Quadratic Residues and Applications in Cryptography
- extended abstract -

by

Anca-Maria Nica

supervisor

Prof. Dr. Cătălin Dima

2020
Doctoral committee:

Conf.Dr. Adrian Iftene - committee chairman
Alexandru Ioan Cuza University of Iași

Prof.Dr. Cătălin Dima - doctoral supervisor
Alexandru Ioan Cuza University of Iași / “Paris Est Creteil - Val de Marne”

Prof.Dr. Constantin Popescu - reviewer
University of Oradea

Prof.Dr. Ferucio Laurențiu Tiplea - reviewer
Alexandru Ioan Cuza University of Iași

Conf.Dr. Octavian Catrina - reviewer
University Politehnica of Bucharest

Conf.Dr. Mihai Dumitru Prunescu - reviewer
University of Bucharest
# Contents

**Preface**  
Preface 1  
Thesis overview 1  
Thesis contributions 3  

List of publications 7  

1 Introduction to cryptography and quadratic residues 9  

2 Prerequisites 11  

3 On the distribution of quadratic residues 13  
3.1 Counting QR and QNR in the set $a + X$  
3.1.1 The case of prime moduli  
3.1.2 The case of RSA moduli  
3.2 Computing probabilities on sets $Y(a + X)$  

4 Applications of QR to IBE 27  
4.1 Cocks’ IBE scheme  
4.1.1 Cocks’ IBE ciphertexts  
4.1.2 Galbraith’s test  
4.1.3 Anonymous Cocks’ schemes  
4.2 BGH’s IBE scheme  
4.2.1 Associated polynomials  
4.2.2 The BGH scheme and its security  
4.2.3 A new security analysis for BasicIBE scheme  
4.3 QR-based IBE schemes that fail security
4.3.1 Jhanwar-Barua scheme .......................... 41
4.3.2 Other insecure IBE schemes based on QR 42
4.4 Continuous mutual authentication ................. 42
  4.4.1 Real privacy management ........................ 42
  4.4.2 RPM description ............................... 43
  4.4.3 Continuous mutual authentication and data security 43
4.5 Pseudo-random generators .......................... 46

5 From IBE to ABE 47
  5.1 Introduction ........................................ 47
  5.2 ABE and the backtracking attack ................. 49
    5.2.1 The secure KP-ABE Scheme_1 .................. 51
  5.3 KP-ABE for Boolean circuits using secret sharing and multilinear maps 53
    5.3.1 The secure KP-ABE Scheme_2 .................. 53

6 Conclusion and open problems ........................ 57

Selected bibliography ................................. 59
A careful analysis of Cocks’ IBE scheme leads to the study of the integers which are obtained by adding a quadratic residue to an integer in $\mathbb{Z}_n^*$, i.e. the set $a + QR_n$, as we will deeply discuss in Chapter 3 of this thesis. This was a starting point in our research, together with the proof of Galbraith’s test, addressed in detail in Section 4.1.2. The anonymization and the security of Cocks’ IBE scheme, was another point of interest in the thesis, as well as some applications of this scheme, and attribute based encryption, which is considerable useful in cloud computing, access control in cloud and other fields. These are the main subjects which we describe in this thesis.

**Thesis overview**

In what follows we will shortly describe each chapter.

**Chapter 1: Introduction to cryptography and quadratic residues**  In the first chapter, after a short review of the thesis, we present some phases in the history of cryptology. We focused on a special case of Public Key Cryptography (PKE) which is Identity-based Encryption (IBE) using quadratic residues (QR). This is one of the areas where we applied some of our mathematical results in Chapter 3.

The security level of a cryptographic scheme is usually proved using security games. The areas where quadratic residues are of
great interest can also be found in this first chapter accompanied by the literature review on the subjects discussed in this thesis.

**Chapter 2: Prerequisites**  This chapter introduces some notations, definitions, and basic results from number theory, probabilities, and complexity which we are going to use along the thesis.

**Chapter 3: On the distribution of quadratic residues**  Our research has lead to important results with exact formulas for the cardinality of a multitude of sets with different Jacobi patterns. These results can be found in this chapter. In Section 3.2 few examples of calculating probabilities using these distributions were presented. These probabilities are of great interest not only for encryption schemes, but also in various issues like security of cryptosystems or pseudo-random generators.

**Chapter 4: Applications of quadratic residues to identity-based encryption**  This chapter presents some applications of our results from Chapter 3. We deeply analyze the cryptotexts of Cocks’ scheme which is useful for the proof of Galbraith’s test. Also a much simple description of Joye’s anonymous variant of Cocks’ IBE scheme is presented in this chapter. This result was detailed in [51]. Starting from the IND-ID-CPA secure BGH scheme [8] we obtained in [62] a better upper bound for the BGH scheme which is described in Section 4.2.3. BGH gets shorter ciphertext with the cost of an expensive encryption. Unfortunately, security flows may easily occur, as we can see in some attempts of improving time efficiency of BGH, as Schipor proved in [60]. These results were clearly presented in [74].

Then a technique for continuous mutual authentication is described, namely RPM. Here we showed how, using Cocks’ scheme in one of the RMP configurations, results an improved variant of cma.
Chapter 5: From identity-based to attribute-based encryption  Chapter 5 presents a generalization of IBE with applications in a huge variety of niches as cloud computing and IoT. It begins with a brief introduction on ABE, the general structure and correctness of an ABE scheme, the backtracking attack and some deeper details on KP-ABE schemes. In Sections 5.2.1 and 5.3 two efficient KP-ABE schemes are presented, accompanied by their security proofs, implementation issues, applications, complexity and comparisons.

Chapter 6: Conclusion and open problems  In this last chapter we draw conclusions and present some open problems regarding the results obtained in the thesis and further work.

Thesis contributions

After the introduction and preliminaries in Chapters 1 and 2, the next chapters expose our work as follows. Chapter 3 presents some results we developed regarding sets such as $\mathbb{QNR}_m(a + QR_m)$, the set of integers of the form $a + QR_m$ which are quadratic non-residues modulo $m$. These sets are very useful for cryptography due to the fact that cryptographic schemes can be created using them [12, 8, 31].

Perron’s work on the distribution of quadratic residues and non-residues in sets like $a + QR_m$ focuses on prime moduli [54]. We extended these results to the case where the modulus is an RSA integer. We also generalized the case $a + QR_m$ and studied sets of the form $a + X$, where $X$ can be one of the sets $\mathbb{Z}_m$, $\mathbb{Z}_m^*$, $QR_m$, $QNR_m$, and the modulus can be either a prime or an RSA integer. In the last case, when $m$ is of the form $p \cdot q$, for some distinct primes $p$ and $q$, $X$ may also be one of the sets $J_m^\pm$ and $J_m^{\mp}$. For all these sets $a + X$ we presented not only their cardinals, but we counted the number of elements for all Jacobi patterns on these sets. Section 3.2 shows how to compute probabilities on these sets,
for example, the probability that \( x \) is in \( J_n^- \) when it is extracted uniformly at random from the set \( a + \mathbb{Z}_n^* \), see Corollary 3.2.1.

In Chapter 4 some applications of the results in Chapter 3 were detailed, together with an interesting combination between a continuous mutual authentication protocol and Cocks’ IBE scheme.

In Section 4.1 we deeply analyzed Cock’s IBE scheme and its cryptotexts’ structure in order to be able to compute the exact probability that a given cryptotext was encrypted for a given identity, see Section 4.1.2. Thus, in Section 4.1.1 we studied the way that the messages are encrypted, and how the sets of cryptotexts outputted by this scheme look like. Thus, the computations in Section 4.1.2 were done using the results achieved in Chapter 3 and the cardinalities in Section 4.1.1. Then we have shown in section 4.1.3 how efficient anonymized Cocks’ cryptotexts can be obtained from non-anonymous ones as an independent process. One such secure universal anonymous scheme is due to G.A. Schipor [61]. Right after this scheme, in Section 4.1.3 we showed how easily the anonymization variant of Cocks’ IBE scheme due to Joye [40] can be described, without using cyclotomic polynomials and algebraic toruses, as it was presented in [51].

Cocks’ IBE scheme, notwithstanding its simplicity and elegance, outputs quite large cryptotexts, \( 2\log n \) bits per bit of plaintext. Section 4.2 describes a solution proposed in 2007 by Boneh et al., the BasicIBE (shortened here into BGH) which improves the length of the cryptotexts at the cost of increasing the time complexity to quartic in the security parameter. This scheme is proven to be IND-ID-CPA secure under the QR assumption for the RSA generator in the random oracle model (ROM), as we can see in Section 4.2.2. A better upper bound for BGH scheme has been obtained in [62] and it is detailed in Section 4.2.3.

Starting from [8] Jhanwar and Barua tried to make the encryption/decryption processes faster, as it is presented in Section 4.3.1 (their scheme will be called here JB for short). The bottleneck of the scheme proposed by Boneh et al. was the algorithm for solving Equation (4.2) on page 37.
In [39], the same two researchers, Jhanwar and Barua, found a very useful probabilistic algorithm for finding solutions to Equation (4.2), instead of the deterministic one of Boneh et al. Unfortunately, the scheme proposed by them is no longer a secure variant of Cocks’ scheme due to the method of combining the solutions of two congruential equations in order to get a third solution to another equation. As A. Schipor showed, the variants of the schemes presented by Elashry, Mu, and Susilo in [23] and [21] suffer from the same security weakness. Thus, for the moment, the QR-based IBE schemes which remain secure are Cocks’ scheme, BGH and their anonymous variants, as it is detailed in [74].

An important contribution of the thesis relies to continuous mutual authentication. When two parts wish to communicate securely they (both) will want to be sure, at each moment during the process, that on the other end of the “line” is the person that they aspect to be and not a third party, not an eavesdropper. In order to achieve this, continuous (mutual) authentication is needed. But what if, at a certain point, an intruder will decode their communication? Is there any possibility that the communication become secure again during the same process, without interrupting it and start it over? This property was first defined by Elashry et al. in [22], who called it resiliency. We found a way to achieve this property using Cocks’ IBE scheme, which perfectly fits to RPM configurations, see Section 4.4.

In the end of Chapter 4 we will see how pseudorandom generators can be created using quadratic residues, which is another important application of QR in cryptography.

In Chapter 5 we outlined the latest ideas developed in the area of KP-ABE schemes based on bilinear maps and secret sharing. We conclude that, for safety, leveled multi-linear maps should be avoided. However, the current solutions for Boolean circuits in general which use bilinear maps are not efficient. So, finding a balanced variant for this kind of circuits remains an open problem.

Chapter 6 draws conclusion and presents some ideas of up-leveling and / or extending the current work.
List of publications


Chapter 1

Introduction to cryptography and quadratic residues

From ancient times cryptology played an important role, especially in the field of military services providing mainly confidentiality, integrity, and authentication. Later on, people began to be interested also in breaking ciphers, so, this gave rise to cryptanalysis. Steganography, the technique of concealing sensitive data in “innocent” messages, is another method for hiding secret data.

In the beginning of cryptography only symmetric ciphers were used in order to provide confidentiality, integrity, and authentication. This type of encryption uses the same key both for encryption and decryption. The cryptographic field has extended to asymmetric encryption since mid-twentieth century. In this case a pair of keys is used as follows: the encryptor uses receiver’s public key while the decryptor needs the correspondent secret key in order to get the original message. Public Key Encryption (PKE) is costly than symmetric encryption but nowadays they complement each other. Usually we first use a public key scheme just to transmit the secret key of a symmetric scheme which will be used to encrypt the rest of the communication due to the fact that it
is faster than PKE. Certifying the public keys in PKE involves a trust chain and a complicate management of the certificates.

In 1984 Adi Shamir proposed Identity Based Encryption (IBE), which is a new type of PKE that avoids the public key infrastructure. Here the public key can be an arbitrary string that uniquely characterize the receiver, as his phone number, email, etc.

It took 17 years until the first concrete implementations [7, 12]. IBE schemes are based mainly on: bilinear pairings on elliptic curves [58, 7], quadratic residues (QR) [12, 8, 39], and lattices [28, 20]. The bilinear maps-based IBE schemes output very short cryptotexts and have a good time complexity, however, they use some mathematical problems that are not completely understood or managed. The lattice-based schemes are of great interest because they are quantum computing resistant but their inconvenient is that the public keys are very big. QR are well understood mathematical tools, simple, and elegant. There is an effervescence now in finding an optimal balance between time and space complexity.

The first who used QR in IBE was Clifford Cocks, who published his scheme in 2001 [12]. His paper has a big impact in cryptography, and contains an important scheme with variants both, in classical PKE and in IBE. This cryptosystem presents a special interest in our thesis and will be detailed in Chapter 4.

If we want to allow the access on encrypted data to a group of people in which all the persons have some common characteristics - “attributes” -, if we have many receivers who’s identity may be unknown, or if new users want to join the system later, then Attribute-based Encryption (ABE) shall be used instead of IBE. This is a generalization of IBE to multiple decryptors and it becomes essential when talking about fine grained access control.

Perfect secrecy [63] is neither easy nor practical to provide due to the key-length and the key-exchange problem. Thus, other levels of security are used, as semantic security, indistinguishability, or non-malleability. The most usual models are COA (ciphertext-only attack), KPA (known-plaintext attack), CPA (chosen-plaintext attack), and CCA1/2 (non-/adaptive chosen-ciphertext attack).
Chapter 2

Prerequisites

In this chapter we will set the notations and give some basic notions from number theory, probabilities, complexity, and quadratic residues that we will use from now on.

Let $\mathbb{Z}$ be the set of integers and $a, b \in \mathbb{Z}$. We will denote their greatest common divisor by $(a, b)$. Let $m$ be a positive integer, $\mathbb{Z}_m$ will denote the set of the equivalence classes induced by the equivalence modulo $m$, i.e. $\{0, 1, 2, \ldots, m - 1\}$, while the set $\mathbb{Z}_m^*$ will be the set of integers $x \in \mathbb{Z}_m$ with $(x, m) = 1$. We will say that $a$ and $b$ are congruent modulo $m$ and denote this by $a \equiv b \mod m$ or $a \equiv_m b$, if $m$ divides $a - b$. The remainder of the integer division of $a$ by $m$, assuming $m \neq 0$, is denoted $(a)_m$. The integer quotient of $a$ and $m$ is denoted by $a \div m$. Positive integers $n = pq$ that are product of two distinct primes $p$ and $q$ will be usually called RSA integers or RSA moduli.

An integer $a$ co-prime with $m$ is a quadratic residue modulo $m$ if $a \equiv_m x^2$, for some integer $x$; the integer $x \in SQRTR_p(a)$ is called a square root of $a$ modulo $m$. The set of all square roots modulo $m$ of all the elements in a set $A$ will be denoted by $SQRTR_p(A)$.

Let $p$ be a prime. The Legendre symbol of an integer $a$ modulo $p$, denoted $\left(\frac{a}{p}\right)$ or $(a|p)$, is 1 if $a$ is a quadratic residue modulo $p$, 0 if $p$ divides $a$, and $-1$ otherwise. The Jacobi symbol extends the Legendre symbol to composite moduli. If $n = p_1^{e_1} \cdots p_m^{e_m}$ is
the prime factorization of the positive integer \( n \), then the Jacobi symbol of \( a \) modulo \( n \) is \( \left( \frac{a}{n} \right) = \left( \frac{p_1}{a} \right)^{e_1} \cdots \left( \frac{p_m}{a} \right)^{e_m} \). For the sake of simplicity we will use the terminology of Jacobi symbol in both cases (prime or composite moduli). For details regarding basic properties of the Jacobi symbol the reader is referred to [49, 65].

Given a positive integer \( n \) and a subset \( A \subseteq \mathbb{Z}_n^* \), \( QR_n(A) \) \((QNR_n(A), J_n^+(A), J_n^-(A))\) stands for the set of quadratic residues (quadratic non-residues, integers with the Jacobi symbol 1, integers with the Jacobi symbol \(-1\), respectively) modulo \( n \) from \( A \). When \( A = \mathbb{Z}_n^* \), the notation will be simplified to \( QR_n \) \((QNR_n, J_n^+, J_n^-)\), respectively). For the case of an RSA modulus \( n = pq \), where \( p < q \) are odd primes, we will also use the notation \( J_n^\pm \) or \( J_n^{+-} \) for the set of integers in \( \mathbb{Z}_n \) which have the Jacobi symbol equal to 1 modulo \( p \), and \(-1\) modulo \( q \). Vice versa, for \( x \in \mathbb{Z}_n \) with \((x|p) = -1 \) and \((x|q) = +1 \) we will use the notations \( J_n^{\mp} \), or some times \( J_n^{+-} \). By \( J_n^{++} \) \((J_n^{--})\) we will denote the set of integers in \( \mathbb{Z}_n \) which are quadratic residues (quadratic non-residues, respectively) both, modulo \( p \) and \( q \). When \( n \) is a prime, \( QR_n(A) = J_n^+(A) \) and \( QNR_n(A) = J_n^-(A) \).

For a set \( A \), \( a \leftarrow A \) means that \( a \) is uniformly at random chosen from \( A \). If \( \mathcal{A} \) is a probabilistic algorithm, then \( a \leftarrow \mathcal{A} \) means that \( a \) is an output of \( \mathcal{A} \) for some given input.

The asymptotic approach to security makes use of security parameters, denoted by \( \lambda \) in our paper. A positive function \( f(\lambda) \) is called negligible if for any positive polynomial \( poly(\lambda) \) there exists \( n_0 \) such that \( f(\lambda) < 1/poly(\lambda) \), for any \( \lambda \geq n_0 \).

Let \( RSAgen(\lambda) \) be a probabilistic polynomial time algorithm that, given a security parameter \( \lambda \), outputs a triple \((n, p, q)\), where \( n = pq \) is an RSA modulus. The quadratic residuosity (QR) assumption holds for \( RSAgen(\lambda) \) if the distance

\[
|P(D(a, n) = 1 : (n, p, q) \leftarrow RSAgen(\lambda), a \leftarrow QR_n) - P(D(a, n) = 1 : (n, p, q) \leftarrow RSAgen(\lambda), a \leftarrow J_n \setminus QR_n)|,
\]

as a function of \( \lambda \), is negligible for all PPT algorithms \( D \).
We focus on quadratic residues because they are elegant, simple and useful mathematical instruments in creating cryptographic tools like PKE schemes. The hard problems from number theory that quadratic residues generate are well-understood in researchers’ community [11] and used in cryptographic schemes, PRBGs, signatures, IBE - see the remarkable cryptosystem of Cocks [12] - and so on [55, 31, 5].

Cocks’ scheme was a starting point in many individual studies [8, 39, 11, 38, 4, 40, 45]. As it was shown in [25, 8], this scheme doesn’t have the property of hiding the identity of the receiver, thus anonymous versions of Cocks were also developed (see Section 4.1.3). In order to analyze the anonymity problem regarding Cocks’ scheme we elaborated a concrete study on the sets $Y(a+X)$, where $Y \in \{ QR_m, J_m^+, J_m^-, J_m^\pm, QNR_m \}$ and $X$ is a subset of $\mathbb{Z}_n$ with some specifications regarding the Jacobi symbols of its elements (called Jacobi patterns).

The study on sets like $a + QR_p$ where $a$ is an integer and $p$ a prime, begun early, at least in the 50s, by the work of Perron [54]. Damgård [16] and Peralta [53] focused on series of characters $\left( \frac{a+i}{p} \right), \left( \frac{a+i+1}{p} \right), \cdots$ proving their randomness and, therefore,
their utility in constructing random number/bit generators. Later on, Benjamin Justus did some studies on quadratic residues and non-residues in the Goldwasser-Micali framework [42]. He also studied the distribution of quadratic residues and non-residues in arithmetic progressions for a large prime modulus [41].

The case of composite moduli is very useful due to the fact that there are many cryptographic schemes using the context of cyclic groups where the modulus is an RSA integer. These distributions are of great interest also in proving security of cryptosystems based on residues, as we will see in Chapter 4. These results were obtained in a joint work with F.L. Țiplea, S. Iftene, and G. Teșeleanu and were published in [75].

3.1 Counting quadratic residues and non-residues in the set $a + X$

In this section we analyze the distribution of quadratic residues and non-residues in sets of the form $(a + X)$, where $a \in \mathbb{Z}_m^*$, $X$ is one of the sets $\mathbb{Z}_m, \mathbb{Z}_m^*, QR_m$ or $QNR_m$, while the modulus $m$ is either an odd prime, in Section 3.1.1, or an RSA integer, in Section 3.1.2.

We begin the study with the case of $m$ being a prime modulus.

3.1.1 The case of prime moduli

As a starting point we mention a common result that one can find in almost any book on number theory like [13, p.27], [64, 65] or [49]. Given $p$ a prime number, in the set $\mathbb{Z}_p^*$ exactly half of elements are quadratic residues and half are quadratic non-residues. When the modulus is a prime number, the subset of residues coincides with the subset of elements that have the Jacobi symbol equal to 1, while the subset of non-residues is the same with the subset of elements with the Legendre and Jacobi symbols equal to $-1$. For
3.1. Counting QR and QNR in the set $a + X$

the case of RSA moduli a pictorial view can be seen in Figure 3.1 on page 18.

**Proposition 3.1.1.** Given $p$ an odd prime and $a$ an integer coprime to $p$, the following properties hold:

a) $a + \mathbb{Z}_p = \mathbb{Z}_p$ and $|(a + \mathbb{Z}_p)^*| = |\mathbb{Z}_p^*| = p - 1$;

b) $a + \mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{a\}$ and $|(a + \mathbb{Z}_p^*)^*| = p - 2$.

Now, we are interested in computing the cardinality of the sets $a + \text{QR}_p$ and $a + \text{QNR}_p$. When thinking on the sets $a + \text{QR}_p$ and $a + \text{QNR}_p$ we must keep in mind that $-a \mod p$ influences their cardinality. Thus, in the set $a + X$, when $(-a)_p \in X$ we will have $a + (-a) \equiv 0$. Regarding the sets of residues in $\text{QR}_p(A)$, where $A$ is a set like $(a + \text{QR}_p)^*$, in contrast to Perron’s results from 1952 [54], we will not include 0 in the set of residues. This brings the following results.

**Corollary 3.1.1.** Let $p$ be an odd prime and $a \in \mathbb{Z}_p^*$.
When $a \in \text{QR}_p$, we have

$$|\text{QR}_p(a + \mathbb{Z}_p^*)| = \frac{p - 3}{2} \quad \text{and} \quad |\text{QNR}_p(a + \mathbb{Z}_p^*)| = \frac{p - 1}{2},$$

but when $a \in \text{QNR}_p$, then

$$|\text{QR}_p(a + \mathbb{Z}_p^*)| = \frac{p - 1}{2} \quad \text{and} \quad |\text{QNR}_p(a + \mathbb{Z}_p^*)| = \frac{p - 3}{2}.$$

**Proposition 3.1.2.** Let $p$ be an odd prime and $a \in \mathbb{Z}_p^*$.
When $-a \in \text{QR}_p$, then

$$|(a + \text{QR}_p)^*| = \frac{p - 3}{2} \quad \text{and} \quad |(a + \text{QNR}_p)^*| = \frac{p - 1}{2}$$

while when $-a \in \text{QNR}_p$, then

$$|(a + \text{QR}_p)^*| = \frac{p - 1}{2} \quad \text{and} \quad |(a + \text{QNR}_p)^*| = \frac{p - 3}{2}.$$
It is important to notice the following facts. When adding a residue to a fixed integer \( a \) we obtain \( a + QR_p \). In order to get a residue from the addition of \( a \) and \( r \), where \( r \in QR_p \), if we consider \( s \) a square root of \( r \) and \( t \) a square root of the sum \( a + r \), then we have to deal with two residues, \( r \) and \( a + r \). So, \( a + r \equiv_p a + s^2 \equiv_p t^2 \). Starting from here, Perron \cite{54} obtained a very important characterization for the quadratic residues in the set \( a + QR_p \), expressed to the following lemma.

**Lemma 3.1.1** (\cite{54}). Let \( p \) be an odd prime, \( a \) an integer in \( Z_p^* \) and \( r \) a residue modulo \( p \). Then, \( a + r \) is a quadratic residue in \( Z_p^* \) if and only if \( r \) can be written as \( r \equiv_p \frac{1}{4} (u - \frac{a}{u})^2 \), where \( u \in Z_p^* \) and \( u \notin SQRT_p(\pm a) \).

Thus, as Perron observed in \cite{54}, in order to count the quadratic residues of the form \( a + r \), where \( r \) is a residue, one can count the number of incongruent residues in \( Z_p^* \) that can be written as \( (u - a/u)^2 \), with the above restrictions for \( u \).

When \( p \equiv_4 3 \), for an integer \( a \in Z_p^* \), either \( a \) or \( -a \) is a residue modulo \( p \); if \( a \in QR_p \) then \( -a \in QNR_p \) and vice versa. While when \( p \equiv_4 1 \), if \( a \) is a residue, it implies that \( -a \) is a residue too and similarly for the case when \( a \) is a non-residue, it means that \( -a \) is a non-residue too. Using these facts, the following theorem is in order.

**Theorem 3.1.1.** Let \( p \) be an odd prime and \( a \) an integer co-prime to \( p \), then

\[
|QR_p(a + QR_p)| = \frac{|Z_p^* \setminus SQRT_p(\pm a)|}{4}.
\]

**Corollary 3.1.2.** When \( p \) is an odd prime, \( p = 4k + i \), where \( i \in \{1, 3\} \) and \( a \in Z_p^* \), then

\[
|QR_p(a + QR_p)| = \begin{cases} 
  k - 1, & \text{if } a \in QR_p \text{ and } i = 1 \\
  k, & \text{otherwise}
\end{cases}
\]
3.1. Counting QR and QNR in the set $a + X$

and

$$|QR_p(a + QR_p)| = \begin{cases} k + 1, & \text{if } a \in QR_p \text{ and } i = 3 \\ k, & \text{otherwise.} \end{cases}$$

For the sets $QR_p(a + QNR_p)$ and $QNR_p(a + QNR_p)$ we have the following results.

**Corollary 3.1.3.** Let $p \equiv 4 \, i$ be an odd prime, with $i \in \{1, 3\}$ and $a \in \mathbb{Z}_p^*$, then

$$|QR_p(a + QNR_p)| = \begin{cases} \frac{p - 3}{4} + 1, & \text{if } i = 3 \text{ and } a \in QNR_p \\ \frac{p - i}{4}, & \text{otherwise} \end{cases}$$

and

$$|QNR_p(a + QNR_p)| = \begin{cases} \frac{p - 1}{4} - 1, & \text{if } i = 1 \text{ and } a \in QNR_p \\ \frac{p - i}{4}, & \text{otherwise.} \end{cases}$$

### 3.1.2 The case of RSA moduli

Quite frequently in cryptography RSA moduli are used (the product of two odd primes). Our goal is to compute the cardinalities of some sets of a given pattern of the Jacobi symbol. The distribution of residues and non-residues in the set $\mathbb{Z}_n^*$ is used to create or analyze cryptosystems, to construct random bit generators and so on. If we can find probability distributions that are statistically indistinguishable then this can be a great mathematical instrument which can be used to prove the security of a scheme or some cryptographic properties, such as anonymity.

If we have an integer $x$ in $\mathbb{Z}_n^*$, we can get its corresponding values in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ by reducing $x$ modulo $p$, modulo $q$ respectively, and obtaining unique values. This is available also vice versa, from
On the distribution of $QR$

$$Z^*_n$$

$QR_n = J_{n}^{++}$

$J_{n} - QR_n = J_{n}^{--}$

$J_{n}^{\pm}$

$J_{n}^{\mp}$

$QNR_n$

$(x)_p$ and $(x)_q$ to $(x)_n$ due to the Chinese Reminder theorem (CRT) and the well known isomorphism $f$ from: $Z^*_n \rightarrow Z^*_p \times Z^*_q$, where $f(x) = (x \mod p, x \mod q), \forall x \in Z^*_n$. Although it is a simple result, it has so many important applications, and we will use it throughout this section in order to obtain new results combining sets and computing their cardinalities, as we can see in the next theorem.

**Theorem 3.1.2.** Let $n$ be an RSA modulus, $n = pq$, $a \in Z^*_p$ and the bijection $f : Z^*_n \rightarrow Z^*_p \times Z^*_q$, given by $f(x) = ((x)_p, (x)_q)$, then $f$ maps the following sets as follows:

1. $(a + Z^*_n)^* \rightarrow ((a)_p + Z^*_p)^* \times ((a)_q + Z^*_q)^*$;
2. $(a + QR_n)^* \rightarrow ((a)_p + QR_p)^* \times ((a)_q + QR_q)^*$;
3. $(a + J_{n}^{+} \setminus QR_n)^* \rightarrow ((a)_p + QNR_p)^* \times ((a)_q + QNR_q)^*$;
4. $(a + J_{n}^{\pm})^* \rightarrow ((a)_p + QR_p)^* \times ((a)_q + QNR_q)^*$;
5. $(a + J_{n}^{\mp})^* \rightarrow ((a)_p + QNR_p)^* \times ((a)_q + QR_q)^*$.
3.1. Counting QR and QNR in the set $a + X$

Figure 3.2: Jacobi patterns on $(a + Z_n^*)^*$

We are now able to count the elements from the sets in Theorem 3.1.2.

**Corollary 3.1.4.** Let $p, q$ be two odd primes, $n = pq$ and $a \in Z_n^*$, then

1. $|(a + Z_n^*)^*| = (p - 2)(q - 2)$;

2. $|(a + QR_n)^*| = \frac{(p - 2 - (-a|p))(q - 2 - (-a|q))}{4}$;

3. $|(a + J_n^\pm QR_n)^*| = \frac{(p - 2 + (-a|p))(q - 2 + (-a|q))}{4}$;

4. $|(a + J_n^\mp)^*| = \frac{(p - 2 - (-a|p))(q - 2 + (-a|q))}{4}$;

5. $|(a + J_n^\pm)^*| = \frac{(p - 2 + (-a|p))(q - 2 - (-a|q))}{4}$;

6. $|(a + J_n^+)^*| = \frac{(p - 2)(q - 2) - (-a|p)(-a|q)}{2}$;

7. $|(a + J_n^-)^*| = \frac{(p - 2)(q - 2) + (-a|p)(-a|q)}{2}$;

8. $|(a + QNR_n)^*| = \frac{3(p - 2)(q - 2) + (-a|p)(q - 2)}{4} + \frac{(-a|q)(p - 2) - (-a|p)(-a|q)}{4}.$
Another result which helps to partition the set \((a + \mathbb{Z}_n^*)\) is obtained also through some mappings that the bijection \(f\) does.

**Theorem 3.1.3.** Let \(n\) be an RSA modulus from the product of the two primes \(p, q\) and \(a \in \mathbb{Z}_n^*\), then the bijection \(f\) in Theorem 3.1.2 maps the sets below as follows:

\[
\begin{align*}
a) & \quad QR_n(a + \mathbb{Z}_n^*) \text{ onto } QR_p((a)_p + \mathbb{Z}_p^*) \times QR_q((a)_q + \mathbb{Z}_q^*); \\
b) & \quad (J_n^+ \setminus QR_n)(a + \mathbb{Z}_n^*) \text{ onto } QNR_p((a)_p + \mathbb{Z}_p^*) \times QNR_q((a)_q + \mathbb{Z}_q^*); \\
c) & \quad J_n^\pm(a + \mathbb{Z}_n^*) \text{ onto } QR_p((a)_p + \mathbb{Z}_p^*) \times QNR_q((a)_q + \mathbb{Z}_q^*); \\
d) & \quad J_n^\mp(a + \mathbb{Z}_n^*) \text{ onto } QNR_p((a)_p + \mathbb{Z}_p^*) \times QR_q((a)_q + \mathbb{Z}_q^*). \\
\end{align*}
\]

Now we can compute the cardinals of the sets mapped by the theorem as it is stated in the next corollary.

**Corollary 3.1.5.** Let \(p < q\) be two odd primes, \(n = pq\) and \(a \in \mathbb{Z}_n^*\), then

\[
\begin{align*}
1. \quad |QR_n(a + \mathbb{Z}_n^*)| &= \frac{(p - 2 - (a | p))(q - 2 - (a | q))}{4}; \\
2. \quad |(J_n^+ \setminus QR_n)(a + \mathbb{Z}_n^*)| &= \frac{(p - 2 + (a | p))(q - 2 + (a | q))}{4}; \\
3. \quad |J_n^\pm(a + \mathbb{Z}_n^*)| &= \frac{(p - 2 - (a | p))(q - 2 + (a | q))}{4}; \\
4. \quad |J_n^\mp(a + \mathbb{Z}_n^*)| &= \frac{(p - 2 + (a | p))(q - 2 - (a | q))}{4}; \\
5. \quad |J_n^+(a + \mathbb{Z}_n^*)| &= \frac{(p - 2)(q - 2) + (a | p)(a | q)}{2}; \\
6. \quad |J_n^-(a + \mathbb{Z}_n^*)| &= \frac{(p - 2)(q - 2) - (a | p)(a | q)}{2};
\end{align*}
\]
3.1. Counting QR and QNR in the set $a + X$

7. $|QNR_n(a + \mathbb{Z}_n^*)| = \frac{3(p - 2)(q - 2) + (a|p)(q - 2)}{4} + \frac{(a|q)(p - 2) - (a|p)(a|q)}{4}$.

The next set we discuss is $A = (a + QR_n)$. We will focus on the subsets $QR_n(A)$, $(J_n^+(\mathbb{Z}_n^*))$, $J_n^\pm(A)$ and $J_n^\mp(A)$. Due to the same isomorphism $f$ in Theorem 3.1.2 these sets are mapped correspondingly as it is shown in the theorem below.

**Theorem 3.1.4.** Let $n = pq$ be an RSA modulus and $a \in \mathbb{Z}_n^*$, then the function $f$ in Theorem 3.1.2 maps the partitioning sets of $A = (a + QR_n)$ as follows:

- a) $QR_n(A)$ onto $QR_p((a)_p + QR_p) \times QR_q((a)_q + QR_q)$;
- b) $(J_n^+(\mathbb{Z}_n^*))$ onto $QNR_p((a)_p + QR_p) \times QNR_q((a)_q + QR_q)$;
- c) $J_n^\pm(A)$ onto $QR_p((a)_p + QR_p) \times QNR_q((a)_q + QR_q)$;
- d) $J_n^\mp(A)$ onto $QNR_p((a)_p + QR_p) \times QR_q((a)_q + QR_q)$.

We are now able to establish in the next corollary how many elements each set discussed above has. In order to avoid too many cases, we will use the notation in [75].

**Notation 3.1.1 ([75]).** Let $p$ be an odd prime, $a$ an integer coprime with $p$, and $i = p \mod 4 \in \{1, 3\}$, then we define:

\[
\tau^i_{p,a} = \begin{cases} 
1, & \text{if } (p)_4 = i \text{ and } (a)_p \in QR_p \\
0, & \text{otherwise}
\end{cases}
\]

and

\[
\bar{\tau}^i_{p,a} = \begin{cases} 
1, & \text{if } (p)_4 = i \text{ and } (a)_p \in QNR_p \\
0, & \text{otherwise}.
\end{cases}
\]

**Corollary 3.1.6.** Given an RSA modulus $n = pq$, with $s = p \div 4$, $t = q \div 4$, and an integer $a \in \mathbb{Z}_n^*$, then
1. \( |QR_n(a + QR_n)| = (s - \tau_p^1)(t - \tau_q^1); \)

2. \( |(J_n^+ \setminus QR_n)(a + QR_n)| = (s + \tau_p^3)(t + \tau_q^3); \)

3. \( |J_n^+(a + QR_n)| = (s - \tau_p^1)(t + \tau_q^3); \)

4. \( |J_n^+(a + QR_n)| = (s + \tau_p^3)(t - \tau_q^1); \)

5. \( |J_n^+(a + QR_n)| = 2st + s(\tau_q^3 - \tau_p^1) + t(\tau_p^3 - \tau_p^1) + \tau_p^1 \tau_q^1 + \tau_p^3 \tau_q^3; \)

6. \( |J_n^-(a + QR_n)| = 2st + s(\tau_q^3 - \tau_p^1) + t(\tau_p^3 - \tau_p^1) - \tau_p^1 \tau_q^1 - \tau_p^3 \tau_q^3; \)

7. \( |QNR_n(a + QR_n)| = 3st + s(2\tau_q^3 - \tau_p^1) + t(2\tau_p^3 - \tau_p^1) - \tau_p^1 \tau_q^3 - \tau_p^3 \tau_q^1 + \tau_p^3 \tau_q^3. \)

We will partition now the set \((a + J_n^+ \setminus QR_n)\) in the next theorem.

**Theorem 3.1.5.** Let \( n \) be an RSA modulus with \( n = pq \), \( a \in \mathbb{Z}_n^* \) and the set \( A = (a + J_n^+ \setminus QR_n) \), then the bijection in Theorem 3.1.2 maps the four sets as follows:

a) \( QR_n(A) \) onto \( QR_p((a)_p + QNR_p) \times QR_q((a)_q + QNR_q); \)

b) \( (J_n^+ \setminus QR_n)(A) \) onto \( QNR_p((a)_p + QNR_p) \times QNR_q((a)_q + QNR_q); \)

c) \( J_n^+(A) \) onto \( QR_p((a)_p + QNR_p) \times QNR_q((a)_q + QNR_q); \)

d) \( J_n^+(A) \) onto \( QNR_p((a)_p + QNR_p) \times QNR_q((a)_q + QNR_q). \)

The corresponding cardinals for the sets in Theorem 3.1.5 are detailed in the corollary below.

**Corollary 3.1.7.** Let \( p, q \) be two odd primes, \( n = pq \) an RSA modulus, \( s = p \text{ div } 4 \), \( t = q \text{ div } 4 \), \( a \in \mathbb{Z}_n^* \), and \( A = (a + J_n^+ \setminus QR_n) \), then
3.1. Counting QR and QNR in the set $a + X$

1. $|QR_n(A)| = (s + \bar{\tau}_{p,a}^3)(t + \bar{\tau}_{q,a}^3)$;

2. $|(J_n^+ \setminus QR_n)(A)| = (s - \bar{\tau}_{p,a}^1)(t - \bar{\tau}_{q,a}^1)$

3. $|J_n^+(A)| = (s + \bar{\tau}_{p,a}^3)(t - \bar{\tau}_{q,a}^1)$;

4. $|J_n^-(A)| = (s - \bar{\tau}_{p,a}^1)(t + \bar{\tau}_{q,a}^3)$;

5. $|J_n^+(A)| = 2st + s(\bar{\tau}_{q,a}^3 - \bar{\tau}_{q,a}^1) + t(\bar{\tau}_{p,a}^3 - \bar{\tau}_{p,a}^1) + \bar{\tau}_{p,a}^3 \bar{\tau}_{q,a} - \bar{\tau}_{p,a} \bar{\tau}_{q,a}^1$;

6. $|J_n^-(A)| = 2st + s(\bar{\tau}_{q,a}^3 - \bar{\tau}_{q,a}^1) + t(\bar{\tau}_{p,a}^3 - \bar{\tau}_{p,a}^1) - \bar{\tau}_{p,a}^1 \bar{\tau}_{q,a}^3 - \bar{\tau}_{p,a} \bar{\tau}_{q,a}^1$;

7. $|QNR_n(A)| = 3st + s(\bar{\tau}_{q,a}^3 - 2\bar{\tau}_{q,a}^1) + t(\bar{\tau}_{p,a}^3 - 2\bar{\tau}_{p,a}^1) - \bar{\tau}_{p,a}^3 \bar{\tau}_{q,a} - \bar{\tau}_{p,a} \bar{\tau}_{q,a}^1$

The next result partitions the set $a + J^\pm$.

**Theorem 3.1.6.** Let $p < q$ be two odd primes, $n = pq$, an RSA modulus, $a \in \mathbb{Z}_n^*$ and $A = (a + J_n^\pm)$. Then the isomorphism $f$ in Theorem 3.1.2 maps the following sets as we can see below:

a) $QR_n(A)$ onto $QR_p((a)_p + QR_p) \times QR_q((a)_q + QNR_q)$;

b) $(J_n^+ \setminus QR_n)(A)$ onto $QNR_p((a)_p + QR_p) \times QNR_q((a)_q + QNR_q)$;

c) $J_n^+(A)$ onto $QR_p((a)_p + QR_p) \times QNR_q((a)_q + QNR_q)$;

d) $J_n^-(A)$ onto $QNR_p((a)_p + QR_p) \times QR_q((a)_q + QNR_q)$.

The number of integers in the sets from the previous theorem are counted in the next corollary.

**Corollary 3.1.8.** Let $n = pq$ be an RSA modulus, $s = p \text{ div } 4$, $t = q \text{ div } 4$, $a \in \mathbb{Z}_n^*$, and $A = (a + J_n^\pm)$, then

1. $|QR_n(A)| = (s - \tau_{p,a}^1)(t + \bar{\tau}_{q,a}^3)$;

2. $|(J_n^+ \setminus QR_n)(A)| = (s + \tau_{p,a}^3)(t - \bar{\tau}_{q,a}^1)$;

3. $|J_n^+(A)| = (s - \tau_{p,a}^1)(t - \bar{\tau}_{q,a}^1)$;
Theorem 3.1.7. Let $A \in \mathbb{Z}_n^*$. Then the bijection $f$ in Theorem 3.1.2 maps the sets that partition $A = (a + J_n^\pm)$ as follows:

a) $QR_n(A)$ onto $QR_p((a)_p + QNR_p) \times QR_q((a)_q + QR_q)$;

b) $(J_n^+ \setminus QR_n)(A)$ onto $QNR_p((a)_p + QNR_p) \times QNR_q((a)_q + QR_q)$;

c) $J_n^\pm(A)$ onto $QR_p((a)_p + QNR_p) \times QNR_q((a)_q + QR_q)$;

d) $J_n^\mp(A)$ onto $QNR_p((a)_p + QNR_p) \times QNR_q((a)_q + QR_q)$.

Corollary 3.1.9. Let $n = pq$, $s = p \div 4$, $t = q \div 4$, $a \in \mathbb{Z}_n^*$, and $A = (a + J_n^\pm)$, then,

1. $|QR_n(A)| = (s + \tau_{p,a}^3)(t - \tau_{q,a}^1)$;

2. $|(J_n^+ \setminus QR_n)(A)| = (s - \tau_{p,a}^1)(t + \tau_{q,a}^3)$;

3. $|J_n^\pm(A)| = (s + \tau_{p,a}^3)(t + \tau_{q,a}^3)$;

4. $|J_n^\mp(A)| = (s - \tau_{p,a}^1)(t + \tau_{q,a}^3)$;

5. $|J_n^+(A)| = 2st + s(\tau_{q,a}^3 - \tau_{q,a}^1) + t(\tau_{p,a}^3 - \tau_{p,a}^1 - \tau_{p,a}^1 \tau_{q,a}^3 - \tau_{p,a}^1 \tau_{q,a}^1)$;

6. $|J_n^-(A)| = 2st + s(\tau_{q,a}^3 - \tau_{q,a}^1) + t(\tau_{p,a}^3 - \tau_{p,a}^1) + \tau_{p,a}^1 \tau_{q,a}^3 + \tau_{p,a}^1 \tau_{q,a}^3$;

7. $|QNR_n(A)| = 3st + s(2\tau_{q,a}^3 - \tau_{q,a}^1) + t(\tau_{p,a}^3 - 2\tau_{p,a}^1) + \tau_{p,a}^3 \tau_{q,a}^3 + \tau_{p,a}^1 \tau_{q,a}^3 - \tau_{p,a}^1 \tau_{q,a}^3$.

The last set we discuss in this section is $(a + J_n^\pm)$.
3.2. Computing probabilities on sets $Y(a + X)$

$$ (a + \mathbb{Z}_n^*) $$

<table>
<thead>
<tr>
<th>$a + QR_n$</th>
<th>$a + (J_n^+ \setminus QR_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a + J_n^+$</td>
<td>$a + J_n^+$</td>
</tr>
</tbody>
</table>

Figure 3.3: The sets partitioning $(a + \mathbb{Z}_n^*)$

Now that we have done with the computation of all the cardinalities of the set partitions of $(a + \mathbb{Z}_n^*)$ together with the sets of different patterns of the Jacobi symbol regarding $p$ and $q$ respectively, we can show in the next section how to compute a probability that depends on the sets mentioned above. For a general view of these partitionings, see Figure 3.3.

3.2 Computing probabilities on sets $Y(a + X)$

QR are important specially in mathematics but also in the field of cryptography. Usually we “define security and analyze schemes using probabilistic experiments involving algorithms making randomized choices” [43, p.25]. Thus we are interested to know how to compute probabilities on sets of different Jacobi patterns. In this section we give a few examples of computing such probabilities, and in the next chapter we will use some of them.

Corollary 3.2.1. Let $n = pq$ be an RSA modulus and $a \in \mathbb{Z}_n^*$. Then:
On the distribution of QR

(1) \( P(x \in QR_n : x \leftarrow (a + Z_n^*)^*) = \begin{cases} \frac{1}{4} + \mathcal{O}\left(\frac{1}{n}\right), & \text{if } a \in J_n^+ \setminus QR_n \\ \frac{1}{4} - \mathcal{O}\left(\frac{1}{n}\right), & \text{otherwise.} \end{cases} \)

(2) \( P(x \in J_n^+ \setminus QR_n : x \leftarrow (a + Z_n^*)^*) = \begin{cases} \frac{1}{4} + \mathcal{O}\left(\frac{1}{n}\right), & \text{if } a \in J_n^+ \setminus QR_n \\ \frac{1}{4} - \mathcal{O}\left(\frac{1}{n}\right), & \text{otherwise.} \end{cases} \)

(3) \( P(x \in J_n^\pm : x \leftarrow (a + Z_n^*)^*) = \begin{cases} \frac{1}{4} + \mathcal{O}\left(\frac{1}{n}\right), & \text{if } a \in J_n^\mp \\ \frac{1}{4} - \mathcal{O}\left(\frac{1}{n}\right), & \text{otherwise.} \end{cases} \)

The distribution of different patterns of Jacobi symbol, especially when the modulus is an RSA moduli, a product of two odd primes, is of great interest not just in mathematics but also in cryptography. In the next chapter we will see how one can use the results in this chapter in order to prove different statements and even how one can avoid using the quadratic residuosity assumption and obtain better results in proving issues regarding security of schemes.
Chapter 4

Applications of QR to IBE

A general IBE scheme, according to [7], has four PPT algorithms. The first one is \textsc{Setup}(\lambda), which, for a security parameter \( \lambda \), outputs the public parameters, \( PP \), and the master secret key, \( msk \). The \textsc{KeyGen}(\( PP, msk, ID \)) algorithm outputs the secret key, \( sk_{ID} \), corresponding to a given identity \( ID \). The following two algorithms are \textsc{Encryption}(\( PP, m \)) and \textsc{Decryption}(\( sk_{ID}, c \)). They encrypt a message \( m \) for a given identity, \( ID \), and get the cryptotext \( c \), and decrypt, respectively, the cryptotext \( c \) using the secret key corresponding to the same identity, \( ID \), of the receiver.

We will present now the first QR-based IBE scheme, due to Cocks [12].

4.1 Cocks’ IBE scheme

Cocks’ construction, Algorithm 1 on page 29, encrypts one bit at a time and works fine for short messages [12]. Although it has a good running time, its bottleneck is the ciphertext expansion, \( O(2 \log n) \) - one bit of cryptotext is encrypted by two integers in \( \mathbb{Z}_n \). The security of the scheme is presented by the next theorem.

\textbf{Theorem 4.1.1} ([12, 30]). The Cocks IBE scheme is IND-CPA secure in the ROM, assuming that QRA holds for the \textsc{RSAgen}.
Chapter 4. Applications of QR to IBE

The ciphertexts outputted by Cocks’ scheme have the following form: $t + at^{-1}$, where $a, t \in \mathbb{Z}_n^*$. In what follows we will analyze the set of possible Cocks ciphertexts, aiming to see why Cocks’ IBE scheme is not anonymous and why Galbraith’s test (GT) works.

### 4.1.1 Cocks’ IBE Ciphertexts

For an RSA modulus $n$ and $a \in J_n^+$ we know that Cocks’ IBE ciphertexts have the form $t + at^{-1} \mod n$. We will denote the set of such elements $C_n(a)$.

If we rewrite a Cocks ciphertext for fixed values of $c, a \in \mathbb{Z}_n^*$ we obtain the general form of a degree two equation in the unknown $t$, i.e. $c \equiv t + at^{-1}/t \iff ct \equiv t^2 + a$, which is equivalent to:

$$t^2 - ct + a = 0 \mod n \quad (4.1)$$

**Theorem 4.1.2.** Let $a \in J_n^+, c \in \mathbb{Z}_n$, and $C_n(a) = \{t + at^{-1} \mod n|t \in \mathbb{Z}_n^*\}$. Then $c \in C_n(a)$ if and only if the discriminant $\Delta$ of Equation (4.1) is either 0 or a quadratic residue modulo $n$. Moreover, $c$ can be 0 and also in $C_n(a)$ if and only if $-a$ is a quadratic residue.

This theorem will be useful in further computations and in Section 4.1.2 where we will need in addition the exact cardinal of the set of Cocks cryptotexts. Thus, in order to get it, we will compute the cardinal for the case of a prime modulus $p$, $C_p(a)$, using the results in Chapter 3 and compute the same cardinal for $C_n(a)$, where $n$ is an RSA modulus, based on the bijection $f$ in Section 3.1.2 thereafter.

If we analyze the set $C_p^*(a) = C_p(a) \cap \mathbb{Z}_p^*$ in the view of Theorem 4.1.2 for a prime modulus $p$, we can express its partitioning like in Figure 4.1 and define it below:

\[
\begin{align*}
\Delta \equiv_p 0 &: \quad C_p^0(a) = \{c \in \mathbb{Z}_n^*|(c^2 - 4a|n) = 0\} \\
\Delta \in QR_p &: \quad C_p^1(a) = \{c \in \mathbb{Z}_n^*|(c^2 - 4a|n) = 1\}
\end{align*}
\]
Algorithm 1 Cocks’ IBE scheme

```plaintext
procedure Setup(λ)
    (p, q) ← RSA\textsubscript{gen}(λ);
    n = pq;
    e ← J^n_+ \setminus QR_n;
    h : \{0,1\}^* ← J^n_+;
    PP = (n, e, h);
    msk = (p, q);
    return (PP, msk).
end procedure

procedure KeyGen(msk, ID)
    a = h(ID);
    if a ∈ QNR\textsubscript{n} then
        a = ea;
        r = a^{(n+5-(p+q))/8};
    end if
    return r.
end procedure

procedure Encrypt(PP, ID, m)
    a = h(ID);
    t_1, t_2 ← Z^*_n such that (t_1|n) = (t_2|n) = (m|n)
    c_1 = t_1 + at_1^{-1};
    c_2 = t_2 + eat_2^{-1};
    return (c_1, c_2).
end procedure

procedure Decrypt(PP, r, (c_1, c_2))
    if r^2 \equiv_n h(ID) then
        c = c_1;
    else c = c_2;
    end if
    m = \left(\frac{c+2r}{n}\right);
    return m.
end procedure
```

\( \triangleright \) such that \( p \equiv_4 q \equiv_4 3 \)

\( \triangleright \) for example \( e \equiv_n -1 \)

\( \triangleright \) hash func. that maps IDs into \( J^n_+ \)
Thus, for an integer $c$ to be in $C^0_p(a)$, it takes an $a \in QR_p$. Now we are able to do the computations for the cardinals of these sets: $|C^0_p(a)|$, $|C^1_p(a)|$, of their union, $|C^*_p(a)|$, and of the set that contains, in addition, $c \equiv_p 0$, i.e. the cardinal of the set $C_p(a)$.

**Corollary 4.1.1.** Let $C^0_p(a)$, $C^1_p(a)$, $C^*_p(a)$ and $C_p(a)$ defined as above, let $p$ be an odd prime, $a \in Z^*_n$ and $k = p \text{ div } 4$. Then

1. $|C^0_p(a)| = 2(\tau^1_{p,a} + \tau^3_{p,a})$
2. $|C^1_p(a)| = 2|QR_p(a + QR_p)| = 2(k - \tau^1_{p,a})$
3. $|C^*_p(a)| = 2(k + \tau^3_{p,a})$
4. $|C_p(a)| = 2(k + \tau^3_{p,a}) + \tau^1_{p,a} + \tilde{\tau}^3_{p,a}$.

In the case of an RSA modulus $n$, for a given $a \in Z^*_n$, function $f$ in Section 3.1.2 will be used in order to analyze the set $C^*_n(a)$. Let $\Delta$ be the determinant of Equation (4.1) and let us denote by $C^{i,j}_n(a) = \{c \in Z^*_n|((\Delta)_p|p) = i, ((\Delta)_q|q) = j\}$. Thus, we can easily go from two odd distinct primes $p$ and $q$ to the RSA modulus $n = pq$ by $f$ in the next theorem.

**Theorem 4.1.3.** Let $p, q$ be two distinct odd primes, $n = pq$ an RSA modulus, and $a \in Z^*_n$. Then, the isomorphism $f$ in Section 3.1.2 does the following mapping:

a) $C^*_n(a)$ onto $C^*_p((a)_p) \times C^*_q((a)_q)$;
b) $C^{0,0}_n(a)$ onto $C^0_p((a)_p) \times C^0_q((a)_q)$;
c) $C^{0,1}_n(a)$ onto $C^0_p((a)_p) \times C^1_q((a)_q)$;
d) $C^{1,0}_n(a)$ onto $C^1_p((a)_p) \times C^0_q((a)_q)$;
4.1. Cocks’ IBE scheme

e) \( C_n^{1,1}(a) \) onto \( C_p^1((a)_p) \times C_q^1((a)_q) \);

f) \( C_n(a) \) onto \( C_p((a)_p) \times C_q((a)_q) \).

**Corollary 4.1.2.** Let \( n \) be an RSA modulus and \( a \in \mathbb{Z}_n^* \). Then the set \( C^*_n(a) \) is a union of the sets \( C_n^{0,0}, C_n^{1,0}, C_n^{0,1} \) and \( C_n^{1,1} \).

We are now in a position to compute the cardinal of each set in Theorem 4.1.3.

**Corollary 4.1.3.** Let \( p, q \) be two odd primes, \( n = pq \) an RSA modulus, \( a \in \mathbb{Z}_n^* \), \( k_1 = p \text{ div} 4 \), and \( k_2 = q \text{ div} 4 \). Then,

1. \( |C_n^*(a)| = |C_p^*(a)_p| \cdot |C_q^*(a)_q| \);

2. \( |C_n^{0,0}(a)| = 4(\tau^1_{p,a} + \tau^3_{p,a})(\tau^1_{q,a} + \tau^3_{q,a}) \);

3. \( |C_n^{0,1}(a)| = 4(\tau^1_{p,a} + \tau^3_{p,a})(k_2 - \tau^1_{q,a}) \);

4. \( |C_n^{1,0}(a)| = 4(\tau^1_{q,a} + \tau^3_{q,a})(k_1 - \tau^1_{p,a}) \);

5. \( |C_n^{1,1}(a)| = 4|QR_n(a + QR_n)| = 4(k_1 - \tau^1_{p,a})(k_2 - \tau^1_{q,a}) \);

6. \( |C_n(a)| = |C_p((a)_p)| \cdot |C_q((a)_q)| \).

Once we have these results we can use them in order to check the anonymity of Cocks’ IBE cryptotexts.

### 4.1.2 Galbraith’s test

When someone encrypts a message using an IBE scheme, an identity is used. Sometimes it is important that this identity, of the intended receiver, remains anonymous. Let \( ID_1, ID_2 \in J_n^+ \) be the correspondent identities of two receivers. We say that an IBE scheme is anonymous (in the sense of [2]) if, when a third party analyses some cryptotexts encrypted for one of the two identities, \( ID_1 \) or \( ID_2 \), he will not be able to say whether the receiver has \( ID_1 \) or \( ID_2 \) other than with negligible probability.
Since 2004, when it was shown in [25, 6], due to Galbraith’s test (GT), that Cocks’ IBE scheme is not anonymous, a couple of anonymous variants were proposed, more or less efficient [14, 11611, 11, 40]. Boneh et al. presented GT in [6], which helps one to establish with overwhelming probability if a cryptotext was encrypted using a given identity or not. In this section we used the results in Chapter 3 in order to give a rigorous proof of GT. The GT algorithm, as it was presented in [1], takes as input a modulus $n$, a cryptotext $c$ and an identity $a$, and returns $\pm 1$ with the following meaning:

$$
\begin{cases}
+1 : & c \text{ is a Cocks cryptotext and there is a probability of } \frac{1}{2} \text{ that it was encrypted for the identity } a; \\
-1 : & c \text{ was not encrypted for the ID } a \text{ (for sure)}.
\end{cases}
$$

We deeply proved why this result is possible and presented the exact cardinals involved in these computations.

**Theorem 4.1.4.** Let $n$ be an RSA modulus, the product of odd primes $p$ and $q$, and $a \in \mathbb{Z}_n^*$. Then the set $G_n(a)$ is partitioned by the sets $C_{n}^{1,1}(a)$ and $C_{n}^{-1,-1}(a)$, and its cardinal is: $4 |QR_n(a+J_n^+)|$.

So, we can test $c$ regarding $a$ in order to find out if $c \in C_n^*(a)$ using Galbraith’s test described in Algorithm 2 and we can compute the exact probability that $c \in C_n^*(a)$, when $(\Delta | n) = +1$, using the results in the previous section:

$$
P(c \in C_n^*(a) : c \leftarrow G_n(a)) = \frac{|C_{n}^{1,1}(a)|}{|G_n(a)|} = \frac{1}{2} - O\left(\frac{1}{\sqrt{n}}\right).
$$
4.1. Cocks’ IBE scheme

Algorithm 2: Galbraith test

Input: \((a, c, n)\) \(\triangleright a \in J_n^+, n \leftarrow RSAGen(\lambda), c - a\) Cocks cryptotext

Output: yes or no

\[\Delta = c^2 - 4a;\]

if \((\Delta \mod n) = 1\) or if \((\Delta \mod n) = 0\) then

return yes \(\triangleright\) meaning: \(P[c \in C_n(a)] = 1/2\)

else

return no \(\triangleright\) meaning: \(c \notin C_n(a)\), for sure

end if

In order to establish the identity of a set of cryptotexts encrypted for the same identity, one can repeat the GT for each cryptotext in this set in order to increase the probability to give a correct answer. Considering that Cocks’ IBE scheme encrypts one bit at a time and that the sender usually encrypts a sequence of bits and not a single bit, we have a set of cryptotexts encrypted using the same identity which can be used in order to identify the receiver.

4.1.3 Anonymous Cocks’ schemes

In 2005, Hayashi and Tanaka extended the anonymity notion, as it was described in the beginning of Section 4.1.2, to a more general case called universal anonymity \[35\]. This property allows anyone, not only the encryptor, to anonymize a ciphertext using the public key of the receiver. In this case the anonymization process is distinct from the encryption one, being a stand alone action.

Cocks’ scheme is not anonymous \[6\], but several variants were proposed by different authors. G. Di Crescenzo and V. Saraswat
Joye’s IBE scheme

The most efficient anonymized variant of Cocks’ IBE scheme is the one proposed by Joye \cite{joye2016}. In this section we present his scheme in a simplified and more direct way, regarding the analysis of Cocks’ ciphertexts \cite{cocks1999}. In order to accomplish this, we appeal to the results in Chapter 3.

Using the analysis of Cocks’ IBE ciphertexts in Section 4.1.1 and taking into consideration that most of them are in $C_n^{1,1}(a)$, we may figure out that, if the sender slightly modifies some of the ciphertexts, randomly choosing which one to modify and which to let it as it is, and if we also find a method such that the receiver be the only one who knows which one was modified and how to deanonymize them, then we will get an anonymized scheme. Thus, given $c \in C_n^{1,1}(a)$, the sender can transform it into a $c'$ such that $GT_{n,a}(c') = \pm 1$ by choosing a fixed $d$ such that $GT_{n,a}(d) = -1$, and then using a binary operation $\circ$ on $\mathbb{Z}_n^*$ such that

$$GT_{n,a}(c \circ d) = GT_{n,a}(c) \cdot GT_{n,a}(d).$$

So, the modified ciphertext may be $c' = c \circ d$. Now we have to find a way such that the sender and the receiver alone might know which ciphertext has to be deanonymized. In order to accomplish this, Galbraith’s test can be used, as we will see below.

We will denote by $\circ$ the following operation:

$$u \circ v = \frac{uv + 4a}{u + v} \mod n,$$

for all $u, v \in \mathbb{Z}_n^*$ with $(u + v, n) = 1$.

This operation has a set of properties, as we can see in this proposition, which ensures the correctness of the scheme.
Proposition 4.1.1. Let $u,v,w \in \mathbb{Z}_n^*$ and $a \in J_n^+$. Then:

1. Associativity. $\circ$ is associative whenever it is defined, so $u \circ (v \circ w) = (u \circ v) \circ w$.

2. Even if $v \circ (-v)$ is not defined, we have $(u \circ v) \circ (-v) = u$ whenever $(u + v, n) = 1$ and $(v^2 - 4a, n) = 1$.

3. When $u \circ v$ is defined, $GT_{n,a}(u \circ v) = GT_{n,a}(u) \cdot GT_{n,a}(v)$.

4. $u \circ u \in G_n(a)$.

In order to identify if a ciphertext $c^*$ has been modified or not, we can use Proposition 4.1.1(4), which says that $u \circ u$ always passes Galbraith’s test and Proposition 4.1.1(3) which, assuming that $u \circ v$ is defined, sets that $u \circ v$ passes Galbraith’s test if and only if Galbraith’s test returns the same result for both $u$ and $v$.

Having this operation together with the properties above, we have all set in order to describe the anonymized scheme, as we can see in Algorithm 3.

Regarding the security of the scheme we have the following theorem.

Theorem 4.1.5 ([51]). Cocks’ AnonIBE scheme is ANON-IND-ID-CPA secure in the random oracle model under the QRA.

4.2 BGH’s IBE scheme

There is a remarkable result obtained in 2007 whose goal was to decrease the ciphertext expansion in Cocks’ scheme [12] - the Boneh-Gentry-Hamburg’s (BGH) scheme [8]. This goal was met but the scheme remains unpractical because its time complexity becomes quartic in the security parameter per message bit.

They managed to encrypt a bit of plaintext by multiplying it by a Jacobi symbol whose value can be computed only by the
Algorithm 3 Cocks' AnonIBE scheme

```latex
\textbf{procedure} \textsc{Setup}(\lambda):
\begin{align*}
PP &= (n, e, d, h) \\
\triangleright \text{where } n \text{ and } e \text{ are as in Cocks' IBE scheme} \\
d &\leftarrow \mathbb{Z}_n^* \text{ and } h : \{0, 1\}^* \rightarrow J_n^+ \text{ are chosen so that} \\
GT_{n,a}(d) &= -1 = GT_{n,ea}(d), \text{ for any output } a \text{ of } h \\
msk &= (p, q) \\
\text{return } (PP, msk).
\end{align*}
\textbf{end procedure}

\textbf{procedure} \textsc{Ext}(msk, ID):
\begin{align*}
a &= h(ID); \\
\text{return } r &\triangleright \text{(private key) rand. sq. root of } a \text{ or } ea \\
\textbf{end procedure}
\end{align*}

\textbf{procedure} \textsc{Enc}(PP, ID, m):
\begin{align*}
a &= h(ID); \\
t_0, t_1 &\leftarrow \mathbb{Z}_n^* \text{ with } J_n(t_0) = m = J_n(t_1); \\
c_0 &\leftarrow \{u, u \circ d\} \text{ where } u = t_0 + at_0^{-1} \mod n; \\
c_1 &\leftarrow \{v, v \circ d\} \text{ where } v = t_1 + eat_1^{-1} \mod n; \\
\text{return } (c_0, c_1).
\end{align*}
\textbf{end procedure}

\textbf{procedure} \textsc{Dec}((c_0, c_1), r):
\begin{align*}
\text{set } b \in \{0, 1\} \text{ such that } e^b a \equiv_n r^2 \mod n; \\
\text{return } m &= \begin{cases} 
J_n(c_b + 2r), & \text{if } GT_{n,e^b a}(c_b) = 1 \\
J_n(c_b \circ (-d) + 2r), & \text{otherwise}
\end{cases}
\end{align*}
\textbf{end procedure}
```
4.2. BGH’s IBE scheme

encryptor due to the fact that only he is the only one who knows some random parameters he chooses.

The decryption process is very innovative. Without knowing the random parameters that the sender chose, the receiver would not be able to get the message from the cryptotext, but the scheme is constructed such that the quantity which the receiver can compute using his secret key has the same value, i.e. the same Jacobi symbol, with the one computed by the encryptor. How is this possible?

We should figure out a way to get the same value in two different ways, that is by using different parameters. Such a method may use a pair of polynomials \( f \) and \( g \) while setting up some conditions. Thus, considering that the receiver has the secret key, let it be \( r \), and the encryptor has his own “secret” parameter \( s \), then the receiver will use \( f(r) \) for decryption while the sender should use \( g(s) \) for encryption, requesting that their Jacobi symbols to be equal modulo some common parameter \( n \). That is, \( (f(r)|n) = (g(s)|n) \). This is the idea used by Boneh et al. in their scheme proposed in [8]. The concept is very beautiful, interesting and thoughtful.

Their idea was to establish some parameters which will be used for the encryption, respective decryption process. These will be obtained by computing solutions to the following congruential equation which is denoted by \( QC_n(a,S) \) and is given by

\[
ax^2 + Sy^2 \equiv_n 1 \tag{4.2}
\]

where \( n \) is an RSA modulus and \( a,S \in \mathbb{Z}_n^* \). Thus, if we consider \((x,y)\) a solution for Equation (4.2), then the two polynomials will be computed as follows:

\[
f(r) = xr + 1 \mod n,
\]

\[
g(s) = 2(ys + 1) \mod n.
\]

The process of finding these solutions \((x,y)\), which are then
used by the two polynomials, is the bottleneck of BGH’s scheme due to its complexity. This was stated in Section 4 of [8], in the instantiation of the abstract algorithm \( Q \) outputting the two polynomials \( f \) and \( g \), which are called associated polynomials. Now let us see the conditions which these two polynomials should fulfill.

### 4.2.1 Associated polynomials

**Definition 4.2.1** ([74, 62]). Let \( n \in \mathbb{N} \), \( a, S \in \mathbb{Z}_n^* \) and \( f, g \in \mathbb{Z}_n^*[x] \), then, if the two conditions below hold, we say that \( f \) and \( g \) are two \((a, S)\)-associated polynomials:

1. \( f(r)g(s) \in QR_n \) whenever \( a, S \in QR_n \), \( \forall r \in SQRT_n(a) \) and \( \forall s \in SQRT_n(S) \).

2. \( f(r)f(-r)S \in QR_n \) whenever \( a \in QR_n \), \( \forall r \in SQRT_n(a) \).
   
   In this case we say that \( f \) is \( a \)-secure.

**Remark 4.2.1.** When \( S \in J^+_n \setminus QR_n \), Condition (2) is equivalent with saying that \( \left( \frac{f(r)}{n} \right) \) is uniformly distributed in \( \{\pm 1\} \) when \( r \) is uniformly chosen from \( SQRT_n(a) \), whenever \( a \) is a residue modulo \( n \) (see Lemma 3.3 in [8]). This will be exploited in Section 4.2.3.

The soundness of the decryption is provided by (1), while (2) is useful in the proof of security, as we will see in Game 6 from the proof of security of BGH scheme.

**Definition 4.2.2** ([8]). If \( Q \) is a deterministic algorithm that on input \( n, a, S \) outputs two \((a, S)\)-associated polynomials, then \( Q \) is called an IBE compatible algorithm.

It is remarked in [8, Lemma 3.3.] that if \( n = pq \) is an RSA modulus, \( a \in QR_n \), \( f \) is a polynomial satisfying (1) in Definition 4.2.1 and \( S \in J^+_n \), then, when \( S \) is a residue, for all values of \( r \) in \( SQRT_n(a) \), \( (f(r)|n) \) is the same, while when \( S \in J^+_n \setminus QR_n \), \( (f(r)|n) \) is +1 or −1 with equal probability, due to the fact that \( (f(r)|n) \) is uniformly distributed in \( \{\pm 1\} \). This last case happens
because of the four possible values in $\text{SQRT}_n(a)$, obtained by the combination of the possible values of $r \in \mathbb{Z}_p^*$ and of $r \in \mathbb{Z}_q^*$ through CRT. This property is used in the last game of the security proof.

### 4.2.2 The BGH scheme and its security

The abstract BasicIBE scheme was proposed by Boneh et al. in [8], while its security (which was discussed in our paper [62]) will be analyzed below, having as a starting point the following theorem.

**Theorem 4.2.1 ([8]).** Let $h$ in the BasicIBE be modeled as a random oracle, and $F$ a PRF function. Assuming that the QRA holds for the RSAgen, then the BasicIBE scheme is IND-ID-CPA secure and the advantage of an efficient adversary $A$ against it will be

$$\text{IBEAdv}_{A,\text{BasicIBE}}(\lambda) \leq \text{PRFAdv}_{B_1,F}(\lambda) + 2 \cdot \text{QRAdv}_{B_2,\text{RSAgen}}(\lambda)$$

for some efficient algorithms $B_1$ and $B_2$, whose running time is about the same as that of $A$.

In order to avoid computing solutions to $2l$ equations of the form of Equation (4.2) during encryption and, instead, compute only $l + 1$ solutions, Boneh et al. proposed a product formula as follows:

**Lemma 4.2.1 ([8]).** Let $(x_i, y_i)$ be a solution for the equation $a_ix^2 + Sy^2 = 1$, where $i \in \{1, 2\}$, then $(x_3, y_3)$ is a solution to the equation

$$a_1a_2 \cdot x^2 + S \cdot y^2 = 1 \quad (4.3)$$

where $x_3 = \frac{x_1x_2}{Sy_1y_2 + 1}$ and $y_3 = \frac{y_1 + y_2}{Sy_1y_2 + 1}$.

This will be used to combine the solutions of the equations $ax^2 + Sy^2 \equiv_n 1$ and $ex^2 + Sy^2 \equiv_n 1$ in order to get a solution for the equation $eax^2 + Sy^2 \equiv_n 1$. 

4.2.3 A new security analysis for BasicIBE scheme

Due to the fact that $QC_n(a, S)$ is symmetric, using Remark 4.2.1, we extend Definition 4.2.1 with a third condition as follows.

**Definition 4.2.3 ([62]).** Let $n \in \mathbb{N}$, $a, S \in \mathbb{Z}_n^*$ and let $Q(n, a, S)$ be a deterministic algorithm outputting two polynomials $f, g \in \mathbb{Z}_n[x]$. We say that $Q$ is extended IBE compatible if it is IBE compatible and the following condition is fulfilled

3. $\left( \frac{g(s)}{n} \right)$ is uniformly distributed in $\{\pm 1\}$ whenever $a \in J_n^+ \setminus QR_n$ and $S \in QR_n$, where $s \in_R SQRT_n(S)$.

Using an extended IBE compatible algorithm we can attain a better upper bound than the one in Theorem 4.2.1 by slightly modifying the security proof of BGH scheme. This result is stated in the following theorem.

**Theorem 4.2.2.** Let $h$ in the BasicIBE be modeled as a random oracle, and $F$ a PRF function. Assuming that the QRA holds for the RSAgen, then the BasicIBE scheme is IND-ID-CPA secure and the advantage of an efficient PPT adversary $A$ against it will be

$$\text{IBEAdv}_{A, BasicIBE}(\lambda) \leq \text{PRFAdv}_{B_1,F}(\lambda) + \text{QRAdv}_{B_2,\text{RSAgen}}(\lambda).$$

for some efficient PPT algorithms $B_1$ and $B_2$, whose running time is about the same as that of $A$. 

4.3 QR-based IBE schemes that fail security

The way of combining the solutions in order to reduce the number of equations to be solved may lead to important security flows which may result in an insecure scheme, as the ones presented in this section. We begin with the Jhanwar-Barua’s scheme due to the fact that its creators came with the idea of modifying the combining lemma in [8]. We want to emphasize here that, despite the fact that their paper does not offer a secure scheme, the authors bring a useful result: a fast algorithm for solving congruential equations of the form of Equation (4.2).

4.3.1 Jhanwar-Barua scheme

In [39], Jhanwar and Barua replaced the deterministic algorithm for finding solutions to Equation (4.2) by a probabilistic one. This solution is computed using just one modular inversion in \( \mathbb{Z}_n \) and the greatest improvement is that, unlike BGH, it requires no generation of primes, which is a costly process.

They use a formula which is different from the one in [8] in order to combine solutions of Equation (4.2). Their goal is to decrease the ciphertext expansion and to obtain a faster scheme.

**Lemma 4.3.1.** Let \((x_i, y_i)\) be a solution for equation \(ax^2 + S_i y^2 = 1\), where \(i \in \{1, 2\}\), then the pair \((x_3, y_3)\) computed as

\[
x_3 = \frac{x_1 + x_2}{ax_1 x_2 + 1}, \quad y_3 = \frac{y_1 y_2}{ax_1 x_2 + 1}
\]

is a solution to the equation \(ax^2 + S_1 S_2 y^2 = 1\).

Unfortunately, using this combination causes a security flow therefore this scheme is not IND-ID-CPA secure [60].
4.3.2 Other insecure IBE schemes based on QR

JB-revisited by Susilo et al.

In [21], Elashri, Mu and Susilo noticed a security flow of [39], different than the one in [60]. They also showed how to avoid it but, despite this fact, their ”fixed“ variant of JB is still vulnerable to the security flow that Adrian Schipor presented in [60] which remains valuable also for the variant of Elashry et al. [21]. They claim that their variant is as secure as BGH’s scheme but, unfortunately, they use the same variant of combining lemma as Jhanwar and Barua [39] so, the security flow found by Schipor in [60] is also happening here.

Although revisited variants of Susilo et al. are not secure, there is place here for new improvements and for finding new ways to speed up BGH.

4.4 Continuous mutual authentication

In some contexts like military or other important fields, communication through an untrusted channel asks that, at any time during the conversation, both parties to be authenticated [34]. This concept is called continuous mutual authentication (CMA) [50, 47, 48].

4.4.1 Real privacy management

Real Privacy Management (RPM) is a method for CMA which was patented in 2008 by Paul McGough (see [47] and [48]). It generates and manages the keys while providing data secrecy. It also offers a solution providing real-time security for networks.

By communication step we will denote the operations done in order to transmit and receive a single message $m$ from one part to the other in a secure and authenticated manner. A series of communication steps we be called session.

The RPM protocol provides perfect secrecy and the security of
it consists of a robust authentication (construction and transmis-
sion) and a secure encryption.

If an eavesdropper, at some point, gets the credentials, he has
access to all the communication that follows. In the case of such an
attack, in order to stop the intruder from accessing the information
that will be transmitted after a successful attack, one can think
to renew the credentials, but the question is: “How to transmit
the new acks avoiding that the attacker also get access to these
new credentials?” . This is the main issue of the protocol that
motivated our study. Thus, we propose in what follows a new
authentication method that prevents this attack [50]. The main
idea is that the update method should be different than the one
used by RPM in order to prevent the attack mentioned above.

4.4.2 RPM description

RPM is a protocol that ensures an authenticated and secure mes-
sage transmission through its key generation and key management
methods. There are four main configuration of RPM which com-
bine these functions in order to provide data security and continu-
ous mutual authentication, as we can see in [71]: PDAF baseline,
PDAF network, CE baseline and CE network. All the four con-
fugurations ensure (continuous) authentication and data security
together with a key establishment and key exchange method. One
can observe that, revealing just the secret key $k$ at some point,
without any other secret information, will lead to the decryption
of the message from that communication step, but neither previous
nor future messages can be decrypted.

4.4.3 Continuous mutual authentication
and data security

MA and data security The methods which are currently known and
used for the initial step are far from being efficient or malleable.
That’s what made us look for an improved method which can
**CMA and data security**

<table>
<thead>
<tr>
<th>Sender</th>
<th>PP: $n, e, h$</th>
<th>Receiver ($r$)</th>
</tr>
</thead>
</table>
| 1. $S \leftarrow \text{RandGen}(\lambda)$  
1.a) $a = H(ID(\text{Receiver}))$  
1.b) $t_1, t_2 \leftarrow \text{RandGen}(\lambda)$ s.t. $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = (-1)^b$  
1.c) $ID_1 = (t_1 + a t_1^{-1}) \mod n$  
1.d) $ID_2 = (t_2 + ea \ t_2^{-1}) \mod n$  
2. $OR_{ID_1} = ID_1 \oplus ms$  
3. $OR_{ID_2} = ID_2 \oplus msk$  
4. $OR_S = ID_1 \oplus S$  
5. $k = f(ID_2, S)$ | | 6. $\text{Enc}\{m\}_k, OR_S, OR_{ID_1}, OR_{ID_2}$ |
| 7. $ID_1 \leftarrow OR_{ID_1}$  
8. $ID_2 \leftarrow OR_{ID_2}$  
9. $S \leftarrow OR_S$  
10. $k = f(ID_2, S)$  
11. $m \leftarrow (\text{Enc}\{m\}_k, k)$  
12. $b \leftarrow (ID_1, ID_2)$ | |

Where $PP$ are the public parameters, $r \in SQRT_n(a)$, and $a = ID(\text{Receiver})$

---

**Figure 4.3:** PDAF network configuration with Cocks method [50]

The method we come up with is based on Cocks’ identity-based encryption scheme, see Section 4.1. The main improvement of our variant of transmitting information is that it doesn’t require any supplementary steps. It uses Cocks’ scheme which is very fast and elegant. It encrypts only one bit at a time, which, in our context, is provide security, efficiency and flexibility. Our result [50] can be used both, for transmitting the initial *acks* and for renewing the *acks* at any moment (during the sessions or after each session). This will provide the confidentiality of the communication in an authenticated manner without interrupting the process.
useful because the (new) credential data can be sent in more than one session step, without being observed by an eavesdropper. Its computational cost is less than one modular exponentiation, so it is a very light alternative, regarding the time complexity, for sending (re)authentication credentials (continuously) at any time. This is illustrated in Figure 4.3.

The scheme will act as follows. The two communicating parties use some public parameters, \((n, e, h)\), and a master secret \(msk\) (which is an exchange key), while the Receiver uses some private key which corresponds to its identity \(a\). The Sender generates a random parameter \(S\), then he computes the identity using a hash function \(H\) on \(ID\), where \(ID\) is now a public parameter which uniquely identifies the Receiver, like his phone number or his e-mail address. Then, the Sender randomly generates two integers in \(Z_n^*\), such that their Jacobi symbol to be equal with the bit to be transmitted. Then the two parameters \(ID_1\) and \(ID_2\) will be computed just like in Cocks’ scheme, \(ID_i = (t_i + e^{(i+1) \mod 2} at_i^{-1})n\), \(i \in \{1, 2\}\). After that, he computes the secret key \(k\) by the function \(f\) applied on \(ID_2\) and \(S\). Finally, he hides \(S, ID_1\) and \(ID_2\) using \(OR\) function and \(msk\) in \(OR_S, OR_{ID_1}, OR_{ID_2}\), respectively, and sends these three values, together with the encrypted message \(Enc\{m\}_k\) to the Receiver. Using the \(msk\), the Receiver recovers \(S, ID_1, ID_2\) from \(OR_S, OR_{ID_1}, OR_{ID_2}\), respectively, computes the encryption key \(k\) and using the function \(f\) on \(ID_2\) and \(S\), recovers the message from the cryptotext and, finally, he also recovers the extra bit that was transmitted.

So we can see that this is a great way to solve the problem of exchanging the \(acks\). This can be done in each communication step, as long as it takes, without any supplementary step or message. Thus, if the two parties establish, outside of the protocol, when they will transmit the supplementary bits, the messages that are sent between the two parties are identical, from an attacker’s perspective, with a transmission of a message without the additional bit. The \(acks\) can be transmitted without modifying or disturbing the usual communication. The steps which are re-
quested by the additional $ack$ sharing do not considerably affect the time complexity - they are less than a modular exponentiation - and will be done only when the parties agree to change the $acks$.

From a security point of view, the protocol remains protected by the randomness of the Cocks ciphertexts, so the two parameters which are computed through Cocks’ IBE scheme, $ID_1$ and $ID_2$ are indistinguishable from some random chosen parameters (assuming the QRA).

4.5 Pseudo-random generators

Pseudo-random generators (PRG) are deterministic algorithms that get as input a seed and output numbers (PRNGs) or bits (PRBGs) emulating a truly random behavior. They have a period which is the distance between the beginning of a sequence and the same output [29, 17, 24]. The goal of PRFs is that the output of the generator to be at least computationally, if not statistically, indistinguishable from a truly random sequence and to stretch the input, the seed, as much as possible, such that the pseudo-randomness to be preserved.

Researchers thought that residues constitutes a good tool in creating pseudo-random generators. Some of these results are due to Damgard [16], Perron [54], Peralta [53], and Tarakanov [68], to only name a few. One such generator which is based on QR is the Blum-Blum-Shub generator [5]. It meets the requirements mentioned above, assuming that the QRA holds. Another example is [59], where the authors describe a way to create a family of finite binary sequences together with a pseudo-random bit generator that outputs such sequences also using QR.
Chapter 5

From IBE to ABE

5.1 Introduction

Attribute-based encryption (ABE)\(^1\) is a generalization of IBE allowing one to many encrypted communication, which defines the identities as sets of attributes characterizing the destination group \(^5\). So, in order to decrypt a message, one should have a valid (accepted) combination of attributes which is called an access structure. ABE is currently implemented using three mathematical tools: bilinear maps \(^{36, 18}\), lattices \(^{46, 15}\) and QR \(^{10}\).

The access structures, depending on their complexity, can be expressed by Boolean formulas for (non-/monotone) Boolean circuits or by general Boolean circuits.

The Boolean circuits with two input wires, i.e. fan-in two, \textsc{and} and \textsc{or} gates but without any \textsc{not} gate are called monotone. The first key-policy (KP-ABE) schemes work for simple access structures which can be expressed as Boolean formulas, for monotone \(^{33}\), or non-monotone \(^{52}\) Boolean circuits. Some access structures are more complex, like multi-level ones \(^{69, 70}\), and they cannot be expressed using Boolean formulas, but using general Boolean circuits instead.

\(^1\)We present in this chapter a study we did in \(^{73}\).
Goyal et al. introduced in [33] the concept of KP-ABE, and the first scheme providing one-to-many encryption. They used secret sharing and a bottom-up reconstruction using a bilinear map in order to allow fine-grained sharing of encrypted data. It works for simple access structures, because it only can use Boolean trees, and not general circuits. Only in 2013 a first solution for general Boolean circuits came by [27]. It is an extension of [33] which uses leveled multilinear maps, so it is less efficient than [33], but it can express more complex access structures. In the same year a lattice-based solution followed [32]. All these schemes are secure in the standard model.

Using only bilinear applications and getting a backtracking attack-resist scheme is a challenge and an open question for KP-ABE model. Since 2014 there were some attempts to meet that need.

Ţiplea and Drăgan extended [33] from the Boolean tree case to (monotone) Boolean circuits by [72], which is more efficient. It uses secret sharing together with a single bilinear map, and has the same level of security as the original scheme. A slight enhance of it is obtained in 2017 by Hu and Gao in [37], with shorter decryption keys and the same level of security.

The solution in [19] is a KP-ABE scheme working for general
5.2. ABE and the backtracking attack

Boolean circuits, like [27], but it is by far more efficient because it uses *chained multilinear maps* which are simplified types of multilinear maps, together with secret sharing techniques, and provides the same level of security.

5.2 ABE and the backtracking attack

In the beginning of this section we give some definitions and set the notation that will be used in this chapter.

*Access structures* [67] are usually represented as Boolean circuits [3]. A circuit has input and output wires (some of them are not gate input wires, while others are not gate output wires), and also some gates which can be \texttt{AND}, \texttt{OR}, and \texttt{NOT}. Pictorially, the input wires are below the gates, and we will count them by *fan-in*, while the output wires are above the gates, and we will count them by *fan-out*. There will be two input wires for each \texttt{AND} and \texttt{OR} gates, while for \texttt{NOT} gates will be a single input wire. Each of them outputs at least a wire. By *circuits* we will understand *Boolean circuits*, unless otherwise mentioned. The *Boolean formulas* are those circuits in which all gates have a fan-out of one. We recall that a *monotone* circuit does not have \texttt{NOT} gates. In this chapter we deal only with monotone circuits having exactly one output wire but, as it is noticed in [27] this fact does not lose generality.

Let $\mathcal{U}$ be a set of attributes, $A$ be a subset of $\mathcal{U}$, and $\mathcal{C}$ a circuit. If the elements in $A$ can be mapped such that they correspond one-to-one with the input wires of the circuit, then $\mathcal{C}$ is considered a Boolean circuit over the set $\mathcal{U}$. The circuit $\mathcal{C}$ will be evaluated for a subset of attributes $A$ to 1 or 0 by setting 1 to all input wires mapped to attributes in $A$, 0, respectively. The input of the circuit is composed of values 1 and 0; each such value is transmitted from the lowest level to the top through the gates in a standard way. The result of evaluating $\mathcal{C}$ for $A$ will be denoted by $\mathcal{C}(A)$. The access structure defined by $\mathcal{C}$ is the set of all $A$ with $\mathcal{C}(A) = 1$. 
Let $\mathcal{U}$ be a set of attributes, then the tuple $(\overline{a}, \overline{U}, S)$ will be called a disjunctive multi-level access structure over $\mathcal{U}$, where $\overline{a} = (a_1, \ldots, a_k)$ and $a_i \in \mathbb{N}$, such that $0 < a_1 < \cdots < a_k$, $\overline{U}$ partitions $\mathcal{U}$ by $= (\mathcal{U}_1, \ldots, \mathcal{U}_k)$, and $S = \{A \subseteq \mathcal{U} | (\exists i \in \{1, \ldots, k\})(|A \cap (\bigcup_{j=1}^{i} \mathcal{U}_j)| \geq a_i)\}$. If we set $S$ such that the above expression is valid for any $i \in \{1, \ldots, k\}$ this will define the conjunctive case of multi-level access structures.

A KP-ABE scheme contains four PPT algorithms: the Setup algorithm, which outputs the master (secret) key $msk$ starting from the security parameter $\lambda$ and the public parameters $PP$. The encryption algorithm $Enc(m, A, PP)$, which uses the input message $m$, a subset of attributes $A \subseteq \mathcal{U}$, and the public parameters $PP$ in order to output the cryptotext $E$. The secret key $sk$ is obtained using the algorithm $KeyGen(C, msk)$ from a Boolean circuit $C$ and $msk$. Finally, the message $m$ is decrypted by the algorithm $Dec(E, sk)$ if a valid $sk$ is used together with the ciphertext $c$.

**The correctness property:** for any pair of public parameters together with the master secret outputted by the Setup algorithm, any circuit $C$ over some defined set of attributes, $\mathcal{U}$, and any subset $A$ of $\mathcal{U}$, let $m$ be any message in the message space and $E$ its encryption under the public parameters $PP$ and the subset $A$. If the circuit is evaluated to 1 for $A$, i.e. $C(A) = 1$ then the decryption of $E$, $Dec(E, sk)$, will return $m$, for all $sk$ outputted by $KeyGen(C, msk)$. This property must be fulfilled by all KP-ABE schemes.

The first KP-ABE solution cannot work for circuits but only for Boolean formulas due to the fact that the value computed to one of an OR gate input wires, can immediately flow down to the other input wires, due to the secret sharing procedure, then, if the same input value (wire) is used by another gate, as in the case of circuits, then this information flow cannot be avoided. This is the backtracking attack which is possible only in the context of a circuit. When talking about Boolean formulas the situation is changed and such an attack is not possible because an input wire
5.2. ABE and the backtracking attack

of an OR gate is never used by another gate (see [72] for a pictorial view of this attack).

5.2.1 The secure KP-ABE_Scheme_1

The scheme is described in the thesis and in [18]. Its correctness results by a simple computation while the next theorem states its security.

**Theorem 5.2.1 ([18]).** The KP-ABE_Scheme_1 is secure in the selective model under the decisional bilinear Diffie-Hellman assumption.

The KP-ABE_Scheme_1 cryptosystem is not efficient when there are many path-connected FO-gates. Nevertheless, the proposed scheme may be even more efficient than the scheme in [27] when the FO gates are at the bottom of the circuit and just a few of them are linked by a path. As an example we will apply it for the multi-level access structures in [66, 69].

It is within reach to see that Boolean formulas are not complex enough to express disjunctive and conjunctive multi-level access structures (see [18] for the proof), therefore Boolean circuits will be used for such access structures. Now, for an easier and clearer representation of the structures, the circuits we use will be enhanced by \((a, b)\)-threshold gates [33], where \(b \geq 2\) and \(1 \leq a \leq b\). These gates will have \(b\) input wires and just one output wire. If we evaluate the output of this type of gates it will return 1, i.e. true, when the threshold is reached, i.e. \(a\) input wires are assigned to 1. Therefore we can say that the threshold is 1 in the case of OR gates, (and denote this by \((1, 2)\)-threshold gates), while it is 2 for the AND gates (then \((2, 2)\)-threshold gates will denote them).

By the use of a probabilistic linear secret sharing scheme KP-ABE_Scheme_1 may naturally extended such that the endowed version will contain, in addition, threshold gates. We can notice that KP-ABE_Scheme_1 works faster than the scheme of Garg et al. in [27] (see [72] for deeper specifications).
Algorithm 4: KP-ABE_Scheme_1

**procedure** SETUP($\lambda$, $n$)

choose a prime $p$; set $G_1$ and $G_2$;  
▷ two multiplicative groups of prime order $p$
set $g$;  
▷ a generator of $G_1$
set $e : G_1 \times G_1 \rightarrow G_2$;  
▷ a bilinear map
$\mathcal{U} \leftarrow \{1, \ldots, n\}$;  
▷ the set of attributes
$y \in \mathbb{Z}_p$ and $t_i \in \mathbb{Z}_p, \forall i \in \mathcal{U}$; 
$PP \leftarrow (p, G_1, G_2, g, e, n, Y = e(g, g)^y, (T_i = g^{t_i} | i \in \mathcal{U}))$;
$msk \leftarrow (y, t_1, \ldots, t_n)$;
return $(PP, msk)$

**end procedure**

**procedure** ENCRYPT($m, A, PP$)

$m \in G_2$ and $s \leftarrow \mathbb{Z}_p$; 
$A \subseteq \mathcal{U}$;  
▷ a non-empty set of attributes
$E \leftarrow (A, E' = mY^s, (E_i = T_i^s = g^{t_is} | i \in A), g^s)$;
return $E$

**end procedure**

**procedure** KEYGEN($C, msk$)

$(S, P) \leftarrow \text{SHARE}(y, C)$;
foreach $i \in \mathcal{U}$ do

$D(i) = (g^{S(i,j)/t_i} | 1 \leq j \leq |S(i)|)$;
$D \leftarrow ((D(i) | i \in \mathcal{U}), P)$
end foreach

return $D$

**end procedure**

**procedure** DECRYPT($E, D$)

$R \leftarrow \text{RECON}(C, P, V_A, g^s)$;
foreach $i \in \mathcal{U}$ and $1 \leq j \leq |S(i)|$ do

if $i \in A$ then

$V_A(i, j) \leftarrow e(E_i, D(i, j)) = e(g^{t_is}, g^{S(i,j)/t_i}) = e(g, g)^{S(i,j)s}$;
else $V_A(i, j) \leftarrow \bot$; $m \leftarrow E'/R(o, 1)$;
end if
end foreach

return $m$

**end procedure**
5.3 KP-ABE for Boolean circuits using secret sharing and multilinear maps

Garg et al. were the first to create a KP-ABE scheme that can be used also for general Boolean formulas. In [27] they renounced to use secret sharing, and implemented a technique which goes bottom-up through the circuit instead, using leveled multilinear maps, in order to change the generator on each level. Apart from the output wire of the circuit, for all the other wires, there are between two and four keys assigned to.

In what follows we present the result from [19] which does not renounce to use secret sharing, but they defend the scheme from the backtracking attack using, in addition, leveled multilinear maps. This combination produces a better solution than the scheme in [27]. For a complete description and additional specifications the reader is invited to see [19].

5.3.1 The secure KP-ABE_Scheme_2

In [19] a simpler form of leveled multilinear maps was used, which is called chained multilinear maps. Let $p$ be a prime, $G_1, \ldots, G_{k+1}$ be multiplicative groups of order $p$, then the notion of chained multilinear map stands for a sequence of bilinear maps $(e_i : G_i \times G_1 \rightarrow G_{i+1}|1 \leq i \leq k)$ such that, if $g_1 \in G_1$ is a generator of $G_1$, then for all $i \in \{1, \ldots, k\}$, a generator $g_{i+1}$ for each group $G_{i+1}$ can be recursively defined by $g_{i+1} = e_i(g_i, g_1)$ (due to the fact that $e_i$ is a bilinear map). Thus, $(e_i|1 \leq i \leq k)$ is also a form of leveled multilinear map, but a simpler one. Now we will see how these constructions are used in [19].

Let $\mathcal{C}$ be a Boolean circuit, we will denote by $r$ its total amount of $\text{FO}$-levels, and $(e_i|1 \leq i \leq r + 1)$ will stand for a chained multilinear map as it was described above. In the setup phase there is a random integer chosen, namely $y \in_R \mathbb{Z}$. The encryption al-
Algorithm works as follows: for a message $m \in G_{r+2}$, it will set a random integer $s \in R \mathbb{Z}$, and compute the corresponding ciphertext as: $mg_{r+2}^{ys}$. The decryption algorithm will use two procedures, for secret sharing, and reconstruction, respectively, to get the $g_{r+2}^{ys}$ quantity, which will be used in order to decrypt and get the message.

$$x_1 a + x_2 \equiv y \mod p, \quad x_3 + x_5 a \equiv y \mod p, \quad x_4 b_1 \equiv x_2 \mod p, \quad x_4 b_2 \equiv x_3 \mod p$$

**Figure 5.2: SHARE(y, C)**

The scheme [19] is described in Algorithm [5]. The correctness of KP-ABE_Scheme_2 follows by a simple computation. Regarding the security of the scheme, we have the following important result.

**Theorem 5.3.1 ([19]).** The KP-ABE_Scheme_2 is secure in the selective model under the decisional multilinear Diffie-Hellman assumption.

The translation presented in [27] to graded encoding systems [26], assuming that leveled multilinear maps exist, can be applied in the same way to KP-ABE_Scheme_2.

The scheme can be enhanced in order to work for circuits which have gates with three or more input wires and in the same time
Algorithm 5 : KP-ABE_Scheme_2

procedure Setup($\lambda, n, r$)
    choose a prime $p$;
    set $G_1, \ldots, G_{r+2}$;  $\triangleright$ multiplicative groups of prime order $p$
    set $g_1 \in G_1$;  $\triangleright$ a generator
    set $(e_i : G_i \times G_1 \to G_{i+1} | 1 \leq i \leq r+1)$;  $\triangleright$ a bilinear maps
    $g_{i+1} = e_i(g_i, g_1)$, for all $1 \leq i \leq r+1$;
    $U = \{1, \ldots, n\}$;  $\triangleright$ the set of attributes
    foreach $i \in U$ do
        $y \leftarrow Z_p$ and $t_i \leftarrow Z_p$;
    end foreach
    $PP \leftarrow (n, r, p, G_1, \ldots, G_{r+2}, g_1, e_1, \ldots, e_{r+1}, Y = g_{r+2}^{y}, (T_i = g_{1}^{t_i} | i \in U))$;
    $msk \leftarrow (y, t_1, \ldots, t_n)$;
    return $(PP, msk)$.
end procedure

procedure Encrypt($m, A, PP$)
    $m \in G_{r+2}$, $s \leftarrow Z_p$ and $A \subseteq U$;  $\triangleright$ where $A$ cannot be empty
    foreach $i \in A$ do
        $E \leftarrow (A, E' = mY^s, (E_i = T_i^s = g_{1}^{t_is}))$;
    end foreach
    return $E$.
end procedure

procedure KeyGen($C, msk$)
    $(S, P, L) \leftarrow \text{Share}(y, C)$;
    foreach $i \in U$ do
        $D(i) = g_{1}^{S(i)/t_i}$ and $D \leftarrow ((D(i)|i \in U), P, L)$
    end foreach
    return $D$.
end procedure

procedure Decrypt($E, D$)
    $R \leftarrow \text{Recon}(C, P, L, A, V_A)$;
    foreach $i \in U$ do
        if $i \in A$ then
            $V_A(i) \leftarrow e_1(E_i, D(i)) = e_1(g_{1}^{t_is}, g_{1}^{S(i)/t_i}) = g_{2}^{S(i)s}$;
            else
                $V_A(i) \leftarrow \perp$ and $m \leftarrow E'/R(o)$;
            end if
    end foreach
    return $m$ or $\perp$.
end procedure
keeping the same size of the key used for decryption. The bounding for the FO-levels is flexible, an extension to an unbounded version at the cost of increasing the number of FO-level-keys can easily be done. The amount of decryption key elements may also be decreased reusing the key of each FO-level for one of the output wire of each (some) gates from the same level.
Chapter 6

Conclusion and open problems

After a study of almost six years on QR we can say that they represent a very powerful tool in many areas such as mathematics, computer science, and many other fields. Due to their simplicity and elegance they are well understood and working with them is much more accurate than with other mathematical tools.

The mathematics part on QR was studied in Chapter 3, where we have computed some useful cardinalities on sets of the form \( a + QR_m \) with different Jacobi patterns, where \( m \) is either a prime or an RSA modulus, then some probabilities which were later applied in Chapter 4. These results were used in Chapter 4 which deeply analyzes Cocks’ IBE scheme [12] and its ciphertexts which helped to rigorously prove the Galbraith test and to present an anonymous variant of Cocks’ scheme, [40], in a very simple manner [51]. Then a contribution to the upper bound in the security proof of the BGH scheme was in order in Section 4.2. A warning flag was then triggered regarding some combining methods used for the ”optimization“ of [8] and [40]. The chapter ends with some applications of Cocks’ scheme to continuous mutual authentication using RPM, Section 4.4.3, and some examples of PRBG-s from QR, Section 4.5.
Chapter 6. Conclusion and open problems

In the last part of the thesis we analyzed attribute-based encryption (ABE) and the backtracking attack, [27]. In the timeline from Figure 5.1 one can easily see the main tools used in creating ABE schemes, and the state of the art regarding KP-ABE. Further on, in the same chapter, two KP-ABE schemes [18, 19] were described, which avoid the backtracking attack and are probably the most efficient at the moment, together with their security proofs. The state of the art on ABE (Section 5.1) shows that the current results are based mainly on pairings, then on lattices, and a few shy results on residues. As a further work we are interested in analyzing how suitable are QR in creating ABE schemes.

Open problems and further work

Even if it took six years to finish this thesis, this is just the beginning of a next and deeper level of my research. I am interested both in developing the ideas we begun to study in our research group and also in finding new issues where quadratic residues and/or higher degree residues can help.

If we were to describe some of the open problems, one of them would be to discover rules for higher levels of the Jacobi patterns [16, 53] for sets of the form $QR_n(a + \cdots QR_n(a + QR_n) \cdots)$ which can help to validating (or not) the pseudo-randomness of residues.

Also an interesting subject to be researched is the extension of Cocks’ scheme such that it will encrypt many bits at once. Some burning tasks left by BGH’s scheme [8] are: to find other ways to improve the costing deterministic algorithm (its bottleneck), or finding a secure and improved way of combining solutions in the encryption/decryption algorithms of this scheme. Another exciting area to be explored is how to use our results on QR distribution, or to extend them, in order to be useful in fields like VoIP, security and defense applications, cloud, big data [10], and so on.

In a further work we are interested to analyze the idea of using QR instead of, or combined with, bilinear maps or lattices, preparing to the post-quantum cryptography [56, 9, 44].
Selected bibliography


