would like to acknowledge the Cryptology Research Society of India (CRSI) for funding my travel to attend conferences within and outside India.

To My Parents

Contents

List of Symbols

$\mathbb{N}$:	set of natural numbers
$\mathbb{Z}$:	ring of integers
$\mathbb{R}$:	field of real numbers
$\mathbb{Z}/m\mathbb{Z}$:	ring of integers modulo m
RSA modulus N :	N is product of two large primes
$a \in_R A$:	random selection of an element a from the set A
$\triangleq$:	to define
$QR(m)$:	set of all quadratic residues modulo m
$NQR(m)$:	set of all quadratic-non residues modulo m
$QR(m)^+$:	set of all signed quadratic residues modulo m
$(\frac{a}{p})$, p is prime :	the Legendre symbol
$(\frac{a}{m})$, m is any odd integer :	the Jacobi symbol
$J(m)$:	$\{a \in \mathbb{Z}/m\mathbb{Z}^* \mid (\frac{a}{m}) = 1\}$

List of Tables

Chapter 1

Introduction

The field of cryptography went through several paradigm shifts and periods of consolidation between these paradigm shifts before it has been established as an important branch of science. Though known from the antiquity and not without some shining milestones; it encountered a very crucial paradigm shift in 1976 through the seminal work of Diffie and Hellman [43], when they introduced the notion of public key cryptography.

Public key cryptography is also known as asymmetric key cryptography. This alternate terminology rightly suggest that prior to the work of Diffie and Hellman, cryptography was thriving in the "symmetric" setting only, i.e., the same secret key was used for encryption as well as decryption. This kind of framework necessitates a secret channel to be established between the communicating parties to exchange the secret key prior to any communication over a public channal. This is, no doubt, a cumbersome business when a large numbers of users want to communicate secretly with each other.

Within two years of publication of Diffie-Hellman's work, the field of cryptography got a milestone in the form of RSA public key encryption [98]. In 1985 came the ElGamal public key encryption [48]. This was soon followed by Koblitz [70] and Miller [80], who independently proposed a public key encryption called the elliptic curve cryptosystem (ECC). RSA and ECC are the two most popular public key encryptions to date.

Despite these early (important and versatile) breakthroughs in the field of public key cryptography, the construction of a public key encryption that is both practical and provably secure in the security model of indistinguishability of en-

cryptions against adaptive chosen ciphertext attack (IND-CCA secure) [94] took a somewhat longer period. There were several attempts in the literature but none seemed provably secure against adaptive chosen ciphertext attack and practical at the same time. Finally, the solution to the above problem appeared in 1998 through the work of Cramer and Shoup [37]. Since then, the subject of building practical IND-CCA secure public key encryption schemes has flourished with many successful results. Even today, new solutions to the above problem is considered extremely important for its theoretical and practical values.

Identity-based cryptography is an extension of the public key paradigm, which was initially suggested by Adi Shamir [106] in 1984. Identity-based encryption (IBE) offers a nice solution to a large fraction of practical problems faced during the deployment of a public key infrastructure. More generally, IBE can simplify systems that manage a large number of public keys. The basic foundational remark of identity-based encryption is that even in the case of unencrypted communications, the user already needs to learn some basic information before he/she can communicate with another user. At the very least, he/she should obtain his/her telephone number, his e-mail address or a similar information. As pointed out by Shamir in his seminal paper [106], it would be extremely nice if this basic information could replace the need for an encryption key altogether.

The construction of an identity-based encryption remained unsolved till 2001. In an active field like that of cryptography, a problem that remains open for seventeen years must be a tough problem. Practical constructions of identity-based encryption first appeared at the turn of the 20th century. Two similar schemes based on bilinear pairings were independently proposed by Sakai, Ohgishi and Kashahara [92] and by Boneh and Franklin [21], followed soon afterwards by a completely different scheme due to Cocks [35]. Things have changed dramatically in the years since 2001. Many improvements have been made to the Boneh-Franklin IBE scheme, and a few brand new approaches to IBE have even been proposed. All of them, however, rely in one way or another upon the notion of bilinear pairings [7, 55]. Pairings are powerful mathematical constructs defined over certain algebraic curves, and whose recently discovered potential for creative cryptographic applications has not ceased to be a source of much amazement. In this regard, Cocks pairing-free approach to IBE remains for the most part an isolated result with its share of limitations. Cocks solution based on quadratic residuosity modulo an RSA number is not very efficient in terms of bandwidth,

i.e., the size of the ciphertext is very large. Since the work of Cocks', an important problem was to construct a space-efficient IBE (a system with short ciphertexts) that does not uses pairings. Recently, Boneh, Gentry and Hamburg [24] have given a non-pairing based IBE which is space-efficient. The trade-off is a substantial increase in encryption time. The difficult and important part of the subsequent research in this direction is to strike a better balance between encryption time and ciphertext size.

The most crucial role behind many of the breakthroughs mentioned in the preceding discussion is played by computational problems. Computational problems play the central role in the field of cryptography. By a computational assumption we mean that a certain computational problem is "hard" to solve. Cryptographic schemes often work with finite groups in such a way that the security of the scheme depends upon an explicit complexity-theoretic assumption about computational problems in that group. Study of various cryptographically important computational problems constitute an important branch within the field of cryptography. The concept of pseudo-free group is a recent development [97] in the bag of cryptographically important computational assumptions. The notion of "pseudo-free group" was first introduced by Hohenberger [66]. Rivest [97], explored this notion and provided an alternative stronger definition. In that paper Rivest define pseudo-free (abelian) groups as computational (abelian) groups such that no polynomial time adversary, given random group elements $a_1, \ldots, a_n$ (chosen using an appropriate sampling procedure), can output (with non negligible probability) an equation which is unsatisfiable over the free abelian group generated by the symbols $a_1, \ldots, a_n$, together with a solution to the equation in the computational group. Pseudo-free assumption implies a number of other well-known cryptographic assumptions. The study of pseudo-freeness by Rivest produced some intriguing open problems and conjectures. The main question that was left open by Rivest in [97] is: do pseudo-free groups exists? Moreover, he made the conjecture that the RSA group $\mathbb{Z}/N\mathbb{Z}^*$ (where $N = P \cdot Q$ is the product of two large primes) is pseudo-free and nicknamed his conjecture the *super strong RSA assumption*. Resolution of this conjecture is considered one of the challenging problems related to the pseudo-free groups and its applications.

The field of public key cryptography render its many primitives and concepts as building blocks for several other topics within the field of Cryptography. The field of publicly verifiable secret sharing (PVSS) is one such. In a secret sharing scheme,

there exists a dealer and n shareholders (sometimes referred to participants). The dealer splits a **secret**, say s, into n different pieces, called **shares**, and sends each share to each shareholder. An access structure describes which subsets of shareholders are qualified to recover the secret. The verifiable secret sharing (VSS) schemes constitute a particular interesting class of schemes as they allow each receiver of information about the secret (share of the secret) to verify that the share is consistent with the other shares. A publicly verifiable secret sharing scheme, proposed by Stadler in [111], is a VSS scheme in which anyone, not only the shareholders, can verify that the secret shares are correctly distributed. In most PVSS schemes, the verification procedure involves interactive proofs of knowledge. These proofs are made non-interactive by means of the Fiat-Shamir technique [50] and thus security for verifiability can only be carried out in the random oracle model [9]. Transforming security analysis of cryptographic primitives from the framework of random oracle model to the standard model has always turned out to be a theoretically important task which is seemingly difficult in most of the cases. Achieving simultaneously the following two features for PVSS is a challenging job: efficient non-interactive public verification and proving security for the public verifiability in the standard model.

1.1 Outline of the Thesis

In this thesis, we further explore the avenues that may lead to the solution of the problems that are mentioned in the Introduction. The thesis is based on the works mentioned in Section 1.2 and is organized as follows.

In Chapter 2, we define the core concepts that are used through-out the thesis. We give precise formal definitions in some of the cases while providing an informal discussion and results for some other and fix the notation.

In Chapter 3, we present our work about pseudo-free groups. Recently, in [77] Micciancio partially resolved the Rivest's conjecture "*the super strong RSA assumption*" by providing an affirmative answer. He proved $\mathbb{Z}/N\mathbb{Z}^*$ is pseudofree under the strong RSA assumption modulo the following constraints: N is product of two "safe primes" (i.e., $N = P \cdot Q$, where P and Q are of the form $2p + 1$ and $2q+1$ respectively such that p and q are primes) and the fact that the proof for the pseudo-freeness of $\mathbb{Z}/N\mathbb{Z}^*$ requires sampling procedure that chooses elements at

random from the subgroup $QR(N)$, the set of quadratic residues modulo N. The problem that was left open by Micciancio is that if one can prove pseudo-freeness of $\mathbb{Z}/N\mathbb{Z}^*$ if elements are sampled uniformly at random from the whole group $\mathbb{Z}/N\mathbb{Z}^*$. We prove $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free when elements are sampled uniformly at random from the subgroup of signed quadratic residues of $\mathbb{Z}/N\mathbb{Z}^*$ in this chapter. Consequently, we believe one can show $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free where elements are sampled from $QR(N) \cup QR(N)^+$, thus enlarging the set from which elements are sampled. The group $QR(N)^+$ has been suggested by Fischlin and Schnorr in [51] (in the different context of hard-core bits of generalized Rabin functions [51, 93]) and later Hoftheinz and Kiltz [65] have shown its cryptographic applications.

In Chapter 4, we describe a identity-based encryption scheme without pairings based on the quadratic residuosity assumption in the random oracle model. Our scheme is more time efficient than Boneh-Gentry-Hamburg's identity-based encryption BasicIBE, but is less space efficient. Compare to Cocks' IBE, our scheme is more space efficient. Time efficiency of our scheme is comparable to Cocks' IBE. We also describe a public key encryption scheme. In the standard model, the scheme is IND-CPA [57] secure based on the combined hardness of quadratic residuosity problem and RSA problem. Our scheme is more time efficient than the public key encryption scheme BasicPKE proposed by Boneh-Gentry-Hamburg in [24]. Space efficiency of BasicPKE and our scheme remain same. Both the schemes largely depends on certain results about the quadratic congruence $Rx^2 + Sy^2 \equiv 1 \pmod{N}$, where N is an RSA modulus and R, S are quadratic residues modulo N. We describe, using elementary methods, a useful characterization of solutions of $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ and a count of the number of solutions of $Rx^2 + Sy^2 \equiv 1 \pmod{N}$.

In Chapter 5, we present a (t, n)-threshold non-interactive publicly verifiable secret sharing (PVSS) scheme. Our proposal satisfies both the following properties: efficient non-interactive public verification and a proof of security for the public verifiability in the standard model. Efficiency of the non-interactive public verification step of the proposed scheme is optimal (in terms of computations of pairings while comparing with the earlier solution in [61]. In public verification step of [61], one needs to compute $2n$ many pairings, where n is the number of shareholders, whereas in our scheme the number of pairing computations is 4 only. This count is irrespective of the number of shareholders. We also provide a formal proof for the semantic security (IND) of our scheme based on the hardness

of a problem that we call the (n, t)-multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework [20]. We also observe that a simple modification to the verification algorithm of [61] reduces the number of pairing computations from $2n$ to $n + 1$. But this modification is done at the cost ([61] enjoys unconditional security for public verifiability) of reducing the security of public verifiability to a new computational problem.

In Chapter 6, we present an IND-CCA secure public key encryption scheme. The first truly practical public key encryption that is provably secure against chosen ciphertext attack was discovered by Cramer and Shoup [37]. The security of this scheme is based on the hardness of the decisional Diffie-Hellman problem. In [39] Cramer and Shoup show that their original scheme is an instance of a more generic paradigm. This paradigm is based on the hash proof systems. In [39] they also show that their paradigm can be also instantiated with the Quadradic Residuosity and Composite Residuosity assumptions [39]. In this chapter we have shown that the Cramer-Shoup paradigm can also be instantiated based on the decisional bilinear Diffie-Hellman problem. In particular, we proved that our scheme is IND-CCA secure by showing it to be a particular instance of the Cramer-Shoup framework.

1.2 List of Publications Related to this Thesis

List of Publications Related to this Thesis (In Order of its Appearance in this Thesis)

1. Mahabir Prasad Jhanwar and Rana Barua. Sampling from Signed Quadratic Residues: RSA Group Is Pseudofree. In Bimal K. Roy and Nicolas Sendrier, editors, Progress in Cryptology - INDOCRYPT 2009, 10th International Conference on Cryptology in India, New Delhi, India, December 13-16, 2009. Proceedings, volume 5922 of Lecture Notes in Computer Science, pages 233-247. Springer Verlag, 2009.

2. Rana Barua and Mahabir Prasad Jhanwar. On the Number of Solutions of the Equation $Rx^2 + Sy^2 = 1 \pmod N$. In *Sankhyā: The Indian Journal of Statistics*. Volume 72-A, Part 1, pages 226-236, 2010. Springer Verlag, 2010.

3. Mahabir Prasad Jhanwar and Rana Barua. A Variant of Boneh-Gentry-Hamburg's Pairing-Free Identity Based Encryption Scheme. In Moti Yung and Peng Liu and Dongdai Lin, editors, Information Security and Cryptology, 4th International Conference, Inscrypt 2008, Beijing, China, December 14-17, 2008. Proceedings, volume 5487 of Lecture Notes in Computer Science, pages 314-331. Springer Verlag, 2009.

4. Mahabir Prasad Jhanwar and Rana Barua. A Semantically Secure Public Key Encryption in the Standard Model. In Alexander Kholosha, Eirik Rosnes, and Matthew Parker, editors, The International Workshop on Coding and Cryptography WCC 2009, Ullensvang, Norway, May 10-15, 2009. Proceedings pages 181-190.

5. Mahabir Prasad Jhanwar. A Practical (Non-interactive) Publicly Verifiable Secret Sharing Scheme. *Communicated*. Available at http://eprint.iacr.org /2010/495.

6. Mahabir Prasad Jhanwar and Rana Barua. A Public Key Encryption in Standard Model using Cramer-Shoup Paradigm. International Cryptology Workshop and Conference 2008, Kuala Lumpur, Malaysia, June 2-12, 2008. Proceedings pages 298-308.

Chapter 2

Preliminaries

We establish here some terminology, notation, and simple facts that will be used throughout the thesis.

2.1 Basic Definitions: Complexity of Computation

The Big O Notation

Suppose that $f(n)$ and $g(n)$ are functions of the positive integers n which take positive real values for all n. Suppose that for all $n \geq n_0$ and for some constant C, f and g satisfy the inequality $f(n) \leq Cg(n)$. Then we say $f = O(g)$.

Size of Numbers

From now on, unless otherwise stated, we shall assume that all of our arithmetic on numbers is performed to the base 2. Throughout this thesis we shall use the notation log to mean $\log_2$. For an integer m, we define its size to be the number of bits in the binary representation of $|m|$; more precisely,

$$\text{size of } m = \begin{cases} \lfloor \log |m| \rfloor + 1 & \text{if } m \neq 0, \\ 1 & \text{if } m = 0. \end{cases}$$

If size of m is λ, we say that m is an λ-bit integer. Notice that if m is a positive, λ-bit integer, then $\log m < \lambda \leq \log m + 1$, or equivalently, $2^{\lambda-1} \leq m < 2^{\lambda}$.

Probabilistic Polynomial Time Algorithm

We now recall a definition that is fundamental in the study of algorithms.

Definition 2.1.1 *An algorithm to perform a computation is said to be a polynomial time algorithm if there exists an integer c such that the number of bit operations required to perform the algorithm on integers of size of at most k is $O(k^c)$. Further, a probabilistic polynomial time (PPT) algorithm is a polynomial time algorithm that includes a random selection of one or more integers during its execution.*

Negligible Functions

Definition 2.1.2 *We call a function $\mu : \mathbb{N} \to \mathbb{R}$ negligible if for every positive polynomial $p(\cdot)$ there exists an n_0 such that for all $n > n_0$,*

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

2.2 Basic Results: Number Theory

Congruences

Let a, b be integers. The integers a, b are called *congruent* modulo the integer m, and this is abbreviated $a \equiv b \mod m$, if $m|(a-b)$, otherwise a and b are called *incongruent* modulo m, and this is abbreviated by $a \not\equiv b \pmod m$. It is clear that for each fixed m, the relation $\equiv \pmod m$ is an *equivalence* relation on the set $\mathbb{Z}$ of all integers. For each integer a, let $[a]$ denote the equivalence class of a, i.e. the set of all integers b such that $b \equiv a \pmod m$. For each m there exists exactly m equivalence classes modulo m, namely $[0], [1], \ldots, [m-1]$. $\mathbb{Z}/m\mathbb{Z} \stackrel{\Delta}{=} \{[0], [1], \ldots, [m-1]\}$ is the set of all equivalence classes modulo m.

One can define two operations , addition (denoted by $+$) and multiplication (denoted by $\cdot$) on the set $\mathbb{Z}/m\mathbb{Z}$ as follows: $[a] + [b] = [a+b]$ and $[a] \cdot [b] = [a \cdot b]$. The set $\mathbb{Z}/m\mathbb{Z}$ endowed with these two operations forms a commutative ring with unity. The multiplicative group of unit (invertible) elements of this ring $\mathbb{Z}/m\mathbb{Z}$ is denoted as $\mathbb{Z}/m\mathbb{Z}^*$. In the next section we will describe $\mathbb{Z}/m\mathbb{Z}^*$ more explicitly. In order to avoid unnecessary complications, from now on and for the rest of this thesis the same symbol a will be used for an integer a and its equivalence class $[a]$

modulo an integer m. This should cause no confusion because it will always be clear from the context.

The linear congruence $ax \equiv b \pmod{m}$

We recall the definition of polynomial congruence modulo m. Quite generally, let $f(x_1, \ldots, x_n)$ be a polynomial in n variables with integer coefficients and consider the congruence $f(x_1, \ldots, x_n) \equiv 0 \pmod{m}$. A solution is an n-tuple of integers $(a_1, \ldots, a_n)$ such that $f(a_1, \ldots, a_n) \equiv 0 \pmod{m}$. If $(b_1, \ldots, b_n)$ is another tuple such that $b_i \equiv a_i$ for $i = 1, \ldots, n$, then it is easy to see that $f(b_1, \ldots, b_n) \equiv 0 \pmod{m}$. We do not want to consider these two solutions as being essentially different. Thus two solutions $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ are called equivalent if $a_i \equiv b_i$ for $i = 1, \ldots, n$. The number of solutions to $f(x_1, \ldots, x_n) \equiv 0 \pmod{m}$ is defined to be the number of inequivalent solutions.

The simplest congruence is $ax \equiv b \pmod{m}$. We state a criterion to test this congruence for solvability, and if it is solvable, give a formula for the number of solutions. Let $d > 0$ be the greatest common divisor of a and m, denoted as $gcd(a, m)$. Set $a' = a/d$ and $m' = m/d$. Then a' and m' are relatively prime.

Proposition 2.2.1 *The congruence* $ax \equiv b \pmod{m}$ *has solution iff* $d|b$. *If* $d|b$, *then there are exactly* d *solutions. If* x_0 *is a solution, then the other solutions are given by* $x_0 + m', x_0 + 2m', \ldots, x_0 + (d-1)m'$.

Corollary 2.2.1 *If* a *and* m *are relatively prime, then* $ax \equiv b \pmod{m}$ *has one and only one solution.*

Corollary 2.2.2 $\mathbb{Z}/m\mathbb{Z}^* = \{a \in \mathbb{Z}/m\mathbb{Z} \mid gcd(a, m) = 1\}$.

When the modulus m of a congruence is composite it is possible to reduce a congruence modulo m to system of simpler congruences. The main theorem of this type is the well known Chinese Remainder Theorem (CRT), which we state below.

Theorem 2.2.1 *Suppose that* $m = m_1 m_2 \cdots m_t$ *and that* $gcd(m_i, m_j) = 1$ *for* $i \neq j$. *Let* $b_1, b_2, \ldots, b_t$ *be integers and consider the system of congruences:*

$$x \equiv b_1 \pmod{m_1}, x \equiv b_2 \pmod{m_2}, \ldots, x \equiv b_t \pmod{m_t}$$

This system always has solutions and any two solutions differ by a multiple of m, *i.e., this system has a unique solution modulo* m.

Corollary 2.2.3 *Let $f(x) \in \mathbb{Z}[x]$ and $m = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$, where p_i's are distinct primes. Let N be the number of solutions to $f(x) \equiv 0 \pmod{m}$ and N_i be the number of solutions to $f(x) \equiv 0 \pmod{p_i^{e_i}}$. Then $N = N_1 N_2 \cdots N_t$.*

2.3 Quadratic Residues

Definition 2.3.1 *Suppose m is a positive integer and a is an integer. If $gcd(a, m) = 1$, a is called quadratic residue mod m if the congruence $x^2 \equiv a \pmod{m}$ has a solution. Otherwise a is called a quadratic non-residue mod m.*

Let $QR(m)$ (respectively $NQR(m)$) be the set of all quadratic residues (respectively non-residues) modulo m. One may note that $QR(m)$ is a subgroup of $\mathbb{Z}/m\mathbb{Z}^*$. This group has many applications in cryptography when m is considered either

- a prime number
- or $m = N = p \cdot q$, where p and q are two distinct odd primes. In this case N is called an RSA modulus [1].

For both forms of m, we shall discuss broadly the following two problems.

Problem-1	
Input:	$a \in_R \mathbb{Z}/m\mathbb{Z}^*$
Output:	to determine if $a \in QR(m)$ or not

Note: Later we shall see that the input to the Problem-1 can be chosen from a proper subgroup of $\mathbb{Z}/m\mathbb{Z}^*$.

Problem-2	
Input:	$a \in_R QR(m)$
Output:	$y \in \mathbb{Z}/m\mathbb{Z}$ such that $y^2 \equiv a \pmod{m}$

[1] Unless otherwise mentioned, by N we always mean an RSA modulus

Problem-1: When m is a prime number

In what follows p will denote an odd prime. We state the following result, known as Euler's criterion, that will give rise to a polynomial time deterministic algorithm to *check* if a given element is quadratic residue or not and thus efficiently solves the Problem-1.

Theorem 2.3.1 *(**Euler's Criterion**) Let $p \nmid a$. Then a is a quadratic residue modulo p iff*

$$a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$$

Thus one has to compute a modular exponentiation to check if an element is quadratic residue or not. One can use the well known SQUARE AND MULTIPLY algorithm [36, Chapter 1] for exponentiation modulo p. The complexity of the algorithm is $O((\log p)^3)$.

We now define *Legendre symbol*, which is an extremely convenient tool for discussing quadratic residues.

Definition 2.3.2 *Suppose p is an odd prime. For any integer a, define the Legendre symbol $\left(\frac{a}{p}\right)$ as follows:*

$$\left(\frac{a}{p}\right) = \left\{ \begin{array}{ll} 0 & \text{if } a \equiv 0 \pmod{p} \\ 1 & \text{if } a \text{ is a quadratic residue mod } p \\ -1 & \text{if } a \text{ is a quadratic non-residue mod } p \end{array} \right\}$$

The following theorem ensures that it is enough to compute Legendre symbol to check if an element is quadratic residue or not.

Theorem 2.3.2 *Let a be an integer and p be a prime, then we have*

$$a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \pmod{p}$$

Later in this section we will recall the definition of a more general terminology, that of Jacobi symbol of which the Legendre symbol is a particular case. We will also see that one can efficiently compute the Jacobi symbol (and hence Legendre symbol). We now list some of the properties of Legendre symbol.

Proposition 2.3.1

1. $a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \pmod{p}$

2. $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$

3. If $a \equiv b \pmod{p}$, then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$

Corollary 2.3.1 *There are as many quadratic residues as quadratic non-residues mod p.*

Corollary 2.3.2 *Quadratic residue × quadratic residue = quadratic residue, quadratic non-residue × quadratic non-residue = quadratic residue, and quadratic residue × quadratic non-residue = quadratic non-residue.*

Corollary 2.3.3 $(-1)^{\frac{p-1}{2}} = \left(\frac{-1}{p}\right)$

Corollary 2.3.3 is particularly interesting. Every odd integer has the form $4k+1$ or $4k+3$. Using this one can restate Corollary 2.3.3 as follows: -1 is a quadratic residue mod p iff p is of the form $4k+1$.

Problem-2: When m is a prime number

Let p be an odd prime number and we are given an element $a \in QR(p)$. Then, by definition, there exists an $x \in \mathbb{Z}/p\mathbb{Z}^*$ such that $x^2 \equiv a \pmod{p}$. How do we find such an x?

There are essentially three algorithms for solving the above problem. One is a special case of a general method for factoring polynomials modulo a prime p [36, Chapter 3]. Another is due to Schoof [103]. This algorithm is non-probabilistic and runs in polynomial time. It is quite complex since it involves the use of elliptic curves. There are several follow up works on this algorithm by Atkin. The third algorithm is due to Tonelli and Shanks, and although probabilistic, it is quite efficient. A detailed discussion on this algorithm can be found in Cohen's book [36, Chapter 1]. Thus computing square roots modulo a prime number is easy (polynomial time).

Problem-1: When $m = N$, an RSA modulus

Given $a \in \mathbb{Z}/N\mathbb{Z}^*$, the task at hand is to decide if a is a quadratic residue or not. It is clear that, if we know the factorization of N, then we can accomplish this task by determining if a is a quadratic residue modulo each prime divisor p and

q (CRT). However, without knowledge of this factorization (which is in general believed to be hard to compute), there is no efficient algorithm known.

We now discuss the distribution of quadratic residues in $\mathbb{Z}/N\mathbb{Z}^*$. To this we need some more number theoretic definitions and results.

The Jacobi symbol

Let m be an odd positive integer. Suppose $m = q_1 \cdots q_k$, where the q_i's are odd primes, not necessarily distinct. For an integer a the Jacobi symbol $\left(\frac{a}{m}\right)$ is defined as

$$\left(\frac{a}{m}\right) \triangleq \left(\frac{a}{q_1}\right) \cdots \left(\frac{a}{q_k}\right)$$

where $\left(\frac{a}{q_i}\right)$ is Legendre symbol. The Jacobi symbol extends the domain of definition of the Legendre symbol. Note that $\left(\frac{a}{m}\right) \in \{0, \pm 1\}$, and that $\left(\frac{a}{m}\right) = 0$ if and only if $gcd(a, m) > 1$. The following theorem summarizes the essential properties of the Jacobi symbol.

Theorem 2.3.3 *Let m, n be two odd positive integers, and let $a, b \in \mathbb{Z}$. Then we have:*

1. $\left(\frac{ab}{m}\right) = \left(\frac{a}{m}\right)\left(\frac{b}{m}\right)$
2. $\left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right)$
3. $a \equiv b \pmod{m}$ *implies* $\left(\frac{a}{m}\right) = \left(\frac{b}{m}\right)$
4. $\left(\frac{-1}{m}\right) = (-1)^{\frac{m-1}{2}}$
5. $\left(\frac{2}{m}\right) = (2)^{\frac{m^2-1}{8}}$
6. $\left(\frac{m}{n}\right) = (-1)^{\frac{m-1}{2}\frac{n-1}{2}}\left(\frac{n}{m}\right)$

This theorem is extremely useful from a computational point of view - with it, one can efficiently (running time $O(\log a \cdot \log m)$) compute $\left(\frac{a}{m}\right)$ without the knowledge of the prime factorization of either a or m.

For an RSA modulus N, $\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right)$. To decide quadratic residuosity character of a given element $a \in \mathbb{Z}/N\mathbb{Z}^*$, one may compute the Jacobi symbol $\left(\frac{a}{N}\right)$; if this is -1, we can conclude that a is not a quadratic residue as $\left(\frac{a}{N}\right) = -1$

implies one of the Legendre symbols $\left(\frac{a}{p}\right), \left(\frac{a}{q}\right)$ is -1, i.e., a is either quadratic non-residue modulo p or modulo q and hence by the Chinese Remainder Theorem one can check that a is a quadratic non-residue modulo N.

The situation is not so easy when $\left(\frac{a}{N}\right) = 1$ as this gives rise to two options; either $\left(\frac{a}{p}\right) = 1$ and $\left(\frac{a}{q}\right) = 1$ or $\left(\frac{a}{p}\right) = -1$ and $\left(\frac{a}{q}\right) = -1$, i.e., in the later case a can be a quadratic non-residue with Jacobi symbol being 1. What is the probability that given a random element $a \in \mathbb{Z}/N\mathbb{Z}^*$, the Jacobi symbol $\left(\frac{a}{N}\right) = 1$? Let us briefly discuss these issues. Note that, based on the discussion so far, the Problem-1 can be re-formulated by having the input as elements from $\mathbb{Z}/N\mathbb{Z}^*$ with Jacobi symbol 1. In fact we will see that these elements form a subgroup of $\mathbb{Z}/N\mathbb{Z}^*$.

One may view the Jacobi symbol as a group homomorphism. Though the following discussion is valid for any odd positive integer, we shall stick to the RSA modulus. Define the Jacobi map between the groups $\mathbb{Z}/N\mathbb{Z}^*$ and $\{\pm 1\}$

$$\begin{aligned} J_N : \mathbb{Z}/N\mathbb{Z}^* &\to \{\pm 1\} \\ a &\to \left(\frac{a}{N}\right) \end{aligned}$$

Theorem 2.3.3 (3) ensures that this definition is unambiguous and (1) ensures that this is indeed a group homomorphism. By $J(N)$ we denote the kernel of this map. Thus $J(N) = \mathrm{Ker}(J_N) = \{a \in \mathbb{Z}/N\mathbb{Z}^* | \left(\frac{a}{N}\right) = 1\}$ and hence is a subgroup of $\mathbb{Z}/N\mathbb{Z}^*$.

Theorem 2.3.4 *Let N be an RSA modulus. Then,*

1. *the index of $J(N)$ in $\mathbb{Z}/N\mathbb{Z}^*$, i.e., $[\mathbb{Z}/N\mathbb{Z}^* : J(N)] = 2$ and*
2. *$[J(N) : QR(N)] = 2$*

Thus for an element a chosen uniformly at random from $\mathbb{Z}/N\mathbb{Z}^*$ (respectively $J(N)$), the probability that it is in $J(N)$ (respectively $QR(N)$) is $\frac{1}{2}$. As of now there is no polynomial time algorithm is known which can determine the quadratic character of a randomly chosen element from $J(N)$ with probability $\frac{1}{2} + \epsilon$, where ϵ is a non-negligible function in the size of N. The assumption that there does not exists any such algorithm is known as **Quadratic Residuosity Assumption**. The problem can formally be defined as follows.

2.3.1 Quadratic Residuosity Assumption

Let RSAgen(λ) be a PPT algorithm that generates two equal size primes p and q, where λ is the security parameter, i.e., p and q are of size $\frac{\lambda}{2}$. The quadratic residuosity (QR) problem is defined as follows:

> Given a tuple (N, V), where $N = p \cdot q$ and $V \in J(N)$, decide whether $V \in QR(N)$ or $V \in J(N) \setminus QR(N)$

A probabilistic algorithm $\mathcal{A}$, which takes as input a tuple (N, V) runs in time $t(\lambda)$ (polynomial in λ) and outputs a bit, has advantage $\epsilon(\lambda)$ in solving the quadratic residuosity problem if

$$\text{QRAdv}_{\text{RSAgen},\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(N, V) = 1 | V \in QR(N)] - Pr[\mathcal{A}(N, V) = 1 | V \in J(N) \setminus QR(N)]| \geq \epsilon(\lambda)$$

where the probability is computed over all the random bits consumed by RSAgen and by $\mathcal{A}$.

The (t, ϵ)-quadratic residuosity assumption holds for RSAgen if no t-time algorithm has advantage at least ϵ in solving the quadratic residuosity problem for RSAgen. We say that the quadratic residuosity problem is (t, ϵ)-hard for RSAgen if the (t, ϵ)-quadratic residuosity assumption holds for RSAgen.

Problem-2: When $m = N$, an RSA modulus

If we know the prime factorization of N, then we can use the efficient algorithms for finding square roots modulo each of the primes p and q, and then use the Chinese Remainder Theorem to get a square root modulo N. However, if the factorization of N is not known, then there is no efficient algorithm known for computing square roots modulo N. In fact, one can show that the problem of finding square roots modulo N is at least as hard as the problem of factoring N [93], in the sense that if there is an efficient algorithm for computing square roots modulo N, then there is an efficient (probabilistic) algorithm for factoring N.

2.4 Signed Quadratic Residues

In this section we discuss the concept of signed quadratic residues. We use them in Chapter 3.

Let m be an odd integer. Elements of $\mathbb{Z}/m\mathbb{Z}$, the set of residue classes modulo m, are usually represented as $\{0, 1, \ldots, m-1\}$, where an element i represents the equivalence class containing the integer i. For the ongoing discussion, we take a different representation of the equivalence classes. They are represented as $\{-\frac{m-1}{2}, \ldots, -1, 0, 1, \ldots, \frac{m-1}{2}\}$. We call them signed integers. Thus for the equivalence class containing the integer $\frac{m+1}{2}$, we shall take $-\frac{m-1}{2}$ as the class representative. For $x \in \mathbb{Z}/m\mathbb{Z}$, we define $|x|$ as the absolute value of x

For a subgroup G of $\mathbb{Z}/m\mathbb{Z}^*$, consider the following set:

$$G^+ \triangleq \{|x| : x \in G\}$$

One may note that elements in G^+ are not necessarily in G. Define an operation '$\circ$' on G^+ as follows. For $g, h \in G^+$, $g \circ h = |g \cdot h \pmod m|$. One may check that $(G^+, \circ)$ becomes a group. For completeness we shall now work out the closure property. The rest of the group properties can be checked easily. For G^+ to be closed under '$\circ$', one has to show that $g, h \in G^+$ implies $g \circ h \in G^+$. For $g, h \in G^+$, the closure property can be shown as follows, where the proof goes by considering all the sub cases individually.

Case-1: $g, h \in G$. This trivially shows that $g \circ h$ belongs to G^+.

Case-2: Either of g or h is not in G. Without loss of generality say g is not in G and $h \in G$. As $g \in G^+$, clearly $-g \in G$. Thus $-g \cdot h \pmod m \in G$. Therefore $|-g \cdot h \pmod m| \in G^+$. But $|-g \cdot h \pmod m| = |g \cdot h \pmod m| = g \circ h$. Thus $g \circ h \in G^+$.

Case-3: Both g, h are not in G. Then $-g, -h \in G$. Thus $(-g) \cdot (-h) \pmod m \in G$. Therefore $|(-g) \cdot (-h) \pmod m| \in G^+$. But $|(-g) \cdot (-h) \pmod m| = |g \cdot h \pmod m| = g \circ h$. Thus $g \circ h \in G^+$.

For an integer x we define,

$$g^{\underline{x}} = g \circ \cdots \circ g = |g^x \pmod m|$$

More complicated expressions in the exponents are computed modulo the group order. For example,

$$g^{\underline{1/2}} = g^{\underline{2^{-1} \bmod \mathrm{ord}(G^+)}}$$

Define the following map $\psi : G \to G^+$ as follows: For $x \in G$, $\psi(x) = |x|$. One may check that, for $x, y \in G$

$$\psi(x \cdot y) = \psi(x) \circ \psi(y)$$

i.e., ψ is a group homomorphism. The kernel of this homomorphism is trivial if -1 is not in G and otherwise it is $\{-1, 1\}$ and in the former case, ψ becomes an isomorphism.

Signed quadratic residues (modulo m) is the group G^+ where G is taken to be the group of quadratic residues modulo m, i.e., $QR(m)$. Thus the group of signed quadratic residues modulo m is denoted as $QR(m)^+$.

Blum Integers

Informally, an integer m is a blum integer if $m = p^{k_1}q^{k_2}$, where p and q are different primes both $\equiv 3 \pmod 4$ and k_1 and k_2 are odd integers. These integers have some special properties [58, 74, 101], and were first used for cryptographic purposes by Blum in [17].

2.5 Bilinear Maps

Let G_1, G_2 and $\tilde{G}$ be three cyclic groups of prime order p. Suppose the group laws for all the three groups are written multiplicatively. A mapping $e : G_1 \times G_2 \to \tilde{G}$ is called an admissible bilinear map (pairing) if it satisfies the following properties:

- **Bilinearity:** $e(g_1^\alpha, g_2^\beta) = e(g_1, g_2)^{\alpha\beta}$ for all $g_1 \in G_1, g_2 \in G_2$ and $\alpha, \beta \in \mathbb{Z}/p\mathbb{Z}$.
- **Non-degeneracy:** $e(g_1, g_2) \neq 1$ unless $g_1 = 1$ or $g_2 = 1$.
- **Computability:** There exist efficient algorithms to compute the group operations in $G_1, G_2, \tilde{G}$ as well as the map $e(\cdot, \cdot)$.

A bilinear map group system is a tuple $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$ composed of the objects as described above. The above bilinear map is defined in *asymmetric* setting [23, 26]. Also in asymmetric setting, existence of an efficiently computable isomorphism $\phi : G_2 \to G_1$ is known [82, 83]. In symmetric setting, we have $G_1 = G_2$. Known examples of $e(\cdot, \cdot)$ usually have G to be a group of Elliptic Curve or Hyper Elliptic Curve points and $\tilde{G}$ to be a subgroup of a multiplicative group of finite field. Modified Weil pairing [21], Tate pairing [7, 55] are some of the practical examples of bilinear maps.

2.6 Decisional Bilinear Diffie-Hellman Assumption

Let $(p, G, \tilde{G}, e(\cdot,\cdot))$ be a bilinear map group system defined in the symmetric setting. Let λ be the size of p. The decisional bilinear Diffie-Hellman problem (DBDH) in $(p, G, \tilde{G}, e(\cdot,\cdot))$ is as follows:

> Given a tuple $(g, g^a, g^b, g^c, Z) \in G^4 \times \tilde{G}$, where g is a generator of G and a, b, c are random elements of $\mathbb{Z}/p\mathbb{Z}$, decide whether $Z = e(g,g)^{abc}$ (which we denote as Z is **real**) or Z is **random**.

A probabilistic algorithm $\mathcal{A}$, which takes as input a tuple $(g, g^a, g^b, g^c, Z) \in G^4 \times \tilde{G}$ runs in time $t(\lambda)$ (i.e., polynomial in λ) and outputs a bit, has an advantage $\epsilon(\lambda)$ in solving the DBDH problem if

$$\mathrm{Adv}_{\mathrm{DBDH},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(g, g^a, g^b, g^c, Z) = 1 | Z \text{ is real}] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, Z) = 1 | Z \text{ is random}]| \geq \epsilon(\lambda)$$

where the probability is computed over all the random choices of $a, b, c \in \mathbb{Z}/p\mathbb{Z}$ as well as the random bits consumed by $\mathcal{A}$.

The (t, ϵ)-DBDH assumption holds in $(G, \tilde{G}, e(\cdot,\cdot))$ if no t-time algorithm has advantage at least ϵ in solving the DBDH problem in $(G, \tilde{G}, e(\cdot,\cdot))$. We say that the DBDH problem is (t, ϵ)-hard in $(G, \tilde{G}, e(\cdot,\cdot))$ if the (t, ϵ)-DBDH assumption holds in $(G, \tilde{G}, e(\cdot,\cdot))$.

2.7 The Lagrange Interpolation

In this section we describe the Lagrange interpolation method. We use this in Chapter 5.

Let $P(x) = \sum_{j=0}^{t-1} \alpha_j x^j$ be a polynomial over a field F with degree $t-1$. Let $(x_1, P(x_1)), \ldots, (x_t, P(x_t))$ be t many distinct points over $P(x)$. Then, for given $(x_1, P(x_1)), \ldots, (x_t, P(x_t))$ one can reconstruct $P(x)$ as

$$P(x) = P(x_1)\lambda_{x_1}(x) + \cdots + P(x_t)\lambda_{x_t}(x)$$

where for $1 \leq j \leq t$

$$\lambda_{x_j}(x) = \frac{(x - x_1)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_t)}{(x_j - x_1)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_t)}$$

2.8 Statistical Distance

Here we discuss a very useful measure of *distance* between two random variables. Statistical distance play an important role in the field of cryptography, in particular when proving security of protocols using sequence of "games".

Let X and Y be two random variables which both take values in a finite set S. We define the *statistical distance* between X and Y as

$$\triangle[X;Y] \stackrel{\triangle}{=} \frac{1}{2}\sum_{s\in S}|P[X=s]-P[Y=s]|$$

The following are some immediate facts about statistical distance.

Theorem 2.8.1

For random variables X, Y, Z, we have

1. $0 \leq \triangle[X;Y] \leq 1$
2. $\triangle[X;X] = 0$
3. $\triangle[X;Y] = \triangle[Y;X]$, and
4. $\triangle[X;Z] \leq \triangle[X;Y] + \triangle[Y;Z]$

One may note that $\triangle[X;Y]$ depends only on the distribution of X and Y, and not on any other properties like the values X and Y takes. As such, it is reasonable to speak of the statistical distance between two distributions, rather than between two random variables.

2.9 Public Key Encryption

The idea of a public key cryptosystem (PKE) was proposed by Diffie and Hellman in their pioneering paper [43] in 1976. Their revolutionary idea was to enable secure message exchange between sender and receiver without ever having to meet in advance to agree on a common secret key. Recall that in symmetric key cryptography each communicating party needed to have a copy of the same secret key. This led to very difficult key management problem. In public key cryptography we replace the use of identical keys with two keys, one public and one private. The

public key can be published in a directory along with the user's name. Anyone who then wishes to send a message to the holder of the associated private key will take the public key, encrypt a message under it and send it to the the owner of the corresponding private key. The idea is that only the holder of the private key will be able to decrypt the message. Formally, a public key encryption scheme is given by a triplet of algorithms **KeyGen, Enc, Dec**. They are as follows.

- The *key generation algorithm* **KeyGen :** This is a probabilistic algorithm that takes a security parameter $\lambda \in \mathbb{N}$ (provided in unary e.g. 1^λ) returns a pair (pk, sk) of matching public and secret keys.
- The *encryption algorithm* **Enc :** This algorithm takes a public key pk and a message $m \in \{0,1\}^*$ to produce a ciphertext $C = \mathbf{Enc}(pk, m)$. This algorithm may be probabilistic. In the latter case, we write $\mathbf{Enc}(pk, m, r)$ where r is the random input to **Enc**.
- The *decryption algorithm* **Dec :** This is a deterministic algorithm which takes a secret key sk and a ciphertext C to produce either a message $m \in \{0,1\}^*$ or a special symbol $\perp$ to indicate that the ciphertext is invalid.

We require that for all (pk, sk) which can be output by $\mathbf{KeyGen}(1^\lambda)$, for all $m \in \{0,1\}^*$ and for all C that can be output by $\mathbf{Enc}(pk, m, r)$, we have that $\mathbf{Dec}(sk, \mathbf{Enc}(pk, m, r)) = m$.

Security

Public key encryption has several goals in terms of protecting the data that is encrypted. A convenient way to organize definitions of secure encryption is by considering separately the various possible *goals* and the various possible *attack models*, and then obtain each definition as a pairing of a particular goal and a particular attack model.

2.9.1 Security Goals

- **Indistinguishability of encryptions (IND):** This notion is due to Goldwasser and Micali [57]. Indistinguishability formalizes an adversary's inability to learn any information about the plaintext m underlying a challenge ciphertext C, capturing a strong notion of privacy.

- **Non-malleability (NM):** This notion is introduced by Dolev, Dwork and Naor [47]. Non-malleability roughly requires that an attacker given a challenge ciphertext C be unable to modify it into another, different ciphertext C' in such a way that the respective plaintexts m, m' underlying the two ciphertexts are "meaningfully related" to each other (for example $m' = m+1$).

2.9.2 Attack Models

Both the security goals, indistinguishability of encryptions and non-malleability, can be considered under the following attack models with increasing severity.

- **Chosen-plaintext attack (CPA):** Under this attack the adversary can obtain ciphertexts of plaintexts of her choice. In the public key setting, giving the adversary the public key suffices to capture these attacks.
- **Non-adaptive chosen-ciphertext attack (CCA1):** This attack model was formalized by Naor and Yung [84]. Under CCA1 the adversary gets, in addition to the public key, access to an oracle for the decryption function. The adversary may use this decryption function only for the period of time preceding her being given the challenge ciphertext C. (The term non-adaptive refers to the fact that queries to the decryption oracle cannot depend on the challenge ciphertext C. Colloquially this attack has also been called a "lunchtime" attack.)
- **Adaptive chosen-ciphertext attack (CCA2):** Under CCA2, due to Rackoff and Simon [94], the adversary again gets (in addition to the public key) access to an oracle for the decryption function, but this time she may use this decryption function even on ciphertexts chosen after obtaining the challenge ciphertext C, the only restriction being that the adversary may not ask for the decryption of C itself. The attack is called adaptive because queries to the decryption oracle can depend on the challenge ciphertext C.

As a mnemonic for the abbreviations CCA1, CCA2, just remember that the bigger number goes with the stronger attack. One can "mix-and-match" the goals {IND, NM} and attacks {CPA, CCA1, CCA2} in any combination, giving rise to six notions of security:

IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2.

Most of these notions are familiar (although under different names). IND-CPA is the notion defined in [57]; IND-CCA1 is the notion defined in [84]; IND-CCA2 is the notion defined in [94]; NM-CPA, NM-CCA1 and NM-CCA2 are from [44, 45, 46].

Relations between these security notions for public key encryption scheme have been deeply studied. One may refer to the papers of Bellare et al. [8] and of Bellare and Sahai [11] for a more vivid description.

In this thesis, we shall mainly be interested in IND-CPA and IND-CCA2 security notion. For notational convenience, hence onwards **IND-CCA2** security will be referred as **IND-CCA**.They can be formally defined as follows.

Definition 2.9.1 *Indistinguishability of encryptions (IND) against chosen-plaintext attack (CPA): IND-CPA*

This security notion is defined via the following game. This game is played between a challenger and a PPT adversary $\mathcal{A}$.

- Given the security parameter λ, the challenger generates a public key/secret key pair (pk, sk) by running the KeyGen algorithm.
- The adversary $\mathcal{A}$ is given the public key pk.
- The adversary $\mathcal{A}$ makes one *challenge query*: it chooses a pair of equal length messages m_0, m_1 and sends these to challenger.
- The challenger chooses $b \in_R \{0, 1\}$ and computes a challenge ciphertext $C^* = \text{Enc}(pk, m_b)$, which is given to $\mathcal{A}$.
- The game ends with $\mathcal{A}$ outputting a bit b'.

We define the IND-CPA advantage of $\mathcal{A}$ as a function of the security parameter as follows:

$$\text{Adv}^{\text{IND-CPA}}_{\text{PKE},\mathcal{A}}(\lambda) \stackrel{\triangle}{=} |\Pr(b' = b) - \frac{1}{2}| \tag{2.1}$$

Definition 2.9.2 *We say that a PKE is IND-CPA secure if for all PPT adversaries $\mathcal{A}$, we have that $Adv^{IND\text{-}CPA}_{PKE,\mathcal{A}}(\lambda)$ is negligible.*

Definition 2.9.3 *Indistinguishability of encryptions (IND) against adaptive chosen-ciphertext (CCA) attack: IND-CCA*

Like IND-CPA security notion, this notion also can be defined via the following game between a challenger and a PPT adversary $\mathcal{A}$.

- Given the security parameter λ, the challenger generates a public key/secret key pair: (pk, sk) by running the KeyGen algorithm.
- The adversary $\mathcal{A}$ is given the public key pk.
- $\mathcal{A}$ makes a number of *decryption queries* to the challenger; each such query is a ciphertext C of his choice; the challenger decrypts these arbitrary ciphertexts and returns the results to $\mathcal{A}$.
- The adversary $\mathcal{A}$ makes one *challenge query*: It chooses a pair of equal length messages m_0, m_1 and sends these to the challenger.
- The challenger chooses $b \in_R \{0, 1\}$ and computes a challenge ciphertext $C^* = \text{Enc}(pk, m_b)$, which is given to $\mathcal{A}$.
- $\mathcal{A}$ makes more *decryption queries*, just as before the *challenge phase*, but with the obvious restriction that $C \neq C^*$.
- The game ends with $\mathcal{A}$ outputting a bit b'.

We define the IND-CCA advantage of $\mathcal{A}$ as a function of the security parameter as follows:

$$\text{Adv}^{\text{IND-CCA}}_{\text{PKE},\mathcal{A}}(\lambda) \stackrel{\triangle}{=} |\Pr(b' = b) - \frac{1}{2}| \tag{2.2}$$

Definition 2.9.4 *We say that a PKE is IND-CCA secure if for all PPT adversaries $\mathcal{A}$, we have that $Adv^{IND\text{-}CCA}_{PKE,\mathcal{A}}(\lambda)$ is negligible.*

2.10 Identity-based Encryption

Identity-based cryptography is an extension of the public key paradigm, which was initially suggested by Adi Shamir [106] at CRYPTO' 84.

In order to better understand identity-based cryptography, it is important to know how traditional public key systems are usually put to use in real life applications. The Chapter 1 of [68] provides in depth motivation towards the need of having identity-based cryptography in place.

An identity-based encryption scheme $\mathcal{E}$ is specified by four randomized algorithms: **Setup, Extract, Encrypt, Decrypt:**.

- **Setup:** takes a security parameter λ and returns the *system (public) parameters* PP and the *master key* msk. The system parameters include a description of a message space $\mathcal{M}$ and a description of a ciphertext space $\mathcal{C}$. Intuitively, the system parameters are publicly known, while the *master key* is known only to the "Private Key Generator" (PKG).

- **Extract:** takes as input PP, msk, and an arbitrary $id \in \{0,1\}^*$, and returns a private key d (usually denoted as d_{id}). Here "id" is an arbitrary string that will be used as a public key, and "d_{id}" is the corresponding decryption key. The **Extract** algorithm extracts a private key from the given public key.

- **Encrypt:** takes as input PP, id, and $m \in \mathcal{M}$. It returns a ciphertext $C \in \mathcal{C}$.

- **Decrypt:** takes as input PP, $C \in \mathcal{C}$, and a private key d. It returns $m \in \mathcal{M} \cup \{\perp\}$. Usually the symbol "$\perp$" refers to the case where the input ciphertext is not valid.

These set of algorithms must satisfy the standard consistency requirement, namely for PP and msk output by **Setup**, let d_{id} be a private key generated by the algorithm **Extract** given a public key id as input, then

$$\forall m \in \mathcal{M} : \textbf{Decrypt}(\textbf{Encrypt}(m, id, PP), d_{id}, PP) = m.$$

2.10.1 Attack Models

As we discussed earlier, indistinguishability against chosen plaintext attack (IND-CPA) and chosen ciphertext attack (IND-CCA) are considered to be the correct notion of security for general-purpose public key encryption schemes. Boneh and Franklin [21] extended this notion of security to identity-based setting terming it as IND-ID-CPA and IND-ID-CCA respectively. As defined for public key encryption, we define semantic security (IND-ID-CPA) for identity-based encryption schemes using the following game. This game is played between a challenger and a PPT adversary $\mathcal{A}$.

IND-ID-CPA Game:

- The challenger takes a security parameter λ and runs the **Setup** algorithm. It gives the adversary $\mathcal{A}$ the resulting *system parameters* PP. It keeps the *master key* msk to itself.
- The adversary issues private key extraction queries for the identities $id_1, \ldots, id_r$. The challenger responds by running algorithm **Extract** to generate the private keys d_{id_j}'s corresponding to the public keys id_j's respectively. It sends d_{id_j}'s to the adversary. These queries may be asked adaptively.
- The adversary makes one challenge query: It outputs two equal length messages $m_0, m_1 \in \mathcal{M}$ and a public key id^* on which it wishes to be challenged. The only constraint is that id^* should not appear earlier in the private key extractions queries made by $\mathcal{A}$.
- The challenger picks a random bit $b \in \{0, 1\}$ and sets $C^* = \textbf{Encrypt}(PP, id^*, m_b)$. It sends C^* as the challenge to the adversary.
- The adversary issues more extraction queries $id_{r+1}, \ldots, id_s$. The only constraint is that $id_j \neq id^*$. The challenger responds with $d_{id_{r+1}}, \ldots, d_{id_s}$.
- The game ends with $\mathcal{A}$ outputting a bit $b' \in \{0, 1\}$.

As defined for public key encryption schemes, the IND-ID-CPA advantage of $\mathcal{A}$ against an IBE scheme is the following function of the security parameter λ,

$$\mathrm{Adv}^{\text{IND-ID-CPA}}_{\text{IBE},\mathcal{A}}(\lambda) \triangleq |\Pr(b' = b) - \frac{1}{2}|$$

Definition 2.10.1 *We say that an IBE scheme is IND-ID-CPA secure if for all PPT adversaries $\mathcal{A}$, we have that $Adv^{IND\text{-}ID\text{-}CPA}_{IBE,\mathcal{A}}(\lambda)$ is a negligible function of λ.*

The stronger security notion, IND-ID-CCA, can be similarly defined as follows.

IND-ID-CCA Game

- The challenger takes a security parameter λ and runs the **Setup** algorithm. It gives the adversary $\mathcal{A}$ the resulting *system parameters* PP. It keeps the *master key* msk to itself.

- The adversary issues queries $q_1, \ldots, q_r$ where q_j is one of:
  - Extraction query $\langle id_j \rangle$. The challenger responds by running algorithm **Extract** to generate the private key d_{id_j} corresponding to the public key id_j. It sends d_{id_j} to the adversary.
  - Decryption query $\langle id_j, C_j \rangle$. The challenger responds by running algorithm **Extract** to generate the private-key d_{id_j} corresponding to the public key id_j. it then runs algorithm **Decrypt** to decrypt the ciphertext C_j using the private key d_{id_j}. It sends the resulting plaintext to the adversary.

  These queries may be asked adaptively.

- The adversary makes one challenge query: It outputs two equal length messages $m_0, m_1 \in \mathcal{M}$ and a public key id^* on which it wishes to be challenged. The only constraint is that id^* did not appear earlier in the private key extractions queries made by $\mathcal{A}$.

- The challenger picks a random bit $b \in \{0, 1\}$ and sets $C^* = \textbf{Encrypt}(PP, id^*, m_b)$. It sends C^* as the challenge to the adversary.

- The adversary issues more queries $q_{r+1}, \ldots, q_s$ where q_j is one of:
  - Extraction query $\langle id_j \rangle$ where $id_j \neq id^*$.
  - Decryption query $\langle id_j, C_j \rangle \neq \langle id^*, C^* \rangle$.

  The Challenger responds to the queries $q_1, \ldots, q_r$. These queries may be asked adaptively.

- The game ends with $\mathcal{A}$ outputting a bit $b' \in \{0, 1\}$.

The IND-ID-CCA advantage of $\mathcal{A}$ against an IBE scheme is the following function of the security parameter λ:

$$\text{Adv}^{\text{IND-ID-CCA}}_{\text{IBE},\mathcal{A}}(\lambda) \triangleq |\Pr(b' = b) - \frac{1}{2}|$$

Definition 2.10.2 *We say that an IBE scheme is IND-ID-CCA secure if for all PPT adversaries $\mathcal{A}$, we have that $Adv^{IND\text{-}ID\text{-}CCA}_{IBE,\mathcal{A}}(\lambda)$ is a negligible function of λ.*

2.11 Publicly Verifiable Secret Sharing

Secret Sharing is one of the most important tools in modern cryptography. The concept and the first realization of secret sharing were presented independently in [105] and in [15]. Since then much work has been put into the investigation of such schemes (see [109, 112] for a list of references). In a secret sharing scheme, there exists a dealer and n shareholders (sometimes referred to participants). The dealer splits a **secret**, say s, into n different pieces, called **shares**, and sends one share to each shareholder. An access structure describes which subsets of shareholders are qualified to recover the secret. By a (t, n)-threshold access structure, $1 \leq t \leq n$, we mean that any subset of t or more shareholders will be able to recover the secret; any smaller subset of shareholders will not be able to gain any information about the secret.

In this section we describe a model for non-interactive publicly verifiable secret sharing (PVSS) scheme. In a PVSS scheme, a dealer D wishes to distribute shares of a secret value "s" among n shareholders $P_1, \ldots, P_n$. In this thesis, we consider (t, n)-threshold access structure, $1 \leq t \leq n$, which means that any subset of t or more shareholders will be able to recover the secret; any smaller subset will not be able to gain any information about the secret, unless a computational assumption is broken. A PVSS scheme is described by the following standard algorithms.

- **Initialization** This algorithm generates all system parameters. Furthermore, each shareholder P_i registers its public-key (may be issued by the dealer with the corresponding secret key). The actual set of shareholders taking part in a run of PVSS scheme must be a subset of the registered shareholders. We assume w.l.o.g. that shareholders $P_1, \ldots, P_n$ are the actual shareholders in the run described below.

- **Distribution** The distribution of the shares of a secret "s" is performed by the dealer D. The dealer computes and publishes the secret commitment value(s) and the share deriving value(s) respectively. The secret commitment value(s) commits the dealer to the value of secret s, whereas the share deriving value(s) can be used with the shareholders' secret keys to yield the share of the secret for the respective shareholders.

- **Verification** It is required that the dealer's commitment to the secret can be verified *publicly*. Thus any party knowing only the publicly available

information may verify that share deriving information is consistent with the share commitment information, i.e., it guarantees that the reconstruction protocol will be able to recover the same secret s. Furthermore, this verification runs non-interactively.

- **Reconstruction** The shareholders construct their shares S_i from the share deriving value using the secret keys. It is not required that all shareholders succeed in doing so, as long as a qualified set of shareholders is successful. These shareholders then release S_i and also the share commitment value(s) to verify that the released shares are correct. The share commitment information is used to exclude the shareholders which are dishonest or fail to reproduce their share S_i correctly. Reconstruction of the secret s can be done from the shares of any qualified set of shareholders.

In non-interactive PVSS schemes it is essential that all commitments can be verified non-interactively. Since any party can verify the output of the dealer, so we don't budget operations for the individual participants to check their own shares. Hence it suffices to have just one public verifier.

2.11.1 Security Model

Such a scheme must satisfy the following properties.

- **Correctness** If the dealer and the shareholders act honestly, every qualified subset of shareholders reconstructs the secret during the reconstruction algorithm.

- **Verifiability** If a dealer passes the verification step, then it implies that the secret commitment values are consistent with the share deriving values, i.e., the information which the dealer outputs for shareholders to derive their respective shares of the secret for which the dealer had published his commitment in terms of secret commitment values.

- **Privacy** The very basic requirement is that, for an honest dealer, the adversary cannot learn any information about the secret at the end of the protocol.

Privacy Following [61, 99], we can more formally define the above privacy notion, under the classical semantic-security notion [57], using a game between an adversary $\mathcal{A}$ and a challenger. The adversary here is a static one, i.e., at the beginning of the game, he is given the secret keys of the corrupted shareholders.
Indistinguishability of Secrets (IND) The security notion is defined via the following game between a challenger and a probabilistic polynomial time (PPT) adversary $\mathcal{A}$. Both the adversary and the challenger are given as input a security parameter λ.

- **Initialization:** The challenger runs Initialization(λ) to obtain the set of public parameters along with the public keys and the secret keys of all the shareholders. Besides all the public keys, the adversary is also given the secret keys of $t-1$ corrupted shareholders.
- **Challenge:** The challenger picks two random secrets T_0 and T_1 and a random bit $b \in \{0,1\}$. Then he runs the distribution algorithm for the secret T_b and sends all the resulting information to $\mathcal{A}$ along with $\{T_0, T_1\}$.
- **Guess:** Finally, the adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for b and wins the game if $b' = b$.

We define the advantage of this static adversary (SA) $\mathcal{A}$ against a (t,n)-threshold PVSS as follows:

$$\mathrm{Adv}^{SA-IND}_{PVSS,\mathcal{A}}(\lambda) = \left|\mathrm{Prob}[b' = b] - \tfrac{1}{2}\right|$$

The advantage is a function of the security parameter λ.

Definition 2.11.1 *We say that a (t,n)-threshold PVSS scheme is SA-IND secure if for all PPT adversaries $\mathcal{A}$, we have that $Adv^{SA-IND}_{PVSS,\mathcal{A}}(\lambda)$ is a negligible function in λ.*

Chapter 3

Pseudo-Free Groups

3.1 Introduction

Given a computational problem (Computational Assumption: the problem is "hard" to solve) over a finite group, often a cryptographic scheme is designed over this group in such a way that security (of the scheme) could be achieved without **extra assumptions**. The only way to formally prove such a fact is by showing that an attacker against the scheme can be used as a sub-part in an algorithm that can break the underlying computational problem.

For example, the RSA public-key cryptosystem [98] is based on the multiplicative group $\mathbb{Z}/N\mathbb{Z}^*$, where N is the product of two large primes. The security of RSA scheme depends upon the "RSA assumption" [98]. Informally this assumption is that it is hard to solve the equation $x^e \equiv a \pmod{N}$ given only $N, a \in \mathbb{Z}/N\mathbb{Z}^*$ and e ($gcd(e, \phi(N) = 1)$, where ϕ is Euler phi function). Similarly, the Cramer-Shoup cryptosystem and signature scheme [37, 38] depend upon the "Strong RSA Assumption" [6, 52], which similarly assumes the hardness of solving the equation $x^e \equiv a \pmod{N}$, though here the adversary is allowed to solve the equation for exponent $e > 1$ of **her choice**.

Within $\mathbb{Z}/N\mathbb{Z}^*$, Rivest [97] took this progression one step further. He examine the situation where the adversary may choose whatever equation (as long as the equation is "nontrivial" - unsatisfiable in the "corresponding" free group, with appropriate care for some details) and try to solve. The assumption $\mathbb{Z}/N\mathbb{Z}^*$ **is pseudo-free** ensures that the adversary can succeed with at most negligible probability. The notion of pseudo-free groups was introduced by Hohenberger [66].

Rivest [97], explored this notion and provided an alternative stronger definition. He defined pseudo-freeness of a family of *computational groups*. The assumption of pseudo-freeness may be made for arbitrary finite group, such as an elliptic curve group or even a noncommutative group.

Rivest [97], studied the assumption that a group is pseudo-free or, more specifically, pseudo-free abelian, and showed how it implies some known standard assumptions on the group. Thus assuming that a finite group is pseudo-free appears to be quite a strong assumption and formulating and studying such a strong assumption may indicate a course taken against the traditional style of making only the minimal complexity theoretic assumptions necessary for a cryptographic scheme. Rivest [97], provides motivation and justifications for studying pseudo-free groups and some of them are as follows:

- $\mathbb{Z}/N\mathbb{Z}^*$ is possibly a natural candidate for a pseudo-free group.
- It may turn out that the pseudo-freeness is in fact not a "stronger" assumption. It may be implied by some standard assumptions.
- Using a stronger assumption may make proofs (security reductions) easier.
- Reasoning in a free group can be quite simple and intuitive, so assuming pseudo-freeness allows one to capture "natural" security proofs in a plausible framework. This was Hohenberger's [66] motivation. In [66], pseudo-freeness has been linked to the construction of specific cryptographic primitives, like directed transitive signature schemes, for which no solution is currently known. See [85] for a recent work in this area.

Free groups are widely used in computer science, and most modern cryptography relies on the hardness of computational problems over finite groups. As argued in [97], pseudo-free groups may turnout to be a very interesting notion from cryptographic perspective. As pointed out by Micciancio [77], (non abelian) free groups were used in the so called Dolev-Yao model [47] for the symbolic analysis of public key cryptographic protocols. In the last few years, there have been several effort to bridge the gap between the symbolic model of [47] and the standard computational model used in cryptography (see for example [1, 5, 67, 78, 79, 81]). An intersting question is whether pseudo-free groups can be used to extend (in a computationally sound way) the Dolev-Yao model (in which encryption and

decryption are viewed as black-box operations with no algebraic properties) with richer data structures and cryptographic functions (e.g., homomorphic encryption schemes) that make fundamental use of computational groups.

The main question that was left open by Rivest in [97] is: do pseudo-free groups exists? Moreover, Rivest [97], made the conjecture that the RSA group $\mathbb{Z}/N\mathbb{Z}^*$ (where $N = P \cdot Q$ is the product of two large primes) is pseudo-free and nicknamed their conjecture the *super strong RSA assumption*.

In [77], Micciancio partially resolved this conjecture by providing an affirmative answer. He proved $\mathbb{Z}/N\mathbb{Z}^*$ is pseudofree under the strong RSA assumption modulo the following constraints: N is product of two "safe primes" (i.e., $N = P \cdot Q$, where P and Q are of the form $2p + 1$ and $2q + 1$ respectively such that p and q are primes) and the fact that the proof for the pseudo-freeness of $\mathbb{Z}/N\mathbb{Z}^*$ requires sampling procedure that chooses elements at random from the subgroup $QR(N)$, the set of quadratic residues modulo N.

The problem that was left open by Micciancio is that if one can prove pseudo-freeness of $\mathbb{Z}/N\mathbb{Z}^*$ if elements are sampled uniformly at random from the whole group $\mathbb{Z}/N\mathbb{Z}^*$.

Our Contribution

In this chapter we prove that $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free when elements are sampled uniformly at random from the subgroup of signed quadratic residues of $\mathbb{Z}/N\mathbb{Z}^*$. Consequently, we believe one can show $\mathbb{Z}/N\mathbb{Z}^*$ pseudo-free where elements are sampled from $QR(N) \cup QR(N)^+$, thus enlarging the set from which elements are sampled. The group $QR(N)^+$ has been suggested by Fischlin and Schnorr in [51] (in the context of hard-core bits of generalized Rabin functions [93],[51]) and later Hoftheinz and Kiltz [65] have shown its cryptographic applications.

3.2 Computational Group

In cryptography, for a mathematical group G, often a "suitable" representation $\langle\cdot\rangle : G \to \{0,1\}^*$ of G is what one looks for. Such a representation $\langle\cdot\rangle : G \to \{0,1\}^*$ is called a computational group implementing the underlying mathematical group. Clearly many computational groups may implement the same mathematical group. Below we define, a family of computational groups implementing a

family of finite mathematical groups.

Let $\mathcal{G} = \{G_N\}_{N \in \mathcal{N}}$ be a family of finite mathematical groups indexed by $N \in \mathcal{N} \subset \{0,1\}^*$. A computational group family implementing $\mathcal{G}$ is defined by a collection of representations $\langle \cdot \rangle_N : G_N \to \{0,1\}^*$ (for $N \in \mathcal{N}$) such that the following operations can be carried out in polynomial time in the bit-size of N.

- Composition: given $N \in \mathcal{N}$, representations $\langle x \rangle_N$ and $\langle y \rangle_N$ of group elements $x, y \in G_N$, compute representation $\langle x \circ y \rangle_N$ of $x \circ y$.
- Identity: given N, compute a representation $\langle 1 \rangle_N$ of the identity element 1 of the group G_N.
- Inverses: given N and $\langle x \rangle_N$ (for some $x \in G_N$), compute $\langle x^{-1} \rangle_N$.
- Recognizing elements from a group: given $N \in \mathcal{N}$ and $x \in \{0,1\}^*$, determine if there exists a $y \in G_N$ such that $x = \langle y \rangle_N$.
- Sampling group elements: on input $N \in \mathcal{N}$, output the representation $\langle x \rangle_N$ of a randomly chosen group element $x \in G_N$ (with not necessarily uniform probability distribution).

3.3 Free Group

A free group G is a group with a generating set $A \subseteq G$ such that there are no relations satisfied by any of the elements in A (A is "free" of relations). We denote G by $\mathcal{F}(A)$, the free group generated by the set A.

For example, for any finite group G, every element $g \in G$ satisfies the following relation: $g^{|G|} = 1$, where $|G|$ is the order of G. So for a finite group G, there is no generating set A of G such that the elements in A are free of relations. Thus finite groups cannot be "free".

Let us consider the additive group of integers $(\mathbb{Z}, +)$. It is an infinite group and 1 is a generator of $\mathbb{Z}$. One can see that 1 satisfies no relation in $\mathbb{Z}$. So $(\mathbb{Z}, +)$ is a example of a free group.

So free groups are necessarily infinite. The converse is not true. Consider the following group. Let

$$G = GL_2(\mathbb{R}), \quad a = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 & 2 \\ \frac{1}{2} & 0 \end{pmatrix}$$

Consider the subgroup $\langle a, b\rangle$ generated by $\{a, b\}$ of G. This subgroup is certainly not "free" as a, b both satisfies the following relations: $a^2 = b^2 = 1$. But $ab = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{pmatrix}$. It is easy to see that ab has infinite order. Thus $\langle a, b\rangle$ is an infinite subgroup of $GL_2(\mathbb{R})$ but it is not a free group.

It is natural at the outset (even before we know A is contained in some group) to formally define $\mathcal{F}(A)$ as follows. We consider A to be a non-empty finite set of distinct symbols.

Let $A = \{a_1, a_2, \ldots, a_\ell\}$. For each symbol a_i, let a_i^{-1} be a new formal symbol. Let A^{-1} denote the set $\{a_i^{-1} | a_i \in A\}$. Let $(a_i^{-1})^{-1} = a_i$ for all $a_i \in A$. Take a singleton set not contained in $A \cup A^{-1}$ and call it $\{1\}$. Let $1^{-1} = 1$.

A word on A is by definition a finite sequence $s_1 s_2 \ldots s_r$, where $s_i \in A \cup A^{-1}$. For example $a_1 a_3^{-1} a_2 a_2^{-1} a_3 a_2$ is a word. A word may be simplified, or reduced, by repeatedly eliminating any two adjacent inverse symbols. The resulting word is equivalent to the original. Thus, the word above is equivalent to $a_1 a_2$. A word that cannot be reduced further is *reduced* or in *canonical form*. Let

$$\mathcal{F}(A) = \{\text{all reduced words on } A\} \cup \{1\}$$

Define an operation $\circ$ on $\mathcal{F}(A)$ as follows. The operation $\circ$ is concatenation followed by simplification. For example, $a_1 a_3 a_2 \circ a_2^{-1} a_1 = a_1 a_3 a_2 a_2^{-1} a_1 = a_1 a_3 a_1$. Let $a_i \circ 1 = 1 \circ a_i = a_i$ for all $a_i \in A$. Thus, "1" will work as identity in $\mathcal{F}(A)$. The inverse of a word is just the reverse of the word, with each symbol replaced by its inverse. The operator $\circ$ is closed and associative [73]. Thus, under the operation $\circ$, $\mathcal{F}(A)$ becomes a group and infact it is a free group.

3.3.1 Free Abelian Group

A free abelian group $\mathcal{FA}(a_1, a_2, \ldots, a_\ell)$ is defined similarly to ordinary free groups, except that the group is abelian. Thus, for any pair of symbols a_i and a_j, we replace the sequence $a_i a_j$ by the sequence $a_j a_i$ and preserve equivalence. Commutativity enables one to define the canonical form for a word in $\mathcal{FA}(a_1, a_2, \ldots, a_\ell)$ to be a word of the form

$$a_1^{\epsilon_1} a_2^{\epsilon_2} \ldots a_\ell^{\epsilon_\ell}$$

where $\epsilon_i \in \mathbb{Z}$. One may see that $a_1^{\epsilon_1} a_2^{\epsilon_2} \ldots a_\ell^{\epsilon_\ell} \rightarrow (\epsilon_1, \epsilon_2, \ldots, \epsilon_\ell)$ gives a natural group isomorphism between $\mathcal{FA}(a_1, a_2, \ldots, a_\ell)$ and the ℓ-fold direct sum $\mathbb{Z} \oplus \mathbb{Z} \oplus$

$\ldots \oplus \mathbb{Z}$. ℓ is called the rank of the free group $\mathcal{FA}(a_1, a_2, \ldots, a_\ell)$. One may note that $\mathcal{FA}(a_1, a_2, \ldots, a_\ell)$ being free means,

$$a_1^{\epsilon_1} \cdots a_\ell^{\epsilon_\ell} = 1 \text{ implies } \epsilon_i = 0 \text{ for all } i.$$

3.3.2 Equations Over Free Groups

Consider a free group $\mathcal{F}(a_1, a_2, \ldots, a_\ell)$. By an equation in $\mathcal{F}(a_1, a_2, \ldots, a_\ell)$ with unknowns (variables) $x_1, x_2, \ldots, x_\lambda$, we mean an equality of the form

$$W(x_1, x_2, \ldots, x_\lambda; a_1, a_2, \ldots, a_\ell) = 1 \tag{3.1}$$

where W is word formed from the letters $x_1, x_2, \ldots, x_\lambda, a_1, a_2, \ldots, a_\ell$ and their inverses. A list of words

$$w_1, w_2, \ldots, w_\lambda$$

in the alphabet $a_1, a_2, \ldots, a_\ell, a_1^{-1}, a_2^{-1}, \ldots, a_\ell^{-1}$ is called a solution of the equation-(3.1) if the word $W(w_1, w_2, \ldots, w_\lambda; a_1, a_2, \ldots, a_\ell)$ is equal to 1 in $\mathcal{F}(a_1, a_2, \ldots, a_\ell)$.

Equations that have solutions in the free group are called *satisfiable*, otherwise they are called *unsatisfiable*. As an example, in $\mathcal{F}(a_1, a_2)$ the equation

$$a_1^{-1} x_1 a_2^{-1} x_2^{-1} = 1 \tag{3.2}$$

has many solutions (x_1, x_2), such as (a_2, a_1^{-1}) or $(a_1 a_2, 1)$ or $(1, a_1^{-1} a_2^{-1})$. Thus the above equation is satisfiable. However, in the free group $\mathcal{F}(a_1, a_2)$, there is no solution to

$$x^3 a_2^{-1} a_1^{-1} = 1$$

Fortunately, whether a given equation in a free group is satisfiable or not, is decidable due to Makanin [75]. In particular, Gutiérrez [59] has recently shown that this problem is decidable and is in PSPACE. Though the definition of a pseudo-free group depends on the ability to distinguish effectively between satisfiable and unsatisfiable equations in a free group, the decision procedure need not be in polynomial time. The problem becomes easier in free abelian group. In a free abelian group, an equation can always be rewritten in the form:

$$x_1^{d_1} x_2^{d_2} \ldots x_\lambda^{d_\lambda} = a_1^{e_1} a_2^{e_2} \ldots a_\ell^{e_\ell} \tag{3.3}$$

where $d_1, d_2, \ldots, d_\lambda, e_1, e_2, \ldots, e_\ell$ are integers. Thus, in the free abelian group $\mathcal{FA}(a_1, a_2)$, the equation (3.2) can now be written as

$$x_1 x_2^{-1} = a_1 a_2 \tag{3.4}$$

Equation of the form (3.3) is satisfiable iff for all i, $1 \leq i \leq \ell$, we have

$$gcd(d_1, d_2, \ldots, d_\lambda) | e_i \tag{3.5}$$

The following useful fact, an equation that is satisfiable in $\mathcal{F}(a_1, a_2, \ldots, a_\ell)$ is also satisfiable in $\mathcal{FA}(a_1, a_2, \ldots, a_\ell)$, provides an easy way to prove that an equation is unsatisfiable in a (non-abelian) free group: merely prove that it is unsatisfiable in the corresponding free abelian group. One may note that an equation that is satisfiable in free abelian group need not possesses a solution in the corresponding non-abelian free group. For example consider the following equation in the free group $\mathcal{F}(a_1, a_2)$,

$$x^2 y^2 = a_1 a_2 a_1^{-1} a_2^{-1} \tag{3.6}$$

It has been shown by Lyndon and Newman [72] that in the free group $\mathcal{F}(a_1, a_2)$, the commutator $[a_1, a_2] = a_1 a_2 a_1^{-1} a_2^{-1}$ is never the product of two squares in $\mathcal{F}(a_1, a_2)$. Thus this equation is not satisfiable in $\mathcal{F}(a_1, a_2)$. But the equation (3.6) becomes $x^2 y^2 = 1$ in the corresponding free abelian group $\mathcal{FA}(a_1, a_2)$ and which certainly possesses a solution.

For the rest of this chapter, we will use the following terminology for solutions of equations over *free abelian groups*. Let X and A be two disjoint sets of variable and constant symbols. Thus, as defined earlier, an equation over variables X and constants A in a free abelian group $\mathcal{FA}(A)$, is $E : w_1 = w_2$, where w_1 and w_2 are words over alphabets $\{X \cup X^{-1}\}$ and $\{A \cup A^{-1}\}$ respectively.

A solution to $E : w_1 = w_2$ (over the free abelian group $\mathcal{FA}(A)$) is a function $\sigma : X \to \mathcal{FA}(A)$ such that $\sigma(w_1) = w_2$ (in $\mathcal{FA}(A)$), where σ is extended to words over $X \cup X^{-1}$ homomorphically in the natural way. For example, a solution σ to the equation (3.4) is defined as $\sigma(x_1) = a_1$ and $\sigma(x_2) = a_2^{-1}$. This is a valid solution as $\sigma(w_1) = \sigma(x_1 x_2^{-1}) = \sigma(x_1)\sigma(x_2)^{-1} = a_1 a_2 = w_2$.

For the rest of this chapter, unless otherwise mentioned, $\mathcal{F}(A)$ will always mean a *free abelian group* generated by the set A.

3.4 Pseudo Free (Abelian) Groups

We define pseudo freeness of *computational abelian groups* (abelian groups that admits an efficient algorithmic implementation). In particular, we shall consider infinite families of computational abelian groups implementing infinite families of *finite* (abelian) mathematical groups.

Let G be a computational group. An equation over G is defined by an equation E over variables X and constants A, and a function $\alpha : A \to G$ (α will sample $|A|$ many element from the group G) and is denoted as E_α. A solution to the equation $E_\alpha : w_1 = w_2$ is a function $\xi : X \to G$ such that $\xi(w_1) = \alpha(w_2)$.

Before we formally introduce pseudo-free groups, observe that for any finite group G, given any element $a \in G$, the following equation $E : x^{|G|+1} = a$ is unsatisfiable over the free group $\mathcal{F}(\{a\})$ but has solution $x = a$ over G. In order to properly define pseudo-free groups we need to consider families of groups $\{G_N\}_{N \in \mathcal{N} \subset \{0,1\}^*}$ (N is chosen at random) so that given a randomly chosen N, it should be hard to compute the corresponding group order $|G_N|$. Technically, we assume the set of indices $\mathcal{N}$ is endowed with a sequence of probability distributions $(\mathcal{N}_k)_{k \in \{0,1\}^*}$ such that $\mathcal{N}_k$ can be sampled in (expected) polynomial (in k) time. Typically, $\mathcal{N}_k$ is the uniform distribution over all strings in $\mathcal{N}$ of length k, but other distributions are possible. The set of indices $\mathcal{N}$ together with the polynomial time sampling algorithm and associated probability distribution $\mathcal{N}_k$ is called a **probability ensemble**.

Definition 3.4.1 *Let $\mathcal{G} = \{G_N\}_{N \in \mathcal{N}}$ be a family of computational groups. $\mathcal{G}$ is called pseudo-free if for any probabilistic polynomial (in k, security parameter) time algorithm $\mathcal{A}$ the probability that, on input a polynomial size set A ($|A| = poly(k)$), a randomly chosen group index $N \in \mathcal{N}_k$, and $\alpha : A \to G_N$ (it selects independently at random $|A|$ many group elements according to the computational group sampling procedure) $\mathcal{A}$ outputs (E, ξ) such that E (equation over variables X and constants A) is unsatisfiable over $\mathcal{F}(A)$ and $\xi : X \to G_N$ is a solution to E_α over G_N, is a negligible function in k.*

Signed Quadratic Residues Modulo a Special Blum Integer

We recall here the group of signed quadratic residues (Section 2.4) for a particular form of m. We take $m = N = P \cdot Q$, the product of two prime numbers where P

and Q are of the form $2p+1$ and $2q+1$ respectively with p, q themselves primes. Clearly $P, Q \equiv 3 \pmod 4$ i.e. N is a Blum integer. Thut. For $x \in \mathbb{Z}/N\mathbb{Z}$, we define $|x|$ as the absolute value of x. We will take G to be the group $QR(N)$ of quadratic residues modulo N. Thus

$$QR(N)^+ = \{|x| : x \in QR(N)\}$$

As N is a Blum integer, -1 is not a quadratic residue and thus $\psi(x \to |x|)$ becomes an isomorphism. Now as $QR(N)$ is a cyclic group of order pq ($|QR(N)| = \frac{\phi(N)}{4} = \frac{4pq}{4} = pq$) and ϕ is an isomorphism, $QR(N)^+$ is also a cyclic group of order pq. Another fundamental property of the group $QR(N)^+$, where it crucially differs from $QR(N)$, is that membership in $QR(N)^+$ is efficiently recognizable whereas computing square root still remains hard. It is due to the fact that as a set $QR(N)^+ = J(N) \cap (0, \frac{N-1}{2}]$ where $J(N) = \left\{x \in \mathbb{Z}/N\mathbb{Z}^* : \text{ Jacobi symbol } \left(\frac{x}{N}\right) = 1\right\}$. One may note that recognizing membership of elements in $J(N)$ can be performed efficiently as in particular Jacobi symbols can be efficiently computed. Thus $QR(N)^+$ is a "gap group" [86], in which the computational problem (i.e computing a square root) is as hard as factoring, whereas the corresponding decisional problem (i.e., deciding if an element is a signed square) is easy.

As we noted earlier in the general case, $x \in QR(N)^+$ does not imply that x is a quadratic residue modulo N. For $-y \in QR(N)$ implies $|-y| = y \in QR(N)^+$. But as $y = (-1) \cdot (-y)$ and -1 is quadratic non-residue, y is quadratic non-residue. Thus elements of $QR(N)^+$ can be characterized as follows. For an element $x \in QR(N)^+$, either $x \in QR(N)$ or there exists a unique quadratic residue $y \in QR(N)$ such that $-x \equiv y \pmod N$. This y is nothing but the element $N - x$.

3.4.1 Strong Signed QR-RSA Assumption

We let, for $k \geq 1$, $\mathcal{N}_k$ be the set of all safe prime products of bit-size bounded by k with some standard probability distribution (used in cryptography) on $\mathcal{N}_k$. We first recall some computational assumptions that are conjectured to be asymptotically hard and are related to this work.

The *Strong RSA assumption* was introduced by Barić and Pfitzmann [6] and by Fujisaki and Okamoto [52] (see also [38]).

This assumption differs from the RSA assumption in that the adversary can select the public exponent e. The adversary's task is to output, given a random

integer $N \in \mathcal{N}_k$, and a randomly chosen group element $\gamma \in \mathbb{Z}/N\mathbb{Z}^*$, an integer $e > 1$ and a group element $\xi \in \mathbb{Z}/N\mathbb{Z}^*$ such that $\xi^e \equiv \gamma \pmod{N}$. This may well be easier than solving the RSA problem, thus the assumption about its hardness is stronger than the RSA assumption. The Strong RSA assumption is basis for a variety of cryptographic constructions.

Strong QR-RSA problem [38]: given a random integer $N \in \mathcal{N}_k$, and a randomly chosen quadratic residue $\gamma \in QR(N)$, output an integer $e > 1$ and a group element $\xi \in \mathbb{Z}/N\mathbb{Z}^*$ such that $\xi^e \equiv \gamma \pmod{N}$.

In [38], it has been observed that strong QR-RSA problem is as hard as strong RSA problem. For our work we propose a new variant of strong RSA problem. We call it strong signed QR-RSA (SQR-RSA) problem.

Strong SQR-RSA problem: given a random integer $N \in \mathcal{N}_k$, and a randomly chosen $\gamma \in QR(N)^+$, output an integer $e > 1$ and a group element $\xi \in \mathbb{Z}/N\mathbb{Z}^*$ such that $\xi^e \equiv \gamma \pmod{N}$.

We now prove strong SQR-RSA problem is as hard as strong QR-RSA problem.

Theorem 3.4.1 *If the strong QR-RSA problem is asymptotically hard where the underlying modulus N is chosen randomly from the set $\mathcal{N}_k$ of all safe prime products of bit size bounded by k, then the strong SQR-RSA problem is also asymptotically hard for the same N.*

Proof : Assume the existence of a PPT algorithm $\mathcal{A}$ which, given an input (N, γ), where $\gamma \in QR(N)^+$, outputs an element $\xi \in \mathbb{Z}/N\mathbb{Z}^*$ and an integer $e > 1$ such that $\xi^e \equiv \gamma \pmod{N}$. We will use $\mathcal{A}$ as an oracle to solve strong QR-RSA problem. Let an input of the strong QR-RSA problem be given as (N, γ) where $\gamma \in QR(N)$. Consider the following cases:
Case-1: Check if $0 < \gamma \leq \frac{N-1}{2}$. If yes, then $\gamma \in QR(N)^+$. Pass (N, γ) to $\mathcal{A}$. Clearly the solution given by $\mathcal{A}$ will also be a solution to this strong QR-RSA instance.
Case-2: Let $-\frac{N-1}{2} \leq \gamma < 0$. Then $\gamma = -t$ where $0 < t \leq \frac{N-1}{2}$. As $\gamma \in QR(N)$, $t = |\gamma| \in QR(N)^+$. With (N, t) as an input to $\mathcal{A}$, it will output an element $\xi \in \mathbb{Z}/N\mathbb{Z}^*$ and a positive integer $e > 1$ such that $\xi^e \equiv t \pmod{N}$. Here we claim that this e is necessarily odd. If e is even, t becomes a quadratic residue which is

not possible as $-t = \gamma \in QR(N)$. Thus we now output $\xi' = -\xi$ as a solution to (N, γ) as we can see,

$$(\xi')^e \equiv (-\xi)^e \equiv -1 \cdot \xi^e \equiv -t \equiv \gamma \pmod{N}$$

□

We also need the following lemma in the proof of our main theorem.

Lemma 3.4.1 *Let $N = p \cdot q$, product of two safe primes. $QR(N)$ and $QR(N)^+$ denotes the subgroup of quadratic residues and signed quadratic residues respectively. For an element x, chosen randomly from $QR(N)^+$, the probability that x also belongs to $QR(N)$ is $\frac{1}{2} + o(1)$, where $o(1)$ is negligible.*

Proof : Note that for an element x chosen uniformly at random in an interval $[1, m]$, where m is a multiple of p (p is prime), $x \pmod{p}$ is uniformly distributed in $[0, p-1]$. So for $x \in_R \mathbb{Z}/N\mathbb{Z}^* \cap [1, \frac{N-1}{2}]$, $x \pmod{p}$ and $x \pmod{q}$ are independent and approximately (elements not co-prime to N are ignored) uniformly distributed in $[1, p-1]$ and $[1, q-1]$ respectively.

So for x chosen uniformly at random in $\mathbb{Z}/N\mathbb{Z}^* \cap [1, \frac{N-1}{2}]$, $\text{Prob}[x \in QR(N)] = \text{Prob}[x \pmod{p} \in QR(p) \text{ and } x \pmod{q} \in QR(q)] = \text{Prob}[x \pmod{p} \in QR(p)] \cdot \text{Prob}[x \pmod{q} \in QR(q)] = (\frac{1}{2} + o(1)) \cdot (\frac{1}{2} + o(1)) = \frac{1}{4} + o(1)$.

So $QR(N)$ constitutes approximately $\frac{1}{4}$th of $\mathbb{Z}/N\mathbb{Z}^* \cap [1, \frac{N-1}{2}]$. The cardinality of $\mathbb{Z}/N\mathbb{Z}^* \cap [1, \frac{N-1}{2}]$ and $J(N) \cap [1, \frac{N-1}{2}]$ are $\frac{\phi(N)}{2}$ and $\frac{\phi(N)}{4}$ respectively. As $QR(N) \cap [1, \frac{N-1}{2}] \subset J(N) \cap [1, \frac{N-1}{2}]$, the portion of $QR(N) \cap [1, \frac{N-1}{2}]$ in $J(N) \cap [1, \frac{N-1}{2}]$ is approximately (i.e. modulo a negligible quantity) $\frac{1}{2}$. Hence for an element x, chosen randomly from $QR(N)^+ = J(N) \cap [1, \frac{N-1}{2}]$, the probability that x also belongs to $QR(N)$ is $\frac{1}{2} + o(1)$. □

Remark 3.4.1 *In 1918, towards finding the distribution of quadratic residues and quadratic non-residues modulo a prime, Polya [91] and Vinogradov [113], proved independently the following remarkable inequality,*

$$\left| \sum_{a=N+1}^{N+M} \left(\frac{a}{p} \right) \right| \leq \sqrt{p} \cdot \log p$$

where p is prime and N, M are arbitrary ($0 \leq N < N + M < p$). Vinogradov also proved a generalization of their result in which the prime p is replaced by a composite k. Sharper estimate of the above inequality for prime modulus was obtained by D.A. Burgess [27, 28].

3.5 $\mathbb{Z}/N\mathbb{Z}^*$ is Pseudo-free

The proof of our main Theorem below is along the lines of the proof of Theorem-2 in [77] with necessary modifications. We have essentially managed to sample elements uniformly at random from an **isomorphic copy** $(QR(N)^+)$ of $QR(N)$ in $\mathbb{Z}/N\mathbb{Z}^*$ to prove $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free.

Theorem 3.5.1 *Assume the strong RSA problem is asymptotically hard with respect to a distribution ensemble $\mathcal{N}$ over the safe prime products. Then the computational group family of $\mathbb{Z}/N\mathbb{Z}^*$ of invertible integers modulo $N \in \mathcal{N}$ (with the modular multiplication group operation, and uniform sampling procedure over the signed quadratic residue group $QR(N)^+$) is pseudo-free with respect to the same distribution ensemble $\mathcal{N}$.*

Assume that $\mathbb{Z}/N\mathbb{Z}^*$ is not pseudofree, i.e. there is a PPT algorithm $\mathcal{A}$ that on input a randomly chosen $N \in \mathcal{N}_k$ and random group elements $\alpha : A \rightarrow QR(N)^+$ (for some polynomial-size set A), outputs an equation $E : w_1 = w_2$ (over constants in A and variables in X) which is unsatisfiable over $\mathcal{F}(A)$, together with a solution $\xi : X \rightarrow \mathbb{Z}/N\mathbb{Z}^*$ to E_α over the group $\mathbb{Z}/N\mathbb{Z}^*$.

We use $\mathcal{A}$ to solve the strong SQR-RSA problem for the same distribution of the modulus N. Thus, given a randomly chosen $N \in \mathcal{N}_k$ and $\gamma \in QR(N)^+$, we compute an integer $e > 1$ and a group element $\xi \in \mathbb{Z}/N\mathbb{Z}^*$ such that $\xi^e \equiv \gamma \pmod{N}$. By Theorem 3.4.1 this also implies an algorithm to solve the strong QR-RSA Problem and which in turn yield an algorithm [38] which will solve strong RSA problem. The reduction works as follows.

Let (N, γ) be an instance of the strong SQR-RSA problem. We begin by checking if γ is a generator of $QR(N)^+$. Below we outline a sufficient condition for γ to be a generator of $QR(N)^+$. For this we need the following result [77] which for elements γ in $QR(N)$ checks if γ is a generator for $QR(N)$.

Lemma 3.5.1 *[77] Let $N = P \cdot Q$ be the products of two distinct safe primes, and $\gamma \in QR(N)$ a quadratic residue. Then γ is a generator for $QR(N)$ iff $gcd(\gamma - 1, N) = 1$.*

Lemma 3.5.2 *Let $N = P \cdot Q$ be the products of two distinct safe primes, and $\gamma \in QR(N)^+$. If,*

$$gcd(\gamma - 1, N) = 1 \text{ and } gcd(-\gamma - 1, N) = 1$$

then γ is a generator for $QR(N)^+$.

Proof : We know for a Blum integer N, the map $\phi : QR(N) \rightarrow QR(N)^+$ ($\phi(x) = |x|$) is a group isomorphism. The inverse of ϕ ($\phi^{-1} : QR(N)^+ \rightarrow QR(N)$) is defined as follows:

$$\phi^{-1}(x) = \begin{array}{ll} x & \text{if } x \in QR(N) \\ -x & \text{if } x \notin QR(N) \end{array}$$

We are given a Blum integer N and $\gamma \in QR(N)^+$ such that $gcd(\gamma - 1, N) = 1$ and $gcd(-\gamma - 1, N) = 1$. We will show that γ is a generator of $QR(N)^+$.

- Case-1: Say $\gamma \in QR(N)$. Then $\phi^{-1}(\gamma) = \gamma$. Now as $gcd(\gamma - 1, N) = 1$, by Lemma 3.5.1, γ is a generator of $QR(N)$. Now as a generator is mapped into a generator under isomorphism and $\phi(\gamma) = |\gamma| = \gamma \in QR(N)^+$, γ is a generator of $QR(N)^+$.

- Case-2: Say $\gamma \notin QR(N)$. Then $\phi^{-1}(\gamma) = -\gamma \in QR(N)$. Now as $gcd(-\gamma - 1, N) = 1$, by Lemma 3.5.1, $-\gamma$ is a generator of $QR(N)$. Similarly as generators are mapped into generators under isomorphism and $\phi(-\gamma) = |-\gamma| = \gamma \in QR(N)^+$, γ is a generator of $QR(N)^+$.

□

So for given $\gamma \in QR(N)^+$ we first compute $g = gcd(\gamma - 1, N)$ and $g' = gcd(-\gamma - 1, N)$. Since $N = P \cdot Q$, we have $g, g' \in \{1, P, Q, PQ\}$. We consider below all the cases.

- Case-1: Either of g or g' is in $\{P, Q\}$. W.l.o.g. say $g \in \{P, Q\}$. Then we can easily compute $\phi(N) = (P - 1) \cdot (Q - 1)$ and output $(\xi, e) = (\gamma, \phi(N) + 1)$ as a solution to the strong SQR-RSA problem input (N, γ).

- Case-2: As $\gamma \in QR(N)^+$, therefore $0 < \gamma \leq \frac{N-1}{2}$. Thus $gcd(-\gamma - 1, N)$ will never be N. So if $g' \notin \{P, Q\}$ then g' must be equal to 1. Now if $g = N$, then $\gamma \equiv 1 \pmod{N}$ and we can immediately output a solution to the strong SQR-RSA problem input (N, γ), e.g., $(\xi, e) = (1, 2)$

- Case-3: g and g' are both equal to 1. Then by Lemma 3.5.2, γ is a generator for $QR(N)^+$.

For rest of the proof we assume that γ is a generator of $QR(N)^+$. Now we generate an input instance (N, α) for algorithm $\mathcal{A}$ where for a polynomial sized set A, α samples $|A|$ many elements from $QR(N)^+$. Since $\mathcal{A}$ works only with non negligible probability, we need the input values $\alpha(a)$ to be distributed (almost) uniformly at random over $QR(N)^+$. The following lemma shows that γ, being a generator of $QR(N)^+$, can be used to sample $QR(N)^+$ almost uniformly at random.

Lemma 3.5.3 *[77] For any cyclic group G and generator $\gamma \in G$, if $\nu \in \{0, \ldots, B-1\}$ is chosen uniformly at random, then the statistical distance between γ^ν and the uniform distribution over G is at most $\frac{|G|}{2B}$.*

So for any polynomial-size set A, we sample $|A|$ many elements almost uniformly at random from $QR(N)^+$ as follows. For $a \in A$, choose $\nu_a \in \{0, \ldots, N \cdot |A| \cdot K - 1\}$ uniformly at random for some super polynomial function $K(k) = k^{\omega(1)}$ and set $\alpha(a) = \gamma^{\nu_a}$ (note that $\gamma^{\nu_a} = |\gamma^{\nu_a} \pmod N|$). There are two things to check here.

1. $\alpha(a)$ is uniformly distributed over $QR(N)^+$
2. Among all the assignments $\alpha : A \to QR(N)^+$ to sample $QR(N)^+$ almost uniformly at random, our choice of α where $\alpha(a) = \gamma^{\nu_a}$ is uniformly selected.

For the first property, By Lemma 3.5.3, the statistical distance between $\alpha(a)$ and the uniform distribution over $QR(N)^+$ is at most $\frac{|QR(N)^+|}{2N|A|K} \leq \frac{1}{2|A|K}$. For later, we know that algorithm $\mathcal{A}$ will be successful with non-negligible probability on input N and assignment $\alpha : A \to QR(N)^+$ provided α is distributed uniformly at random. Since the value $\alpha(a)$ are independently chosen, the statistical distance between α and an uniformly chosen assignment is at most $\frac{1}{2K} = \frac{1}{k^{\omega(1)}}$ and thus $\mathcal{A}$ succeeds on input α ($\alpha(a) = \gamma^{\nu_a}$) with non-negligible probability $\delta(k) - \frac{1}{K(k)}$ where $\delta(k)$ is the non-negligible probability of $\mathcal{A}$'s success on input (N, α) when the assignment α is distributed uniformly at random.

In the rest of the proof, we assume $\mathcal{A}$ is successful, and we consider the conditional success probability of the reduction and show that it will turn out to be at least $\frac{3}{16} + o(1)$, where $o(1)$ is negligible.

We now first workout some more details about our assignment α. We know, for every $a \in A$, we set $\alpha(a) = \gamma^{\nu_a}$ for a randomly chosen $\nu_a \in \{0, \ldots, N \cdot |A| \cdot K - 1\}$. With each of this ν_a one can associate two unique numbers modulo $\frac{\phi(N)}{4} = pq$. They are respectively, the remainder w_a and the quotient z_a of ν_a when divided

by pq. Thus $w_a \equiv \nu_a \pmod{pq}$ and $z_a = \frac{\nu_a - w_a}{pq}$. Eventhough they exist, given ν_a, it is hard to compute z_a and w_a due to the unavailability of $\phi(N)$. Thus we will use w_a and z_a only in the analysis of the reduction.

Notice that, given w_a, the conditional distribution of z_a is uniform over the set

$$S_a = \left\{0, \ldots, \lfloor \tfrac{N \cdot |A| \cdot K - 1 - w_a}{pq} \rfloor \right\}$$

The size of S_a is at least

$$|S_a| \geq 1 + \lfloor \tfrac{N \cdot |A| \cdot K - 1 - w_a}{pq} \rfloor \overset{[w_a \leq pq-1]}{\geq} \lfloor \tfrac{N \cdot |A| \cdot K}{pq} \rfloor \overset{[N > 4pq]}{\geq} 4 \cdot |A| \cdot K \geq 4$$

Also, given w_a, the value of $\alpha(a) = \gamma^{\underline{\nu_a}} = \gamma^{\underline{w_a}}$ is uniquely determined, and z_a is uniformly distributed over the set S_a independently from α, E and ξ.

Assume that $\mathcal{A}$ is successful, i.e., $E : w_1 = w_2$ is not satisfiable over $\mathcal{F}(A)$, and $\xi : X \to \mathbb{Z}/N\mathbb{Z}^*$ is a valid solution to E_α, i.e. $\xi(w_1) = \alpha(w_2)$. Like typical reduction, we use equation E and solution ξ to output a solution to strong SQR-RSA problem input (N, γ). This is done in two steps. First, we transform equation E and solution ξ (to E_α) into a new equation E' (unsatisfiable over the same $\mathcal{F}(A)$) containing only one variable and a solution ξ' to E'_α. Then E' and ξ' will be used to solve the given instance (N, γ) of strong SQR-RSA problem.The following lemma will show how to transform (E, ξ) into an univariate equation and solution (E', ξ').

Lemma 3.5.4 *[77] For any computational group family $\mathcal{G}$, there is a polynomial time algorithm that on input a group G from $\mathcal{G}$, and an equation E, over a set X of variables and set A of constants, and a variable assignment $\xi : X \to G$, outputs a univariate equation E', over the same set A of constants, and a value $\xi' \in G$, such that*

- *Prop-1: if E is unsatisfiable over the free group $\mathcal{F}(A)$, then E' is also unsatisfiable over $\mathcal{F}(A)$, and*
- *Prop-2: for any assignment $\alpha : A \to G$, if ξ is a solution to E_α then ξ' is a solution to E'_α.*

In particular, for input equation $E : \prod_{x \in X} x^{e_x} = \prod_{a \in A} a^{d_a}$, where $e_x, d_a \in \mathbb{Z}$ and input assignment $\xi : X \to G$, the algorithm $\mathcal{A}$ outputs equation $E' : x^e = \prod_{a \in A} a^{d_a}$ and the value ξ' in G, $\xi' = \prod_{x \in X} \xi(x)^{\frac{e_x}{e}}$.

At this point, as an output of Lemma 3.5.4, we are having a univariate equation $E' : x^e = \prod_{a \in A} a^{d_a}$ which is unsatisfiable over $\mathcal{F}(A)$ and a solution $\xi' \in \mathbb{Z}/N\mathbb{Z}^*$ to E'_α, i.e.,

$$(\xi')^e = \prod_{a \in A} \alpha(a)^{d_a} = \gamma^{\underline{d}} \tag{3.7}$$

where $d = \sum_{a \in A} \nu_a d_a$. Notice that E' is satisfiable over the free group $\mathcal{F}(A)$ iff $e \mid gcd(d_a : a \in A)$. So necessarily here, $e \nmid gcd(d_a : a \in A)$.

In the rest of the proof we distinguish various cases, depending on the all possible values of $gcd(e, pq)$ and they are,

- **Case-1:** $e = 0$.
- **Case-2:** $e \neq 0$ and $gcd(e, pq) = pq$.
- **Case-3:** $e \neq 0$ and $gcd(e, pq) \in \{p, q\}$.
- **Case-4:** $e \neq 0$ and $gcd(e, pq) = 1$.

Case-1: In this case we first calculate the probability that $d = \sum_a \nu_a d_a \neq 0$.

Lemma 3.5.5 *[77] Given α, $e = 0$ and $\{d_a : a \in A\}$ such that $e \nmid gcd\{d_a : a \in A\}$, the conditional probability that $d = \sum_{a \in A} \nu_a d_a \neq 0$ is at least $\frac{3}{4}$.*

Assuming $d \neq 0$ (which, by Lemma 3.5.5, happens with probability at least $\frac{3}{4}$), we have $|d| + 1 > |d| \geq 1$. Now,

$$\gamma^{\underline{|d|}} = \gamma^{\underline{\pm d}} \overset{\text{[equation 3.7]}}{=} (\xi')^{\pm e} = (\xi')^{\pm 0} = 1$$

But $\gamma^{\underline{\pm d}} = 1$,
$\Rightarrow |\gamma^{\pm d} \pmod N| = 1$,
$\Rightarrow \gamma^{\pm d} \pmod N = \pm 1$.
We want to rule out the case when $\gamma^{\pm d} \pmod N = -1$. γ is given in $QR(N)^+$. By Lemma 3.4.1, the probability that γ is also in $QR(N)$ is $\frac{1}{2} + o(1)$. Now assume $\gamma \in QR(N)$ (happens with probability $\frac{1}{2} + o(1)$), then $\gamma^{\pm d} \pmod N \neq -1$ as $-1 \notin QR(N)$. Thus we have $\gamma^{\pm d} \pmod N = 1$. Now we can output $(\gamma, |d| + 1)$ as a valid solution to strong SQR-RSA problem input (N, γ) as

$$\gamma^{|d|+1} = \gamma \cdot \gamma^{|d|} = \gamma \cdot \gamma^{\pm d} \overset{[\gamma^{\pm d} \pmod N = 1]}{=} \gamma$$

Thus $(\gamma, |d|+1))$ is a valid solution provided $d \neq 0$ and $\gamma \in QR(N)$. These two events are clearly independent. Thus
Prob $[(\gamma, |d|+1)$ is a valid solution]
$\geq$ Prob $[d \neq 0$ and $\gamma \in QR(N)]$
$=$ Prob $[d \neq 0]\cdot$Prob $[\gamma \in QR(N)] \overset{[\text{Lemma 3.5.5,Lemma 3.4.1}]}{\geq} \frac{3}{4}\cdot(\frac{1}{2}+o(1)) = \frac{3}{8}+o(1)$.
Case-2: As $\gamma \in QR(N)^+$, $\gamma^{\underline{\phi(N)/4}} = \gamma^{\underline{pq}} = 1$ and therefore as $pq \mid e$, $\gamma^{\underline{e}} = 1$. Now $\gamma^{\underline{e}} = |\gamma^e \pmod N|$. Thus $\gamma^{\underline{e}} = 1$ implies $|\gamma^e \pmod N| = 1$, i.e., $\gamma^e \pmod N = \pm 1$. We want to rule out the case when $\gamma^e \pmod N = -1$. Like earlier, assuming γ to be quadratic residue will help us rule out this case and $\gamma \in QR(N)$ happens with probability $\frac{1}{2}+o(1)$. So we now assume that $\gamma \in QR(N)$ and out put $(\gamma, |e|+1)$ as a valid solution to the strong SQR-RSA problem instance (N, γ) as

$$\gamma^{|e|+1} = \gamma \cdot \gamma^{\pm e} = \gamma \cdot 1 = \gamma$$

So Prob$[(\gamma, |e|+1)$ is a valid solution$] \geq$ Prob$[\gamma \in QR(N)] = \frac{1}{2}+o(1)$. We remark that, although we cannot compute $gcd(e, pq)$ (or even check if $gcd(e, pq) = pq$) because pq is not known, we can guess that this is the case, and simply check if $(\gamma, |e|+1)$ is indeed a solution to the given strong SQR-RSA problem input. Similar remarks apply to the other cases below.
Case-3: Here $o(\gamma^{\underline{e}}) = \frac{pq}{gcd(e,pq)} \in \{p, q\}$. Thus $\gamma^{\underline{e}}$ is not a generator of $QR(N)^+$. Assume that $\gamma^{\underline{e}}$ also belongs to $QR(N)$ and this happens with probability $\frac{1}{2}+o(1)$. Now as $\gamma^{\underline{e}}$ is not a generator of $QR(N)^+$, it is also not a generator of $QR(N)$. Then by Lemma 3.5.1 $gcd(\gamma^{\underline{e}} - 1, N) \neq 1$. Again as γ is a generator of $QR(N)^+$ and $gcd(e, pq) \in \{p, q\}$ implies that $\gamma^{\underline{e}} \not\equiv 1 \pmod N$. Thus $gcd(\gamma^{\underline{e}} - 1, N) \neq N$. Then $gcd(\gamma^{\underline{e}} - 1, N) \in \{P, Q\}$. So we can compute $\phi(N)$, and output the solution $(\gamma, \phi(N)+1)$ to the strong SQR-RSA problem input (N, γ). So the probability of success in outputting a valid solution in this case depends on the probability that given $\gamma \in QR(N)^+$ is also in $QR(N)$ and this probability is $\frac{1}{2} + o(1)$.
Case-4: In this case, first we see that $e \nmid d$ with probability at least $\frac{3}{8}$.

Lemma 3.5.6 *[77] Given $\alpha, gcd(e, pq) = 1$, and $\{d_a : a \in A\}$ such that $e \nmid gcd\{d_a : a \in A\}$, the conditional probability that e does not divide $d = \sum_{a \in A} \nu_a d_a$ is at least $\frac{3}{8}$.*

Let $e' = \frac{e}{t}$ and $d' = \frac{d}{t}$ where $t = gcd(e, d)$. Assuming $e \nmid d$ (which, by Lemma 3.5.6, happens with probability at least $\frac{3}{8}$), we have $t \neq e$, and consequently $e' = \frac{e}{t} > 1$. Also note that from $gcd(e, pq) = 1$ and $t \mid e$, we get $gcd(t, |QR(N)^+|) = gcd(t, pq) = 1$. Now we have as output (equation 3.7) from the algorithm $\mathcal{A}$,
$(\xi')^e = \gamma^{\underline{d}}$
i.e. $(\xi')^e \equiv \pm\gamma^d \pmod N$
i.e. $((\xi')^e)^2 \equiv (\pm\gamma^d)^2 \equiv \gamma^{2d} \pmod N$
i.e. $(\xi')^{2e't} \equiv \gamma^{2d't} \pmod N$.
Now clearly $(\xi')^{2e'}, \gamma^{2d'} \in QR(N)$, and as $gcd(t, |QR(N)|) = gcd(t, |QR(N)^+|) = 1$ we have $(\xi')^{2e'} \equiv (\xi')^{2et^{-1} \pmod{|QR(N)|}} \equiv \gamma^{2dt^{-1} \pmod{|QR(N)|}} \equiv \gamma^{2d'} \pmod N$. At this point, we have (γ, ξ', e', d') such that $(\xi')^{2e'} \equiv \gamma^{2d'} \pmod N$, $e' > 1$ and $gcd(e', d') = 1$. $(\xi')^{2e'} \equiv \gamma^{2d'} \pmod N$ tells that $N \mid ((\xi')^{e'} - \gamma^{d'})((\xi')^{e'} + \gamma^{d'})$. If $(\xi')^{e'} \neq \pm\gamma^{d'}$, then computing $gcd(N, (\xi')^{e'} - \gamma^{d'})$ and $gcd(N, (\xi')^{e'} + \gamma^{d'})$ will surely yields $\{P, Q\}$ and we can immediately output a solution $(\gamma, \phi(N) + 1)$ to the strong SQR-RSA problem input (N, γ). Now we consider the case when $(\xi')^{e'} = \pm\gamma^{d'}$. If $(\xi')^{e'} = \gamma^{d'}$, use the Euclidean algorithm to compute two integers e'' and d'' such that $e'e'' + d'd'' = gcd(e', d') = 1$. Now we output $((\xi')^{d''}\gamma^{e''}, e')$ as a valid solution to the strong SQR-RSA problem input (N, γ) as $e' > 1$ (as a consequence of Lemma 3.5.6) and $((\xi')^{d''}\gamma^{e''})^{e'} = (\xi')^{e'd''}\gamma^{e'e''} = \gamma^{d'd''+e'e''} = \gamma$. Finally we consider the case when $(\xi')^{e'} = -\gamma^{d'}$. In this case we assume that γ belongs to $QR(N)$ and this happens with probability $\frac{1}{2} + o(1)$. As $\gamma \in QR(N)$, implies $\gamma^{d'}$ is also in $QR(N)$. Also as N is a Blum integer, $-1 \notin QR(N)$ and thus $-\gamma^{d'} \notin QR(N)$. As $(\xi')^{e'} = -\gamma^{d'}$, therefore e' is necessarily odd (e' even implies $(\xi')^{e'} = -\gamma^{d'} \in QR(N)$). Thus $\gamma^{d'} = -(\xi')^{e'} = (-1)^{e'} \cdot (\xi')^{e'} = (-\xi')^{e'}$. So by replacing ξ' with $-\xi'$, this last case $(\xi')^{e'} = -\gamma^{d'}$ reduces to the previous one $(\xi')^{e'} = \gamma^{d'}$.

So in Case-4 ($e \neq 0$ and $gcd(e, pq) = 1$), the probability that we will successfully output a valid solution depends on the two independent events and they are $e \nmid d$ and $\gamma \in QR(N)$ and the probability that both these events occur is atleast $\frac{3}{8} \cdot (\frac{1}{2} + o(1)) = \frac{3}{16} + o(1)$.

3.6 Conclusion

In this chapter we have proved that the RSA group $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free modulo the following constraints:

- N is the product of two safe primes,
- elements are sampled uniformly at random from the subgroup $QR(N)^+$ of signed quadratic residues,
- the proof is based on the hardness assumption of the strong RSA problem.

Some of the immediate open problems that remains are:

- whether $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free even when N is product of two arbitrary primes and elements are sampled uniformly at random from the whole group $\mathbb{Z}/N\mathbb{Z}^*$. There is some hope for the first part of the above problem, as one expects the adversary to face more difficulty when N is a product of two arbitrary primes than the well structured safe prime product case. But then the challenge is to modify the proof in a framework where the structured results for Blum integers are no longer available.
- Prove that $\mathbb{Z}/N\mathbb{Z}^*$ is pseudo-free assuming that factoring N is hard. This problem is apparently very hard, as it would imply that solving the RSA problem is at least as hard as factoring, a long standing open problem in cryptography.

Chapter 4

The Congruence $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ and its Applications

4.1 Introduction

The quadratic congruence $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ arose (in)directly many a times in the field of cryptography. Special forms of this congruence, depending on the associated large numbers N, R, S, are extremly interesting and useful object for cryptography applications. We primarily discuss and use the following form

$$Rx^2 + Sy^2 \equiv 1 \pmod{N} \tag{4.1}$$

where N is an RSA modulus (product of two distinct primes p, q) and $R, S \in \mathbb{Z}/N\mathbb{Z}^*$. Among several cryptography applications of (4.1), the following are interesting and related to our work.

- In the well known signature scheme by Ong, Schnorr, and Shamir [87], the public key consists of integers N and k, where N is a large odd composite integer (say an RSA modulus) whose factorization is kept secret; k is in general of similar size to N. A valid signature of the message m, where $m \in \mathbb{Z}/N\mathbb{Z}$, is any pair of integers x, y such that

  $$x^2 + ky^2 \equiv m \pmod{N}$$

  The equation $x^2+ky^2 \equiv m \pmod{N}$ can be rewritten as $(m^{-1})x^2+(m^{-1}k)y^2 \equiv 1 \pmod{N}$ for $m \in \mathbb{Z}/N\mathbb{Z}^*$. Pollard and Schnorr [90] provided as part of

their cryptanalysis of this signature scheme, a beautiful algorithm that finds solutions to the equation $x^2 + ky^2 \equiv m \pmod{N}$ without knowing factorization of N. This algorithm runs in polynomial time under the assumption of the generalized Riemann hypothesis (GRH). Indeed, a solution to $x^2 + ky^2 \equiv m \pmod{N}$ gives a solution to the equation (4.1).

- As part of solving a long standing open problem, i.e., construction of a space efficient identity-based encryption scheme without using pairings, Boneh-Gentry-Hamburg (BGH) [24] had to design an algorithm, that outputs a **unique** solution to the equation (4.1) corresponding to the input parameters, without the knowledge of the factorization of N. To solve this problem, they lift (4.1) to the integers and consider the ternary quadratic form

$$\tilde{R}x^2 + \tilde{S}y^2 - z^2 = 0 \tag{4.2}$$

where $\tilde{R}, \tilde{S}, x, y, z \in \mathbb{Z}$ and $\tilde{R} \equiv R \pmod{N}$, $\tilde{S} \equiv S \pmod{N}$. A solution to (4.2) in $\mathbb{Z}$ gives a solution to (4.1) in $\mathbb{Z}/N\mathbb{Z}$. At this juncture a couple of things that need to be addressed are: first the existence of an efficient algorithm to solve (4.2) and secondly to obtain unique solutions to equation (4.1) corresponding to the input parameters. The equation (4.2) is known as Legendre equation and a great deal of algorithms exist to solve them. For example, in [104, Chap. IV Section 3] or [110, Chap. IV Section 3.3], the solution of $ax^2+by^2+cz^2 = 0$ is deduced from the solution of $\tilde{a}x^2+\tilde{b}y^2+\tilde{c}z^2 = 0$, where the new coefficients are smaller than the old ones. However, this reduction depends on the possibility of extracting square roots modulo a, b or c, which is only possible if we know the factorization of abc. During the whole algorithm, the total number of factorization is quite large. The worst drawback is certainly that the number that have to be factored may also be quite large. Solving quadratic equations with these algorithms uses only a few lines in theory, but is extremly slow in practice. Fortunately, a couple of algorithms exist, which do not factor any other integers than a, b, and c. It seems possible, in general, to avoid these three factorizations. Such an algorithm is given in [40] and in practice, it indeed runs fast. The important part of BGH's work is to adapt the algorithm of [40] to obtain unique solutions to equation (4.1) corresponding to the input parameters. An overview of this method is given at the end of the Section 4.3.3.

In this chapter we present our work on the following three topics. The topics are:

A. Characterization and counting of solutions of the quadratic congruence $Rx^2 + Sy^2 \equiv 1 \pmod{N}$, where N is an RSA modulus and $R, S \in \mathbb{Z}/N\mathbb{Z}$.

B. An efficient (time and space) identity-based encryption scheme without pairing, a variant of BGH scheme [24].

C. An IND-CPA secure public key encryption scheme in the standard model.

The topics are dependent on each other, in particular schemes given in B and C use properties of the quadratic congruence given in A and the scheme from C depends upon the scheme given in B.

4.2 A. Characterization and Counting of Solutions to $Rx^2 + Sy^2 \equiv 1 \pmod{N}$

It must have been apparent by now that the primary concern and difficulty related to the quadratic congruence (4.1) is to efficiently find solutions when the factorization of the modulus N is not known. This problem becomes much easy when a square root of either R or S (modulo N) is known. In this Section, we characterize solutions of (4.1) in terms of the square roots of the coefficients (R, S). The results obtained here will be used in Section (4.3) and Section (4.4). We provide, using elementary methods,

- a characterization of solutions (x_0, y_0) of (4.1) and
- a count of the number of solutions (x_0, y_0) of (4.1),

when S is a quadratic residue modulo N. We divide the proof into several subcases. To begin with, we have the following lemma.

Lemma 4.2.1 *Let N be a prime and S be a quadratic residue modulo N. Then the following are true:*

(a) For any solution (x_0, y_0) with $gcd(x_0, N) = 1$ of equation (4.1), there exist a $t \in \mathbb{Z}/N\mathbb{Z}^*$ with $R + St^2 \in \mathbb{Z}/N\mathbb{Z}^*$ such that

$$(x_0, y_0) = \left(\frac{-2st}{R + St^2}, \frac{R - St^2}{s(R + St^2)} \right)$$

where s is a square root of S modulo N.

(b) There is an one-one correspondence between the set $A_N \stackrel{\Delta}{=} \{t \in \mathbb{Z}/N\mathbb{Z}^* : gcd(R + St^2, N) = 1\}$ *and the solutions* (x_0, y_0) *with* $gcd(x_0, N) = 1$ *of equation (4.1).*

(c) The number of solutions (x_0, y_0) *with* $gcd(x_0, N) = 1$ *of equation (4.1) is*

$$= \begin{cases} N-1 & \text{if } -R \in NQR(N) \\ N-3 & \text{if } -R \in QR(N) \end{cases}$$

Proof : Let (x_0, y_0) with $gcd(x_0, N) = 1$ be a solution of equation (4.1). Define $t = \frac{(y_0 - 1/s)}{x_0}$ where $s^2 \equiv S \pmod N$. Then $gcd(t, N) = 1$. Compute,

$$R + St^2 = R + S\frac{(y_0 - 1/s)^2}{x_0^2} = \frac{Rx_0^2 + Sy_0^2 + 1 - 2y_0 s}{x_0^2} = \frac{2(1 - y_0 s)}{x_0^2} \tag{4.3}$$

Clearly, $gcd(R + St^2, N) = 1$. If $R + St^2 \equiv 0 \pmod N$, then by (4.3), $y_0 \equiv 1/s \pmod N$. But (x_0, y_0) is a solution of $Rx^2 + Sy^2 = 1 \pmod N$ and this implies $x_0 \equiv 0 \pmod N$, contradicting the fact that $gcd(x_0, N) = 1$. Thus we must have, $gcd(R + St^2, N) = 1$. Compute, $R - St^2 = R - S\frac{(y_0 - 1/s)^2}{x_0^2} = \frac{Rx_0^2 - Sy_0^2 - 1 + 2y_0 s}{x_0^2}$ $= \frac{2y_0 s(1 - y_0 s)}{x_0^2} = y_0 s(R + St^2)$, which gives $y_0 = \frac{R - St^2}{s(R + St^2)}$. Further,

$$-2st = -2s\frac{(y_0 - 1/s)}{x_0} = \frac{2(1 - y_0 s)}{x_0} = \frac{1}{x_0} 2(1 - y_0 s) \stackrel{\text{equation (4.3)}}{=} \frac{1}{x_0}(R + St^2)x_0^2$$

This yields,

$$x_0 = \frac{-2st}{R + St^2}$$

Thus (a) is proved. For (b), one may see that for any $t \in A_N$, the pair $(\frac{-2st}{R+St^2}, \frac{R-St^2}{s(R+St^2)})$ is a solution to (4.1). Moreover, if $\frac{-2st}{R+St^2} = \frac{-2st'}{R+S(t')^2}$ and $\frac{R-St^2}{s(R+St^2)} = \frac{R-S(t')^2}{s(R+S(t')^2)}$, then from the second identity we have $t^2 = (t')^2$. Thus, the first identity implies $t = t'$.

For (c), note that if $-R \in NQR(N)$ then for any $t \in \mathbb{Z}/N\mathbb{Z}^*, R + St^2 \not\equiv 0 \pmod N$ and hence $|A_N| = N - 1$. If $-R \in QR(N)$, then for $t = \pm\sqrt{-R/S}$, $R + St^2 \equiv 0 \pmod N$. Otherwise, $R + St^2 \not\equiv 0 \pmod N$. Hence in this case, the number of solutions is $N - 3$. □

Remark 4.2.1 *An analogous result can be proved when* R *is a quadratic residue modulo* N*.*

Corollary 4.2.1 *Let $N = p \cdot q$, where p, q are primes. Then any solution (x_0, y_0) with $gcd(x_0, N) = 1$ of (4.1) is of the form*

$$\left(\frac{-2st}{R+St^2}, \frac{R-St^2}{s(R+St^2)}\right),$$

for some $t \in \mathbb{Z}/N\mathbb{Z}$ such that $R + St^2 \in \mathbb{Z}/N\mathbb{Z}^*$.

Proof : Since (x_0, y_0) is a solution to both the equations,

$$Rx^2 + Sy^2 \equiv 1 \pmod{p}$$

$$Rx^2 + Sy^2 \equiv 1 \pmod{q}$$

we have by Lemma 4.2.1,

$$x_0 \equiv \frac{-2st_1}{R+St_1^2} \pmod{p}, \quad y_0 \equiv \frac{R-St_1^2}{s(R+St_1^2)} \pmod{p}$$

$$x_0 \equiv \frac{-2st_2}{R+St_2^2} \pmod{q}, \quad y_0 \equiv \frac{R-St_2^2}{s(R+St_2^2)} \pmod{q}$$

where $t_1 \in \mathbb{Z}/p\mathbb{Z}^*, t_2 \in \mathbb{Z}/q\mathbb{Z}^*$ and $\gcd(R + St_1^2, p) = 1, \gcd(R + St_2^2, q) = 1$. Let t be the unique integer in $\mathbb{Z}/N\mathbb{Z}^*$ such that $t \equiv t_1 \pmod{p}$ and $t \equiv t_2 \pmod{q}$. Clearly for this t, we have $gcd(R + St^2, N) = 1$. Thus, we have,

$$x_0 \equiv \frac{-2st}{R+St^2} \pmod{N}, \quad y_0 \equiv \frac{R-St^2}{s(R+St^2)} \pmod{N}$$

□

We now look at solutions (x, y) when x is not co-prime to N.

Lemma 4.2.2 *Suppose N is prime. The number of solutions (x_0, y_0) with $x_0 \equiv 0 \pmod{N}$ is*

$$= \begin{cases} 0 & \text{if } S \in NQR(N) \\ 2 & \text{if } S \in QR(N) \end{cases}$$

Proof : If $S \in NQR(N)$, then a solution of the form $(0, y_0)$ to (4.1) implies $S \equiv (\frac{1}{y_0})^2 \pmod{N}$, i.e., $S \in QR(N)$, a contradiction. Thus there is no such solution in this case. Now suppose, $S \in QR(N)$ and let s be a square root of S modulo N. Then $(0, \pm s^{-1})$ are the required solutions to (4.1). □

Thus we have proved the following theorem.

Theorem 4.2.1 *Let N be prime and S a square modulo N. Let η_N denote the number of solutions of equation (4.1). Then*

$$\eta_N = \begin{cases} N-1 & \text{if } -R \in QR(N) \\ N+1 & \text{if } -R \in NQR(N) \end{cases}$$

Theorem 4.2.2 *Let $N = p \cdot q$, where p and q are primes. Suppose S is a quadratic residue modulo N. Let η_N denote the number of solutions of equation (4.1). Then*

$$= \begin{cases} (p-1)(q-1) & \text{if } -R \in QR(N) \\ (p-1)(q+1) & \text{if } -R \in QR(p) \text{ and } -R \in NQR(q) \\ (p+1)(q-1) & \text{if } -R \in NQR(p) \text{ and } -R \in QR(q) \\ (p+1)(q+1) & \text{if } -R \in NQR(p) \text{ and } -R \in NQR(q) \end{cases}$$

Proof : The Chinese Remainder Theorem implies $\eta_N = \eta_p \times \eta_q$. Now η_p (respectively η_q) is $p-1$ or $p+1$ (respectively $q-1$ or $q+1$) according as $-R$ is square or non-square modulo p (respectively q). The result now follows immediately. □

The General Case:

We now consider the case when R or S may not be a quadratic residue. Again, we first assume that N is a prime and that a solution of (4.1) is known. Fix such a solution and denote it as (x^*, y^*).

Lemma 4.2.3 *Let $(x_0, y_0) \in \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ with $gcd(x_0 - x^*, N) = 1$ be any solution of (4.1). Then there exists a $t \in \mathbb{Z}/N\mathbb{Z}^*$ with $gcd(R + St^2, N) = 1$ such that*

$$x_0 = -\frac{(R - St^2)x^* + 2Sty^*}{R + St^2}; y_0 = \frac{-2Rtx^* + (R - St^2)y^*}{R + St^2} \tag{4.4}$$

Moreover, there is an one-one correspondence between all such solutions and the set A_N of Lemma (4.2.1).

Proof : Let (x_0, y_0) be a solution of equation (4.1), where $gcd(x_0 - x^*, N) = 1$. Let $t = \frac{y_0 - y^*}{x_0 - x^*}$. Then

$$R + St^2 = R + S\frac{(y_0 - y^*)^2}{(x_0 - x^*)^2} = \frac{2 - 2(Rx_0x^* + Sy_0y^*)}{(x_0 - x^*)^2} \tag{4.5}$$

If $R + St^2 \equiv 0 \pmod{N}$, then

$$Rx_0x^* + Sy_0y^* \equiv 1 \pmod{N} \tag{4.6}$$

Also, we have

$$Rx_0^2 + Sy_0^2 \equiv 1 \pmod{N} \tag{4.7}$$

$$R(x^*)^2 + S(y^*)^2 \equiv 1 \pmod{N} \tag{4.8}$$

By subtraction we obtain

$$Rx_0(x^* - x_0) + Sy_0(y^* - y_0) \equiv 0 \pmod{N}$$

$$Rx^*(x^* - x_0) + Sy^*(y^* - y_0) \equiv 0 \pmod{N}$$

Hence $\frac{x_0}{x^*} \equiv \frac{y_0}{x^*} \equiv k \pmod{N}$, say. So $x_0 \equiv kx^* \pmod{N}$ and $y_0 \equiv ky^* \pmod{N}$. Substitution in equation (4.6) yields $k \equiv 1 \pmod{N}$ which contradicts the fact that $gcd(x_0 - x^*, N) = 1$. Thus we must have $gcd(R + St^2, N) = 1$.

Now, we have

$$Sy_0y^* = Sy^*(y_0 - y^*) + S(y^*)^2 = Sty^*(x_0 - x^*) \tag{4.9}$$

Hence by (4.5) and (4.9) we obtain

$$\begin{aligned}(x_0 - x^*)^2 &= 2\frac{1 - (Rx_0x^* + Sy^*(y_0 - y^*) + S(y^*)^2)}{R + St^2} \\ &= 2\frac{1 - (Rx_0x^* + St(x_0 - x^*) + S(y^*)^2)}{R + St^2} \\ &= 2\frac{R(x^*)^2 - Rx_0x^* - St(x_0 - x^*)}{R + St^2}\end{aligned}$$

which yields,

$$x_0 - x^* = -2\frac{Rx^* + Sty^*}{R + St^2}$$

On simplification, we obtain,

$$x_0 = -\frac{(R - St^2)x^* + 2Sty^*}{R + St^2}$$

Also we have, $y_0 - y^* = t(x_0 - x^*)$. Putting the value of x_0 we get

$$y_0 = \frac{-2Rtx^* + (R - St^2)y^*}{R + St^2}$$

Denote the righthand-sides of equations (4.4) by x_t and y_t respectively. Suppose $x_t = x_{t'}$ and $y_t = y_{t'}$. Then from the last relation one obtains $y^* + t(x_t - x^*) = y^* + t'(x_{t'} - x^*)$ and hence $t = t'$. This establises the 1-1 correspondence. □

Like Lemma (4.2.2), the following is easy to prove.

Lemma 4.2.4 *Suppose N is prime. Then the number of solutions (x_0, y_0), where $x_0 \equiv x^* \pmod N$, is 2.*

Thus we have the following theorem.

Theorem 4.2.3 *Suppose $N = pq$, where p and q are primes. Let η_N denote the number of solutions of equation (4.1). Then*

$$\eta_N = \begin{cases} (p-1)(q-1) & \text{if } -R/S \in QR(N) \\ (p-1)(q+1) & \text{if } -R/S \in QR(p) \text{ and } -R/S \in NQR(q) \\ (p+1)(q-1) & \text{if } -R/S \in NQR(p) \text{ and } -R/S \in QR(q) \\ (p+1)(q+1) & \text{if } -R/S \in NQR(p) \text{ and } -R/S \in NQR(q) \end{cases}$$

Proof : We have $\eta_N = \eta_p \times \eta_q$. But by Lemmas 4.2.3 and 4.2.4,

$$\eta_p = |A_p| + 2 = \begin{cases} p-1 & \text{if } -R/S \in QR(p) \\ p+1 & \text{if } -R/S \in NQR(p) \end{cases}$$

□

Similar expression holds for η_q. The expressions now follows immediately.

4.2.1 Efficient Algorithm to Find a Random Solution of $Rx^2 + Sy^2 \equiv 1 \pmod N$

As an immediate implication of Corollary 4.2.1 we have the following simple algorithm to obtain a random solution of (4.1) when S is a quadratic residue and its square root is known. This algorithm plays a very important role in our constructions of Identity based encryption scheme and public key encryption scheme given in Section (4.3) and Section (4.4) respectively.

A Random Solution of $Rx^2 + Sy^2 \equiv 1 \bmod N$

Let N be a RSA modulus. Suppose $S \in QR(N)$ and s a square root of $S \bmod N$. Then a random solution to (4.1) can be obtained as follows.

Algorithm $\mathcal{R}$:
Consider $Rx^2 + Sy^2 = 1$ as a curve over $\mathbb{Z}/N\mathbb{Z}$.
Set $P = (0, 1/s)$ and choose $t \in_R \mathbb{Z}/N\mathbb{Z}$ such that $\gcd(R + St^2, N) = 1$.
Consider the line $L : y = tx + 1/s$ through P with gradient t
L intersects the curve at the point (x_0, y_0), where

$$x_0 = -\frac{2st}{R + St^2}, \quad y_0 = tx_0 + \frac{1}{s} \tag{4.10}$$

Output (x_0, y_0).

4.3 B. Identity-based Encryption Without Pairing

Though the notion of identity-based encryption (IBE) was introduced by Shamir [106] in 1984, it took more than two decades for the cryptography community to come up with a practical identity-based encryption scheme. Practical constructions of identity-based encryption first appeared at the beginning of the 21st century. Two similar schemes based on bilinear maps on elliptic curves (bilinear pairings, [14]) were independently proposed by Sakai et al. [92] and by Boneh and Franklin [21], followed soon afterwords by a completely different scheme (without using pairings) due to Cocks [35].

Things have changed rapidly in the years since 2000. Many improvements have been made to the Boneh-Franklin IBE scheme, and a few brand new approaches to IBE have been proposed [100, 19, 114, 56]. All of them, however, rely in one way or another upon the notion of pairings.

In this regard, Cocks' pairing free approach to IBE remains for the most part an isolated result with its share of limitations. Cocks' solution based on quadratic residuosity modulo an RSA number is not very efficient in terms of bandwidth. Briefly, the idea is the following.

- The public parameters of the system consists of $N = p \cdot q$; a random $u \in J(N) \setminus QR(N)$; and a hash function $H(\cdot)$ which maps identities into $J(N)$.
- The secret key corresponding to an identity id is obtained by first computing $R = H(id)$ and then r to be either $\sqrt{R}$ or $\sqrt{uR}$ according as R is quadratic

residue modulo N or not. The secret key for id is r.

- To encrypt a bit $m \in \{0,1\}$ using an identity id, compute R from id as above; randomly choose t_0, t_1 from $\mathbb{Z}/N\mathbb{Z}$ and compute $d_a = \frac{t_a^2 + u^a R}{t_a}$ and $c_a = m \cdot \left(\frac{t_a}{N}\right)$. The ciphertext consists of (d_0, d_1, c_0, c_1).

- For decryption using identity id and secret key r, first set $a \in \{0,1\}$ such that $r^2 = u^a R$, where R is obtained from id as above. Set $g = d_a + 2r$. Note that $g = \frac{(t_a + r)^2}{t_a}$ and hence , $\left(\frac{g}{N}\right) = \left(\frac{t_a}{N}\right)$. So the receiver can compute $m = c_a \cdot \left(\frac{g}{N}\right)$.

While this is an interesting system, sending a single encrypted bit is achieved by transmitting a value modulo the RSA number, which essentially has the same size as the RSA number itself. In particular, for an ℓ-bit message it outputs a ciphertext consisting of 2ℓ elements of $\mathbb{Z}/N\mathbb{Z}$ with 2ℓ additional bits, thus making it space-inefficient (high bandwidth). Since [35], an important problem was to construct an space-efficient IBE (a system with short ciphertexts) that does not uses pairings.

Recently Boneh-Gentry-Hamburg (BGH) proposed a very elegant IBE system, BasicIBE [24]. The scheme BasicIBE has solved this long standing open problem. The bandwidth problem was significantly improved in BasicIBE, in particular, the ciphertext overhead was reduced from 2ℓ elements of $\mathbb{Z}/N\mathbb{Z}$ to merely one. Though BasicIBE is highly space efficient, the concrete instantiations of BasicIBE is obtained at the cost of much less efficient encryption and decryption algorithms. To encrypt an ℓ-bit message with BasicIBE,

- the encryptor has to solve $\ell + 1$ many equations of the form (4.1),

- the encryptor and decryptor has to agree on the **same** solutions to each of these $\ell + 1$ equations. This is a major requirement (Our earlier discussions implies equations of type (4.1) has "many" solutions. We will see that it is difficult to construct an algorithm that finds unique solution).

As discussed at the beginning of this chapter, to solve this problem, BGH lift (4.1) to the integers and consider the ternary quadratic form (4.2). Factorization free algorithm of [40] is used to solve these equations. The algorithm of [40] requires what is called a *solubility certificate*. BGH [24] provides an algorithm that produce these certificates. This algorithm requires **generation of primes** that are

of size similar to the size of N. The generation of primes is the main bottleneck of the BasicIBE. Various ways of reducing prime generations are discussed in [24]. In [24] Section 5.3, a time-space tradeoff method for reducing prime generations by a factor of $\sqrt{\ell}$ has been proposed. This is obtained at the cost of increasing the ciphertext size from a single element to $\lceil\sqrt{\ell}\rceil$ many $\mathbb{Z}/N\mathbb{Z}$ elements. For completeness an insight to the Boneh-Gentry-Hamburg's method to solve equations of the form (4.1) is given at the end of the Section 4.3.3.

Based on BasicIBE, Boneh-Gentry-Hamburg obtained an *anonymous* IBE system (Sec 6, [24]) under interactive quadratic residuocity [24] assumption in the standard model. In this scheme, for an ℓ-bit message the ciphertext consists of only a single $\mathbb{Z}/N\mathbb{Z}$ element and $\ell + 1$ additional bits.

Our Contribution

In this chapter, we present an IBE system by suitably modifying the BasicIBE [24]. Our system achieves the following

- The encryptor needs to solve $2\lceil\sqrt{\ell}\rceil$ equations of the form $Rx^2 + Sy^2 \equiv 1 \bmod N$. We use the algorithm $\mathcal{R}$ of Section 4.2.1 to solve these equations. The algorithm $\mathcal{R}$ takes a square root of $S \bmod N$ as input and finds a random solution to $RX^2 + SY^2 = 1 \bmod N$ using only *one inversion* in $\mathbb{Z}/N\mathbb{Z}^*$. Thus we do not use the algorithm of BGH to solve these equations. Hence encryption algorithm completely avoids the primes generation cost.

- The decryptor need not solve any equation of the above type. This has two consequences. Firstly, this increases the efficiency of the decryption algorithm as no prime generation will be needed. Secondly, as we discuss below, in BasicIBE both encryptor and decryptor has to follow *exactly the same algorithm* for obtaining solutions of the above equations. Otherwise, error in decryption may take place (see Section 4.3.1 below). However, in our scheme, the decryptor need not solve any equation. So the encryptor is free to use any algorithm to solve these equations.

- The private key consists of only a single element of $\mathbb{Z}/N\mathbb{Z}$, whereas in BasicIBE, it consists of ℓ such elements.

- The ciphertext length increases from a single element to $2\lceil\sqrt{\ell}\rceil$ elements of $\mathbb{Z}/N\mathbb{Z}$. This is much worse that the BGH BasicIBE as far as space efficiency

is concerned. However, the space-time tradeoff of BGH (discussed briefly above) increases the ciphertext length from one element to $\sqrt{\ell}$ elements of $\mathbb{Z}/N\mathbb{Z}$. Our scheme performs much better as we have completely avoided generation of primes while solving equations of type (4.1).

4.3.1 Boneh-Gentry-Hamburg Identity-based Encryption Scheme

We recall the concrete BasicIBE scheme of [24] and discuss some important issues about BasicIBE scheme. A key step of this system is a deterministic algorithm $\mathcal{Q}$ with the following properties.

Definition 4.3.1 *The deterministic algorithm $\mathcal{Q}$ takes as input (N, R, S) where N is an RSA modulus and $R, S \in \mathbb{Z}/N\mathbb{Z}^*$. The algorithm outputs two polynomials $f, g \in \mathbb{Z}/N\mathbb{Z}[X]$ satisfying the following two properties:*

Prop 1 If R and S are quadratic residues modulo N, then $f(r)g(s)$ is a quadratic residue modulo N for all square roots r of R and s of S modulo N and hence the following Jacobi symbols are equal

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

Prop 2 If R is a quadratic residue modulo N, then $f(r)f(-r)S$ is a quadratic residue modulo N for all square roots r of R modulo N.

Concrete Instantiation of $\mathcal{Q}$ for BasicIBE

The concrete instantiation of algorithm $\mathcal{Q}(N, R, S)$ takes input R, S from $\mathbb{Z}/N\mathbb{Z}$. It works as follows:

- Solve (by running the deterministic algorithm of BGH cf.[24])

$$Rx^2 + Sy^2 \equiv 1 \bmod N \quad \text{for} \quad x \quad \text{and} \quad y$$

- Output polynomials f, g in variables r, s respectively with a fixed solution (x, y) of (4.1) as follows

$$f(r) = xr + 1 \quad \text{and} \quad g(s) = 2ys + 2 \tag{4.11}$$

As observed in [24], the polynomials satisfy **Prop-1** and **Prop-2**. Let r and s be square roots of R and S modulo N respectively. Then

- **Prop 1**

$$f(r) \cdot g(s) = 2xrys + 2xr + 2ys + 2 + (Rx^2 + Sy^2 - 1) = (xr + ys + 1)^2 \bmod N \tag{4.12}$$

- **Prop 2** $f(r) \cdot f(-r) \cdot S = (1 - Rx^2) \cdot S = (Sy)^2 \bmod N$

The decryption algorithm of BasicIBE uses **Prop 1** and **Prop 2** is needed for the security analysis of BasicIBE. Like BasicIBE, in our proposed IBE **Prop 1** is required for decryption. But unlike BasicIBE, **Prop 2** is not required to prove the security of our proposed IBE. Instead we observe that polynomials f and g satisfy a similar property which will be useful for the security proof of our scheme. We state it below:

Prop 3 If S is a quadratic residue modulo N, then $g(s)g(-s)R$ is a quadratic residue for all square roots s of S modulo N. This holds, since

$$g(s) \cdot g(-s) \cdot R = (2ys + 2) \cdot (-2ys + 2) \cdot R = 4 \cdot (1 - Sy^2) \cdot R = (2Rx)^2 \bmod N$$

Next we describe BGH's BasicIBE scheme for encrypting ℓ-bit messages.

BasicIBE

1. **Setup(λ):** Generate $(p, q) \leftarrow \text{RSAgen}(\lambda)$ and compute $N = p \cdot q$. Choose $u \in_R J(N) \setminus QR(N)$. Public parameters $PP = (N, u, H)$ where H is a hash function $H : \mathcal{ID} \times \{1, 2, \ldots, \ell\} \rightarrow J(N)$ and $\mathcal{ID} = \{0,1\}^*$.
   The master key msk is the factorization of N and a random key K for a pseudorandom function $F_K : \mathcal{ID} \times \{1, 2, \ldots, \ell\} \rightarrow \{0, 1, 2, 3\}$.

2. **Extract**(msk, id, ℓ): Takes as input msk, id and a message length parameter ℓ. It generates private key for decrypting encryptions of ℓ-bit messages. For $j = 1, \ldots, \ell$ do:
   $H(id, j) = R_j \in J(N)$ and $F_K(id, j) = w \in \{0, 1, 2, 3\}$
   let $a \in \{0, 1\}$ be such that $u^a R_j \in QR(N)$
   let $\{z_0, z_1, z_2, z_3\}$ be the four square roots of $u^a R_j$ in $\mathbb{Z}/N\mathbb{Z}$
   Set $r_j = z_w$.
   Output the private key $d_{id} = (PP, r_1, \ldots, r_\ell)$. The PRF F_K ensures that the key generation always outputs the same square roots for a given id.

3. **Encrypt**(PP, id, m): Takes as input PP, a user id and a message $m = m_1, \ldots, m_\ell \in \{-1, 1\}^\ell$. It generates a random $s \in \mathbb{Z}/N\mathbb{Z}$ and sets $S \equiv s^2 \pmod N$. For $j = 1, \ldots, \ell$ do:

   - $R_j = H(id, j)$, $(f_j, g_j) \leftarrow Q(N, R_j, S)$, and $(\tilde{f}_j, \tilde{g}_j) \leftarrow \mathcal{Q}(N, uR_j, S)$
   - $c_j = m_j \cdot (\frac{g_j(s)}{N})$ and $\tilde{c}_j = m_j \cdot (\frac{\tilde{g}_j(s)}{N})$.

   Set $c = c_1, \ldots, c_\ell$ and $\tilde{c} = \tilde{c}_1, \ldots, \tilde{c}_\ell$ and output the ciphertext $C = (S, c, \tilde{c})$.

4. **Decrypt**(C, d_{id}): Let $d_{id} = (PP, r_1, \ldots, r_\ell)$. For $j = 1, \ldots, \ell$, let $R_j = H(id, j)$ and do:

   - if $r_j^2 = R_j$ run $(f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S)$ and set $m_j = c_j \cdot (\frac{f_j(r_j)}{N})$
   - if $r_j^2 = uR_j$ run $(\tilde{f}_j, \tilde{g}_j) \leftarrow \mathcal{Q}(N, uR_j, S)$ and set $m_j = \tilde{c}_j \cdot (\frac{\tilde{f}_j(r_j)}{N})$

   Output $m = m_1, \ldots, m_\ell$

Soundness of decryption follows from **Prop 1**.

Optimizations in [24] for BasicIBE

In BasicIBE, to encrypt an ℓ-bit message, one has to solve 2ℓ many equations of type (4.1). In particular, for $i = 1, \ldots, \ell$, one needs pairs (x_i, y_i), $(\tilde{x}_i, \tilde{y}_i) \in \mathbb{Z}/N\mathbb{Z}$ such that

$$R_i \cdot x_i^2 + S \cdot y_i^2 \equiv 1 \bmod N \quad \text{and} \quad (uR_i) \cdot \tilde{x}_i^2 + S \cdot \tilde{y}_i^2 \equiv 1 \bmod N \tag{4.13}$$

The decryptor needs the same solutions to these equations. By using the following product formula (cf. [24]), it is easy to see that the encryptor needs to solve only $\ell + 1$ equations.

Lemma 4.3.1 *[24, Lemma 5.1] Let (x_i, y_i) be solutions to $R_i x^2 + S y^2 \equiv 1 \pmod N$ for $i = 1, 2$. Then (x_3, y_3) is a solution to*

$$(R_1 R_2) \cdot x^2 + S \cdot y^2 \equiv 1 \pmod N \tag{4.14}$$

where $x_3 = \left(\frac{x_1 x_2}{S y_1 y_2 + 1}\right)$ and $y_3 = \left(\frac{y_1 + y_2}{S y_1 y_2 + 1}\right)$

Thus during encryption, one first finds solutions to the $\ell + 1$ equations

$$ux^2 + Sy^2 \equiv 1 \bmod N \ \text{ and } \ R_i x^2 + S y_i^2 \equiv 1 \bmod N \ \text{ for } i = 1, \ldots, \ell \tag{4.15}$$

Then apply Lemma 4.3.1 to obtain a solution to $(uR_i)x^2 + Sy^2 \equiv 1 \bmod N$. If required, decryptor does the same.

Security

The security of the BasicIBE in the random oracle model is based on the quadratic residuosity assumption.

Theorem 4.3.1 *[24] Suppose the Quadratic Residue assumption holds for RSAgen and F is a secure PRF. Then the IBE 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*

$$\text{IBEAdv}^{\text{IND-ID-CPA}}_{\text{BasicIBE},\mathcal{A}}(\lambda) \leq 2 \cdot \text{QRAdv}_{\text{RSAgen},\mathcal{B}_1}(\lambda) + \text{PRFAdv}_{F,\mathcal{B}_2}(\lambda)$$

Some observations

Observation 1: Note that both the encryptor and the decryptor have to solve equations of the form (4.1). The encryptor computes $g(s)$, while the decryptor computes $f(r)$, where r and s are square roots of R and S mod$N$ respectively, and (x, y) a solution of (4.1). The soundness follows from the fact that $\left(\frac{f(r)}{N}\right) = \left(\frac{g(s)}{N}\right)$. This, however, is true provided both $f(r)$ and $g(s)$ are coprime to N. It is possible to end up with polynomials f and g such that $\left(\frac{f(r)}{N}\right) = 0$ but $\left(\frac{g(s)}{N}\right) = \pm 1$. Consider the following example:

Example 4.3.1 *Take p=23, q=19 and N=$p \times q$=437. Now consider the following equation $Rx^2 + Sy^2 \equiv 1 \bmod N$ where R=348, S=36 and r=214, s=63 are the respective square roots modulo N. One can see that $(x = 15, y = 76)$ is a solution to the above equation. For this solution one can verify that $\left(\frac{g(s)}{N}\right) = 1$ whereas $\left(\frac{f(r)}{N}\right) = 0$.*

One may note that the probability of such an event is very small.

However, one can easily prevent such an event from occuring. Observe that $x \equiv \pm r^{-1} \bmod N$, $y \equiv 0 \mod N$ and $x \equiv 0 \bmod N$, $y \equiv \pm s^{-1} \bmod N$ are solutions to equation (4.1). To avoid such an event, *just ensure that the algorithm Q does not output such (trivial) solutions.* This will ensure that $(\frac{f(r)}{N}) \neq 0 \neq (\frac{g(s)}{N})$. To see this, suppose $(\frac{f(r)}{N}) = 0$. Then $f(r) = 0 \bmod p$ or $f(r) = 0 \bmod q$. W.l.o.g. we may assume that $f(r) = 0 \bmod N$ (otherwise, we obtain a factorization of N). Then by equation (4.12), $f(r)g(s) = (rx +$

$sy+1)^2 = s^2y^2 = Sy^2 = 0 \bmod N$ and so $y = 0 \bmod N$, since $(S, N) = 1$. This will contradict that (x, y) is a non-trivial solution. Similar arguments hold for $g(s)$.

Observation 2: As we have mentioned, the main bottleneck is to solve equations of the form $Rx^2 + Sy^2 \equiv 1 \bmod N$. We shall argue that great care is needed to solve such equations. For solving these equations, Boneh-Gentry-Hamburg considered the ternary quadratic form of the type (4.2), where $\tilde{R}, \tilde{S}, x, y, z \in \mathbb{Z}$ and $\tilde{R} \equiv R \bmod N$, $\tilde{S} \equiv S \bmod N$. Clearly a solution to (4.2) in $\mathbb{Z}$ gives a solution to (4.1) in $\mathbb{Z}/N\mathbb{Z}$. BGH uses lattice basis reduction algorithm of Cremona-Rusin [40] to solve (4.2). But this algorithm needs the following square roots:

$$r^2 \equiv \tilde{R} \bmod \tilde{S} \text{ and } s^2 \equiv \tilde{S} \bmod \tilde{R} \tag{4.16}$$

The congruences (4.16) can be solved efficiently when $\tilde{R}$ and $\tilde{S}$ are primes. So one has to look for (probable) primes $\tilde{R}$ and $\tilde{S}$ in the arithmetic progressions $\tilde{R} \equiv R \bmod N$, $\tilde{S} \equiv S \bmod N$, a major reason for the slow encryption. See [24] for details of the algorithm.

We shall now show that both the encryptor and the decryptor must follow the **same** steps while executing the algorithm. In fact, they should come up with same $\tilde{R}$ and $\tilde{S}$ while generating the primes so that on invoking the Cremona-Rusin Algorithm, one obtains the same solution. If both encryptor and decryptor do not follow the same algorithm for solving equations of the form (1), then they may obtain different solutions of (4.1). This, possibly, may lead to an error (see below). Thus as part of the implementation of BasicIBE, the use of the same algorithm for solving (4.1) by both the encryptor and decryptor is mandatory. This is because of the following simple observation.

> Suppose (x, y) and $(\tilde{x}, \tilde{y})$ are different solutions to (1). Let f, g and $\tilde{f}, \tilde{g}$ be the corresponding polynomials. Then, it is **not** always the case that
>
> $$\left(\frac{f(r)}{N}\right) = \left(\frac{\tilde{f}(r)}{N}\right) \tag{4.17}$$
>
> where r, s are square roots of R, S modulo N respectively.
> However, $\left(\frac{f(r)}{N}\right) = \left(\frac{g(s)}{N}\right)$ and $\left(\frac{\tilde{f}(r)}{N}\right) = \left(\frac{\tilde{g}(s)}{N}\right)$ may still be valid.

Again consider Example 4.3.1. Here, for the following two different solutions $(x_1 = 8, y_1 = 145)$ and $(x_2 = 10, y_2 = 276)$ the corresponding $f(r)$ and $\tilde{f}(r)$ respectively have the following Jacobi symbols:

$$\left(\frac{f(r)}{N}\right) = 1 \quad \text{and} \quad \left(\frac{\tilde{f}(r)}{N}\right) = -1$$

Thus, for this example, the decryptor will erroneously decrypt the encrypted bit.

Table 4.1 gives the number of solutions of equation (4.1) and the Jacobi symbols for several values of the parameters involved. It shows that in equal number of cases, the Jacobi symbols are likely to be different.

Table 4.1: Examples: Number of Solutions to $Rx^2 + Sy^2 \equiv 1 \pmod{N}$

p	q	N	r	R	s	S	K	K_1	K_2
23	19	437	214	348	63	36	480	198	198
43	59	2537	2461	702	1437	2388	2640	1218	1218
67	73	4891	3916	1771	1872	2428	4896	2310	2310
151	179	27029	13009	5512	10440	12672	27360	13350	13350

Here K denotes the number of solutions (x, y) to the equation (4.1). Among K solutions, K_1 deontes the number of solutions for which the corresponding f satisfy $\left(\frac{f(r)}{N}\right) = 1$ and K_2 denotes the number of solutions for which f satisfy $\left(\frac{f(r)}{N}\right) = -1$.

Thus both the encryptor and decryptor should obtain the same polynomials as output from the algorithm $\mathcal{Q}(N, R, S)$. As a consequence, the algorithm has to be precisely documented as part of the public parameters so that each time it is invoked it gives the same output. It is worth mentioning that the BGH scheme is sound since both encryptor and decryptor use the same deterministc algorithm Q. It is evident from Section 4.2 that the equation $Rx^2 + Sy^2 \equiv 1 \bmod N$ has many solutions. Thus a scheme which does not require such constraints is likely to be more efficient. We shall exploit this in our scheme.

Observation 3: For the ith bit, the decryptor, using his private key $d_{id} = (r_1, \ldots, r_\ell)$, needs to solve $uR_ix^2 + Sy^2 \equiv 1 \bmod N$ *in the case* $r_i^2 \equiv uR_i \bmod$

N. Since the encryptor solves this equation by solving both $ux^2 + Sy^2 \equiv 1 \bmod N$ and $R_i x^2 + Sy^2 \equiv 1 \bmod N$ and then uses the product formula (Lemma 4.3.1), the decryptor also has to solve these equations. Solving $uR_i x^2 + Sy^2 \equiv 1 \bmod N$ directly, the decryptor may end up with a different solution and by **Observation 2** this might create a problem. Thus the decryptor needs to solve an additional equation $ux^2 + Sy^2 \equiv 1 \bmod N$ (except in the very unlikely case that all R_i are squares). In our proposed scheme we avoid such restrictions since the decryptor need not even solve any such equations.

4.3.2 The Modified Boneh-Gentry-Hamburg's IBE Scheme

In this section we present our scheme. We call it M-BasicIBE.

Generalized Product Formula

We shall first need the following product formula, which is analogue to Lemma 4.3.1.

Lemma 4.3.2 *Let (x_i, y_i) be solutions to the equations $Rx^2 + U_i y^2 = 1$ for $i = 1, 2$. Then a solution to the equation $Rx^2 + U_1 U_2 y^2 = 1$ is given by*

$$\left(\frac{x_1 + x_2}{Rx_1 x_2 + 1}, \frac{y_1 y_2}{Rx_1 x_2 + 1} \right)$$

Remark 4.3.1 *Observe that the x part of the solution to the third equation depends only on x_1, x_2 and R. We exploit this in our scheme to reduce the ciphertext length.*

We now present our scheme.

M-BasicIBE

- **Setup** (λ): Generate two primes p and q and compute $N = p \cdot q$. Choose $u \in_R J(N) \setminus QR(N)$.
  - Public parameters $PP = (N, u, H)$ where H is a hash function $H : \mathcal{ID} \to J(N)$.

  - The master key msk is the factorization of N and a random key K for a pseudorandom function $F_K : \mathcal{ID} \to \{0, 1, 2, 3\}$.

- **KeyExtraction** (msk, id, ℓ): Takes as input msk, id and a message length parameter ℓ. It generates a private key for decrypting encryptions of ℓ-bit messages as follows:
  - Set $R = H(id) \in J(N)$ and $w = F_K(id) \in \{0, 1, 2, 3\}$.
  - Selects $a \in \{0, 1\}$ such that $u^a R \in QR(N)$.
  - Let $\{z_0, z_1, z_2, z_3\}$ be the four square roots of $u^a R$ in $\mathbb{Z}/N\mathbb{Z}$.
  - Set $r = z_w$ and output the private key $d_{id} = r$.

- **Encryption** (PP, id, m): To encrypt an ℓ-bit message $m = m_0, \ldots, m_{\ell-1} \in \{-1, 1\}^\ell$ using id do the following. Write $\kappa = \lceil \sqrt{\ell} \rceil$.
  - Compute $H(id) = R$. Choose $u_0, u_1, \ldots, u_{\kappa-1} \in_R \mathbb{Z}/N\mathbb{Z}$ and compute $u_i^2 \equiv U_i \mod N$ for $0 \leq i \leq \kappa - 1$.
  - Solve, using algorithm $\mathcal{R}$, the following set of equations for $0 \leq i \leq \kappa - 1$.

$$Rx^2 + U_i y^2 \equiv 1 \bmod N; \quad uRx^2 + U_i y^2 \equiv 1 \bmod N \tag{4.18}$$

  - Let (x_i, y_i) and $(\tilde{x}_i, \tilde{y}_i)$ be respectively the solutions. Now to encrypt the jth bit m_j one does the following:
    If $j \leq \kappa - 1$, define $g_j(u_j) = 2y_j u_j + 2$ and $\tilde{g}_j(u_j) = 2\tilde{y}_j u_j + 2$ and compute

$$c_j = m_j \cdot \left(\tfrac{g_j(u_j)}{N}\right); \quad \tilde{c}_j = m_j \cdot \left(\frac{\tilde{g}_j(u_j)}{N}\right) \tag{4.19}$$

    If $j > \kappa - 1$, find the unique integers j_1, j_2 such that

$$j = \kappa \cdot j_1 + j_2, 0 \leq j_1, j_2 \leq \kappa - 1$$

  - Set $y_{j_1 j_2} = \frac{y_{j_1} y_{j_2}}{R x_{j_1} x_{j_2} + 1}$ and $\tilde{y}_{j_1 j_2} = \frac{\tilde{y}_{j_1} \tilde{y}_{j_2}}{uR\tilde{x}_{j_1}\tilde{x}_{j_2} + 1}$. By (Lemma 4.3.2), $y_{j_1 j_2}, \tilde{y}_{j_1 j_2}$ are the y-coordinates of the solutions to the equations $Rx^2 + U_{j_1} U_{j_2} y^2 \equiv 1 \pmod{N}$ and $uRx^2 + U_{j_1} U_{j_2} y^2 \equiv 1 \pmod{N}$ respectively.
  - Define $g_{j_1 j_2}(u_{j_1} u_{j_2}) = 2y_{j_1 j_2} u_{j_1} u_{j_2} + 2$ and $\tilde{g}_{j_1 j_2}(u_{j_1} u_{j_2}) = 2\tilde{y}_{j_1 j_2} u_{j_1} u_{j_2} + 2$.

– Compute,

$$c_j = m_j \cdot \left(\frac{g_{j_1 j_2}(u_{j_1} u_{j_2})}{N}\right); \quad \tilde{c}_j = m_j \cdot \left(\frac{\tilde{g}_{j_1 j_2}(u_{j_1} u_{j_2})}{N}\right) \tag{4.20}$$

– Set $\mathbf{c} =< c_1, \ldots, c_\ell >, \tilde{\mathbf{c}} =< \tilde{c}_1, \ldots, \tilde{c}_\ell >, \mathbf{x} = (x_1, \ldots, x_\kappa), \tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_\kappa)$.

– The cipherext is $C = (\mathbf{c}, \tilde{\mathbf{c}}, \mathbf{x}, \tilde{\mathbf{x}})$.

- **Decryption** (C, d_{id}): Decrypt the cipherext $C = (\mathbf{c}, \tilde{\mathbf{c}}, \mathbf{x}, \tilde{\mathbf{x}})$ with the private key $d_{id} = r$ as follows.

  – Compute $H(id) = R$.

  – If $r^2 \equiv R \bmod N$, then for $j \leq \kappa - 1$, set $f_j(r) = x_j r + 1$ to decrypt the jth bit of the ciphertext as follows,

  $$m_j = c_j \cdot \left(\frac{f_j(r)}{N}\right)$$

  If $j > \kappa - 1$, write j as $j = \kappa \cdot j_1 + j_2$, $0 \leq j_1, j_2 \leq \kappa - 1$. Set $x_{j_1 j_2} = \frac{x_{j_1} + x_{j_2}}{R x_{j_1} x_{j_2} + 1}$. By (Lemma-4.3.2), $x_{j_1 j_2}$ is the x-coordinate of the solution to the equation $Rx^2 + U_{j_1} U_{j_2} y^2 \equiv 1 \pmod N$. Define $f_{j_1 j_2}(r) = x_{j_1 j_2} r + 1$. Compute,

  $$m_j = c_j \cdot \left(\frac{f_{j_1 j_2}(r)}{N}\right)$$

  – For the case when $r^2 \equiv uR \pmod N$, decrypt the jth bit as follows: If $j \leq \kappa - 1$, define $\tilde{f}_j(r) = \tilde{x}_j r + 1$ and compute

  $$m_j = \tilde{c}_j \cdot \left(\frac{\tilde{f}_j(r)}{N}\right)$$

  If $j > \kappa - 1$, write j as $j = \kappa \cdot j_1 + j_2$, $0 \leq j_1, j_2 \leq \kappa - 1$. Set $\tilde{x}_{j_1 j_2} = \frac{\tilde{x}_{j_1} + \tilde{x}_{j_2}}{uR \tilde{x}_{j_1} \tilde{x}_{j_2} + 1}$. By (4.3.2), $\tilde{x}_{j_1 j_2}$ is the x-coordinate of the solution to the equation $uRx^2 + U_{j_1} U_{j_2} y^2 \equiv 1 \pmod N$. Define $\tilde{f}_{j_1 j_2}(r) = \tilde{x}_{j_1 j_2} r + 1$. Compute,

  $$m_j = \tilde{c}_j \cdot \left(\frac{\tilde{f}_{j_1 j_2}(r)}{N}\right)$$

Remark 4.3.2

- An abstract formulation of the above scheme can be made as in [24] with some additional assumption.

- The encryptor has to solve $2\lceil\sqrt{\ell}\rceil$ equations of the form (4.1) while the decryptor solves none. Hence, the encryptor can use any efficient algorithm to solve those equations. Using algorithm $\mathcal{R}$, the encryptor solves these equations using $2\lceil\sqrt{\ell}\rceil$ inversions in $\mathbb{Z}/N\mathbb{Z}^*$. Thus *no generations of primes is needed for our IBE system.*

4.3.3 Security

Since there is a generic method of converting an IND-ID-CPA secure IBE scheme into a chosen ciphertext secure IBE scheme in the random oracle model [13], we shall only consider IND-ID-CPA security. We first state a lemma which is essential in proving the security of M-BasicIBE.

Lemma 4.3.3 *[24] Let $N = p \cdot q$ be an RSA modulus, $S \in QR(N)$, and $R \in J(N)$. Let s be a random variable uniformly chosen among the four square roots of S modulo N. Let g be a polynomial such that $g(s)g(-s)R$ is a quadratic residue modulo N for all four values of s. Then*

- *when $R \in J(N)\backslash QR(N)$ the Jacobi symbol $\left(\frac{g(s)}{N}\right)$ is uniformly distributed in $\{-1, +1\}$;*

- *when $R \in QR(N)$ then the Jacobi symbol $\left(\frac{g(s)}{N}\right)$ is constant, i.e. the same for all four values of s.*

Theorem 4.3.2 *Suppose the Quadratic Residue assumption holds for RSAgen and F is a secure PRF. Then the IBE system M-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*

$$\mathrm{Adv}^{\text{IND-ID-CPA}}_{\text{M-BasicIBE},\mathcal{A}}(\lambda) \leq 2\mathrm{QRAdv}_{\mathrm{RSAgen},\mathcal{B}_1}(\lambda) + \mathrm{PRFAdv}_{\mathrm{F},\mathcal{B}_2}(\lambda) + \frac{1}{2^{\kappa}}$$

where $\kappa = \lceil\sqrt{\ell}\rceil$.

Proof : We define a sequence of games and let W_i denote the event that adversary $\mathcal{A}$ wins the ith game.

- Game 0: This is the usual adversarial game for defining IND-ID-CPA security of IBE protocols. The challenger picks the random oracle $H : \mathcal{ID} \rightarrow J(N)$ at random from the set of all such functions in the Setup algorithm and allows $\mathcal{A}$ to query H at arbitrary points. Thus, we have

$$|\Pr[\mathrm{W}_0] - \frac{1}{2}| = \mathrm{Adv}^{\text{IND-ID-CPA}}_{\text{M-BasicIBE},\mathcal{A}}(\lambda) \tag{4.21}$$

- Game 1: This is the same as Game 0, with the following change. In Setup algorithm, instead of using a PRF F to respond to $\mathcal{A}$'s private key queries we use a truly random function $f : \mathcal{ID} \rightarrow \{0, 1, 2, 3\}$. If F is a secure PRF, $\mathcal{A}$ will not notice the difference between Game 0 and Game 1. In particular, there exists an algorithm $\mathcal{B}_1$ (whose running time is about the same as that of $\mathcal{A}$) such that,

$$|\Pr[\mathrm{W}_1] - \Pr[\mathrm{W}_0]| = \mathrm{PRFAdv}_{\mathrm{F},\mathcal{B}_1}(\lambda) \tag{4.22}$$

- Game 2: (N, u, H) are the public parameters PP given to $\mathcal{A}$ in the previous game where u is uniform in $J(N)\backslash QR(N)$ and the random oracle H is a random function $H : \mathcal{ID} \rightarrow J(N)$. We make the following change in the random oracle H in this game. The challenger responds to a query to $H(ID)$ by picking picking $a \in_R \{0, 1\}$ and $v \in_R \mathbb{Z}/N\mathbb{Z}$ and setting $H(ID)$=$u^a \cdot v^2$. Thus the challenger implements a random function $H : \mathcal{ID} \rightarrow J(N)$ as in the previous game. The challenger responds to a private key query as follows.

  Suppose $R = H(ID) = u^a \cdot v^2$ for some $a \in \{0, 1\}$ and $v \in_R \mathbb{Z}/N\mathbb{Z}$. The challenger responds to a private key query for ID by setting either $R^{1/2} = v$ (when $a = 0$) or $(uR)^{1/2} = uv$ (when $a = 1$). Since v is uniform in $\mathbb{Z}/N\mathbb{Z}$ this will produce a square root of R or uR which is also uniform among the four square roots, as in the previous game. Thus, $\mathcal{A}$'s views in Game 1 and Game 2 are identical and therefore,

$$\Pr[\mathrm{W}_2] = \Pr[\mathrm{W}_1] \tag{4.23}$$

- Game 3: We make the following change to obtain Game 3. Note that the public parameters PP given to $\mathcal{A}$ consist of (N, u, H) where u is uniformly distributed in $J(N)\backslash QR(N)$. In this game, in Setup algorithm, the challenger chooses u, uniformly from $QR(N)$ instead of $J(N)\backslash QR(N)$. Since

this is the only change between Game 2 and Game 3, adversary $\mathcal{A}$ will not notice the difference assuming that the QR assumption holds for RSAgen. In particular, there exists an algorithm $\mathcal{B}_2$ (whose running time is about the same as that of $\mathcal{A}$) such that,

$$|\Pr[\mathrm{W}_3] - \Pr[\mathrm{W}_2]| = \mathrm{QRAdv}_{\mathrm{RSAgen},\mathcal{B}_2}(\lambda) \tag{4.24}$$

- Game 4: In this game we make a change in the challenge phase. We describe below in detail how the challenger responds to the encryption query $(ID, m^{(0)}, m^{(1)})$ from $\mathcal{A}$ in the challenge phase.
  - Choose $b \in_R \{0,1\}$ and write $m^{(b)} = m_0, \ldots, m_{\ell-1} \in \{-1,1\}^\ell$.
  - Choose $R \in_R J(N) \backslash QR(N)$ and set $H(ID) = R$. (*)
  - Write $\kappa = \lceil \sqrt{\ell} \rceil$. Choose $u_0, u_1, \ldots, u_\kappa \in_R \mathbb{Z}/N\mathbb{Z}$ and compute $u_i^2 \equiv U_i \mod N$ for $0 \leq i \leq \kappa - 1$.
  - Solve, using algorithm $\mathcal{R}$, the following set of equations for $0 \leq i \leq \kappa - 1$.

$$Rx^2 + U_i y^2 \equiv 1 \bmod N; \quad uRx^2 + U_i y^2 \equiv 1 \bmod N \tag{4.25}$$

  - Let (x_i, y_i) and $(\tilde{x}_i, \tilde{y}_i)$ be respectively the solutions. Now to encrypt the jth bit m_j one does the following:
    If $j \leq \kappa - 1$, compute

$$c_j = m_j \cdot \left(\frac{2y_j u_j + 2}{N}\right); \tilde{c}_j = m_j \cdot \left(\frac{2\tilde{y}_j u_j + 2}{N}\right) \ldots\ldots (\dagger)$$

    If $j > \kappa - 1$, find the unique integers j_1, j_2 such that

$$j = \kappa \cdot j_1 + j_2, 0 \leq j_1, j_2 \leq \kappa - 1$$

  - Set $y_{j_1 j_2} = \frac{y_{j_1} y_{j_2}}{R x_{j_1} x_{j_2} + 1}$ and $\tilde{y}_{j_1 j_2} = \frac{\tilde{y}_{j_1} \tilde{y}_{j_2}}{uR \tilde{x}_{j_1} \tilde{x}_{j_2} + 1}$.
  - Compute,

$$c_j = m_j \cdot \left(\frac{2y_{j_1 j_2} u_{j_1} u_{j_2} + 2}{N}\right); \tilde{c}_j = m_j \cdot \left(\frac{2\tilde{y}_{j_1 j_2} u_{j_1} u_{j_2} + 2}{N}\right) \ldots\ldots (\ddagger)$$

  - Set $\mathbf{c} =< c_1, \ldots, c_\ell >, \tilde{\mathbf{c}} =< \tilde{c}_1, \ldots, \tilde{c}_\ell >, \mathbf{x} = (x_1, \ldots, x_\kappa)$, and $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_\kappa)$.

– The cipherext is $C = (\mathbf{c}, \tilde{\mathbf{c}}, \mathbf{x}, \tilde{\mathbf{x}})$.

Note that the response by the challenger is as in the scheme except for the line marked with (*), where the challenger chooses R uniformly from the set of quadratic non-residues (maintaining a list, if necessary). Since this is the only change between Game 3 and Game 4, adversary $\mathcal{A}$ will not notice the difference assuming the QR assumption holds for RSAgen. Thus,

$$|\Pr[\mathrm{W}_4] - \Pr[\mathrm{W}_3]| = \mathrm{QRAdv}_{\mathrm{RSAgen},\mathcal{B}_2}(\lambda) \tag{4.26}$$

- Game 5: This is the same as Game 4, with the following change in the line marked with (†) in the challenge phase in Game 4. The change is as follows:

  – Choose $z_j \in_R \{-1, +1\}, 0 \leq j \leq \kappa - 1$.

  – For $j \leq \kappa - 1$, set $c_j = z_j \cdot \left(\frac{2y_j u_j + 2}{N}\right)$ and $\tilde{c}_j = z_j \cdot \left(\frac{2\tilde{y}_j u_j + 2}{N}\right)$ in (†)

  We now show,

  $$\Pr[\mathrm{W}_5] = \Pr[\mathrm{W}_4] \tag{4.27}$$

  In Game 4, consider the process of encrypting a single bit m_j for $0 \leq j \leq \kappa - 1$. As R is chosen to be quadratic non-residue by the challenger and as u is taken to be quadratic residue, we have uR to be quadratic non-residue. So by **Prop-3** and Lemma 4.3.3, for $0 \leq j \leq \kappa - 1$, the Jacobi symbols $\left(\frac{2y_j u_j + 2}{N}\right)$, $\left(\frac{2\tilde{y}_j u_j + 2}{N}\right)$ are uniformly and independently distributed in $\{-1, +1\}$, which in turn makes the distribution of first κ bits of $\mathbf{c}, \tilde{\mathbf{c}}$ uniform and independent. This proves equation (4.27).

- Game 6: This is the same as Game 5, with the following change in the line marked with (‡) in the challenge phase in Game 4 (hence in Game 5). This change makes the challenge ciphertext C^* independent of the challenge bit b. We change the line (‡) in the challenge phase of Game 5 as follows:

  – Choose $z_j \in_R \{-1, +1\}, \kappa \leq j \leq \ell - 1$.

  – For $j > \kappa - 1$, set $c_j = z_j \cdot \left(\frac{2y_{j_1 j_2} u_{j_1} u_{j_2} + 2}{N}\right)$ and $\tilde{c}_j = z_j \cdot \left(\frac{2\tilde{y}_{j_1 j_2} u_{j_1} u_{j_2} + 2}{N}\right)$ in (‡).

We first show the following,

$$|\Pr[\mathrm{W}_6] - \Pr[\mathrm{W}_5]| \leq \frac{1}{2^\kappa} \tag{4.28}$$

One can see that the distribution of the $\ell - \kappa$ ($\kappa \leq j \leq \ell - 1$) bits $\left(\frac{2y_{j_1 j_2} u_{j_1} u_{j_2} + 2}{N}\right)$ and $\left(\frac{2\tilde{y}_{j_1 j_2} u_{j_1} u_{j_2} + 2}{N}\right)$ in $\mathbf{c}, \tilde{\mathbf{c}}$ depends on the distribution of the first κ bits $\left(\frac{2y_j u_j + 2}{N}\right)$ and $\left(\frac{2\tilde{y}_j u_j + 2}{N}\right)$ in $\mathbf{c}, \tilde{\mathbf{c}}$ respectively. In adversary's view the ciphertexts are indistinguishable unless he correctly guess the distribution of the first κ many Jacobi symbols of $\mathbf{c}$ or $\tilde{\mathbf{c}}$. As the first κ many Jacobi symbols are distributed uniformly and independently (in $\mathbf{c}$ and $\tilde{\mathbf{c}}$ respectively), to $\mathcal{A}$'s view, the latter bits are also independently and uniformly distributed. Hence,

$$|\Pr[\mathrm{W}_6] - \Pr[\mathrm{W}_5]| \leq \frac{1}{2^\kappa}$$

Also in Game 6 we clearly have,

$$\Pr[\mathrm{W}_6] = \frac{1}{2} \tag{4.29}$$

Now combining equations (4.21) through (4.29) proves the theorem. □

Efficiency Comparison

We use $\mathcal{CIBE}$ to denote Cocks scheme [35], $\mathcal{BGHB}$ to denote BasicIBE of [24], $\mathcal{BGHT}$ to denote BasicIBE with tradeoff of [24] and $\mathcal{JBMB}$ to denote the scheme proposed in this chapter. Message length is denoted by ℓ and elements are referred as elts.

Table 4.2: Efficiency Comparison of Our IBE Scheme

Algorithms	Schemes	Hashing	Solution of Equations	Inversion
Encrypt	$\mathcal{CIBE}$	1	0	2ℓ
	$\mathcal{BGHB}$	ℓ	$\ell+1$	$\approx \ell$
	$\mathcal{JBMB}$	1	$2\lceil\sqrt{\ell}\rceil$	2ℓ
Decrypt	$\mathcal{CIBE}$	1	0	0
	$\mathcal{BGHB}$	ℓ	$\ell+1$	$\approx \ell/2$
	$\mathcal{JBMB}$	1	0	$\ell - \lceil\sqrt{\ell}\rceil$

Table 4.3: Comparison of Ciphertext and Private Key Length of Our PKE Scheme

		$\mathcal{CIBE}$	$\mathcal{BGHB}$	$\mathcal{BGHT}$	$\mathcal{JBMB}$
Ciphertext	$\mathbb{Z}/N\mathbb{Z}$ elts	2ℓ	1	$\sqrt{\ell}$	$2\lceil\sqrt{\ell}\rceil$
	Bits	2ℓ	2ℓ	2ℓ	2ℓ
Private key	$\mathbb{Z}/N\mathbb{Z}$ elts	1	ℓ	ℓ	1

BGH's Method to Find Solutions of $Rx^2 + Sy^2 = 1$ in $\mathbb{Z}/N\mathbb{Z}$

We discuss the method by BGH [24] which uses lattice based Cremona-Rusin algorithm [40] to solve (4.1).

Cremona-Rusin Method

We first describe the lattice based method to solve the quadratic form,

$$aX^2 + bY^2 + cZ^2 = 0 \tag{4.30}$$

where $abc \neq 0$. Equation (4.30) is often called Legendre's equation. We also assume that $a > 0, b > 0$ and $c < 0$.

Definition 4.3.2 *[40] A triple $(k_1, k_2, k_3) \in \mathbb{Z}^3$ is called a solubility certificate for (4.30) if it gives a solution to the following congruences*

$$X_1^2 = -bc(\text{mod } a); \quad X_2^2 = -ca(\text{mod } b); \quad X_3^2 = -ab(\text{mod } c) \tag{4.31}$$

Theorem 4.3.3 *[40] Let a, b and c be nonzero integers with abc squarefree, not all of the same sign. Then (4.30) has a rational solution if and only if a solubility certificate exists.*

If a, b and c are pairwise coprime (but not necessarily square-free), then the existence of a solubility certificate is sufficient, but no longer necessary, for the existence of a rational solution to (4.30).

Thus Theorem 4.3.3 implies the *existence* of a rational solution to (4.30) if a solubility certificate exists. For a given certificate, a solution can be found as follows. To the triple of coefficients (a, b, c) and the certificate (k_1, k_2, k_3), associate a 3-dimensional sublattice $\mathcal{L} = \mathcal{L}(a, b, c; k_1, k_2, k_3)$ of $\mathbb{Z}^3$ as follows:

$$\mathcal{L}(a, b, c; k_1, k_2, k_3) = \{(x, y, z) \in \mathbb{Z}^3 \mid by \equiv k_1 z \pmod{a}, cz \equiv k_2 x \pmod{b}, ax \equiv k_3 y \pmod{c}\}$$

The index of $\mathcal{L}(a, b, c; k_1, k_2, k_3)$ in $\mathbb{Z}^3$ is $|abc|$. One easily checks that for $(x, y, z) \in \mathbb{L}$, we have $ax^2+by^2+cz^2 \equiv 0 \pmod{abc}$. Moreover, Minkowski's theorem implies that $\mathcal{L}$ contains a nonzero vector (x, y, z) satisfying,

$$\max(|a|x^2, |b|y^2, |c|z^2) \leq |abc| \tag{4.32}$$

Inequality (4.32) implies that,

$$-|abc| < ax^2 + by^2 + cz^2 < 2|abc|$$

So either $ax^2+by^2+cz^2 = 0$ or $ax^2+by^2+cz^2 = |abc|$. In the former case, we have a integer solution to (4.30), but in the latter case we do not have a solution. To fix this problem, Cochrane and Mitchell in [34], impose extra 2-adic condition to define a sublattice $\mathcal{L}'$ of index 2 in $\mathcal{L}$ such that points $(x, y, z) \in \mathcal{L}'$ satisfy $ax^2 + by^2 + cz^2 \equiv 0 \pmod{2abc}$, and apply a theorem of Gauss to assert the existence of a point $(x, y, z) \in \mathcal{L}'$ with $|ax^2 + by^2 + cz^2| < 2|abc|$, giving a solution. In order to solve this problem into an algorithm for solving equation (4.1)in practice, one needs methods of finding short vectors in 3-dimensional lattices, since the shortest vector in $\mathcal{L}'$ certainly gives a solution. Though finding short vectors in lattices is difficult, Cremona-Rusin shows that in 3 dimensional it is easy [40, Lemma 2.7].

BGH Method: Solution of (4.1)

Given $Rx^2 + Sy^2 = 1 \pmod{N}$ one does the following. Consider the ternary quadratic form of the type (4.2), where $\tilde{R}, \tilde{S}, x, y, z \in \mathbb{Z}$ and $\tilde{R} \equiv R \pmod{N}$,

$\tilde{S} \equiv S \pmod{N}$. Clearly a solution to (4.2) in $\mathbb{Z}$ gives a solution to (4.1) in $\mathbb{Z}/N\mathbb{Z}$. Now (4.2) is of the form (4.30). So one can call Cremona-Rusin algorithm to solve (4.2). To this end one has to provide a solubility certificate for (4.2). By (4.31), for certificate one has to solve the following congruences,

$$r^2 \equiv \tilde{R} \pmod{\tilde{S}} \text{ and } s^2 \equiv \tilde{S} \pmod{\tilde{R}} \tag{4.33}$$

Congruences (4.33) are easy to solve if $\tilde{R}$ and $\tilde{S}$ are primes. Thus BGH finds primes $\tilde{R}$ and $\tilde{S}$ in the arithmetic progressions $\tilde{R} \equiv R \pmod{N}$, $\tilde{S} \equiv S \pmod{N}$ respectively. The compulsion of finding same solution of (4.1) in BasicIBE implies, the encryptor and the decryptor should find same primes $\tilde{R}, \tilde{S}$ as input for Cremona-Rusin algorithm.

4.4 C. A Public Key Encryption

In [24], following the idea of BasicIBE, Boneh-Gentry-Hamburg also proposed a public key encryption scheme (BasicPKE). This scheme is IND-CPA secure in the standard model. The encryption time of this scheme is not ideal due to the similar reasons that we have discussed earlier for BasicIBE. In this Section we present an efficient public key encryption scheme. The scheme is a modification of BasicPKE. Our scheme is more time efficient than the Boneh-Gentry-Hamburg's BasicPKE scheme. Space efficiency of BasicPKE and our scheme remain same. In the standard model, the scheme is IND-CPA secure based on the combined hardness of quadratic residuosity problem and RSA problem.

4.4.1 Boneh-Gentry-Hamburg Public Key Encryption

We recall BGH's public key encryption BasicPKE verbatim from [24].

- **KeyGen:** Fix a positive integer ℓ for message length parameter and λ for security parameter.
  - Generate primes $(p, q) \leftarrow \text{RSAgen}(\lambda)$. Compute $N = p \cdot q$.
  - For $j = 1, \ldots, \ell$, picks random $r_j \in \mathbb{Z}/N\mathbb{Z}$ and sets $R_j \equiv r_j^2 \pmod{N}$.
  - Output public key $pk = (R_1, \ldots, R_\ell)$ and secret key $sk = (r_1, \ldots, r_\ell)$.

- **Enc:** Encrypt a message $m = m_1, \ldots, m_\ell \in \{-1, 1\}^\ell$ with the publik key $(R_1, \ldots, R_\ell)$ as follows:
  - Picks a random $s \in \mathbb{Z}/N\mathbb{Z}$ and sets $S \equiv s^2 \pmod{N}$.
  - For $j = 1, \ldots, \ell$, compute:
$$(f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S) \text{ and } c_j = m_j \cdot \left(\frac{g_j(s)}{N}\right)$$
  - Output ciphertext $(S, c_1, \ldots, c_\ell)$.
- **Dec:** Decrypt ciphertext $(S, c_1, \ldots, c_\ell)$ with the secret key $(r_1, \ldots, r_\ell)$ as follows.
  - For $j = 1, \ldots, \ell$ compute:
$$R_j \equiv r_j^2 \pmod{N}, (f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S), m_j = c_j \cdot \left(\frac{f_j(r_j)}{N}\right)$$
  - Output $m = m_1, \ldots, m_\ell$.

Security

We state the security theorem of BasicPKE.

Theorem 4.4.1 *[24] The public key encryption BasicPKE is IND-CPA secure in the standard model if the quadratic residuosity 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 about the same as that of $\mathcal{A}$) such that*

$$\mathrm{Adv}^{\text{IND-CPA}}_{\text{BasicPKE},\mathcal{A}}(\lambda) = \mathrm{QRAdv}_{\text{RSAgen},\mathcal{B}}(\lambda)$$

4.4.2 The Modified Boneh-Gentry-Hamburg's PKE Scheme

We now present our scheme. We call it SSPke.

- **KeyGen:** Fix a positive integer ℓ for message length parameter and λ for security parameter.
  - Generate primes $(p, q) \leftarrow \mathrm{RSAgen}(\lambda)$ and compute $N = p \cdot q$.

- Choose $r_1, \ldots, r_\ell \in_R \mathbb{Z}/N\mathbb{Z}$ and compute $R_j \equiv r_j^2 \pmod N$, $1 \leq j \leq \ell$.
- Choose $e, d \in_R \mathbb{Z}/N\mathbb{Z}$ such that $e \cdot d \equiv 1 \pmod{\phi(N)}$.
- Output public key $pk = (R_1, \ldots, R_\ell, e)$ and secret key $sk = (r_1, \ldots, r_\ell, d)$.

- **Enc:** Encrypt message $m = m_1, \ldots, m_\ell \in \{-1, 1\}^\ell$ with the public key $(R_1, \ldots, R_\ell, e)$ as follows:
  - Choose $s, t \in_R \mathbb{Z}/N\mathbb{Z}$.
  - For $1 \leq j \leq \ell$, compute $(x_j, y_j) = \left(\frac{-2st}{R_j + s^2t^2}, \frac{R_j - s^2t^2}{s \cdot (R_j + s^2t^2)}\right)$ (Note that for $1 \leq j \leq \ell$, (x_j, y_j) is a solution of the curve $R_j x^2 + Sy^2 \equiv 1 \pmod N$, where $S \equiv s^2 \pmod N$).
  - For $1 \leq j \leq \ell$, compute $c_j = m_j \cdot \left(\frac{2y_j s + 2}{N}\right)$.
  - Compute $Y \equiv (st)^e \pmod N$.
  - Output the ciphertext $C = (c_1, \ldots, c_\ell, Y)$.

- **Dec:** Decrypt ciphertext $C = (c_1, \ldots, c_\ell, Y)$ with the secret key $(r_1, \ldots, r_\ell, d)$ as follows:
  - Compute $st \equiv Y^d \pmod N$.
  - For $1 \leq j \leq \ell$, compute $x_j = \frac{-2st}{R_j + s^2t^2}$.
  - Compute $m_j = c_j \cdot \left(\frac{x_j r_j + 1}{N}\right)$.
  - Output plaintext $m = m_1, \ldots, m_\ell$.

This completes the description of the public key encryption SSPke. Soundness follows from **Prop** 1 of Definition 4.3.1.

Remark 4.4.1 *A variant of the above scheme can be obtained as follows. In the encryption algorithm, one may use Rabin encryption [93] to hide st instead of RSA function. As inverting Rabin function is as hard as factoring, the security of our scheme will reduce to quadratic residuosity assumption. The disadvantage in using Rabin encryption is that one has to employ disambiguation technique to obtain the requisite unique st out of the four square roots of* $(st)^2$.

4.4.3 Security

We present the semantic security of the scheme against a CPA adversary.

Theorem 4.4.2 *Let $\mathcal{A}$ ba a polynomial time IND-CPA adversary attacking the above scheme SSPke. Then there exists efficient algorithms, QR algorithm $\mathcal{B}_1$ and RSA algorithm $\mathcal{B}_2$ respectively (whose running time is about the same as that of $\mathcal{A}$) such that,*

$$\mathrm{Adv}^{\text{IND-CPA}}_{\text{SSPke},\mathcal{A}}(\lambda) \leq \mathrm{QRAdv}_{\mathrm{RSAgen},\mathcal{B}_1}(\lambda) + \mathrm{RSAAdv}_{\mathrm{RSAgen},\mathcal{B}_2}(\lambda)$$

Proof : Let $\mathcal{A}$ be a IND-CPA adversary against the semantic security of the encryption scheme SSPke. We show that under the hypothesis of the theorem, $\mathrm{Adv}^{\text{IND-CPA}}_{\text{SSPke},\mathcal{A}}$ is a negligible function in λ. In the proof below, we incrementally define a sequence of games starting at the real Game 0 and ending up at Game 2. In each of the games, the challenger chooses a bit $b \in_R \{0,1\}$ and the adversary makes a guess b'. By W_i we will denote the event that the bit b' is equal to the bit b in the ith game.

- Game 0: This is the usual adversial game for defining IND-CPA security of public key encryption schemes. Thus we have,

$$|\Pr[\mathrm{W}_0] - \frac{1}{2}| = \mathrm{Adv}^{\text{IND-CPA}}_{\text{SSPke},\mathcal{A}}(\lambda) \tag{4.34}$$

- Game 1: This is a modification of Game 0 whereby the KeyGen algorithm is modified. The challenger picks $R_1, \ldots, R_\ell \in_R J(N) \backslash QR(N)$ and $e, d \in \mathbb{Z}/N\mathbb{Z}$ such that $e \cdot d \equiv 1 \pmod{\phi(N)}$. Then adversary $\mathcal{A}$ is fed with $pk = (R_1, \ldots, R_\ell, e)$. $\mathcal{A}$ outputs a pair of messages (m^0, m^1). Next a challenge ciphertext is produced by choosing $b \in_R \{0,1\}$ and encrypting $m^b = m^b_1, \ldots, m^b_\ell$ as follows:
  - Choose $s, t \in_R \mathbb{Z}/N\mathbb{Z}$.
  - For $1 \leq j \leq \ell$, compute $(x_j, y_j) = \left(\frac{-2st}{R_j + s^2t^2}, \frac{R_j - s^2t^2}{s \cdot (R_j + s^2t^2)}\right)$ (Note that these solutions are random and independent).
  - Encrypt jth bit m^b_j as follows,

$$c_j = m^b_j \cdot \left(\frac{2y_j s + 2}{N}\right) - - - - - - - - - - - - - (*)$$

– Compute $Y \equiv (st)^e \pmod{N}$.

– Output challenge ciphertext $C = (c_1, \ldots, c_\ell, Y)$.

The only change between Game 0 and Game 1 is that $R_1, \ldots, R_\ell$ are chosen from $J(N) \backslash QR(N)$ instead of $QR(N)$. $\mathcal{A}$ will not notice the difference assuming that the QR assumption holds for RSAgen. In particular, there exists an algorithm $\mathcal{B}_1$ (whose running time is about the same as that of $\mathcal{A}$) such that,

$$|\Pr[\mathrm{W}_1] - \Pr[\mathrm{W}_0]| \leq \mathrm{QRAdv}_{\mathrm{RSAgen},\mathcal{B}_1}(\lambda) \tag{4.35}$$

- Game 2: This is identical to Game 1 except the following change which makes the challenge ciphertext C be independent of the challenge bit b. We change the line marked with $(*)$ in Game 1 as follows:

$$\text{For } 1 \leq j \leq \ell, \text{ choose } z_j \in_R \{-1, 1\} \text{ and set } c_j = z_j \cdot \left(\frac{2y_j s + 2}{N}\right)$$

As a result, the challenge ciphertext C is an encryption of a random message $z_1, \ldots, z_\ell$, independent of the bit b. We argue that because R_j's, $(1 \leq j \leq \ell)$ are chosen to be quadratic non-residues, by Lemma 4.3.3, each of the ℓ many Jacobi symbols $\left(\frac{2y_j s+2}{N}\right)$ are uniformly distributed over $\{-1, +1\}$ and as (x_j, y_j)'s are random points on the ℓ many curves $R_j x^2 + Sy^2 \equiv 1 \pmod{N}$, these Jacobi symbols are independently distributed. This makes the Game 2 and Game 1 indistinguishable unless the adversary finds out the eth root st of $Y((st)^e \equiv Y \pmod{N})$ as with the knowledge of st, $\mathcal{A}$ can compute the Jacobi symbol of $2y_j s + 2 = 2 \cdot \frac{R_j - s^2t^2}{s \cdot (R_j + s^2t^2)} \cdot s + 2 = 2 \cdot \left(\frac{R_j - s^2t^2}{R_j + s^2t^2}\right) + 2$. But computation of st will amount to solve the RSA problem. Thus first ℓ bits of the ciphertexts in Game 2 and Game 1 are indistinguishable. As we apply RSA encryption on a random number st and as st is independent of the bit b the last part of the ciphertexts is also indistinguishable. Thus, there exists an algorithm $\mathcal{B}_2$ (whose running time is about the same as that of $\mathcal{A}$) such that,

$$|\Pr[\mathrm{W}_2] - \Pr[\mathrm{W}_1]| \leq \mathrm{RSAAdv}_{\mathrm{RSAgen},\mathcal{B}_2}(\lambda) \tag{4.36}$$

Clearly in Game 2, we have

$$\Pr[\mathrm{W}_2] = \frac{1}{2} \tag{4.37}$$

Now combining equations (4.34) through (4.37) proves,

$$\mathrm{Adv}^{\text{IND-CPA}}_{\text{SSPke},\mathcal{A}}(\lambda) \leq \mathrm{QRAdv}_{\mathrm{RSAgen},\mathcal{B}_1}(\lambda) + \mathrm{RSAAdv}_{\mathrm{RSAgen},\mathcal{B}_2}(\lambda)$$

□

Efficiency Comparison

We compare BasicPKE with SSPke. Ciphertext length remain same for both the schemes. The encryption time of SSPke is much faster than BasicPKE. In SSPke, to encrypt ℓ a bit message, the encryptor has to solve ℓ many equations of the form (4.1). These equations can be solved using the algorithm $\mathcal{R}$ (requires just one inversion in $\mathbb{Z}/N\mathbb{Z}$ for each equation, see Section 4.2.1). In BasicPKE, the encryptor has to use the BGH's time consuming algorithm to solve ℓ many equation of the form (4.1). Decryptor does not solve any equation in SSPke whereas in BasicPKE like encryptor, the decryptor does the same amount of work.

We restrict the comparison with BasicPKE only. Though in literature, there are number of famous IND-CPA secure schemes based on quadratic residuosity assumption. One may note that in 1982 Goldwasser and Micali [57] proposed a semantically secure public-key encryption based on quadratic residuosity assumption. Ciphertext overhead is much high in their scheme. Later Blum and Goldwaser [18] proposed a very efficient scheme. The scheme was semantically secure and the underlying hardness assumption was the intractibility of factoring problem. Their scheme uses the Blum-Blum-Shub pseudorandom generator [16] to obtain an efficient hard-core function with linear output length.

4.5 Conclusion

In this chapter we discussed the quadratic congruence $Rx^2 + Sy^2 \equiv 1 \pmod{N}$, where N is an RSA modulus and R, S are quadratic residues modulo N. We described, using elementary methods, a useful characterization of solutions of $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ and a count of the number of solutions of $Rx^2 + Sy^2 \equiv 1 \pmod{N}$.

Further, we described a identity-based encryption scheme without pairings based on the quadratic residuosity assumption in the random oracle model by

suitably modifying Boneh-Gentry-Hamburg's BasicIBE. Our scheme is more time efficient than Boneh-Gentry-Hamburg's BasicIBE, but is less space efficient. Compare to Cocks' IBE, our scheme is more space efficient. Time efficiency of our scheme is comparable to Cocks' IBE. Some of the immediate open problems that remain are:

- Construction of an IBE scheme without pairings based on the quadratic residuosity assumption such that it is both: space and time efficient.
- Another natural and theoretical important problem is to design an IBE scheme secure without random oracles based on quadratic residuosity. Currently, all the three schemes, Cocks, BGH and ours need the random oracle model to prove the security based on quadratic residuosity assumption.

We also described a public key encryption scheme. In the standard model, the scheme is IND-CPA secure based on the combined hardness of quadratic residuosity problem and RSA problem. Our scheme is more time efficient than the Boneh-Gentry-Hamburg's BasicPKE scheme. Space efficiency of BasicPKE and our scheme remain same.

Chapter 5

A Practical (Non-interactive) Publicly Verifiable Secret Sharing Scheme

5.1 Introduction

The verifiable secret sharing (VSS) schemes constitute a particular interesting class of schemes as they allow each receiver of information about the secret (share of the secret) to verify that the share is consistent with the other shares. If the dealer trusts one of the shareholders completely, he could share the 'whole' secret with the person and thus altogether avoid the trouble of using a secret sharing scheme. Therefore in many applications the dealer doesn't trust the shareholders completely, and therefore it is reasonable to expect that (some of) the shareholders do not trust the dealer either. For this reason efficient verifiable secret sharing schemes are necessary in practice. Verifiable secret sharing was proposed first in [33]. In a VSS scheme, the shareholders can verify the validity of their shares and thus overcome the problem of dishonest dealers. VSS is known to play important roles in various cryptographic protocols such as the multiparty protocols [12, 32], key-escrow cryptosystems [76], and threshold cryptography. A VSS scheme is called **non-interactive** if the shareholders can verify their share without talking-to each other or the dealer. Proposals by [49, 88] contributed to non-interactiveness and improved efficiency.

(Non-interactive) Publicly Verifiable Secret Sharing: The first proposed

VSS scheme [33] has the special property that anyone, not only the shareholders, can verify that the shares were correctly distributed. In [111], the property was named **public verifiability** and the VSS schemes with the above property were named publicly verifiable secret sharing schemes (PVSS). Some of the important PVSS schemes were presented in [53, 102, 111].

In most PVSS schemes, the verification procedure involves interactive proofs of knowledge. These proofs are made non-interactive by means of the Fiat-Shamir technique [50] and thus security for verifiability can only be carried out in the random oracle model [9]. Transforming security analysis of cryptographic primitives from the framework of random oracle model to the standard model has always turned out to be a theoretically important task which is seemingly difficult in most of the cases. Some of these problems were dealt in [61, 99]. Some of the positive features of [61] are: non-interactive PVSS, Fiat-Shamir technique is not used, unconditional security for public verifiability and security for indistinguishability of secrets.

Although, [61] successfully avoids Fiat-Shamir technique, their public verification algorithm is inefficient. In particular, for n shareholders, one has to compute $2n$ many pairings in the public verification algorithm. This number of pairing computations is expensive. Therefore, an important problem was to reduce the number of pairing computations during the public verification algorithm.

Our Contribution

In this chapter we propose a practical and provably secure non-interactive (t, n)-threshold PVSS scheme. Our scheme has the following features:

- Public verification algorithm is non-interactive and is obtained without using Fiat-Shamir zero knowledge proofs.

- Comparing with the public verification step of [61], our scheme provides optimal efficiency in terms of the number of pairing computations. In public verification step of [61], one needs to compute $2n$ many pairings, where n is the number of shareholders, whereas in our scheme the number of pairing computations is 4 only. This count is irrespective of the number of shareholders. We also observe that a simple modification to the verification algorithm of [61] reduces the number of pairing computations from $2n$ to

$n+1$. But this modification is done at the cost ([61] enjoys unconditional security for public verifiability) of reducing the security of public verifiability to a new computational problem.

- The scheme is provably secure against a SA-IND (see Section 2.11.1) adversary. The security relies on the hardness of a problem that we call the (n,t)- multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework [20].

5.2 Heidarvand and Villar's PVSS Scheme

Here, we recall the PVSS scheme of [61]. A bilinear map group system $(p, G, \tilde{G}, e(\cdot,\cdot))$ is generated where the bilinear map is $e : G \times G \rightarrow \tilde{G}$. The aim is to share efficiently a random value from $\tilde{G}$. The Dealer, $\mathcal{D}$, will achieve this by first randomly selecting two independent generators $g, h \in G$, $s \in \mathbb{F}_p$ and then distributing shares of the secret $e(h,h)^s$.

- **Initialization:** Participants P_i's generates their respective private keys $x_i \in_R \mathbb{F}_p^*$ and registers $y_i = h^{x_i}$ as their respective public keys.

- **Distribution:** The dealer wishes to distribute the secret among participants $P_1, \ldots, P_n$. The dealer picks a random polynomial P of degree $t-1$ in $\mathbb{F}_p[x]$:

$$P(x) = \sum_{j=0}^{t-1} \alpha_j x^j$$

and sets $s = \alpha_0$. The dealer keeps this polynomial secret but publishes the secret commitment values $C_j = g^{\alpha_j}$, $0 \leq j \leq t-1$. The dealer also publishes the shares deriving values $Y_i = y_i^{P(i)}$, $1 \leq i \leq n$, where y_i's are the public keys of the participants.

- **Verification:** An external verifier can check the correctness of the shares as follows. For $i = 1$ to n, it computes

$$X_i = \prod_{j=0}^{t-1} C_j^{i^j}$$

and checks if the following equalities holds.

$$e(X_i, y_i) = e(g, Y_i)$$

If the verification fails, all participants exit the protocol. **Note that the verification step requires $2n$ pairing computations**.

- **Reconstruction:** Using its private key x_i, each participant finds the share $S_i = h^{P(i)}$ from Y_i by computing $S_i = Y_i^{x_i^{-1}}$. Then all participants pool their shares. All shares can be verified by other participants by checking the equation $e(S_i, y_i) = e(Y_i, h)$. After the verification, if there are at least t correct shares, then for an arbitrary set A of t participants who have pooled correct shares can get h^s by Lagrange interpolation:

$$\prod_{P_i \in A} S_i^{\lambda_i} = \prod_{P_i \in A} (h^{P(i)})^{\lambda_i} = h^{\sum_{P_i \in A} \lambda_i P(i)} = h^{P(0)} = h^s$$

where $\lambda_i = \prod_{P_j \in A \setminus \{P_i\}} \frac{j}{j-i}$ is a Lagrange coefficient. The secret will be recovered by computing $e(h^s, h)$.

Correctness of Verifiability

$$\begin{aligned} e(X_i, y_i) &= e(\prod_{j=0}^{t-1} C_j^{i^j}, y_i) \\ &= e(\prod_{j=0}^{t-1} g^{\alpha_j \cdot i^j}, y_i) \\ &= e(g^{P(i)}, y_i) = e(g, y_i^{P(i)}) = e(g, Y_i) \end{aligned}$$

Security

We state the security theorems of Heidarvand and Villar's PVSS Scheme [61].

Theorem 5.2.1 *[61] The PVSS scheme is publicly verfiable in the presence of an unbounded adversary.*

Theorem 5.2.2 *[61] The PVSS scheme is SA-IND secure based on the Decisional Bilinear Square assumption [61, Section 5.3].*

5.3 The (n, t)-MSE-DDH (Multi-sequence of Exponents Diffie-Hellman) Assumption

In this Section we define the (n, t)- multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). Our scheme's security relies on the hardness of this problem. (n, t)-MSE-DDH problem falls under the general Diffie-Hellman exponent problem framework [20]. Some of the problems that are similar to (n, t)-MSE-DDH, were considered in [41, 42, 62] and all of them fit the framework of general Diffie-Hellman exponent problem. [20] provides an intractability bound for the general Diffie-Hellman exponent problem in the generic model [107], where the underlying groups are equipped with pairings. Thus the generic complexity of (n, t)-MSE-DDH and the other similar problems mentioned in [41, 42, 62] are covered by the analysis in [20]. A proof to show the (n, t)-MSE-DDH problem as a particular instance of general Diffie-Hellman exponent problem is similar to the proof of [42], where it has been shown that the (l, m, t)-MSE-DDH [42] (l, m, t are integers) problem fits the framework of general Diffie-Hellman exponent problem.

Let $G_1, G_2, \tilde{G}$ be three groups of the same prime order p, and let $e : G_1 \times G_2 \rightarrow \tilde{G}$ be a non-degenerate and efficiently computable bilinear map. Let g_1 be a generator of G_1 and g_2 be a generator of G_2.

Let n, t be two positive integers ($t \leq n$). The (n, t)-multi-sequence of exponents Diffie-Hellman problem ((n, t)-MSE-DDH) related to the group triplet $(G_1, G_2, \tilde{G})$ is as follows:

- **Input:** Two polynomials θ_1, θ_2 as

$$\theta_1(x) = \prod_{i=1}^{n}(x + a_i) \text{ and } \theta_2(x) = \prod_{i=1}^{n-t+1}(x + b_i)$$

  where $a_1, \ldots, a_n$ and $b_1, \ldots, b_{n-t+1}$ are all distinct elements in $\mathbb{F}_p$. Thus degrees of θ_1, θ_2 are n and $n - t + 1$ respectively. We call $a_1, \ldots, a_n$ and $b_1, \ldots, b_{n-t+1}$ to be the **negative roots** of θ_1, θ_2 respectively. Beside polynomials θ_1, θ_2, the following sequences of exponentiations are also given as input,

  - $\hat{g_1} := [g_1, g_1^{\alpha}, \{g_1^{\gamma^i}\}_{i=1}^{n+t-2}, \{g_1^{\alpha\gamma^i}\}_{i=1}^{n+t}$ and $g_1^{k\alpha\theta_1(\gamma)}]$,
  - $\hat{g_2} := [g_2, g_2^{\alpha}, \{g_2^{\gamma^i}\}_{i=1}^{n-t-1}, \{g_2^{\alpha\gamma^i}\}_{i=1}^{n}$ and $g_2^{k\theta_2(\gamma)}]$,

– an element $T \in \tilde{G}$,

where $k, \alpha, \gamma \in \mathbb{F}_p^*$ and are not known.

- **Output:** a bit $b \in \{0, 1\}$ as,

$$b = \begin{cases} 1 & \text{if } T = e(g_1, g_2)^{k\theta_1(\gamma)} \\ 0 & \text{if } T \text{ is a random element of } \tilde{G} \end{cases}$$

Thus the problem is to distinguish if T is a random value or if it is equal to $e(g_1, g_2)^{k\theta_1(\gamma)}$. To be more precise, let us denote by **real** the event that $T = e(g_1, g_2)^{k\theta_1(\gamma)}$, by **random** the event that T is a random element from $\tilde{G}$ and by $\mathcal{I}(\theta_1, \theta_2, \hat{g_1}, \hat{g_2}, T)$ the input of the problem. Let λ be the size of the underlying group order. We define the **advantage** of an algorithm $\mathcal{A}$ in solving (n, t)-MSE-DDH problem as

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{(n,t)\text{–MSE-DDH}}(\lambda) = \Big|&\Pr[\mathcal{A}(\mathcal{I}(\theta_1, \theta_2, [g_1], [g_2], T)) = \\ &1|\text{real}] - \Pr[\mathcal{A}(\mathcal{I}(\theta_1, \theta_2, [g_1], [g_2], T)) = 1|\text{random}]\Big| \end{aligned}$$

where the probability is taken over all the random coins consumed by $\mathcal{A}$.

The (τ, ϵ)-(n, t)-MSE-DDH assumption holds in $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$ if no τ-time algorithm has advantage at least ϵ in solving the (n, t)-MSE-DDH problem in $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$. We say that the problem (n, t)-MSE-DDH is (τ, ϵ) hard in $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$ if the (τ, ϵ)-(n, t)-MSE-DDH assumption holds in $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$.

5.4 The Proposed (t, n)-threshold PVSS Scheme

The earlier proposals for (publicly) verifiable secret sharing scheme mostly rely on the idea of interpolating a polynomial on the exponent of a generator of a group. A sketch of the idea can be given as follows:

- Fix a cyclic group G of prime order p and a generator $g \in G$.
- Choose a polynomial $f \in \mathbb{F}_p[x]$ of degree $t - 1$, say $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$.

- The polynomial is kept secret but a **commitment** to the polynomial is published by publicly distributing the coefficients of f on the exponent of g. Shares (usually $f(i)$'s for the ith shareholder) are also published on the exponents of g.
- When t or more participants come together, they can interpolate f on the exponent of g, i.e., $g^{f(x)}$.

Our proposal, though works with polynomial interpolation, is based on a different approach. This idea is very prominent in threshold cryptography, e.g., broadcast encryption, threshold encryption, attribute based encryption etc. The approach for our scheme is inspired by the work of [41, 42, 62]. An overview of this idea can briefly be described as follows:

- Fix a cyclic group G of prime order p and a generator $g \in G$.
- Choose a polynomial $f \in \mathbb{F}_p[x]$ and **publish** it (unlike the earlier approach, f is not kept secret).
- Instead what is kept secret is a **value** (say $\gamma \in \mathbb{F}_p$) where this polynomial would later be evaluated. Some public information is made available so that one can compute $g^{f(\gamma)}$.

Scheme: Now we describe a (t, n)-threshold publicly verifiable secret sharing scheme. A special property of this scheme is that the participants are initially issued secret keys such that for every new secret that the dealer wants to share, the participants can use the same secret keys to derive the respective shares of the secret in question. Let λ be the underlying security parameter of the system.

- **Initialization:** This algorithm consists of two steps:
  - Setting up public parameters: Generates a bilinear map group system $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$. Let $\phi : G_2 \rightarrow G_1$ be an efficiently computable group isomorphism [82, 83]. Also, two generators $g \in G_1$ and $h \in G_2$ are randomly selected as well as the secret values $\alpha, \gamma \in \mathbb{F}_p^*$. The dealer then computes and publishes

$$\left(u = g^{\alpha\gamma}, h, h^{\alpha}, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n}\right)$$

– User keys generation: There are n participants $P_1, \ldots, P_n$ and each of them is given a pair of public key and secret key as: the dealer first randomly selects n many distinct elements $a_1, \ldots, a_n \in \mathbb{F}_p^*$ and consider the following polynomial,

$$f(x) = \prod_{i=1}^{n}(x + a_i)$$

Then the ith participant P_i is given public key and secret key as

$$(pk_i, sk_i) = (a_i, g^{\frac{1}{\gamma + a_i}})$$

Thus $\{a_i\}$'s are known to all, i.e., f is public. The Remark 5.4.1 below describes how the participants can verify the correctness of their respective secret keys. Also the dealer can publicly send the encrypted secret keys using any standard ElGamal like public key encryption scheme.

- **Distribution:** The dealer wishes to share a secret, which is an element in $\tilde{G}$. The secret is of the form $e(g, h)^{\alpha k}$, where k is selected randomly from $\mathbb{F}_p^*$. The dealer then computes and publishes the following values:

  – Share commitment element (SCE): This value binds the dealer's commitment to the secret and is given as,

  $$\text{SCE} = u^{-k} = g^{-k\alpha\gamma}$$

  – Share deriving element (SDE): This value contains information about all the shares of the secret for which the dealer rendered his commitment. Participants will get their share by using the respective secret keys with SDE. This value is given as,

  $$\text{SDE} = h^{\alpha k f(\gamma)}$$

  The ith participant gets his share S_i by computing,

  $$S_i = e(g^{\frac{1}{\gamma + a_i}}, \text{SDE})$$

- **Verification:** Any (external) verifier first computes and checks the following equality,

$$e(\phi(h^{\alpha}), h^{\sum_{i=0}^{n-t-1}\gamma^i}) = e(\phi(h), h^{\sum_{i=0}^{n-t-1}\alpha\gamma^i})$$

Then it computes,

$$\text{SCE}' = u^{-1} \text{ and } \text{SDE}' = h^{\alpha f(\gamma)}$$

One may note that $h^{\alpha f(\gamma)}$ can be computed using $\{h^{\alpha\gamma^i}\}_{i=1}^{n}$. The verifier then check the correctness by checking

$$e(\text{SCE}', \text{SDE}) = e(\text{SCE}, \text{SDE}')$$

One should also note that the share deriving element SDE is consistent with the share commitment element SCE if and only if there exists a scalar k such that $\text{SCE} = (\text{SCE}')^k$ and $\text{SDE} = (\text{SDE}')^k$. If the verification fails, all participants exit the protocol.

- **Reconstruction:** Let A be a qualified set of participants, i.e. it consists of at least t many participants. Let the public-keys of the participants are $a_{r_1}, \ldots, a_{r_s}$, $s \geq t$. Together with their respective shares $e(g^{\frac{1}{\gamma+a_{r_i}}}, h^{\alpha k f(\gamma)})$'s, they reconstruct the secret as follows. They first compute

$$R_1 = e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)}, \text{ where } f_{r_1,\ldots,r_s}(\gamma) = \frac{f(\gamma)}{\prod_{i=1}^{s}(\gamma + a_{r_i})}$$

[The computation of R_1 is done recursively. A simple case is described here for convenience. With $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_1}}}$ and $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_2}}}$, the element $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_1})(\gamma+a_{r_2})}}$ is derived as:

$$\left(\frac{e(g, h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_1}}}}{e(g, h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_2}}}} \right)^{\frac{1}{(a_{r_2}-a_{r_1})}}$$

Thus, in order to compute $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_1})(\gamma+a_{r_2})(\gamma+a_{r_3})}}$, one can repeat the above technique twice: first with the inputs $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_2}}}$ and $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_3}}}$ (which will output $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_2})(\gamma+a_{r_3})}}$) and secondly with the inputs $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_1})(\gamma+a_{r_2})}}$ and $e(g, h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_2})(\gamma+a_{r_3})}}$]

Next they compute,

$$R_2 = h^{\frac{1}{\gamma}(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0))}$$

The computation of R_2 can successfully be carried out using $\{h^{\gamma^i}\}_{i=1}^{n-t-1}$ as degree of $\frac{1}{\gamma}(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0)) = n - s - 1$ and $n - s - 1 \leq n - t - 1$

(as $t \leq s$). Now compute

$$\begin{aligned} e(\text{SCE}, R_2) \cdot R_1 &= e(g^{-k\alpha\gamma}, h^{\frac{1}{\gamma}(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0))}) \cdot e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)} \\ &= e(g,h)^{-k\alpha(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0))} \cdot e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)} \\ &= e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(0)} \end{aligned}$$

Finally the secret is reconstructed by computing

$$e(g,h)^{k\alpha} = \left(e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(0)}\right)^{\frac{1}{f_{r_1,\ldots,r_s}(0)}}$$

Remark 5.4.1 *The ith participant P_i, with its pair of keys*

$$(pk_i, sk_i) = (a_i, g^{\frac{1}{\gamma + a_i}})$$

can check the correctness of its secret key as follows. It first computes $h^{\alpha\gamma a_i}$ and checks if

$$\begin{aligned} e(sk_i, h^{\alpha\gamma^2} \cdot h^{\alpha\gamma a_i}) &= e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha\gamma^2} \cdot h^{\alpha\gamma a_i}) \\ &= e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha\gamma(\gamma + a_i)}) \\ &= e(g^{\alpha\gamma}, h) = e(u, h) \end{aligned}$$

Remark 5.4.2 *Our scheme couldn't provide a satisfactory answer to the following problem. During the reconstruction phase when a shareholder releases his* ***share commitment value*** $e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha k f(\gamma)})$*, there seems to be no obvious method to verify this value. One, not so interesting, wayout is to publish the hash digests of $e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha k f(\gamma)})$'s, $(1 \leq i \leq n)$ during the distribution step of the scheme. But then this would mean that the correctness of the verification can only be carried out in the random oracle model.*

Remark 5.4.3 *The* ***Reconstruction*** *algorithm of the scheme requires the computation of $R_1 = e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)}$ given $\{a_{r_i}\}_{i=1}^{s}$ and $\{e(g^{\frac{1}{\gamma + a_{r_i}}}, h^{\alpha k f(\gamma)})\}_{i=1}^{s}$. The recursive method (described in the scheme) takes time that is bounded by $\frac{(s-1)s}{2} \cdot (T_p + T_{\tilde{G}})$, where T_p is the total time of a subtraction and an inversion in $\mathbb{F}_p$ and $T_{\tilde{G}}$ the total time of a division and exponentiation in $\tilde{G}$. One may note that the computation of the elements $\left(\frac{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_i}}}}{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_j}}}}\right)^{\frac{1}{(a_{r_j} - a_{r_i})}}$'s is done by exponentiation (not by computing the high order roots) as $(a_{r_j} - a_{r_i})$ is invertible modulo the order of the elements $\left(\frac{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_i}}}}{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_j}}}}\right)$ which is p.*

5.5 Security

In this Section we present the security analysis of our scheme.

5.5.1 Verifiability

We describe that a dishonest dealer cannot cheat the shareholders without being detected in the verification.

Lemma 5.5.1 *If the dealer passes the verification step, then all qualified subsets of honest shareholders will reconstruct the same secret that dealer had wished to share.*

The dealer puts forward its commitment to the secret $e(g,h)^{\alpha k}$ by binding its essential value k as part of the secret commitment element SCE $= u^{-k} = g^{-k\alpha\gamma}$. Thus, the consistency of the public parameters $(u = g^{\alpha\gamma}, h, h^{\alpha}, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n})$ and the consistency of the share deriving element SDE$=h^{k\alpha f(\gamma)}$ with the secret commitment element SCE follows from the facts that,

- $e(\phi(h^{\alpha}), h^{\sum_{i=0}^{n-t-1}\gamma^i}) = e(\phi(h), h^{\sum_{i=0}^{n-t-1}\alpha\gamma^i})$,
- SDE$'$ $(= h^{\alpha f(\gamma)})$ and SCE$'$ are obtained respectively from the dealer's publicly committed values $\{h^{\alpha\gamma^i}\}_{i=1}^{n}$ and $u = g^{\alpha\gamma}$,
- and the equality $e(\text{SCE}', \text{SDE}) = e(\text{SCE}, \text{SDE}')$ which essentially ensures that the scalar k is same such that SCE $= (\text{SCE}')^k$ and SDE $= (\text{SDE}')^k$.

5.5.2 Indistinguishability of Secrets (IND)

In this section, we show that our (t,n)-threshold PVSS scheme is SA-IND secure, assuming that the (n,t)-MSE-DDH problem is hard to solve.

Theorem 5.5.1 *Let n, t $(t \leq n)$ be two positive integers. For any PPT adversary $\mathcal{A}$ against the SA-IND security of our (t,n)-threshold PVSS scheme, there exists an algorithm $\mathcal{B}$ that distinguishes the two distributions of the (n,t)-MSE-DDH problem, such that*

$$\mathrm{Adv}_{\mathcal{A}}^{\text{SA-IND}}(\lambda) \leq 2 \cdot \mathrm{Adv}_{\mathcal{B}}^{(n,t)\text{–MSE-DDH}}(\lambda)$$

Proof : The security reduction is to show that if there is an adversary ($\mathcal{A}$) which can break our (t, n)-threshold PVSS then one obtains an algorithm to solve (n, t)-MDE-DDH. The heart of such an algorithm is a simulator ($\mathcal{B}$) which is constructed as follows. Given an instance of (n, t)-MSE-DDH as input, the simulator plays the security game SA-IND with an adversary against (t, n)-threshold PVSS. The simulator sets up the (t, n)-threshold PVSS based on the (n, t)-MSE-DDH instance. The simulator gives the public parameters to the adversary and continues the game by answering all queries made by the adversary. The queries include, public keys of n participants and private keys of $t - 1$ corrupted participants. In the process, it randomly chooses a bit b and distributes the shares of the secret T_b using the (n, t)-MSE-DDH instance provided as input. Finally, the adversary outputs a bit b'. Based on the value of b and b', the simulator decides whether the instance it received is **real** or **random**. Intuitively, if the adversary has an advantage in breaking the scheme, the simulator also has an advantage in distinguishing between **real** and **random** instances. This leads to an upper bound on the advantage of the adversary in terms of the advantage of the simulator in solving (n, t)-MSE-DDH.

- **(n, t)-MSE-DDH Instance:** The simulator, $\mathcal{B}$, receives an instance of (n, t)-MSE-DDH as described in Section 5.3. Thus $\mathcal{B}$ is given a bilinear map group system $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$ where the size of p is λ. $\mathcal{B}$ is further given polynomials $\theta_1, \theta_2 \in \mathbb{F}_p[x]$ of degrees n and $n-t+1$ respectively as described in Section 5.3. The negative roots of θ_1 and θ_2 are denoted by $a_1, \ldots, a_n$ and $a_{n+t}, \ldots, a_{2n}$ respectively. This instance also includes,
  - $\hat{g_1} := [g_1, g_1^{\alpha}, \{g_1^{\gamma^i}\}_{i=1}^{n+t-2}, \{g_1^{\alpha\gamma^i}\}_{i=1}^{n+t}$ and $g_1^{k\gamma\theta_1(\gamma)}]$,
  - $\hat{g_2} := [g_2, g_2^{\alpha}, \{g_2^{\gamma^i}\}_{i=1}^{n-t-1}, \{g_2^{\alpha\gamma^i}\}_{i=1}^{n}$ and $g_2^{k\theta_2(\gamma)}]$,
  - an element $T \in \tilde{G}$.
- **Initialization:** $\mathcal{B}$ selects randomly $t - 1$ elements $a_{n+1}, \ldots, a_{n+t-1} \in \mathbb{F}_p^*$ (different from the input a_i's) and construct a polynomial of degree $t-1$ as,

$$\theta_0(x) = \prod_{i=n+1}^{n+t-1} (x + a_i)$$

The public parameters are defined and published in the following manner.

- $g = g_1^{\theta_1(\gamma)\theta_0(\gamma)}$,
- $h = g_2$,
- $h^\alpha, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n}$,
- $u = g_1^{\alpha\gamma\theta_1(\gamma)\theta_0(\gamma)} = (g_1^{\theta_1(\gamma)\theta_0(\gamma)})^{\alpha\gamma} = g^{\alpha\gamma}$.

One may note that $\mathcal{B}$ cannot compute g, as degree of $\theta_1(x)\theta_0(x)$ is $n+t-1$. As we see subsequently that the form of g is required only for the security analysis and $\mathcal{B}$ doesn't have to publish it. Of course $\mathcal{B}$ can compute u with $\{g_1^{\alpha\gamma^i}\}_{i=1}^{n+t}$. Thus to be precise the published parameters are,

$$u, h, h^\alpha, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n}$$

- **User Keys Generation:** There are n participants and $t-1$ of them are assumed to be corrupted, i.e., $\mathcal{B}$ will issue respective public key and secret key pairs to $\mathcal{A}$ for $t-1$ corrupted participants and only public keys for the remaining $n-t+1$ participants.
  - Corrupted participants: For $t-1$ corrupted participants $(P_{w_{n+1}}, \ldots, P_{w_{n+t-1}})$, the key pairs are issued to $\mathcal{A}$ as

$$(pk_{w_i}, sk_{w_i}) = (a_i, g_1^{\frac{\theta_1(\gamma)\theta_0(\gamma)}{\gamma+a_i}}) = (a_i, g^{\frac{1}{\gamma+a_i}}), i = n+1 \text{ to } n+t-1$$

  - Honest Participants: The remaining $n-t+1$ participants assigned their respective public keys from

$$\{a_i\}_{i=n+t}^{2n}$$

One may note that $\mathcal{B}$ can compute $g_1^{\frac{\theta_1(\gamma)\theta_0(\gamma)}{\gamma+a_i}}$'s using $\{g_1^{\gamma^i}\}_{i=1}^{n+t-2}$.

- **Distribution of secret commitment element and share deriving element:** $\mathcal{B}$ defines polynomial f, as described in the scheme, whose negative roots correspond to the public keys of all the participants. Thus,

$$f(x) = \theta_0(x)\theta_2(x)$$

$\mathcal{B}$ then proceeds to select T as the **secret** that it intends to share among the n participants by publishing the secret commitment element and share deriving element as,

$$(\text{SCE}, \text{SDE}) = (g_1^{k\alpha\theta_1(\gamma)}, g_2^{k\theta_2(\gamma)})$$

One may note that, if we set $k' = \frac{k}{\alpha\theta_0(\gamma)}$, then

$$\begin{aligned} \text{SCE} &= g_1^{-k\gamma\theta_1(\gamma)} \\ &= g_1^{-k'\alpha\gamma\theta_0(\gamma)\theta_1(\gamma)} \\ &= (g_1^{\theta_1(\gamma)\theta_0(\gamma)})^{-k'\alpha\gamma} \\ &= (g)^{-k'\alpha\gamma} = u^{-k'} \end{aligned}$$

and

$$\begin{aligned} \text{SDE} &= g_2^{k\theta_2(\gamma)} \\ &= g_2^{\alpha k'\theta_0(\gamma)\theta_2(\gamma)} \\ &= h^{\alpha k' f(\gamma)} \end{aligned}$$

Further, if T is **real**, then

$$\begin{aligned} T &= e(g_1, g_2)^{k\theta_1(\gamma)} \\ &= e(g_1, g_2)^{\alpha k'\theta_0(\gamma)\theta_1(\gamma)} \\ &= e(g, h)^{\alpha k'} \end{aligned}$$

Thus the secret T is of required form as described in the scheme. With this, the simulator now randomly selects a bit $b \in \{0, 1\}$ and sets $T_b = T$ and assigns a random value in the secret space $\tilde{G}$ to T_{1-b}. $\mathcal{A}$ is then issued

$$(\text{SCE}, \text{SDE}, T_0, T_1)$$

- **Guess:** Finally $\mathcal{A}$ outputs its guess, a bit b' for b.

Based on the value of b and b', $\mathcal{B}$ goes on to solve the (n, t)-MSE-DDH problem instance at hand as follows:

- if $b' = b$, $\mathcal{B}$ answers 1, meaning that $T = e(g_1, g_2)^{k\theta_1(\gamma)}$,
- otherwise, $\mathcal{B}$ answers 0, meaning that T is a random element of $\tilde{G}$.

Thus, the advantage of the algorithm $\mathcal{B}$ in solving the input (n, t)-MSE-DDH problem is

$$\text{Adv}_{\mathcal{B}}^{(n,t)\text{–MSE-DDH}}(\lambda)$$

$$= \Big|\Pr[\mathcal{B}(\mathcal{I}(\theta_1, \theta_2, [g_1], [g_2], T)) = 1|\text{real}] - \Pr[\mathcal{B}(\mathcal{I}(\theta_1, \theta_2, [g_1], \hat{g_2}, T)) = 1|\text{random}]\Big|$$

$$= \Big|\Pr[b' = b|\text{real}] - \Pr[b' = b|\text{random}]\Big|.$$

In the above simulation, when the event **real** occurs, the simulator $\mathcal{B}$ poses as a real challenger for $\mathcal{A}$, i.e., the distribution of all the parameters during the simulation perfectly comply with the IND security game and therefore $\Big|\Pr[b' = b|\text{real}] - \frac{1}{2}\Big| = \frac{1}{2}\text{Adv}_{\mathcal{A}}^{SA-IND}(\lambda)$. Whereas, when the event **random** occurs, the distribution of the guess bit b' is completely independent of the distribution of the bit b and thus $\Pr[b' = b]$ is equal to $\frac{1}{2}$. Putting it altogether, we obtain

$$\text{Adv}_{\mathcal{A}}^{\text{SA-IND}}(\lambda) \leq 2 \cdot \text{Adv}_{\mathcal{B}}^{(n,t)\text{–MSE-DDH}}(\lambda)$$

□

5.6 Efficiency Comparison

We use $\mathcal{HV}$ and $\mathcal{PS}$ to denote the schemes proposed in [61] and this chapter respectively. In Table 5.1, we compare $\mathcal{PS}$ with $\mathcal{HV}$ in terms of exponentiations (in the underlying groups) and pairing computations. $\mathcal{HV}$ uses *symmetric* pairing and $\mathcal{PS}$ is based on *asymmetric* pairing. Thus the group exponents are computed in G (see Section 5.2) for $\mathcal{HV}$, and in G_1, G_2 for $\mathcal{PS}$ (see Section 5.4). A list of points to better understand the comparison table is given as follows:

- n, t bears the usual meaning.
- For Reconstruction algorithm, comparison is done based on the number of operations required for t shareholders to reconstruct the secret.
- $\mathcal{HV}$ requires computation of $2t$ pairings to verify the shares, released by the t shareholders during the Reconstruction algorithm. But this number has not been counted in the comparison table as $\mathcal{PS}$ does not satisfy this property.

A Modification of Heidarvand and Villar's PVSS Scheme (Modified-$\mathcal{HV}$ Scheme)

In this Section we propose a simple modification to the verification algorithm of [61]. This simple modification reduces the number of pairing computations from

Table 5.1: Efficiency Comparison of Our PVSS Scheme

Algorithms	Schemes	Exponentiation in G, G_1 or G_2	Exponentiation in $\tilde{G}$	Pairing
Setup	$\mathcal{HV}$	n	$-$	$-$
	$\mathcal{PS}$	$3n-t+1$	$-$	$-$
Distribution	$\mathcal{HV}$	$n+t$	$-$	$-$
	$\mathcal{PS}$	2	$-$	n
Verification	$\mathcal{HV}$	$n \cdot t$	$-$	$2n$
	$\mathcal{PS}$	n	$-$	4
Reconstruction	$\mathcal{HV}$	$2t$	$-$	1
	$\mathcal{PS}$	$n-t-1$	$\approx \frac{(t-1)t}{2}$	1

$2n$ to $n+1$ in the verification algorithm. Thus we present only the new verification algorithm.

- **Verification**: An external verifier can check the correctness of the shares as follows. For $i = 1$ to n, it computes

$$X_i = \prod_{j=0}^{t-1} C_j^{i^j}$$

and checks if the following equality holds,

$$\prod_{i=1}^{n} e(X_i, y_i) = e(g, \prod_{i=1}^{n} Y_i)$$

If the verification fails, all participants exit the protocol. **Note that the verification step requires $n+1$ pairing computations**.

The (n,t)-RDHE (The Representation of (n,t) Diffie-Hellman Exponent) Problem

We relate the security of the public verifiability of Modified-$\mathcal{HV}$ to a variant of computational Diffie-Hellman problem. We call it, the (n,t)-RDHE (The Representation of (n,t) Diffie-Hellman Exponent) Problem.

Let G be a multiplicative group with prime order p and n, t be two positive integers ($t \leq n$). We assume that p is significantly larger than n. The representation of (n, t) Diffie-Hellman exponent problem related to the group G is described as follows:

- **Input:** A degree $(t-1)$ polynomial in $\mathbb{F}_p[x]$,

$$P(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_{t-1} x^{t-1}$$

  Beside $P(x)$, the tuple of exponents $[h, h^{d_1}, \ldots, h^{d_n}]$ is also given as input, where $h \in G$ and $d_1, \ldots, d_n \in \mathbb{F}_p$.

- **Output:** A tuple $(h^{d_1})^{a_1}, \ldots, (h^{d_n})^{a_n}$ such that,

$$(a_1, \ldots, a_n) \neq (P(1), \ldots, P(n)) \text{ but } h^{\sum_{i=1}^{n} d_i a_i} = h^{\sum_{i=1}^{n} d_i P(i)}$$

  where $a_1, \ldots, a_n \in \mathbb{F}_p$.

Let λ be the size of the underlying group order. We define the advantage of an algorithm $\mathcal{A}$ in solving (n, t)-RDHE problem as

$$\Pr[\mathcal{A}(P(x), h, h^{d_1}, \ldots, h^{d_n}) = (h_1^d)^{a_1}, \ldots, (h_n^d)^{a_n}) \mid (a_1, \ldots, a_n) \neq (P(1), \ldots, P(n)) \text{ but } h^{\sum_{i=1}^{n} d_i a_i} = h^{\sum_{i=1}^{n} d_i P(i)}]$$

where the probability is over the random choice of generator h in G, the random choice of $P(x) \in \mathbb{F}_p[x]$, and the random bits consumed by $\mathcal{A}$.

Definition 5.6.1 *We say that the (τ, ϵ)-(n, t)-RDHE assumption holds on G if no τ-time algorithm has advantage at least ϵ in solving the (n, t)-RDHE problem on G.*

Security

The following theorem states that if the dealer passes the verification, then all participants in the protocol must behave honestly or will be detected.

Theorem 5.6.1 *Under the (n, t)-RDHE assumption, if verifier accepts, then there exists a unique polynomial $P(x)$ such that the encrypted share of participant P_i is $Y_i = y_i^{P(i)}$ for $1 \leq i \leq n$.*

Proof: The security reduction is to show that if there is a dishonest dealer $\mathcal{A}$ who can successfully cheat in the verification algorithm of the Modified-$\mathcal{HV}$ scheme, then one obtains an algorithm $\mathcal{B}$ to efficiently solve (n,t)-RDHE problem. We describe the algorithm $\mathcal{B}$ as follows:

- $\mathcal{B}$ is given an instance of the (n,t)-RDHE problem. This instance includes a polynomial $P(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_{t-1}x^{t-1}$, and a tuple $[h, h^{d_1}, \ldots, h^{d_n}]$ of elements from the group G of order p.

- $\mathcal{B}$ selects randomly an element $g \in G$. It then feeds $\mathcal{A}$ with $(P(x), g, h, h^{d_1}, \ldots, h^{d_n})$ and asks $\mathcal{A}$ to setup the Modified-$\mathcal{HV}$ scheme with $y_i = h^{d_i}$ as the public key of the ith participant P_i, $1 \leq i \leq n$.

- $\mathcal{A}$ then sets $e(h,h)^{\alpha_0}$ as the secret, where $\alpha_0 = P(0)$.

- $\mathcal{A}$ then publishes the commitment values (for the polynomial $P(x)$) $C_j = g^{\alpha_j}$, $0 \leq j \leq t-1$. Next it selects $a_1, \ldots, a_n$ and publishes $Y_i = y_i^{a_i}$, $1 \leq i \leq n$. $\mathcal{A}$ then claims that Y_i's are the respective encrypted shares for the participants P_i's. Thus $\mathcal{A}$ can successfully cheat in the above claim (amounts to cheating in verification) if

$$(a_1, \ldots, a_n) \neq (P(1), \ldots, P(n)) \text{ but } \prod_{i=1}^{n} e(X_i, y_i) = e(g, \prod_{i=1}^{n} Y_i) \qquad (5.1)$$

- Finally $\mathcal{B}$ outputs $(Y_1, \ldots, Y_n)$ as the solution to the given (n,t)-RDHE problem instance.

This completes the description of $\mathcal{B}$. Equation (5.1) implies

$$(a_1, \ldots, a_n) \neq (P(1), \ldots, P(n)) \text{ but } h^{\sum_{i=1}^{n} d_i a_i} = h^{\sum_{i=1}^{n} d_i P(i)}$$

as

$$\begin{aligned}
\prod_{i=1}^{n} e(X_i, y_i) = e(g, \prod_{i=1}^{n} Y_i) \quad &=> \quad \prod_{i=1}^{n} e(g^{P(i)}, h^{d_i}) = e(g, \prod_{i=1}^{n} y_i^{a_i}) \\
&=> \quad \prod_{i=1}^{n} e(g, h^{d_i \cdot P(i)}) = e(g, \prod_{i=1}^{n} h^{d_i \cdot a_i}) \\
&=> \quad e(g, h^{\sum_{i=1}^{n} d_i \cdot P(i)}) = e(g, h^{\sum_{i=1}^{n} d_i \cdot a_i}) \\
&=> \quad h^{\sum_{i=1}^{n} d_i P(i)} = h^{\sum_{i=1}^{n} d_i a_i}
\end{aligned}$$

Thus if the dealer $\mathcal{A}$ successfully cheats in the verification algorithm then its advantage translates to the advantage of $\mathcal{B}$ in solving the given (n, t)-RDHE problem instance.

5.7 Conclusion

We have proposed in this chapter a practical (t, n)-threshold PVSS scheme that achieves simultaneously:

- Efficient non-interactive public verification.
- Provable security for the public verifiability in the standard model.

Efficiency of the non-interactive public verification step of our scheme is optimal while comparing with the earlier proposal [61]. We provide formal security proof for our scheme, against indistinguishability of secrets (IND) attack model, based on the hardness of a problem that we call the (n, t)- multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework [20]. The security proof (for indistinguishability of secrets) could handle only static adversaries. An interesting task would be to modify the scheme accordingly so that an adaptive adversary can be handled during the security analysis. The other challenging task is to provide the security proof under a more standard assumption.

Chapter 6

An IND-CCA Secure Public Key Encryption Scheme

6.1 Introduction

In this chapter, we present an IND-CCA (indistinguishability of encryption against chosen ciphertext attack) secure public key encryption scheme. The scheme is an instance of the famous Cramer-Shoup Paradigm [39].

Chosen ciphertext security is nowdays considered as a standard notion of security for public key encryption in practice. Furthermore, this security also implies universal composable security [29] and also, as shown in [44], is equivalent to the notion of non-malleability. For these reasons, the notion of chosen ciphertext security has emerged as the "right" notion of security for encryption schemes. Indeed it can be shown that in order to model encryption as a "secure envelope", the encryption scheme used must be chosen ciphertext secure.

So far, many CCA secure public key encryption schemes have been proposed in the literature. The first schemes were presented in [44, 84, 94], but they were quite impractical. The first truly practical public key encryption that is provably secure against chosen ciphertext attack was discovered by Cramer and Shoup [37]. The security of this scheme is based on the hardness of the decisional Diffie-Hellman problem. In [39] Cramer and Shoup show that their original scheme is an instance of a more generic paradigm. This paradigm is based on the hash proof systems. In [39] they also show that their paradigm can be also instantiated with the Quadradic Residuosity and Composite Residuosity assumptions [39].

Further, Shoup [108] proposed a general KEM/DEM framework, and extended the Cramer-shoup [37] scheme to a hybrid encryption scheme. Kurosawa and Desmedt [71] improved efficiency of the hybrid version of Cramer-Shoup scheme. Abe et al. [2] proposed the Tag-KEM/DEM framework, and explained the security of Kurosawa-Desmedt scheme [71] in this framework. Hofheinz and Kiltz [63] proposed another paradigm for building hybrid encryption with strictly weakened KEM. Canetti, Halevi, and Katz (CHK) [30] proposed a generic method for converting a selectively secure identity-based encryption scheme into a CCA secure public key encryption scheme, and Boneh and Katz [22] improved its efficiency. In [69], a more relaxed condition for achieving CCA security was discussed by Kiltz. Based on CHK paradigm, in [25] Boyen, Mei and Waters presented practical CCA secure schemes. The above schemes are proven secure in the standard model and mostly based on decisional assumptions. Under the random oracle methodology [9], there exists many practical schemes, e.g. [10, 54]. Recently, remarkable results about building CCA secure public key encryption schemes based on computational problems are proposed [31, 60, 64]. An entirely different approach based on the Ajtai's seminal work on lattice-based one-way functions [3] is putting this field on a very strong footing. Some of the initial public key encryption schemes based on this approach are [4, 95]. But the above cryptosystems do not satisfy IND-CCA security. A recent interesting result is the work of Peikert and Waters [89] who, building on the cryptosystem of [96], were able to design a lattice-based cryptosystem achieving CCA security.

Thus, from the above docussion, it is very much apparent about the quantum of good work that has been done in this field. The work that we present in this chapter shows that the Cramer-Shoup paradigm can also be instantiated with the decisional bilinear Diffie-Hellman problem. This scheme can be seen as a work of theoretical importance as after the seminal paper of Cramer-Shoup, the subject of building *practical CCA secure encryption* has come a long way.

6.2 Primitives for the Cramer-Shoup Paradigm

The essential primitives for the Cramer-Shoup paradigm are the so-called *projective hash families*, *subset membership problems* and *hash proof systems*. We include an informal summary of these notions. We refer to [39] for more detailed

information.

6.2.1 Universal Projective Hashing

In this section we review universal projective hashing introduced by Cramer and Shoup [39].

Definition 6.2.1 *Let X and Π be finite, non empty sets. Let $H = (H_k)_{k \in K}$ be a collection of functions indexed by K, so that for every $k \in K$, H_k is a function from X into Π. Note that we may have $H_k = H_{k'}$ for $k \neq k'$. We call $\boldsymbol{F} = (H, K, X, \Pi)$ a hash family and each H_k a hash function.*

We now recall the concept of universal projective hashing. Let $\mathbf{F} = (H, K, X, \Pi)$ be a hash family. Let L be a nonempty, proper subset of X. Let S be a finite, nonempty set, and let $\alpha : K \to S$ be a function.

Definition 6.2.2 *[39] Set $\boldsymbol{H} = (H, K, X, L, \Pi, S, \alpha)$. We say $\boldsymbol{H}$ is a projective hash family for (X, L) if for all $k \in K$, the action of H_k on L is determined by $\alpha(k)$, i.e., given $\alpha(k)$ and $x \in L$, the value of $H_k(x)$ is uniquely determined.*

Informally, we say $\mathbf{H}$ is ϵ-*universal* if for any $x \in X \setminus L$ and for a randomly chosen k, the probability of correctly guessing $H_k(x)$ from x and $\alpha(k)$ is at most ϵ. Moreover, we say $\mathbf{H}$ is ϵ-*universal*$_2$ if even knowing the value of H_k at some $x^* \in X \setminus \{x\}$, the value of $H_k(x)$ can be only guessed correctly with probability at most ϵ. They are formally defined as follows.

Definition 6.2.3 *[39] Let $\boldsymbol{H} = (H, K, X, L, \Pi, S, \alpha)$ be a projective hash family , and let $\epsilon \geq 0$ be a real number. Consider the probability space defined by choosing $k \in K$ at random.*

- *Then $\boldsymbol{H}$ is said to be ϵ-universal if for all $s \in S$, $x \in X \setminus L$ and $\pi \in \Pi$, it holds that*

$$Pr[H_k(x) = \pi \wedge \alpha(k) = s] \leq \epsilon Pr[\alpha(k) = s]$$

- *$\boldsymbol{H}$ is said to be ϵ-universal$_2$ if for all $s \in S$, $x, x^* \in X$ and $\pi, \pi^* \in \Pi$ with $x \notin L \cup \{x^*\}$, it holds that*

$$Pr[H_k(x) = \pi \wedge H_k(x^*) = \pi^* \wedge \alpha(k) = s] \leq \epsilon Pr[H_k(x^*) = \pi^* \wedge \alpha(k) = s]$$

We say $\mathbf{H}$ is ϵ-*smooth* if the probability distribution of $(x, s, H_k(x))$ and (x, s, π), where k, x and π are chosen uniformly at random in K, $X \setminus L$ and Π respectively, and $s = \alpha(k)$, are ϵ-close.

Definition 6.2.4 *Let $\boldsymbol{H} = (H, K, X, L, \Pi, S, \alpha)$ be a projective hash family. Consider the probability space defined by choosing $k \in_R K$, $x \in_R X \setminus L$ and $\pi' \in_R \Pi$. We say for $\epsilon \geq 0$, a real number, the above family is ϵ-smooth if*

$$|\Pr[H_k(x) = \pi'] - \frac{1}{|\Pi|}| \leq \epsilon$$

6.2.2 Group-theoretic Constructions of Universal Projective Hash Families

Following [39], we present the group-theoretic constructions of universal projective hash family and universal$_2$ projective hash family.

Diverse Group System

Let X, L and Π be finite abelian groups such that L be a proper subgroup of X. Let $\mathrm{Hom}(X, \Pi)$ denote the group of all homomorphisms $\phi : X \to \Pi$. $\mathrm{Hom}(X, \Pi)$ is a finite abelian group. If group operations in X, L and Π are written additively, then the group operations in $\mathrm{Hom}(X, \Pi)$ are written as follows. For $\phi, \phi' \in \mathrm{Hom}(X, \Pi)$, $x \in X$, and $\alpha \in Z$, we have $(\phi + \phi')(x) = \phi(x) + \phi'(x), (\phi - \phi')(x) = \phi(x) - \phi'(x)$, and $(a\phi)(x) = a\phi(x) = \phi(ax)$. The zero element of $\mathrm{Hom}(X, \Pi)$ sends all elements of X to $0 \in \Pi$.

Definition 6.2.5 *Let X, L, Π be as above. Let H be a subgroup of $Hom(X, \Pi)$. Then $\boldsymbol{G} = (H, X, L, \Pi)$ is called a group system.*

Definition 6.2.6 *Let $\boldsymbol{G} = (H, X, L, \Pi)$ be a group system. We say that $\boldsymbol{G}$ is diverse if for all $x \in X \setminus L$, there exists $\phi \in H$ such that $\phi(L) = \langle 0 \rangle$, i.e., ϕ vanishes on L but $\phi(x) \neq 0$.*

Let $\mathbf{G} = (H, X, L, \Pi)$ be a group system. Let $g_1, \ldots, g_d \in L$ be a set of generators for L. Set $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$, where for randomly chosen $k \in K$, H_k is uniformly distributed over H, $S = \Pi^d$, and the map $\alpha : K \to S$ sends $k \in K$ to $(\phi(g_1), \ldots, \phi(g_d))$, where $\phi = H_k$. One may check that $\mathbf{H}$ is a projective hash family.

Definition 6.2.7 *Let $\boldsymbol{G}$ be a group system as above and let $\boldsymbol{H}$ be described as above. Then we say that $\boldsymbol{H}$ is a projective hash family derived from $\boldsymbol{G}$.*

Theorem 6.2.1 *[39] Let $\boldsymbol{G} = (H, X, L, \Pi)$ be a group system, $\boldsymbol{H} = (H, K, X, L, \Pi, S, \alpha)$ be the projective hash family derived from $\boldsymbol{G}$, and $\tilde{p}$ denoted the smallest prime dividing $|X/L|$. If $\boldsymbol{G}$ is diverse group system, then $\boldsymbol{H}$ is $\frac{1}{\tilde{p}}$-universal projective hash family.*

We now proceed to construct a universal$_2$ projective hash family. Let E be an arbitrary finite set. Fix an injective encoding function $\Gamma : X \times E \to \{0, \ldots, \tilde{p}-1\}^n$, where n is a positive integer. Let $\hat{\mathbf{H}} = (\hat{H}, K^{n+1}, X \times E, L \times E, \Pi, S^{n+1}, \hat{\alpha})$, where $\hat{H}$ and $\hat{\alpha}$ are defined as follows. For $\hat{k} = (k', k_1, \ldots, k_n) \in K^{n+1}$, $x \in X$, and $e \in E$, we define

$$\hat{H}_{\hat{k}}(x, e) = H_{k'}(x) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x)$$

where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e)$ and define $\hat{\alpha}(\hat{k}) = (\alpha(k'), \alpha(k_1), \ldots, \alpha(k_n))$. One may check that $\hat{\mathbf{H}}$ is a projective hash family for $(X \times E, L \times E)$.

Theorem 6.2.2 *[39] Let $\hat{\boldsymbol{H}}$ be as above. If the underlying group system $\boldsymbol{G}$ is diverse, then $\hat{\boldsymbol{H}}$ is $\frac{1}{\tilde{p}}$-universal$_2$ projective hash family.*

6.2.3 Subset Membership Problem

A subset membership problem (SMP) $\mathbf{M}$ specifies a collection $\{I_l\}_{l \in \mathbb{N}}$ of distributions. For every value of the security parameter $l \in \mathbb{N}$, I_l is a probability distribution of instance descriptions. An instance description Λ specifies the following:

- Non-empty sets, X, L and W such that L is a proper subset of X.
- A binary relation $R \subset X \times W$ such that $x \in L$ iff $(x, w) \in R$ for some witness $w \in W$.

SMP requires that the following probabilistic polynomial time algorithms exist.

- Instance sampling: samples an instance Λ according to the distribution I_l on system parameter 1^l.

- Subset sampling: outputs a random $x \in L$ together with a witness $w \in W$ for x on input 1^l and $\Lambda[X, L, W, R]$.
- Element sampling: outputs a random $x \in X$.

The subset membership problem is said to be hard if (Λ, x_0) and (Λ, x_1) are indistinguishable for a random $x_0 \in L$ and a random $x_1 \in X \setminus L$.

6.2.4 Hash Proof Systems

Let $\mathbf{M}$ be a susbet membership problem specifying a sequence $(I_l)_{l \geq 0}$ of instance distributions. A hash proof system (HPS) $\mathbf{P}$ for the subset membership problem $\mathbf{M}$ is described as follows:

- It associates, with each instance $\Lambda[X, L, W, R]$ of $\mathbf{M}$, a projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ for (X, L).
- Provides efficient algorithms to carry out basic operations present in the associated projective hash family; namely, sampling $k \in K$ at random, computing $\alpha(k) \in S$ given $k \in K$.
- Efficient computation of $H_k(x) \in \Pi$ given $k \in K$ and $x \in X$. This algorithm is called private evaluation algorithm for $\mathbf{P}$.
- Moreover, a crucial property is that the system provides an efficient algorithm to compute $H_k(x) \in \Pi$, given $\alpha(k) \in S$, $x \in L$ and $w \in W$, where w is a witness for x. This algorithm is called public evaluation algorithm for $\mathbf{P}$.

Definition 6.2.8 *[39] Let $\epsilon(l)$ be a function mapping non-negative integers to non-negative reals. Let $\boldsymbol{M}$ be a subset membership problem specifying a sequence $(I_l)_{l \in \mathbb{N}}$ of instance distributions. Let $\boldsymbol{P}$ be an HPS for $\boldsymbol{M}$. We say that $\boldsymbol{P}$ is $\epsilon(l)$-universal (respectively -universal$_2$, -smooth) if there exists a negligible function $\delta(l)$ such that for all $l \geq 0$ and for all $\Lambda[X, L, W, R] \in [I_l]$, the projective hash family $\boldsymbol{H} = (H, K, X, L, \Pi, S, \alpha)$ that $\boldsymbol{P}$ associates with Λ is $\delta(l)$-close to an $\epsilon(l)$-universal (respectively -universal$_2$, smooth) projective hash family $\boldsymbol{H}^* = (H^*, K^*, X, L, \Pi, S, \alpha^*)$.*

Moreover, if this is the case, and $\epsilon(l)$ is a negligible function, then we say that $\mathbf{P}$ is strongly universal (respectively, universal$_2$, smooth).

Extended Hash Proof Systems

Definition 6.2.9 *[39] For $l \geq 0$ and for all $\Lambda[X, L, W, R] \in [I_l]$ an extended HPS $\boldsymbol{P}$ for $\boldsymbol{M}$ associates with Λ a finite set E along with a projective hash family $\boldsymbol{H} = (H, K, X \times E, L \times E, \Pi, S, \alpha)$ for $(X \times E, L \times E)$. All the related properties of HPS can similarly be defined for extended HPS.*

6.3 The Generic Cramer-Shoup Scheme

We now recall Cramer-Shoup generic method for constructing a secure public-key encryption scheme. Let $\mathbf{M}$ be a subset membership problem specifying a sequence of $(I_l)_{l \in \mathbb{N}}$ instance distributions. Let $\mathbf{P}$ be a strongly smooth hash proof system for $\mathbf{M}$ and $\hat{\mathbf{P}}$ be strongly universal$_2$ extended hash proof system for $\mathbf{M}$. To simplify the notation, we will describe the scheme with respect to a fixed instance description $\Lambda[X, L, W, R] \in [I_l]$ given by the instance sampling algorithm (provided by $\mathbf{M}$). This fixed value of l is the underlying security parameter. With this fixed instance Λ, Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be the projective hash family that $\mathbf{P}$ associates with Λ and, let $\hat{\mathbf{H}} = (\hat{H}, \hat{K}, X \times \Pi, L \times \Pi, \hat{\Pi}, \hat{S}, \hat{\alpha})$ be the projective hash family that $\hat{\mathbf{P}}$ associates with Λ. We require that Π is an abelian group and it is written additively. The group operations within Π can be done efficiently. We now describe the scheme. The message space is Π.

- **KeyGen:**
  - Randomly select $k \in K$ and $\hat{k} \in \hat{K}$, and compute $s = \alpha(k) \in S$ and $\hat{s} = \hat{\alpha}(\hat{k}) \in \hat{S}$.
  - The public key is $(s, \hat{s})$.
  - The secret key is $(k, \hat{k})$.
- **Enc:** Encrypt a message $m \in \Pi$ with the public key as follows:
  - Generate a random $x \in L$, together with a witness $w \in W$. This is done using the subset membership algorithm provided by $\mathbf{M}$.
  - Compute $\pi = H_k(x) \in \Pi$. This is done using the public evaluation algorithm for $\mathbf{P}$ on inputs s, x and w.
  - Compute $\theta = m + \pi \in \Pi$.

- Compute $\hat{\pi} = \hat{H}_{\hat{k}}(x, \theta) \in \hat{\Pi}$. This is done using the public evaluation algorithm for $\hat{\mathbf{P}}$ on inputs $\hat{s}$, x, θ and w.
- The ciphertext is $(x, \theta, \hat{\pi})$.

- **Dec:** Decrypt a ciphertext $(x, \theta, \hat{\pi})$ with the corresponding secret key as follows:
  - Compute $\hat{\pi}' = \hat{H}_{\hat{k}}(x, \theta) \in \hat{\Pi}$. This is done using the private evaluation algorithm for $\hat{\mathbf{P}}$ on inputs $\hat{k}$, x and θ.
  - Check whether $\hat{\pi}' = \hat{\pi}$; if not, then output **reject** and halt.
  - Compute $\pi = H_k(x) \in \Pi$. This is done using using the private evaluation algorithm for $\mathbf{P}$ on inputs k and x.
  - Compute $m = \theta - \pi$ and output the message m.

Theorem 6.3.1 *[39] The above scheme is IND-CCA secure, assuming the underlying subset membership problem* $\boldsymbol{M}$ *is hard.*

6.4 The Proposed Scheme

In this Section we show that the Cramer-Shoup paradigm can also be instantiated based on the decisional bilinear Diffie-Hellman problem. In particular we present a public key encryption scheme and prove that it is IND-CCA secure by showing it to be a particular instance of the Cramer-Shoup framework. We relate the hardness of the unerlying subset membership problem to the decisional bilinear Diffie-Hellman problem problem.

A bilinear map group system $(q, G, \tilde{G}, e(\cdot, \cdot))$ is generated where the bilinear map is $e : G \times G \to \tilde{G}$. We use additive notation for the group operation in G and multiplicative for $\tilde{G}$. Let $\Gamma : G \times \tilde{G} \times \tilde{G} \to \mathbb{Z}/q\mathbb{Z}^*$ be a collision resistant hash function and $f : G \to \tilde{G}$ be an efficiently computable isomorphism (see Remark 6.4.1). The message space is $\tilde{G}$.

- **KeyGen:**

– Choose $g_0, g_1, g_2 \in_R G$ and $k_0, k_1, k_{00}, k_{01}, k_{10}, k_{11} \in_R \mathbb{Z}/q\mathbb{Z}^*$. Compute

$$\begin{aligned} s_0 &= f(g_0)^{k_0} e(g_1, g_2)^{k_1} \\ s_1 &= f(g_0)^{k_{00}} e(g_1, g_2)^{k_{01}} \\ s_2 &= f(g_0)^{k_{10}} e(g_1, g_2)^{k_{11}} \end{aligned}$$

– The public key is $pk = (g_0, g_1, g_2, s_0, s_1, s_2)$.

– The secret key is $sk = (k_0, k_1, k_{00}, k_{01}, k_{10}, k_{11})$.

- **Enc:** Encrypt a message $m \in \tilde{G}$ with the public key $(g_0, g_1, g_2, s_0, s_1, s_2)$ as follows:

  – choose $w \in_R \mathbb{Z}/q\mathbb{Z}^*$,

  – compute $(x, y)=(wg_0, e(g_1, g_2)^w)$,

  – compute $\pi = s_0^w$,

  – compute $\theta = \pi m$,

  – compute $\hat{\pi}=(s_1 s_2^t)^w$, where $t = \Gamma(x, y, \theta)$,

  – output the ciphertext $(x, y, \theta, \hat{\pi})$.

- **Dec:** Decrypt the ciphertext $(x, y, \theta, \hat{\pi})$ with the secret key $(k_0, k_1, k_{00}, k_{01}, k_{10}, k_{11})$ as follows:

  – compute $\hat{\pi}' = (f(x)^{k_{00}} y^{k_{01}})(f(x)^{k_{10}} y^{k_{11}})^t$, where $t = \Gamma(x, y, \theta)$,

  – check if $\hat{\pi} = \hat{\pi}'$; if not, then output **reject** and halt,

  – compute $\pi = (f(x)^{k_0} y^{k_1})$,

  – compute $m = \theta\pi^{-1}$.

Correctness

We check the correctness of the scheme. For a valid ciphertext $(x, y, \theta, \hat{\pi})$ corresponding to the message m the decryption algorithm first computes $\hat{\pi}'$ which is

equal to $\hat{\pi}$ as follows:

$$\begin{aligned}
\hat{\pi}' = & \left(f(x)^{k_{00}} y^{k_{01}}\right)\left(f(x)^{k_{10}} y^{k_{11}}\right)^t \\
= & \left(f(wg_0)^{k_{00}}(e(g_1,g_2)^w)^{k_{01}}\right)\left(f(wg_0)^{k_{10}}(e(g_1,g_2)^w)^{k_{11}}\right)^t \\
= & \left(f(g_0)^{k_{00}}(e(g_1,g_2))^{k_{01}}\right)^w\left(f(g_0)^{k_{10}}(e(g_1,g_2))^{k_{11}}\right)^{wt} \\
= & \left((f(g_0)^{k_{00}}(e(g_1,g_2))^{k_{01}})(f(g_0)^{k_{10}}(e(g_1,g_2))^{k_{11}})^t\right)^w \\
= & (s_1 s_2^t)^w = \hat{\pi}.
\end{aligned}$$

Next we have,

$$\begin{aligned}
(f(x)^{k_0} y^{k_1}) &= (f(wg_0)^{k_0}(e(g_1,g_2)^w)^{k_1}) \\
&= (f(g_0)^{k_0}(e(g_1,g_2))^{k_1})^w = s_0^w = \pi.
\end{aligned}$$

Remark 6.4.1 *The map f in the scheme can be instantiated as follows. For fixed non-zero element $g_0 \in G$, $f(x) = e(g_0, x)$ for all $x \in G$. This is clearly an isomorphism from $G \to \tilde{G}$. Also $e(g_1, g_2)$ can be taken as part of the pulic key.*

6.5 Security

In this Section we present the security analysis of our scheme.

Theorem 6.5.1 *Assuming the hardness of decisional bilinear Diffie-Hellman (DBDH) problem, the proposed public key encryption scheme is IND-CCA secure.*

Proof : We prove the security by showing the scheme to be a particular instance of the Cramer-Shoup framework. We describe the following primitives that constitute the scheme according to the framework of Cramer-Shoup. They are

- a subset membership problem $\mathbf{M}$.
- a smooth projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ which can be obtained given an instance $\Lambda[X, L, W, R]$ of the subset membership problem $\mathbf{M}$.
- a universal$_2$ projective hash family $\hat{\mathbf{H}} = (\hat{H}, \hat{K}, X \times \Pi, L \times \Pi, \hat{\Pi}, \hat{S}, \hat{\alpha})$ which can be obtained given the instance $\Lambda[X, L, W, R]$.

In particular, efficient constructions of these primitive describes the underlying Hash Proof Systems.

The Subset-Membership Problem

In this Section we describe the underlying subset membership problem $\mathbf{M}$. We relate the hardness of $\mathbf{M}$ to the decisional bilinear Diffie-Hellman problem. To simplify the notation, we describe an instance of $\mathbf{M}$.

A bilinear map group system $(q, G, \tilde{G}, e(\cdot,\cdot))$ is generated where the bilinear map is $e : G \times G \to \tilde{G}$. We use additive notation for the group operation in G and multiplicative for $\tilde{G}$. Let $g_0,\ g_1,\ g_2$ be randomly chosen elements of G. Define $X = G \times \tilde{G}$ and $L = \langle (g_0, e(g_1, g_2)) \rangle$ be the subgroup of X generated by the element $(g_0, e(g_1, g_2))$. For any element $(x_0, x_1) \in X$, a witness is a $w \in \mathbb{Z}/q\mathbb{Z}$ such that $(x_0, x_1) = (wg_0, e(g_1, g_2)^w)$, i.e., a witness for (x_0, x_1) ensures its membership in L. One can efficiently sample a random element of L together with a witness by generating a $w \in \mathbb{Z}/q\mathbb{Z}$ at random and then computing the element as $(wg_0, e(g_1, g_2)^w)$. We now define the subset membership problem: efficiently distinguish between elements of L and $X \setminus L$. To be precise, consider the following two distributions:

$$\mathcal{D}_L = \{g_0, g_1, g_2, rg_0, \bar{A} = e(g_1, g_2)^r \mid g_0, g_1, g_2 \in G, r \in \mathbb{Z}/q\mathbb{Z}\}$$

$$\mathcal{D}_{X\setminus L} = \{g_0, g_1, g_2, rg_0, \bar{A} = e(g_1, g_2)^{r'} \mid g_0, g_1, g_2 \in G, r, r' \in \mathbb{Z}/q\mathbb{Z}, r \neq r'\}$$

The subset membership problem is to distinguish these two distributions. We now show that the hardness of this problem is equivalent to the DBDH problem. We assume the existence of an efficient algorithm $\mathcal{O}_{\mathbf{M}}$ that solves $\mathbf{M}$. We use this algorithm to solve the DBDH problem. Let an input instance of the DBDH problem is given as (g, ag, bg, cg, Z). We fed $\mathcal{O}_{\mathbf{M}}$ with the tuple $(g_0 = g, g_1 = bg, g_2 = cg, rg_0 = ag_0 = ag, \bar{A} = Z)$. If $Z = e(g, g)^{abc}$, then $A = Z = e(g, g)^{abc} = e(bg, cg)^a = e(g_1, g_2)^r$. Thus a real input instance of DBDH problem corresponds to a input member of the distribution $\mathcal{D}_L$.

We now assume the existence of an efficient algorithm $\mathcal{O}_{\text{DBDH}}$ that solves the $DBDH$ problem. We use this algorithm to solve the subset membership problem $\mathbf{M}$. Let an input instance of $\mathbf{M}$ is given as $(g_0, g_1, g_2, rg_0, \bar{A})$. We fed $\mathcal{O}_{\text{DBDH}}$ with the tuple $(g = g_0, ag = rg_0, bg = bg_0 = g_1, cg = cg_0 = g_2, Z = \bar{A})$. If $\bar{A} = e(g_1, g_2)^r$, then $Z = \bar{A} = e(g_1, g_2)^r = e(bg, cg)^a = e(g, g)^{abc}$. Thus a input member of the distribution $\mathcal{D}_L$ corresponds to a real input instance of the DBDH problem. Hence, the hardness of the subset membership is equivalent to the hardness of DBDH problem.

Construction of Smooth Projective Hash Family and Universal$_2$ Projective Hash Family

We now proceed for a group-theoretic construction of smooth projective hash family and universal$_2$ projective hash family. As dicussed in Section 6.2.2, we first require a diverse group system. With X, L defined as above and f as in the scheme, set $K = \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$. Define for each $(k_0, k_1) \in K$, a map $H_{k_0,k_1} : G \times \tilde{G} \to \tilde{G}$ as follows. For $(x, y) \in X = G \times \tilde{G}$,

$$H_{k_0,k_1}(x, y) = f(x)^{k_0} y^{k_1}$$

It can be checked that the correspondence $(k_0, k_1) \to H_{k_0,k_1}$ is a bijection between K and $\mathrm{Hom}(X, \tilde{G})$, the set of all group homomorphisms from X to $\tilde{G}$. Set $H = \mathrm{Hom}(X, \tilde{G})$ and consider the group system $\mathbf{G} = (H, X, L, \tilde{G})$. We show that $\mathbf{G}$ is a diverse group system. Set $X' = \tilde{G} \times \tilde{G}$ and $L' = \langle (f(g_0), e(g_1, g_2)) \rangle$, subgroup of X' generated by $(f(g_0), e(g_1, g_2))$. Define $H' = \mathrm{Hom}(X', \tilde{G})$. The map $(k_0, k_1) \to H'_{(k_0,k_1)}$, where for $(x, y) \in X'$, $H'_{k_0,k_1}(x, y) = x^{k_0} y^{k_1}$, is an 1-1 correspondence between $K = \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ and $H' = \mathrm{Hom}(X', \tilde{G})$. We now have two group systems:

$$\mathbf{G} = (H, X, L, \tilde{G}) \text{ and } \mathbf{G}' = (H', X', L', \tilde{G})$$

It follows from [39] that $\mathbf{G}' = (H', X', L', \tilde{G})$ is a diverse group system. We use this fact to show that $\mathbf{G} = (H, X, L, \tilde{G})$ is also a diverse group system.

The map $(k_0, k_1) \to H_{(k_0,k_1)}$, where $H_{k_0,k_1}(x, y) = f(x)^{k_0} y^{k_1}$, is an 1-1 correspondence between $\mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ and H. Let (x', y') be any element in $X \setminus L$. Then $(x', y') = (ag_0, e(g_1, g_2)^b)$ for some $a \neq b$, $a, b \in \mathbb{Z}/q\mathbb{Z}$. We have to show that there exists $(k_0, k_1) \in \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ such that $H_{(k_0,k_1)}(x', y') = f(x')^{k_0} y'^{k_1} \neq 0$ but $H_{(k_0,k_1)}$ vanishes on L. Now $(f(x'), y') = (f(g_0)^a, e(g_1, g_2)^b)$, clearly $(f(x'), y') \in X' \setminus L'$. As $\mathbf{G}'$ is a diverse group system, there exists a tuple $(k_0, k_1) \in \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ such that the corresponding homomorphism ${H'}_{k_0,k_1}$ vanishes on L' but ${H'}_{k_0,k_1}(f(x'), y') = f(x')^{k_0} y'^{k_1} \neq 0$. Since ${H'}_{k_0,k_1}(f(x), y) = H_{k_0,k_1}(x, y)$ for all $(x, y) \in X$, $H_{(k_0,k_1)}$ vanishes on L and $H_{(k_0,k_1)}(x', y') \neq 0$. This proves that $\mathbf{G} = (H, X, L, \tilde{G})$ is a diverse group system.

We now construct a $\frac{1}{q}$-universal projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ from the diverse group system $\mathbf{G} = (H, X, L, \tilde{G})$ as follows. With H, K, X, L as above, set $\Pi = \tilde{G}$ and $S = \tilde{G}$. The map $\alpha : K \to S$ is defined as follows :

$$\alpha(k_0, k_1) = H_{k_0,k_1}(g_0, e(g_1, g_2)) = f(g_0)^{k_0} e(g_1, g_2)^{k_1}$$

where $(k_0, k_1) \in K$. Theorem 6.2.1 implies that the resulting family is a $\frac{1}{q}$-universal projective hash family. We recall that $\tilde{p}$ here is q as $|X| = q^2$ and $|L| = q$. Further this family is 0-smooth.

We now construct a $\frac{1}{q}$-universal$_2$ projective hash family using the above family. Let $\Gamma : X \times \tilde{G} \rightarrow \mathbb{Z}/q\mathbb{Z}$ be a collision resistant hash function. Set $\hat{\mathbf{H}} = (\hat{H}, K \times K, X \times \tilde{G}, L \times \tilde{G}, \tilde{G}, S \times S, \hat{\alpha})$, where for $((k_0, k_1), (k_0', k_1')) \in K \times K$, define $\hat{H}_{((k_0,k_1),(k_0',k_1'))} \in \hat{H}$ as follows,

$$\hat{H}_{((k_0,k_1),(k_0',k_1'))}((x, y), \theta) = H_{(k_0,k_1)}(x, y).(H_{(k_0',k_1')}(x, y))^t$$

where $(x, y) \in X$, $\theta \in \tilde{G}$ and $\Gamma((x, y), \theta) = t$. Define the map $\hat{\alpha}$ as follows:

$$\hat{\alpha}((k_0, k_1), (k_0', k_1')) = (\alpha(k_0, k_1), \alpha((k_0', k_1'))$$

where $((k_0, k_1), (k_0', k_1')) \in K \times K$. Theorem 6.2.2 ensure that this family is a $\frac{1}{q}$-universal$_2$ projective hash family. One can check that these two families are used in the proposed scheme according to the Cramer-Shoup paradigm. This completes the proof. □

6.6 Conclusion

In this chapter we have shown that the Cramer-Shoup paradigm can also be instantiated based on the decisional bilinear Diffie-Hellman problem. In particular we present a public key encryption scheme and prove that it is IND-CCA secure by showing it to be a particular instance of the Cramer-Shoup framework. We relate the hardness of the underlying subset membership problem to the decisional bilinear Diffie-Hellman problem.

Bibliography

[1] M. Abadi and P. Rogaway. Reconciling Two Views of Cryptography: The Computational Soundness of Formal Encryption. *Journal of Cryptology*, 15(2):103–127, 2002.

[2] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 128–146. Springer, 2005.

[3] M. Ajtai. Generating Hard Instances of Lattice Problems. Complexity of Computations and Proofs. *Quaderni di Matematica 13. Preliminary version in STOC 1996*, pages 1–32, 2004.

[4] M. Ajtai and C. Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. *ACM Symposium on Theory of Computing*, pages 284–293, 1997.

[5] M. Backes, B. Pfitzmann, and M. Waidner. A Composable Cryptographic Library with Nested Operations. *ACM Conference on Computer and Communications Security*, pages 220–230, 2003.

[6] N. Bari and B. Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer, 1997.

[7] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.

[8] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.

[9] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[10] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption. In *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.

[11] M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 519–536. Springer, 1999.

[12] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract). *ACM Symposium on Theory of Computing*, pages 1–10, 1988.

[13] K. Bentahar, P. Farshim, J. Malone-Lee, and N.P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. *Journal of Cryptology*, 21(2):178–199, 2008.

[14] I. Blake, G. Seroussi, and N. Smart. *Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series)*. Cambridge University Press New York, NY, USA, 2005.

[15] G. Blakley. Safeguarding Cryptographic Keys. In *AFIPS National Computer Conference*, volume 48, pages 313–317, 1979.

[16] L. Blum, M. Blum, and M. Shub. A Simple Unpredictable Pseudo-Random Number Generator. *SIAM Journal on Computing*, 15(2):364–383, 1986.

[17] M. Blum. Coin Flipping by Telephone: A Protocol for Solving Impossible Problems. *SIGACT News*, 15(1):23–27, 1983.

[18] M. Blum and S. Goldwasser. An Efficient Probabilistic Public-Key Encryption Scheme Which Hides All Partial Information. In *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 289–302. Springer, 1984.

[19] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[20] D. Boneh, X. Boyen, and E. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.

[21] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. Earlier version appeared in the proceedings of CRYPTO 2001.

[22] D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer, 2005.

[23] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001.

[24] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-Efficient Identity Based Encryption Without Pairings. *IEEE Symposium on Foundations of Computer Science*, pages 647–657, 2007.

[25] X. Boyen, Q. Mei, and B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. In *ACM Conference on Computer and Communications Security*, pages 320–329, 2005.

[26] X. Boyen and B. Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.

[27] D. A. Burgess. The Distribution of Quadratic Residues and Non-residues. *Mathematika*, 4:106–112, 1957.

[28] D. A. Burgess. On Character Sums and Primitive Roots. *Proceedings of London Mathematical Society*, 12(3):179–192, 1962.

[29] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. *IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2001.

[30] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.

[31] D. Cash, E. Kiltz, and V. Shoup. The Twin Diffie-Hellman Problem and Applications. *Journal of Cryptology*, 22(4):470–504, 2009.

[32] D. Chaum, C. Crépeau, and I. Damgård. Multiparty Unconditionally Secure Protocols (Extended Abstract). *ACM Symposium on Theory of Computing*, pages 11–19, 1988.

[33] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). *IEEE Symposium on Foundations of Computer Science*, pages 383–395, 1985.

[34] T. Cochrane and P. Mitchell. Small Solutions of the Legendre Equation. *Journal of Number Theory*, 70:62–66, 1980.

[35] C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In *Cryptography and Coding: 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.

[36] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1993.

[37] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 1998.

[38] R. Cramer and V. Shoup. Signature Schemes Based on the Strong RSA Assumption. In *ACM Conference on Computer and Communications Security*, pages 46–51, 1999.

[39] R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.

[40] J. E. Cremona and D. Rusin. Efficient Solution of Rational Conics. *Mathematics of Computation*, 72(243):1417–1441, 2003.

[41] C. Delerablée, P. Paillier, and D. Pointcheval. Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys. In *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59. Springer, 2007.

[42] C. Delerablée and D. Pointcheval. Dynamic Threshold Public-Key Encryption. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 317–334. Springer, 2008.

[43] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[44] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography (Extended Abstract). *ACM Symposium on Theory of Computing*, pages 542–552, 1991.

[45] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *Technical Report CS95-27, Weizmann Institute of Science*, 1995.

[46] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[47] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[48] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469 – 472, 1985.

[49] P. Feldman. A Practical Scheme for Non-Interactive Verifiable Secret Sharing. *IEEE Symposium on Foundations of Computer Science*, 0:427–438, 1987.

[50] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[51] R. Fischlin and C. P. Schnorr. Stronger Security Proofs for RSA and Rabin Bits. *Journal of Cryptology*, 13(2):221–244, 2000.

[52] E. Fujisaki and T. Okamoto. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 1997.

[53] E. Fujisaki and T. Okamoto. A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications. In *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1998.

[54] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[55] S. D. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate Pairing. In *ANTS*, volume 2369 of *Lecture Notes in Computer Science*, pages 324–337. Springer, 2002.

[56] C. Gentry. Practical Identity-Based Encryption without Random Oracles. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.

[57] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and Systems Sciences*, 28(2):270–299, 1984.

[58] J. Graaf and R. Peralta. A Simple and Secure Way to Show the Validity of Your Public Key. In *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 128–134. Springer, 1987.

[59] C. Gutiérrez. Satisfiability of Equations in Free Groups is in PSPACE. *ACM Symposium on Theory of Computing*, pages 21–27, 2000.

[60] G. Hanaoka and K. Kurosawa. Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 308–325. Springer, 2008.

[61] S. Heidarvand and J. L. Villar. Public Verifiability from Pairings in Secret Sharing Schemes. In *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 294–308. Springer, 2008.

[62] J. Herranz, F. Laguillaumie, and C. Ràfols. Constant Size Ciphertexts in Threshold Attribute-Based Encryption. In *PKC*, volume 6056 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2010.

[63] D. Hofheinz and E. Kiltz. Secure Hybrid Encryption from Weakened Key Encapsulation. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 553–571. Springer, 2007.

[64] D. Hofheinz and E. Kiltz. Practical Chosen Ciphertext Secure Encryption from Factoring. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 313–332. Springer, 2009.

[65] D. Hofheinz and E. Kiltz. The Group of Signed Quadratic Residues and Applications. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 637–653. Springer, 2009.

[66] S. Hohenberger. *The Cryptographic Iimpact of Groups with Infeasible Inversion*. Master's thesis, EECS Dept., MIT, 2003.

[67] R. Impagliazzo and B. M. Kapron. Logics for Reasoning about Cryptographic Constructions. *IEEE Symposium on Foundations of Computer Science*, pages 372–383, 2003.

[68] M. Joye and G. Neven, editors. *Identity-Based Cryptography*. IOS Press Cryptology and Information Security Series, 2008.

[69] E. Kiltz. Chosen-Ciphertext Security from Tag-Based Encryption. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006.

[70] N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[71] K. Kurosawa and Y. Desmedt. A New Paradigm of Hybrid Encryption Scheme. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2004.

[72] R. C. Lyndon and M. Newman. Commutators as Products of Squares. *Proceedings Americal Mathematical Society*, 39(2):267–272, 1973.

[73] R. C. Lyndon and P. E. Schupp. Combinatorial Group Theory. *Springer*, 1977.

[74] S. Micali M. Blum, A. D. Santis and G. Persiano. Non-Interactive Zero-Knowledge. In *SIAM Journal on Computing*, volume 20(6), pages 1084–1118, 1991.

[75] G. S. Makanin. Equations in Free Groups. *Izvestiya NA SSSR*, 46:1199–1273, 1982. English translation in Math USSR Izvestiya, 21 (1983), 483-546.

[76] S. Micali. Fair Public-Key Cryptosystems. In *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 113–138. Springer, 1992.

[77] D. Micciancio. The RSA group is Pseudo-free. *Journal of Cryptology*, 23(2):169–186, 2010. A preliminary version appeared in Eurocrypt 2005.

[78] D. Micciancio and S. Panjwani. Adaptive Security of Symbolic Encryption. In *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 169–187. Springer, 2005.

[79] D. Micciancio and B. Warinschi. Soundness of Formal Encryption in the Presence of Active Adversaries. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[80] V. S. Miller. Use of Elliptic Curves in Cryptography. In *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1986.

[81] J. C. Mitchell, A. Ramanathan, A. Scedrov, and V. Teague. A Probabilistic Polynomial-time Process Calculus for the Analysis of Cryptographic Protocols. *Theoretical Computer Science*, 353(1-3):118–164, 2006.

[82] Atsuko Miyaji, Masaki Nakabayashi, and Shunzo Takano. Characterization of Elliptic Curve Traces under FR-Reduction. In *ICISC*, volume 2015 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2000.

[83] Atsuko Miyaji, Masaki Nakabayashi, and Shunzo Takano. New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001.

[84] M. Naor and M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. *ACM Symposium on Theory of Computing*, pages 427–437, 1990.

[85] G. Neven. A Simple Transitive Signature Scheme for Directed Trees. *Theoretical Computer Science*, 396(1-3):277–282, 2008.

[86] T. Okamoto and D. Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2001.

[87] H. Ong, C. P. Schnorr, and A. Shamir. An Efficient Signature Scheme Based on Quadratic Equations. *ACM Symposium on Theory of Computing*, pages 208–216, 1984.

[88] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[89] C. Peikert and B. Waters. Lossy Trapdoor Functions and their Applications. *ACM Symposium on Theory of Computing*, pages 187–196, 2008.

[90] J. M. Pollard and C. P. Schnorr. An Efficient Solution of the Congruence $x^2 + ky^2 = m \pmod{n}$. *IEEE Transactions on Information Theory*, 33(5):702–709, 1987.

[91] G. Pólya. Über Die Verteilung Der Quadratischen Reste und Nichtreste. *Göttinger Nachrichte*, pages 21–29, 1918.

[92] K. Ohgishi R. Sakai and M. Kashahara. Cryptosystems Based on Pairings. In *Symposium on Cryptography and Information Security*, 2000.

[93] M. O. Rabin. Digital Signatures and Public Key Functions as Intractable as Factorization. *Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology*, 1979.

[94] C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[95] O. Regev. New Lattice Based Cryptographic Constructions. *ACM Symposium on Theory of Computing*, pages 407–416, 2003.

[96] O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *ACM Symposium on Theory of Computing*, pages 84–93, 2005.

[97] R. L. Rivest. On the Notion of Pseudo-Free Groups. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 505–521. Springer, 2004.

[98] R. L. Rivest, A. Shamir, and L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[99] A. Ruiz and J. L. Villar. Publicly Verfiable Secret Sharing from Paillier's Cryptosystem. *Lecture Notes in Informatics*, pages 98–108, 2005.

[100] R. Sakai and M. Kasahara. ID based Cryptosystems with Pairing on Elliptic curve. *Symposium on Cryptography and Information Security*, 2003.

[101] A. De Santis, G. D. Crescenzo, and G. Persiano. Secret Sharing and Perfect Zero Knowledge. In *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 73–84. Springer, 1993.

[102] B. Schoenmakers. A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic. In *CRYPTO*, volume 5381 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 1999.

[103] R. Schoof. Elliptic Curves Over Finite Fields and the Computation of Square Roots (mod p). *Mathematics of Computation*, 44(170):483–494, 1985.

[104] J. P. Serre. *Cours d'arithmetique*. P.U.F. 3rd edition, 1988.

[105] A. Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.

[106] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[107] V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997.

[108] V. Shoup. Using Hash Functions as a Hedge against Chosen Ciphertext Attack. In *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 275–288. Springer, 2000.

[109] G. J. Simmons. How to (Really) Share a Secret. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 390–448. Springer, 1988.

[110] N. P. Smart. *The Algorithmic Resolution of Diophantine Equations*. London Mathematical Society Student Texts, Cambridge University Press, 1998.

[111] M. Stadler. Publicly Verifiable Secret Sharing. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 190–199. Springer, 1996.

[112] D. R. Stinson. An Explication of Secret Sharing Schemes. *Design, Codes and Cryptography*, 2(4):357–390, 1992.

[113] I. M. Vinogradov. Sur la distribution des résidus et des non-résidus despuissances. *J. Phys.-Math. Soc. Perm. No. 1*, pages 94–96, 1918.

[114] B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.