九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Zero-Knowledge Protocols for Code-Based Public-Key Encryption and Their Applications

胡, 荣

https://doi.org/10.15017/1543929

出版情報：九州大学, 2015, 博士（数理学）, 課程博士
バージョン：
権利関係：全文ファイル公表済

KYUSHU UNIVERSITY

DOCTORAL THESIS

Zero-Knowledge Protocols for Code-Based Public-Key Encryption and Their Applications

Author:
Rong HU

Supervisor:
Prof. Tsuyoshi TAKAGI

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Mathematics

in the

Graduate School of Mathematics

July 20, 2015

Declaration of Authorship

I, Rong Hu, declare that this thesis titled, 'Zero-Knowledge Protocols for Code-Based Public-Key Encryption and Their Applications' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Doctor degree at Kyushu University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Abstract

Cryptography relies on Mathematics in all its aspects, beginning from the constructions relying on various mathematical theories, continuing with security evaluation of cryptographic systems, and proving their security, and finally ending in implementation.

Recently, new security threats are posed by the emerging quantum computing technology. Specifically, quantum algorithms can break some public-key encryption schemes such as RSA and Elgamal, which are widely used for protection of computer systems and networks. This issue demands us to develop a new generation of cryptographic systems, which will serve as secure alternatives to the currently used ones. Such the new systems are referred to as the post-quantum cryptography.

One promising direction in post-quantum cryptography is the systems whose security is based on hardness of mathematical problems arising in the context of coding theory. In particular, the problem of decoding random linear codes has been studied for over 30 years, and still no polynomial-time solution has been proposed, even when using quantum algorithms. In this thesis, we focus on this area, which is called the code-based cryptography.

The first code-based public-key encryption (PKE) scheme was introduced by R.J. McEliece in 1978. Since then, various code-based public-key encryption, digital signature and identification schemes were introduced, but currently, one of the main challenges is to introduce more advanced cryptographic functionalities based on coding.

In this thesis, first, we give a brief introduction about post-quantum cryptography and code-based cryptography, and then we provide the background information about the cryptographic primitives, which we will study, as well as the relevant notions and results from coding theory and cryptography.

Next, we introduce our contributions as follows. Firstly, we study zero-knowledge (ZK) identification schemes based q-ary linear codes. We show that when $q < 5$, a straightforward generalization of Stern's ZK identification scheme (1993) is more efficient in terms of both communication and computation, as compared to the ZK identification scheme by Cayrel, Véron and El Yousfi Alaoui (2010), which is specifically designed for q-ary codes.

Secondly, we introduce the first proof of plaintext knowledge (PPK) for the McEliece PKE and the Niederreiter PKE. These protocols allow the encryptor to prove the knowledge of the plaintext contained in a given ciphertext to any party, who does not hold the secret key for decryption. We also provide a performance evaluation for the proposed schemes.

As an application of the above PPK for the McEliece PKE, we present the first verifiable public-key encryption, which allows the encryptor to prove to any party that a given plaintext is contained in a given ciphertext, again without decrypting it. We also discuss why this idea cannot be applied to the case of Niederreiter PKE. We also provide a performance evaluation for our proposal.

Lastly, the first designated confirmer signatures based on coding is constructed following the framework of El Aimani (2010). This type of signature can only be verified via the interaction with the signer or the designated party called a "confirmer". The above proposal for verifiable PKE is applied to construct the signature verification protocol.

Finally, we provide our concluding remarks and discuss the future work.

Acknowledgements

I am grateful to the God for the good health and wellbeing that were necessary to complete this thesis. My highest gratitude goes first and foremost to Professor Takagi, my supervisor, for his constant encouragement and guidance. Without his support and illuminating instructions, this thesis could not have reached its present form.

I would like to express my heartfelt gratitude to Dr. Morozov, my co-advisor, who led me into the world of coding theory and code-based cryptography. Now that I finally reached the end of my Ph.D studies, I would like to express my gratitude for this opportunity to conduct a joint research with him.

Last but not least, my thanks go to my beloved family and the members of Takagi Laboratory.

Contents

List of Figures

List of Tables

Abbreviations

DCS	**D**esignated **C**onfirmer **S**ignatures
GC-D	**G**oppa **C**ode **D**istinguishing
G-SD	**G**eneral **S**yndrome **D**ecoding
IND-CPA	**Ind**istinguishability under **C**hosen-**P**laintext **A**ttack
IND-CCA1	**Ind**istinguishability under **C**hosen **C**iphertext **A**ttack
IND-CCA2	**Ind**istinguishability under **A**daptive **C**hosen **C**iphertext **A**ttack
INV-CMA	**Inv**isibility under a **C**hosen **M**essage **A**ttack
PKE	**P**ublic **K**ey **E**ncryption
PKI	**P**ublic **K**ey **I**nfrastructure
PPK	**P**roof of **P**laintext **K**nowledge
PPT	**P**robabilistic **P**olynomial **T**ime
SD	**S**yndrome **D**ecoding
(S)EUF-CMA	(**S**trong) **E**xistentially **Unf**orgeable under Adaptive **C**hosen **M**essage **A**ttacks
US	**U**ndeniable **S**ignatures
ZK	**Z**ero **K**nowledge

Symbols

$\mathcal{A}$	an adversary
$\mathbb{F}_q^{k\times n}$	k by n matrix over q-ary field
H	parity check matrix
G	generator matrix
I	the identity matrix
1^k	a sequence of k unary symbols
pk	public key
sk	secret key
P	prover
$\widetilde{P}$	cheating prover
V	verifier
$\widetilde{V}$	cheating verifier
$\mathcal{K}$	key generating algorithm
wt	the Hamming weight of a binary vector
$\mathbb{Z}$	a set of integer numbers
$\mathbb{N}$	a set of natural numbers
$\mathbb{R}$	a set of real numbers
$\mathbb{R}^+$	a set of positive real numbers
$\|x\|$	the length of a vector x
$negl$	negligible function
$\perp$	reject
Pr	probability

Dedicated to my parents and China Scholarship Council

Chapter 1

Introduction

1.1 Cryptography

Originated from the ancient art of transposition and substitution of letters of messages in secret communication, such as the famous Caesar cipher[1], cryptography has now become a highly developed research subject which aims at protecting any system from being attacked by an adversary.

The development of cryptography embodies in mainly two aspects.

- The border of cryptography, in particular, has been enriched from merely secret communication to a variety of topics including confidentiality, integrity, authentication, secure communication, electronic transaction.
- Different from classical ciphers, which are more of an art form, modern cryptography uses accurate and precise security definitions and admits rigorous proofs. However, computationally secure cryptographic systems are currently all based on some well-studied assumptions (i.e., some problems that are believed to be hard to solve).

In this section we review some important areas of modern cryptography on which we focus in this thesis.

1.1.1 Public-Key Encryption

Secret communication is the most immediate application of cryptography, where two parties want to communicate securely in the presence of a third party who can eavesdrop over the

[1]The Caesar cipher substitutes each English letter by some fixed number of positions down the alphabet.

communication channel. Encryption schemes are developed to solve the problem of secret communication (confidentiality issue). In an encryption scheme, the sender encrypts the message (formally called the ”plaintext”) using some encryption key. The result of the encryption (called ”ciphertext”) is sent over the public channel. Upon receiving the ciphertext, the receiver can decrypt it using the decryption key and get the plaintext back. At the same time, the adversary who does not have the secret key cannot decrypt the ciphertext thus the confidentiality of the communicated message is preserved. Depending on whether the encryption and decryption are the same function, encryption schemes can be divided into two categories:
private-key encryption schemes and **public-key encryption**(PKE) schemes. In private-key encryption schemes, the sender and receiver hold the same key both for encryption and decryption. The private-key encryption schemes are usually more simple and efficient than public-key encryption schemes, but a serious challenge of secure key distribution is present. Since the key must be kept away from the adversary, the sender and the receiver have to figure out a way to create a secure channel for distributing the encryption / decryption key. This challenge stipulated the introduction of the notion of public-key cryptography proposed by Diffie and Hellman in 1976[DH76]. The advantage of a public-key encryption scheme is that the encryption and decryption are detached. The sender can encrypt a plaintext using the public key of the receiver which is publicly available. The receiver who owns the private key can decrypt the ciphertext encrypted under her public key and get back the intended plaintext.

Besides the direct application in secret communication and data confidentiality, public-key encryption can be used in constructing other cryptographic primitives such as commitment schemes, and identification schemes.

1.1.2 Post-Quantum Cryptography

As mentioned earlier, modern cryptography are based on hardness assumptions such as the problems of integer factorization and discrete logarithm problems assumed to be hard to solve. Nowadays, the most widely used encryption scheme, RSA[RSA78], is based on integer factorization. Many popular cryptographic schemes such as ElGamal encryption[ElG85] and the elliptic curve cryptography[Mil86, CMO98] are based on the discrete logarithm problem. It is important to review the hard problems since whether they are indeed hard has direct impact on the cryptographic schemes based on them. Although currently we do not know any algorithm that can efficiently (i.e., running in polynomial time in the problem size) solve integer factorization and discrete logarithm problem on a **Turing Machine**, the discovery of Shor[Sho94] shows that there are exist polynomial algorithms running on a quantum machine which can easily solve integer factorization and discrete logarithm problem.

Although the quantum computers that can practically run a quantum algorithm are still being researched, but if a quantum computer can be put into practiced use in the near future, almost all currently used cryptographic schemes will be broken. Thus it is necessary to investigate alternative cryptographic schemes which will be secure even against quantum computing. Under this situation, the new cryptosystems which can resist attacks of quantum computers need to be studied, which is done by a so-called post-quantum cryptography.

Currently, post-quantum cryptography has the following main topics[BBD09]:

1. Lattice-based cryptography;
2. Multivariate cryptography;
3. Hash-based cryptography;
4. Code-based cryptography;
5. Super singular Elliptic Curve Isogeny cryptography.

In this thesis, we focus on code-based cryptography.
If the attacker wants to break a coding assumption, it will have to solve the problem of decoding a random code, and although there are some algorithms attempting to solve this problems, all of them are exponential[Bar98]. Two important candidates for the post-quantum public key encryption are the code-based PKE schemes by McEliece [McE78] and Niederreiter [Nie86].

1.1.3 Code-Based Public-Key Encryption

There are many kinds of cryptosystems in code-based public-key encryption cryptography. The security of them is based on the hardness of decoding in a linear code. The famous code-based hardness is syndrome decoding(SD) Problem and the general syndrome decoding(G-SD) Problem[Rot06]. From two code-based hard assumption mentioned above, there are two important candidates for the code-based public key encryption. One is the code-based PKE schemes by McEliece [McE78] which based on general decoding problem, the other is Niederreiter [Nie86] which is based on syndrome decoding problem. Their security is based on hardness of decoding of Goppa code, which is a well-studied cryptographic assumption [BMVT78, Sen02, EOS07, Ber10, BLP11, DMR11a, DMR11b, FGUO+13, BJMM12].

The McEliece encryption is the first code-based PKE been proposed. It uses the error-correcting codes by Goppa [Gop70, Mat80]. While a more advanced PKE pointed out by Bernstein et al.[BLP11] using a different family of codes has a asymptoticly good performance towards the origin McEliece PKE. The other code-based PKE system is Niederreiter cryptosystem, it uses the random irreducible Goppa code, The benefit is that it can use pseudorandom generator to construct the commitment functions which does not have trapdoor. So the adversary who does not know the secret key will face the syndrome decoding problem. It means that, until now, there is no polynomial time algorithm to decode the ciphertext from Niederreiter cryptosystem.

By comparison, the speed of Niederreiter cryptosystem encryption is close to other public key cryptosystems which use pseudorandom permutation, for instance, the block ciphers.

Since breaking both of these PKE cryptosystems is believed to be infeasible for properly chosen parameters [EOS07, BLP11, FGUO+13], even for adversaries equipped with quantum computers [Ber10, DMR11a, DMR11b]. That makes these PKE schemes candidates for the post-quantum world. Existing results include, for instance, identification schemes and digital signatures – see e.g. the surveys [EOS07, OS09, CM10], and also [Pie12] for related results. Nevertheless, the current challenge in the code-based cryptography is to enrich the variety of code-based cryptographic protocols, which gave the direction of our main task in this thesis, that is, to study and research how to design code-based cryptosystems.

1.1.4 Digital Signatures

In generally, the basic framework of a digital signature scheme consists of two algorithms and a pair of keys (pk, sk), where pk is the signer's public key and sk is its secret key. This algorithm can identify every signer uniquely so that no user can deny his signature. The other algorithm is the verification algorithm available to everyone by using pk.

1.1.4.1 Undeniable Signatures

Undeniable signature [CVA90] introduced by Chaum and van Antwerpen is a digital signature which can only be verified with the help from the signer. In other words, verification procedure includes two *interactive* protocols: the one for confirmation of the valid signature and the one for denial of the invalid one. In a generalized version of this primitive called designated confirmer signature (DCS) [Cha94] proposed by Chaum, confirmation can be delegated to the third party called designated confirmer.

1.1.4.2 Designated Confirmer Signatures

Designated confirmer signatures (or just confirmer signatures, for short) are beneficial for the scenarios where the signer would like to control the process of verification. This property is important in a number of privacy protecting protocols ranging from software licensing [CVA90] to electronic auctions [SM00].

1.2 Zero-knowledge proofs

The first zero-knowledge proof was proposed by Goldwasser et al. in 1985[GMR85]. Here by we denote ZK as the abbreviation of zero-knowledge. This paper proposed the zero-knowledge protocol which had two parties, prover and verifier. Prover can prove an assertion to the verifier without providing any useful information. In 1991, Goldreich et al.[GMW91] gave an introduction on how to apply the zero-knowledge proof to all languages in NP. Their assertions guaranteed that the execution of the two parties protocol does not leak any information (in computational, or in statistical sense) to the cheating polynomial time verifier $\widetilde{P}$, who might decide on an arbitrary strategy for choosing her challenges. Zero-knowledge proofs are one aspect of the most useful tools in cryptography. The important result of zero-knowledge proofs is how to construct constant-round zero-knowledge proof systems[FS90, GK96]. Generally speaking, zero-knowledge proof is classified as follows:

1. Interactive zero-knowledge proof;
2. Non-interactive zero-knowledge proof[BFM88, RS92];
3. Multi-prover zero-knowledge proof.

1.2.1 Zero-Knowledge Identification

In identification protocol, there are two parties which are prover and verifier. The prover can identify his unique statement in a set, the verifier can check the correctness which prover want to prove. The identification protocol is to allow prover which can identify with control center in this protocol.

A zero-knowledge identification protocol requires to run the zero-knowledge proof to prove the statement about the secret key but does not give any information on it to other party. Note that assuming that one-way functions exist, one could achieve the results which presented in this work using general ZK proofs for NP-statements [GMW91], however such constructions would be prohibitively inefficient.

In order to build zero-knowledge identification scheme more effectively, we can use xLPN problem. The xLPN problem is equivalent to that of general decoding for the same parameters. More specifically, the only difference is that in xLPN, the generator matrix of the code is chosen uniformly at random rather than being full-rank. However, it is possible to choose the appropriate code parameters such that the probability for the uniform matrix to be full-rank is overwhelming.

Form constructing zero-knowledge identification scheme, we should emphasize we use interactive ZK proofs. According to an observation made in [GK05]: "... known constructions for non-interactive zero-knowledge proofs (NIZK) [DDN03] for NP languages (which are a

central tool in constructing CCA2 secure non-interactive public-key encryption given semantically secure public key encryption algorithms) require trapdoor permutations." At the same time, the hardness assumptions related to coding give rise to only trapdoor function candidates [McE78, Nie86].

Recently, Jain et al.[JKPT12] constructed efficient ZK proofs for NP-statements assuming hardness of (x)LPN problem. Their proposal is based on a proof of valid opening for their commitment scheme. Their proof of valid opening also implies a ZK identification scheme based on xLPN. Our zero-Knowledge identification schemes are based on McEliece PKE and Niederreiter PKE so that we can use the proof from Jain et al.

1.2.2 Proof of Plaintext Knowledge

Proof of Plaintext Knowledge(PPK) were first introduced by Aumann and Rabin [AR01] in the generic case of any PKE, and then investigated by Katz [Kat03], who presented efficient PPK for RSA, Rabin, ElGamal and Paillier PKE's. The first PPK in the post-quantum setting for the lattice-based Ajtai-Dwork PKE was presented by Goldwasser and Kharchenko [GK05], and with a number of works for the lattice-based schemes followed [XKT07, XT09, BD10]. It is worth that the works by Xagawa et al. [XKT07] and Xagawa and Tanaka [XT09]. [XKT07] successfully achieved PPK scheme based on [Reg05], and [XT09] use a modification of the NTRU identification scheme. The Stern scheme was also used by Kobara et al. [KMO08] for the similar purpose in a code-based oblivious transfer.

Recently, Morozov and Takagi [MT12] presented PPK and verifiable encryption for the McEliece PKE using the code-based Véron identification scheme [Vér97], which can be seen as a dual version of the Stern scheme [Ste96]. However, Jain et al. [JKPT12] pointed out a gap in the proof of the zero-knowledge property of the Véron scheme. This also created a gap in the proofs of both primitives in [MT12].

1.2.3 Verifiable Public-Key Encryption

Verifiable encryption was first introduced by Stadler [Sta96] as a tool for publicly verifiable secret sharing, and later generalized by Asokan et al. [ASW98] and applied to fair exchange of digital signatures. The works by Camenisch and Damgård [CD00] and by Camenisch and Shoup [CS03] made further advances on this topic. Verifiable encryption for the Ajtai-Dwork PKE was presented in [GK05].

1.3 Commitments

Commitment schemes find their applications, in particular, in zero-knowledge proofs. In the random oracle model, a random oracle itself can be readily used for commitments. In the standard model, one can construct commitments based on pseudorandom generators using Naor's paradigm [Nao90]. Jain et al. [JKPT12] presented an efficient commitment scheme which is directly based on xLPN, a variant of the learning parity with noise (LPN) problem, where the weight of the error vector is fixed [Pie12]. Note that the structure of the Jain et al. commitment scheme is essentially the same as that of the IND-CPA McEliece PKE [NIKM08], if the public key in the latter is substituted with a random matrix.

1.4 Our Contribution and Thesis Outline

In this thesis, our contribution are four-fold:
1. We show that the straight forward generalization of Stern's identification scheme to q-ary codes;
2. We give the first code-based proof of plaintext knowledge protocol;
3. We construct the first verifiable encryption based on the McEliece PKE;
4. We construct the first confirmer signature based on coding theory.

The first result is a construction has been made of the identification scheme by Cayrel et al.[CVA10] designed specifically, for q-ary codes. The research results was published in:
Rong Hu, Kirill Morozov, Tsuyoshi Takagi: On Zero-Knowledge Identification Based on Q-ary Syndrome Decoding. The 8th Asia Joint Conference on Information Security 2013: 12-18.
Next, we observe that Stern's identification scheme can be used to construct proof of plaintext knowledge protocol for code-based cryptosystem. Specifically, we design the proof of plaintext knowledge protocol for both McEliece and Niederreiter PKE. The research results were published in:
Rong Hu, Kirill Morozov, Tsuyoshi Takagi: Proof of plaintext knowledge for code-based public-key encryption revisited. The 8th ACM Symposium on Information, Computer and Communications Security 2013: 535-540.
We introduce a new approach for designing verifiable code-based encryption scheme, which combines two proof of plaintext knowledge protocols. This scheme allows the verifier to check if a ciphertext contains a given plaintext, without decrypting the ciphertext. The research results were published in:
Rong Hu, Kirill Morozov, Tsuyoshi Takagi: Zero-Knowledge Protocols for Code-Based Public-Key Encryption. The Institute of Electronics, Information and Communication Engineers.(accept with condition).

Finally, we present that the undeniable signatures and designated confirmer signatures. This thesis gives the first designated confirmer signature based on coding assumption. We should emphasize the fact that the designated confirmer signature can be easily degraded to undeniable signatures. The research results were published in:
Rong Hu, Kirill Morozov, Rui Zhang, Tsuyoshi Takagi: Confirmer Signatures from McEliece Assumptions. The 31st Symposium on Cryptography and Information Security 2014.

The remaining chapters of this thesis are structured as follows:
Chapter 2: Preliminaries. This chapter provides a background on the results and techniques from cryptography and coding theory used in the thesis.
Chapter 3: Zero-knowledge identification from codes. This chapter studies the identification schemes which are based on q-ary codes. We investigate their communication and computation costs. At last, we give the performance showing the advantages of our schemes.
Chapter 4: Proof of plaintext knowledge for McEliece and Niederreiter PKE schemes. Also, this chapter evaluates the performance of the different security parameters which run on our schemes.
Chapter 5: Verifiable code-based encryption. This chapter provides an analysis of verifiable code-based encryption from McEliece cryptosystem.
Chapter 6: Designated confirmer signatures from codes. This chapter focuses on the design of a designated confirmer signature scheme which is based on coding theory. This scheme argues non-transferability, unforgeability, security for the verifier, security for the signer, invisibility, INV-CMA, EUF-CMA and SEUF-CMA security.
Chapter 7: The conclusions, open questions and plans for future research are presented in this chapter.

Chapter 2

Preliminaries

2.1 Basic Notions

In this thesis, some classic notation will be sustained such as different kinds of numbers, say, the set of positive integers $\mathbb{N}$, the relative integers $\mathbb{Z}$, the real numbers $\mathbb{R}$ (The set of non negative real numbers is denoted by $\mathbb{R}^+$.). For generating an element x uniformly at random with a given set S, it will be written $x \in_\$ S$ or $x \xleftarrow{\$} S$. We use the set $\{0,1\}^*$ to generate the bit strings of arbitrary length. The bit strings $x \in \{0,1\}^n$ means that x be generate by the set $\{0,1\}^*$ with a fixed length n. The notation $|x|$ denotes its bit length.

Definition 2.1. We say one problem $\mathcal{P}_1$ **reduce**s to the problem $\mathcal{P}_2$ if there exists an oracle machine which solves the problem $\mathcal{P}_1$ using access to an oracle solving the problem $\mathcal{P}_2$.

Definition 2.2. We can also say that $\mathcal{P}_2$ is at least as hard as $\mathcal{P}_1$ and we denote this fact by $\mathcal{P}_2 \geq \mathcal{P}_1$.

2.2 Some Definitions of Computer Science

Turing machine. A Turing machine[Min67] is a hypothetical machine consisted of a tape with an infinite number of positions within a symbol. Every position can write and read symbols on the tape and shift left and right. Depending on the symbol read by the head and the internal state[Sha57], a transition function which orders the machine needs to execute the next operation. There are several work tapes of Turing machine but only one input tape. Probabilistic Turing machines can apply probabilistic algorithms to possess an additional tape called random tape containing uniformly random symbols (a symbol represents a bit.).
In this these, we will very often mention the term polynomial which means that the size of the input of a polynomial time algorithm. The complexity of a Turing machine is assumed that the

expected complexity over the random tape distribution. The complexity can also be measured by the number of steps executed by the Turing machine until it reaches the final state.

Definition 2.3. A function $f : \mathbb{N} \to \mathbb{R}^+$ is called negligible if for any polynomial $p : \mathbb{N} \to \mathbb{R}^+$ there exists an integer x_1 such that

$$n \geq x_1 \Rightarrow f(x) < \frac{1}{p(x)}.$$

Oracle machine. An additional term will also be referred which we call it oracle. This ways of access are able to perform a special functionality in associating the solution of a computational(hard) problem. Oracle, or oracle machine gets an access via a special tape on which could function as an input. Different from classic state, it has a special state "invocation", that is, the oracle input is replaced by its output. For instance, a factorization oracle would return the prime factors of a given integer.

Random oracles. A concept of idealized objects called random oracle was introduced in [FS87], and [BR93] to facilitate the security analysis of cryptographic schemes. A random oracle implements a uniformly distributed random function from $\{0,1\}^m$ to $\{0,1\}^n$ for some given positive integers m, n. The generator of oracle which is picked uniformly at random among all possible functions with a m-bit input and an n-bit output. In the model of random oracle, all participants have access to such idealized functions through this random oracles. Such an oracle can be achieved as follows. Firstly, a query function make a list which has the input-output pairs, it can be empty originally. When any new query come to the oracle, one checks whether it is already stored as input, if this is the right pair, one answers the corresponding output stored in the list. If it is not in the queue inside , one picks a uniformly random output of n bits and adds this new pair in the list.
One can makes security proof easier by using the random oracle because a challenge can be included to a hard problem in the simulation of random oracles. Although it has shown that a cryptographic scheme secure in the random oracle model may lead to insecure instantiations in the standard mode. It is commonly believed that a security proof in this model shows the well design of related schemes.

One-Way Functions. The property of one-way functions[KL14] is that they are easy to compute, but hard to invert. Since the popular computational assumption in cryptography that is almost always hard to solve in polynomial-time. So the definition of a polynomial-time adversary is formalized by saying the hard-to-invert requirement. For one-way function, the polynomial-time adversary will always fail to invert the function, except with negligible probability.We

emphasize that a one-way function is hard to invert if and only if the input is uniformly random.

Definition 2.4. (One-way function) The function picks up $\{0,1\}^n$ to $\{0,1\}^n$ and satisfies two conditions as follow:
1. Easy to compute: There exists a polynomial-time algorithm **M** such that on input any $x \in \{0,1\}^n$, M_f outputs $f(x)$ (i.e., $M_f(x) = f(x)$ for every x).
2. Hard to invert: For every probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function *negl* such that $\Pr[\mathcal{A}(f(x)) \in f^{-1}(f(x))] \leq negl(n)$.
where the probability is taken over the uniform choice of x in $\{0,1\}^n$ and the random coin tosses of $\mathcal{A}$.

Hard-Core Predicates. To prove our identification schemes and proof of knowledge schemes needs a notion called pseudorandom generators.

Definition 2.5. (Hard-core predicate):
1. The polynomial-time computable hard-core predicate *hc* function[KL14] picks up $\{0,1\}^n$ to $\{0,1\}$.
2. For every probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function *negl* such that

$$Pr[\mathcal{A}(f(x)) = hc(x) | x \leftarrow \{0,1\}^n] \leq \frac{1}{2} + negl(n)$$

where the probability is taken over the uniform choice of $x \leftarrow \{0,1\}^n$ and the random coin tosses of $\mathcal{A}$.

(Hash Functions) Hash functions plays a great role at the design of practical digital signatures and protocols. There many constructions of hash functions from different assumptions, in our protocol we will choose the efficient one. Now we describe hash function following the definition of Katz[Kat10].

Definition 2.6. (Hash functions) A Hash function is consisted of two probabilistic polynomial-time algorithms $(Gen; H)$ such that:
1. The probabilistic algorithm *Gen* picks up $\{0,1\}^m$ and outputs a key s. (We assume that $\{0,1\}^m$ is implicit in s.)
2. There exists a polynomial function l such that H with a key s and $x \in \{0,1\}^*$, and outputs a string $H_s(x) \in \{0,1\}^{l(m)}$). (Note that the running time of H is allowed to depend on $|x|$.)
If H_s is defined only for inputs $x \in \{0,1\}^{l'(m)}$), where $l(m) < l'(m)$ for any k, then we say that $(Gen; H)$ is a fixed-length hash function for inputs of length $l'(m)$.

For every hash function there are two basic security properties which define as follow:

Definition 2.7. (Collision-resistant) For every probabilistic polynomial-time adversary $\mathcal{A}$, a hash function $(Gen; H)$ is collision-resistant if the following property is negligible:

$$Pr[s \leftarrow Gen(1^k); (x, x') \leftarrow \mathcal{A}(1^k, s) := x' \wedge H_s(x) = H_s(x')]$$

Definition 2.8. (Universal one way) For every probabilistic polynomial-time adversary $\mathcal{A}$, a hash function $(Gen; H)$ is universal one way if the following property is negligible:

$$Pr[x \leftarrow \mathcal{A}(1^k); s \leftarrow Gen(1^k); x' \leftarrow \mathcal{A}(1^k, s) := x' \wedge H_s(x) = H_s(x')]$$

2.3 Public-Key Encryption

The **PKE** is one of the main important achievements of modern cryptography. It is suggest that readers should acknowledge them or refer to the famous textbook written by N. Koblitz [Kob94]. PKE transmit the public-key to communicate in a confidential way with the help of an authenticated channel. Any messages encrypted by public-key will only decrypted by the owner who has the corresponding secret key. Next we introduce the method based on Katz [KL14] to build a PKE scheme. We denote the message space by $\mathcal{M}$ and the ciphertext space by C.

Indistinguishability. With a view to expound public-Key encryption, primarily, we need two definitions [KW92] to describe two different kinds of indistinguishability. Our description is based on the definitions of Goldreich [Gol08].

Definition 2.9. (Probability Ensembles) A probability ensemble is a family of stochastic variable $\chi = \{X_i\}_{i\in\mathbb{N}}$ where $\{X_i\}$ is a probability distribution.

Definition 2.10. (Uniform Ensemble) A uniform ensemble $U = \{U_i\}_{i\in\mathbb{N}}$ is a distribution ensemble means that it uniformly distribute over strings of length n.

Definition 2.11. (Statistical Distance) The statistical distance Θ between two discrete random variables X_1 and X_2 with range φ is

$$\Theta(X_1, X_2) := \frac{1}{2} \sum_{x\in\varphi} |Pr[X_1 = x] - Pr[X_2 = x]|.$$

Next we need the same range φ to use define the notion of statistical and computational indistinguishability between two sequences of random variables.

Definition 2.12. (Statistical indistinguishability) Let $(\Psi_i)_{i\in I}$ and $(\Omega_i)_{i\in I}$ be two probability ensembles of discrete stochastic variables such that Ψ_i and Ω_i have the same range for any $i \in \mathbf{I}$. The ensembles $(\Psi_i)_{i\in I}$ and $(\Omega_i)_{i\in I}$ are statistical indistinguishable if

$$\Theta(\Psi_i, \Omega_i) = negl(|i|).$$

More specifically, if $\Theta(\Psi_i, \Omega_i) = 0$ for any $i \in \mathbf{I}$, it means that these ensembles are perfectly indistinguishable.

Definition 2.13. (Computational Indistinguishability) Let $(\Psi_i)_{i\in I}$ and $(\Omega_i)_{i\in I}$ be two ensembles of discrete random variable such that X_i and Y_i have the same range for any $i \in \mathbf{I}$. The ensembles $(\Psi_i)_{i\in I}$ and $(\Omega_i)_{i\in I}$ are computationally indistinguishable if for any probabilistic polynomial-time (with respect to $|i|$) algorithm **CI**.

$$|Pr[\mathcal{D}(i, \Psi_i) = 1] - Pr[\mathcal{D}(i, \Omega_i) = 1]| = negl(|i|).$$

Note that if the PKE cryptosystem need large index, The above notations can change to the case $\mathbf{I} = \mathbb{N}$. The negligible function *negl* takes $n \in \mathbb{N}$ as input which is easily converted into its length. This representativeness is included in the above general definitions if we represent any integer $n \in \mathbb{N}$ with a unary statement[Gol08]. A PKE scheme is composed of three polynomial time algorithms.

Definition 2.14. A public-key encryption scheme consists of the following three algorithms: **Setup(Key Generation)**, **Encryption**, and **Decryption**.

- **Setup** The key generator is a probabilistic algorithm which on input of the security parameter 1^k outputs a key pair. We have $(pk, sk) \leftarrow Setup(1^k)$.
- **Enc** The encryption algorithm is a probabilistic algorithm which takes a message $m \in \mathcal{M}$ and the public key pk as its input and outputs a ciphertext c. We have $c \leftarrow Enc(m, pk)$.
- **Dec** The decryption algorithm is a deterministic algorithm which takes a ciphertext $c \in C$ and the secret key sk to output a message $m \in \mathcal{M}$. We have $m \leftarrow Dec(c, sk)$.

Correctness. Pke require that any encrypted can be retrieved with the decryption algorithm. For any message $m \in \mathcal{M}$ and any pair (pk, sk), PKE can do the following functions: $c \leftarrow Enc(m, pk); m \leftarrow Dec(c, sk)$.
Next we present a security notion called indistinguishability under a chosen plaintext attack (IND-CPA as abbreviation) which is based on the word of [GM84]. This notion formalizes the fact that two ciphertexts of two different messages should be indistinguishable for any adversary which does not know sk. Here we assume the adversary is only given pk which allows him to encrypt any message m. In all features of indistinguishability, the weakest for an adversary can think of is to perform a chosen-plaintext attack.

IND-CPA Security Notion. For describing indistinguishability under chosen-plaintext attack, let us first consider an adversary $\mathcal{A}$ modeled by a probabilistic polynomial algorithm and the two following games corresponding to $b \in \{0, 1\}$.
(**Game**$^{ind-cpa-b}$.) First, $\mathcal{A}$ is fed by a public key pk generated by $(pk, sk) \leftarrow$ **Setup**(1^k).
After a given time, the adversary $\mathcal{A}$ submits two messages m_0, m_1.
The challenger answers $c \leftarrow Enc(m_b, pk)$. Then $\mathcal{A}$ outputs a bit $b^{'}$.

Definition 2.15. We define the advantage of the adversary as follows

$$Adv_{\mathcal{A}}^{ind-cpa} := |Pr[b^{'} = 1 \ in \ Game^{ind-cpa-1}] - Pr[b^{'} = 1 \ in \ Game^{ind-cpa-1}]|,$$

where the probabilisties are over the random tapes of the involved algorithms. We say that a public-key encryption scheme satisfied indistinguishability under a chosen plaintext attack (IND-CPA) if there exists no probabilistic polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{ind-cpa}$ is non-negligible.

IND-CCA1 Security Notion. For describing indistinguishability under chosen ciphertext attack,Let $\mathcal{A}$ be an adversary, which we model as an arbitrary non-uniform **PPT** machine (polynomial in the implicit security parameter of the encryption scheme). We define the following game played against A:
(**Game**$^{ind-cca1-b}$) $\mathcal{A}$ get a key according to the key generation algorithm: $(pk, sk) \leftarrow$ **Setup**(1^k). We choose a random bit b $\leftarrow \{0, 1\}$. $\mathcal{A}$ is allowed to query an oracle that computes the functionality Enc_K. Challenge: $\mathcal{A}$ outputs two messages, m_0, and m_1. Response: We give $\mathcal{A}$ the ciphertext $\mathsf{Enc}_K(m_b)$ A outputs b', (a guess from b). We say that the advantage of $\mathcal{A}$ in this experiment is $\Pr[b = b'] - 1/2$.

Definition 2.16. We define the advantage of the adversary as follows

$$Adv_{\mathcal{A}}^{ind-cca1} := |Pr[b^{'} = 1 \ in \ Game^{ind-cca-1}] - Pr[b^{'} = 1 \ in \ Game^{ind-cca-1}]|,$$

where the probabilisties are over the random tapes of the involved algorithms. We say that a public-key encryption scheme satisfied indistinguishability under a chosen ciphertext attack (IND-CCA1) if there exists no probabilistic polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{ind-cca-1}$ is non-negligible.

IND-CCA2 Security Notion. In the non-adaptive case (IND-CCA1), the adversary $\mathcal{A}$ does not make any decryption oracle calls after receiving the challenge ciphertext C. In contrast, in the adaptive case (IND-CCA2), the adversary $\mathcal{A}$ is allowed to make the decryption oracle calls even after receiving C under restriction that C is not submitted to the oracle.

2.4 Elements of Coding Theory

In this section, we explain the coding theory following the description of [MS77, Rot06]. In communications and information processing, we use code (error-correcting code) to recover the error happened caused by something like electromagnetic or light interference or simple channel noises. Error-correcting codes, by the feature of their structures, includes two classes, linear codes and nonlinear codes. For instance, for any codeword c_1 and c_2 in a linear code C, we have $c_3 = c_1 \oplus c_2$ also a codeword in C. While it is not true for a nonlinear code. In our thesis, we focus on linear codes.

2.4.1 Notions

For linear codes, there are two kinds of codes. One is the error-detecting code and the other is error-correcting code. Roughly speaking, error-correcting codes are usually strengthened version than the error-detecting codes. Since error-detecting codes only can detect the error Both of kinds can allow us to not only to detect errors but also to correct from a wrong state. Generally speaking, whatever kind of error-correcting code it is, the main idea to correct or detect errors is to add redundancy, which means the coder would put some more information bits to the origin codes. Error-detection and correction schemes can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm.

A binary (n, k)-code C is a k-dimensional subspace of the vector space $\mathbb{F}_2^n$, n and k are called the *length* and the *dimension* of the code, respectively. We call C an (n, k, d)-code, if its so-called *minimum distance* is $d := \min_{x,y \in C; x \neq y} d_H(x, y)$. For a generator matrix $G \in \mathbb{F}_2^{k \times n}$, we will denote the corresponding code as $C(G)$. For the relevant topics in the coding theory, we refer the reader to [Rot06, Mat80]. Suppose there is a telegraph wire from Boston to New York down which 0's and 1's can be sent. Usually when a 0 is sent it is received as a 0, but occasionally a 0 will be received as 1, or a 1 as a 0. Let us say that ont her average 1 out of every 100 symbols will be in error. i.e., for each symbol there is a probability $p = 1/1000$ that the channel will make a mistake. This is called a binary symmetric channel.

There are a lot of important messages to be sent ddown this wire, and they must be sent as quickly and reliably as possible. The messages are already written as a string of 0's and 1's-perhaps they are being produced by a computer.

We are going to encode these messages to give them some protection against errors on the channel. A block of k messages symbols $u = u_1 u_2 \dots u_k$ ($u_i = 0$ or 1) will be encoded into a codeword $x = x_1 x_2 \dots x_n$ ($x_i = 0$ or 1) where $n \geq k$; these codewords form a code.

The method of encoding we are about to describe produces what is called a linear code. The

first part of the codeword consists of the message itself:

$$x_1 = u_1, x_2 = u_2, \ldots x_k = u_k,$$

followed by $n - k$ check symbols

$$x_{k+1}, \ldots, x_n.$$

The check symbols are chosen so that the codewords satisfy

$$H \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = Hx^\top = 0, \tag{2.1}$$

where the $(n-k) \times k$ matrix H is the parity check matrix of the code, given by

$$H = [A|I_{n-k}], \tag{2.2}$$

A is the $(n-k) \times (n-k)$ unit matrix. The arithmetic in Equation (2.1) is to be performed modulo 2,, i.e. $0 + 1 = 1, 1 + 1 = 0, -1 = +1$. We shall refer to this as binary arithmetic.

Example 2.1. *The parity check matrix of a code*

$$H = \left(\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \tag{2.3}$$

defines a code with $k = 3$ *and* $n = 6$*. For this code*

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \tag{2.4}$$

The message $u_1u_2u_3$ is encoded into the codeword $x = x_1x_2x_3x_4x_5x_6$, which begins with the message itself:

$$x_1 = u_1, \quad x_2 = u_2, \quad x_3 = u_3,$$

followed by three check symbols $x_4x_5x_6$ chosen so that $Hx^\top = 0$, i.e. so that

$$\begin{aligned} x_2 + x_3 + x_4 &= 0, \\ x_1 + x_3 + x_5 &= 0, \\ x_1 + x_2 + x_6 &= 0. \end{aligned} \tag{2.5}$$

If the message is $u = 011$, then $x_=0, x_2 = 1, x_3 = 1$, and the check symbols are

$$x_4 = -1 - 1 = 1 + 1 = 2 = 0,$$

$$x_5 = -1 = 1, \quad x_6 = -1 = 1$$

so the codeword is $x = 011011$.
The equations (2.5) are called the parity check equations, or simply parity checks, of the code. We take (2.1) as our general definition:

Definition 2.17. (Parity Check Matrix) Let H be any binary matrix. The linear code with parity check matrix H consists of all vector x such that

$$Hx^\top = 0$$

where this equation is to be interpreted modulo 2.

It is convenient, but not essential, if H has the form shown in (2.2) and (2.3), in which case the first k symbols in each codeword are message or information symbols, and the last $n - k$ are check symbols.
Linear codes are the most important for practical applications and are the simplest to understand. In this thesis, the detail of nonlinear codes will not be inferred for the sake of concise.

Proposition 2.18. *For every two codewords $c_1, c_2 \in C$ and two scalars $a_1, a_2 \in GF(q)$, we have $a_1c_1 + a_2c_2 \in C$.*

Example 2.2. *The* $(3, 4, 2)$ *parity code over* $GF(2)$ *is a linear [3,2,2] code spanned by* $(1\,0\,1)$ *and* $(0\,1\,1)$.

Example 2.3. *The* $(3, 2, 3)$ *repetition code over* $GF(2)$ *is a linear* $[3, 1, 3]$ *code generated by*

$$G = (1 \quad 1 \quad 1).$$

In general, the $[n, 1, n]$ *repetition code over a field F is defined as the code with a generate metrix*

$$G = (1 \quad 1 \quad \dots \quad 1).$$

How to check a word is codeword $\in C$? We can use the parity check matrix.

- The definition again:$x = x_1x_2 \dots x_n$ is a codeword if and only if

$$Hx^\top = 0. \tag{2.6}$$

- Usually the parity check matrix H is an $(n-k)\times n$ matrix of the form

$$H = [A|I_{n-k}] \tag{2.7}$$

and as we have seen there are 2^k codewords satisfying (2.6). (This is still true even if H doesn't have this form, provided H has n columns and $n-k$ linearly independent rows.) When H has the form (2.7), the codewords look like as follows:

$$x = message\ symbols,\ check\ symbols = x_1 \dots x_k,\ x_{k+1} \dots x_n.$$

- If the message is $u = u_1 \dots u_k$, what is the corresponding codeword $x = x_1 \dots x_n$? First $x_1 = u_1, \dots, x_k = u_k$, or

$$\begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = I_k \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix}, I_k = unit\ matrix. \tag{2.8}$$

Then from 2.6 and 2.7, since

$$[A|I_{n-k}] \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = 0. \tag{2.9}$$

We have,

$$\begin{pmatrix} x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = -A \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = -A \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix}, \quad \text{from 2.8.} \tag{2.10}$$

In the binary case $-A = A$, but later we shall treat case where $-A \neq A$. put the 2.8 on top of 2.10:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{bmatrix} I_k \\ \hline -A \end{bmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix} \tag{2.11}$$

We get

$$x = uG \tag{2.12}$$

where

$$G = [I_k| - A^T] \tag{2.13}$$

We can easily obtained H from G.

Definition 2.19. (The Generator Matrix) G is called a generator matrix of the code for 2.12 we says that the codewrods are all possible linear combinations of the row of G. 2.6 and 2.12 together imply G and H are related by

$$GH^T = 0, \ or \ HG^T = 0. \tag{2.14}$$

2.4.2 Security of Linear Codes

Definition 2.20 (Gilbert-Varshamov Bound [Rot06])**.** Let C is a q-ary code with length n and minimum hamming weight d,$C_q(n,d)$ is the maximum possible size of C, we have $C_q(n,d) \geqslant \frac{q^n}{\sum_{i=0}^{d-1}\binom{n}{i}(q-1)^i}$. For simplicity, we can change the formula as follow:

Let $H_q(x) = x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x)$ be the q-ary entropy function. Let $d/n = \zeta$, and the rate of a q-ary linear code $R = k/n$. If $0 \leq \zeta \leq (q-1)q$, then for any n, there exists an (n,k,d) code such that

$$R \leq 1 - H_q(\zeta).$$

Definition 2.21. (Goppa codes [MS77]). A polynomial $G(x)$ called the Goppa polynomial, having coefficients from $GF(q^m)$, fix m and a subset $L = \{b_1, ..., b_n\}$ over $GF(q^m)$ ($G(b_i) \neq 0$ for all element form L.) We do not consider the case where the element form L is zero.
With any vector $a = (a_1, ..., a_n)$ over $GF(q)$, there is the rational function

$$R_a(x) = \sum_{i=1}^{n} \frac{a^i}{x - b^i}.$$

The Goppa code Γ consists of all vectors a such that

$$R_a(x) \equiv 0 \, mod \, G(x),$$

or equivalently such that $R_a(x) = 0$ in the polynomial ring $GF(q^m)[x]/G(x)$. If $G(x)$ is irreducible then Γ is called an irreducible Goppa code.

2.4.3 Hardness Assumptions

The hard problem for code-based cryptographic systems are based on the problem as follows. We will use assumption related to security of Niederreiter public key encryption

Definition 2.22. Syndrome Decoding (SD) Problem.

Input: $H \xleftarrow{\$} \mathbb{F}_q^{(n-k)\times n}$, $y \xleftarrow{\$} \mathbb{F}_q^{n-k}$ and $0 < \omega \in \mathbb{N}$.
Decide: If there exists $e \in \mathbb{F}_q^n$ s.t. $He^T = y$ and $w_H(e) \leq \omega$.

This problem was shown to be NP-complete by Berlekamp et al [BMVT78]. The best algorithm for solving it is the Information Set Decoding algorithm by Peters [Pet10].

We will use two assumptions related to security of McEliece public key encryption [McE78, Sen02].

Definition 2.23. General Syndrome Decoding (G-SD) Problem.

Input: $G \xleftarrow{\$} \mathbb{F}_2^{k\times n}$, $y \xleftarrow{\$} \mathbb{F}_2^n$ and $0 < t \in \mathbb{N}$.

Output: $x \in \mathbb{F}_2^k$, $e \in \mathbb{F}_2^n$ s.t. $w_H(e) \leq t$, $xG \oplus e = y$.

This problem was shown NP-complete by Berlekamp et al. [BMVT78].

Definition 2.24. Goppa Code Distinguishing(GCD) Problem.

Input: $H \in \mathbb{F}_2^{(n-k)\times n}$.

Decide: If H is a parity check matrix of an (n, k) irreducible Goppa code in Definition 2.21, or of a random (n, k)-code?

Currently, there are no polynomial algorithms for either of these problems [Sen09] in the general case. We remark that the Goppa Code Distinguishing Problem can be solved efficiently for high-rate Goppa code [FGUO+13]. However, a careful choice of code parameters will allow us to avoid the attack of [FGUO+13].

We assume both of the above problems to be hard.

2.4.4 Code-based PKE

2.4.4.1 McEliece PKE

The McEliece PKE consists of the following triplet of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with system parameters $n, t \in \mathbb{N}$:

• Key generation algorithm $\mathcal{K}$: On input 1^κ, choose the appropriate (n, k, t), and generate the following matrices:

– $G \in \mathbb{F}_2^{k\times n}$ – the generator matrix of an irreducible (n, k) binary Goppa code correcting up to t errors, where $k \geq n - t \cdot \log n$. Its decoding algorithm is denoted as $\mathsf{Dec}_{\mathcal{G}}$.

– $S \in \mathbb{F}_2^{k\times k}$ – a random non-singular matrix.

– $P \in \mathcal{S}_n$ – a random permutation matrix (of size n).

– $G^{pub} = SGP \in \mathbb{F}_2^{k\times n}$.

Output the public key $pk = (G^{pub}, t)$ and the secret key $sk = (S, G, P)$.

• Encryption algorithm $\mathcal{E}$: On input a plaintext $m \in \mathbb{F}_2^k$ and the public key pk, choose a vector $e \in \mathbb{F}_2^n$ of weight t at random, and output the ciphertext $c = mG^{pub} + e$.

• Decryption algorithm $\mathcal{D}$: On input c and the secret key sk, calculate:

– $cP^{-1} = (mS)G + eP^{-1}$.

– $mSG = \mathsf{Dec}_{\mathcal{G}}(cP^{-1})$.

– Let $J \subseteq \{1, \ldots, n\}$ be such that G_J is invertible.

Output $m = (mSG)_J(G_J)^{-1}S^{-1}$.

It is easy to check that the decryption algorithm correctly recovers the plaintext: Since in the first step of decryption, the permuted error vector eP^{-1} is again of weight t, the decoding algorithm $\mathsf{Dec}_{\mathcal{G}}$ successfully corrects these errors in the next step.
There some attacks can broken the McEliece cryptosystem, but we believe if we construct the McEliece cryptosystem based on the binary irreducible Goppa code, it is security [BLP08].

2.4.4.2 Niederreiter PKE

For a survey on the material presented in this and the next subsections, we refer the reader to [EOS07, OS09].

The Niederreiter PKE consists of the following triplet of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with the system parameters $n, t \in \mathbb{N}$:

• Key generation algorithm $\mathcal{K}$: On input 1^κ, choose the appropriate (n, k, t), and generate the following matrices:

– $H \in \mathbb{F}_2^{(n-k)\times n}$ – the parity check matrix of an irreducible (n, k) binary Goppa code which can correct at most t errors, where $k \geq n - t \cdot \log n$. The decoding algorithm of this code is denoted as $\mathsf{Dec}_{\mathcal{H}}$.

– $M \in \mathbb{F}_2^{(n-k)\times(n-k)}$ – a random non-singular matrix.

– $P \in \mathcal{S}_n$ – a random permutation matrix.

– $H^{pub} = MHP \in \mathbb{F}_2^{(n-k)\times n}$.

Output the public key $pk = (H^{pub}, t)$ and the secret key $sk = (M, H, P)$.

• Encryption algorithm $\mathcal{E}$: On input a plaintext $m \in \mathbb{F}_2^n$ such that $w_H(m) = t$, and the public key pk, output the ciphertext $c = H^{pub}m^T$.

• Decryption algorithm $\mathcal{D}$: On input a ciphertext c and the secret key sk, calculate:

– $M^{-1}c = (HP)m^T$.

– Since $(HP)m^T = H(Pm^T)$, use the decoding algorithm $\mathsf{Dec}_{\mathcal{H}}$ to recover Pm^T.

–Output $m^T = P^{-1}Pm^T$.

It is easy to check correctness of the decryption algorithm: After the first step of decryption, we obtain a syndrome of the permuted plaintext Pm^T. Since the decoding algorithm $\mathsf{Dec}_{\mathcal{H}}$ is known, it is easy to recover the plaintext.

We note that the plaintext space of the Niederreiter PKE is the set of weight-t binary vectors. For representation of arbitrary binary vectors (of an appropriate length) as a valid plaintext, see the work by Cover [Cov73] and its improvements by Sendrier [Sen05].

2.4.4.3 IND-CPA Variants

The hardness of SD problem and G-SD problem implies the one-wayness under chosen-plaintext attacks (OW-CPA) for the respective PKE's. This basic security notion states that any PPT adversary has only negligible probability in winning the following game. The challenger generates a public key pk and a uniformly random plaintext m, encrypts the latter into the ciphertext c, and submits pk and c to the adversary. The adversary wins, if she outputs m. The probability is taken over the random coins used to generate the key pair, the choice of m, and the randomness of the adversary.

Nojima et al. [NIKM08] introduced the efficient IND-CPA variants of the Niederreiter and the McEliece PKE's under hardness of the Syndrome Decoding and the LPN problems, respectively, plus hardness of the GD problem (in both cases). The IND-CPA variant is constructed by a uniform padding of the plaintext. We refer the reader to [NIKM08] for details.

In the standard model, Nojima et al. [NIKM08] show that the McEliece encryption with a random padding of the plaintext (which is multi-bit) is IND-CPA secure under hardness of the learning parities with noise (LPN) problem for a formal definition of LPN problem – it is similar to G-SD problem except that in the error vector e, each bit has Bernoulli distribution with fixed p, $0 < p < 0.5$. and SD problem. Let n, k be the integers such that $n \geq k$, then $\mathsf{Binary}(n, k)$ denotes a set of $\mathbb{F}_2^{k \times n}$ (i.e. the set of $(k \times n)$ binary matrices) of rank k.

The IND-CPA Niederreiter encryption is constructed in the same way, as described above, except that the ciphertext is computed as follows $c = H^{pub}(r|m)^T$, where $r \in \mathbb{F}^{n_0}$, $w_H(r) = t_0$, $m \in \mathbb{F}_2^{n_1}$, $w_H(m) = t_1$, and $n = n_0 + n_1$, $t = t_0 + t_1$.

The IND-CPA McEliece encryption is constructed in the same way as described above, except that the ciphertext $c = (r|m)G^{pub} + e$, where $r \xleftarrow{\$} \mathbb{F}_2^{k_0}$, $m \in \mathbb{F}_2^{k_1}$, $k = k_0 + k_1$.

We refer the reader to [NIKM08] for the security arguments and the recommended parameters.

2.5 Zero-Knowledge Proofs

From the practical point of view, it is preferable to construct zero-knowledge proofs which are secure against malicious verifier.

One can imagine various applications, where the receiver of the ciphertext might want to ensure in advance that the sender knows the plaintext inside it. For instance, in case of an auction, when the encrypted bid is sent, an adversary might intercept it and re-send it on his own behalf, in attempt to bring the auction to a draw. Clearly, deployment of PPK would prevent the above attack. One might argue that an authentication mechanism such as digital signature might be a cheaper solution, however it depends on a particular application, since PPK does *not* bind the bidder to the bid. Therefore, the interactive nature of PPK would allow the bidder to authenticate the bid, but deny it later for privacy reasons. One practical application of zero-knowledge PPK is a secure service for cloud storage, where an encrypted message from a sender P is first held by a cloud storage provider (playing a role of the verifier V) and then, after some time, transferred to the intended receiver. In this scenario, PPK allows the provider V to verify that the ciphertext coming from P is indeed valid. Hereby, a dishonest $\widetilde{P}$ cannot submit an invalid ciphertext, causing the receiver to blame V for mishandling the ciphertext. As compared to the use of digital signature, PPK allows V to verify correctness of the ciphertext, but it does not allow a dishonest $\widetilde{V}$ to convince anybody else that the ciphertext comes from P, due to the zero-knowledge property. An immediate application of this result is the interactive chosen-ciphertext secure encryption. Here, the sender uses an IND-CPA secure PKE to encrypt a message for the receiver, who must be online. Along with transmitting the ciphertext, the sender also uses the interactive PPK to convince the receiver that he knows the message. According to the observation by Katz [Kat03], this construction results in an interactive IND-CCA1 PKE [GHY86, Gol01]. Combined with the IND-CPA secure McEliece encryption by Nojima et al. [NIKM08], this yields the first code-based interactive IND-CCA1 PKE in the standard model.

2.6 Commitments

ZK proof systems require commitments. A commitment scheme consists of two phases: the first one is *commitment*, where a sender P provides a receiver V with an evidence about an input m. The cheating receiver $\widetilde{V}$ cannot learn m at this stage – this property is called *hiding*. In the

second phase, called *opening*, P reveals m to V. The cheating sender $\widetilde{P}$ cannot successfully any other message apart from m – this property is called *binding*. Let us denote by $[P, V]_{A,st}$ the *view* of the party $A \in \{P, V\}$ at the stage st, which is a concatenation of all the messages sent and received by A, along with its local randomness.

We denote by $Com(x_1, x_2, \ldots)$ a commitment to values $(x_1, x_2, \ldots)$. For simplicity, we henceforth denote a commitment to a value x, respectively a pair of values (x, y), by $Com(x)$, respectively $Com(x, y)$, and omit the mentioning of the public commitment key.

Definition 2.25. A triple of algorithms $(\mathsf{KGen}, com, \mathsf{Ver})$ is called a commitment scheme, if the following holds:

- On input 1^κ, the key generation algorithm KGen outputs a public commitment key pk.
- The commitment algorithm com takes as inputs a message m from a message space $\mathcal{M}$ and a commitment key pk, and outputs a commitment/opening pair (c, d).
- The verification algorithm Ver takes pk, m, a commitment c and an opening d and outputs 1 or 0.

The commitment scheme is secure, if it satisfies the following three properties. We present them somewhat informally, and refer the reader to [Dam99, Sch] for details.

- *Correctness*: Ver evaluates to 1 whenever the inputs were computed by honest parties P and V, i.e., for any $m \in \mathcal{M}$, given $pk \xleftarrow{\$} \mathsf{KGen}(1^\kappa)$, and $(c, d) \xleftarrow{\$} com(m, pk)$, we have that $\mathsf{Ver}(pk, m, c, d) = 1$.
- *Binding*: For any $\widetilde{P}$, the probability of generating (d, d') satisfying

$$(\mathsf{Ver}(pk, m, c, d) = 1) \wedge (\mathsf{Ver}(pk, m, c, d') = 1)$$

  must be negligible over the random coins of KGen and $\widetilde{P}$.
- *Hiding*: For any $m, m' \in \mathcal{M}$, and $(c, d) \xleftarrow{\$} com(m, pk)$ and $(c', d') \xleftarrow{\$} com(m', pk)$, the distributions of c and c' are indistinguishable, where the random variables are over the random coins of com.

If $\widetilde{P}$ is restricted to run in PPT, then the binding is called *computational* and otherwise it is *statistical*. The hiding is called *computational*, respectively *statistical*, if in its definition, the indistinguishability is computational, respectively statistical. We adapt the following definition from [JKPT12].

Definition 2.26. A protocol is said to *securely implement string commitment*, if at the end of its execution by PPT Turing machines P (with input $b \in \mathbb{F}_2^l$, $l \in \mathbb{N}$) and V, the following properties hold:

(Correctness) $\Pr[\langle P(b), V\rangle = 1]$ with overwhelming probability.

(Hiding) For any PPT $\widetilde{V}$, any $l \in \mathbb{N}$, any $b \in \mathbb{F}_2^l$ and $b' \in \mathbb{F}_2^l$ such that $b' \neq b$, after the committing stage, but before the opening stage, the distributions

$$[P(b), \widetilde{V}]_{\widetilde{V},Commit} \quad \text{and} \quad [P(b'), \widetilde{V}]_{\widetilde{V},Commit}$$

are indistinguishable. Depending of the type of indistinguishability, hiding can be *statistical* or *computational*.

(Binding) For any $\widetilde{P}$, any $l \in \mathbb{N}$, any commitment can be successfully opened in a unique way, i.e.,

$$\langle \widetilde{P}(b), V\rangle$$

is non-negligible. If $\widetilde{P}$ is restricted to run in PPT, then binding is called *computational*, if $\widetilde{P}$'s computing power is not restricted, then the binding is *statistical*.

In the standard model, an efficient computationally hiding and statistically binding commitment commitment schemes is presented by Jain et al. [JKPT12].

The fact that the commitment is statistically binding, allows the prover in our zero-knowledge proof system to be computationally unbounded.

Note that committing to binary vectors does not pose a problem, since mathematical objects, which we are working with, will eventually be represented as binary vectors for the actual implementation.

In the random oracle model (ROM) [BR93], a string commitment which is both computationally hiding and binding can be implemented using (idealized) cryptographic hash function. $h : \{0,1\}^* \to \{0,1\}^{l_c}$, for some fixed l_c. In order to commit to a binary vector x, one computes $Com(x) = h(r\|x)$, where "‖" denotes concatenation, and $r \xleftarrow{\$} \{0,1\}^{l_r}$, for some fixed l_r. Then, the opening simply consists of announcing (r, x). In the standard model, a computationally hiding and statistically binding code-based commitment commitment schemes are known [DvdGMQN08, JKPT12].

Perfectly Binding and Computational Hiding Commitments from Coding Theory.
It is necessary for us to study the commitments function based on coding theory which have the properties perfectly binding and perfectly hiding. A theorem for proof perfectly binding and computational hiding is presented in this section. An outline of the research from the definition

[JKPT12] that is we can make the perfectly binding and computational hiding commitments based on SD problem and G-SD problem. We strongly recommend readers get more details from [JKPT12].

2.7 Code-Based Identification Scheme

The first practical identification scheme based on coding theory is that the Stern's scheme [Ste96] schemes. This scheme is valid for any code, for which SD problem is hard, not just a random one. Clearly, the SD problem implies hardness of decoding of the code represented by the parity check matrix H.

Note that in the following protocol, the probability for $\widetilde{P}$ to past the protocol (i.e. to make V accept the proof without knowledge of the witness (m, e)) at least $2/3$. This scheme can be reduced to an arbitrary small value $(2/3)^s$ by running the protocol for s times.

Witness: s, $s \in \mathbb{F}_2^k$, t an integer, where the parameters n, k, t are described in Section 2.4.1.

Common data: $(H \in \mathbb{F}_2^{(n-k)\times n}, t)$ – the parity check matrix, and $i = Hs^T$ – the syndrome of message s.

1. P chooses randomly a word $y \in \mathbb{F}_2^n$ of n bits and a permutation σ of $\{1, 2, ..., n\}$ and sends three commitments:
   – $C_1 = com(\sigma|Hy^T)$,
   – $C_2 = com((y\sigma)$,
   – $C_3 = com((y \oplus s)\sigma)$.
2. V sends $b \overset{\$}{\leftarrow} \{0, 1, 2\}$.
3. In this step, V checks the validity of the quantities presented by P, and rejects if it does not hold:

- If $b = 0$,
  – P sends y and σ, and opens C_1, C_2.
  – V checks validity of C_1 and C_2, since H is public.
- If $b = 1$,
  – P sends $(y \oplus s)$ and σ, and opens C_1, C_3.
  – V checks that the validity of C_2, C_3
  (using that $Hy^T = (y \oplus s)^T + i$).

- If $b = 2$,
  - P sends $y\sigma$, $s\sigma$, and opens C_2, C_3.
  - V checks the validity of C_1, C_3 by using $y\sigma \oplus s\sigma = (y \oplus s)\sigma$,and check that $wt(s) = t$.

2.8 Proof of Plaintext Knowledge

We borrow some parts of the presentation in this section from [MT12]. All of these ll of these concepts will be used in section 4. The Hamming distance between $x, y \in \mathbb{F}_2^n$ (i.e. the number of positions where they differ) is denoted as $d_H(x, y)$. The distance of $x \in \mathbb{F}_2^n$ to the zero-vector 0^n denoted by $w_H(x) := d_H(x, 0^n)$ is called the *weight* of x. Henceforth, we will write $\nvdash$ for the zero-vector of an appropriate length which will be clear from the context. For $x, y \in \mathbb{F}_2^n$, $x + y$ will denote the bitwise exclusive-or. Let J be an ordered subset as follows: $\{j_i, \ldots, j_m\} = J \subseteq \{1, \ldots, n\}$, then we denote a vector $(x_{j_1}, \ldots, x_{j_m}) \in \mathbb{F}_2^m$ by x_J. Similarly, we denote by $M_J \in \mathbb{F}_2^{k \times |J|}$ a restriction of the matrix $M \in \mathbb{F}_2^{k \times n}$ to the columns with indices in J. A concatenation of matrices $X \in \mathbb{F}_2^{k \times n_0}$ and $Y \in \mathbb{F}_2^{k \times n_1}$ is written as $(X|Y) \in \mathbb{F}_2^{k \times (n_0 + n_1)}$, and for $k = 1$ this will denote concatenation of vectors. We denote by $x \xleftarrow{\$} \mathcal{X}$ a uniformly random sampling of an element from its domain $\mathcal{X}$. A set of $(n \times n)$ permutation matrices is denoted by $\mathcal{S}_n$. By $\mathsf{Binary}(k, n)$, we denote the set of the matrices in $\mathbb{F}_2^{k \times n}$ of rank k. All the logarithms are to the base 2, unless otherwise stated. If A is a PPT algorithm which on inputs $(x_1, \ldots, x_n)$ computes the outputs $(y_1, \ldots, y_n)$, we write it as $(y_1, \ldots, y_n) \xleftarrow{\$} \mathsf{A}(x_1, \ldots, x_n)$.

We model a party taking part in an interactive two-party protocol as an interactive Turing machine. We denote by $\langle A(a), B(b)\rangle(c)$ a random variable representing the output of a party B following an execution of an interactive two-party protocol between a party A with private input a and B with private input b on a joint input c, where A and B have uniformly distributed random tapes. If a party, say A, has no input, then we omit it by writing just A (instead of $A(a)$) in the above notation. In our two-party protocols, we will denote an honest prover by P and an honest verifier by V, while a dishonest party will be denoted by $\widetilde{P}$ and $\widetilde{V}$, respectively.

We call a function $\epsilon(\kappa)$ *negligible in some parameter* κ, if $\epsilon(\kappa) = 2^{-\omega(\log \kappa)}$. We call a probability $1 - \epsilon(\kappa)$ *overwhelming*, when $\epsilon(\kappa)$ is negligible. Occasionally, we may omit mentioning the security parameter, when it is clear from the context. In these cases, by saying that a quantity is negligible (overwhelming), we mean that it is *negligible (overwhelming) in the security parameter*.

We adapt the following definition from [Kat03]. For a PKE scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, denote by $c = \mathcal{E}_{pk}(m; R)$ a ciphertext of a plaintext m under the public key pk using the randomness R. We will call (m, R) a *witness* to the decryption of c under pk. Informally, in a PPK protocol, a sender P

proves to a receiver V the knowledge of a witness to the decryption for some ciphertext c under the known public key pk.

Definition 2.27. Let $\Pi = (P, V)$ be a tuple of PPT algorithms. Π is a *proof of plaintext knowledge* for an encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ if the following conditions hold:

(Completeness) For any pk output by $\mathcal{K}(1^\kappa)$ and any c with witness w to the decryption of c under pk, we have that $\langle P(w), V\rangle(pk, c) = 1$ (when V outputs 1 we say it *accepts*).

(Soundness) For a public key pk output by $\mathcal{K}(1^\kappa)$, a ciphertext c produced under such the pk, and for any expected PPT $\widetilde{P}$, we have that $\Pr[\langle \widetilde{P}, V\rangle(pk, c) = 1]$ is negligible, where the probability is taken over the random coins of $\mathcal{K}$, and the random tapes of both $\widetilde{P}$ and V.

(Zero-knowledge) There exists a PPT Turing machine $\mathcal{SIM}$ (called a *simulator*) such that, for any pk output by $\mathcal{K}(1^\kappa)$, any PPT $\widetilde{V}$, and any w, the following distributions are indistinguishable:

$$\{c = \mathcal{E}_{pk}(m; R) : \langle P(w), \widetilde{V}\rangle(pk, c)\},$$

$$\{c = \mathcal{E}_{pk}(m; R) : \langle \mathcal{SIM}, \widetilde{V}\rangle(pk, c)\},$$

where the probability is taken over the random tapes of both P and $\widetilde{V}$. In the case of *statistical*, respectively *computational* indistinguishability, we call the property *statistical*, respectively *computational* zero-knowledge (ZK).

2.9 Verifiable Public-Key Encryption

Definition 2.28. Let $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme, let R be a binary relation and let $L_R = \{x | \exists w : (x, w) \in R\}$. A *secure verifiable encryption* scheme for a relation R consists of a two-party protocol between P and V such that the following conditions hold:

(Completeness) For any pk output by $\mathcal{K}(1^n)$ and any $x \in L_R$, we have $\langle P(x), V\rangle(pk, c) = 1$ (when V outputs 1 we say it *accepts*.)

(Soundness) For any pk output by $\mathcal{K}(1^n)$, any $x' \notin L_R$, and for any expected PPT $\widetilde{P}$, $\Pr[\langle \widetilde{P}(x'), V\rangle(pk, c) = 1]$ is negligible, where the probability is taken over the random tapes of both $\widetilde{P}$ and V.

(Zero-knowledge) There exists a PPT simulator $\mathcal{SIM}$ such that, for any pk output by $\mathcal{K}(1^n)$, any PPT $\widetilde{V}$, and any $x \in L_R$, the following distributions are computationally indistinguishable:

$$\{x \in L_R : \langle P(x), \widetilde{V}\rangle(pk, c)\},\ \{x \in L_R : \langle \mathcal{SIM}, \widetilde{V}\rangle(pk, c)\},$$

where the probability is taken over the random tapes of both P and $\widetilde{V}$.

Note that this definition captures only the properties related to verifiability. We implicitly assume that a scheme in question is indeed a public-key encryption scheme. For a formal definition, see e.g. [Gol04, Ch. 5].

2.10 Digital Signatures

During the past decades, the digital signatures have been implemented using in internet, personal computer and so on. The digital signatures are the online/offline protocols which contain a signer and the other a set of potential verifier. For signing the message, the signer will use a secret key from himself, the message after sign and be published we call it as digital signature. In the standard model, any potential verifier can check this signature by using public key. To check the signature, there are some trusted parts we call Public Key Infrastructures (PKI) who can identify a valid digital certificate, all of them can publish the public key. We have a lot of ways to construct the digital signatures. For example, due to the contribution of Rivest, Shamir, and Adleman in [RSA78], one of digital signatures was proposed by using the RSA trapdoor one-way permutation. From the above description, we can see the digital signatures may create by PKE schemes like identification scheme.
A digital signature scheme consists of the three following polynomial-time algorithms[Kat10].

- **Setup** This probabilistic algorithm generates a key pair which is associated with the signer. We have $(pk, sk) \leftarrow Setup(1^k)$.
- **Sign** This probabilistic algorithm generates a signature for a given message $m \in \mathcal{M}$ with respect to the above generated key pair. We have $\delta \leftarrow Sign(m, sk)$.
- **Verify** The verification algorithm is usually deterministic and takes a message signature pair $(m, \delta) \in \mathcal{M} \times \Sigma$ and the public key pk as input and outputs with repect to the key pair (pk, sk). The output bit 1 means that the signature is valid.

It is supposed that the signer should transmit his public key pk to the verifier through an authenticated channel so that the verifier is ensured that pk really corresponds to the right signer. Once this step is performed,, the signer can authenticate messages using digital signatures even through an insecure communication channel.
Here we introduce the notion *existential unforgeability* under an adaptive message attack. It was first put forward by Goldwasser, Micali, and Rivest in [GMR88].

2.10.1 Existential Unforgeability Digital Signatures

To obtain a code-based EUF-CMA signature[Kat10], we propose to apply Fiat-Shamir transform[FS87, Kat10] to a code-based identification scheme by Stern[Ste96] or to that by Jain et al.[JKPT12]. We use the definition from Katz's book[Kat10] as following transform. **Fiat-Shamir Transform.**

Definition 2.29. Let$\Xi = (Gen; \mathbf{P}; \mathbf{V})$ be a standard identification scheme where the verifier's challenges are chosen uniformly from Ψ. Let $H : \{0,1\}^* \rightarrow \Psi$ be a hash function.
Key generation: Run $Gen(1^k)$ to generate keys (pk, sk). These are the public and secret keys, respectively.
Signature generation: To sign message m using secret key sk, do: 1. Run the prover algorithm $\mathbf{P}(sk)$ to generate an initial message I. 2. Compute $c := H(I; m)$. 3. By using$P(sk)$, Compute the appropriate response r to the challenge c. The signature is (I, r).
Signature verification: To verify the signature (I, r) on message m with respect to public key pk, proceed as follows: 1. Compute $c := H(I; m)$. 2. Accept the signature iff (pk, I, c, r) is an accepting transcript.

Any EUF-CMA (code-based) digital signature can be used. A number of code-based signature algorithms exist [KKS97, CFS01, GG07a, CM10, GS12] but their security remains a matter of dispute to various extents [SMEYA11, Fin11a, Fin11b].

A safe (but not the most efficient) instantiation is the construction based on Fiat-Shamir transform using a zero-knowledge identification scheme.

Let $\Pi = (KeyGen, P, V)$ be a canonical identification scheme that is secure againse a passive attack. Since Fiat-Shamir transform can only be proven secure in the random oracle model, we are forced to take this assumption for our construction as well. There are existentially unforgeable under an adaptive chosen message attack signature scheme Σ from the Fiat-Shamir transform based on random oracle model[Kat10].

Existential Unforgeability. First we introduce O a signing oracle. Precisely speaking, O is a kind of machine which on any message m sent to it answers a valid signature, i.e., it implements the algorithm *sign*. We denote by L the list of all messages queried to O. A signature scheme is secure against an existential forgery under an adaptive chosen-message attack, if for any probabilistic polynomial time forger(algorithm) $\mathcal{F}$, we have

$$Pr[1 \leftarrow Verify(m, \delta, pk) \wedge m \notin L | \begin{matrix}(pk, sk) \leftarrow Setup(1^k); \\ (m, \delta) \leftarrow \mathcal{F}^O(pk)\end{matrix}] = negl(k),$$

where the probability is taken over the random tapes of the involved algorithms. Each invocation to the oracle O is counted in the complexity of $\mathcal{F}$ so that the number of queries made to the O must also be polynomially bounded in k.

2.10.2 Undeniable Signatures

Undeniable signature schemes, devised by David Chaum and van Antwerpen [CVA90], are non-self-authenticating signature schemes. These signatures can only be verified with the communication from signer. However, if a signature is only verifiable with the aid of a signer, a dishonest signer may deny the signature which he published before. Undeniable signatures can solve this problem after adding a new component called the disavowal protocol.

After one years, David Chaum, Eugéne van Heijst and Birgit Pfitzmann [CvHP91] show an important property of undeniable signature, where the security of the Signer must be unconditional. The scheme is implemented using public-key cryptography based on the discrete logarithm problem. The signature part of the scheme is similar to other discrete logarithm signature schemes. In this scheme, the probability of a dishonest signer which try to pass the protocol that is only $1/p$ where p is the prime number depend on the signer's private key. If the protocol use the large size private key for security, so the computation of signer. We can see, if the signer does not have powerful computation ability, she will ask the other party to help her, it may the confirmer.

2.10.3 Designated Confirmer Signatures

Undeniable signatures, whose verification requires the cooperation of the signer in an interactive protocol, where are widely used for constructing the cryptosystems. From the above section, we can see sometimes the signer need the other party to help him. Designated confirmer signatures were introduced as an improvement of undeniable signatures where the signer becomes unavailable or powerless in undeniable signatures. One of attention of this thesis is how to design and analysis of designated confirmer signatures. Firstly, we can achieve an efficient designated confirmer signature by using Fiat-Shamir transform, and such a construction is secure in the random oracle model under code-based hard problems. Secondly, we build a generic DCS scheme which based on El Aimani's DCS scheme[EA10] by constructing the confirmer's ability to disavow on code-based cryptography. The new generic DCS scheme is proved to be secure in the standard model, and can be implemented to obtain an efficient instantiation.

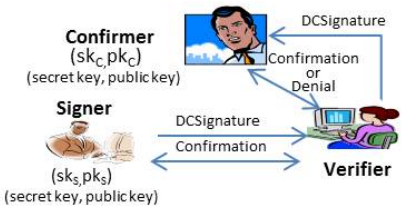

Figure 2.1

2.10.3.1 Syntax

Key generation: Generates probabilistically key pairs (K^S_{pk}, K^S_{sk}) and (K^C_{pk}, K^C_{sk}) for the signer and for the confirmer respectively, consisting of the private and the public key.
ConfirmerSign: On input (K^S_{sk}, K^C_{pk}) and a message m, outputs a confirmer signature μ, then interacts with the signature recipient to convince him of the validity of the just generated signature.

Confirm/Denial: These are interactive protocols between the confirmer and a verifier. Their common input consists of, in addition to K^S_{pk} and K^C_{pk}, the alleged signature μ, and the message m in question. The confirmer uses his private key K^C_{sk} to convince the verifier of the validity(invalidity) of the signature μ on m. At the end, the verifier either accepts or rejects the proof.

Selective conversion: This is an algorithm run by the confirmer using K^C_{sk}, in addition to K^C_{pk} and K^S_{pk}. The result is either $\perp$ or a string which allow ied as a valid digital signature. Some models, e.g. [Wik07], require that the confirmer issues a protocol of the correctness of the conversion incase of a valid signature.
Selective verification: This is an algorithm for verifying converted signatures. It inputs the converted signature, the message and K^S_{pk} and outputs either 0 or 1.

2.10.3.2 Security Model

In this section, we follow [GMR05, WBWB07, Kat10] for the formal definition.
Since designated confirmer signature construction, we show three informal properties corresponding three parties as follow.

Security for the verifier. This property means that an adversary who compromises the private keys of both the signer and the confirmer cannot convince the verifier of the validity of an invalid confirmer signature. In the other hand, the adversary cannot make the verifier to verify invalidity of a valid confirmer signature.
Non Transferability. This property requires that the transcript resulting from the interaction of

the verifier with the signer/confirmer during these protocols is indistinguishable from the transcript resulting from the interaction of the verifier with a simulator (which can be rewound) which does not have the private inputs of the signer/confirmer but is allowed to one oracle call to learn the validity/invalidity of the alleged signature w.r.t. the message in question. We refer to [CM00] for the formal definition (after considering the fix proposed by [Wik07], namely, the possibility of rewinding the simulator).

Unforgeability. It is defined through the following game: the adversary **A** is given the public parameters of the DCS scheme, namely pk_s and pk_c, in addition to sk_c. **A** is further allowed to query the signature on polynomially many messages, say q_s. At the end, **A** outputs a pair consisting of a message m, that has not been queried yet, and a string μ. A wins the game if μ is a valid confirmer signature on m. We say that a DCS scheme is (t, ϵ, q_s) – **EUF-CMA** secure if there is no adversary, operating in time t, that wins the above game with probability greater than ϵ.

Strong Unforgeability. It is defined through the following game: the adversary **A** is given two tuples of the public parameters of the DCS scheme, one is namely pk_s and pk_c, in addition to sk_c. the other one is pk'_s and pk'_c, and sk'_c. In this case, **A** will use sk'_c to sign the signature and get a new signature. After then, **A** is allowed to query the new signature on polynomially many messages, say q'_s. At the end, A outputs a pair consisting of a message m, that has not been queried yet, and a string μ'. A wins the game if μ' is a valid confirmer signature on m. We say that a DCS scheme is (t, ϵ, q'_s) – **SEUF-CMA** secure if there is no adversary, operating in time t, that wins the above game with probability greater than ϵ.

Invisibility. Invisibility against a chosen message attack **(INV-CMA)** is defined through the following game between an attacker **A** and his challenger **R**:

After **A** gets the public parameters of the scheme from **R**, he starts Phase 1 where he queries the signing, confirmation/denial, selective conversion oracles in an adaptive way. Once **A** decides that Phase 1 is over, he outputs two messages m_0, m_1 that have not been queried before to the signing oracle and requests a challenge signature $\mu^{'}$. **R** picks uniformly at random a bit $b \in 0, 1$. Then, $\mu^{'}$ is generated using the signing oracle on the message m_b. Next, **A** starts adaptively querying the previous oracles (Phase 2), with the exception of not querying m_0, m_1 to the signing oracle and $(m_i, \mu^{'}), i = 0, 1$ to the confirmation/denial and selective conversion oracles. At the end, **A** outputs a bit $b^{'}$. He wins the game if $b = b^{'}$. We define **A** advantage as $adv(\mathbf{A}) = |\ \mathbf{Pr}[b = b^{'}] - \frac{1}{2}\ |$. We say that a DCS scheme is $(t, \epsilon, q_s, q_v, q_{sc})$ –**INV-CMA** secure if no adversary operating in time t, issuing q_s queries to the signing oracle, q_v queries to the confirmation/denial oracles and q_{sc} queries to the selective conversion oracle wins the above game with advantage greater that ϵ.

Anonymity of signatures. In some applications, it is required that the confirmer signatures are anonymous,i.e., do not leak the identity (public key) of the signer.We refer to [GM03] for the

formal definition of anonymity of confirmer signatures under a chosen message attack (ANO-CMA).

Chapter 3

Zero-Knowledge Identification from Codes

3.1 Background

Identification protocols serve the goal of entity authentication. Their applications include authentication and access control services such as remote login, credit card purchases and many others. Usually, these are interactive two-party protocols, where one party (called a *prover*) wants to prove a possession of some private identification information to another party (called a *verifier*).

In the public-key setting, on which we focus in our work, most of practically used schemes are challenge-and-response. In this case, *zero-knowledge (ZK)* identification schemes have an advantage in the sense that no information on the private key is released to the verifier. If the eavesdropper observes the communication between the prover and the verifier, then clearly she does not gain any information on the private key as well. Such the scheme is usually constructed as a ZK proof of knowledge with private key as a witness. This approach was pioneered by Fiat and Shamir in [FS87]. We refer the reader to [Sti05, Ch. 9] for more information on identification protocols.

We focus on code-based identification protocols because their security is based on hardness of decoding random codes – the problem which is not known to have an efficient solution even using quantum computation. Although quantum computers remain at the early prototype stage of development, it is desirable to have a secure *postquantum* alternative for the schemes based on hardness of discrete logarithm or integer factorization [Sti05, Ch. 9]. In Niederreiter cryptosystem we use the random irreducible Goppa code, so we can use pseudorandom generator to construct the commitment functions which they are not trapdoor. So the adversary who do not

know the secret key he will fact the syndrome decoding problem. It means that, until now, there is no polynomial time algorithm to decode a Niederreiter cryptosystem. Compare other public key cryptosystem, the speed of Niederreiter cryptosystem encryption is close to pseudorandom permutation, for instance, the block ciphers.

To break coding assumption, it also try to solve the problem of decoding a random code, there are some algorithms attempt doing this matter, but all these algorithms are exponential[Bar98]. Our scheme is based on hardness of Syndrome Decoding – a well studied problem – see e.g. [LB88, Leo88, Ste89, CCC98, Sen02, BJMM12]. Currently, the most efficient attack against the parameters relevant to our scheme is Information Set Decoding (ISD) [Pet10].

3.1.1 Related Works

The first code-based zero-knowledge identification protocol was proposed by Harari [Har88] in 1988, however Véron showed it insecure [Vér95], and only recently Malek and Miri [MM12] fixed the problem. The first secure zero-knowledge identification protocol based on coding was presented by Stern [Ste96]. It is a 3-pass protocol with soundness error $2/3$, based on hardness of syndrome decoding of binary codes. Girault showed a 3-pass identification scheme [Gir90], but it was not practical. Véron proposed a protocol [Vér97] based on (binary) General Decoding problem (a dual of Syndrome Decoding) but this scheme was recently shown insecure by Jain et al [JKPT12] who also presented a secure alternative. Gaborit and Girault [GG07b] proposed, in particular, a q-ary code based authentication scheme, but it was based on specific double-circulant binary codes. Kawachi et al proposed a q-ary identification scheme in the context of lattices, which is similar to ours [KTX08, Xag10]. Xagawa and Tanaka [XT09] modified the scheme [KTX08] to get a proof of plaintext knowledge for NTRU public key encryption. Recently, Cayrel, Véron and Alaoui [CVA10] presented a ZK identification scheme (we will call it the CVA scheme) based on syndrome decoding of codes over $\mathbb{F}_q$ ($q > 2$) where the soundness error is reduced to $\frac{q}{2(q-1)}$, which is essentially $1/2$ for large q.

3.1.2 Contribution of this Chapter

The main motivation for our work was to construct the code-based zero-knowledge identification scheme for small $q > 2$, this is also a new q-ary code based scheme based on stern's ID scheme. Such schemes may be applied for instance for proof of plaintext knowledge [MT12] for public-key encryption based on codes over $\mathbb{F}_q$ [BLP10].

We constructed a generalization of Stern scheme for q-ary case with soundness error $2/3$ and confirmed that its communication cost is superior as compared to that of CVA scheme for $q = \{3, 4\}$.

In particular, let us consider the 80-bit equivalent security level, and overall soundness error 2^{-16}. Then for $q = 3$, our proposed scheme has 28 rounds and communication cost of 4.79 kilobytes against that of 39 rounds and 7.50 kilobytes, respectively, with CVA scheme. When $q = 4$, both schemes have 28 rounds, and we have 4.33 kilobytes with our scheme against 4.69 kilobytes with CVA scheme.

It is worth noting, however, that already for $q \geq 5$, CVA scheme comes on top in terms of communication cost. In particular, for $q = 5$, our proposal required 28 rounds and 5.08 kilobytes of communication against 24 rounds and 4.99 kilobytes with CVA scheme.

Section 6.3 presents necessary definitions and tools. Our proposed scheme is presented in Section 3.3, its security argued in Section 3.3.1. Performance evaluation is given in Section 3.3.2 contains concluding remarks and open questions.

We define by $wt(x)$ the Hamming weight of $x \in \mathbb{F}_q^n$. The set of all permutations of n elements is denoted by $\mathcal{S}_n$. For $x, y \in \mathbb{F}_q^n$, we denote by $x + y$ an element-wise addition of x and y over $\mathbb{F}_q$.

An (n, k, d) linear code is a k-dimensional subspace of an n-dimensional vector space over a finite field $\mathbb{F}_q$, and d is the *minimal distance* of the code [Rot06]. We define by **q-ary** (n,k) the elements of $\mathbb{F}_q^{k\times n}$ with rank k.

3.2 Definitions and Notions

q-ary Linear Relations. We take the basic idea from [CVA10], and we change it as follow:

Definition 3.1 (q-ary Linear Relations)**.** Let $\Sigma \in \mathcal{S}_n$ and $\gamma = (\gamma_1, ..., \gamma_n) \in \mathbb{F}_q^n$ such that $\forall i, \gamma_i \neq 0$. We define the transformation $\Pi_{\gamma,\Sigma}$ as follows:

$$\begin{aligned}\Pi_{\gamma,\Sigma} : \mathbb{F}_q^n &\to \mathbb{F}_q^n \\ v &\longmapsto (\gamma_{\Sigma(1)} v_{\Sigma(1)}, ..., \gamma_{\Sigma(n)} v_{\Sigma(n)}).\end{aligned}$$

It was noted in [CVA10] that $\forall v \in \mathbb{F}_q^n$ and $\alpha \in \mathbb{F}_q$, we have $\Pi_{\gamma,\Sigma}(\alpha v) = \alpha \Pi_{\gamma,\Sigma}(v)$, and that this transformation preserves the weight, i.e. $\forall v \in \mathbb{F}_q^n$, $wt(\Pi_{\gamma,\Sigma}(v)) = wt(v)$. In addition, we observe that $\forall v, w \in \mathbb{F}_q^n$: $\Pi_{\gamma,\Sigma}(v + w) = \Pi_{\gamma,\Sigma}(v) + \Pi_{\gamma,\Sigma}(w)$, which implies that $\Pi_{\gamma,\Sigma}$ is a *linear* transformation.

We set $V, W \in \mathbb{F}_q^n$, $V = (v_1, , ..., v_n)$, $W = (w_1, , ..., w_n)$, $V + W = (v_1 + w_1, , ..., v_n + w_n)$
$(V + W)_{\Sigma(i)} = (V_{\Sigma(i)} + W_{\Sigma(i)})$, $\forall i : 1 \leqslant i \leqslant n$ $\Pi_{\gamma,\Sigma}(v) + \Pi_{\gamma,\Sigma}(w) = \Pi_{\gamma,\Sigma}(v + w)$
RHS=$(\gamma_{\Sigma(1)}(v + w)_{\Sigma(1)}, ..., \gamma_{\Sigma(n)}(v + w)_{\Sigma(n)})$;
LHS=$[\gamma_{\Sigma(1)} v_{\Sigma(1)}, ..., \gamma_{\Sigma(n)} v_{\Sigma(n)}] + [\gamma_{\Sigma(1)} w_{\Sigma(1)}, ..., \gamma_{\Sigma(n)} w_{\Sigma(n)}]$
$= \gamma_{\Sigma(1)}(v_{\Sigma(1)} + w_{\Sigma(1)}), ..., \gamma_{\Sigma(n)}(v_{\Sigma(n)} + w_{\Sigma(n)})$

$\because V, W \in \mathbb{F}_q^n, \therefore LHS = RHS$

After the proof, we can say this transformation is also linear in our scheme, and this permutation is random and no information can been detected by structure attack.

q-ary Commitment Schemes.

In the case when the q-ary assumption of commitment seems to be too restrictive , the size of q-ary commitment scheme with large computation, i.e. Given by a q-ary set over the q-ary space, may be applied Zero-knowledge proofs use q-ary commitment schemes as a building block. For this section, we borrow the presentation of [MT12]. A q-ary commitment scheme consists of two phases: the first one is *committing*, where a sender P provides a receiver V with an evidence about input b. The cheating receiver $\widetilde{V}$ cannot learn b before the second phase, called *opening*, when P reveals b to V. The cheating sender $\widetilde{P}$ cannot successfully open $b' \neq b$.

Let us denote by $\langle P, V\rangle_{A,st}$ the *view* of the party $A \in \{P, V\}$ at the stage st, which is a concatenation of all the messages sent and received by A, along with its local randomness.We denote by $Com(x_1, x_2, \ldots)$ a commitment to values $(x_1, x_2, \ldots)$.

Definition 3.2. A protocol is said to *securely implement string commitment*, if at the end of its execution by PPT Turing machines P (with input $b \in \mathbb{F}_2^l$, $l \in \mathbb{N}$) and V, the following properties hold:

(Correctness) $\Pr[\langle P(b), V\rangle_{V,Open} = \text{“ACCEPT”}]$ with overwhelming probability.

(Hiding) For any PPT $\widetilde{V}$, any $l \in \mathbb{N}$, any $b \in \mathbb{F}_2^l$ and $b' \in \mathbb{F}_2^l$ such that $b' \neq b$, after the committing stage, but before the opening stage, the distributions

$$\langle P(b), \widetilde{V}\rangle_{\widetilde{V},Commit} \quad \text{and} \quad \langle P(b'), \widetilde{V}\rangle_{\widetilde{V},Commit}$$

are indistinguishable. Depending of the type of indistinguishability, hiding can be *statistical* or *computational*.

(Binding) For any $\widetilde{P}$, any $l \in \mathbb{N}$, and $b' \in \mathbb{F}_2^l$ there exists $b \in \mathbb{F}_2^l$ which can be computed by P after the committing stage, such that the probability

$$\Pr[\langle \widetilde{P}(b'), V\rangle_{V,Open} = \text{“ACCEPT”}]$$

is negligible. If $\widetilde{P}$ restricted to run in PPT, then binding is called *computational*, if $\widetilde{P}$'s computing power is not restricted, then the binding is *statistical*.

Note that committing to binary vectors does not pose a problem, since mathematical objects, which we are working with, will eventually be represented as binary vectors for the actual implementation.

In the random oracle model (ROM) [BR93], a string commitment which is both computationally hiding and binding can be implemented using (idealized) cryptographic hash function. In the standard model, a computationally hiding and statistically binding code-based commitment commitment schemes are known [DvdGMQN08, JKPT12].

3.3 Variant of Stern's Scheme Based on q-ary Codes

We present our proposed zero-knowledge identification protocol, which is a generalization of (binary) Stern identification scheme [Ste96] to q-ary case. In the context of lattice-based cryptography, a similar scheme was presented in [KTX08, Xag10]. The main difference is that an ordinary permutation cannot be used now, since a permutation of the q-ary witness does indeed release some information on it, hereby violating the zero-knowledge requirement. We avoid this problem by employing the generalized permutation $\Pi_{\gamma,\Sigma}$ introduced in [CVA10] exactly for this purpose. Indeed, it is easy to check that for any input in $\{x \in \mathbb{F}_q^n | wt(x) = \omega\}$, this transformation outputs a vector with uniform distribution in $\{x \in \mathbb{F}_q^n | wt(x) = \omega\}$ given that (uniformly chosen) γ and Σ are unknown. Since transformation $\Pi_{\gamma,\Sigma}$ happens to be linear as noted in above, it can be used directly in the construction of [Ste96].

Key Generation.

Input: Given a security parameter and q, choose n, k, and ω, then compute:
$H \xleftarrow{\$} \mathbb{F}_q^{(n-k)\times n}$, $e \xleftarrow{\$} \mathbb{F}_q^n$ such that $wt(e) = \omega$, and $y \leftarrow He^T$.

Output: the private key sk, and the public key (identification) pk: $(\mathrm{sk}, \mathrm{pk}) = (e, (y, H, \omega))$.

Technically, our protocol is an interactive zero-knowledge proof of knowledge [Gol01] for the predcate:

$$\mathcal{P}(sk, pk) = \text{“With respect to } pk,\ e \text{ is such that}$$

$$y = He^T \text{ and } wt(e) = \omega\text{”},$$

where pk is a common data, and sk is a witness.

Our protocol is presented in Table 3.1. It has soundness error $2/3$, which can be reduced to $(2/3)^\delta$ by iterating this protocol independently for δ rounds.

3.3.1 Security Proof

Our proof uses the approach presented in [Ste96, Vér97]. We also follow [MT12] by presenting our proof in the standard model.

Figure 3.1: q-ary Stern Identification Protocol.

Prover P		Verifier V
$(sk, pk) = (e, (y, H, \omega))$ $u \xleftarrow{\$} \mathbb{F}_q^n$ $\Sigma \xleftarrow{\$} \mathcal{S}_n, \gamma \xleftarrow{\$} (\mathbb{F}_q^*)^n$		$pk = (y, H, \omega)$
$c_1 = Com(\gamma, \Sigma, Hu^T)$ $c_2 = Com(\Pi_{\gamma,\Sigma}(u))$ $c_3 = Com(\Pi_{\gamma,\Sigma}(u + e))$		
	$c_1, c_2, c_3 \longrightarrow$	
	$\longleftarrow b$	$b \xleftarrow{\$} \{0, 1, 2\}$
If $b = 0$	$u, \gamma, \Sigma \longrightarrow$	P opens c_1 and c_2, V checks correctness of c_1, c_2, using H which is public.
If $b = 1$	$u + e, \gamma, \Sigma \longrightarrow$	P opens c_1 and c_3, V checks correctness using $Hu^T = H(u + e)^T - y$.
If $b = 2$	$\Pi_{\gamma,\Sigma}(u), \Pi_{\gamma,\Sigma}(e) \longrightarrow$	P opens c_2 and c_3, and checks correctness using $\Pi_{\gamma,\Sigma}(u)+\Pi_{\gamma,\Sigma}(e) = \Pi_{\gamma,\Sigma}(u+e)$, and checks that $wt(\Pi_{\gamma,\Sigma}(e)) = \omega$.

Let us denote the cheating prover by $\widetilde{P}$ and a cheating verifier by $\widetilde{V}$.

Proposition 3.3. *The protocol in Table 3.1 is an interactive zero-knowldge proof of the predicate $\mathcal{P}(sk, pk)$, in the standard model.*

Proof. **Completeness:** If P knows the witness, it is easy to show that she can answer each challenge correctly, in particular one will need to use the fact that the transformation $\Pi_{\gamma,\Sigma}$ is linear and that it preserves the weight.

Soundness:

Lemma 3.4. *If V accepts $\widetilde{P}$'s proof with probability at least $\left(\frac{2}{3}\right)^r + \epsilon$, then there exists a PPT algorithm which with overwhelming probability computes the witness e.*

Proof. Let T be an execution tree of the protocol between $\widetilde{P}$ and V that corresponds to all possible challenges by V. Verifier V can send 3 possible challenges in each round. Suppose that the binding property of the underlying commitment scheme holds. Then, we present a PPT algorithm (called *witness extractor*) that computes the witness e from a vertex with 3 descendants.

Suppose there exists a vertex with 3 descendants. This implies that all the three challenges were correctly answered. Suppose now that the following responses were provied by $\widetilde{P}$:

- $b = 0 : (u_0, \gamma_0, \Sigma_0)$,
- $b = 1 : (w_1, \gamma_1, \Sigma_1)$ (w_1 corresponds to $u + e$),
- $b = 2 : (z_2, t_2)$ (correspond to $\Pi_{\gamma,\Sigma}(u)$ and $\Pi_{\gamma,\Sigma}(e)$, respectively).

Now, according to the checks perfomed by V, we have : $(\gamma_0, \Sigma_0, Hu_0^T) = Open(c_1) = (\gamma_1, \Sigma_1, Hw_1^T - y)$. Remember that the binding property is assumed to hold, then we have $\gamma_0 = \gamma_1$, $\Sigma_0 = \Sigma_1$, and $Hu_0^T = Hw_1^T - y$. Using correctness of c_2 and c_3, we also have $z_2 = \Pi_{\gamma_0,\Sigma_0}(u_0)$, $z_2 + t_2 = \Pi_{\gamma_1,\Sigma_1}(w_1)$, and $wt(t_2) = \omega$. This implies $t_2 = (t_2+z_2)-z_2 = \Pi_{\gamma_0,\Sigma_0}(w_1-u_0)$ where $wt(w_1-u_0) = \omega$. Therefore, the expression $H(w_1 - u_0)^T = Hw_1^T - Hu_0^T = y$ shows that $w_1 - y_0$ is a valid witness.

The rest of the proof of this lemma is exactly as in [Vér97]. The omitted part shows that the probability for T to have a vertex with 3 descendants is at least ϵ. □

Zero-knowledge: This property states that the cheating polynomial-time verifier $\widetilde{V}$ learns no information on the witness irrespective of her cheating strategy.

Lemma 3.5. *Protocol in Table 3.1 is zero-knowledge if the used commitment scheme is hiding.*

Proof. The flavor of zero-knowledge – computational or statistical – depends on the corresponding flavor of the hiding property of the underlying commitment scheme.

Next, we present a PPT algorithm called the *simulator* which works in expected polynomial time, and which constructs a protocol transcript whose distribution is indistinguishable from the transcript of the protocol execution between honest P and V.

Since the zero-knowledge property must hold irrespective of $\widetilde{V}$'s strategy, we denote this strategy by $St(c_1, c_2, c_3)$, and assume that the challenges are chosen according to it. The simulator works as follows:

1. Pick a challenge $b \xleftarrow{\$} \{0, 1, 2\}$.
   - If $b = 0$, choose $u \xleftarrow{\$} \mathbb{F}_q^n$, $\gamma \xleftarrow{\$} (\mathbb{F}_q^*)^n$, $\Sigma \xleftarrow{\$} \mathcal{S}_n$, compute $c_1 = Com(\gamma, \Sigma, Hu^T)$, $c_2 = Com(\Pi_{\gamma,\Sigma}(u))$, $c_3 = Com(0)$, and $Response = (u, \gamma, \Sigma)$.
   It is easy to check that the distributions of c_1, c_2, c_3 and $Response$ are identical to the corresponding distributions in the actual protocol transcript.
   - If $b = 1$, choose $u \xleftarrow{\$} \mathbb{F}_q^n$, $\gamma \xleftarrow{\$} (\mathbb{F}_q^*)^n$, $\Sigma \xleftarrow{\$} \mathcal{S}_n$, and $w = u + z$, where $z \in \mathbb{F}_q^n$ is such that $Hz^T = y$, $z \neq e$, $wt(z) \neq \omega$. Then, compute $c_1 = Com(\gamma, \Sigma, Hu^T)$, $c_2 = Com(0)$, $C_3 = Com(\Pi_{\gamma,\Sigma}(w))$, and $Response = (w, \gamma, \Sigma)$. Again, it is easy to check that the values in $Response$ are consistent with the checks, and also that distributions of the

commitments and *Response* are identical to those in the actual protocol transcript. In particular, a uniform u serves as a one-time pad for z.
• If $b = 2$, choose $u \xleftarrow{\$} \mathbb{F}_q^n$, $\gamma \xleftarrow{\$} (\mathbb{F}_q^*)^n$, $\Sigma \xleftarrow{\$} \mathcal{S}_n$, and $z \xleftarrow{\$} \{x \in \mathbb{F}_q^n | wt(x) = \omega\}$. Then, compute $c_1 = Com(0)$, $c_2 = Com(\Pi_{\gamma,\Sigma}(u))$, $c_3 = Com(\Pi_{\gamma,\Sigma}(u + z))$, and $Response = (\Pi_{\gamma,\Sigma}(u), \Pi_{\gamma,\Sigma}(z))$. The values in *Response* are consistent with the checks, and it is easy to see that distributions of the commitments and *Response* are identical to those in the actual transcript.

2. The simulator computes $b' = St(c_1, c_2, c_3)$.

3. If $b = b'$, then the simulator outputs a transcript which includes H, b and *Response*, otherwise go to Step 1.

Now, in 3δ iterations on the average, the above algorithm constructs the protocol transcript which is indistinguishable from the transcript of the protocol execution between the honest parties. □

Now, Lemmas 1 and 2 conclude the proof of Proposition 1. □

3.3.2 Performance Evaluation

For simplicity of our calculations, we assume that commitments are implemented using random oracles, in the same way as it is done in [Ste96, CVA10]. Then the commitment function will be computed as some cryptographic hash function h with output size $l_h = 160$ bits. When checking the prover's responses, the verifier will simply check that the output of the hash functions on the values contained in the response are the same as those sent as commitments. For instance, when $b = 0$, the verifier will check if $h(\gamma, \Sigma, Hu^T) = c_1$ and $h(\Pi_{\gamma,\Sigma}(u)) = c_2$.

We will assume that the values u and (γ, Σ) are computed using pseudorandom generators (PRG) with seed of length $l_s = 128$ bits. Hereby, only the seeds can be sent to save on communication, in the same way, as it is done in [Ste96, CVA10].

The syndrome decoding problem is hardest to solve when $k \approx n/2$ [CCC98] and ω is chosen slightly below the Gilbert-Varshamov bound (see Section 6.3). We take $k = n/2$, and hereby we only need to choose n (as an even integer).

In order to compute the recommended parameters for a given security level, we will use the currently best algorithm for solving q-ary Syndrome Decoding problem that is the Information Set Decoding algorithm by Peters [Pet10]. Since the equations for its parameters do not have a short analytic representation, we use the C program introduced by Peters, which is available at `https://bitbucket.org/cbcrypto/isdfq/src`.

We let N be the number of bits needed to represent an element of $\mathbb{F}_q$, and we set $N = \lceil \log_2 q \rceil$. We take $l_\Sigma = n\lceil \log_2 n \rceil$ to be the size of representation of a permutation Σ, and $l_\gamma = nL$ to be the size of representation of a vector γ, both values are in bits. Let δ be the number of rounds. We assume that the common matrix H is in the systematic form.

Our scheme has the following parameters.

Size of the common matrix: k^2N.
Size of the public identification: kN.
Size of the secret key: nN.

In fact, the above parameters are the same as in CVA scheme.

Total communication cost, in bits:
$\delta(3l_h + 2 + (3l_s + 3nN)/3)$.
Prover's computation complexity over $\mathbb{F}_q$, per round:
$(k^2 + n + \omega)$ multiplications and $(k^2 + 2\omega)$ additions.

Concerning calculation of the communication cost, we note that the above values are the average assuming that the challenges are made uniformly at random. In addition, we point out that although opening of the commitment do not requires sending of its contents, because we only need to get the hash values to check the commitments function.

in most of the cases we can avoid it, since the same value is sent as a part of response and there is no need to send the value twice. For instance, if $b = 0$, opening of c_1 does not require sending of its contents because γ and Σ have already been sent as a part of the response, and Hu^T can be computed by the verifier, as u is another part of the response, using H which is public. On the other hand, if $b = 1$, then Hu^T cannot be efficiently computed from the response, then hence it must be sent by the prover. In fact, it is easy to check that the later value is the only overhead that is incurred by opening the commitments, apart from the need to sent the randomness used for commitments.

Cayrel et al [CVA10] offer the following estimate for the communication cost of their protocol:

$$\delta(2l_h + N + nN + 1 + (l_\gamma + l_\Sigma + nN)/2),$$

where $l_\gamma = l_\Sigma = l_s$ is the length of a seeds of some PRG used to obtain γ and Σ. For a fair comparison with our scheme, we assume that (γ, Σ) are generated from a single seed in their scheme.

By applying this estimate to the recommended parameters suggested in [CVA10], we can see that the actual communication cost of their proposal is about twice as large, as compared to the claimed estimates. For example, for 87-bit equivalent security level with $q = 256$, $n = 128$,

Table 3.1: Performance Comparison of Our Proposal and CVA Scheme for $q = 3$, 80bits

$q = 3, n = 396, k = 198, \omega = 62$	CVA [CVA10]	Our Proposal
Number of Rounds	39	28
Matrix size (kilobytes)	9.57	9.57
Public key (bits)	396	396
Secret key (bits)	792	792
Communication (kilobytes)	7.50	4.79
Prover's Computation over $\mathbb{F}_3$	$2^{20.58}$ multiplications, $2^{20.54}$ additions	$2^{20.08}$ multiplications, $2^{20.07}$ additions

Table 3.2: Performance Comparison of Our Proposal and CVA Scheme for $q = 4$, 80bits

$q = 4, n = 328, k = 164, w = 61$	CVA [CVA10]	Our Proposal
Number of Rounds	28	28
Matrix size (kilobytes)	6.57	6.57
Public key (bits)	328	328
Secret key (bits)	656	656
Communication (kilobytes)	4.69	4.33
Prover's Computation over $\mathbb{F}_4$	$2^{19.56}$ multiplications, $2^{19.53}$ additions	$2^{19.54}$ multiplications, $2^{19.53}$ additions

Table 3.3: Performance Comparison of Our Proposal and CVA Scheme for $q = 5$, 80bits

$q = 5, n = 292, k = 146, w = 60$	CVA [CVA10]	Our Proposal
Number of Rounds	24	28
Matrix size (kilobytes)	7.81	7.81
Public key (bits)	438	438
Secret key (bits)	876	876
Communication (kilobytes)	4.99	5.08
Prover's Computation over $\mathbb{F}_5$	$2^{19.01}$ multiplications, $2^{18.97}$ additions	$2^{19.21}$ multiplications, $2^{19.20}$ additions

$k = 64$, $\omega = 49$, and soundness error 2^{-16} (provided by 16 rounds), we obtain the communication cost of 7.27 kilobytes rather than 3.89 kilobytes as claimed. For 128-bit equivalent security level with $q = 256$, $n = 208$, $k = 104$, $\omega = 78$, and the same soundness error, we obtain 11.46 kilobytes against 5.77 kilobytes as claimed.

Next, we provide recommended parameters for the equivalent security level of 80 bits and 128

Table 3.4: Performance Comparison of Our Proposal and CVA Scheme for $q = 3$, 128bits

$q = 3, n = 654, k = 327, \omega = 103$	CVA [CVA10]	Our Proposal
Number of Rounds	39	28
Matrix size (kilobytes)	26.11	26.11
Public key (bits)	654	654
Secret key (bits)	1308	1308
Communication (kilobytes)	11.18	6.56
Prover's Computation over $\mathbb{F}_3$	$2^{22.01}$ multiplications, $2^{21.99}$ additions	$2^{21.52}$ multiplications, $2^{21.52}$ additions

Table 3.5: Performance Comparison of Our Proposal and CVA Scheme for $q = 4$, 128bits

$q = 4, n = 540, k = 270, w = 101$	CVA [CVA10]	Our Proposal
Number of Rounds	28	28
Matrix size (kilobytes)	17.80	17.80
Public key (bits)	540	540
Secret key (bits)	1080	1080
Communication (kilobytes)	6.86	5.78
Prover's Computation over $\mathbb{F}_4$	$2^{20.99}$ multiplications, $2^{20.96}$ additions	$2^{20.97}$ multiplications, $2^{20.97}$ additions

Table 3.6: Performance Comparison of Our Proposal and CVA Scheme for $q = 5$, 128bits

$q = 5, n = 480, k = 240, w = 99$	CVA [CVA10]	Our Proposal
Number of Rounds	24	28
Matrix size (kilobytes)	21.09	21.09
Public key (bits)	720	720
Secret key (bits)	1440	1440
Communication (kilobytes)	7.46	7.01
Prover's Computation over $\mathbb{F}_5$	$2^{20.43}$ multiplications, $2^{20.40}$ additions	$2^{20.64}$ multiplications, $2^{20.63}$ additions

bits soundness error 2^{-16} – to satisfy the later requirement our scheme needs 28 rounds (independently of q). Our recommended parameters for $q = 3$, along with the corresponding parameters for Cayrel et al scheme for comparison, are presented in Table 3.4. In this case, our scheme requires 4.79 kilobytes of communication cost that is by 36% smaller than the communication cost of CVA scheme being 7.50 kilobytes; besides their scheme requires 39 rounds. When $q = 4$ (see Table 3.5), our scheme requires 4.33 kilobytes of communication that is by 8% smaller

than 4.69 kilobytes of communication required for CVA scheme; the number of rounds is 28 in both protocols. For $q = 5$, the performance evaluation results are given in Table 3.6. Now, our scheme requires 5.08 kilobytes of communication, as compared to 24 rounds and 4.99 kilobytes with CVA, which is by 2% smaller than with ours. For $q > 5$, the CVA scheme is superior to our scheme in terms of communication cost, since the soundness error of their scheme approaches $1/2$, when q is growing. We note that the complexity of Information Set Decoding is growing when q increases, hence larger q require smaller n for the same security level.

3.4 Conclusion

In this section, we presented a zero-knowledge identification scheme based on q-ary syndrome decoding with soundness error $2/3$, which is a generalization of (binary) Stern scheme to q-ary case. Our scheme is superior to the CVA scheme [CVA10] in terms of communication cost, but only for $q = \{3, 4\}$.

An open question is to reduce the soundness error to exactly $1/2$ in the case of small q, most importantly for $q = 2$, since a scheme working over the binary field is expected to have a fast implementation. Reducing the size of the public matrix is another natural open question.

Chapter 4

Proof of Plaintext Knowledge for Code-Based Encryption

4.1 Introduction

Proof of Plaintext Knowledge cryptography encompasses schemes which remain secure even without decryption. In this chapter we will use coded-based assumption to construct PPK scheme. Two important candidates for the code-based PPK scheme are the code-based PKE schemes by McEliece [McE78] and Niederreiter [Nie86]. Their security is based on hardness of decoding, which is a well-studied cryptographic assumption [BMVT78, Sen02, EOS07, Ber10, FGUO+13, BJMM12].

The McEliece encryption is the first code-based PKE. It uses the error-correcting codes by Goppa [Gop70, Mat80]. Breaking of both of these PKE's is believed to be infeasible for properly chosen parameters [EOS07, BLP11, FGUO+13], even for adversaries equipped with quantum computers [Ber10, DMR11a, DMR11b]. The later fact makes these PKE a prospective candidate for the post-quantum world. One more advantage pointed out by Bernstein et al. [BLP11, App. A] is a good asymptotic performance of the McEliece PKE.

One of the challenges in the code-based cryptography is to enrich the variety of code-based cryptographic protocols. Existing results include, for instance, identification schemes and digital signatures – see e.g. the surveys [EOS07, OS09, CM10], and also [Pie12] for related results.

In this chapter, we focus on the *proof of plaintext knowledge (PPK)* for the code-based PKE, and their applications. Suppose that a prover P encrypted the plaintext m into ciphertext c on the public key pk. Now, a PPK allows P to convince a verifier V, who does not have the secret key, that P knows m. If such the proof is zero-knowledge, it will not reveal any additional information on m, apart from the above statement itself. We will apply our PPK to construct a

code-based verifiable public-key encryption. For such an encryption with respect to some binary relation R on the plaintexts, there exists a zero-knowledge proof on the public inputs pk, c, and δ that allows a prover P to convince a verifier V that c is a ciphertext of m under pk such that $(m, \delta) \in R$. In this paper, we will focus on the case when R is an an equality relation, and $\delta = m$. In other words, the verifier will be convinced that a given ciphertext c contains a given plaintext m.

4.1.1 Zero-Knowledge Proofs for PPK Schemes

Note that assuming that one-way functions exist, one could achieve the results presented in this work using general ZK proofs for NP-statements [GMW91], however such constructions would be prohibitively inefficient.

Jain et al. [JKPT12] constructed efficient ZK proofs for NP-statements assuming hardness of (x)LPN problem. Their proposal is based on a proof of valid opening for their commitment scheme. Their proof of valid opening also implies a ZK identification scheme based on xLPN.

Note that xLPN problem is equivalent to that of general decoding for the parameters in question. More specifically, the only difference is that in xLPN, the generator matrix of the code is chosen uniformly at random rather than being full-rank. However, it is possible to choose the appropriate code parameters such that the probability for the uniform matrix to be full-rank is overwhelming. Finally, note that we use interactive ZK proofs. According to an observation made in [GK05]: "... known constructions for non-interactive zero-knowledge proofs (NIZK) [DDN03] for NP languages (which are a central tool in constructing CCA2 secure non-interactive public-key encryption given semantically secure public key encryption algorithms) require trapdoor permutations." At the same time, the hardness assumptions related to coding give rise to only trapdoor function candidates [McE78, Nie86].

4.1.2 Main Results of This Chapter and Discussion

Our contribution is two-fold: We present the first proof of plaintext knowledge for the above mentioned code-based PKE's, and then we apply it to obtain the first code-based verifiable encryption in next section.

Our main observation is that the ZK identification schemes by Stern [Ste96] and by Jain et al. [JKPT12] can be directly used to construct the PPK for the Niederreiter and the McEliece PKE's, respectively. We emphasize that this fact does not immediately follow from the previous works due to the following observations.

First of all, we note that the public key of each PKE will have to be used as a common matrix of the corresponding ZK identification scheme, which is supposed to be uniformly random. Our security proof shows that, in fact, the requirement on the randomness of the common matrix is not necessary. The only essential requirement is that the general decoding for the matrix in question is hard.

Secondly, we emphasize that our proof technique for the McEliece PPK is different from that of Jain et al., while being along the lines of [Ste96, Vér97]. Hereby, it follows from our result that the Jain et al. identification scheme (and the proof of valid opening of their commitment) is secure against a malicious verifier. In detail, Jain et al. observed that the proof of valid opening of their commitment is in fact a special variant of Σ-protocols where soundness error is larger than $1/2$. They claimed that the major results for Σ-protocols hold for their variant as well. We however choose to disregard this observation, and to construct the security argument for our PPK as for the standard ZK proof of knowledge. The reason is that typically for Σ-protocols, a special honest-verifier ZK property is proved first, and then the security against malicious verifier is obtained via the transformation by Damgård et al. [DGOW95]. Now, our observation allows us to circumvent the use of this transformation, hereby making our protocol more efficient as compared to that of Jain et al.

Third, the setting of PPK is different from that of identification, since the prover does not necessary provide a valid ciphertext. In the setting of identification, it would mean that the key generation is not necessarily performed correctly. Since the identification schemes by Stern and by Jain et al. are based on the ZK proof systems, we note that in the latter setting, this problem corresponds to the case when a valid witness does not exist. Therefore, for our protocols, we prove the soundness property in the strong sense (it is called a strong validity in the survey by Bellare and Goldreich [BG93]), meaning that one must make sure that the prover is always rejected by the honest verifier, if the witness does not exist. In turn, our proofs imply a better security for the above mentioned identification schemes.

Next, we provide a performance evaluation for the proposed protocols and suggest secure parameter sets. In particular, for 80-bit equivalent security, the PPK for the Niederreiter PKE requires about 9.1 kilobytes of communication, while the PPK for the McEliece PKE requires about 16 kilobytes of communication, both with 28 rounds, which can be considered practical.

Next, we apply the PPK for the McEliece PKE to construct the verifiable IND-CPA McEliece encryption for an equality relation. Note that the original McEliece encryption cannot be used here, since given pk and $c = Enc_{pk}(m)$, it is trivial to check whether or not c is a ciphertext of m. Therefore, we use an IND-CPA secure McEliece encryption [NIKM08] instead. It is interesting to note that in the lattice-based constructions [GK05, XKT07], one first constructs a verifiable encryption for an equality relation, and then uses it as a building block for the PPK, while in our case it works the other way around. In our construction, we must impose that the rows of the

McEliece public key matrix are linearly independent, i.e., that this matrix is full-rank. This is a natural assumption, as the McEliece public key is typically chosen this way.

In our constructions, we assume that both the prover and the verifier are assured that the public key pk is valid. This assumption will require a trusted third party who generates public keys – this can be, for instance, an entity in the public key infrastructure.

Throughout our work, we assume the presence of the public key infrastructure (or any other key authentication mechanism), which ensures validity of receiver's public key to be used in PPK.

Currently, the most efficient code-based commitment scheme is the construction of Jain et al. [JKPT12].

The type of zero-knowledge which we obtain, being statistical or computational, depends solely on the employed commitment scheme. In the case of the schemes [DvdGMQN08, JKPT12], which are statistically binding and computationally hiding, the ZK property holds in the computational sense. In principle, we can achieve efficient statistical ZK proofs, if we assume existence of collision-resistant hashing, [1] due to statistically hiding (and computationally binding) commitments by Halevi and Micali [HM96].

As a side remark, we mention a purely theoretical implication of our result for interactive code-based IND-CPA PKE in the standard model [GHY86, Gol01, Kat03], when the IND-CPA secure McEliece encryption by Nojima et al. [NIKM08] is combined with the presented PPK. One might also consider the applications of PPK suggested by Katz [Kat03] such as deniable authentication or password-based authenticated key exchange. However, those are more relevant to PKE's which do *not* have efficient IND-CCA2 secure instantiations in the standard model. Recently, such the instantiation was presented by Mathew et al. [MVVR12], therefore it is more suitable for the above applications, as compared to the interactive solution.

4.2 PPK for Niederreiter PKE

We construct the proof of plaintext knowledge for the Niederreiter PKE using the Stern ZK identification scheme [Ste96]. We take the public key of the PKE, i.e. a permuted and scrambled parity-check matrix of an irreducible binary Goppa code correcting up to t errors as the common data. The plaintext is used as witness and the ciphertext is used as public identification.

We observe that the security proof of the Stern scheme [Ste96] does not use the fact that the common code is random. It is only important that the common code has a particular minimum distance, which is provided by construction, and that the syndrome decoding problem for this

[1] Since we are not aware of efficient constructions of collision-resistant hashing from the standard code-based assumptions.

code is hard. The latter is ensured by the hardness of the Niederreiter PKE (see Section 2.4.4.2). Since the Niederreiter PKE is deterministic, we consider the string R representing the randomness in Definition 2.27 as empty.

In our proofs throughout this chapter, similarly to the works [Ste96, Vér97, Kat03, CVA10, JKPT12], we assume commitments to be ideal. This is done in order to simplify our arguments, and to abstract from a specific security of the underlying commitment scheme, whether computational or statistical, because this scheme can be used as a black box in our constructions.
Witness: $m \in \mathbb{F}_2^n$, $w_H(m) = t$, where the parameters n and t are described in Section 2.4.4.2.

Common data: (H^{pub}, t) such that $H^{pub} \in \mathbb{F}_2^{(n-k)\times n}$ – the public key of the Niederreiter PKE, and $c = H^{pub}m^T$ – the ciphertext of the Niederreiter PKE.

Protocol 1 (Niederreiter PPK).

1. P computes $y \xleftarrow{\$} \mathbb{F}_2^n$, $\pi \xleftarrow{\$} \mathcal{S}_n$, and sends the following three commitments to V:
   – $C_1 = Com(\pi, H^{pub}y^T)$,
   – $C_2 = Com(y\pi)$,
   – $C_3 = Com((y + m)\pi)$.
2. V sends a challenge $b \xleftarrow{\$} \{0, 1, 2\}$.
3. P replies as follows, while V performs the following checks and rejects, if any check fails:
   – If $b = 0$,
     – P sends y, π and opens C_1 and C_2.
     – V checks validity of the opened values.
   – If $b = 1$,
     – P sends $y + m, \pi$ and opens C_1 and C_3.
     – V checks validity by computing $H^{pub}y^T = H^{pub}(y + m)^T + c$, and then verifying that the opening of C_1 is $(\pi, H^{pub}(y + m)^T + c)$, and the opening of C_3 is $(y + m)\pi$.
   – If $b = 2$,
     – P sends $y\pi, m\pi$ and opens C_2 and C_3.
     – V checks validity of the opened values by verifying that C_2 opens to $y\pi$, C_3 opens to $y\pi + m\pi$, and that $w_H(m\pi) = t$.

Denote a protocol consisting of r independent sequential iterations of Protocol 1 by $\mathsf{PPK}_N(H^{pub}, c; m)$, with some appropriately chosen r.

Theorem 4.1. *The above protocol, which is denoted as* $\mathsf{PPK}_N(H^{pub}, c; m)$*, is a proof of plaintext knowledge for the Niederreiter public key encryption in the standard model, according to Definition 2.27 assuming that the Niederreiter problem is hard, and the underlying commitment scheme satisfies Definition 9.*

Proof. We generally follow the proof of [Ste96], but for the proof of soundness we use the argument from [Vér97], since it is shorter. We emphasize that the gap in the proof of [Vér97] pointed out in [JKPT12] concerned only the proof of the zero-knowledge property.

Completeness. It is easy to check that P who knows the plaintext m can answer all of three challenges correctly. This implies that $\langle P(m), V\rangle(pk, c) = 1$.

Soundness. First, we note that similarly to [Kat03], we will prove the soundness in the strong sense – for the details, we refer the reader to the survey by Bellare and Goldreich [BG93], where such the property is called a strong validity. Technically, we will first prove that if there exists no witness – or in other words, if the prover presents an invalid ciphertext – then she has only a negligible probability to be accepted. At the same time, if the prover did manage to be accepted with a non-negligible probability, then we construct a so called witness extractor – an algorithm running in expected polynomial time which computes the witness.

Lemma 4.2. *Protocol 1 is sound according to Definition 2.27, if the underlying commitment scheme is binding, the Niederreiter problem is hard, and* $r(\kappa) = \omega(\log \kappa)$.

This lemma follows from the following two auxiliary lemmas. In their proofs, we will omit mentioning the fact that the parameters r and ϵ depend on the security parameter κ, for simplicity.

Lemma 4.3. *If the witness does not exist, then the probability for* $\widetilde{P}$ *to be accepted in the above protocol is at most* $\left(\frac{2}{3}\right)^r$*, after* r *rounds.*

Proof. We show that if $\widetilde{P}$'s replies to all the three challenges are accepted, then a (valid) witness can be computed from them. This will contradict the assumption, and imply that $\widetilde{P}$ is not able to answer all the three challenges at the same time, hence his probability to be accepted is at most $\frac{2}{3}$ in every round.

Consider the following challenge-response pairs:

- $b = 0 : (y_0, \pi_0)$,
- $b = 1 : (w_1, \pi_1)$ (w_1 corresponds to $y + m$),
- $b = 2 : (z_2, t_2)$ (correspond to $y\pi$ and $m\pi$, respectively).

Since, the information in the opened commitments is consistent by assumption, we have: $(\pi_0, H^{pub}y_0^T) = Open(C_1) = (\pi_1, H^{pub}w_1^T + c)$. Since binding holds, we conclude that $\pi_0 = \pi_1$ and $H^{pub}y_0^T = H^{pub}w_1^T + c$. Similarly, by consistency of the commitments C_2 and C_3, and by the binding property, we can show that $z_2 = y_0\pi_0$, $z_2 + t_2 = w_1\pi_1$, and $w_H(t_2) = t$. Therefore, we

have that $t_2 = z_2 + (t_2 + z_2) = (y_0 + w_1)\pi_0$ such that $w_H(y_0 + w_1) = t$. Now from $H^{pub}(y_0 + w_1)^T = H^{pub}y_0^T + H^{pub}w_1^T = c$, we conclude that $y_0 + w_1$ is a valid witness. □

Lemma 4.4. *If V accepts $\widetilde{P}$'s proof with probability at least $(\frac{2}{3})^r + \epsilon$, then there exists an expected PPT algorithm WE which, with overwhelming probability, computes a witness m.*

Proof. Let $\mathcal{T}(RA)$ be an execution tree of the protocol $(\widetilde{P}, V)$, where RA is the random tape of $\widetilde{P}$. This tree is constructed as follows. A vertex will represent the commitments made by $\widetilde{P}$, and the edges will be labeled by the challenges of V. An edge will be present only if $\widetilde{P}$ is able to correctly reply to the challenge. Remember that V can send 3 possible challenges at each stage. First, we will argue that as long as the binding property of the commitment holds, a witness m can be computed from a vertex with 3 descendants, that is from the correct answers to three challenges. Next, we will show that a PPT WE can find such a vertex in $\mathcal{T}(RA)$ with overwhelming probability.

Let v be a vertex with 3 descendants. This corresponds to a situation, where 3 commitments C_1, C_2 and C_3 have been made and where the three challenges were correctly answered. Then, the witness can be computed from these correct answers as described in Lemma 4.3.

Next, we can use the argument from [Vér97] (also used in [MT12]) to show that the probability for $\mathcal{T}(RA)$ to have a vertex with 3 descendants is at least ϵ. We give this argument here for the sake of completeness.

Let us consider the random tape RA of $\widetilde{P}$ as a set of μ elements, from which $\widetilde{P}$ randomly picks its values and let $Q = \{1, 2, 3\}$. These two sets are considered as probability spaces, both of them with uniform distribution.

A pair $(a, b) \in (RA \times Q)^r$ represents the commitments, challenges and responses communicated between $\widetilde{P}$ and V. This is indeed the case, since the random tape of the prover, along with the challenges, uniquely defines all the messages sent by her during the protocol. A pair (a, b) is called *valid*, if the execution of $(\widetilde{P}, V)$ is accepted.

Let V be the subset of valid pairs in $(RA \times Q)^r$. By the hypothesis of the lemma,

$$\frac{|V|}{|(RA \times Q)^r|} \geq \left(\frac{2}{3}\right)^r + \epsilon.$$

Let $\Omega_r \subset RA^r$ be such that:
– If $a \in \Omega_r$, then $2^r + 1 \leq |\{b : (a, b) \text{ are valid}\}| \leq 3^r$,
– If $a \in RA^r \setminus \Omega_r$, then $0 \leq |\{b : (a, b) \text{ are valid}\}| \leq 2^r$.

Then, we write $V = \{\text{valid } (a,b), a \in \Omega_r\} \cup \text{valid } (a,b), a \in RA^r \setminus \Omega_r\}$, therefore $|V| \leq |\Omega_r| \cdot 3^r + (\mu^r - |\Omega_r|) \cdot 2^r$. Taking into account that $|RA^r| = \mu^r$ and $|Q^r| = 3^r$, we have

$$\frac{|V|}{|(RA \times Q)^r|} \leq \left(\frac{|\Omega_r|}{|RA^r|} + 2^r\left(3^{-r} - \frac{|\Omega_r|}{|(RA \times Q)^r|}\right)\right) \leq \frac{|\Omega_r|}{|RA^r|} + \left(\frac{2}{3}\right)^r.$$

Now, it follows that $|\Omega_r|/|RA^r| \geq \epsilon$, which shows that the probability that $\widetilde{P}$ replies correctly to at least $2^r + 1$ challenges, by choosing random values from RA, is at least ϵ. Moreover, in this case, $\mathcal{T}(RA)$ has at least $2^r + 1$ leaves. Indeed, by construction of $\mathcal{T}(RA)$, a correctly answered challenge corresponds to an edge, and therefore, the number of leaves is lower bounded by the number of correctly answered challenges. This implies that $\mathcal{T}(RA)$ has at least one vertex with 3 descendants. Now, the machine WE will simply rewind the above $\widetilde{P}$ polynomially many times, hereby finding an execution tree containing a vertex with 3 descendants with overwhelming probability, as claimed. Specifically, we can directly use the analysis by Stern [Ste96, Lemma 1] to verify that the number of necessary rewindings is $\frac{10}{\epsilon^3}$.

Finally, we note that the machine WE, constructed in the proof, finds a valid witness, hereby contradicting the hardness of the Niederreiter problem, unless the binding property of the commitment is violated. Therefore, for a cheating prover $\widetilde{P}$, we must have $\Pr[\langle \widetilde{P}, V\rangle(pk, c) = 1] \leq (2/3)^r + \epsilon$, which is negligible in κ. □

We emphasize that the above proof does not require an indistinguishability of the Niederreiter public key from the random code.

Zero-knowledge. Let us denote by $\mathcal{R}$ the communication tape for P and V, that is a concatenation of all bits they exchange during the protocol. We consider the probability distributions on $\mathcal{R}$.

Lemma 4.5. *Protocol 1 is computational (respectively statistical) zero-knowledge according to Definition 2.27, if the underlying commitment scheme is computationally (respectively statistically) hiding.*

Proof. We construct a simulator $\mathcal{SIM}$, which generates, in expected PPT, a communication tape $\mathcal{R}_s$ whose distribution is indistinguishable from that of $\mathcal{R}$ in a computational or statistical sense (depending on the type of commitments which are used).

Suppose that $\widetilde{V}$ chose a particular strategy depending on the information received from P. Denote this strategy by $St(C_1, C_2, C_3)$.

The simulator $\mathcal{SIM}$ works as follows:

1. Pick a challenge $b \xleftarrow{\$} \{0, 1, 2\}$.
   – If $b = 0$, choose $y \xleftarrow{\$} \mathbb{F}_2^n$, $\pi \xleftarrow{\$} \mathcal{S}_n$, compute $C_1 = Com(\pi, H^{pub}y^T)$, $C_2 = Com(y\pi)$, $C_3 = Com(0)$, and $Rep = (y, \pi)$, where by Rep, we denote the reply of the prover.
   Clearly, the distributions of C_1, C_2, C_3 and Rep are identical to those from the communication tape of the actual protocol.
   – If $b = 1$, choose $y \xleftarrow{\$} \mathbb{F}_2^n$, $\pi \xleftarrow{\$} \mathcal{S}_n$, and $w = y + z$, where $z \in \mathbb{F}_2^n$ is such that $H^{pub}z^T = c$, $z \neq m$, $w_H(z) \neq t$. Note that such the vector w can be computed in polynomial time as shown in [OS09, Proposition 1]. Then, compute $C_1 = Com(\pi, H^{pub}y^T)$, $C_2 = Com(0)$, $C_3 = Com(w\pi)$, and $Rep = (w, \pi)$. It is easy to check that the openings of the above commitments and Rep will pass the verification of Step 3 in Protocol 1, and also that distributions of the commitments and Rep are identical to those in the actual protocol. In particular, in the simulation, the distribution of w is uniform over $\mathbb{F}_2^n$, and hence the contents of C_3 has the distribution identical to that in the Protocol 1.
   – If $b = 2$, choose $y \xleftarrow{\$} \mathbb{F}_2^n$, $\pi \xleftarrow{\$} \mathcal{S}_n$, and $z \xleftarrow{\$} \{x \in \mathbb{F}_2^n | w_H(x) = t\}$. Then, compute $C_1 = Com(0)$, $C_2 = Com(y\pi)$, $C_3 = Com((y + z)\pi)$, and $Rep = (y\pi, z\pi)$. It is again easy to check that the values in Rep will pass the verification of Step 3 in Protocol 1, and that distributions of the commitments and Rep are identical to those in the actual protocol.

2. $\mathcal{SIM}$ computes $b' = St(C_1, C_2, C_3)$.

3. If $b = b'$, then $\mathcal{SIM}$ writes on the tape $\mathcal{R}_s$ the values H^{pub}, b, Rep, otherwise it goes to Step 1.

Note that in the above simulator, in the case of commitments to zero, we use the hiding property of the commitment to ensure that the distributions in question are identical.

We can see that in $3r$ rounds on the average, $\mathcal{SIM}$ produces the communication tape $\mathcal{R}_s$ which is indistinguishable from the communications tape $\mathcal{R}$ produced by the honest parties who execute r rounds of Protocol 1.

We conclude that $\langle P(m), \widetilde{V}\rangle(pk, c)$ and $\langle \mathcal{SIM}, \widetilde{V}\rangle(pk, c)$ are indistinguishable. Note that the simulation is perfect by itself, and the type of indistinguishability, statistical or computational – and hence the type of the ZK proof which we obtain – depends solely on the underlying commitment scheme. □

Using Lemmas 4.2 and 4.5, and the observation on the completeness, we conclude the proof of Theorem 4.1. □

4.3 PPK for McEliece PKE

We construct the proof of plaintext knowledge for the McEliece PKE using Jain et al. ZK identification scheme [JKPT12], using the approach similar to that in the previous subsection. Again, our main observation is that the randomness of the common matrix is not required in the security proof.

We remind that $C(G^{pub})$ denotes the code defined by G^{pub} as a generator matrix. It is easy to check, if a vector $v \in \mathbb{F}_2^n$ belongs to $C(G^{pub})$ by computing its syndrome $syn(v)$ as follows [Rot06]. One can compute the parity-check matrix $H \in \mathbb{F}_2^{n-k \times n}$ corresponding to G^{pub}, i.e., such that $H(G^{pub})^T = \nvDash$, and then $syn(v) = Hv^T$. If $syn(v) = \nvDash$, then v belongs to $C(G^{pub})$, and does not belong to it otherwise.

We note that Jain et al. present an alternative proof using the fact that their protocol is a special variant of Σ-protocols where soundness error is larger than $1/2$. They claimed that the major results for Σ-protocols hold for their variant as well. However, we choose to use the standard security argument for the ZK proof of knowledge. The reason is that in Σ-protocols, a special honest-verifier ZK property is proved, while the security against malicious verifier is obtained via the transformation by Damgård et al. [DGOW95]. However, we observe that the Jain et al. protocol is readily secure against malicious verifier. Therefore, our proposal does not need the aforementioned transformation.

Witness: $m \in \mathbb{F}_2^k$, $e \in \mathbb{F}_2^n$ such that $w_H(e) = t$, where the parameters n and t are described in Section 2.4.4.2.

Common data: (G^{pub}, t) such that $G^{pub} \in \mathbb{F}_2^{k \times n}$ – the public key of the McEliece PKE, and $c = mG^{pub} + e$ – the ciphertext of the McEliece PKE.

Protocol 2 (McEliece PPK).

1. P computes $u \xleftarrow{\$} \mathbb{F}_2^k$, $y \xleftarrow{\$} \mathbb{F}_2^n$, $\pi \xleftarrow{\$} \mathcal{S}_n$, and sends three commitments to V:
   $C_1 = Com(\pi, uG^{pub} + y)$,
   $C_2 = Com(y\pi)$,
   $C_3 = Com((y + e)\pi)$.
2. V sends $b \xleftarrow{\$} \{0, 1, 2\}$.
3. P replies as follows, while V performs the following checks and rejects, if any check fails:
   – If $b = 0$,
   – P sends π, $uG^{pub} + y$, and $y\pi$ and opens C_1 and C_2.
   – V checks validity by computing $(uG^{pub}+y)+(y\pi)\pi^{-1}$ and verifying that it is in $C(G^{pub})$.
   – If $b = 1$,
   – P sends π, $uG^{pub} + y$, and $(y + e)\pi$, and opens C_1 and C_3.
   – V checks validity by computing $(uG^{pub} + y) + ((y + e)\pi)\pi^{-1} + c$ and verifying that it is in $C(G^{pub})$.
   – If $b = 2$,
   – P sends $y\pi$ and $(y + e)\pi$, and opens C_2 and C_3.
   – V checks validity by computing $w_H(y\pi + (y + e)\pi)$ and verifying that it is equal to t.

Theorem 4.6. *The above protocol, which is denoted as* $\mathsf{PPK}_M(G^{pub}, c; m, e)$*, is a proof of plaintext knowledge for the McEliece PKE in the standard model, according to Definition 2.27 assuming that the McEliece problem is hard, and the underlying commitment scheme satisfies Definition 9.*

Proof. In this proof, we combine the techniques from [JKPT12] and [Vér97].

Completeness. It is easy to check that P who knows the plaintext can answer all of three challenges correctly. This implies that $\langle P(m, e), V\rangle(pk, c) = 1$.

Lemma 4.7. *Protocol 2 is sound according to Definition 2.27, if the underlying commitment scheme is binding, the McEliece problem is hard, and* $r(\kappa) = \omega(\log \kappa)$.

This lemma follows from the following two auxiliary lemmas. We will omit mentioning the dependence of r and ϵ on κ, for simplicity.

Lemma 4.8. *If the witness does not exist, then the probability for* $\widetilde{P}$ *to be accepted in the above protocol is at most* $\left(\frac{2}{3}\right)^r$*, after r rounds.*

Proof. We follow the same strategy as in Lemma 4.3.

Consider the following challenge-response pairs:

- $b = 0 : (\pi_0, w_0, w_1)$ (corresponds to π, $mG^{pub} + e$, and $y\pi$, respectively),
- $b = 1 : (\pi_0, w_0, w_2)$ (corresponds to π, $mG^{pub} + e$, and $(y + e)\pi$ respectively),
- $b = 2 : (w_1, w_2)$.

Since the information in the opened commitments is consistent by assumption, and by the binding property of the underlying commitments, we observe that combining the verification for the challenges $b = 0$ and $b = 1$ yields $(w_1 + w_2)\pi_0^{-1} + c \in C(G^{pub})$ and hence $c = m_0 G^{pub} + (w_1 + w_2)\pi_0^{-1}$, where m_0 can be computed by the elementary linear algebra. Since $w_H(w_1 + w_2) = t$ according to the verification for the challenge $b = 2$, a valid witness is computed as $(m_0, (w_1 + w_2)\pi_0^{-1})$. □

The following lemma can be proven using the same counting argument as in Lemma 4.4. Since the proof is very similar to that of Lemma 4.4, it is omitted. We emphasize that, similarly to the previous section, this proof does not require any additional assumptions on the McEliece public key, except for those made in the statement of the McEliece problem in Section 2.4.4.1.

Lemma 4.9. *If V accepts $\widetilde{P}$'s proof with probability at least $(\frac{2}{3})^r + \epsilon$, then there exists an expected PPT algorithm WE which, with overwhelming probability, computes a witness (m, e).*

Zero-knowledge. Let us denote by $\mathcal{R}$ the communication tape for P and V, that is a concatenation of all bits they exchange during the protocol. We consider the probability distributions on $\mathcal{R}$.

Lemma 4.10. *Protocol 2 is computational (respectively statistical) zero-knowledge according to Definition 2.27, if the underlying commitment scheme is computationally (respectively statistically) hiding.*

Proof. We construct a simulator $\mathcal{SIM}$, which generates, in PPT, a communication tape $\mathcal{R}_s$ whose distribution is indistinguishable from that of $\mathcal{R}$ in a computational or statistical sense (depending on the type of commitments which are used).

Suppose that $\widetilde{V}$ chose a particular strategy depending on the information received from P. Denote this strategy by $St(C_1, C_2, C_3)$.

The simulator $\mathcal{SIM}$ works as follows:

1. Pick a challenge $b \xleftarrow{\$} \{0, 1, 2\}$.
   – If $b = 0$, choose $u \xleftarrow{\$} \mathbb{F}_2^k$, $y \xleftarrow{\$} \mathbb{F}_2^n$, $\pi \xleftarrow{\$} \mathcal{S}_n$, and compute $C_1 = Com(\pi, uG^{pub} + y)$, $C_2 = Com(y\pi)$, and $C_3 = Com(0)$, and $Rep = (\pi, uG^{pub} + y, y\pi)$, where by Rep, we denote the reply of the prover.
   Clearly, the distributions of C_1, C_2, C_3 and Rep are identical to those from the communication tape of the actual protocol.
   – If $b = 1$, choose $\pi \xleftarrow{\$} \mathcal{S}_n$, $z \xleftarrow{\$} \mathbb{F}_2^n$, and $x \xleftarrow{\$} \mathbb{F}_2^k$. Then, compute $C_1 = Com(\pi, xG^{pub} + c + z)$, $C_2 = Com(0)$ and $C_3 = Com(z\pi)$, and $Rep = (\pi, xG^{pub} + c + z, z\pi)$. It is easy to check that the openings of the above commitments and Rep will pass the verification of Step 3 in Protocol 2, and also that distributions of the commitments and Rep are identical to those in the actual protocol. In particular, in Protocol 2, P sends $uG^{pub} + y$, where $u \xleftarrow{\$} \mathbb{F}_2^k$ and $y \xleftarrow{\$} \mathbb{F}_2^n$, while in the simulation, we have $xG^{pub} + c + z = (x + m)G^{pub} + (e + z)$, where $x + m$ and $e + z$ are uniform over $\mathbb{F}_2^k$ and $\mathbb{F}_2^n$, respectively.
   – If $b = 2$, choose $z \xleftarrow{\$} \mathbb{F}_2^n$ and $x \xleftarrow{\$} \{a \in \mathbb{F}_2^n | w_H(a) = t\}$, and compute $C_1 = Com(0)$, $C_2 = Com(z)$, $C_3 = Com(z + x)$, and $Rep = (z, x)$. It is again easy to check that the openings of the commitments and Rep will pass the verification of Step 3 in Protocol 1, and that distributions of the commitments and Rep are identical to those in the actual protocol.

2. $\mathcal{SIM}$ computes $b' = St(C_1, C_2, C_3)$.

3. If $b = b'$, then $\mathcal{SIM}$ writes on the tape $\mathcal{R}_s$ the values G^{pub}, b, Rep, otherwise it goes to Step 1.

In the above simulator, in the case of commitments to zero, we use the hiding property of the commitment to ensure that the distributions in question are identical.

We can see that in $3r$ rounds on the average, $\mathcal{SIM}$ produces the communication tape $\mathcal{R}_s$ which is indistinguishable from the communication tape $\mathcal{R}$ produced by the honest parties who execute r rounds of Protocol 2.

We conclude that $\langle P(m), \widetilde{V}\rangle(pk, c)$ and $\langle \mathcal{SIM}, \widetilde{V}\rangle(pk, c)$ are indistinguishable. Note that, as in Lemma 4.5, the simulation is perfect by itself, and the type of indistinguishability, statistical or computational – and hence the type of ZK proof which we obtain – depends solely on the underlying commitment scheme. □

Using Lemmas 4.7 and 4.10, and the observation on the completeness, we conclude the proof of Theorem 4.6. □

4.4 Performance Evaluation

In this section, we estimate the security and performance of the proposed schemes. Since breaking security of the Niederreiter PKE is polynomially equivalent to breaking security of the McEliece PKE [LDW94], the parameter sets (n, k, t) provide the same security levels. The difference will be in the public key sizes, plaintext and ciphertext sizes and the communication and computational costs.

In fact, both the public key and the ciphertext sizes are smaller for the Niederreiter PKE, as compared to the McEliece PKE, for the same security level. Therefore, the Niederreiter PKE is more suitable for the lightweight security applications [SCKI10].

For our security evaluation, we chose the estimation of the currently best attack for the relevant parameters - the information set decoding algorithm proposed and implemented by Peters [Pet10, Pet]. We first choose the code length n (as a power of 2, for convenience) and $k = n - t \log_2 n$ (again for convenience), then we find the smallest value of t which provides us with 80 or 128 bits of security equivalent to symmetric encryption. Specifically, we choose the parameters (n, k, t) according to [Pet10, Pet] as shown in Table 4.1. We fix the soundness failure probability in our PPK to be at most 2^{-16}, since this value is a minimal requirement in the ISO/IEC-9798-5 standard for the zero-knowledge techniques for entity authentication. Remembering that the soundness failure probability is $2/3$ in each round, we will need 28 rounds in total.

4.4.1 PPK for Niederreiter PKE

We consider the original Niederreiter public key encryption, with binary irreducible Goppa codes used for generation of public keys, without any optimizations. Then, the size of the public key is $(n - k)n$ bits, the ciphertext size is $n - k$ bits.

In order to keep estimation simple, we assume the random oracle model, and construct commitments using idealized hash functions $h : \{0, 1\}^* \to \{0, 1\}^{l_c}$, taking $l_c = 160$. In order to commit to a value x (we will think of a binary representation of x), P will simply compute $h(x)$. Then,

Table 4.1: Code Parameters for Performance Evaluation of Code-Based PPK.

Equivalent security (bits)	80	128
Code length n	2048	4096
Code dimension k	1806	3676
Weight of error vector t	22	35

V's checks in our protocol will be performed as in [Ste96] by computing the corresponding hash values. For instance, we will compute $C_1 = h(\pi, H^{pub}y^T)$, and when $b = 0$, V will use the values π and y received from P to compute $h(\pi, H^{pub}y^T)$ and check whether it is equal to C_1 received in Step 1. In our estimations, we follow the approach of [CVA10] and implement the commitments without randomization due to the fact that we commit only to random values. As in [CVA10], we assume that the representations of y and π, respectively are generated using pseudorandom generators with seed length $l_s = 128$. Sending seeds instead of actual values will help us to save on communication.

The expected communication cost of our PPK for the Niederreiter PKE is as follows:

$$r(3l_c + 2 + l_s + n) \text{ bits},$$

where r is the number of rounds. We call it "expected", since the challenges are supposed to be random and so that we average over the sizes of replies to the three challenges. Specifically, per round, 3 commitments contribute $3l_c$ bits, the challenge contributes 2 bits, and the third term is the average over the replies, remembering that the seeds of pseudorandom generators are used to send y and π yielding $(3l_s + 3n)/3$ which gives us $l_s + n$.

Prover's computation cost, which is also an upper bound on that cost for the verifier, is calculated as follows:

$$r((n-1)(n-k) + n) \text{ binary summations}.$$

The dominant term is given by $H^{pub}y^T$ contributing at most $n - 1$ summations of columns of H^{pub}, which are $(n-k)$-bit vectors. In addition, n summations are contributed by the term $y + e$. This is just a rough estimate of the total computation cost, since the costs of computing permutations, commitments, and pseudorandom generation – which are implementation-specific – are not included. For completeness, we note that our protocol requires 3 invocations of commitment, 2 invocations of pseudorandom generator and 2 permutations of n elements per round.

Our recommended parameters and the resulting costs are summarized in Table 4.2.

Table 4.2: Parameters and Performance of the Niederreiter PPK Protocol.

Equivalent security (bits)	80	128
Public key size (Kbytes)	61	210
Ciphertext size (bits)	242	420
Communication (Kbytes)	9.1	16.1
Prover's computation (operations over $\mathbb{F}_2$)	$2^{23.73}$	$2^{25.52}$

We can see that the communication cost of the proposed protocol appears to be within the practical feasibility range. For instance, for the equivalent 80 bit security, it is 9.1 kilobytes. For comparison, it is by 3% smaller than the (non-optimized) public key size of the Niederreiter PKE, which is equal to 62.3 kilobytes for the relevant parameters.

4.4.2 PPK for McEliece PKE

Making calculation in the same setting as in the previous section, we estimate the performance of our PPK for the McEliece PKE. We consider the original McEliece PKE, with binary irreducible Goppa codes used for generation of public keys, without any optimizations. Then, the size of the public key is kn bits, the ciphertext size is n bits.

The expected communication cost of our proposal is:

$$r(3l_c + 2 + 2l_s/3 + 2n) \text{ bits},$$

where r is the number of rounds. The last two terms are computed by averaging over the replies yielding $(2l_s + 6n)/3$ which gives us $2l_s/3 + 2n$.

Prover's computation cost, which is also an upper bound on that cost for the verifier, is calculated as follows:

$$r(k + 1)n \text{ binary summations}.$$

Henceforth, we consider only summations, since we are working over $\mathbb{F}_2$, and therefore the scalar product of two binary vectors x and y can be implemented as the summation of the bits of, say, x corresponding to that of y which are non-zero. Hence, the binary multiplication can be avoided.

The computation of uG^{pub} contributes at most $(k - 1)n$ summations of rows of G^{pub}, which are n-bit vectors. Moreover, adding y to uG^{pub}, as well as computing of $y + e$ contributes $2n$ summations. Altogether, it gives us at most $(k + 1)n$ summations per round. As in the previous subsection, the costs of computing permutations, commitments, and pseudorandom generation

Table 4.3: Parameters and Performance of the McEliece PPK Protocol.

Equivalent security (bits)	80	128
Public key size (Kbytes)	452	1838
Ciphertext size (bits)	2048	4096
Communication (Kbytes)	16.0	30.0
Prover's computation (operations over $\mathbb{F}_2$)	$2^{26.63}$	$2^{28.65}$

– which are implementation-specific – are not included. For completeness, we note that our protocol requires 3 invocations of commitment, 1 invocation of pseudorandom generator and 2 permutations of n elements per round.

Our recommended parameters and the resulting costs are summarized in Table 4.3. As for the previous protocol, we obtain practically feasible communication cost, for instance, for the equivalent 80 bit security, it is 16 kilobytes.

4.5 Conclusion

We presented the first proof of plaintext knowledge for the code-based PKE's by Niederreiter and by McEliece, and applied it to achieve the first code-based verifiable McEliece PKE. A natural extension of this work is to construct PPK for q-ary versions of the McEliece and the Niederreiter PKE's, for instance, using the identification scheme by Cayrel et al. [CVA10].

The construction of the verifiable Niederreiter PKE remains an open problem. Another important open question is to reduce the communication cost and the round complexity of our proposals.

Chapter 5

Verifiable Code-Based Encryption

5.1 Background

Informally, a Verifiable encryption is a two party protocol between a prover P and a verifier V which allows the former to convince the latter that it knows some secret piece of information without revealing anything about it. A bit more precisely, in a verifiable proof for a binary relation R, the parties have common input y and the prover has private input s such that $(y; s) \in$ R. The protocol must then satisfy the following three properties:
(i) For an honest prover, the verifier always accepts (completeness).
(ii) For every potentially malicious verifier V' there exists a PPT simulator only taking y asan input whose output is indistinguishable from conversations of V' with an honest prover(Validity).
(iii) From every prover P which can make the verifier accept with a probability larger than a thresholdω (the knowledge error), a s' satisfying $(y; s') \in$ R can be extracted efficiently in a rewindable blackbox way (proof of knowledge).

5.2 Our Protocol

Note that in the case of the original McEliece encryption, the problem of checking the equality relation for some $m' \in \mathbb{F}_2^k$ is trivial and does not require interactions. Specifically, given the ciphertext $c = mG^{pub} + e$ (as defined in Section 2.4.4.1), one can compute $y = m'G^{pub} + c$, and check if $w_H(y) = t$. Then, $y = (m + m')G^{pub} + e$, and if $m' = m$, we have that $y = e$. At the same time, if $m' \neq m$, then taking into account that $C(G^{pub})$ by construction has a minimum distance at least $2t + 1$, we have that $w_H((m + m')G^{pub}) \geq 2t + 1$ and hence $w_H(y) \geq t + 1$.

Based the above idea, we apply the PPK for the McEliece PKE to obtain the verifiable McEliece encryption for an equality relation in Section 4. Informally, P can use this protocol to convince

V that given a binary vector m', a ciphertext c and a public key pk, the vector m' is indeed the plaintext which is contained in c computed under pk. Verifiable encryption for more general relations on the encrypted data is briefly discussed in Section 5.2.1. However, the same technique as above cannot be directly applied to the Niederreiter PKE – we discuss this point in Section 5.3.

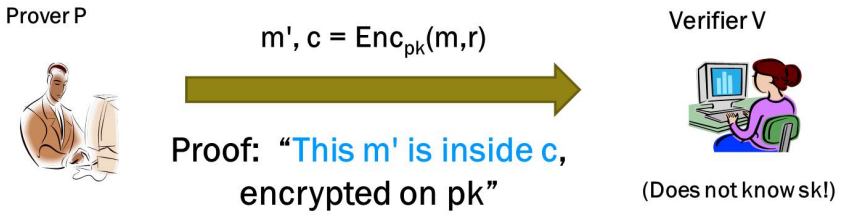

Figure 5.1: Verifiable Encryption

5.2.1 Proof of Equality for McEliece Plaintext

Jain et al. [JKPT12] present the efficient zero-knowledge proofs for relations between committed values. Specifically, their results allow one to prove that certain committed messages $m_0, m_1, \ldots, m_l$ satisfy $m_0 = C(m_1, \ldots, m_l)$, where C is a boolean circuit with l inputs.

Observing that their commitments have the same structure as the IND-CPA McEliece encryption [NIKM08], except that a random matrix is used rather than the McEliece public key, it may be possible to extend the above protocol to verifiable encryption for general relations. Informally, their protocol works as follows: First, P runs PPK on the input ciphertext $c = (r|m)G^{pub} + e = rG_0^{pub} + mG_1^{pub} + e$, where $(G^{pub})^T = [(G_0^{pub})^T, (G_1^{pub})^T]$, $G_0^{pub} \in \mathbb{F}_2^{k_0 \times n}$ and $G_1^{pub} \in \mathbb{F}_2^{k_1 \times n}$ are the sub-matrices of G^{pub} corresponding to randomness and plaintext, respectively. Secondly, both players compute $c' = c + mG_1^{pub} = rG_0^{pub} + e$, hereby canceling out the plaintext, and then run PPK with c' as ciphertext and G_0^{pub} as public key.

Now, we can see that Protocol 1 can be used as PPK in the above construction. Hereby, we fix the McEliece verifiable encryption scheme of [MT12].

Remark 5.1. It was noted in [MT12], that although the resulting PPK is zero-knowledge, the fact that V learns m (together with the fact that c is a valid McEliece ciphertext) implies that $\widetilde{V}$ can now produce a valid encryption $c_a = rG_0^{pub} + e + m_a G_1^{pub}$ for an arbitrary $m_a \in \mathbb{F}_2^k$. Note that although the IND-CPA McEliece PKE is clearly malleable, the above attack is not feasible for $\widetilde{V}$ prior to the protocol execution. This is not a problem of the protocol, but rather the property of the IND-CPA McEliece encryption. Therefore, some authentication technique must be applied in order to avoid this attack. We will need to prove the following statement: "A ciphertext c encrypted on G^{pub} contains the plaintext m'", with the mentioned values being public, and the

witness being $((r|m), e)$. This can be directly achieved using the proof for affine relation R_{ALPN} defined in [JKPT12, Sec. 4.2].

The proof of inequality will introduce at next chapter. It will use the proof that the weight of the error vector in the McEliece encryption is upper-bounded by specific value – such the proof is presented in Jain et al. for proving valid opening of commitments based on LPN problem.

5.2.2 Our Verifiable Encryption

The fact that c is properly formed can be verified using the PPK for the McEliece PKE.

Therefore, we assume that the IND-CPA variant of the McEliece PKE, as described in Section 2.4.4.3, is used for encryption. Specifically, the ciphertext is computed as $c = (r|m)G^{pub} + e$, where in particular, $r \in \mathbb{F}_2^{k_0}$, $m \in \mathbb{F}_2^{k_1}$, $k = k_0 + k_1$. Moreover, we assume that the ciphertext presented by the prover is correctly formed. In principle, this assumption can be removed, if we require the prover to run the McEliece PPK Protocol as $\mathsf{PPK}_M(G^{pub}, c; (r|m), e)$, i.e., the part of the witness m in McEliece PPK Protocol will be replaced with $(r|m)$. We observe that here, a PPK will play the role of the proof of correctness of the ciphertext.

We make an additional assumption that all the rows of the public key $G^{pub} \in \mathbb{F}_2^k$ are linearly independent. However, this assumption is natural as the public key is typically constructed this way.

Witness: $r \in \mathbb{F}_2^{k_0}$, $m \in \mathbb{F}_2^{k_1}$, and $e \in \mathbb{F}_2^n$, where the parameters k_0, k_1 and n are described in Section 2.4.4.3.

Common data: $m' \in \mathbb{F}_2^k$, (G^{pub}, t) such that $G^{pub} \in \mathbb{F}_2^{k \times n}$ – the public key the IND-CPA McEliece PKE, and $c = (r|m)G^{pub} + e$ – the ciphertext of the IND-CPA McEliece PKE, where $r \in \mathbb{F}_2^{k_0}$, $m \in \mathbb{F}_2^{k_1}$, $k = k_0 + k_1$.

Consider the partition of the public key as follows $(G^{pub})^T = ((G_0^{pub})^T | (G_1^{pub})^T)$, where $G_0^{pub} \in \mathbb{F}_2^{k_0 \times n}$ and $G_1^{pub} \in \mathbb{F}_2^{k_1 \times n}$ are the sub-matrices of G^{pub} corresponding to the randomness and the plaintext, respectively.

(Verifiable McEliece PKE).

1. P and V both compute $c' = c + m' G_1^{pub}$.
2. P and V run the protocol $\mathsf{PPK}_M(G_0^{pub}, c'; r, e)$.

Theorem 5.2. *The above protocol is the verifiable McEliece PKE for equality relation in the standard model, according to Definition 2.28 assuming that the variant of the McEliece PKE described in Section 2.4.4.3 is IND-CPA.*

Proof. **Completeness.** If $c = (r|m)G^{pub} + e$, then we can write $c' = rG_0^{pub} + mG_1^{pub} + e + m'G_1^{pub}$. Therefore, $m' = m$ implies $c' = rG_0^{pub} + e$ and hence the completeness will hold similarly to that in Theorem 4.6.

Soundness. Suppose that $m' \neq m$, then we have that $c' = rG_0^{pub} + e + (m + m')G_1^{pub}$, where $m + m' \neq \nvdash$. Since $C(G_1^{pub})$ is a sub-code of $C(G^{pub})$, it has the same minimum weight, and then $w_H((m + m')G_1^{pub}) \geq 2t + 1$ so that $w_H((m + m')G_1^{pub} + e) \geq t + 1$, and therefore the valid witness does not exist. The only possibility for this argument to fail is when a codeword from $C(G_0^{pub})$ is equal to a codeword from $C(G_1^{pub})$, but this would contradict the assumption that the rows of G^{pub} are linearly independent.

Then, the soundness will follow by the same argument as in Theorem 4.6, taking into account that the IND-CPA property [NIKM08] of the variant of the McEliece PKE described in Section 2.4.4.3 implies that it is OW-CPA.

Zero-Knowledge. This property will follow by the same argument as in Theorem 4.6. □

Informally, their protocol works as follows: First, P runs PPK on the input ciphertext $c = (r|m)G^{pub} + e = rG_0^{pub} + mG_1^{pub} + e$, where $(G^{pub})^T = [(G_0^{pub})^T, (G_1^{pub})^T]$, $G_0^{pub} \in \mathbb{F}_2^{k_0 \times n}$ and $G_1^{pub} \in \mathbb{F}_2^{k_1 \times n}$ are the sub-matrices of G^{pub} corresponding to randomness and plaintext, respectively. Secondly, both players compute $c' = c + mG_1^{pub} = rG_0^{pub} + e$, hereby canceling out the plaintext, and then run PPK with c' as ciphertext and G_0^{pub} as public key.

Now, we can see that Protocol 1 can be used as PPK in the above construction. Hereby, we fix the McEliece verifiable encryption scheme of [MT12].

Remark 5.3. It was noted in [MT12], that although the resulting PPK is zero-knowledge, the fact that V learns m (together with the fact that c is a valid McEliece ciphertext) implies that $\widetilde{V}$ can now produce a valid encryption $c_a = rG_0^{pub} + e + m_aG_1^{pub}$ for an arbitrary $m_a \in \mathbb{F}_2^k$. Note that although the IND-CPA McEliece PKE is clearly malleable, the above attack is not feasible for $\widetilde{V}$ prior to the protocol execution. This is not a problem of the protocol, but rather the property of the IND-CPA McEliece encryption. Therefore, some authentication technique must be applied in order to avoid this attack.

5.3 Discussion

We note that the approach from Section 5.2.1 cannot be generalized immediately to the case of the IND-CPA Niederreiter PKE. Consider the ciphertext $c = H^{pub}(r|m)^T$ as described in Section 2.4.4.2. We have $c = [H_0^{pub}|H_1^{pub}](r|m)^T = H_0^{pub}r^T + H_1^{pub}m^T$, where $H_0^{pub} \in \mathbb{F}_2^{n-k\times n_0}$ and $H_1^{pub} \in \mathbb{F}_2^{n-k\times n_1}$ are the sub-matrices of H^{pub} corresponding to the randomness and the plaintext, respectively. Firstly, Let us consider the players will first run Protocol PPK for Nie with c, H^{pub} and m as inputs, then will compute $c' = c + H_1^{pub}m^T$. Secondly, the players will run Protocol 1 with c', H_0^{pub} and r as inputs. Thirdly, Suppose that $\widetilde{P}$ presents $m' \in \mathbb{F}_2^{n_1}$ such that $m' \neq m$ and $w_H(m') = t_1$ (if the weight is different from t_1, $\widetilde{V}$ must reject him). They get the results are $c' = c + H_1^{pub}(m')^T = H_1^{pub}(m + m')^T$, where $m + m' \neq \not{\epsilon}$. In order to use the same argument, we need to require that the columns of H^{pub} are linearly independent, which is clearly impossible.

Observing that the columns of H^{pub} are non-zero by construction [EOS07, Mat80] (otherwise the "equivalent" code length would be smaller than n), but the columns are not linearly independent, the cheating prover will succeed because c will be consisted with c', but the honest prover also use it, the proof can not distinguish.

we conclude that $c' \neq H_0^{pub}r^T$ and therefore, $\widetilde{P}$ must not succeed in PPK of Protocol 1 by the soundness property. Remark 5.3 applies to the above construction as well.

Chapter 6

Designated Confirmer Signatures from Codes

6.1 Background

In this work, we focus on cryptographic primitives whose security is based on hardness of decoding random codes. Such the primitives are prospective candidates for postquantum cryptography which must resist attacks using quantum computation.

6.1.1 Related Works

Okamoto [Oka94] showed that DCS and public key encryption are equivalent. His construction used commitment and digital signature as ingredients. Conformation and denial were preformed using general zero-knowledge proofs. A number of works followed with efforts directed at improving efficiency and fixing security model [MS98, CM00, GW04, GMR05, WBWB07].

In a recent work [EA10], El Aimani proposed a "Signature of Encryption" paradigm for constructing DCS. It required EUF-CMA signature and IND-CPA PKE.

6.1.2 Contribution of This Chapter

We construct the first code-based designated confirmer signature in the random oracle model. As IND-PKE, we use the Randomized McEliece PKE by Nojima et al. [NIKM08]. As EUF-CMA signature [Kat10], we may use the scheme obtained by applying Fiat-Shamir transform [FS87] to a code-based identification scheme [Ste96, JKPT12]. We emphasize that the random oracle model is only required for the Fiat-Shamir transform. In other words, given a code-based

EUF-CMA signature in the standard model, our scheme would be also secure in this model. Unfortunately, we are not aware of such the signature, at the moment. In a further aspect, we should claim our scheme is the first *efficient* code-based DCS, because straightforwardly instantiating the El Aimani's generic construction with code-based primitives will require general zero-knowledge proofs, which are inefficient.

For confirmation, respectively denial, a confirmer needs to prove to a verifier an equality, respectively an inequality, of a given message to that contained in a given McEliece plaintext. For the proof of equality, we will directly use the zero-knowledge proof of affine relations from [JKPT12]. Note that Jain et al. devise their proof in particular based on what they call xLPN problem, where error vector has fixed weight. This is equivalent to the problem of decoding random linear codes.

We construct the proof of inequality similarly to the construction of Hu et al. [HMT13]. In fact, the idea by [HMT13] that the proof of equality was independently presented by Hu et al. [HMT13]. It was less efficient compared to that of [JKPT12] as it consisted of two steps, each of which used an independent zero-knowledge protocol. We adapt their idea for ZK proof of inequality. This primitive may be of independent interest.

6.2 Security Model and Definitions

6.2.1 Security Model

Now we introduce the details of the security model and definitions of DCS which the confirmer do not have communication with signer. Specifically, the syntax of DCS is the same as given in Gentry et al.′s exposition in [GMR05]. In general, a DCS scheme has three different roles of parties: a signer S, a verifier V, and a confirmer C.

Definition 6.1. (**Syntax**). A typical designated confirmer signature (DCS) scheme consists of a tuple $(DCGen, Sign, Verify, Extract, ComfirmedSign_{(S,V)}, Confirm_{(C,V)}, Disavowal_{(C,V)})$ which contains probabilistic polynomial-time algorithms and interactive protocols, we will describe the specific details as follow.

- *DCGen*: With the security parameter 1^{λ} input and output two pair of keys (sk_S, pk_S) and (sk_C, pk_C) where (sk_S, pk_S) are the signer S's signing and verification keys respectively, while (sk_C, pk_C) are the confirmer C's private and public keys respectively.
- *Sign*: Input a message m and a signing key sk_C, and outputs a basic signature σ ensure that $Verify(m, \sigma, pk_S)$ is valid. We call this state when the signature is valid the state $ACCEPT$.

- $Vertify$: Input a triple tuple (m, σ, pk_S), and outputs $ACCEPT$ if σ is an output of $Sign(m, sk_S)$ or $\perp$ otherwise.

- $Extract$: Input a tuple (m, σ', sk_C, pk_S), and outputs is string σ, such $Verify(m, \sigma, pk_S) = ACCEPT$ that σ is an output of $Sign(m, sk_S)$, otherwise $\perp$ is the output.

- $ConfirmedSign_{(S,V)}$: An interactive protocol has two parties, one is the signer S (with private input sk_S), the other is a verifier V, they have the common input (m, pk_S, pk_C). The output of V is a pair (b, σ') where $b \in \{ACCEPT, \perp\}$ and σ' is S's designated confirmer signature on message m. For some verifier V, the $ConfirmedSign$ protocol should be complete and sound.

  Completeness : There is some signer S such that for any (valid) signer and confirmer keys, any message m, the $ConfirmedSign$ protocol outputs $(ACCEPT, \sigma')$, where $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S)$ outputs $ACCEPT$.

  Soundness : For any signer S', if $ComfirmedSign_{(S',V)}(m, pk_S, pk_C)$ outputs $(ACCEPT, \sigma')$, then

  $$Pr[Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = \perp] < negl(\lambda). \tag{6.1}$$

  This property ensures that even a cheating signer S' cannot convince an honest verify V that an "un-extractable" DCS σ' is valid.

  There are two different protocols between confirmer and verifier.

- $Confirm_{(C,V)}$: When the confirmer C check the signature DCS σ' is valid, he will use $Confirm_{(C,V)}$ protocol to confirm it. The common input of this protocol is (m, σ', pk_S, pk_C), the and the input sk_C for C. The output is $b \in \{ACCEPT, \perp\}$. We show the confirm protocol is both complete and sound.

  Completeness : Suppose $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = ACCEPT$, then $Confirm_{(C,V)}(m, \sigma', pk_S, pk_C)$ will also output $ACCEPT$.

  Soundness : For any confirmer C', suppose $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = \perp$,then

  $$Pr[Confirm_{(C',V)}(m, \sigma', pk_S, pk_C) = ACCEPT] < negl(\lambda). \tag{6.2}$$

  This property ensures that even a cheating confirmer C' cannot convince an honest verifier V that an "un-extractable" DCS σ' is valid.

- $Disavowal_{(C,V)}$: For the confirmer C and a verifier V, $Disavowal_{(C,V)}$ disavow an invalid DCS σ' with the input (m, σ', pk_S, pk_C) and sk_C input into C. The output is $b \in \{ACCEPT, \perp\}$. We show the $Disavowal$ must be complete and sound.

  Completeness : Suppose $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = ACCEPT$, then $Disavowal_{(C,V)}(m, \sigma', pk_S, pk_C)$ will also output $ACCEPT$.

Soundness : For any confirmer C', suppose $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = \perp$,then

$$Pr[Disavowal_{(C',V)}(m, \sigma', pk_S, pk_C) = ACCEPT] < negl(\lambda). \tag{6.3}$$

This property ensures that even a cheating confirmer C' cannot convince an honest verifier V that an "extractable" DCS σ' is valid.

Above we showed three security requirements of a DCS scheme. To sum up, A DCS scheme should include three features. First, it should be secure for verifiers, confirmed DCS should be un-extractable. Second, it should be secure for signer, that is, for anyone but for signer, no one could forge a DCS on a message unsigned by the signer. Third, the scheme should ensure that only the confirmer can confirm or disavowal an alleged DCS.

A new two-move protocol $OutputDCS_{(s,v)}$ is introduced though it is only a version of the former $ConfirmSign_{(S,V)}$ for the sake of security [GMR05]. In which V queries m and S outputs a DCS σ' on m without confirming its correctness. Now suppose the adversary $\mathcal{A}$ is allowed to access all oracles of O =($Extract$,$ComfirmedSign_{(S,\mathcal{A})}$,$Confirm_{(C,\mathcal{A})}$,$Disavowal_{(C,\mathcal{A})}$) with restraint condition

- The adversary can receive a confirmed signature by his own choice.
- The adversary can execute the interactive $Confirm_{(C,\mathcal{A})}$.
- The adversary can run the interactive protocol $Disavowal_{(C,\mathcal{A})}$.
- The adversary can get a signature from a designated confirmer signature through the $Extract$ oracle.

The security for verifiers requires that even if the adversary $\mathcal{A}$ compromises the private keys of both the confirmer C and the signer S simultaneously, it is still unable to create a pair (m, σ'). We define the situation as the unfoolability. By the way, all of the definitions as follow we keep them as [GMR05], since the [WBWB07, Wik07, EA08, EA10] also kept the same as [GMR05].

Definition 6.2. (**Unfoolability: security for Verifiers**). We say a DCS scheme is secure for verifiers if for any PPT algorithms $\mathcal{A}$ involved in the experiment **Exp1-Unfoolverifier**, its advantage $Adv^{fool}\mathcal{A} := Pr[b_{fool} = 1] < negl(\lambda)$, where b_{fool} is the one bit information returned by the experiment.

Exp1-UnfoolVerifier:

1. $(sk_S, pk_S, sk_C, pk_C) \leftarrow DCGen(1^\lambda)$

2. $(m, \sigma', \tau_1, \tau_2, \tau_3) \leftarrow \mathcal{A}_0^{\mathcal{O}}(sk_S, sk_C, 1^\lambda, pk_S, pk_C)$
3. $(b_1, \sigma') \leftarrow ConfirmedSign_{(\mathcal{A}_1(\tau_1),V)}(m, 1^\lambda, pk_S, pk_C)$ in Case 1
4. $b_2 \leftarrow Confirm_{(\mathcal{A}_2(\tau_2),V)}(m, \sigma', 1^\lambda, pk_S, pk_C)$ in Case 1
5. $b_3 \leftarrow Disavowal_{(\mathcal{A}_3(\tau_3),V)}(m, \sigma', 1^\lambda, pk_S, pk_C)$ in Case 2
6. Return $b_{fool} = (b_1 = ACCEPT \vee b_2 = ACCEPT \vee b_3 = ACCEPT)$

Here Case 1 and Case 2 refer to the restraint conditions on the adversary′s output (m, σ'):

Case 1 : $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = \perp$, i.e.,$\sigma'$ is un-extractable.

Case 2 : $Verify(m, Extract(m, \sigma', sk_C, pk_S), pk_S) = ACCEPT$, i.e.,$\sigma'$ is extractable.

Next we show the definition of the security for the signer, For an adversary, $\mathcal{A}$ must not to forge a valid DCS pair (m, σ') for a new message m, even he may have the ability to create an extractable or confirmable (m, σ'') for a signed message m.

Definition 6.3. (**Unforgeability: Security for the Signer**). We say a DCS scheme is secure is secure for the signer if for any PPT adversary $\mathcal{A}$ involved in the following experiment **Exp2-Unforge**, its advantage $Adv^{forge}(\mathcal{A}) := Pr[b_{forge} = 1] < negl(\lambda)$, where b_{forge} is the one bit information returned by the experiment.

Exp2-Unforge:

1. $(sk_S, pk_S, sk_C, pk_C) \leftarrow DCGen(1^\lambda)$
2. $(m, \sigma') \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda, pk_S, pk_C, sk_C)$
3. $b \leftarrow Verify(m, \sigma, pk_S)$ for $\sigma = EXtract(m, \sigma', sk_C, pk_S)$
4. Return $b_{forge} = (b = ACCEPT \wedge (m, \sigma') \notin L_{sig})$

Where L_{sig} denotes the list of all message-signature pairs (m_i, σ_i') output by the $ConfirmedSign$ Oracle in Step 2 and all (m_i, σ_i'') such that $\sigma_i' = \sigma_i''$

In the definition of the secure for the confirmer, we assume that $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_1'$ represent verifier V_1, verifier V_2 and a simulation algorithm, respectively. If $\mathcal{A}_2$ has only negligible advantage to guess whether its input τ came from , $\mathcal{A}_1$ or , $\mathcal{A}_1'$, which means , $\mathcal{A}_1$'s potentially authentic transcript showing that m_0 was signed is no more convincing or informative than , $\mathcal{A}_1'$'s simulated transcript showing that m_1 was signed.

Definition 6.4. (**Transcript Simulatablity: Securtiy for the Confirmer**). We say a DCS scheme is secure for the confirmer if for any PPT adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ involved in the following experiment **Exp3-Transcript Simulatability**, there exists a PPT algorithm $\mathcal{A}'_1$ such that $\mathcal{A}$'s advantage respect to $\mathcal{A}'_1$ is negligible int he security parameter. That is, $Adv^{trans}(\mathcal{A}, \mathcal{A}'_1) := |Pr[b_{trans} = 1] - 1/2| < negl(\lambda)$, where b_{trans} is the one bit information returned by the experiment.

Exp3-TranscriptSimulatability:

1. $(sk_S, pk_S, sk_C, pk_C) \leftarrow DCGen(1^\lambda)$
2. $(m_0, m_1, s) \leftarrow \mathcal{A}_0^{O}(1^\lambda, pk_S, pk_C, sk_C, sk_S)$
3. $b \leftarrow_R \{0, 1\}$
4. $(ACCEPT, \sigma') \leftarrow ConfirmedSign(1^\lambda, pk_S, pk_C, m_b)$
5. If $b = 0$, $\tau \leftarrow \mathcal{A}_1^{O_1}(1^\lambda, pk_S, pk_C, b, m_0, m_1, s, \sigma')$;
   else, $\tau \leftarrow \mathcal{A}_1'^{OutputDCS}(1^\lambda, pk_S, pk_C, b, m_0, m_1, s, \sigma')$
6. $b' \leftarrow \mathcal{A}_2^{O_2}(1^\lambda, pk_S, pk_C, b, m_0, m_1, \tau, \sigma')$
7. Return $b_{trans} = ((b' = b) \wedge ((m_0, \sigma') \notin L_{ext}) \wedge ((m_1, \sigma') \notin L_{ext}))$

where L_{ext} is a list consisting of each (m_i, σ'_i) that has been queried by $\mathcal{A}_1$ to the *Extract* oracle.

Definition 6.5. (**Security of a DCS Scheme**). We say a designated confirmer signature scheme is secure, if it satisfies security for verifier, the signer and the confirmer we mentioned in Definition 6.2, 6.3 and 6.4.

6.2.2 Proof of Inequality for McEliece Plaintexts

The prover needs to convince the verifier that a plaintext m' is not contained in a ciphertext $c = (r|m)G^{pub} + e$ encrypted using the key G^{pub}. The idea for the following construction comes from [HMT13]. Let $(G^{pub})^T = ((G_0^{pub})^T|(G_1^{pub})^T)$, $G_0^{pub} \in \mathbb{F}_2^{k_0 \times n}$ and $G_1^{pub} \in \mathbb{F}_2^{k_1 \times n}$ are the sub-matrices of G^{pub} corresponding to randomness and plaintext, respectively.

1. A prover completes the proof of valid opening with (c, G^{pub}) as a public data and $((r|m), e)$ as witness.

2. Both prover and verifier compute $c' = c + m'G_1^{pub}$.

3. A prover completes the proof of valid opening for LPN [JKPT12] with (c', G_0^{pub}) and $t + 1 \leq w \leq n$ as a public data and $((r|m), e)$ as witness.

In [JKPT12], the proof of valid opening for LPN allows the weight of the error vector to lie withing a certain range rather than being fixed.

The first ZK proof works as a proof of correctness of the ciphertext c. The second ZK proof assures that $m \neq m'$, since should $m = m'$, the contribution from the part G_1^{pub} will be canceled yielding $c' = rG_0^{pub} + e$. Hence the weight of error vector will be t, violating the soundness of valid opening. On the other hand, if indeed $m \neq m'$, then $(m+m')G_1^{pub}$ is of weight at least $2t+1$ (as it is a codeword of G) resulting in $c' = rG_0^{pub} + e + (m + m')G_1^{pub}$, where the weight of the "error vector" $e + (m + m')G_1^{pub}$ is at least $t + 1$ satisfying correctness.

6.3 Our Protocol

This scheme closely follows the "Signature of an Encryption" paradigm by El Aimani [EA10].

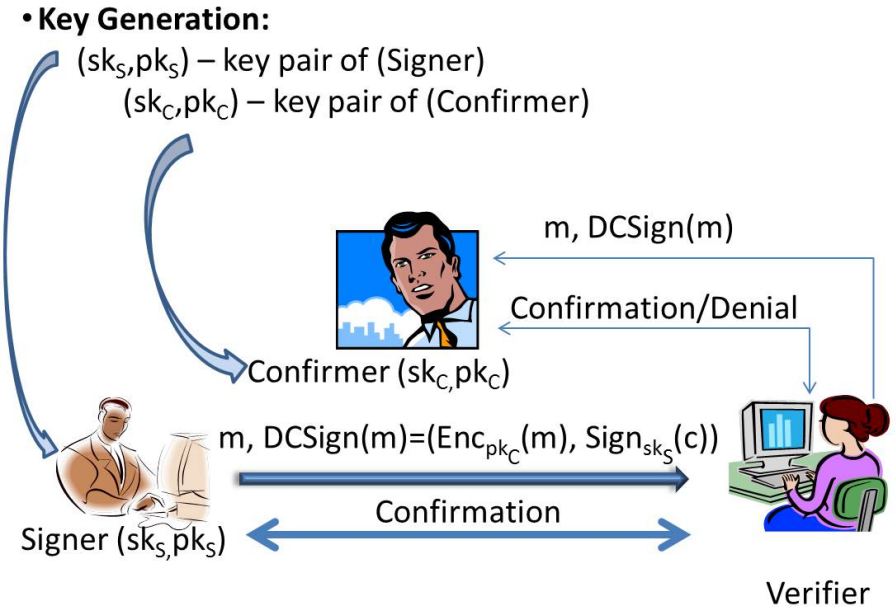

Figure 6.1: Designated Confirmer Signatures

Key Generation: Signer's key pair is (sk_S, pk_S) - the keys of EUF-CMA signature scheme; Confirmer's key is (sk_C, G^{pub}) - the keys of IND-CPA PKE scheme.

ConfirmedSign: On input message m, compute $c = Enc(m, G^{pub})$, and its (ordinary) signature $s = Sign(m, sk_S)$.

Confirmation/Denial: On input a message m, and a signature $\sigma = (\sigma_0, \sigma_1)$, the Confirmer checks validity of σ_1 on σ_0 using pk_S. If invalid, output $\perp$. Otherwise, decrypt m' from σ_0 using sk_C, check if m is equal to m' and use the proof of Sec. 5.3 for confirmation or the proof of Sec. 6.2.2 for denial. Selective conversion: The confirmer with the input sk_C, pk_C and pk_S, the output is $\perp$ or a string which can be universally verified as a valid digital signature. This is an algorithm whose constructions from the "encryption of a signatur" paradigm, to a proof that a given ciphertext decrypts to a given message. Selective verification: This is an algorithm for verifying converted signatures. It inputs the converted signature, the message and pk_S, the outputs is either $\perp$ or Accept.

6.4 Security Proof

In this section, we will follow the proof from [EA08, EA10], but we will make it more formally.

Lemma 6.6. *Our proposed scheme is (t, ϵ, q_s)-EUF-CMA secure if the underlying digital signature is also (t, ϵ, q_s)-EUF-CMA secure.*

Proof. The adversary R against the signature underlying the construction will get the parameters of the digital signature he is trying to attack from his challenger. Then, he will choose a suitable cryptosystem. Simulation of signatures is simple; on a query m_i, R will first compute an encryption c_i of m_i, then request his challenger for a signature on c_i. Let σ_i be the answer of such a query. R will then output (c_i, σ_i) and produces a ZK proof that c_i decrypts in m_i. Such a proof, in addition to all the proofs involved in the verification/conversion queries is possible for R to give with the knowledge of the cryptosystem private key.
At some time, the adversary **A** against the construction will output a forgery (c', σ') on a message m', that has never been queried before. (σ' is by definition a digital signature on c'. The former has never been queried by R for digital signature, since otherwise m' would have been queried before. We conclude that (c', σ') is also a valid forgery on the signature scheme. □

Lemma 6.7. *The above construction is $(t, \epsilon, q_s, q_v, q_{sc})$-INV1-CMA secure if it uses a (t, ϵ', q_s)-SEUF-CMA secure digital signature and a $(t, \epsilon, q_s(q_v + q_{sc}), \epsilon(1 - \epsilon')^{q_v+q_{sc}})$-IND-CPA secure cryptosystem.*

Let A be the invisibility adversary against the construction,we construct an IND-CPA adversary R against the underlying cryptosystem as follows. R gets the parameters of the target cryptosystem from his challenger, and chooses a suitable digital signature scheme. For a confirmed Sign query on m_i, R will proceed as in the real algorithm, with the exception of maintaining a list L of records that consists of the query, its encryption, the randomness used to produce the encryption, and finally the digital signature on the encryption. R can produce digital signatures

on any encryption with the knowledge of the signature scheme private key. Moreover, he can confirm any signature he has just generated with the knowledge of the randomness used in the encryption.

For a verification query (c_i, σ_i) on m_i, R will check R (after checking of course the validity of σ on m_i), if the record R_i appears in the list, then he will issue a proof that c_i decrypts in m_i using the third component of the record. Otherwise, he will simulate a proof of the inequality of the decryption of c_i and m_i using the rewinding technique. For a conversion query, R will proceed as in a verification query with the exception of providing the non-interactive variant of the proof he would issue if the signature is valid, and the symbol $\perp$ otherwise.

This simulation differs from the real one when the queried signature (c_i, σ_i) is valid on m_i however c_i does not appear in the list (as first field of the output confirmer signatures). We distinguish two cases, either the message in question m_i has not been queried before for signature, in which case such a query would correspond to a valid existential forgery on the construction, and thus on the underling signature scheme. The queried signature is on a message that has been queried before, which corresponds to an existential forgery on the underlying signature scheme. Since the signature scheme underlying the construction is (t, ϵ, q_s) -EUF-CMA secure, this scenario does not happen with probability at least $(1 - \epsilon')^{q_v+q_{sc}}$ at some point, A produces two messages m_0, m_1. R will forward the same messages to his challenger and obtain a ciphertext C, encryption of m_b for some $b \overset{\$}{\leftarrow} \{0, 1\}$ will produce a digital signature on c and give the result in addition to c to A as a challenge confirmer signature.

It easy to see that $A's$ answer is sufficient for R to conclude. Note that after the challenge phase, A is allowed to issue confirmed **Sign**, verification and conversion queries and R can handle them as previously. Namely the probability that the adversary does not issue a verification/conversion query of the type $(c,)$ is at least $(1 - \epsilon')^{q_v+q_{sc}}$ since the signature scheme underlying the construction is (t, ϵ, q_s)-SEUF-CMA secure.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

In this thesis, we showed our results which based on the coded-based cryptography. We got there results on this topic: Firstly, we constructed a zero-knowledge identification scheme based on syndrome decoding problem over finite field Z_q. This scheme with soundness error $2/3$, which is a generalization of (binary) Stern scheme to q-ary case. Our scheme is superior to the CVA scheme [CVA10] in terms of communication cost, but only for $q = \{3, 4\}$ under 80 bits security. Secondly,we presented the first proof of plaintext knowledge for an equality relation for the Niederreiter and McEliece PKE. Our constructions are proved secure in the standard model, under hardness of the McEliece assumptions related to coding theory. Thirdly, we present a new version of the verifiable encryption which apply the PPK for the McEliece PKE to construct the verifiable IND-CPA McEliece encryption for an equality relation. We also discuss this idea could not apply to construct verifiable encryption. We also try to give the improving the computational complexity on this scheme. Lastly, we constructed the first designated confirmer signatures scheme based on McEliece PKE, as our best knowledge, it is also the first post-quantum confirmer signatures scheme. Another important open question is to extend our verifiable encryption to more general relations and to verifiable decryption. We construct the general construction about code-based undeniable signature, in addition, we also give the general construction about code-based confirm and disavowal signature. To the best of our construction, we design a practical scheme, this schemes is the first code-based undeniable signature scheme.

7.2 Future Works

The future works is to solve the open questions on our two schemes: The open question of proof of plaintext knowledge scheme based on code-based theory is that to construct the more fast verifiable encryption based on McEliece encryption, reduce the communication cost and computation cost. We also need to try to solve the gap of verifiable encryption based on Niederreiter encryption. The other important open question of proof of plaintext knowledge scheme is to upgrade our scheme to non-malleable security. According to [Kat03], this will allow us to construct password-based authentication and key exchange, as well as deniable authentication based on coding.

The open question of identification scheme based on code-based theory is to reduce the soundness error to exactly $1/2$ in the case of small q, most importantly for $q = 2$, since a scheme working over the binary field is expected to have a fast implementation. Reducing the size of the public matrix is another natural open question.

Bibliography

[AR01] Y Aumann and MO Rabin. A proof of plaintext knowledge protocol and applications. *Manuscript. June*, 2001.

[ASW98] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures (extended abstract). In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606. Springer, 1998.

[Bar98] Alexander Barg. Complexity issues in coding theory, 1998.

[BBD09] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-quantum cryptography*. Springer Science & Business Media, 2009.

[BD10] Rikke Bendlin and Ivan Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *Theory of Cryptography*, pages 201–218. Springer, 2010.

[Ber10] Daniel J Bernstein. Grover vs. mceliece. In *Post-Quantum Cryptography*, pages 73–80. Springer, 2010.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM, 1988.

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptolog–CRYPTO 1992*, pages 390–420. Springer, 1993.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1+ 1= 0 improves information set decoding. In *Advances in Cryptology–EUROCRYPT 2012*, pages 520–536. Springer, 2012.

[BLP08] Daniel J Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the mceliece cryptosystem. In *Post-Quantum Cryptography*, pages 31–46. Springer, 2008.

[BLP10] Daniel J Bernstein, Tanja Lange, and Christiane Peters. Wild mceliece. In *Selected Areas in Cryptography*, pages 143–158. Springer, 2010.

[BLP11] Daniel J Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: ball-collision decoding. In *Advances in Cryptology–CRYPTO 2011*, pages 743–760. Springer, 2011.

[BMVT78] Elwyn R Berlekamp, Robert J McEliece, and Henk CA Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[CCC98] Florent Chabaud, Anne Canteaut, and Et Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to primitive narrow-sense bch codes of length 511. 1998.

[CD00] Jan Camenisch and Ivan Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Advances in Cryptology–ASIACRYPT 2000*, pages 331–345. Springer, 2000.

[CFS01] Nicolas T Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *Advances in Cryptology–CRYPTO 2001*, pages 157–174. Springer, 2001.

[Cha94] David Chaum. Designated confirmer signatures. In *Advances in Cryptology–CRYPTOEUROCRYPT 1994*, pages 86–91. Springer, 1994.

[CM00] Jan Camenisch and Markus Michels. Confirmer signature schemes secure against adaptive adversaries. In *Advances in Cryptology–EUROCRYPT 2000*, pages 243–258. Springer, 2000.

[CM10] Pierre-Louis Cayrel and Mohammed Meziani. Post-quantum cryptography: code-based signatures. In *Advances in Computer Science and Information Technology*, pages 82–99. Springer, 2010.

[CMO98] Henri Cohen, Atsuko Miyaji, and Takatoshi Ono. Efficient elliptic curve exponentiation using mixed coordinates. In *Advances in Cryptology–ASIACRYPT 1998*, pages 51–65. Springer, 1998.

[Cov73] Thomas M Cover. Enumerative source encoding. *Information Theory, IEEE Transactions on*, 19(1):73–77, 1973.

[CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Advances in Cryptology–CRYPTO 2003*, pages 126–144. Springer, 2003.

[CVA90] David Chaum and Hans Van Antwerpen. Undeniable signatures. In *Advances in Cryptology–CRYPTO 1989 Proceedings*, pages 212–216. Springer, 1990.

[CVA10] Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In *Selected Areas in Cryptography*, pages 171–186. Springer, 2010.

[CvHP91] David Chaum, Eugéne van Heijst, and Birgit Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *Advances in Cryptology–CRYPTO 91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 1991.

[Dam99] Ivan Damgard. *Lectures on data security: modern cryptology in theory and practice*, volume 1561. Springer Science & Business Media, 1999.

[DDN03] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM review*, 45(4):727–784, 2003.

[DGOW95] Ivan Damgård, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. Honest verifier vs dishonest verifier in public coin zero-knowledge proofs. In *Advances in Cryptology–CRYPTO 1995*, pages 325–338. Springer, 1995.

[DH76] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[DMR11a] Hang Dinh, Cristopher Moore, and Alexander Russell. Mceliece and niederreiter cryptosystems that resist quantum fourier sampling attacks. In *Advances in Cryptology–Crypto 2011*, pages 761–779. Springer, 2011.

[DMR11b] Hang Dinh, Cristopher Moore, and Alexander Russell. Quantum fourier sampling, code equivalence, and the quantum security of the mceliece and sidelnikov cryptosystems. *arXiv preprint arXiv:1111.4382*, 2011.

[DvdGMQN08] Rafael Dowsley, Jeroen van de Graaf, Jörn Müller-Quade, and Anderson CA Nascimento. Oblivious transfer based on the mceliece assumptions. In *Information Theoretic Security*, pages 107–117. Springer, 2008.

[EA08] Laila El Aimani. Toward a generic construction of universally convertible undeniable signatures from pairing-based signatures. In *Progress in Cryptology-INDOCRYPT 2008*, pages 145–157. Springer, 2008.

[EA10] Laila El Aimani. Efficient confirmer signatures from the "signature of a commitment" paradigm. In *Provable Security*, pages 87–101. Springer, 2010.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in cryptology*, pages 10–18. Springer, 1985.

[EOS07] Daniela Engelbert, Raphael Overbeck, and Arthur Schmidt. A summary of mceliece-type cryptosystems and their security. *J. Mathematical Cryptology*, 1(2):151–199, 2007.

[FGUO+13] J-C Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and J-P Tillich. A distinguisher for high-rate mceliece cryptosystems. *Information Theory, IEEE Transactions on*, 59(10):6830–6844, 2013.

[Fin11a] Matthieu Finiasz. Parallel-cfs. In *Selected areas in cryptography*, pages 159–170. Springer, 2011.

[Fin11b] Matthieu Finiasz. Parallel-cfs. In *Selected areas in cryptography*, pages 159–170. Springer, 2011.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology–CRYPTO 1986*, pages 186–194. Springer, 1987.

[FS90] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In *Advances in Cryptology–CRYPTO 1989 Proceedings*, pages 526–544. Springer, 1990.

[GG07a] Philippe Gaborit and Marc Girault. Lightweight code-based identification and signature. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 191–195. IEEE, 2007.

[GG07b] Philippe Gaborit and Marc Girault. Lightweight code-based identification and signature. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 191–195. IEEE, 2007.

[GHY86] Zvi Galil, Stuart Haber, and Moti Yung. Symmetric public-key encryption. In *Advances in Cryptology–CRYPTO 1985 Proceedings*, pages 128–137. Springer, 1986.

[Gir90] Marc Girault. A (non-practical) three-pass identification protocol using coding theory. In *Advances in Cryptology–AUSCRYPT 1990*, pages 265–272. Springer, 1990.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for np. *Journal of Cryptology*, 9(3):167–189, 1996.

[GK05] Shafi Goldwasser and Dmitriy Kharchenko. Proof of plaintext knowledge for the ajtai-dwork cryptosystem. In *Theory of Cryptography*, pages 529–555. Springer, 2005.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GM03] Steven D Galbraith and Wenbo Mao. Invisibility and anonymity of undeniable and confirmer signatures. In *Topics in Cryptology–CT–RSA 2003*, pages 80–97. Springer, 2003.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 291–304. ACM, 1985.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[GMR05] Craig Gentry, David Molnar, and Zulfikar Ramzan. Efficient designated confirmer signatures without random oracles or general zero-knowledge proofs. In *Advances in Cryptology–ASIACRYPT 2005*, pages 662–681. Springer, 2005.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[Gol01] Oded Goldreich. *Foundations of cryptography: volume I, basic applications*. Cambridge university press, 2001.

[Gol04] Oded Goldreich. *Foundations of cryptography: volume II, basic applications*. Cambridge university press, 2004.

[Gol08] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[Gop70] Valerii Denisovich Goppa. A new class of linear correcting codes. *Problemy Peredachi Informatsii*, 6(3):24–30, 1970.

[GS12] Philippe Gaborit and Julien Schrek. Efficient code-based one-time signature from automorphism groups with syndrome compatibility. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 1982–1986. IEEE, 2012.

[GW04] Shafi Goldwasser and Erez Waisbard. Transformation of digital signature schemes into designated confirmer signature schemes. In *Theory of Cryptography*, pages 77–100. Springer, 2004.

[Har88] Sami Harari. A new authentication algorithm. In *Coding Theory and Applications*, pages 91–105. Springer, 1988.

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Advances in Cryptology–CRYPTO 1996*, pages 201–215. Springer, 1996.

[HMT13] Rong Hu, Kirill Morozov, and Tsuyoshi Takagi. Proof of plaintext knowledge for code-based public-key encryption revisited. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 535–540. ACM, 2013.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *Advances in Cryptology–ASIACRYPT 2012*, pages 663–680. Springer, 2012.

[Kat03] Jonathan Katz. Efficient and non-malleable proofs of plaintext knowledge and applications. In *Advances in Cryptology–EUROCRYPT 2003*, pages 211–228. Springer, 2003.

[Kat10] Jonathan Katz. *Digital Signatures*. Springer Science & Business Media, 2010.

[KKS97] Gregory Kabatianskii, E Krouk, and Ben Smeets. A digital signature scheme based on random error-correcting codes. In *Crytography and Coding*, pages 161–167. Springer, 1997.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC Press, 2014.

[KMO08] Kazukuni Kobara, Kirill Morozov, and Raphael Overbeck. Coding-based oblivious transfer. In *Mathematical Methods in Computer Science*, pages 142–156. Springer, 2008.

[Kob94] Neal Koblitz. *A course in number theory and cryptography*, volume 114. Springer Science & Business Media, 1994.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Advances in Cryptology–ASIACRYPT 2008*, pages 372–389. Springer, 2008.

[KW92] Kaoru Kurosawa and Osamu Watanabe. Computational and statistical indistinguishabilities. In *Algorithms and Computation*, pages 430–438. Springer, 1992.

[LB88] Pil Joong Lee and Ernest F Brickell. An observation on the security of mceliece's public-key cryptosystem. In *Advances in Cryptology–EUROCRYPT 1988*, pages 275–280. Springer, 1988.

[LDW94] Yuan Xing Li, Robert H Deng, and Xin Mei Wang. On the equivalence of mceliece's and niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.

[Leo88] Jeffrey Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *Information Theory, IEEE Transactions on*, 34(5):1354–1359, 1988.

[Mat80] HF Mattson, Jr. The theory of error-correcting codes (fj macwilliams and nja sloane). *SIAM Review*, 22(4):513–519, 1980.

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978.

[Mil86] Victor Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology–CRYPTO 1985 Proceedings*, pages 417–426. Springer, 1986.

[Min67] Marvin L Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

[MM12] Behzad Malek and Ali Miri. Securing harari's authentication scheme. *IJ Network Security*, 14(4):206–210, 2012.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error correcting codes*. Elsevier, 1977.

[MS98] Markus Michels and Markus Stadler. Generic constructions for secure and efficient confirmer signature schemes. In *Advances in Cryptology–EUROCRYPT 1998*, pages 406–421. Springer, 1998.

[MT12] Kirill Morozov and Tsuyoshi Takagi. Zero-knowledge protocols for the mceliece encryption. In *Information Security and Privacy*, pages 180–193. Springer, 2012.

[MVVR12] K Preetha Mathew, Sachin Vasant, Sridhar Venkatesan, and C Pandu Rangan. An efficient ind-cca2 secure variant of the niederreiter encryption scheme in the standard model. In *Information Security and Privacy*, pages 166–179. Springer, 2012.

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In *Advances in Cryptology–CRYPTO 1889 Proceedings*, pages 128–136. Springer, 1990.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *PROBLEMS OF CONTROL AND INFORMATION THEORY-PROBLEMY UPRAVLENIYA I TEORII INFORMATSII*, 15(2):159–166, 1986.

[NIKM08] Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Designs, Codes and Cryptography*, 49(1-3):289–305, 2008.

[Oka94] Tatsuaki Okamoto. Designated confirmer signatures and public-key encryption are equivalent. In *Advances in Cryptology–CRYPTO 1994*, pages 61–74. Springer, 1994.

[OS09] Raphael Overbeck and Nicolas Sendrier. Code-based cryptography. In *Post-quantum cryptography*, pages 95–145. Springer, 2009.

[Pet] C Peters. Iteration and operation count for information-set decoding over fq. jan 2010.

[Pet10] Christiane Peters. Information-set decoding for linear codes over f q. In *Post-Quantum Cryptography*, pages 81–94. Springer, 2010.

[Pie12] Krzysztof Pietrzak. Cryptography from learning parity with noise. In *SOFSEM 2012: Theory and Practice of Computer Science*, pages 99–114. Springer, 2012.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

[Rot06] Ron Roth. *Introduction to coding theory*. Cambridge University Press, 2006.

[RS92] Charles Rackoff and Daniel R Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology–CRYPTO 1991*, pages 433–444. Springer, 1992.

[RSA78] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sch] B Schoenmakers. Lecture notes cryptographic protocols, version 1.0.

[SCKI10] Tomohiro Sekino, Yang Cui, Kazukuni Kobara, and Hideki Imai. Privacy enhanced rfid using quasi-dyadic fix domain shrinking. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[Sen02] Nicolas Sendrier. On the security of the mceliece public-key cryptosystem. In *Information, coding and mathematics*, pages 141–163. Springer, 2002.

[Sen05] N. Sendrier. Encoding information into constant weight words. In *IEEE Conference, ISIT 2005*, pages 435–438, Adelaide, Australia, 2005.

[Sen09] Nicolas Sendrier. On the use of structured codes in code based cryptography. *Coding Theory and Cryptography III, The Royal Flemish Academy of Belgium for Science and the Arts*, pages 59–68, 2009.

[Sha57] Claude E Shannon. A universal turing machine with two internal states. *Automata studies*, 34:157–165, 1957.

[Sho94] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.

[SM00] Kouichi Sakurai and Shingo Miyazaki. An anonymous electronic bidding protocol based on a new convertible group signature scheme. In *Information Security and Privacy*, pages 385–399. Springer, 2000.

[SMEYA11] Mohammed Meziani Sidi Mohamed El Yousfi Alaoui, Pierre-Louis Cayrel. Improved identity-based identification and signature schemes using quasi-dyadic goppa codes. In *Information Security and Assurance*, pages 146–155. Springer, 2011.

[Sta96] Markus Stadler. Publicly verifiable secret sharing. In *Advances in Cryptology–EUROCRYPT 1996*, pages 190–199. Springer, 1996.

[Ste89] Jacques Stern. A method for finding codewords of small weight. In *Coding theory and applications*, pages 106–113. Springer, 1989.

[Ste96] Jacques Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6):1757–1768, 1996.

[Sti05] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[Vér95] Pascal Véron. Cryptanalysis of harari's identification scheme. In *Cryptography and Coding*, pages 264–269. Springer, 1995.

[Vér97] Pascal Véron. Improved identification schemes based on error-correcting codes. *Applicable Algebra in Engineering, Communication and Computing*, 8(1):57–69, 1997.

[WBWB07] Guilin Wang, Joonsang Baek, Duncan S Wong, and Feng Bao. On the generic and efficient constructions of secure designated confirmer signatures. In *Public Key Cryptography–PKC 2007*, pages 43–60. Springer, 2007.

[Wik07] Douglas Wikström. Designated confirmer signatures revisited. In *Theory of Cryptography*, pages 342–361. Springer, 2007.

[Xag10] Keita Xagawa. Cryptography with lattices. 2010.

[XKT07] Keita Xagawa, Akinori Kawachi, and Keisuke Tanaka. Proof of plaintext knowledge for the regev cryptosystems. 2007.

[XT09] Keita Xagawa and Keisuke Tanaka. Zero-knowledge protocols for ntru: Application to identification and proof of plaintext knowledge. In *Provable Security*, pages 198–213. Springer, 2009.

Publications

This is the list of our publications, all of them are peer-reviewed.

[1] Rong Hu, Kirill Morozov and Tsuyoshi Takagi. Proof of plaintext knowledge for code-based public-key encryption revisited. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security 2013, pages 535-540. ACM, 2013.

[2] Rong Hu, Kirill Morozov and Tsuyoshi Takagi. On Zero-Knowledge Identification Based on Q-ary Syndrome Decoding. In The 8th Asia Joint Conference on Information Security 2013, pages 12-18. IEEE, 2013.

[3] Rong Hu, Kirill Morozov and Tsuyoshi Takagi. Zero-Knowledge Protocols for Code-Based Public-Key Encryption. IEICE Transactions on Fundamentals on Fundamentals of Electronics, Communications and Computer Sciences. Conditionally Accepted.

[4]Youwen Zhu, Tsuyoshi Takagi and Rong Hu. Security Analysis of Collusion-Resistant Nearest Neighbor Query Scheme on Encrypted Cloud Data. IEICE TRANSACTIONS on Information and Systems, 97-D(2): pages 326-330. The Institute of Electronics, Information and Communication Engineers, 2014.