POST QUANTUM SIGNATURE SCHEMES FOR THE BLOCKCHAIN TECHNOLOGY

By

WU ZI FENG

A project report submitted in partial fulfilment of the
requirements for the award of Master of Mathematics

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2019

DECLARATION OF ORIGINALITY

I hereby declare that this project report entitled "**POST QUANTUM SIGNATURE SCHEMES FOR THE BLOCKCHAIN TECHNOLOGY**" is my own work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : _____

ID No. : _____

Date : _____

APPROVAL FOR SUBMISSION

I certify that this project report entitled "**POST QUANTUM SIGNATURE SCHEMES FOR THE BLOCKCHAIN TECHNOLOGY**" was prepared by **WU ZI FENG** has met the required standard for submission in partial fulfilment of the requirements for the award of Master of Mathematics at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : _____

Date : _____

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of University Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2019, WU ZI FENG. All rights reserved.

ACKNOWLEDGEMENTS

I have successfully completed this research project with the assistance from various authorities and I would like to grab this opportunity to convey my acknowledgement to all the people that assisted me during this period.

First of all, I would like to express my special thanks of gratitude to my supervisor Dr. Denis Wong Chee Keong, who provided insight and expertise that greatly assisted this research. Besides that, I am extremely thankful and indebted to him for his sharing of expertise, guidance and encouragement extended to me.

Next, I am also thankful towards my friends and family for the unceasing encouragement, support and attention throughout the progress of this journey. I sincerely thank all the people for their lend of hands both directly and indirectly contributing to my research project.

WU ZI FENG

POST QUANTUM SIGNATURE SCHEMES FOR THE BLOCKCHAIN TECHNOLOGY

WU ZI FENG

ABSTRACT

Blockchain technology is a booming topic since the invention of cryptocurrency which relies on the security of signature schemes. However a quantum computer is believed to be able to break all security models that are number theory based. This may cause the blockchain technology to lose its security and cryptocurrencies to lose all values if quantum computers were to be released to the public.

The purpose of this study is to construct a suitable signature scheme for the blockchain technology that is quantum resistant, since conventional security models were proven to be easily broken by using quantum algorithm. First we study the different properties and uses of signature schemes, and how Elliptic Curve Digital Signature Algorithm is used in the blockchain technology. Then we review previous researches on lattice-based signature schemes to construct our own scheme which is suitable for the blockchain technology. Finally we show that our scheme is possibly usable for the security in quantum computers.

TABLE OF CONTENTS

TITLE i

DECLARATION OF ORIGINALITY ii

ACKNOWLEDGEMENTS vii

ABSTRACT viii

LIST OF FIGURES xi

LIST OF TABLES xii

CHAPTER 1 Introduction 1

1-1 Objectives 1

1-2 Problem Statements 2

1-3 Methodology and Planning 2

1-4 Project Plan 3

1-5 Research Flow Chart 4

1-6 Expected Outcome 4

CHAPTER 2 Signature Schemes 5

2-1 Introduction 5

2-2 Integer Factorization Problem (IFP) 6

2-3 Discrete Logarithm Problem (DLP) 9

CHAPTER 3 The Blockchain Technology 17

3-1 Game Theory 20

3-2 Computer Science 22

3-3 Cryptography 23

3-4 Applications of the Blockchain Technology 24

3-4-1 Banking Sector 25

3-4-2 Insurance Sector 26

3-4-3 Media Industry 27

3-4-4 Healthcare Sector 27

3-4-5 Government Sector 27

3-4-6 Supply Chain Sector 28

3-5 Islamic Applications 28
3-6 Other Applications 28

CHAPTER 4 Post Quantum Signature Schemes 30

4-1 Quantum Computing 30
4-2 Lattice-based Cryptography 30
4-3 Hardness Problems 31
4-3-1 Shortest Vector Problem (SVP) 31
4-3-2 Closest Vector Problem (CVP) 32
4-3-3 Short Integer Solution (SIS) 32
4-3-4 Learning With Errors (LWE) 32
4-4 Lattice Reduction 33
4-5 Semisimple Cyclic and Abelian Codes 34

CHAPTER 5 Introduction to Lattice Based Schemes 38

5-1 Ajtai-Dwork Public Key Cryptosystem 38
5-2 NTRU Encryption Scheme 39
5-3 The GGH Encryption Scheme 41
5-4 NTRU Signature Scheme (NSS) 43
5-5 Dihedral Group Algebra 44
5-6 Proposed Scheme 47

CHAPTER 6 Conclusion 50

LIST OF FIGURES

1.1 Project planning throughout 21 weeks 3
1.2 The research flow chart for this research project 4

3.1 Simple illustration of the structure of the blockchain 17
3.2 Block structure and header structure of Bitcoin blockchain 20
3.3 Illustration of the longest-chain-criterion 21
3.4 Illustration of the heaviest-chain-criterion 22
3.5 Applications of the blockchain technology 26

LIST OF TABLES

2.1 Comparisim between Signature Schemes 16

3.1 Descriptions of the block structure and header structure from Figure 3.2 18

3.2 A brief history of blockchain (Emphasizing on Bitcoin & Ethereum) (Gichigi, 2018) 19

3.3 How different layers can affect the blockchain system 25

5.1 Comparisim between Lattice-Based Schemes 44

CHAPTER 1: INTRODUCTION

In this project, various signature schemes based on the hard problems in lattice, originated and constructed by different researchers over the years are studied. By comparing different schemes, we can understand the balance between the security and also the efficiency of the signature scheme for use in different applications. It is because there are several uses of signature schemes such as verification process with smaller devices (smart cards) or large networks (blockchain).

One of the well-known system that use blockchain technology is cryptocurrency Bitcoin, proposed Satoshi Nakamoto (Nakamoto, 2008). Since then, the community have been exploring the potential of blockchain technology through various use cases such as security, insurance, medical, or even supply chain sectors. However, engineers are developing a new type of computer which operates in quantum bits (qubits) called quantum computers, which is proven to be able to break commonly used encryption schemes like RSA and Elliptic Curve Digital Signature Algorithm (ECDSA) by Peter Shor (Shor, 1994).

Currently there are several potential hard mathematical problems that have not been solved on a quantum computer. For example, code-based and lattice-based cryptosystems (Lauter, 2017). In this project, we will be focusing on three different directions of studies. First, we will perform a thorough review on the blockchain technology. Then, a detailed study for different types of lattice-based signature schemes will be conducted. Finally, we will be constructing a lattice-based signature scheme which is suitable for the blockchain technology.

1-1 Objectives

1. To understand the usage of Elliptic Curve Digital Signature Algorithm in the blockchain technology.

2. To construct a post quantum lattice-based signature scheme which is suitable for the blockchain technology applications.

1-2 Problem Statements

1. What is the underlying hardness problem used in Elliptic Curve Digital Signature Algorithm and the efficiency of the scheme?

2. What are the hardness problems in lattice which is suitable for constructing a Post Quantum lattice-based signature scheme?

1-3 Methodology and Planning

1. Perform a thorough literature review by emphasizing on:

   (a) Applications of ECDSA in blockchain.

   (b) Lattice-based hardness problems that able to resist quantum attack.

   (c) Existing post quantum signature schemes.

   (d) Standard Model of post quantum blockchain signature schemes.

2. Perform a thorough study on existing lattice-based signature scheme and modify to suit blockchain applications by:

   (a) Reducing the size of signature to increase the efficiency of signing process.

   (b) Modify the verification phase to preserve the correctness of proposed scheme.

1-4 Project Plan

In this section, we stated the plan throughout project 1 and project 2 to accomplish this project.

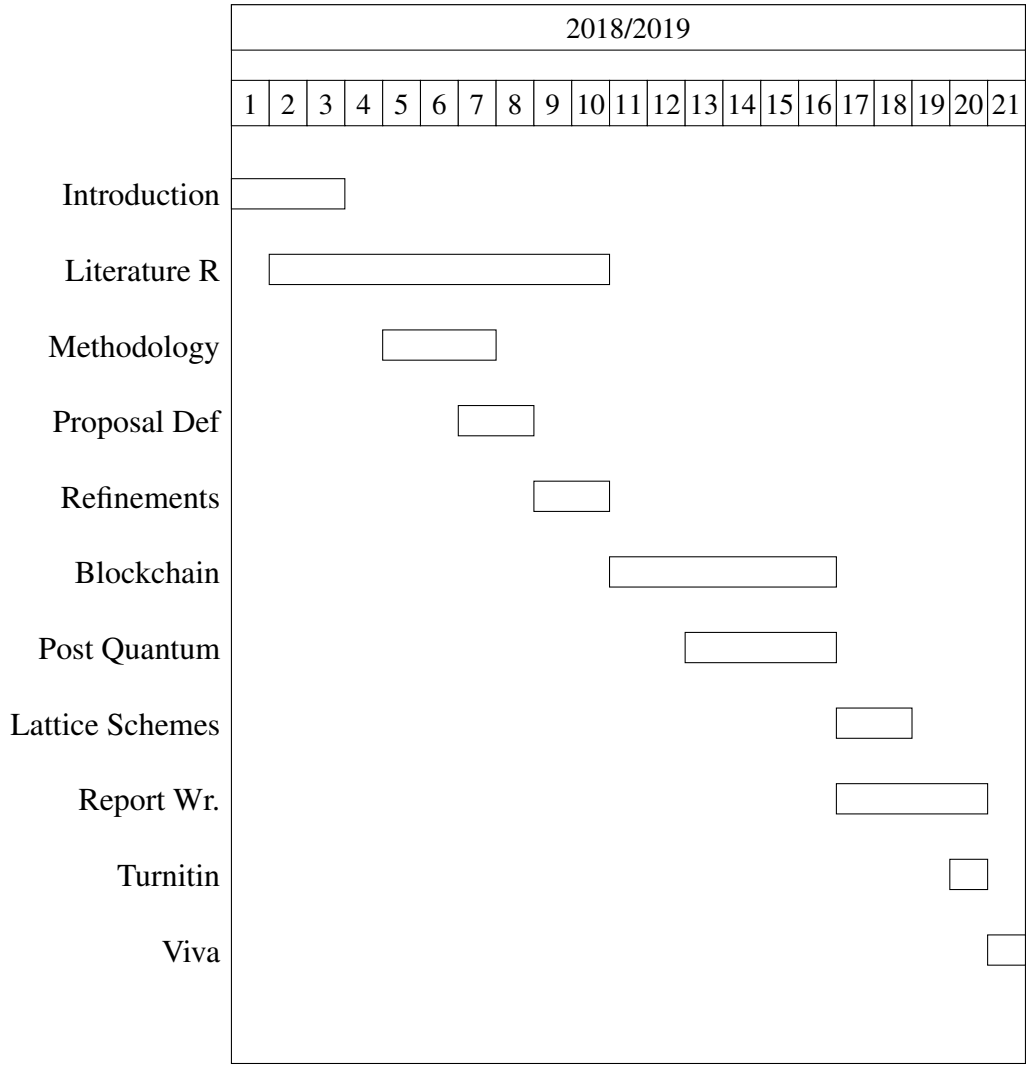

Figure 1.1: Project planning throughout 21 weeks

1-5 Research Flow Chart

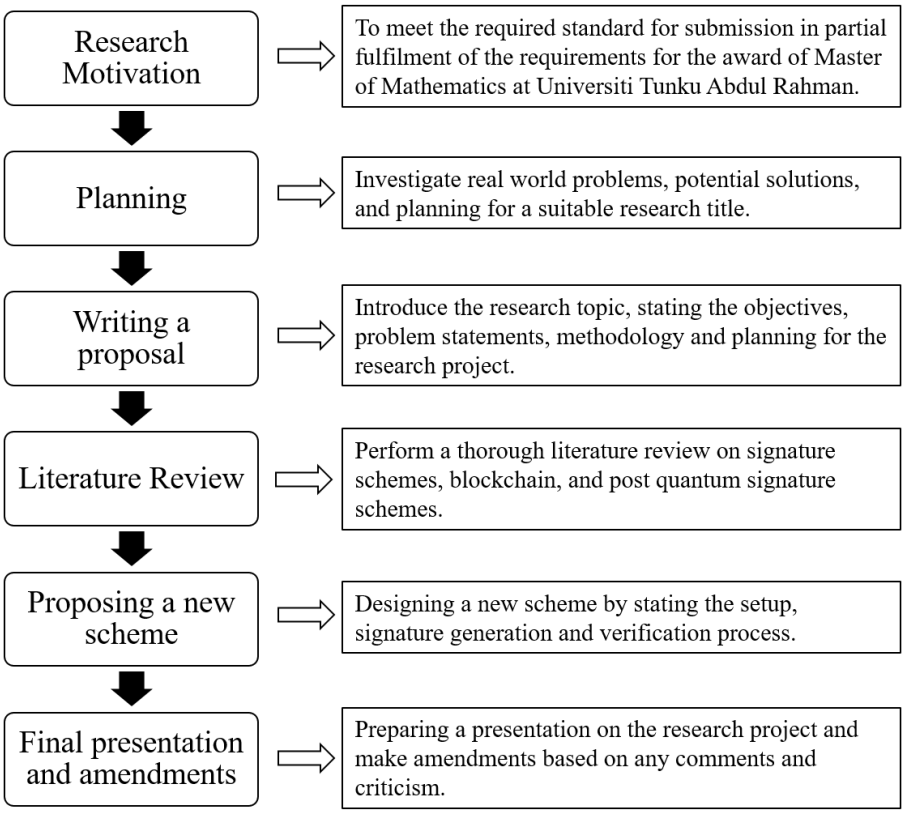

Figure 1.2: The research flow chart for this research project

1-6 Expected Outcome

The proposed signature scheme can be used in the blockchain technology, to verify new transactions or information in forming a new block. This lattice-based signature scheme also can resist attack from quantum computers, which number theory based schemes could not. Other than that, this research project can act as a survey paper for the other future researchers who are interested in the blockchain technology or lattice-based signature schemes.

CHAPTER 2: SIGNATURE SCHEMES

In this chapter, we will be discussing about a brief history on signature schemes and several designs of schemes based on various applications. More detailed explanations and examples can also be found on undergraduate cryptography textbooks (Stinson, 2006, Mollin, 2003). This chapter consists of introduction, and two popular hardness problems: integer factorization problem and discrete logarithm problem.

2-1 Introduction

A unique hand signed signature is often used in letters and contracts as a proof of ownership or agreement. Signature schemes, or digital signatures, are used to sign a message stored in binary computers, which is then transmitted through the computer network (Stinson, 2006). The main weakness of a conventional signature comes from signature forgeries, which can be difficult to verify through the naked eye, while a digital signature that sign using a secret key can easily be identified with a publicly known verification algorithm and the corresponding public key.

The concept of secret and public keys was first proposed by Diffie and Hellman (Diffie and Hellman, 1976), which is also known as asymmetric cryptography this day. Diffie and Hellman mentioned two methods in transfering messages over insecure channels while having no information leaked to any third parties, which are public key cryptosystem and public key distribution system. In public key cryptosystem, the encryption and decryption algorithms are constructed using two distinct keys, public key, K_{pk} and secret key, K_{sk}, such that computing of K_{sk} from K_{pk} is computationally infeasible. Because of this, the public key can be used by anyone to encrypt a message that only can be deciphered by intended receiver through an insecure channel. On the other hand, public key distribution system requires two parties to exchange partial key information back and forth and compute a common secret key, such that eavesdroppers must not be able to compute their common secret key from the insecure channel.

Unlike a conventional encryption scheme which involves encrypting and decrypting a message, a signature scheme consists of a signing algorithm $\mathbf{sig}_{K_{sk}}$ and a ver-

ification algorithm $\mathbf{ver}_{K_{pk}}$. For a message, m, and signed message $s = \mathbf{sig}_{K_{sk}}(m)$, verification process $\mathbf{ver}_{K_{pk}}(m, s)$ will decide on whether s is a authentic signature by the signer for m and returns answer true or false. Every signature schemes are constructed by assuming that there is a hard problem, which is known to be computationally difficult to solve by a computer in polynomial time. The most common hardness problems found in signature schemes are Integer Factorization Problem (IFP) and Discrete Logarithm Problem (DLP).

2-2 Integer Factorization Problem (IFP)

Definition 1. Given a positive integer $n \in \mathbb{N}$, find its prime factorization $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ where the p_i are pairwise distinct primes and $e_i > 0$.

One of the earliest and well-known signature scheme is RSA, constructed by Rivest, Shamir and Adleman (Rivest et al., 1978). RSA uses the hardness of IFP in key generation in the following signature scheme:

RSA Signature Scheme

Key Generation

Choose p and q be two distinct large primes.

Compute RSA modulus, $n = p \cdot q$ and $\phi(n) = (p-1)(q-1)$.

Compute RSA enciphering exponent, $ed \equiv 1 \pmod{\phi(n)}$,

where d can be any prime number greater than $max(p, q)$.

Signing

Let m be the message and s be the signature,

$$\mathbf{sig}_k(m) \equiv m^d \pmod{n} \equiv s,$$

Verification

$$\mathbf{ver}_k(m, s) = \text{true iff } m \equiv c^e \pmod{n}.$$

Thus, the generated public key of RSA is (n, e) and the secret key is (d, p, q), also known as RSA deciphering exponent (Mollin, 2003). Factoring n would enable an

enemy cryptanalyst to break the signature scheme, and the fastest factoring algorithm by Richard Schroeppel (unpublished) can factor n in approximately:

$$exp\sqrt{ln(n) \cdot ln(ln(n))} = n^{\sqrt{ln(ln(n))/ln(n)}} = (ln(n))^{\sqrt{ln(n)/ln(ln(n))}}$$

It was recommended that the RSA modulus n be about 200 digits long, in order to provide a margin of safety against future developments, such as a faster computer. Some other signature schemes that use IFP are Rabin, Fiat-Shamir and Goldwasser-Micali-Rivest.

In Rabin's research paper (Rabin, 1979), he claimed that his scheme has a much faster signature verification compared to RSA. It is because breaking the RSA function is at most as hard as factorization but is not known to be equivalent to factorization. The hardness problem of Rabin's scheme relates directly to IFP, and the scheme is existentially unforgeable in the random oracle model. Similar to RSA in computing $n = pq$, the signature verification requires the hashing of message with random suffix word u, $H(mu)$ by hashing mu to a fixed length binary string, $(0, 1)^k$.

Rabin Signature Scheme

Key Generation

Compute $n = p \cdot q$, where $p, q \equiv 3 \pmod 4$.

Choose $b \in \{1, 2, \ldots, n\}$. Public key is (n, b), secret key is (p, q).

Signing

Choose padding u to hash message, m, compute $H(mu)$ square modulo n.

Compute x such that $x(x + b) \equiv H(mu) \pmod n$.

Signature pair is (u, x).

Verification

The signature (u, x) is valid on m if and only if

$$x(x + b) \pmod n \equiv H(mu).$$

Fiat and Shamir (Fiat and Shamir, 1987) constructed a scheme based on IFP with the intention to reduce the modular multiplications as per RSA scheme, so that it is

ideally suited for microprocessor-based devices such as smart cards and remote control systems. The scheme is based on the difficulty of extracting modular square roots when the factorization of n is unknown. Lastly, the scheme has been proven to be secure against chosen message attack.

Fiat Shamir Signature Scheme

Key Generation

Let $n = pq$, a pseudorandom function f that maps arbitrary strings to a range from 0 to n, user string I and s_j from smart card, and random matrix e_{ij}.

Signing

Randomly choose $r_1, r_2, \ldots, r_t \in [0, n)$, compute $x_i = r_i^2 \pmod{n}$.

Compute $f(m, x_1, \ldots, x_t)$ and uses the first kt bits as e_{ij} values $(1 \leq i \leq t, 1 \leq j \leq k)$.

Compute $y_i = r_i \prod_{r_{ij}=1} s_j \pmod{n}$ for $i = 1, 2, \ldots, t$.

Verification

Compute $v_j = f(I, j)$ for $j = 1, 2, \ldots, k$.

Compute $z_i = y_i^2 \prod_{e_{ij}=1} v_j \pmod{n}$ for $i = 1, 2, \ldots, t$.

Verifies that the first kt bits of $f(m, z_1, \ldots, z_t)$ are e_{ij}.

On the other hand, Goldwasser-Micali-Rivest have constructed a signature scheme, and proven that it is secure against adaptive chosen message attack (Goldwasser et al., 1998). It is the first research paper that mention different kinds of attacks faced by previous signature schemes. A generic chosen message attack is defined as how an attacker is allowed to choose a fixed list of messages $m_1, m_2, \cdots, m_i$ and corresponding signatures $s_1, s_2, \cdots, s_i$. An adaptive chosen message attack is more severe, such that an attacker may request additional signatures of messages which depend additionally on previous signatures, $s_{i+1}, s_{i+2}, \cdots, s_{i+k}$. Both of these attacks are mentioned as an example of information leak in real world situation, and to prove the security of the scheme such that any attackers could not forge a new signature based on messages of their choice.

GMR Signature Scheme

Key Generation

Let G be a claw-free permutation, input 1^k.

Randomly choose two quintuples

$$(d_f, f_0, f_0^{-1}, f_1, f_1^{-1}) \in [G(1^k)]$$

$$(d_g, g_0, g_0^{-1}, g_1, g_1^{-1}) \in [G(1^k)]$$

Randomly choose $r_\varepsilon^f \in D_f = [d_f(\ )]$

We get $f = (d_f, f_0, f_1)$ and $g = (d_g, g_0, g_1)$

Output Public Key is $(f, r_\varepsilon^f, g, 2^b)$ and Secret Key is (f^{-1}, g^{-1}).

Signing

To sign message m_i, choose $(f - 1^b)$-tree T with 2^b leaves,

Compute g-item G_i with root $r_i^g \in D_g$ which will be the ith leaf of T.

Output signature G_i, with beginning tree root r_ε^f, and ending tree leaf r_i^g.

Verification

To verify that the signature is valid for the message m_i for the first $b+1$ elements,

- Check that the f-chain starts at r_ε^f (tree root) and ends at r_i^g tree leaf.
- Check that the signature G_i has r_i^g as its tree root and m_i as its only child.

If both conditions are true, the signature is valid.

2-3 Discrete Logarithm Problem (DLP)

Definition 2. Suppose p is a prime number, choose g to be a primitive root mod p. Then, for any integer $A \in \{1, 2, \ldots, p-1\}$, there exist an exponent $a \in \{0, 1, 2, \ldots, p-2\}$ that satisfies the following congruence:

$$A \equiv g^a \pmod{p}$$

ElGamal uses the hardness of DLP in the construction of his signature scheme (Elgamal, 1985). The scheme is non-deterministic, whereby there are many valid signatures for any given message, and the verification algorithm must be able to accept

any of these valid signatures as authentic. This prevents probable text attack where the attacker tries to find out the message by attempting to produce a matching signature.

ElGamal Signature Scheme

Key Generation

Choose a prime p such that DLP in $\mathbb{Z}_p$ is intractable.

Choose $\alpha \in {\mathbb{Z}_p}^*$ be a primitive element.

Let $\mathcal{P} = {\mathbb{Z}_p}^*$, and $\mathcal{A} = {\mathbb{Z}_p}^* \times \mathbb{Z}_{p-1}$, define

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}.$$

Output Public Key (p, α, β), and Secret Key (a).

Signing

To sign a messge m, choose a secret random number $k \in {\mathbb{Z}_{p-1}}^*$, define

$$\mathbf{sig}_K(m, k) = (\gamma, \delta)$$

$$\gamma = \alpha^k \pmod{p},\ \delta = (m - a\gamma)k^{-1} \pmod{p-1}$$

Verification

$$\mathbf{ver}_K(m, (\gamma, \delta)) = \text{true iff } \beta^\gamma \gamma^\delta \equiv \alpha^m \pmod{p}$$

Schnorr signature scheme (Schnorr, 1991), is the improved version of ElGamal signature scheme in terms of generation speed, verification speed, and bit length of signature. The idea is to use an integer $\alpha \in \mathbb{Z}_p$ such that the order of α is a sufficiently large prime q. This scheme was designed for the limited computational power of smart cards to generate a signature, thus only requiring the multiplication of a 72-bit integer with a 140-bit integer.

Schnorr Signature Scheme

Key Generation

Choose two distinct primes p and q such that $q|(p-1)$, where p and q are chosen to be $q \geq 2^{140}$ and $p \geq 2^{512}$.

Choose $\alpha \in \mathbb{Z}_p^q$ and $\alpha \neq 1$.

Let the hash function be, $H : \mathbb{Z}_q \times \mathbb{Z} \to \{0, 1, \ldots, 2^t - 1\}$,

where $2^t = exp\sqrt{ln(p)lnln(p)}$.

Choose a random secret key, $s \in \{1, 2, \ldots, q\}$ and compute public key, $v = \alpha^{-s} \pmod{p}$.

Signing

Choose random number $r \in \{1, 2, \ldots, q\}$, and compute $x := \alpha^r \pmod{p}$.

Using the hash function on message m to get $e := H(x, m) \in \{0, 1, \ldots, 2^t - 1\}$.

Compute $y := r + se \pmod{q}$, output signature (e, y).

Verification

Compute $\bar{x} = \alpha^y v^e \pmod{p}$,

The signature (e, y) is valid on m if and only if

$$e = H(\bar{x}, m).$$

Other than smart cards, there are also different signature schemes designed for different purposes. For example, Lamport signature scheme (Lamport, 1979), is a one-time signature, which is secure if only one message is signed, and can be verified an arbitrary number of times. The scheme relies on the nature of one-way function, f, which given any value v, it is computationally infeasible to find d, such that $f(d) = v$.

Lamport Signature Scheme

Key Generation

Choose a positive integer k, and set of messages $P = \{0, 1\}^k$.

Choose $f : Y \to Z$ be a one-way function.

Choose randomly $y_{i,j} \in Y$ for $1 \leq i \leq k$ and $j \in \{0, 1\}$.

Compute $z_{i,j} = f(y_{i,j})$.

Output $2k$ values $y_{i,j}$ as the public key.

Output $2k$ values $z_{i,j}$ as the secret key.

Signing

Let $m = m_1, m_2, \ldots, m_k \in \{0, 1\}^k$ be a message.

$sig(m_1, \ldots, m_k) = (y_{1,m_1}, \ldots, y_{k,m_k}) = (s_1, \ldots, s_k)$.

Verification

Check whether $f(s_1, \ldots, s_k) = z_{i,m}$ for all $1 \leq i \leq k$.

If yes, then $(m_1, \ldots, m_k)$ is a valid signed message with signature $(s_1, \ldots, s_k)$.

Chaum and vanAntwerpen constructed an undeniable signature scheme (Chaum and vanAntwerpen, 1990), which requires the signer's response rather than relying on the public key in order to verify the authenticity of the signature. The next part shows the undeniable signature scheme which consists a signing algorithm, a verification protocol. If the signer (Alice) refuse to admit as the she signed the message, there is also a follow-up disavowal protocol to verify the signature does belong to the signer.

Chaum-vanAntwerpen Signature Scheme

Key Generation

Choose two prime, $p = 2q + 1$ and q. Assume the DLP in $\mathbb{Z}_p^*$ is intractable.

Choose an element $\alpha \in \mathbb{Z}_p^*$ of order q.

Let $1 \leq a \leq q - 1$, compute $\beta = \alpha^a \bmod p$.

Suppose G is a multiplicative subgroup of $\mathbb{Z}_p^*$ of order q,

Let $\mathcal{P} = \mathcal{A} = G$, and define

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}.$$

Output public key (p, α, β) and secret key (a).

Signing

To sign a message $m \in G$, define signature s,

$$s = \mathbf{sig}_K(m) = m^a \pmod{p}.$$

Verification

For $m, s \in G$, Bob will verify Alice's signature by the following steps:

1. Bob choose two random integers $e_1, e_2 \in \mathbb{Z}_q$.
2. Bob calculates $c = s^{e_1}\beta^{e_2} \pmod{p}$ and sends it to Alice.
3. Alice calculates $d = c^{a^{-1} \pmod{q}} \pmod{p}$ and replies it back to Bob.
4. Depending on Alice's reply, Bob verifies the signature by computing

$$d \equiv m^{e_1}\alpha^{e_2} \pmod{p}.$$

Disavowal Protocol:

$$d \neq m^{e_1}\alpha^{e_2} \pmod{p}.$$

1. Bob chooses two random integers $f_1, f_2 \in \mathbb{Z}_q^*$.
2. Bob calculates $C = s^{f_1}\beta^{f_2} \pmod{p}$ and sends it to Alice.
3. Alice calculates $D = C^{a^{-1} \pmod{q}} \pmod{p}$ and replies it back to Bob.
4. Bob checks if Alice was lying about her signature by verifying that $D \neq m^{f_1}\alpha^{f_2} \pmod{p}$.
5. Bob concludes that s is a forgery if and only if

$$(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod{p}.$$

Next, vanHeyst-Pedersen proposed another fail-stop signature scheme (vanHeyst and Pedersen, 1993). By taking the properties of undeniable signatures and one-time signatures, system will be stopped once everyone knows scheme has been broken, such that the signer able to prove that the adversary's signature is a forgery. This is useful for the online payment systems, because we assume that the financial intermediaries does not able to forge the signatures.

vanHeyst and Pedersen Signature Scheme

Key Generation

Choose two primes $p = 2q + 1$ and q is also a prime. Assume the DLP in $\mathbb{Z}_p^*$ is intractable.

Choose an element $\alpha \in \mathbb{Z}_p^*$ of order q.

Let $1 \leq a_0 \leq q - 1$ and define $\beta = \alpha^{a_0} \pmod{p}$.

A trusted third-party of authority generates the values $(p, q, \alpha, \beta, a_0)$, such that (p, q, α, β) is known to everyone, while a_0 is kept secret.

Let $\mathcal{P} = \mathbb{Z}_q$ and $\mathcal{A} = \mathbb{Z}_q \times \mathbb{Z}_q$. A key has the form

$$K = (\gamma_1, \gamma_2, a_1, a_2, b_1, b_2),$$

where $a_1, a_2, b_1, b_2 \in \mathbb{Z}_q$,

$$\gamma_1 = \alpha^{a_1}\beta^{a_2} \text{ (mod } p),$$

$$\gamma_2 = \alpha^{b_1}\beta^{b_2} \text{ (mod } p).$$

(γ_1, γ_2) is the public key and (a_1, a_2, b_1, b_2) is the secret key.

Signing

To sign a message $m \in \mathbb{Z}_q$, define

$$\mathbf{sig}_K(m) = (y_1, y_2),$$

where

$$y_1 = a_1 + mb_1 \text{ (mod } q),$$

$$y_2 = a_2 + mb_2 \text{ (mod } q).$$

Verification

To verify the signature $y = (y_1, y_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, we have

$$\mathbf{ver}_K(m, y) = \text{ true iff } \gamma_1\gamma_2^m \equiv \alpha^{y_1}\beta^{y_2} \text{ (mod } p).$$

Elliptic Curve Digital Signature Algorithm (ECDSA) (Johnson et al., 2001), hardness problem lies in the computational intractability of the elliptic curve discrete logarithm problem (ECDLP), which appears to be significantly harder than discrete logarithm problem. Thus, a scheme can be made with smaller parameters but with equivalent levels of security. An elliptic curve is defined over $\mathbb{F}_p$ with equation $y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$. The main difference between the DSA and ECDSA is that, in DSA, the value a^k mod p is reduced modulo q to yield a value γ which is the first component of the signature (γ, δ). In the ECDSA, the analogous value is r, which is the x-co-ordinate of the elliptic curve point kA, reduced modulo q. This value r is the first component of the signature (r, s). Finally, in the ECDSA, the value s is computed from r, m, k, and the message x in exactly the same way as δ is computed from γ, a, k and the message x in the DSA.

Elliptic Curve Digital Signature Algorithm

Key Generation

Suppose that E is an elliptic curve defined over $\mathbb{F}_p$.

Choose large prime number p.

Choose a point A with prime order q on E, such that the DLP in $\langle A \rangle$ is infeasible.

Let $\mathcal{P} = \{0,1\}^*$, $\mathcal{A} = \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, and define

$$\mathcal{K} = \{(p, q, E, A, x, B) : B = xA\},$$

where $0 \leq x \leq q-1$.

We get public key (p, q, E, A, B) and secret key (x).

Signing

To sign message m with hash function SHA-1.

Randomly choose secret number k, where $1 \leq k \leq q-1$.

Compute $kA = (u, v)$, and $r = u \pmod{q}$.

Compute $s = k^{-1}(\text{SHA-1}(m) + xr) \pmod{q}$.

Output signature pair (r, s).

(If $r = 0$ or $s = 0$, choose another value for k and repeat steps above)

Verification

To verify for message $m \in \{0,1\}^*$ and signature pairs $r, s \in \mathbb{Z}_q^*$,

Compute $w = s^{-1} \pmod{q}$.

Compute $i = w\text{SHA-1}(m) \pmod{q}$.

Compute $j = wr \pmod{q}$.

We get $(u, v) = iA + jB$.

The signature pair is valid if and only if

$$u \pmod{q} = r.$$

Table 2.1: Comparisim between Signature Schemes

Schemes	Public Key	Secret Key	Signing	Verification	Hardness
RSA	n, e	d, p, q	1 exp	1 exp	IFP
Rabin	n, b	p, q	1 add, 1 mult	1 add, 1 mult	IFP
Fiat Shamir	f, I	r, s_j	1 exp, 1 func, 1 mult, 1 add	1 func, 1 exp, 1 mult, 1 add	IFP
GMR	f, r_ε^f, g, 2^b	f^{-1}, g^{-1}	1 H, 1 claw-free permutation	1 H, 1 claw-free permutation	IFP
ElGamal	p, α, β	a	1 exp, 2 mult, 1 add	3 exp	DLP
Schnorr	v	s	1 exp, 1 H, 1 mult, 1 add	2 exp, 1 multi	DLP
Lamport	$y_{i,j}$	$z_{i,j}$	1 one-way func	1 one-way func	DLP
Chaum vanAntwerpen	p, α, β	a	1 exp	5 exp, 2 mult	DLP
vanHeyst Pedersen	γ_1, γ_2	a_1, a_2, b_1, b_2	2 mult, 2 add	3 exp, 2 mult	DLP
ECDSA	p, q, E, A, B	x	1 H, 1 add, 3 mult	1 H, 4 mult, 1 add	ECDLP

(add = addition, mult = multiplication, exp = exponent, func = function, H = hash function)

CHAPTER 3: THE BLOCKCHAIN TECHNOLOGY

The first blockchain was conceptualized by Satoshi Nakamoto (Nakamoto, 2008) as a core component for the cryptocurrency, bitcoin. The purpose of this cryptocurrency is to allow online payments to be sent directly from one party to another by using a peer-to-peer network through the design of blockchain technology. Blockchain is a peer-to-peer system of transacting values with no trusted third parties in between (Singhal et al., 2018). This means that the blockchain technology is a distributed and decentralized ledger of information or transactions that is duplicated across all nodes within the system. Furthermore it cannot be changed or altered due to the design of the blockchain technology which we will be discussing in the later sections. Nowadays this open-sourced blockchain can also be configured to be used in various other Internet technologies, other than the forementioned cryptocurrency. The invention of blockchain evolves around three main areas: Game Theory, Computer Science and Cryptography. Next, figure 3.1 shows a simple illustration of the blockchain structure, while figure 3.2 and table 3.1 shows the structure of a single block in blockchain. Finally, table 3.1 gives a brief history on how blockchain has evolved and developed over the years (Gichigi, 2018).

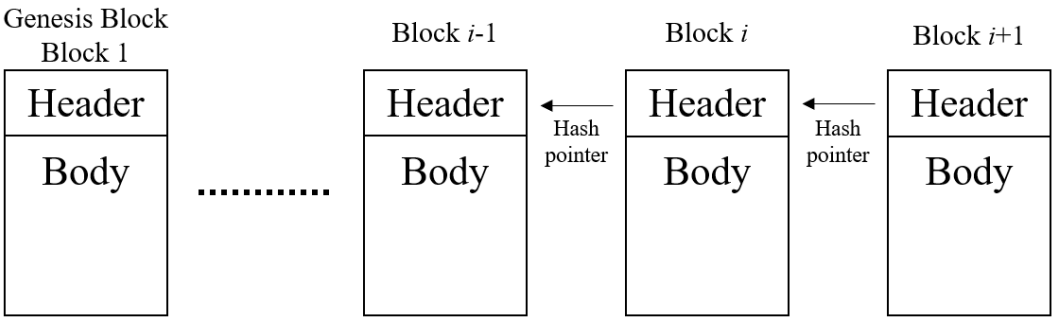

Figure 3.1: Simple illustration of the structure of the blockchain

Block Structure Components

Field	Description
Magic Number	It shows whether the block is from the mainnet or the production network. For example, Bitcoin has a fixed value of 0xD9B4BEF9 as the magic number, which also indicates the beginning of the block.
Block Size	It shows the size of the block, which is 1MB for Bitcoin.
Block Header	It holds a large information of the blockchain.
Transaction Counter	It shows the number of transactions (or other data) with various sizes included within the block.
Transaction List	All the remaining space in a block which lists all the transactions within a given block.

Block Header Components

Field	Description
Version	The same version number is assigned to each node that runs the blockchain protocol.
Previous Block Hash	It is obtained by hashing the block header of the $i-1$ block as shown in figure 3.1.
Merkle Root	It is the root hash of a Merkle tree, which is formed by hashing the transactions or data in a block.
Timestamp	It shows the approximate time that a block is formed using the Unix time format.
Difficulty Target	The difficulty level of hash puzzle set in the process of forming a block (Bitcoin mining).
Nonce	The random number that satisfy the proof-of-work hash puzzle.

Table 3.1: Descriptions of the block structure and header structure from Figure 3.2

Table 3.2: A brief history of blockchain (Emphasizing on Bitcoin & Ethereum) (Gichigi, 2018)

Year	Progression of Blockchain
2009	Satoshi officially launches Bitcoin, and the genesis block was mined (Wallace, 2011).
2010	Lazlo Hanyecz had bought two Papa John's pizzas with 10,000 BTC on 22nd May (Bitcoin Pizza Day) (Wallace, 2011).
2011	An illegal online marketplace, Silk Road started using Bitcoin in transactions (Gayathri, 2011).
2012	More cryptocurrencies are created (Namecoin, Litecoin, Swiftcoin, Tether, OpenCoin).
2013	FBI shuts down Silk Road (Hill, 2013). Ethererum, which introduced Smart Contract (blockchain 2.0) was proposed by Vitalik Buterin (Buterin, 2015).
2014	Ethereum kickstarted with Initial Coin Offering (ICO).
2015	Ethereum launches own blockchain.
2016	Ethereum lost 50 million USD to a hacker due to weakness in code (Popper, 2016). Google joins IBM, Amazon and Microsoft in testing blockchain services with clients (Bloomberg, 2016).
2017	More countries such as Japan and Russia started allowing cryptocurrency as legal currency (Kharpal, 2017).
2018	Companies started exploring the case and uses of blockchain technology.

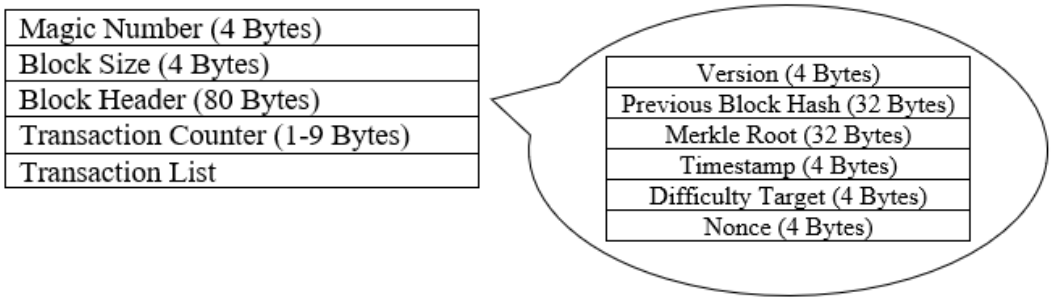

Figure 3.2: Block structure and header structure of Bitcoin blockchain

3-1 Game Theory

The purpose of Game Theory is to solve complicated real-life problems by explaining in the form of games being played. Any form of decision making where two or more players involved utilizing strategic behaviors are considered game theory. This theory is applied in blockchain to ensure the users or participants play by the rules, while maintaining the best possible outcome for everyone to ensure the stability of the blockchain. In this section, we cover four different problems in game theory.

Nash equilibrium states that there exists at least one equilibrium scenario (or win-win situation), whereby all the players can get the most profit while no one will gain more by changing their strategies, assuming all the players already know the strategies of the game. This means that every player in the game wants the best outcome for themselves, playing the best strategies and assuming the worst for other players, will eventually leads to the Nash equilibrium. On the other hand, the **prisoner's dilemma** takes account the global optimum rather than the best move as an individual. It is because in certain situation, being selfless and compromising with other players, may yield a much better outcome for everyone.

Both of these theories are used in verifying and adding transactions or information into the blockchain, also known as 'Bitcoin mining' for cryptocurrency Bitcoin (Drescher, 2017). The blockchain is completely open and accessible to everyone, including dishonest people. As a result, it cannot be guaranteed that the transactions sent through the network are correct. This means that everyone can receive the maximum profit if and only if they all work together and never verify each other's mistakes within the blockchain system, similar to prisoner's dilemma. Due to the rules of the blockchain-algorithm, all nodes of the system have an incentive to process transactions

correctly and to supervise and point out any mistakes made by the other peers. This concept relies on the selfishness of an individual to maximize own profitability to allow only valid blocks in the blockchain system.

The **Byzantine Generals' Problem** was encountered by the Byzantine's generals while conquering a new land. In reality the total army is large enough to win the war, but only if all the generals agree to invade the city altogether. But there are too many challenges to consider in fighting a war such as setting up a reliable communication system and ensuring no traitors among the generals. This is similar to real-life problems on how to reach a consensus between stakeholders in a company, or how to make sure that the distributed database or ledger is consistent, for example Bitcoin. Finally the **zero-sum game** states that one player's gain and another player's loss is equivalent, which is often used in transactions such as how financial intermediaries charge transactions costs.

In order to maintain the consistent state of ledger and to reach a consensus among the nodes, the blockchain relies heavily on the Byzantine Generals' Problem. Due to the network latency issues, the information in forming a new block may get lost, delayed, or arrive in any order. In a real world situation, people often reach a consensus by following the majority vote, or by following the individual who holds the most power. Similarly, the blockchain system will follow the longest-chain-criterion, and heaviest-chain-criterion.

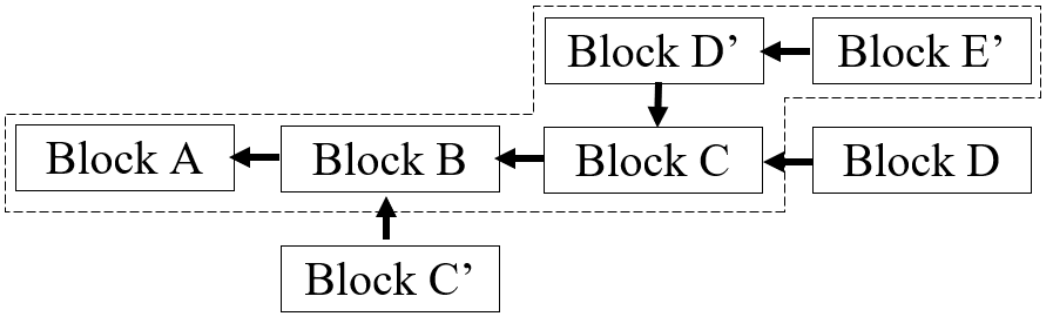

Figure 3.3: Illustration of the longest-chain-criterion

As shown in Figure 3.3, assuming the blockchain has already formed Block A to D. However, a node with slower network formed another Block D' and Block E' before broadcasting to the system. This tells all the other nodes to follow the longest-chain-criterion, and continue forming new blocks from Block E', thus causing Block

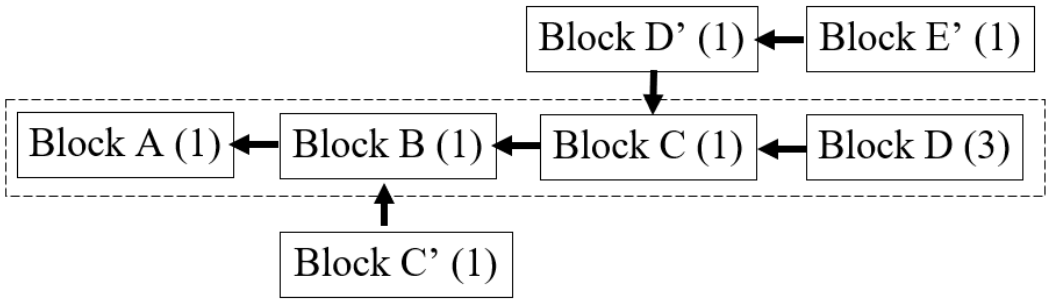

Figure 3.4: Illustration of the heaviest-chain-criterion

C' and Block D to be orphan blocks and no rewards are given. However, in the case of heterogeneous difficulty levels (mixture of different hash puzzles), heaviest-chain-criterion is used instead, as shown in Figure 3.4. The difficulty level of each block is stated in bracket, and the heaviest chain has a total weightage of 6 $(1 + 1 + 1 + 3)$, which is higher than the chain from Block A to Block E' (weightage of 5). Thus, the orphan blocks are Block C', Block D' and Block E'.

3-2 Computer Science

The Blockchain by definition, is a group of blocks containing information linked together with hash pointers. Referring back to figure 3.1, a hash pointer is the hashed output of block $i - 1$, which is then included in block i pointing to the previous block. Next, after the block i is fully completed and filled with data or transactions, block i is hashed again and stored in block $i + 1$, and so forth. All of the blocks contain information of the previous block until the first block, also known as the genesis block. Knowing the nature of hashing being deterministic, it is impossible to alter the information within a block, because having a different hash output allows the system to detect an invalid block in the blockchain.

Since it is impossible to change the history or the previous blocks in the blockchain, any attempt in tempering with the block requires an adversary to keep changing the hash output in every block header until the final or the most recent block. In addition, the distributed nature of the blockchain means that the blockchain is known by everyone at all times, making it much more difficult and computationally expensive to alter the data within a block of the blockchain. To compute the hash output, we can use either the SHA-2 or SHA-3 family of hash functions that will be covered in the next

subsection.

Other than using hash functions on the blocks, it is also used on the transactions within the blocks, forming a Merkle Tree, which was invented and named by Ralph Merkle. First, the each individual transaction or data is paired and hashed at the leaf level. Then the hashed output is paired with another hashed output of paired data to be hashed again, continuing all the way up to the Merkle root, similar to a tree diagram. This Merkle tree is able to efficiently verify a single transaction from the large number of blocks in the blockchain. Let n be the number of transactions in a Merkle tree, then the verification process will only take $log(n)$ computation time.

3-3 Cryptography

One of the most important usage of cryptography in blockchain is the Cryptographic Hash Functions, which is Secure Hash Algorithm (SHA) and is mentioned extensively in the previous sections. SHA-256 is the version being used in blockchain currently. The properties of hash functions are listed as follows:

1. The output of the hash function is fixed length, determined by the hash family, while the input and be string of any length.

2. For any given message, the hashing of the message should be efficiently computed.

3. A hash function is deterministic function, that is, same input will always generate the same hash output.

4. It is computationally infeasible to compute the input, given hash output.

5. It is infeasible to find two different inputs that hash to the same value (collision resistance).

6. First preimage resistant: It is infeasible to find the input X from the output $H(X)$.

7. Second preimage resistant: It is infeasible to find an input Y, such that $H(Y) = H(X)$.

Next, we talk about how ECDSA is used in the blockchain technology. The algorithm contains three parts: key generation, signature generation and signature verification. For any information or transactions to be recorded in the blockchain, they need to be verified by the system with a signature scheme. The ECDSA is mentioned in Chapter 2: Signature Schemes. ECDSA is currently used in cryptocurrencies such as Bitcoin and Ethereum with parameters $secp256k1$ (Koblitz curve for 256-bit elliptic curve domain parameters over $\mathbb{F}_p$), as defined in Standards for Efficient Cryptography (SEC) (Brown, 2010). In Bitcoin, every users will have an account or wallet which generates a private key and public key. Next, the public key is hashed twice with SHA-256 and RIPEMD-160 in order to generate an address which can be used to receive transactions, while the private key is used to verify or approve a transaction in the blockchain system.

3-4 Applications of the Blockchain Technology

Even though blockchain was meant for cryptocurrency initially, researchers have been trying to explore other applications for the blockchain technology due to it being open-sourced. In order to specifically redesign the blockchain technology for other real-world applications, we need to understand how to build blockchain solutions from the bottom. First we need to understand the functionality of different layers of blockchain before we get to the potential of blockchain outside of cryptocurrency. The following layers are defined by: Application, Execution, Semantic, Propagation and Consensus (Singhal et al., 2018).

In the **Application layer**, users can rely on the characteristics of blockchain, such as distributed, decentralized, immutable ledger, and use it as a simple and secure data storage or data processing. In other words, using blockchain as an off-chain network or backend system that deals with the heavy lifting for information processing. In the **Execution layer**, as the name suggests, executes instructions in the form of smart contracts or deterministic functions ordered by application layer, in order to ensure consistencies in data processing. As for the **Semantic layer**, its role as a logical layer is to validate every piece of information such as Bitcoin transactions and smart contracts. This layer also defines how the blocks in the blockchain are linked together, through

hashing. Next, we have the **Propagation layer**, which can be known as peer-to-peer communication layer because its role is to broadcast any new updates to every nodes in the blockchain system. This will make sure that the whole network will always be the updated, regardless of the latency issues due to the capacity of the nodes or network bandwidth. Finally we have the **Consensus layer**. The main purpose of this layer is to ensure the consistency of the blockchain system. There are several methods to enforce consensus among users such as Proof of Work (PoW) and Proof of Stake (PoS), to ensure the safety and security of the blockchain.

Table 3.3: How different layers can affect the blockchain system

Layers	Use Cases
Application	Act as an simple data storage system which is immutable and widely accessible to everyone.
Execution	Adding programming scripts or additional rules into the blockchain system.
Semantic	Controls how information is treated and processed, either as transactions or personal data.
Propagation	To adjust the size of blockchain, to suit the usage of private or public blockchain.
Consensus	Choosing how reward-punishment system works, such as using PoW or PoS.

Other than using blockchain technology in cryptocurrencies, companies such as Deloitte, IBM and Malaysia Industry-Government Group for High Technology (MIGHT) (Deloitte, 2016, IBM, 2018, Thambyrajah and Lee, 2018) have started brainstorming potential use cases for the blockchain technology. Even though most of them are just concepts or still under proposals, all of them shows promising improvements towards many different sectors with blockchain technology.

3-4-1 Banking Sector

Currently one of the most common issue faced by **banking sector** is money laundering. In addition, using the Know Your Customer (KYC) requests will often require roughly 1 to 2 months to complete to obtain a full result. Blockchain technology can help

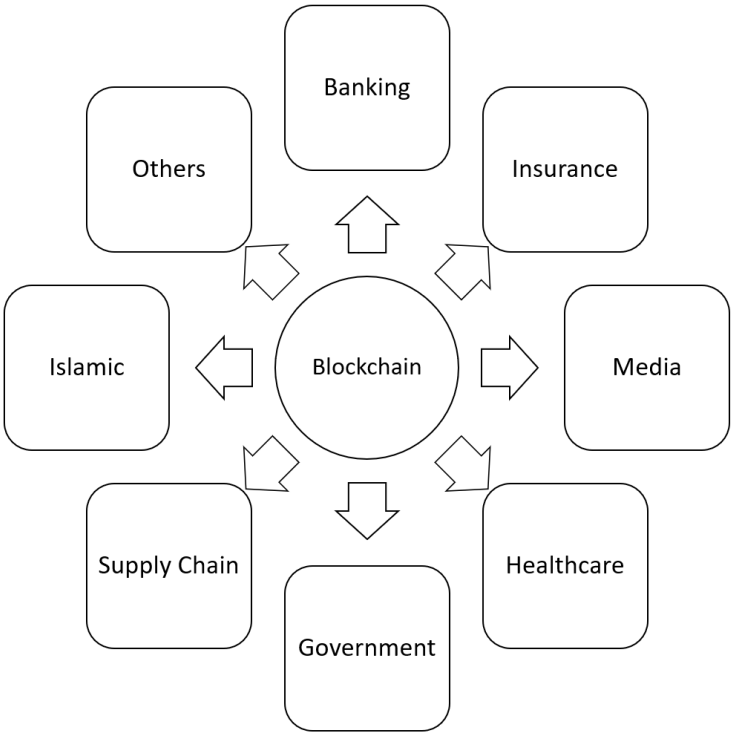

Figure 3.5: Applications of the blockchain technology

speeds up the KYC compliance using a distributed database of customers' details. Other than that, the peer-to-peer factor of blockchain allows the data to be updated to all participating banks in a short amount of time, and remove the repeated efforts in carrying out KYC checks on certain individual, thus reducing the computational cost.

3-4-2 Insurance Sector

In **insurance sector**, customers often faced with a complex and drawnout claims process due to complicated legal language used in insurance contracts. On the other hand, insurers are also facing threats from insurance fraud such as fake claims across multiple insurance companies. By adopting the smart contracts in blockchain technology, customers and insurers can manage claims in a transparent, responsive and irrefutable manner. Contracts and claims are recorded and validated with blockchain, so that smart contracts can trigger payments automatically when certain conditions are met.

3-4-3 Media Industry

The bottleneck of **media industry** or any form of intellectual properties is the inability to keep up with digital technologies. Traditionally, artists' royalties are paid according to the airplay statistics and copyright claims maintained by music label companies. Nowadays revenue are generated through the advertisements in streaming websites such as YouTube, which causes lack of transparency for the content creators. This problem can also be solved using smart contracts, by executing specific instructions, the content creators would be properly compensated whenever there is revenue on the content created. In addition, by using the cryptographic hash function from blockchain technology on the contents, consumers can ensure the originality of the contents such as movies and music.

3-4-4 Healthcare Sector

Current challenges in **healthcare sector** includes difficulty in clinical trial management, disparate record-keeping systems, and quantifying the worth of medical care. By storing patients' medical data using blockchain technology, patients have more control over their own information through verifiable consent, in addition allowing clinical data to be automatically aggregated, replicated and distributed among researchers and practitioners with greater auditability, provenance tracking and control. Other than that, having a complete historical data of medical costs in the form of blockchain can avoid patients being charged with unnecessary hidden charges.

3-4-5 Government Sector

There are also many frauds involving identity scam such as credit cards, passports and ownerships of assets. The **government sector** can rely on the blockchain technology by storing citizens' data into a single and trustworthy collection of digital identity documents. A private blockchain can be used in this situation to only allow trusted parties or government administrators to access, identify, or verify citizens' information. Additional information can also be stored under each citizen such as ownership of houses, businesses, or debts to a single, shared ledger with blockchain technology.

3-4-6 Supply Chain Sector

In the **supply chain sector**, there are many parties involved such as manufacturing, processing, harvesting and transporting departments. There are cases involving counterfeit parts in automobile assembly or the process of Halal products, whereby it is difficult to trace the source of the problem. Blockchain technology allows the information across the supply chain to be accessible to component vendors, transportation owners, maintenance crew, and regulators. This allows full transparency throughout the supply chain, from raw products to final products, especially the process of Halal products for the Islamic countries.

3-5 Islamic Applications

Malaysia government committees have partnered up with South Korean blockchain lab, IncuBlock, to work on constructing a Syariah compliant blockchain technology (Zuckerman, 2018). It is because Islamic teachings stated that they must not profit without effort, which means that Islamic people are not allowed to profit from conventional interest rates from banks, thus Islamic Finance is enforced under Syariah Law. Furthermore the cryptocurrency, Bitcoin is still to new and cannot be concretely defined as money according to Islam, thus not considered as "halal investment".

In addition, blockchain technology improves the transparency and efficiency of the Zakat process (alms-giving), in which how the funds are distributed to the Islamic people in need (Pikri, 2018). Aidatech, a Dublin-based fintech company created an application which allows muslims to choose and also trace which project they would like to contribute, such as schools or shelter for the needy.

3-6 Other Applications

Universiti Tunku Abdul Rahman (UTAR), with the support from Silverlake Symmetry and Technology Research Sdn. Bhd., has launched the Key Generation of UTAR integrated Cumulative Grade Point Average (iCGPA) and Blockchain Certification System at Sungai Long Campus on 9 February 2018 (UTAR, 2018). By relying on the nature of blockchain being immutable, accessible, distributed, sustainable, verifiable and se-

cure, digital certificates can be stored inside the blockchain throughout lifetime without any centralized database. Furthermore the UTAR Blockchain Certificate (in the form of PDF file) can be used to verify the authenticity of the certificate issued by the University. This allows hiring companies to easily verify certificates of the university's graduates online using the blockchain technology.

The blockchain technology can also be used in voting systems, such as the General Election in Malaysia. There have been alleged claims that thousands of foreign workers from Borneo came to Malaysia to vote for a particular party, without having an authentic Malaysian Identification Card (IC) (Ibrahim, 2013). Dr. Mahathir Bin Mohamad also claimed that there was a deliberate attempt to delay the election result by the commission refusing to sign the form, which was later refuted by election commission chairman, Hashim Abdullah (Rajendran, 2018).

First, we take a look at the illegal participants of the general election. If we store every citizens information in a private government blockchain system, every voting centre will have easy access to this database, thus easily verifying the biometric signature, or fingerprint of the voters. We can configure the **Application Layer** to store new data every time a Malaysian IC is issued. Whenever there is a general election, we can use the **Execution Layer** to produce the updated list of citizen who reached the legal voting age, by adding a simple programming rule in the blockchain system. In the **Semantic Layer**, similar to how personal password is stored, every votes are anonymously stored inside the system because of the secure hash algorithm as discussed in the previous chapter. By binding the votes with voters, we can also make sure that there will be no duplicate votes from a single voter. **Propagation Layer** allows the blockchain to have restricted access for the public and act as a private blockchain, and to ensure no delay in voting results due to the peer-to-peer communication behavior of this layer. Finally we can configure how the votes are verified and counted in the **Consensus Layer**, for example having a social credit system (implemented in China) and issue additional credits for the people who help verifying the votes.

CHAPTER 4: POST QUANTUM SIGNATURE SCHEMES

4-1 Quantum Computing

In classical binary computers, every bit can only be either 0 or 1. However, quantum computers run in qubits, which can exist in quantum superposition whereby every qubit exists in both state of 0 and 1, allowing a much faster parallel data processing. In quantum physics, any calculation or processes acts as if it can run on all paths simultaneously. The probability of any particular outcome of the experiment is then proportional to the squared of the absolute value of the sum of the amplitude of all the paths leading to that outcome. A quantum computer behaves the same way by proceeding down all possible paths at once, with complex amplitude in each of the paths (Shor, 1994). Shor's proposed quantum algorithms for finding discrete logarithms and factoring integers on a quantum computer can break the RSA, DSA and ECDSA algorithms, which will easily breach every security system using classical cryptographic algorithm.

Previous researchers have proposed different signature schemes that resist the quantum computing attack, especially lattice-based cryptography (Lauter, 2017). The concept behind lattice-based cryptography is that adding enough noise or error vectors to the inner products of the secret basis, which in turn causing a hard decoding problem. This does not rely on linear algebra or number theories as a hard problem, thus lattice is a potential solution that resists quantum attacks. More information on lattice-based cryptography will be discussed in the next chapter.

4-2 Lattice-based Cryptography

This chapter provides more literature review on lattice-based hardness problems and the proposed security models from the previous researchers. Two main references for this chapter are from Micciancio and Regev (2009) and Peikert (2016), together with lecture notes on lattice cryptography from Bar-Ilan University (Peikert et al., 2012).

A lattice is a set of points in n-dimensional space with a periodic structure. Given n-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$, the lattice generated by them is the set of vectors

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}$$

Lattice-based cryptography carries the following properties:

1. Conjectured security against quantum attacks - Peter Shor (Shor, 1994) gave efficient quantum algorithm which would render number-theoretic systems (such as integer factorization problem and discrete logarithm problem) insecure in a future. Currently there is no quantum algorithms that able to solve lattice-based hard problems efficiently.

2. Simple, efficient, parallel algorithms - The lattice-based cryptosystems consist mostly linear operations on vectors and matrices, thus having a more efficient constructions of certain lattice rings such as the NTRU cryptosystem.

3. Secure against worst-case hardness - In classical cryptosystems, certain hard problems often turn out to be easier on the average case hardness. However, Miklos Ajtai proved for lattice that certain problems are hard on the average as long as some related lattice problems are hard on the worst case (Ajtai, 1996).

4. Constructions of versatile and powerful cryptographic objects - Fully homomorphic encryption (FHE) allows any third party user to run calculations on the encrypted data, without decrypting or gain any information on the original data. Craig Gentry proposed the first lattice-based cryptosystem with the characteristic of FHE (Gentry, 2009).

4-3 Hardness Problems

4-3-1 Shortest Vector Problem (SVP)

Definition 3. Given a basis $B \in \mathbb{Z}^{m \times n}$ and $\gamma > 0$, find a nonzero lattice vector $v \in B\mathbb{Z}\backslash\{0\}$ such that $\|v\|_p \leq \gamma \lambda_1^p(B)$.

Remark: If not stated otherwise, $\gamma = 1, p = 2$.

The γ serves as an approximation factor in the lattice hard problem. Ajtai-Dwork public key cryptosystem has proven that their scheme is secure under worst-case (Ajtai and Dwork, 1997).

4-3-2 Closest Vector Problem (CVP)

Definition 4. Given a basis $B \in \mathbb{Z}^{m\times n}$ and $\gamma > 0$, and $t \in B\mathbb{R}^n$, find a nonzero lattice vector $v \in B\mathbb{Z}^n$ such that $\|t - v\|_p \leq \gamma\lambda_1^p(B)$.

Remark: If not stated otherwise, $\gamma = 1, p = 2$. Inhomogeneous version of SVP.

The CVP is a generalization of the SVP. Supposed that the input to SVP is the basis, $B = [b_1, b_2, \ldots, b_n]$. Consider another basis $B^i = [b_1, \ldots, 2b_i, \ldots, b_n]$ and let x_i be the vector returned by CVP(B^i, b_i). The claim is that the shortest vector in the set $\{x_i, -b_i\}$ is the shortest vector in the given lattice, thus the hardness reduction from SVP is to CVP.

The next two hardness problems are modular lattice problems, which are typically defined as average-case hardness.

4-3-3 Short Integer Solution (SIS)

Definition 5. Let q be a prime and $A \in \mathbb{Z}_q^{m\times n}$, where A is chosen from a distribution negligibly close to uniform over $\mathbb{Z}_p^{m\times n}$. Then $\mathcal{L}_q^{\perp}(A) = \{x \in \mathbb{Z}^n : Ax \equiv \overrightarrow{0} \in \mathbb{Z}^m \pmod q\}$ is an n-dimensional lattice. The task is to find a vector $v \in \mathcal{L}_q^{\perp}(A)$ with $\|v\|_p \leq \beta$.

The concept of SIS is that, given m uniformly random vectors $a_i \in \mathbb{Z}^n$ (matrix $A \in \mathbb{Z}^{m\times n}$). It is computationally hard to find a nonzero integer combination that sums to zero. Ajtai has proven that solving SIS is at least as hard as solving the decisional approximate SVP (Ajtai, 1996).

4-3-4 Learning With Errors (LWE)

The parameters of LWE are $n, q, \chi \in \mathbb{Z}$, where χ is the relative error rate, taken to be a discrete Gaussian of width αq for some $\alpha < 1$ (Regev, 2009).

Definition 6. For a secret vector $s \in \mathbb{Z}_q^n$, the LWE distribution $A_{s,\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $a \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e \pmod q)$.

- Search-LWE: Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ drawn from $A_{s,\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ (fixed for all samples), find s.
- Decision-LWE: Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ distinguish whether the sample is uniformly distributed, or according to $A_{s,\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$.

The search version of LWE is to find the secret given LWE samples, while the decision version is to distinguish between LWE samples and uniformly random samples. Regev has proven that solving LWE is at least as hard as quantumly solving decisional approximate SVP (Regev, 2009).

4-4 Lattice Reduction

The lattice reduction allows us to find short vectors in a basis, which relates to the hardness of SVP. This is important because most of the lattice hardness problems can be reduced to SVP, thus solving SVP can lead to breaking schemes based on SIS or LWE hardness problem. The most commonly used lattice reduction method is the LLL algorithm (Lenstra et al., 1982). This is a polynomial time algorithm for SVP that achieves an approximation factor of $2^{\mathcal{O}(n)}$ where n is the dimension of the lattice. According to the LLL algorithm:

1. $(b_1, \ldots, b_n)$ is a basis of $\mathcal{L}$.
2. $|b_1| \leq 2^{(n-1)/2} \times \lambda$.
3. $|b_1| \leq 2^{n(n-1)/2} \times (\Delta(\mathcal{L}))^{1/n}$.

where λ_1 is the length of the shortest non zero vector of $\mathcal{L}$, and $\Delta(\mathcal{L})$ is the determinant of $\mathcal{L}$, that is the euclidean volume of the q-dimensional parallelepiped enclosed by $b_1, \ldots, b_n$. Condition (3) states that the length of b_1 is shorter than a basis with mutually orthogonal vectors of equal length. LLL algorithm also stated that, $|b_i| \leq 2^{(n-1)/2} \times$

λ_i. This means that the algorithm can compute the sublattice, if i linearly dependent vectors of the lattice are very small.

Let m be the dimension of a space, n be the dimension of the lattice, $B_{\mathcal{L}}$ be the basis of lattice, d be the maximum number of bits in matrix $B_{\mathcal{L}}$. Then the running time of LLL algorithm is polynomial $\mathcal{O}(mn^5d^3)$ (Joux and Stern, 1998).

Finally, we can make a conjecture that, there is no polynomial time algorithm that approximates lattice problems to within polynomial factors. Thus, proving that SVP is indeed a valid hardness problem for lattice-based encryption schemes.

4-5 Semisimple Cyclic and Abelian Codes

In this section, we consider the group algebra code which is an ideal of the group algebra KG, where G is a finite abelian group written multiplicatively, and K is a field with $char(K)$ does not divide $|G|$. In this case, KG is semisimple, that is, KG decomposes into the direct sum of minimal ideals as follows:

$$KG = I_1 + \cdots + I_s. \tag{4.1}$$

This decomposition corresponds to the decomposition of the unit of the group algebra into the sum of mutually orthogonal idempotents;

$$1 = e_1 + \cdots + e_s, e_ie_j = 0 \text{ for all } i \neq j, e_i^2 = e_i, \text{ and}$$

$$I_i = KGe_i \text{ for } i = 1, 2, \ldots, s.$$

Each ideal I_i is isomorphic to the field $K(\eta_i)$, where η_i is some root of unity whose order divides $|G|$. The idempotent e_i is the unit of the ideal I_i, for $i = 1, 2, \ldots, s$. Each minimal ideal I_i of the group algebra KG defines an irreducible representation τ_i of group G on field K. Call the kernel N_i of the representation τ_i the kernel of the idempotent e_i and the ideal I_i corresponding to it. The idempotent e_i is called exact if the kernel of the e_i contains only the unit of the group G.

An arbitrary ideal V of KG can be uniquely expressed as the sum of some ideals I_j of the equation (4.1),

$$V = I_{i_1} + \cdots + I_{i_q}. \tag{4.2}$$

Thus, there are $2^s - 1$ distinct nonzero ideals of KG. When $q < s$, it is possible to carry out a direct expansion in terms of the ideal V of the group algebra KG as

follows: $KG = V - V_1$, where $V_1 = I_{j_1} + \cdots + I_{j_{q-s}}$ is the direct sum of those ideals of equation (4.1) which are not included in equation (4.2). The ideal V can be expressed in the following form:

$$V = \{x \in KG : xe_{j_1} = 0, \ldots, xe_{js-q} = 0\},$$

where e_{j_t} is the minimal idempotent of the group algebra KG and $I_{j_t} = KGe_{j_t}$.

Next, we consider the field $T = K(\eta)$, where η is the primitive $|G|$-th root of unity. Then, TG decomposes into the direct sum of n ideals isomorphic to the field T as follows:

$$TG = I'_1 + \cdots + I'_n. \tag{4.3}$$

The minimal idempotents e'_i of the group algebra TG corresponding to the ideals I'_i for $i = 1, \ldots, n$ are of the form

$$e'_i = \frac{1}{n} \sum_{g \in G} \chi_i(g)g, \tag{4.4}$$

where $\chi_1, \ldots, \chi_n$ are the characters of group G on field T (each character χ_i defined a homomorphism of group G on the group of n-th roots of unity of field T).

Next, we have idempotents $M = \{e'_i\}$ which decomposes into intersecting subsets in one-to-one correspondence with the minimal idempotents e_i for $i = 1, 2, \ldots, s$ of the group algebra KG,

$$M = M_1 \cup \cdots \cup M_s;$$

$$M_i \cap M_j = \{\} \text{ for } i \neq j;$$

$$M_i = \{e'_{i_1}, \ldots, e'_{i_{m_t}}\}.$$

The idempotents e'_j of one subset M_i are K-conjugate to each other, such that they undergo automorphisms $\epsilon \to \epsilon^\mu$ of the Galois group of field $K(\epsilon)$ over K (the automorphisms are applied to the coefficients $\frac{1}{n}\chi_i(g)$ of the idempotents e'_i). As such, we can prove that:

$$e_i = e'_{i_1} + \cdots + e'_{i_{m_i}}, \text{ for } i = 1, 2, \ldots, s. \tag{4.5}$$

All the idempotents $e_{i_1}, \ldots, e_{i_{m_i}}$ of the group algebra TG have one and the same kernel $N_i \subseteq G$, which simultaneously coincides with kernel of the minimal idempotent e_i of the group algebra KG. Hence the idempotent e_i will be exact if and only if each

of the minimal idempotents $e_{i_1}, \ldots, e_{i_{m_i}}$ of the group algebra TG occurring in the expansion of e_i is exact.

Let H be a subgroup of the abelian group G such that $G/H = \langle aH \rangle$ is a cyclic group of order m. Let $e_1, \ldots, e_s$ be all the exact minimal idempotents of the group algebra KG_1, where $G_1 = \langle b \rangle \cong G/H$. Let us compare with each idempotent $e_i = \sum \lambda_{il} b^l$ for $\lambda_{il} \in K$, the element $\overline{e_i} = \sum \lambda_{ij} a^j$ of the group algebra KG. Then the elements $u_i = \frac{1}{|H|}(\sum_{g \in H} g)\overline{e_i}$ for $i = 1, \ldots, s$ form the complete set of minimal idempotents of the group algebra KG with kernel H.

Each ideal $V = I_{i_1} + \cdots + I_{i_q}$ of the group algebra KG generates the ideal $V' = TGe_{i_1} + \cdots + TGe_{i_g}$ of the algebra TG. Since $V \subseteq V'$, therefore $d(V) \geq d(V')$.

Let $\chi_1, \ldots, \chi_n$ be some numbers in such a way that $V' = \{x \in TG : xe'_1 = 0, \ldots, xe'_r = 0\}$.

If $x = \sum_{i=1}^{n} a_i g_i$, then $xe_i = (\sum_i a_i \chi_i(g_i^{-1}))e_i$. Hence, the element $x \in TG$ belongs to the ideals V' if and only if the coefficients $\alpha_j \in K$ belong to the system of linear equations $\sum_i^n a_i \chi_1(g_i^{-1}) = 0, \ldots, \sum_i^n a_i \chi_r(g_i^{-1}) = 0$.

The matrix of the system is of the form

$$A = \begin{bmatrix} \chi_1(g_1^{-1}) & \cdots & \chi_1(g_n^{-1}) \\ \chi_2(g_1^{-1}) & \cdots & \chi_2(g_n^{-1}) \\ \vdots & \ddots & \vdots \\ \chi_r(g_1^{-1}) & \cdots & \chi_r(g_n^{-1}) \end{bmatrix}$$

The rank of the matrix A equals to r because the rows are mutually orthogonal. Because any additional $r + 1$ columns of the matrix A are linearly dependent, there exists an element $x \in V$ for which $l(x) = r - 1$. Consequently, $d(V') \leq r - 1$. We can see that $d(V') = r$ if and only if any minor of order r constructed from matrix $A \neq 0$. If $G = \langle a : a^n = 1 \rangle$ is a cyclic group, then it is always possible to choose the matrix A so that the last condition is satisfied. For this it is sufficient to put $\chi_i(a) = \eta^{-i-1}, i = 1, 2, \ldots, r$, where η is a primitive n-th root of unity. Then

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \eta & \cdots & \eta^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \eta^{r-1} & \cdots & \eta^{r(n-1)} \end{bmatrix}$$

An arbitrary minor of the r-th order of the matrix A is of the form

$$D_1 = \begin{bmatrix} 1 & \cdots & 1 \\ \eta^{i_1} & \cdots & \eta^{i_r} \\ \vdots & \ddots & \vdots \\ \eta^{i_1(r-1)} & \cdots & \eta^{i_r(r-1)} \end{bmatrix}$$

$|D_1|$ is Vandermonde's determinant and so $|D_1| \neq 0$. Therefore, if $G = \langle a \rangle$ is a cyclic group and the matrix A is as above, then $d(V') = r-1$. Obviously, $d(V) \geq r+1$. This code V is the well-known cyclic BCH code.

CHAPTER 5: INTRODUCTION TO LATTICE BASED SCHEMES

5-1 Ajtai-Dwork Public Key Cryptosystem

The public key cryptosytem constructed by Ajtai and Dwork is secure unless the worst case of unique SVP (uSVP) can be solved in polynomial time (Ajtai and Dwork, 1997). The main difference of SVP and uSVP is that, the shortest vector v is unique because any other vectors with length at most $n^c\|v\|$ is parallel to v, where c is a constant.

However, this cryptosystem is impractical because based on lattice dimension n, the public key size is $\tilde{\mathcal{O}}(n^4)$, the secret key and ciphertexts size are $\tilde{\mathcal{O}}(n^2)$ (Peikert, 2016). In order to prevent attacks on the hidden hyperplane problem, the dimension of lattice n need to be several hundreds, causing the public key size to be in the order of several gigabytes. Even so, most of the lattice-based encryption schemes are constructed similar to this system, thus this is one of the earlier popular lattice-based scheme.

Ajtai-Dwork Public Key Cryptosystem

Key Generation

Generate random $n-1$ dimensional lattice $\mathcal{L}'$ with basis $b_1, \ldots, b_{n-1}$ such that $\|b_i\| \leq M$. Let H be the $n-1$ dimentional subspace containing $\mathcal{L}'$.

Choose $d \geq n^c M$, randomly choose b_n of distance $d \leq d_L \leq 2d$ from H.

Randomly choose a basis B' with the same lattice.

Output Secret key is H, public key is (B', M).

Encryption

Let $K \geq 2^n d$, U^n is n-dimension unit cube, choose random lattice point v in cube KU^n.

Let $R \in \mathbb{R}$, $m \in \mathbb{Z}$, pertubation(R, m).

For $m : c_0 n, c_0 \geq 4, R = n^3 M$, select a value w of pertubation(R, m).

To encrypt 0, the ciphertext is $v + w$.

To encrypt 1, the ciphertext is a random point in KU^n.

Decryption

Suppose u_H is the unit vector orthogonal to subspace H, and d_L be distance between consecutive hyperplanes.

To decrypt z, compute fractional $(u_H \cdot z)/d_L$.

If within $\frac{mR}{d_L}$ of 0 or 1, $\mathbf{dec}(z) = 0$, otherwise 1.

5-2 NTRU Encryption Scheme

The NTRU encryption scheme (Hoffstein et al., 1998) is a lattice-based public key cryptosystem based on the shortest vector problem. This is the first scheme that uses polynomial rings in terms of algebraically structured lattice, which is said to be difficult to factorize the polynomials in a truncated polynomial ring.

NTRU operations are based on objects in a truncated polynomial ring $R = \mathbf{Z}[X]/(X^N - 1)$ with convolution multiplication and all polynomials in the ring have integer coefficients and degree at most $N - 1$ as follows

$$a = a_0 + a_1X + a_2X^2 + \cdots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}.$$

The integer parameters of NTRU are prime numbers $(N, p, q) \in \mathbb{Z}$, where $q > p$ and $gcd(p, q) = 1$. Whereas the polynomial parameters are (L_f, L_g, L_m, L_r), where L_f is the set of polynomials which are part of the secret key, L_g is the set of polynomials for generation of the public key, L_m is the set of polynomials representing the messages and L_r is the set of polynomial representing the blinding values, all of degree $\leq N-1$.

NTRU Encryption Scheme

Key Generation

Suppose there are two distinct polynomials f and g, with degree $\leq N-1$ and with coefficients in $\{-1, 0, 1\}$. $f \in L_f$ must satisfy

$$f f_p \equiv 1 \pmod{p},$$

and

$$f f_q \equiv 1 \pmod{q}.$$

If f is not invertible, choose another f and repeat above steps.

$$h = pf_q q \text{ (mod } q).$$

Secret Key are (f, f_p) and the public key is h.

Encryption

Choose message m in polynomial form with coefficients $\{-1, 0, 1\}$.

Choose random polynomial r as a blindling value with small coefficients.

To encrypt a message e,

$$e = rh + m \text{ (mod } q)$$

Decryption

Computing $a = fe \text{ (mod } q) = f(rh + m) \text{ (mod } q) = f(rpf_q q + m) \text{ (mod } q) = prg + fm \text{ (mod } q)$.

Choose the coefficients of a between $[-q/2, q/2]$, compute a (mod p).

$$b = a \text{ (mod } p) = fm \text{ (mod } p).$$

To recover the message m using secret key f_p,

$$c = f_p b = f_p fm \text{ (mod } p) = m \text{ (mod } p).$$

Attack: An attack on a basic cryptosystem is to recover the plaintext from the ciphertext. However, in signature schemes, the attack is to find a secret key f in order to forge a signature for any given message. Let Eve be an adversary, the following are four most common types of approach in order to recover the secret key.

Eve can launch a **brute force** attack by trying to compute every possible values of f, assuming f has very few nonzero coefficients. In order to check if f^* is the secret key, Eve can compute f^*h (mod q) and check if it has a small coefficient. Next, Eve can try to decrypt a message using f^* on any messages that is encrypted using the public key. If both conditions are valid, Eve has successfully forged the secret key $f^* = f$.

There is also a faster **meet-in-the-middle** attack. Supposed that $fh = g$ (mod q), Eve wants to find f_1 and f_2 such that $f = f_1 + f_2$ holds and such that they have the property

$$(f_1 + f_2)h = g \text{ (mod q)},$$

$$f_1 h = g - f_2 h \text{ (mod q)}.$$

If the secret key f has d number of 1's and $N - d$ number of 0's, then Eve computes every single solutions for f_1 and f_2 of equal length $1/2N$ with $d/2$ number of one's, such that f_1 are the solutions for the lower coefficients of f and f_2 for the higher coefficients. Next, compute $f_1 h$ (mod q) and $f_2 h$ (mod q) in seperate bins. Eve will be able to find the secret key f, if $f_1 h \equiv g - f_2 h$ (mod q).

Next, is the aforementioned **lattice reduction** attack, which is the most common method to break the NTRU Encryption Scheme, and the most commonly used algorithm is the Lenstra-Lenstra-Lovsz algorithm (Lenstra et al., 1982). Eve can use the LLL algorithm on the public key h in order to find the secret key f. However this lattice reduction attack can only be used on smaller lattice dimensions whereby the shortest vector are not too long.

Finally is the **Chosen Message Attack** (CMA), where Eve will try to obtain as many signature-message pairs as she can, for example from the signer's historical data. Next, Eve will try to compute the relation of the signature-message pairs and sign her own message. If Eve is able to verify her signature is valid, then she has successfully forged the signature.

5-3 The GGH Encryption Scheme

In the research of Oded Goldreich, Shafi Goldwasser and Shai Halevi, they have proposed a lattice-based encryption scheme and also signature scheme (Goldreich et al., 1997). Both of the schemes are based on the closest vector problem, and uses a trap door one way function. By adding a small error vector to any lattice basis, it will be hard to compute a vector which is close to a lattice point, unless there is a special basis as the trap door function.

GGH Encyption Scheme

Key Generation

Choose a basis R of lattice $\mathcal{L}$ and a unimodular matrix T.

Compute another basis B of the same lattice such that $B = RT$.

Secret Key is R and Public Key is B.

Encryption

Let message space $m = (m_1, m_2, \ldots, m_n)$, $-M \leq m_i \leq M$ for $i = 1, 2, \ldots, n$.

Choose a small error e, encrypted message, c:

$$\textbf{enc}(m) = mB + e = c.$$

Decryption

$$\textbf{dec}(c) = cR^{-1}T^{-1} = m$$

. In the decryption process, the Babai's rounding technique is used to remove the small error term eR^{-1}

Babai's rounding technique: Let $b_1, b_2, \ldots, b_n$ be the basis for a lattice in $\mathbb{R}^n$, let $w \in \mathbb{R}^n$ be a target vector. Then $w = \sum_{i=1}^{n} l_i b_i$ with $l_i \in \mathbb{R}$. Next, we approximate the coefficients to the nearest integers, such that $v = \sum_{i=1}^{n} \lfloor l_i \rceil b_i$.

GGH Signature Scheme

Key Generation

Choose private basis R which has small dual orthogonality defect.

Compute public basis B where $B = RT$ for some unimodular transformation matrix T.

Signature Generation

Choose message m, using encoding funciton to get $u \leftarrow Encode(m)$.

Secret key (R^{-1}, T), compute $v \leftarrow T[R^{-1}u]$.

v is the signature on m.

Verification

Public key (B, τ), $\tau > 0$ is a threshold which defines how close should the lattice point be to the given vector.

Compute $u \leftarrow Encode(m)$.

Compute $p \leftarrow Bv$.

Signature is valid if the Euclidean distance is less than τ,

$\|Encode(m) - Bv\| < \tau$.

5-4 NTRU Signature Scheme (NSS)

The GGH signature scheme is the earliest proposal for lattice-based signature scheme, while NSS (or NTRUsign) constructed using the similar concept from the GGH scheme, with NTRU lattices (Hoffstein et al., 2001). The NSS takes the hardness of finding a short vector in NTRU lattice as the foundation of the scheme. In addition a signer, Bob, can show that $h \equiv f^{-1} * g \pmod{q}$ without leaking any information on f, which means that NSS allows Bob to show that he knows the secret key without revealing its value.

NTRU Signature Scheme

Parameters: $(N, p, q, D_{min}, D_{max})$

Key Generation

Let $f = f_0 + pf_1, g = g_0 + pg_1$,

where f_0, g_0 are fixed universal polynomials, f_1, g_1 are polynomials with small coefficient.

Compute f^{-1}, where $f^{-1} * f \equiv 1 \pmod{q}$, and $h \equiv f^{-1} * g \pmod{q}$.

Secret Key is polynomial f.

Public Key is polynomial h.

Signature Generation

Let message be polynomial $m \pmod{p}$.

Choose polynomial $w \in \mathbb{F}_w$ such that $w = m + w_1 + pw_2$, where w_1, w_2 are small polynomials.

Compute $s \equiv f * w \pmod{q}$.

Signed: (m, s).

Verification

1. Check if deviation $(s, f_0 * m)$ is between D_{min}, D_{max}. ($D_{min} \leq Dev(s, f_0 * m) \leq D_{max}$).
2. Compute $t \equiv h * s \pmod{q}$.
3. Check if deviation satisfy $D_{min} \leq Dev(t, g_0 * m) \leq D_{max}$.

If the signature passes (1) and (3), the signature is valid.

Each signing process leaks some information on the secret key, such that it is possible to compute the secret key with enough message-signature pairs. This can be solved by perturbation technique: by using a more difficult body in the signing algorithms. However, this causes the scheme to have a larger secret key size and also slowing down the signature generation. Even so, NSS does not have any security proof (Micciancio and Regev, 2009).

Table 5.1: Comparisim between Lattice-Based Schemes

Schemes	Public Key	Secret Key	Signing / Encrypt	Verification / Decrypt	Hardness
Ajtai-Dwork	$B'^{(1)}$, $M^{(1)}$	$H^{(1)}$	1 add, 1 exp, 1 mult	1 mult, 1 div	SVP
NTRU	$h^{(2)}$	$f^{(2)}$, $f_p^{(2)}$	1 mult, 1 add	4 mult, 1 add	SVP
GGH	$B^{(1)}$	$R^{(1)}$	1 mult, 1 add	2 mult	CVP
GGH Sign	$B^{(1)}$	$R^{(1)}$	1 enc, 1 mult, 1 func	1 enc, 1 mult, 1 func, 1 add	CVP
NSS	$h^{(2)}$	$f^{(2)}$	2 add, 2 mult	1 mult, 1 deviation	SVP

($^{(1)}$ in basis, $^{(2)}$ in polynomial, add = addition, mult = multiplication, div = division, exp = exponent, func = function, enc = encode)

5-5 Dihedral Group Algebra

Let p, q be distinct odd primes. Consider the dihedral group of order $2pq$ with the following group presentation:

$$D_{2pq} = \langle r, s | r^{pq} = s^2 = 1, rs = sr^{-1} \rangle.$$

Suppose r is another odd prime different from p and q. Consider the dihedral group algebra $\mathbf{Z}_r[D_{2pq}] = \{\sum_{g \in D_{2pq}} a_g g | a_g \in \mathbf{Z}_r\}$ which can be regarded as a free $\mathbf{Z}_r$-module over D_{2pq} and D_{2pq} can be viewed as an Z_r-basis for $\mathbf{Z}_r[D_{2pq}]$. The addition

and scalar multiplication are defined as follows. For any $u = \sum_{g \in D_{2pq}} a_g g$, $v = \sum_{g \in D_{2pq}} b_g g \in \mathbf{Z}_r[D_{2pq}]$ and $c \in \mathbf{Z}_r$,

$$u + v = \sum_{g \in D_{2pq}} (a_g + b_g)g,$$

and

$$cu = \sum_{g \in D_{2pq}} ca_g g.$$

Moreover, multiplication in D_{2pq} induces multiplication in $\mathbf{Z}_r[D_{2pq}]$ as

$$u \cdot v = \sum_{k \in D_{2pq}} d_k k,$$

where

$$d_k = \sum_{gh=k \in D_{2pq}} e_g f_h.$$

By these operations, $\mathbf{Z}_r[D_{2pq}]$ is an associative $\mathbf{Z}_r$-algebra with identity 1.

From the choice of p, q and r, we easily see that $gcd(r, 2pq) = 1$ and so $\mathbf{Z}_r[D_{2pq}]$ is a semisimple group algebra. According to Maschke's theorem, $\mathbf{Z}_r[D_{2pq}]$ preserve the following decomposition into direct sum of minimal ideals each generated by an orthogonal idempotent:

$$\mathbf{Z}_r[D_{2pq}] = \bigoplus_{i=1}^{s} \mathbf{Z}_r[D_{2pq}]e_i \bigoplus_{j=1}^{t} \mathbf{Z}_r[D_{2pq}]f_j.$$

Any ideal I of $\mathbf{Z}_r[D_{2pq}]$ can also be written in the form

$$I = \bigoplus_{k=1}^{s_1} \mathbf{Z}_r[D_{2pq}]e_{i_k} \bigoplus_{h=1}^{t_1} \mathbf{Z}_r[D_{2pq}]f_{j_h},$$

where $s_1 \leq s$ and $t_1 \leq t$. Let $\mu_L = \{e_1, e_2, \ldots, e_t\}$ and $\mu_N = \{f_1, f_2, \ldots, f_s\}$ be the set of all linear idempotents and the set of all nonlinear idempotents of $\mathbf{Z}_r[D_{2pq}]$, respectively. We note that e_i and f_j, for all $1 \leq i \leq t$ and $1 \leq j \leq s$, are pairwise primitive orthogonal idempotents. For all $e_i \in \mu_L$, e_i is constructed by using the linear character χ of D_{2pq}, and all $f_j \in \mu_N$ are constructed by using the non-linear character ρ of D_{2pq}. Note that $deg(\chi) = 1$ and $deg(\rho) = 2$. We further let $\mu = \mu_L \cup \mu_N$ be the set of all idempotents of D_{2pq}.

Consider

$$I = \bigoplus_{e_i \in A_L} \mathbf{Z}_r[D_{2pq}]e_i \bigoplus_{f_j \in A_N} \mathbf{Z}_r[D_{2pq}]f_j$$

where $A_L \subset \mu_L$ and $A_N \subset \mu_N$. For all $\alpha \in I$, we see that $\alpha = \sum_{i=1}^{|A_L|} \beta_i e_i + \sum_{j=1}^{|A_N} \theta_j e_j$, where $\beta_i, \theta_j \in \mathbf{Z}_r[D_{2pq}]$ for all i and j. Consider $e \in M \backslash (A_L \cup A_N)$, then we have

$$\alpha e = \sum_{i=1}^{|A_L|} \beta_i e_i e + \sum_{j=1}^{|A_N|} \theta_j e_j e = 0 + 0 = 0$$

Therefore, we have proved that

$$I = \{\alpha \in \mathbf{Z}_r[D_{2pq}] | \alpha e = 0 \text{ for all } e \in M \backslash (A_L \cup A_N)\}$$

For convenient, we write I as $I_{A_L \cup A_N}$. The length n of $I_{A_L \cup A_N}$ is defined to be $|D_{2pq}| = 2pq$. The weight of any element $u = \sum_{g \in D_{2pq}} a_g g$ is equal to $|\{a_g : a_g \neq 0\}|$ and is denoted by $wt(u)$. If the dimension of $I_{A_L \cup A_N}$ is k and minimum distance $d = d(I_{A_L \cup A_N}) = min\{wt(u) : 0 \neq u \in I_{A_L \cup A_N}\}$, then $I_{A_L \cup A_N}$ is called an $[n, k, d]$-dihedral group algebra code. We can regard $I_{A_L \cup A_N}$ as a left $\mathbf{Z}_r[D_{2pq}]$-module and so $dim(I_{A_L \cup A_N}) = 2pq - |A_L| - 2^2|A_N|$. Suppose $B = \{b_1, b_2, \ldots, b_{2pq-|A_L|-2^2|A_N|}\}$ is a basis of $I_{A_L \cup A_N}$. Arrange all elements of basis B of $I_{A_L \cup A_N}$ as rows in the following $(2pq - |A_L| - 2^2|A_N|) \times 2pq$ matrix G:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{|D_{2pq}|-|A_L|-2^2|A_N|} \end{bmatrix}$$

Note that G is a generator matrix of $I_{A_L \cup A_N}$. Thus, the parity check matrix of $I_{A_L \cup A_N}$ is H with size $2pq \times (|A_L| + 2^2|A_N|)$. The $|A_L| + 2^2|A_N|$ columns of H forms a basis of

$$I^{\perp}_{A_L \cup A_N} = I_{M \backslash (A_L \cup A_N)} = \{\alpha \in \mathbf{Z}_r | D_{2pq}] | \alpha e = 0 \text{ for all } e \in A_L \cup A_N\}$$

We note that G is a r-ary $(2pq - |A_L| - 4|A_N|) \times 2pq$ matrix, and we can define the $2pq$-dimensional q-ary lattices as follows:

$A_r(G) = \{y \in \mathbf{Z}^{2pq} : y = G^t s \text{ (mod r) for some } s \in \mathbf{Z}^{2pq-|A_L|-4|A_N|}\}$ or equivalently

$$A_r(G) = \{y \in \mathbf{Z}^{|A_L|+4|A_N|} : Hy^t = 0 \text{ (mod } r)\}.$$

Theorem 5.1. $\mathbf{Z}_r[D_{2pq}] \cong \mathbf{Z}_r^{2pq}$ *as vector space.*

Proof: Define the following mapping θ from $\mathbf{Z}_r[D_{2pq}]$ to $\mathbf{Z}_r^{2pq}$ as follows:

$$\theta\left(\sum_{i=1}^{pq} a_i r^i + \sum_{j=1}^{pq} b_j r^j s\right) = (a_1, a_2, \ldots, a_{pq}, b_1, b_2, \ldots, b_{pq})$$

Consider any two elements $\alpha = \sum_{i=1}^{pq} a_i r^i + \sum_{j=1}^{pq} b_j r^j s$, $\beta = \sum_{i=1}^{pq} c_i r^i + \sum_{j=1}^{pq} d_j r^j s \in \mathbf{Z}_r[D_{2pq}]$. Then, we see that $\alpha + \beta = \sum_{i=1}^{pq}(a_i + c_i)r^i + \sum_{j-1}^{pq}(b_j + d_j)r^j s$. Thus,

$$\begin{aligned}\theta(\alpha+\beta) &= (a_1 + c_1, \ldots, a_{pq} + c_{pq}, b_1 + d_1, \ldots, b_{pq} + d_{pq}) \\ &= (a_1, \ldots, a_{pq}, b_1, \ldots, b_{pq}) + (c_1, \ldots, c_{pq}, d_1, \ldots, d_{pq}) \\ &= \theta(\alpha) + \theta(\beta)\end{aligned}$$

Next, for any $a \in \mathbf{Z}_r$, we have $\theta(a\alpha) = a\theta(\alpha)$. Thus, θ is a linear transformation.

Suppose $\alpha = \sum_{i=1}^{pq} a_i r^i + \sum_{j=1}^{pq} b_j r^j s \in ker(\theta)$. Then $\theta(\alpha) = (0, \ldots, 0, 0, \ldots, 0)$. Hence, we have $a_1 = \cdots = a_{pq} = b_1 = \cdots = b_{pq} = 0$, and so $\alpha = 0$ which implies $ker(\theta) = \{0\}$. Clearly, θ is onto. Therefore, the isomorphic property follows directly. **Q.E.D.**

From previous theorem, we see that from now onward, we may identify an element in $\mathbf{Z}_r[D_{2pq}]$ as a group algebra element or as a vector of length $2pq$.

Corollary 1. $\mathbf{Z}_r[\langle r\rangle] \subset \mathbf{Z}_r[D_{2pq}]$. Thus, $\mathbf{Z}_r[\langle r\rangle]$ is a pq-dimensional vector space over $\mathbf{Z}_r$. Furthermore, any element in $\mathbf{Z}_r[\langle r\rangle]$ can be viewed as an element in $\mathbf{Z}_r[D_{2pq}]$ through the following identification: $(a_1, \ldots, a_{pq}) \in \mathbf{Z}_r[\langle r\rangle] \leftrightarrow (a_1, \ldots, a_{pq}, 0, \ldots, 0) \in \mathbf{Z}_r[D_{2pq}]$.

5-6 Proposed Scheme

In this section, first we will explain two family of hash functions which will be used as a random oracle in our proposed scheme.

Choose D_{2pq} with generators s and r. Fix a set $A = \{0, 1\}$, and a one-to-one mapping from s and r to 0 and 1, respectively. The hashing function $h' : A^{2pq} \to D_{2pq}$ associated to D_{2pq}, define s, r and f: To any $x = x_1 x_2 \ldots x_{2pq} \in A^{2pq}$,

$$h'(x) = f(x_1)f(x_2)\ldots f(x_{2pq}).$$

Next, we consider another function from $g : \mathbf{Z}_r[D_{2pq}] \to A^{2pq}$ which is defined as

follows:

$$g\left(\sum_{i=1}^{pq} a_i r^i + \sum_{j=1}^{pq} b_j r^j s\right) = a_1 a_2 \dots a_{pq} b_1 b_2 \dots b_{pq} \text{ (mod 2)}.$$

Clearly, $h = h' \circ g$ is a hashing function from $\mathbf{Z}_r[D_{2pq}]$ to D_{2pq}.

Let $\mathbf{G}$ be the directed Cayley graph with D_{2pq}, r, s, with directed edge between vertices v and w if and only if $w = vt$ and $t \in \{r, s\}$. Next, we can input a directed path of text x in $\mathbf{G}$, starting with identity vertex and ends with hash output $h'(x)$.

Let x and y be two distinct texts, we need to avoid collisions, that is, we need to avoid having $h'(x) = h'(y)$. Because of this, we need to avoid having two texts with the same path in $\mathbf{G}$.

We will also use another family of hash function, h_1 such that

$$h_1 : D_{2pq} \times \mathbf{Z}_{2pq} \to \mathbb{F}_2^k,$$

where k is the dimension of $I_{A_L \cup A_N}$

System Parameters

- Choose n as a product of two large distinct primes p and q.
- Form the dihedral group D_{2pq} (together with the multiplication table of D_{2pq}, directed Cayley graph and character table of D_{2pq}).
- Form the dihedral group codes $I_{A_L \cup A_N}$ (or $I^{\perp}_{A_L \cup A_N}$) with minimum distance $d \geq 2t + 1$.
- Use the hash function $h : \mathbf{Z}_r[D_{2pq}] \to D_{2pq}$ corresponds to the directed Cayley graph.

Key Generation

Choose public key to be n and H (equivalently, a basis of $I^{\perp}_{A_L \cup A_N}$).

Choose secret key to be p, q and G (equivalently, a basis of $I_{A_L \cup A_N}$).

Signing

To sign a message $m \in \mathbf{Z}_r[D_{2pq}]$, randomly choose $r' \in \mathbf{Z}_{2pq}$.

Compute hashing $h(m) = r$.

Compute second hashing $h_1(r, r') = \widehat{m}$.

Compute $\hat{m}G + e = \sigma$, where $e \in \mathbb{F}_2^n$ with $wt(e) \leq t$.

Output message-signature pair (m, σ).

Verification

Compute hashing $h(m) = r$.

Choose random r'' such that $h_1(r, r'') = \widehat{m'}$.

Compute $\sigma H = (\widehat{m'}G + e)H = \widehat{m'}GH + eH = eH$.

Decode (σH) = Decode (eH) = $\widehat{m}$.

σ is a valid signature on message m if and only if

$$\widehat{m} = h(r, r'')$$

Remarks:

1. The order of the multiplications $\widehat{m}G$, eH, σH are important as dihedral group is noncommutative.

2. The condition distance $d \geq 2t+1$ is necessary to ensure the constructed dihedral group algebra code is a t-error correcting code.

In the following, we state some security properties of the proposed scheme.

1. The first security assumption is the hardness of integer factorization (IFP), which is well-known can only be solved exponentially.

2. The basis used as a component in the secret key is typically a good basis. Algorithmically, good bases allow to efficiently solve certain instance of the closest vector problem in $I_{A_L \cup A_N}$. Thus, our second security assumption is the hardness of solving closest vector problem (CVP) in lattice.

3. Given H, to deduce the generator matrix G is equivalent to knowing all vectors in $I^{\perp}_{A_L \cup A_N}$ which are orthogonal to $I_{A_L \cup A_N}$ which turn out to be equivalent to the hardness of solving the complete decoding problem (CDP).

CHAPTER 6: CONCLUSION

In conclusion, we have successfully completed our objectives of this project. First, we have done a thorough study of the blockchain technology and the security system behind, which is signature scheme. Next, we address the potential threats of quantum computers toward the security system of blockchain technology, and the potential solutions by previous researchers. Finally we have constructed a post quantum lattice-based signature scheme which is suitable for the blockchain technology applications.

We hope that this research project can provide sufficient information and ideas for other researchers and blockchain developers. The proposed scheme has not been fully tested due to our limited knowledge in computer science, thus we hope that other researchers can further improve our signature scheme, or use it as an inspiration to construct better post quantum signature schemes for the blockchain technology.

REFERENCES

Ajtai, M., 1996. 'Generating hard instances of lattice problems', *Quaderni di Matematica* **13**, pp. 1–32.

Ajtai, M. and Dwork, C., 1997. 'A public-key cryptosystem with worst-case/average-case equivalence', *Proc. 29th Annual ACM Symp. on Theory of Computing (STOC)* pp. 284–293.

Bloomberg, 2016. 'Google cloud joins amazon, IBM in serving blockchain clients'. `https://www.datacenterknowledge.com/archives/2016/09/23/google-cloud-joins-amazon-ibm-in-serving-blockchain-clients`.

Brown, D. R. L., 2010. 'SEC 2: Recommended elliptic curve domain parameters'. `http://www.secg.org/sec2-v2.pdf`.

Buterin, V., 2015. 'A next-generation smart contract and decentralized application platform'. `https://github.com/ethereum/wiki/wiki/White-Paper`.

Chaum, D. and vanAntwerpen, H., 1990. 'Undeniable signatures', *Lecture Notes in Computer Science Crypto '89* **435**, pp. 212–216.

Deloitte, 2016. 'Blockchain: Enigma, paradox, opportunity'. `https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/Innovation/deloitte-uk-blockchain-full-report.pdf`.

Diffie, W. and Hellman, M. E., 1976. 'New directions in cryptography', *IEEE Transactions on Information Theory* **22**(6), pp. 644–654.

Drescher, D., 2017. *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, Apress, pp. 66–210.

Elgamal, T., 1985. 'A public key cryptosystem and a signature scheme based on discrete logarithms', *IEEE Transactions on Information Theory* **31**(4), pp. 469–472.

Fiat, A. and Shamir, A., 1987. 'How to prove yourself: Practical solutions to identification and signature problems', *Lecture Notes in Computer Science CRYPTO '86* **263**, pp. 186–194.

Gayathri, A., 2011. 'From marijuana to LSD, now illegal drugs delivered on your doorstep'. https://www.ibtimes.com/marijuana-lsd-now-illegal-drugs-delivered-your-doorstep-290021.

Gentry, C., 2009. A Fully Homomorphic Encryption Scheme, PhD thesis, Stanford University. crypto.stanford.edu/craig.

Gichigi, T., 2018. 'A brief history of blockchain'. https://medium.com/coinmonks/a-brief-history-of-blockchain-70c519d3053.

Goldreich, O., Goldwasser, S. and Halevi, S., 1997. 'Public-key cryptosystems from lattice reduction problems', *CRYPTO* pp. 112–131.

Goldwasser, S., Micali, S. and Rivest, R. L., 1998. 'A digital signature scheme secure against adaptive chosen-message attacks', *SIAM Journal on Computing* **17**(2), pp. 281–308.

Hill, K., 2013. 'The FBI's plan for the millions worth of bitcoins seized from silk road'. https://www.forbes.com/sites/kashmirhill/2013/10/04/fbi-silk-road-bitcoin-seizure/#270397a12848.

Hoffstein, J., Pipher, J. and Silverman, J. H., 1998. 'NTRU: A ring-based public key cryptosystem', *ANTS* pp. 267–288.

Hoffstein, J., Pipher, J. and Silverman, J. H., 2001. 'NSS: An NTRU lattice-based signature scheme', *EUROCRYPT* pp. 211–228.

IBM, 2018. 'Rewire your industry with ibm blockchain'. https://www.ibm.com/blockchain/industries.

Ibrahim, A., 2013. 'Election fraud in malaysia'. https://www.huffingtonpost.com/azeem-ibrahim/malaysia-election-fraud_b_3211954.html.

Johnson, D., Menezes, A. and Vanstone, S., 2001. 'The elliptic curve digital signature algorithm (ECDSA)', *International Journal on Information Security* **1**, pp. 36–63.

Joux, A. and Stern, J., 1998. 'Lattice reduction: a toolbox for the cryptanalyst', *Journal of Cryptology* **11**(3), pp. 161–185.

Kharpal, A., 2017. 'Bitcoin value rises over $1 billion as japan, russia move to legitimize cryptocurrency'. https://www.cnbc.com/2017/04/12/bitcoin-price-rises-japan-russia-regulation.html.

Lamport, L., 1979. 'Constructing digital signatures from a one way function'. SRI International CSL-98.

Lauter, K., 2017. 'Postquantum opportunities: Lattices, homomorphic encryption, and supersingular isogeny graphs', *IEEE Security & Privacy* **15**(4), pp. 22–27.

Lenstra, A. K., Lenstra, H. W. and Lovasz, L., 1982. 'Factoring polynomials with rational coefficients', *Mathematische Annalen* **261**(4), pp. 515–534.

Micciancio, D. and Regev, O., 2009. *Post Quantum Cryptography*, Springer, chapter Lattice-based Cryptography, pp. 147–191.

Mollin, R. A., 2003. *RSA and Public-Key Crptography*, Discrete Mathematics and its Applications, Chapman & Hall/CRC.

Nakamoto, S., 2008. 'Bitcoin: A peer-to-peer electronic cash system'. https://bitcoin.org/bitcoin.pdf/.

Peikert, C., 2016. *A Decade of Lattice Cryptography*, Now Publishers.

Peikert, C., Gentry, C., Regev, O. and Lyubashevsky, V., 2012. 'The 2nd BIU winter school: Lattice-based cryptography and applications'. https://cyber.biu.ac.il/event/the-2nd-biu-winter-school/.

Pikri, E., 2018. '7 cool blockchain projects made right here in malaysia'. https://fintechnews.my/18476/blockchain/blockchain-malaysia-projects/.

Popper, N., 2016. 'A hacking of more than $50 million dashes hopes in the world of virtual currency'. https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html.

Rabin, M. O., 1979. 'Digitalized signatures and public-key functions as intraciable as factorization'. MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-212.

Rajendran, N., 2018. 'Night of drama, confusion at 14th malaysian general elections'. https://www.todayonline.com/malaysian-ge/night-drama-confusion-14th-malaysian-general-elections.

Regev, O., 2009. 'On lattices, learning with errors, random linear codes, and cryptography', *J.ACM* **56**(6), pp. 1–40.

Rivest, R. L., Shamir, A. and Adleman, L., 1978. 'A method for obtaining digital signatures and public-key cryptosystems', *Communications of the ACM* **21**, pp. 120–126.

Schnorr, C. P., 1991. 'Efficient signature generation by smart cards', *Journal of Cryptology* **4**, pp. 161–174.

Shor, P. W., 1994. 'Algorithms for quantum computation: Discrete log and factoring', *Proc. 35th Annu. Symp. Found. Comput. Sci.* pp. 124–134.

Singhal, B., Dhameja, G. and Panda, P. S., 2018. *Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions*, Apress, pp. 1–148.

Stinson, D. R., 2006. *Cryptography: Theory and Practice*, Discrete Mathematics and its Applications, 3 edn, Chapman & Hall/CRC.

Thambyrajah, S. and Lee, T., 2018. 'M'sian halal products focus area for blockchain tech'. http://www.enterpriseitnews.com.my/msian-halal-products-focus-area-for-blockchain-tech/.

UTAR, 2018. 'UTAR iCGPA and blockchain certification system'. http://www.utar.edu.my/econtent_sub.jsp?fcatid=16&fcontentid=118227.

vanHeyst, E. and Pedersen, T. P., 1993. 'How to make efficient fail-stop signatures', *Lecture Notes in Computer Science EUROCRYPT '92* **658**, pp. 366–377.

Wallace, B., 2011. 'The rise and fall of bitcoin'. https://www.wired.com/2011/11/mf-bitcoin/.

Zuckerman, M. J., 2018. 'Malaysian gov't committee partners with korean lab to develop sharia-compliant blockchain'. https://cointelegraph.com/news/malaysian-gov-t-committee-partners-with-korean-lab-to-develop-sharia-compliant-blockchain.