ZERO-KNOWLEDGE RANGE PROOFS AND APPLICATIONS ON DECENTRALIZED CONSTRUCTIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ESRA GÜNSAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

FEBRUARY 2021

Approval of the thesis:

ZERO-KNOWLEDGE RANGE PROOFS AND APPLICATIONS ON DECENTRALIZED CONSTRUCTIONS

submitted by **ESRA GÜNSAY** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Selçuk-Kestel
Director, Graduate School of **Applied Mathematics** ______________

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** ______________

Assoc. Prof. Dr. Murat Cenk
Supervisor, **Cryptography, METU** ______________

Examining Committee Members:

Assoc. Prof. Dr. Zülfükar Saygı
Mathematics, TOBB ETU ______________

Assoc. Prof. Dr. Murat Cenk
Cryptography, METU ______________

Assoc. Prof. Dr. Oğuz Yayla
Cryptography, METU ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ESRA GÜNSAY

Signature :

ABSTRACT

ZERO-KNOWLEDGE RANGE PROOFS AND APPLICATIONS ON DECENTRALIZED CONSTRUCTIONS

Günsay, Esra
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Murat Cenk

February 2021, 54 pages

Appropriate, effective, and efficient use of cryptographic protocols contributes to many novel advances in real-world privacy-preserving constructions. One of the most important cryptographic protocols is the zero-knowledge proofs. The zero-knowledge proofs have recently gained the utmost importance in terms of decentralized systems, especially in the context of privacy. In many decentralized systems, such as electronic voting, e-cash, e-auctions, or anonymous credentials, the zero-knowledge range proofs are used as the building blocks. In this thesis, we examine, summarise and compare range proofs based on zero-knowledge proofs, and examine their applications in decentralized systems such as distributed ledgers, confidential assets and smart contracts. We also, investigate different basis of OR-proofs and compare the efficiency of different basis approaches. To this end, we have modified the Mao's range proof [31] to base-3 with a modified OR-proof [16]. For each basis, we derive the number of computations in modulo exponentiations and the cost of numbers exchanged between parties. Then, we have generalized these costs for base-u construction. At the end of these comparisons, we observe that comparing the number of computations in modulo exponentiations with other base approaches, the base-3 approach is 5.5% more efficient. In addition, comparing the cost of numbers exchanged between prover and verifier, base-3 approach is 7% more efficient than other base approaches.

Keywords: zero knowledge proof, range proof, blockchain, decentralized systems, commitment scheme, sigma protocol.

ÖZ

SIFIR BİLGİ ARALIK ISPATLARI VE DAĞITIK SİSTEMLERDEKİ UYGULAMALARI

Günsay, Esra
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Murat Cenk

Şubat 2021, 54 sayfa

Kriptografik protokollerin uygun, etkili ve verimli kullanımı, gerçek dünyadaki gizlilik koruma temelli birçok yeni ilerlemeye katkıda bulunur. En önemli kriptografik protokollerden biri de sıfır bilgi ispatlarıdır. Sıfır bilgi ispatları, özellikle son zamanlarda gizlilik bağlamında, merkezi olmayan sistemler açısından önemli hale gelmiştir. Elektronik oylama, e-nakit, e-açık artırmalar veya anonim kimlik bilgileri gibi birçok merkezi olmayan sistemde, sıfır bilgi aralığı ispatları yapı taşı olarak kullanılmaktadır. Bu tezde, sıfır bilgi ispatlarına dayalı aralık ispatları açıklanıp karşılaştırıldı ve dağıtılmış defterler, gizli varlıklar ve akıllı sözleşmeler gibi merkezi olmayan sistemlerdeki uygulamaları incelendi. Ayrıca, farklı tabanlardaki aralık ispatları incelenip verimlilikleri karşılaştırıldı. Bu amaçla, Mao'nun aralık ispatı [31] 3 bazında modifiye edildi. Bunun için OR şeması [16] 3 tabanında inşa edildi. Her bir değişik taban için gerekli modülar formda üst alma işlemi maaliyeti ve partiler arası değiş tokuş edilen sayıların maaliyetleri hesaplandı. Daha sonrasında, bu hesaplamalar u-tabanı için genelleştirildi. Bu karşılaştırmaların sonunda, modüler üstelleştirme hesaplamaların maliyetinin diğer baz yaklaşımlarıyla karşılaştırıldığında, taban-3 yaklaşımının yaklaşık $\%5.5$ daha verimli olduğunu gözlemlendi. Ayrıca, kanıtlayıcı ve doğrulayıcı arasında değiş tokuş edilen sayıların maliyetini karşılaştırdığımızda, taban-3 yaklaşımı diğer taban yaklaşımlarına göre yaklaşık $\%7$ daha verimli olduğu da görüldü.

Anahtar Kelimeler: sıfır bilgi ispatı, aralık ispatı, blokzincir, dağıtık sistemler, taahhüt şemaları, sigma protokolü.

To my family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Murat Cenk for his motivation and support during the development and preparation of this thesis.

I would also like to thank my examining committee members Assoc. Prof. Dr. Oğuz Yayla and Assoc. Prof. Dr. Zülfükar Saygı for their time they spared for me.

I am grateful for my colleagues and friends Kübra Kaytancı, H. Bartu Yünüak, Pelin Çiloğlu, Cem Yavrum and Mustafa Kütük for their friendship and making institude such great and warm place for me.

I deliver my utmost thanks and appreciation to my dear fiancé S. Serdar Kozat, for encouraging me to a bright future with his support and love.

I would like to express my sincere appreciations to my family. They are always with me whenever I need their love and support.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DApp	Decentralized Application
DCR	Decisional Composite Residuosity
DDH	Decisional Diffie-Hellman
DHP	Diffie-Hellman Problem
DLP	Discrete Logarithm Problem
ECC	Elliptic Curve Cryptography
HVZK	Honest-Verifier Zero-Knowledge
NIZK	Non-interactive Zero-Knowledge
PoK	Proof of Knowledge
q-SDH	q-Strong Diffie-Hellman
SDH	Strong Diffie-Hellman
sHVZK	Special Honest-Verifier Zero-Knowledgeness
ZKP	Zero-Knowledge Proof

CHAPTER 1

INTRODUCTION

Zero-knowledge proofs are one of the most important building blocks of many privacy-preserving systems. The basic principle of the zero-knowledge protocol construction is to prove the information you know to someone else without revealing any information about the secret itself. The method of proving this hidden knowledge is often the output of a cryptographic function. There exist various types of zero-knowledge proofs used in the area of computer science and applied mathematics.

Among these various different proofs, one very important them is the zero-knowledge range proofs (ZKRPs), which prove that an integer lies in an interval without revealing the integer itself. There are numerous areas of usage of ZKRPs in the real-world [3] [12] [11]. One basic example as widely popular is to assume one person wants to prove that she is over 18 in order to validate that he/she can consume an age-restricted service, say he/she wants to vote. In this example, a cryptographic ZKRP algorithm is used to prove this without revealing the age.

Another common example, especially in the banking industry, assume a party wants to transfer x (which can be money, services, etc.). To do so, the party needs to prove that the amount he/she tries to send is positive. Otherwise, this transaction works in opposite direction. The problem is how one person can prove that he has enough money without revealing the transaction amount. At this point, the ZKRP goes into the work to solve the problem.

Due to this importance, there exist many up-to-date studies to explain and compare existing range proofs [36][19][26][15]. Apart from these studies, the motivation of this thesis is to focus on underlying cryptographic primitives, explain some more recent schemes and decentralised applications to lead the establishment of new decentralized systems, and the improvement of existing systems. Also, in this study, we examine different base approaches, analyse the complexity for these basis by aiming to find most efficient approach.

As we mentioned, there are numerous techniques to achieve range proofs. In this thesis, we separated them into 3 main categories:

1. Strong RSA problem based method,
2. Diffie-Hellman problem method,
3. Discrete logarithm problem based method.

For each catogory, we investigate one or two leading examples, which can be useful in context of decentralised applications (DApps), or different base approach.

CHAPTER 2

PRELIMINARY TO THE SUBJECT

In this chapter, some of the basic primitives, notations, and definitions will be introduced to provide the reader with sufficient theoretical background about the zero-knowledge range proofs.

There are 5 main sections in this chapter. Firstly notation of the thesis will be introduced. Secondly, some algebraic backgrounds will be summarized, and then bilinear pairings will be defined. Due to its important role in ZKRP systems, a comprehensive explanation of commitment schemes will be given. The last section will explain digital signatures and define one particular digital signature due to its role in the next chapters.

2.1 Notations

Let $\mathbb{G}$ and $\mathbb{Q}$ be two cyclic groups of order p and q respectively, where both p and q are large primes. Let $\mathbb{Z}_p$ denotes residue class ring of modulo p. By both $r \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, and $r \in_R \mathbb{Z}_p^*$, we denote that r is randomly chosen over $\mathbb{Z}_p^*$. To notate the protocols, we choose to use the Boudot's representation as $PK_{type}(x : \mathcal{R}(x))$, which denotes proof of x when $x \in \mathcal{R}(x)$.

Also, some related but specific definitions and notations will be given when needed in the following chapters.

2.2 Finite Field Cryptography

In public-key cryptography, many applications are constructed over algebraic structures. It is crucial to know these bases to get the cryptographic protocols.

A set $\mathbb{G}$ with a binary operation $*$ is called as a *group* and shown as $(\mathbb{G}, *)$. A group has 4 properties to hold:

1. $\forall a, b \in (\mathbb{G}, *)$, $a * b \in (\mathbb{G}, *)$, and uniquely defined.
2. $\forall a, b, c \in (\mathbb{G}, *)$, $a * (b * c) = (a * b) * c$.
3. $\exists e \in (\mathbb{G}, *)$ s.t $a * e = e * a = a, \forall a \in (\mathbb{G}, *)$.
4. $\exists a^{-1} \in (\mathbb{G}, *)$ s.t $a * a^{-1} = a^{-1} * a = e, \forall a \in (\mathbb{G}, *)$.

A set with at least two binary operations $(+, *)$ is called as *field* and satisfies:

1. $(\mathbb{F}, +)$, needs to be a commutative group, and its identity element is $I_+ = 0$.
2. $(\mathbb{F}^*, *)$, needs to be a commutative group, and its identity element is $I_* = 1$.
3. $\forall a, b, c \in \mathbb{F}$, $a * (b + c) = a * b + a * c$.

Based on these explanations, a *finite field* is a field with a finite number of elements. Now we will give some definitions on finite fields.

Definition 2.2.1. (**RSA assumption**) Let n be a RSA modulus, and x be a RSA exponent. *RSA assumption* states that given element $s \in \mathbb{Z}_n^*$ it is hard to find such m providing:

$$m^x = s \ (\text{mod}\ n).$$

Definition 2.2.2. (**Strong RSA assumption**) Let n be a composite number which is $n = pq$. p and q are t-bit random primes and let $\tau \in \mathbb{Z}$ be a security parameter. For a secret $s \in \mathbb{Z}_n^*$, *Strong RSA assumption* states that it is hard to find such x and m providing:

$$m^x = s \ (\text{mod}\ n),$$

where $x \neq \pm 1$.

Definition 2.2.3. (**Discreate Logarithm problem**) Let $\mathbb{G}$ be a multiplicative cyclic group and g be a generator over $\mathbb{G}$ and h be an element of $\mathbb{G}$. *Discrete Logarithm problem* states that it is hard to find such x providing:

$$g^x = h.$$

2.3 Bilinear Pairings

In pairing-based cryptography, bilinear maps are a special form of pairings.

Definition 2.3.1. (bilinear map)[5] Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups with sufficiently large order p..

A bilinear pairing $bp := (\hat{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is a map:

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T,$$

with satisfying all the following requirements:

- **bilinearity** Let $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$. $\forall a, b \in \mathbb{Z}$,

  $$\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}.$$

- **non-degeneracy** non-degeneracy means map should be non-trivial. $\forall u \in \mathbb{G}_1,$ different than identity element of $\mathbb{G}_1,$ there exist $v \in \mathbb{G}_2$ s.t.

  $$\hat{e}(u, v) \neq 1.$$

  Similarly, $\forall v \in \mathbb{G}_1,$ different than identity element of $\mathbb{G}_2,$ there exist $u \in \mathbb{G}_2$ s.t.

  $$\hat{e}(u, v) \neq 1.$$

- **efficient computability** operations within the group and membership decisions in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ should be efficiently computable.

Here *order* might be a prime composite number. Usually, *prime orders* are preferred. It is also correct to write groups $\mathbb{G}_1$ and $\mathbb{G}_2$ additively, this time definition becomes $\hat{e}(au, bv) = \hat{e}(u, v)^{ab}$ which is still correct.

Weil and Tate pairings [25] are examples of pairings where the above properties hold an elliptic curve over $\mathbb{G}_1$ and finite field $\mathbb{G}_2$.

Definition 2.3.2. (**bilinear Diffie-Hellman Problem**) Let $u \in \mathbb{G}_1$, for given $u \in \mathbb{G}_1$ u^a, u^b, u^c one must be able to find u^{ab} then compute $\hat{e}(u^{ab}, u^c) = \hat{e}(u, vu)^{abc}$.

q-Strong Diffie-Hellman Problem is another important tool for the following proofs. Boneh and Boyen proposed it for the first time related to Boneh-Boyen short signature scheme [6].

Definition 2.3.3. (**q-Strong Diffie-Hellman Problem (q-SDH)**) [42]

Let $bp := (\hat{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be a bilinear map. $\mathbb{G}_1$ and $\mathbb{G}_2$ has order p and g be the generator in $\mathbb{G}_1$. Then for any random chosen $x \in \mathbb{Z}_p$, given $g_1, g_1^x, g_1^{x^2} \ldots g_1^{x^q}$ computing a pair $g_1^{\frac{1}{x+c}}$, where $c \in \mathbb{Z}_p$.

2.4 Commitment Schemes

Commitments are important building blocks due to their major role in many cryptographic applications such as zero-knowledge proofs or secure multiparty computations. These schemes were introduced in 1998, by Brassard, Chaum $\&$ Crépeau [7].

Definition 2.4.1. (Commitment Scheme) A commitment scheme a deterministic polynomial-time algorithm, and a commitment is $\{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^*$ where k is a security parameter.

A commitment scheme has 3 polynomial algorithms (Gen, Com, Open) defined as follows:

Gen: Public commitment key c_{pub} is generated $c_{pub} \leftarrow \{1^k\}$.

Com: Using public commitment key c_{pub} it generates the commitment **c**.

Open: It takes randomness **r**, message **m**, and opens the committed value and reveals the message.

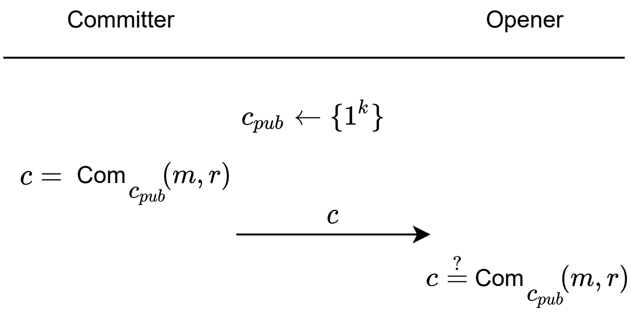

Committer
Opener
$c_{pub} \leftarrow \{1^k\}$
$c = \mathsf{Com}_{c_{pub}}(m, r)$
c
$c \overset{?}{=} \mathsf{Com}_{c_{pub}}(m, r)$

Figure 2.1: Commitment scheme workflow.

A commitment scheme has 2 stages, namely *committing* and *revealing*, as shown in figure 2.1. Here 1^k is the security parameter. At the committing stage, using the deterministic **Com** algorithm sender hides the message m. At the end of this step, both sender and receiver have the same output. The revealing stage is non-interactively occurs. In the revealing stage, the sender sends the random value d, and message m to the receiver. The receiver uses the **Com** algorithm and checks the equality of the new and previous commitments. Depending on the result, it accepts or rejects.

As a security requirement, a commitment scheme should satisfy hiding and binding properties.

Hiding: For any adversary $\mathcal{A}$, it is computationally infeasible to find $Com(m_0) = Com(m_1)$, when $m_0 \neq m_1$. This property assures that given committed value c, adversary cannot learn anything about message m. For all sufficiently large λ:

$$Pr\big[b = b^{'} \mid c_{pub} \leftarrow \textbf{Gen}\{1^\lambda\}; b \leftarrow \{0,1\};$$
$$(com, d) \leftarrow \textbf{Com}(c_{pub}, m_b) : b^{'} \leftarrow \mathcal{A}(com)\big] < \tfrac{1}{2} + negl(\lambda).$$

If a probabilistic polynomial-time (PPT) adversary is in the case, then the commitment scheme is called *computationally hiding*. Otherwise, considering the adversary has unlimited power sources, the scheme is called *information-theoretically hiding*.

It is called *statistically binding* commitment scheme if for any adversary having unlimited sources and for sufficiently large λ following holds:

$$Pr\big[b = b^{'} \mid c_{pub} \leftarrow \textbf{Gen}\{1^\lambda\}; b \leftarrow \{0,1\};$$
$$(com, d) \leftarrow \textbf{Com}(c_{pub}, m_b) : b^{'} \leftarrow \mathcal{A}(com)\big] < \tfrac{1}{2} + negl(\lambda).$$

It is called a *perfectly binding* commitment scheme if for any adversary having unlimited sources and for sufficiently large λ following holds:

$$Pr\big[b = b^{'} \mid c_{pub} \leftarrow \textbf{Gen}\{1^{\lambda}\}; b \leftarrow \{0,1\};\\ (com, d) \leftarrow \textbf{Com}(c_{pub}, m_b) : b^{'} \leftarrow \mathcal{A}(com)\big] = \tfrac{1}{2}.$$

Binding: it is infeasible to find two different openings from one committed value which guarantees committer cannot forge the system even if she changes her mind. More formally, for any adversary $\mathcal{A}$:

$$Pr\big[c_{pub} \leftarrow \textbf{Gen}\{1^{\lambda}\}; (com, m_0, m_1, d_0, d_1) \leftarrow \mathcal{A}(c_{pub}) :\\ m_0 \neq m_1 \textit{ and } \textbf{Open}(c_{pub}, com, d_0, m_0) = \textbf{Open}(c_{pub}, com, d_1, m_1) = 1\big] \leq negl(\lambda).$$

If PPT adversary is in the case, then the commitment scheme is called *computationally binding*. Otherwise, considering the adversary has unlimited power sources, the scheme is called *information-theoretically binding*.

It is called *statistically binding* commitment scheme, if for any sender having unlimited sources and for sufficiently large λ following holds:

$$Pr\big[c_{pub} \leftarrow \textbf{Gen}\{1^{\lambda}\}; (com, m_0, m_1, d_0, d_1) \leftarrow \mathcal{A}(c_{pub}) :\\ m_0 \neq m_1 \textit{ and } \textbf{Open}(c_{pub}, com, d_0, m_0) = \textbf{Open}(c_{pub}, com, d_1, m_1) = 1\big] \leq negl(\lambda).$$

It is called a *perfectly binding* commitment scheme, if for any sender having unlimited sources and for sufficiently large λ following holds:

$$Pr\big[c_{pub} \leftarrow \textbf{Gen}\{1^{\lambda}\}; (com, m_0, m_1, d_0, d_1) \leftarrow \mathcal{A}(c_{pub}) :\\ m_0 \neq m_1 \textit{ and } \textbf{Open}(c_{pub}, com, d_0, m_0) = \textbf{Open}(c_{pub}, com, d_1, m_1) = 1\big] = 0.$$

Definition 2.4.2. (Bit commitment scheme) Bit commitment scheme is a function $\{0,1\} \times \{0,1\}^* \rightarrow \{0,1\}^*$. In a language $\mathcal{L}$ a bit commitment scheme is a protocol user chooses a bit b and a random element r and produce a commitment $Com(b, r)$. In a bit commitment scheme following properties must be satisfied:

1. It would be computationally infeasible to reveal a b for a given $Com(b, r)$.

2. $Com(b, r)$ must be openable simply with revealing r by the committer.

3. It would be computationally infeasible to find an $r^{'}$ s.t. $Com(1, r^{'}) = Com(0, r^{'})$

Definition 2.4.3. (Homomorphic commitment scheme) Let $Comm$ be a commitment, and m,$m^{'}$ be two messages, r,$r^{'}$ two randomness values. A commitment scheme $Com(m, r)$ is said to be homomorphic if in the case of multiplying two commitments result is a new commitment that includes both messages under operation.

$$Com(m, r)Com(m^{'}, r^{'}) = Com(mm^{'}, rr^{'})$$

Now, we will move on with some particular commitment schemes, which will be used constructing range proof schemes.

2.4.1 Pedersen Commitment

The idea of Pedersen commitment with perfectly hiding and computationally binding properties is presented for the firts time in [33] [34]. The security of the scheme is based on the hardness of the discrete logarithm problem (DLP).

In the setup phase receiver picks uniformly random primes p and q with $p \mid (q - 1)$. Suppose $\mathbb{G}$ be the subgroup of $\mathbb{Z}_p^*$. Receiver picks an element h generator $g, \in \mathbb{G}$ randomly where $\log_g h$ is unknown.

In the committing phase, to commit a secret $x \in \mathbb{Z}_q^*$, she computes $Com(x, r) = g^x h^r$. Opening phase is quite similar, committer reveals x and corresponding r for opener to compute $Com(x, r) = g^x h^r$ to check its correctness.

- **perfectly hiding** It is computationally infeasible to reveal x for a given $Com(x, r)$.
- **binding** This property directly holds due to DLP assumption.

2.4.2 Fujisaki-Okomoto Commitment

Fujisaki-Okomoto Commitment scheme is firstly presented in [21]. In 2002, it was revisited by Damgard and Fujisaki [17]. It has statistically hiding and computationally binding properties. Security of the scheme is based on the hardness of the *strong-RSA assumption*.

In the setup phase 3 elements are generated n, h, g. Uniformly random primes p and q are chosen and $n = p * q$ is computed. Quadratic resudue modulo n generated by generator $h \leftarrow$ **Gen**(1^λ). g is an element of subgroup generated by h. Finally, a random element r is chosen from the set $\mathbb{Z}_{2^{N+\lambda}}$.

Now the commitment is of the form $Com(x, r) = g^x h^r \pmod{n}$

- **perfectly hiding** Since random element r is used in the commit phase, the result of the commitment scheme is a random group element.
- **binding** It is computationally infeasible to find two openings due to *strong-RSA assumption.*

2.5 Digital Signatures

A digital signature scheme is formed with 3 main functions (Gen, Sign, Verify), key generation, signing, and verifying, respectively. Gen algorithm outputs private-public key pairs $(k_{sk}k_{pk})$. Sign algorithm generates the signature σ by using k_{sk}. Verify algorithm checks the correctness of the signature by using corresponding k_{pk} The message is signed with the private key and verified with the corresponding public key.

2.5.1 Boneh-Boyen Signature Scheme

As provided in [24], the signature scheme has 3 main functions (Gen, Sign, Verify), respectively key generation, signing, and verifying algorithms.

Gen: Random element $x \in_R \mathbb{Z}_p^*$ chosen and set as secret key. Then public key set as $y \leftarrow g^x$.

Sign: Signing process is hold as $\zeta \leftarrow g^{\frac{1}{x+m}}$.

Verify: To verify the given signature following should be checked $\hat{e}(\zeta, y.g^m) = \hat{e}(g, g)$.

CHAPTER 3

PROOF SYSTEMS AND PROTOCOLS

A *proof system* is a workflow between two parties, namely the prover and the verifier. Usually, it has 3 phases: commit phase, challenge phase, and response phase.

A *cryptographic protocol* is a construction performing security related functions using cryptographic techniques such as encryption, hashing, signing, etc.

Since this thesis' main topic is range proofs, due to their important property, so-called zero-knowledgeness, we explain zero-knowledge proofs with their security definitions. Since range proofs are a particular kind of proof of knowledge (PoK) protocols, it is also important to get the idea of the proof of knowledge.

In this sense, we are going to use interactive or non-interactive proofs to classify types of the range proofs in the next chapters.

Also, since many of the range proofs are based on Σ-protocol, we explain them with compositions of them at the end of this chapter in detail.

3.1 Interactive Proof Systems

The idea of interactive proofs was introduced in 1985, by Goldwasser, Micali, and Rackoff for cryptographic applications [23]. Interactive proofs take place between two parties: prover and verifier. These parties are interactive Turing machines that can send and receive messages. In most cases, we assume prover has infinite computer power while verifier is a polynomial-time computer, which means his computations are in bounded-error probabilistic polynomial time (BPP). Verifier sends some random challenge, namely x, to the verifier and wants to convince that x in a language, namely $\mathcal{L}$. Prover, responds to the challenge after computing a polynomial-time function for x. Depending on the result/response verifier decides whether it accepts or rejects the proof, as shown in the figure 3.1.

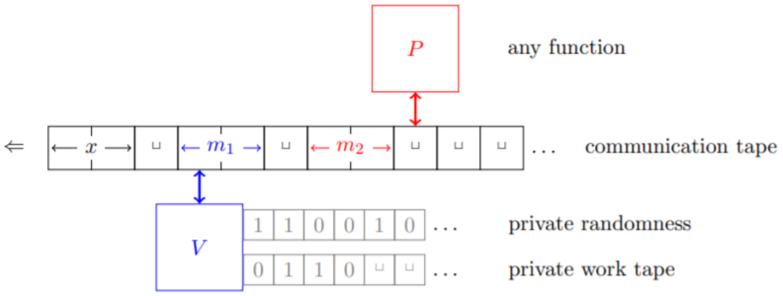


Figure 3.1: Interactive proof system [1].

Definition 3.1.1. (Interactive Proof) For the proof system pair (P, V), the statement in language $\mathcal{L} \subset \{0,1\}^*$ system accepts x in time t is

$$Pr_t\big[(V \leftrightarrow P) \quad accepts \quad x\big] = Pr_t\big[(V \leftrightarrow P)(x, r) = \quad accepts\big],$$

where verifier's private randomness r is generated uniformly. It should satisfy the following 2 properties:

Completeness: For every true claim, verifier needs to be convinced.

Soundness: It should be computationally infeasible to convince the verifier for a false claim.

3.2 Proof of Knowledge Systems

In a secure computation, PoK systems are used as building blocks in almost every protocol. The impacts of underlying PoKs on the degree of security and efficiency of the system cannot be ignored.

In the PoK systems given a binary relation set $\mathcal{R} = (x, w)$ for any x and witness set $w(x)$, membership can be tested in polynomial time. It is a type of proof of membership. The witness set is all w, providing $(x, w) \in \mathcal{R}$.

Definition 3.2.1. (Proof of Knowledge) The relation $\mathcal{R}$ with knowledge error κ where $\kappa : \{0,1\}^* \rightarrow [0,1]$ is a function, a proof of work system is a pair (P, V), which satisfies the followings:

Completeness: $\forall(x, w) \in \mathcal{R}$

$$Pr\big[(V \leftrightarrow P) \quad accepts \quad x\big] > 1 - \kappa(x),$$

which means if $(x, w) \in \mathcal{R}$, then verifier always accepts.

Soundness: $\exists E$ as an extractor which is a probabilistic oracle machine, for a constant $c > 0, \forall P^{'}, \quad \forall x, \quad \forall w^{'}$ satisfies that:

If

$$Pr\big[(V \leftrightarrow P^{'})(x, w^{'} \quad accepts\big] > \kappa(x),$$

then outputting string w by the extractor E $(x, w) \in \mathcal{R}$ can be done in steps bounded by:

$$\frac{|x|^c}{\epsilon(x) - \kappa(x)},$$

and the probability is shown as:

$$Pr\big[(V \leftrightarrow P^{'})(x, w^{'} \quad accepts\big] < Pr\big[(E(x, w^{'}; P^{'} \in w(x))\big] + \kappa(n).$$

A well-known example of this protocol is knowledge of a discrete logarithm. Let $\mathbb{G}$ be a group of prime order q. Let x is a common input and $x = g^w$ when $g \neq 1$

Protocol workflow is depicted in the figure 3.2.

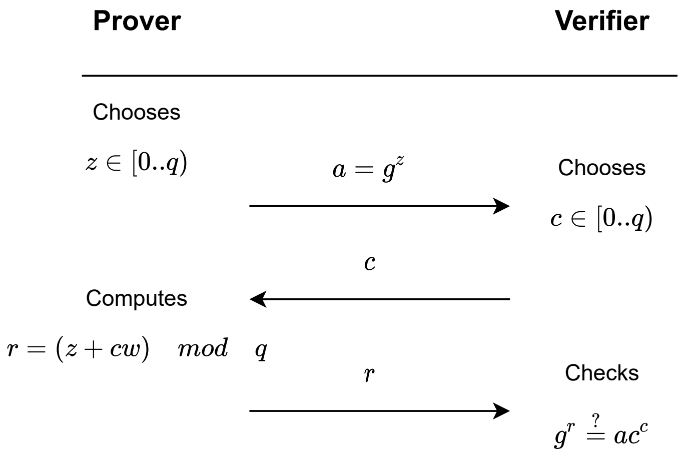


Figure 3.2: Proof of Knowledge construction workflow.

3.3 Zero-Knowledge Proofs

The class non-deterministic polynomial-time (NP) problems are those, assuming prover has infinite computing power, and the verifier is polynomially bounded. So in this set of decision problems, the "yes" answer can be verified with proofs in a polynomial time by a deterministic Turing machine. We know that any NP problem instance can be proven to be true in zero-knowledge [22].

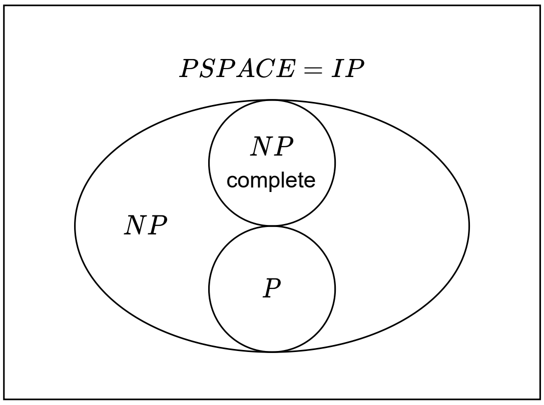


Figure 3.3: Space scheme.

The class of IP denotes the class of problems solvable by an interactive proof system. The class of $PSPACE$, on the other hand, denotes the class of problems that can be solvable by a Turing machine using a polynomial amount of space. We direct the reference [41], for proof of the claim of $PSPACE = IP$. Assuming a prover with unlimited power wants to convince a polynomially bounded verifier that a statement is valid. It is quite similar to interactive proof systems. It holds completeness and soundness properties. As a difference, in addition, zero-knowledgeness should be satisfied.

Definition 3.3.1. (Zero-Knowledge) For the interactive protocol system pair (P, V), the statement in language $\mathcal{L} \subset \{0,1\}^*$, if for all polynomially bounded verifier V' there exist a probabilistic expected polynomial time simulator $S_{V'}$,

$$x \in \mathcal{L}, \quad \big[(V' \leftrightarrow P)\big](x, w) \text{ and } \big\{S_{V'}(x)\big\}.$$

it is said that interactive proof is zero-knowledge.

Definition 3.3.2. (Honest Verifier Zero-Knowledge) For the interactive protocol system pair (P, V), the statement in language $\mathcal{L} \subset \{0,1\}^*$, if there exist an honest probabilistic expected polynomial time simulator $S_{V'}$,

$$x \in \mathcal{L}, \quad \big[(V \leftrightarrow P)\big](x, w) \text{ and } \big\{S_V(x)\big\}.$$

it is said that interactive proof is zero-knowledge.

A *simulator* S is a polynomial-time probabilistic machine for an interaction of a verifier and a prover if for every element in the language L the output of simulator is the information that the verifier possibly have gained, namely a *transcript*).

It is clear that the idea of zero-knowledge is related to the concept of a simulation. The relation of the set of valid transcripts and the set of possible simulations specifies the security.

An interactive proof system (P, V) is *computationally zero-knowledge* if the set of simulations and the set of real-world valid transcripts are computationally indistinguishable. While ξ is an auxiliary symbol:

$$S_{V'}(x) \sim_c (P, V^{'}(\xi))(x).$$

An interactive proof system (P, V) is said to be *perfect zero-knowledge* if the set of simulations and real-world valid transcripts are identical.

$$S_{V'}(x) \sim_p (P, V^{'}(\xi))(x).$$

When the difference between these sets is just a small statistical distance, it is called *statistical zero-knowledge.*

$$S_{V'}(x) \sim_s (P, V^{'}(\xi))(x).$$

A zero-knowledge proof is used to be sure that the malicious parties do not cheat. This proving process may take several steps. Depending on the response, at each step verifier decides whether it accepts or rejects the proof. It should satisfy the following 2 properties:

Completeness: For every true claim verifier needs to be convinced.

Soundness: It should be computationally infeasible to convince the verifier for a false claim.

Zero-knowledge: At the end of the process prover reveals nothing but her claim.

Arbitrary zero-knowledge proofs of $\mathcal{NP}$ statements are usually considered as somehow expensive and therefore not efficient enough for protocols. Although this is true still there exist many languages having extraordinary efficient zero-knowledge proofs.

3.4 $\Sigma-$Protocols

$\Sigma-$ protocols are special types of interactive 3-move honest verifier zero-knowledge proofs (HVZK). The first movement is usually a committed value sent by the prover. The second movement is by the verifier, and usually, it is a large enough uniformly random challenge. The third movement comes from the prover to aim that the verifier will be able to run a proof of knowledge with some specific steps.

In most of the protocols, non-interactive proofs are preferred because they do not take direct input from the verifier. This problem can be passed simply by using Fiat-Shamir transform, which hashes the message and the prover's statement. Some of these protocols are considered very efficient in applications. Here provide the necessary definitions:

Definition 3.4.1. (Σ-Protocol) For the protocol system pair (P, V), a Σ-Protocol for the relation $\mathcal{R}$, is of the following form:

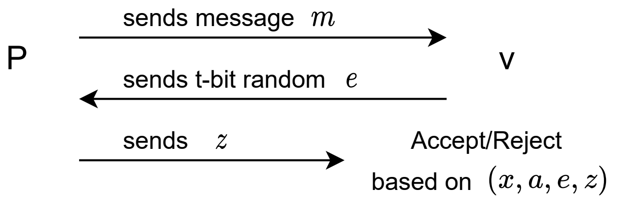


Figure 3.4: 3-move HVZK.

Both P and V have x as common input, P has private input w, where $(x, w) \in \mathcal{R}$. A typical Σ-protocol needs to achieve 3 security parameters. These are (perfect) completeness, special soundness, and Special honest-verifier zero-knowledgeness (sHVZK).

Completeness: If $(x, w) \in \mathcal{R}$ then the Verifier always accepts.

Special soundness: Let a, e, z and $a, e^{'}, z^{'}$ be two accepting transcripts with $e \neq e^{'}$. Then there exists a PPT knowledge extractor E, and always there exists a probabilistic polynomial time knowledge extractor that can output a witness w satisfying $(x, w) \in \mathcal{R}$.

sHVZK: There exists a PPT simulator $\mathcal{S}$, for any given input x, and $t-$bit random challenge e, it outputs the (x, a, e, z) with the probability distribution of outputting it between honest P and V.

As an example, *Schnorr's Protocol for discrete logarithm* can be given as we depicted in the figure 3.5.

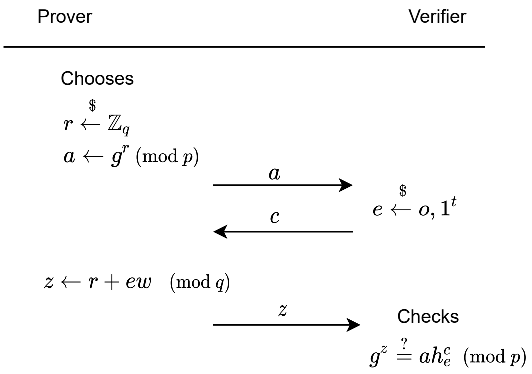

Prover
Verifier
Chooses
$r \overset{\$}{\leftarrow} \mathbb{Z}_q$
$a \leftarrow g^r \pmod p$
a
$e \overset{\$}{\leftarrow} o, 1^t$
c
$z \leftarrow r + ew \pmod q$
z
Checks
$g^z \overset{?}{=} ah_e^c \pmod p$

Figure 3.5: Schnorr's Protocol for Discreate Log.

As common input both parties have (p, q, g, h). Here p and q are primes. Prover has a value $w \in \mathbb{Z}_q$ and wants to convince the Verifier that she knows $h = g^w$. This is a 3-step protocol.

1. First prover chooses uniformly a random r and computes $a = g^z \bmod p$, and then sends a to the Verifier.

2. Verifier chooses a t-bit random challenge e, where $2^t < q$.

3. Prover computes $z = (r + ew) \bmod q$ sends z to Verifier to check weather $g^z = ah^e \bmod p$ holds. This equation holds if only $\text{ord}(g) = \text{ord}(h) = q$.

It is easy to see the completeness of the protocol:

$$\text{Since } z = r + ew, \quad g^z = g^{r+ew} = g^r(g^x)^e = ah^e.$$

Now we will examine the case that the Verifier sends two different challenges $e, e^{'}$. Then, Prover computes two different values $z = (r + ew)$ and $z^{'} = (r + e^{'}w)$ in mod q. This will lead computing $g^z = ah^e$ mod p and $g^{z^{'}} = ah^{e^{'}}$ mod p.

For the soundness property, it needs to answer two different challenges correctly so that it should be hold that:

$$g^z.h^{-e} = g^{z^{'}}.h^{-e^{'}},$$

$$z + w.(-e) = z^{'} + w.(-e^{'}).$$

So that we get:

$$w \longleftarrow \frac{z - z^{'}}{e - e^{'}}.$$

It is known that $e \neq e^{'}$. Also, $z, z^{'}, e, e^{'}$ are all known by the prover. Hence, the value of w can be recovered by the prover. Then this protocol has *special soundness property* and it is said to be proof of knowledge. Note that fraction fails in the situation of $e = e^{'}$, which happens probability of 2^{-t}, so the error probability is 2^{-t}.

3.4.1 Compositions of Σ-Protocol

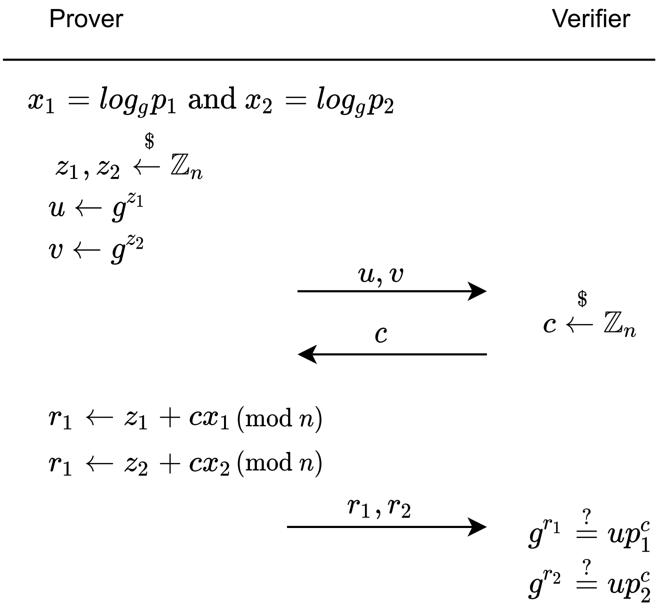
Combining the existing protocol for achieving different aims results compositions. There are many compositions such as AND, parallel, EQ, NEQ, OR compositions; however, in the context of this thesis, only AND and OR compositions will be examined by sampling the Schnorr protocol. We will define these compositions.

3.4.1.1 AND-Composition

AND-Composition is a special form of parallel composition. It checks that two given statements are correct individually. Suppose $\mathcal{R}_1$ and $\mathcal{R}_2$ be two relations in language $\mathcal{L}$. Prover now wants to prove that:

For public keys p_1 and p_2, prover is able to compute $x_1 = log_g p_1$, $x_2 = log_g p_2$ respectively. To prove these statements, two $\Sigma-$protocols work in parallel.

The figure 3.6shows the workflow of the AND-composition.



Prover

Verifier

$x_1 = log_g p_1$ and $x_2 = log_g p_2$

$z_1, z_2 \overset{\$}{\leftarrow} \mathbb{Z}_n$

$u \leftarrow g^{z_1}$

$v \leftarrow g^{z_2}$

$\xrightarrow{u, v}$

$c \overset{\$}{\leftarrow} \mathbb{Z}_n$

$\xleftarrow{c}$

$r_1 \leftarrow z_1 + cx_1 \pmod{n}$

$r_1 \leftarrow z_2 + cx_2 \pmod{n}$

$\xrightarrow{r_1, r_2}$

$g^{r_1} \overset{?}{=} up_1^c$

$g^{r_2} \overset{?}{=} up_2^c$

Figure 3.6: AND-composition of $\Sigma-$protocol for Schnorr protocol.

Proof of the relevant properties can be shown as follows:

- *soundness* property let AND(a, r) and AND$(a^{'}, r^{'})$ be two accepted conversation. Suppose they have different challenges $c \neq c^{'}$. We know that $g^{r_1} = up_1^c$ and $g^{r_1^{'}} = up_1^{c^{'}}$. Dividing both sides one can get $g^{r_1 - r_1^{'}} = p_1^{c-c^{'}}$. From this equality p_1 can be taken as $p_1 = g^{\frac{r_1 - r_1^{'}}{c-c^{'}}}$. Since p_1 also equals to $p_1 = g^{x_1}$, one can obtain x_1 as $x_1 = (r_1 - r_1^{'})/(c - c^{'})$. Similarly $x_2 = (r_2 - r_2^{'})/(c - c^{'})$.

- *Completeness* is almost straight forward. Since $r_1 = z_1 + cx_1$

  $$g^{r_1} = g^{z_1 + cx_1} = g^{z_1} g^{cx_1} = ug^{(x_1)^c} = up_1^c.$$

  Similarly,

  $$g^{r_2} = g^{z_2 + cx_2} = g^{z_2} g^{cx_2} = ug^{(x_2)^c} = up_2^c.$$

- *sHVZK* property, under the simulation both verifier and simulator has the same probability of distribution.

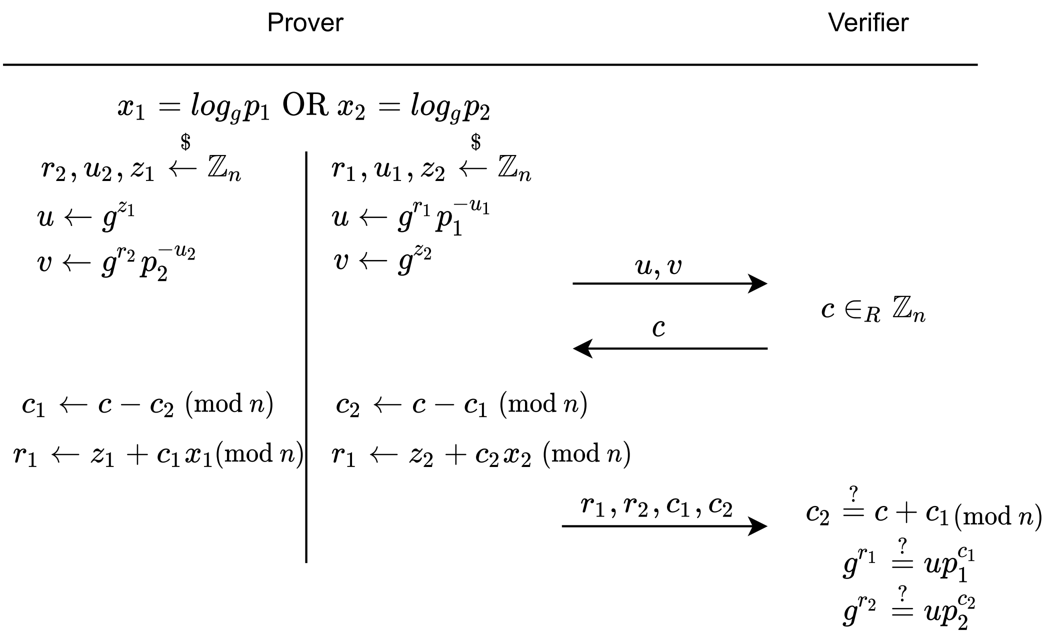

Prover
Verifier
$x_1 = log_g p_1$ OR $x_2 = log_g p_2$
$r_2, u_2, z_1 \overset{\$}{\leftarrow} \mathbb{Z}_n$
$u \leftarrow g^{z_1}$
$v \leftarrow g^{r_2} p_2^{-u_2}$
$r_1, u_1, z_2 \overset{\$}{\leftarrow} \mathbb{Z}_n$
$u \leftarrow g^{r_1} p_1^{-u_1}$
$v \leftarrow g^{z_2}$
u, v
$c \in_R \mathbb{Z}_n$
c
$c_1 \leftarrow c - c_2 \pmod{n}$
$r_1 \leftarrow z_1 + c_1 x_1 (\mathrm{mod}\, n)$
$c_2 \leftarrow c - c_1 \pmod{n}$
$r_1 \leftarrow z_2 + c_2 x_2 \pmod{n}$
r_1, r_2, c_1, c_2
$c_2 \overset{?}{=} c + c_1 (\mathrm{mod}\, n)$
$g^{r_1} \overset{?}{=} u p_1^{c_1}$
$g^{r_2} \overset{?}{=} u p_2^{c_2}$

Figure 3.7: OR-composition of $\Sigma-$protocol for Schnorr protocol.

3.4.1.2 OR-Composition

OR composition is a type of Σ-protocol that one of the statements is correct. Again under assumption of $\mathcal{R}_1$ and $\mathcal{R}_2$ being two relations, let the prover knows private input which belongs to herself (namely witness) belongs to $\mathcal{R}_1$. So that she knows her private input is in $\mathcal{R}_1 \vee \mathcal{R}_2$. To prove that, she needs to run an OR-proof as seen in figure 3.7. Proofs of the relevant properties can be shown as follows:

– *special soundness* property, let OR(a, r) and OR$(a^{'}, r^{'})$ be two accepted conversation. Suppose they have different challenges $c \neq c^{'}$. Since $c = c_1 + c_2$ and $c^{'} = c_1^{'} + c_2^{'}$, at least one of the followings should hold $c_1 \neq c_1^{'}$ or $c_2 \neq c_2^{'}$.

  On the other hand, we know that $g^{r_1} = up_1^{c_1}$, $g^{r_2} = up_2^{c_2}$ and $g^{r_1^{'}} = up_1^{c_1^{'}}$, $g^{r_2^{'}} = up_2^{c_2^{'}}$. Dividing both sides one can get the followings.

$$g^{r_1 - r_1^{'}} = p_1^{c_1 - c_1^{'}}, g^{r_2 - r_2^{'}} = p_2^{c_2 - c_2^{'}}.$$

  After that in situation of $c_1 \neq c_1^{'}$ from this equality p_1 can be taken as $p_1 = g^{\frac{r_1 - r_1^{'}}{c - c^{'}}}$. Since p_1 also equals to $p_1 = g^{x_1}$, one can obtain x_1 as $x_1 = (r_1 - r_1^{'})/(c - c^{'})$.

Else if $c_2 \neq c_2^{'}$ similarly one can obtain x_1 as $x_2 = (r_2 - r_2^{'})/(c - c^{'})$. As a result at least one of this situations will be obtained.

– *Completeness* We know that $c = c_1 + c_2$ (mod n)

$$g^{r_1} = g^{z_1+cx_1} = g^{z_1}g^{cx_1} = ug^{(x_1)^c} = up_1^c,$$

$$g^{r_2} = up_2^c.$$

Similarly,

$$g^{r_1} = up_1^c,$$

$$g^{r_2} = g^{z_2+cx_2} = g^{z_2}g^{cx_2} = ug^{(x_2)^c} = up_2^c.$$

– *sHVZK* property, under the simulation both honest verifier and simulator has the same probability of distribution for any given challenge.

3.5 Non-Interactive Proof Systems

There exist various methods to achieve a non-interactive proofs. Throughout this thesis, the structures we examined can be easily turned into non-interactive by using Fiat-Shamir Heuristic. Fiat-Shamir Heuristic is a transform to make interactive proofs into non-interactive ones by using a hash algorithm. This technique is presented by Amos Fiat and Adi Shamir in 1986 [20]. According to their original publication, the formal definition is:

Definition 3.5.1. (Fiat-Shamir Heuristic) Let (P, V) be a Σ-Protocol pair for the relation $\mathcal{R}$ and $\mathcal{H}$ be hash function. Let δ=(Gen,Com,Open) denotes the result of the commitment scheme. The proof system, depending on the δ, generating the challenge value by hashing the δ as $\mathcal{H}(\delta)$, and the resulting response is defined as *The weak Fiat-Shamir transform.*

Similarly, generating the challenge value by hashing the δ, Y as $\mathcal{H}(\delta, Y)$, where y is the input statement, is defined as *The strong Fiat-Shamir transform.*

CHAPTER 4

RANGE PROOFS

Brickell, Chaum, Damgård $\&$ van de Graaf proposed the first range proof in 1987 [8]. Their construction was based on discrete logarithm with using a *bit commitment*. The check accuracy of the proof they use $\Sigma-$protocol. This construction has many negative features, especially in ranging.

In 1995, Damgard proposed a ZKRP construction[18]. Soon later, in 1997, Fujisaki $\&$ Okamoto proposed another construction [21]. Although they work properly, the problem with both of these schemes was that they were inefficient to use in real-world cases.

In 1997, Bellare and Goldwasser presented *binary decomposition range proof* [14]. In this construction, secret s is represented in binary. This structure allows to $s \in [0, 2^{k-1}]$ where s has k bits. This construction is quite similar to Mao's construction. Each bit s_i of the secret s, written as a committed value to prove that it is actually a binary element. To do this, a special technique of proof of knowledge is used. This technique is also known as *OR-proof*, explained in Chapter 3, section 3.4. In 2001, Boudot proposed the first useful scheme [36], which we will explain in later this chapter in detail.

The main purpose of range proofs is to prove that a committed number lies in an interval without revealing the number itself. There exist various methods/algorithms to achieve this goal.

ZKRPs are important due to their several use cases. Some of them can be listed as e-cash schemes, group signature schemes, verifiable secret sharing (VSS) schemes, and they can also be used as sub-protocol in other zero-knowledge systems.

First, one should know zero-knowledge set membership proofs because any set-membership proof can be easily converted into a range proof [10]. The problem is transforming such proofs is not effective in many ways. Therefore some specific techniques are developed for this specific purpose.

This part will briefly give information about the set membership proofs, transform them into range proofs, and explain why these proofs are considered inefficient.

Some of the range proofs can be considered as special subsets of set membership proofs. Suppose δ=(Gen, Com, Open) be a denotation of a commitment scheme. A proof of set membership is a construction of a proof of knowledge with respect to challenge c and the set Ψ.

$$PK\big[(u,w) : \delta \leftarrow Com(u,w) \wedge u \in \Psi\big].$$

Set membership proofs are used when a party wants to prove that an $\mathcal{S}$ is a member of a language $\mathcal{L}$. As an example, [28], [10], [38] can be given for these varieties of constructions.

In the literature, there exist 2 main ways to commit the secret wanted to prove. These are *integer* or *binary*. In this thesis, we divide it to 3 main methods. We add an extra method namely $u-$*ary method*.

1. **binary method** This way allows to check that binary representation of a committed value is in the interval $[0, 2^k - 1]$. Due to the inefficiency of this method, we consider this method impracticable.

2. **integer method** In this way, it is enough to check whether a committed number belongs to interval $\mathcal{I}$ or not. Usually, $\mathcal{I}$ is chosen as a much larger interval space.

3. $u-$**ary method** This way allows to check that u-ary representation of a committed value is in the interval $[0, u^k - 1]$. It is stated in the literature, this method itself does not add extra efficiency to an algorithm.

We will also mention which of these methods the given structures use.

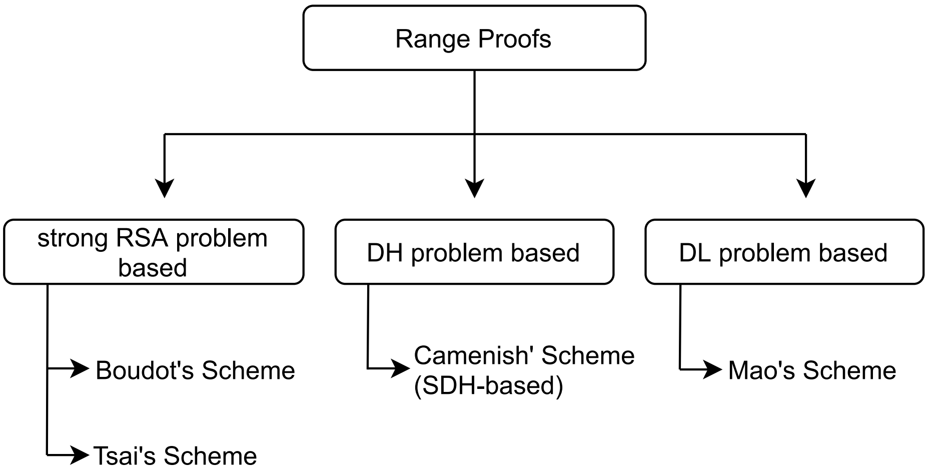


Figure 4.1: Classification of zero-knowledge range proofs.

In this thesis, existing schemes will be examined based on their hardness assumption under these three main headlines as shown in the Figure 4.1:

1. Strong RSA problem-based method,
2. Diffie-Hellman problem-based method,
3. Discrete logarithm problem-based method.

For each category, we will explain one or two constructions. In the DLP-based method, we will analyse the efficiency of one of the existing protocols for different basis.

4.1 Strong RSA problem based ZKRP

We have already defined strong RSA problem in chapter. One of the very old and important schemes is Boudots, depending on this problem. Later Lipmaa has proposed another construction. We can say that also Grooth's studies were based on this problem. To keep it short, at first, we will explain Boudot's scheme and then we will give Tsai et.al.'s scheme which has common underlying protocols with Boudot's scheme.

4.1.1 Boudot's Scheme

To prove an integer x lies in interval $[a, b]$ it is enough to prove both $x - a$ and $b - x$ are both positive. Indeed the problems turn out to proving the positivity of an integer.

To do so, there exist three more proofs one needs to know beforehand. These are:

1. *the proof of the committed number is a square*,
2. *the proof of two commitments hides the same secret*,
3. *the CFT proof.*

First, we will explain these, and after that, we will explain Boudot's construction. Suppose $E(x, r) = g^x h^r \pmod{n}$ is a homomorphic commitment where $g \in \mathbb{Z}_n^*$ and h is element of a group generated by g both $\log_g h$ and $\log_h g$ are unknown by prover. n is composite number and factors of it are unknown for both prover and verifier. Lastly $r \in_R [-2^\kappa n + 1, 2^\kappa n - 1]$.

First sub-protocol used in the construction is the proof that two committed values hide the same secret. Where (t, l, κ) are security parameters:

$$PK_{ss}[x, r, \hat{r} | E = E(x, r) \wedge \hat{E} = E(x, \hat{r})].$$

Prover	Verifier
Randomly chooses:	
$\mu \xleftarrow{\$} [1, 2^{l+t}b - 1]$	
$\tau_1 \xleftarrow{\$} [1, 2^{l+t+\kappa_1}n - 1]$	
$\tau_2 \xleftarrow{\$} [1, 2^{l+t+\kappa_2}n - 1]$	
Computes:	
$\Psi_1 \leftarrow g^\mu h^{\tau_1} \pmod{n}$ and $\Psi_2 \leftarrow \hat{g}^\mu \hat{h}^{\tau_2} \pmod{n}$	
$c \leftarrow H(\Psi_1 \Vert \Psi_2)$	
$W_0 \leftarrow \mu + cx_1$	
$W_1 \leftarrow \tau_1 + cr$, $W_2 \leftarrow \mu + c\hat{r}$	
	$\xrightarrow{c, W_0, W_1, W_2}$
	To verify checks the following:
	$c \stackrel{?}{=} Hash(g^{W_0} h^{W_1} E^{-c} \Vert \hat{g}^{W_0} \hat{h}^{W_2} \hat{E}^{-c})$

Figure 4.2: Proof of two commitments hide the same secret PK_{ss}

The second sub-protocol used in this construction is the proof of the committed value is a square. Where (t, l, κ) are security parameters:

$$PK[x, r^{'} | \hat{E} = E(x^2, r^{'})], \text{ where } r^{'} = xr + \hat{r}.$$

Prover		Verifier
Randomly chooses:		
$\hat{r} \overset{\$}{\leftarrow} [-2^{\kappa} n^1, 2^{\kappa} n - 1]$		
$\hat{E} \leftarrow g^x h^{\hat{r}}$		
$\tilde{r} \leftarrow r - \hat{r} x$		
$E \leftarrow \hat{E}^x h^{\tilde{r}}$		
Calls for PK_{ss}		
$PK_{ss}(x, \hat{r}, \tilde{r} \vert \hat{E} = g^x h^{\hat{r}}$ mod $n \wedge E = \hat{E}^x h^{\tilde{r}}$ mod n		
From PK_{ss} she gets c, W_0, W_1, W_2		
	$\xrightarrow{\hat{E}, c, W_0, W_1, W_2}$	
		Checks if the following is valid to verification:
		$PK_{ss}(x, \hat{r}, \tilde{r} \vert \hat{E} = g^x h^{\hat{r}}$ mod $n \wedge E = \hat{E}^x h^{\tilde{r}}$ mod n

Figure 4.3: Proof of committed value is a square PK_{sq}

The last sub-protocol is a variation of the CFT protocol to proving $x \in [-\Theta, \Theta]$, the absolute value of committed value x is less than Θ.

$$PK_{CFT}[x, r, E = E(x, r) \wedge x \in [a, b]].$$

At the and of this protocol verifier convinces that committed value $x \in [-2^{t+l} b, 2^{t+l} b]$

After these building blocks, to make the overall system to work, Boudot's ZKRP inputs are chosen as follows. Let $\Lambda = |b - a|$. Set $T = 2(t + l + 1) + \Lambda$, $X = 2^T x$ and $R = 2^T r$.

Prover	Verifier
Randomly chooses:	
$\mu \overset{\$}{\leftarrow} [0, 2^{l+t}b - 1]$	
$\tau \overset{\$}{\leftarrow} [1 - 2^{l+t+\kappa_1}n + 1, 2^{l+t+\kappa_1}n - 1]$	
Computes:	
$\Psi_1 \leftarrow g^{\mu}h^{\tau}$ (mod n)	
$c \leftarrow H(\Psi)$	
$c^{'} \leftarrow c($ mod $2^t)$	
$W_1 \leftarrow \mu + xc^{'}$	
$W_2 \leftarrow \tau + xc^{'}$	
If $W_1 \notin [cb, 2^{t+l}b - 1]$ starts again from the beginning	
$\xrightarrow{c, W_1, W_2}$	
	Verification is done by checking:
	$W_1 \overset{?}{=} [cb, 2^{t+l}b - 1]$
	$c^{'} \overset{?}{=} Hash(g^{W_1}h^{W_2}E^{-c}$

Figure 4.4: Proof of CFT for larger interval PK_{CFT}

Prover	Verifier
Computes:	Computes:
$E_A = E/g^a$ (mod n)	$E_A = E/g^a$ (mod n)
$E_B = g^b/E$ (mod n)	$E_B = g^b/E$ (mod n)
Sets:	
$\hat{x} \leftarrow x - a$ and $\tilde{x} \leftarrow b - x$	
Sets:	
$\hat{x}_1 = \lfloor\sqrt{\hat{x}}\rfloor$ and $\hat{x}_2 = \hat{x} - \hat{x}_1^2$	
$\tilde{x}_1 = \lfloor\sqrt{\tilde{x}}\rfloor$ and $\tilde{x}_2 = \tilde{x} - \tilde{x}_1^2$	
$\hat{r}_1, \hat{r}_1 \in_R [-2^{\kappa}n + 1, \ldots, 2^{\kappa}n - 1]$ s.t $\hat{r}_1 + \hat{r}_2 = r$	
Similarly,	
$\tilde{r}_1, \tilde{r}_2 \in_R [-2^{\kappa}n + 1, \ldots, 2^{\kappa}n - 1]$ s.t $\tilde{r}_1 + \tilde{r}_2 = -r$	
Computes new commitments:	
$E_{A1} = g^{\hat{x}_1^2}h^{\hat{r}_1}$ and $E_{A2} = g^{\hat{x}_2}h^{\hat{r}_2}$	
$E_{B1} = g^{\tilde{x}_1^2}h^{\tilde{r}_1}$ and $E_{B2} = g^{\tilde{x}_2}h^{\tilde{r}_2}$	
$\xrightarrow{E_{A1} \& E_{B1}}$	
	Validation of the Commitments:
	$PK_{sq}(\hat{x}_1, \hat{r}_1, E_{A1} = (\hat{x}_1^2, \hat{r}_1))$
	$PK_{sq}(\tilde{x}_1, \tilde{r}_1, E_{B1} = (\tilde{x}_1^2, \tilde{r}_1))$
	$PK_{CFT}(\hat{x}_2, \hat{r}_2, E_{A2} = E(\hat{x}_2, \hat{r}_2) \wedge \hat{x}_2 \in [-\theta, \theta])$
	$PK_{CFT}(\tilde{x}_2, \tilde{r}_2, E_{B2} = E(\tilde{x}_2, \tilde{r}_2) \wedge \tilde{x}_2 \in [-\theta, \theta])$

Figure 4.5: Boudots ZKRP with Proof with Tolerance

To compute the cost of this construction in terms of exponentiations, we first consider the prover side. 8 exponentiations come from the PK_{SS}, 4 exponentiations come from the PK_{SQ}, and CFT proof only has 3 exponentiations. The main scheme has 12 exponentiations. Overall scheme costs 27 $\mathscr{E}$. Similar computations can be done for the verifier side and results 24 $\mathscr{E}$. There exist 4 elements to exchanged which are c, W_0, W_1, W_2.

4.1.2 Tsai's Scheme

In 2019. Tsai et. al. proposed an improved range proof for decentralized applications [43]. For simplicity, we call this construction as Tsai's scheme in this thesis. The idea is that to prove an integer x lies in interval $[a, b]$ it is enough to prove $(x - a + 1)(b - x + 1) > 0$, assuming both a and b positive integers.

With this aim, to prevent information leakages, for random chosen integer δ, they prove $\delta^2(x - a + 1)(b - x + 1) > 0$. Let $S + P = \delta^2(x - a + 1)(b - x + 1)$. It uses PK_{SS} and PK_{SQ} as sub-protocols which we have explained in Boudots scheme. But we use non-interactive versions of these proofs. It is stated in the original paper, non-interactive range proofs are more suited options comparing with interactive ones. So that, they construct a non-interactive ZKRP that has flexiable range form. We will directly give the workflow of the scheme as follows:

This construction provides completeness, soundness and zero-knowledgeness properties. To see the proofs of these properties, original paper in [43], can be checked.

4.2 DHP-based ZKRP

Strong Diffie-Hellman (SDH) and Decisional Diffie-Hellman (DDH) assumptions can be used in this method. A suitable digital signature algorithm signs each element in the range. Then, the prover proves that he knows a secret signature in a blind way.

There are not many structures proposed with this method. In the literature, Camenisch et al. proposed an efficient construction based on q-Strong Diffie-Hellman assump-

Prover		Verifier
Computes:		Computes:
$E_1 = E/g^{a-1}$ (mod n)		$E_1 = E/g^{a-1}$ (mod n)
$E_2 = g^{b+1}/E$ (mod n)		$E_2 = g^{b+1}/E$ (mod n)
Computes:		
$\hat{E} = E_1^{(b-x+1)} h^{\hat{r}}$ (mod n)		
$\tilde{E} = E_1^{\delta^2} h^{\tilde{r}}$ (mod n)		
Choses		
$\lambda \xleftarrow{\$} \mathbb{Z}^+$		
Computes:		
$S = \lambda^2$ where $S < \delta^2(x-a+1)(b-x+1)$		
$P = \delta^2(x-a+1)(b-x+1) - S$		
Sets $r_1, r_2 \in \mathbb{Z}^+$, s.t.		
$r_1 + r_2 = \delta^2((b-x+1)+r+\hat{r}) + \tilde{r} - S$		
Computes new commitments:		
$E_1^{'} = g^S h^{r_1}$ (mod n)		
$E_2^{'} = g^P h^{r_2}$ (mod n)		
	$\xrightarrow{E,\hat{E},\tilde{E},E_1^{'},E_2^{'}\&P}$	
		Validation:
		$PK_{ss}(E_2, \hat{E}, b-x+1, -r, \hat{r})$
		$PK_{sq}(\delta, \tilde{r}_1, \tilde{E} = (\delta^2, \tilde{r}))$
		$\tilde{E} \overset{?}{=} E_1^{'} E_2^{'}$ (mod n)
		$PK_{sq}(\alpha, r_1, E_1^{'} = (S, r_1)$
		$P \overset{?}{>} 0$

Figure 4.6: Tsai's non-interactive ZKRP

tion for the ranges of the type $[0, n^l]$. Applying the proof twice can also be used for ranges of the type $[-n^l, n^l]$.

4.2.1 Camenisch's Scheme

In 2008, Camenisch et al. proposed an efficient zero-knowledge set membership proof and a range proof application [10]. Their construction inspired an oblivious transfer scheme proposed by Camenisch et al. [13], and it is based on bilinear group signatures and Strong RSA assumption.

The main idea of range proof is defined in previous sections. In this construction, instead of using bit commitment schemes, base$-u$ representation is used. Then every element of the set is encoded with a digital signature with a key by the verifier. So that the problem becomes *"prove that signed value refers to a committed element of the set* $\mathcal{C}$*."* Prover gets subject matter signatures, after signing blinds them, to hide the committed element.

They used the Boneh-Boyen signature algorithm. Security of the algorithm relies on the hardness of the q-Strong Diffie-Hellman assumption.

Lemma 4.2.1 (4). *Suppose* $\hat{e} : (\mathbb{G}_1, \mathbb{G}_t)$ *is a bilinear map. In the case of q-Strong Diffie-Hellman assumption valid on* $\hat{e}$*, then under a weak chosen message attack, Boneh-Boyen signature scheme is* $q-$*secure against an existential forgery.*

Prover		Verifier
		First verifier picks random x,$x \overset{\$}{\leftarrow} \mathbb{Z}_p$ And computes the corresponding y as $y \leftarrow g^x$ For each $i \in Z_u$, computes $\mathcal{A}_i \leftarrow g^{\frac{1}{x+1}}$ $x_A \leftarrow x - a$ and $x_B \leftarrow b - x$
	$\xleftarrow{\text{sends the values of } y\&A_i}$	
Prover binds the signatures For each $j \in Z_p$,randomly picks v_j Computes$V_j \leftarrow A_{\sigma_j}^{v_j}$ s.t. $\sigma = \sum_j(\sigma_j u^j)$		
	$\xrightarrow{\text{sends the values of } V_j}$	
Prover and verifier performs proof of knowledge		
For each $j \in Z_p$,randomly picks s, t Computes $a_j \leftarrow e(V_j, g)^{-s_j} e(g, g)_j^t$, and $D \leftarrow \prod_j (g^{u^j s_j}) h^{m_j}$,		
	$\xrightarrow{\text{sends the values of } (a,D)_(j \in \mathbb{Z}_p)}$	

Figure 4.7: Proof Generation of ZKRP by Camenisch et al.

So far, the construction above results correct solutions for ranges of the form $[0, B)$ or $[0, u^l)$. In real-world applications, this is not always the case. To handle arbitrary ranges that might be of the form $[A, B]$, an improvement on Berry Schoenmakers folklore suggestion can be used.

Let upper bound B can be expressed as $u^{l-1} < B < u^l$. Our aim is to prove $\sigma \in [A, B)$. Then it is straigtforword to show $\sigma \in [A, A + u^l)$ and $\sigma \in [B - u^l, B)$ is

Prover	Verifier
	Verification of PoK
	Verifier randomly chooses $c \leftarrow \mathbb{Z}_p$
$\xleftarrow{\text{random challange } c}$	
For each $j \in Z_l$ computes $z_{\sigma_j} \leftarrow s_j - \sigma_j c$ and $z_{v_j} \leftarrow t_j - v_j c$ then computes $z_r \leftarrow m - rc$	
	In Verification step verifier checks the followings
	$D \overset{?}{=} C^c h^{z_r} \prod_j (g_j^{z_{\sigma_j}})$
	For every j values in $\mathbb{Z}_l$ checks the following
	$a_j \overset{?}{=} e(V_j, y)^c e(V_j, g)^{-z_{\sigma_j}} e(g, g)^{z_{v_j}}$

Figure 4.8: Verification of ZKRP by Camenisch et al.

enough.

If $\sigma \in [B - u^l, B)$, one can also represent this as $\sigma - B + u^l \in [0, u^l)$. Similarly $\sigma \in [A, A + u^l)$ can be represented as $\sigma - A \in [0, u^l)$.

It is enough to send for once the verification key vk and u signatures for both of the sets above. Therefore in the scenario arbitrary range $[A, B]$, the verifier needs to check

$$D \overset{?}{=} C^c g^{-B+u^l} h^{z_r} \prod_j (g_j^{z_{\sigma_j}})$$

$$D \overset{?}{=} C^c g^{-A} h^{z_r} \prod_j (g_j^{z_{\sigma_j}})$$

There also exist more optimized version in the version of $A + u^{l-1} < B$, the set can be represented as $[A, B) = [B - u^{l-1}, B) \cup [A, A + u^{l-1})$. Here OR-composition is used.

4.3 DLP-based ZKRP

Working on DLP has many positive features. One of them is, it makes schemes easily applicable on elliptic curves. There are many old studies on this method. In [31], Mao proposed a binary algorithm to prove the secret s lies in the given interval which we will examine deeply in this section.

Schoenmakers, on the other hand, has many researches and iterative algorithms in this decomposition. Suppose we have an interval $\mathcal{I} = [0, b)$ he suggested to write upper bound as a variant of 2: $\mathcal{I} = [0, 2^n)$. He has two approaches: one is done by using AND-proofs and the other is done by using OR-proof.

While $2^{n-1} < b < 2^n$, interval can be rewritten as

$$\mathcal{I}[0, b) = [0, 2^{n-1}) \cap (b - 2^{n-1}, b),$$

with using AND-composition of Σ-protocol (AND-proof). An equivalent representation is given by,

$$\mathcal{I} = [0, b) = [0, 2^n) \cup (b - 2^n, b),$$

with using OR-composition of Σ-protocol (OR-proof). In this scenario $4(\hat{n} - 1)$ Schnorr-type OR-proofs are needed.

To increase the efficiency, another suggestion Berry Schoenmakers made is, in case of $2^{\hat{n}-1} < b < 2^{\hat{n}}$, precedent upper bound b can be rewritten as $b = 2^{\hat{n}} + \kappa$, where $2^{n^{'}-1} < \kappa < 2^{n^{'}}$. Here $n^{'} \leq \hat{n}$. So the OR proof case will become:

$$\mathcal{I} = [0, b) = [0, 2^{\hat{n}}) \cup (b - 2^{n^{'}}, b).$$

Bulletproofs are also very important DLP-based range proofs. The aim of to protocol is to prove that commitment $Com(s, r) = xH + rG$, hides the secret element $s \in [0, 2^k)$. To do so, the protocol uses *inner product argument*, which based on the fact that *we can represent any number as an inner product of two vector.* When $(a_1, a_2, \ldots, a_k)$ is the bits of the secret s, we show that $s = (a_1, a_2, a_3 \ldots, a_k) \times (2^0, 2^1, 2^3, \ldots, 2^{k-1})$, and this leads us the result of $s \in [0, 2^k)$ [44].

We want to hide the bits of the secret in a single vector Pedersen commitment. Since we have vectors of size k, the cost of exponentiations is quite expensive. The homomorphic property of Pedersen commitments, allows us to halve the vectors. By doing this halving $\log_2 k$ times, we get a single element. We apply this method to the inner product vectors until we get a single multi-exponentiation instead of k-times which makes the system more efficient.

In the workflow of the protocol, some combinations of constrictions and challenges over $\mathbb{Z}_p$ are sent to the prover by the verifier. With these combinations, the prover computes the inner product containing vector a, and blinding vectors. With the help of the Fiat-Shamir heuristic whole scheme can be made non-interactive easily.

Now we will examine Mao's bit-wise construction, after that, we will analyze the efficiency of the scheme if we modify the scheme for other bases.

4.3.1 Mao's Construction [31]

Let p be a large prime and let $p = kq + 1$ where k is an even number. Let $g, h \in \mathbb{Z}_p^*$ is an element of order q and discrete logarithm of h in base g, $\log_g(f)$ is unknown by Prover.

When $r \in_R \mathbb{Z}_p^*$, we say that $E = g^x h^r \bmod p$ is a commitment to hide x.

Firstly, when $k = |x| + 1$ binary representation of x can be written as:

$$x = x_0 2^0 + x_1 2^1 + \cdots + x_{k-1} 2^{k-1} \text{ for } x_i \in 0, 1 \text{ and } i = 0, 1, \ldots, k-1.$$

Prover chooses $r_0, r_1, \ldots, r_k$ randomly s.t. $\sum_{i=0,\ldots k-1} r_i = r$ then computes the following bit commitment scheme:

$$E_i = E(x_i, r_i) = g^{x_i} h^{r_i} \bmod p \text{ for } i = 0, 1, \ldots, k-1.$$

After that, the prover proves that, in each step, the value committed by $E(x_i, r_i)$ is whether 0 or 1. For this purpose, one can use a zero-knowledge sub-protocol, namely the OR-composition of Σ-protocol, and shows that she knows whether E_i in base h or E_i/g in base h as shown in Figure 4.9.

After all, verifier gets each E_i and r, with help of homomorophic property, he also needs to check:

$$\prod_{i=0}^{k-1} E(x_i, r_i) \stackrel{?}{=} E(x, r) \bmod p.$$

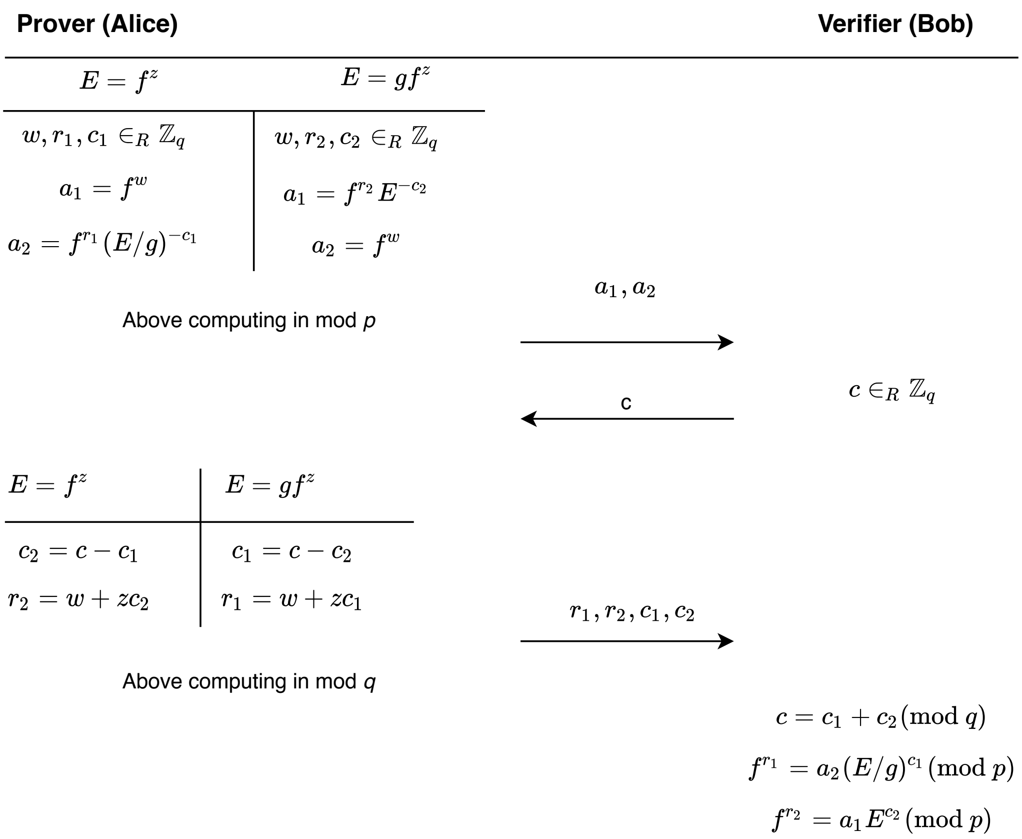


Figure 4.9: bit-lenght-based OR-Proof [31]

Here we say, a large enough p is 1024-bits, so q is 1023-bits. By $\mathscr{E}$, we denote exponentiation. Similarly, $\mathscr{I}$ denotes inverse, $\mathscr{M}$ denotes multiplication.

The complexity of the proof depends on the size of x. We supposed $k = |x| + 1$ at the beginning. Since both prover and verifier need to compute 4 exponentiations for each bit of x, in total we have $4k$ exponentiations.

We also have 7 integers: $a_1, a_2, c, r_1, r_2, c_1, c_2$ exchanged between prover and verifier. Say $k^{'} = |q|$. In total, cost of exchanging integers equals $7kk^{'}$.

CHAPTER 5

BASE-U RANGE PROOF

In this chapter, we investigate existing bit-wise range proofs on different bases. To do so, the secret we want to commit is represented in base-u. For each bit of the secret, we call the scheme we want to use. Here, though complexity is increasing on the scheme, we expect to decrease overall complexity since we call the scheme fewer times.

In this thesis, we take Mao's construction and modified it to base-3, and after that, we generalize it for base-u.

5.1 Mao's ZKRP with Base-u OR-Proof

In this section we will observe Mao's construction for different basis. At first we use ternary-lenght-based representation instead of bit-length-based. So that, we denote the secret we want to prove as ternary representation as follows:

$$x = x_0 3^0 + x_1 3^1 + \cdots + x_{\tilde{k}-1} 3^{\tilde{k}-1} \text{ for } x_i \in 0,1,2 \text{ and } i = 0,1,\ldots,\tilde{k}-1.$$

After that, for randomly chosen $r_0, r_1, \ldots, r_{\tilde{k}-1}$ values s.t. $\sum_{i=0,\ldots\tilde{k}-1} r_i = r$, we compute the commitments as:

$$E_i = E(x_i, r_i) = g^{x_i} h^{r_i} \bmod p \text{ for } i = 0,1,\ldots,\tilde{k}-1.$$

After this step, we need to use OR-proof. But with classical base-2 OR-proof, we cannot check each commitment for once in our case. So, instead of base-2 OR-Proof we construct base-3 OR-proof to prove that committed value is weather equal E_i, or E_i/g, or E_i/g^2 as follows.

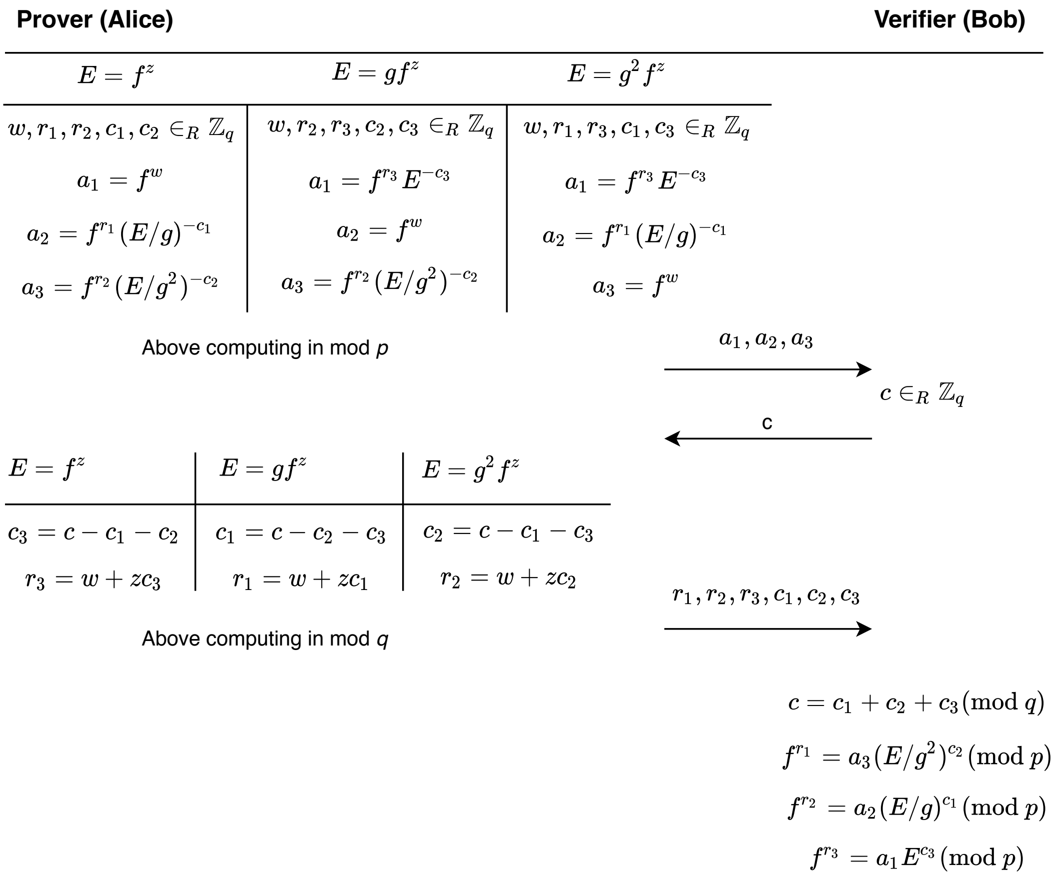

Prover (Alice)
Verifier (Bob)
$E = f^z$
$E = gf^z$
$E = g^2 f^z$
$w, r_1, r_2, c_1, c_2 \in_R \mathbb{Z}_q$
$a_1 = f^w$
$a_2 = f^{r_1}(E/g)^{-c_1}$
$a_3 = f^{r_2}(E/g^2)^{-c_2}$
$w, r_2, r_3, c_2, c_3 \in_R \mathbb{Z}_q$
$a_1 = f^{r_3}E^{-c_3}$
$a_2 = f^w$
$a_3 = f^{r_2}(E/g^2)^{-c_2}$
$w, r_1, r_3, c_1, c_3 \in_R \mathbb{Z}_q$
$a_1 = f^{r_3}E^{-c_3}$
$a_2 = f^{r_1}(E/g)^{-c_1}$
$a_3 = f^w$
Above computing in mod p
a_1, a_2, a_3
$c \in_R \mathbb{Z}_q$
c
$E = f^z$
$E = gf^z$
$E = g^2 f^z$
$c_3 = c - c_1 - c_2$
$r_3 = w + zc_3$
$c_1 = c - c_2 - c_3$
$r_1 = w + zc_1$
$c_2 = c - c_1 - c_3$
$r_2 = w + zc_2$
Above computing in mod q
$r_1, r_2, r_3, c_1, c_2, c_3$
$c = c_1 + c_2 + c_3 (\mathrm{mod}\, q)$
$f^{r_1} = a_3(E/g^2)^{c_2} (\mathrm{mod}\, p)$
$f^{r_2} = a_2(E/g)^{c_1} (\mathrm{mod}\, p)$
$f^{r_3} = a_1 E^{c_3} (\mathrm{mod}\, p)$

Figure 5.1: 3-base OR-proof.

Again, verifier gets each E_i and r, he needs to check the following with help of homomorophic property:

$$\prod_{i=0}^{\tilde{k}-1} E(x_i, r_i) \overset{?}{=} E(x, r) \bmod p.$$

5.2 Comparisons and Discussions

Clearly, this base-3 scheme succeeds in our case. For the complexity of the base-3 construction, both prover and verifier need to compute 6 exponentiations in each step. In total, we need $6\tilde{k}$ exponentiations. On the other hand, we will consider the numbers exchanged between the prover and the verifier. In this scenario, there exist 10 integers to exchange between prover and verifier in each step. The cost of exchange then equals $10\tilde{k}k^{'}$.

Based on this, we can write the requirements for base-4. This time, 2 more exponentiations needed for both prover and verifier, so we need $8\tilde{k}$ exponentiations. This time there would be 13 numbers to exchange which will cost $13\tilde{k}k^{'}$ in total.

Before generalizing this scheme to base-u, remember $\mathscr{E} \approx 8\mathscr{I}$, which means inverse operation cost much less comparing exponentiation. Also $\mathscr{E} \approx 1000\mathscr{M}$. That is why other operations are can be seen as negligible. So that, it is enough to check exponentiations here. Now in general, when we work on base-u, both prover and the verifier needs to compute $2u$ exponentiations in each step. And since there will be $3u + 1$ numbers to exchange in each step on base-u. One may see the required exponentiations at the Table 5.1.

Table 5.1: Table to compare requirings for different basis in each step

Basis Cost type	Base-2	Base-3	Base-4	...	Base-u
exponentiations	4	6	8	...	$2u$
numbers to exchange	7	10	13	...	$3u + 1$

These exponentiations and exchangings will tekrar edicek length of secret. Remember length of the secret x is denoted $k = |x|+1$ in base 2 representation and $\tilde{k} = |x|+1$ in base 3 representation. We know that, $\log 2 = 0,30102$ and $\log 3 = 0,47771$ so in base 3 representation $\tilde{k} = \frac{\log 2}{\log 3}m \approx 0,63k$. Now we can compare these 2 worst case complexities by exponentiations of both proof generation and verification as:

$$\frac{0,63k(6\mathscr{E})}{k(4\mathscr{E})} = \frac{3,78\mathscr{E}}{4\mathscr{E}}$$

It can be seen that we have 5.5% efficiency in both proof generation and verification if we use ternary-length-based representation instead of bit-length-based. We also analyze other basis complexities either in the same way and generalize this for base-u. For base-4, since $\frac{\log 2}{\log 4} \approx 0,5$, so $4k$ exponentiations required. For base-u, number of required exponentiations can be formalised as $\frac{\log 2}{\log u}(2u)k$. Similar computations can be done for total numbers exchanged between prover and verifier. In the bit-length-based approach, $7kk^{'}$ bits exchanged. In ternary-length-based $10\tilde{k}k^{'}$ bits exchanged

Table 5.2: Table to compare requirings for different basis in total

Basis Cost type	Base-2	Base-3	Base-4	Base-5	...	Base-u
exponentiations	$4k$	$3.785k$	$4k$	$4.3k$	...	$\frac{\log 2}{\log u}(2u)k$
numbers to exchange	7	6.309	6.5	6.88	...	$\frac{\log 2}{\log u}(3u+1)$

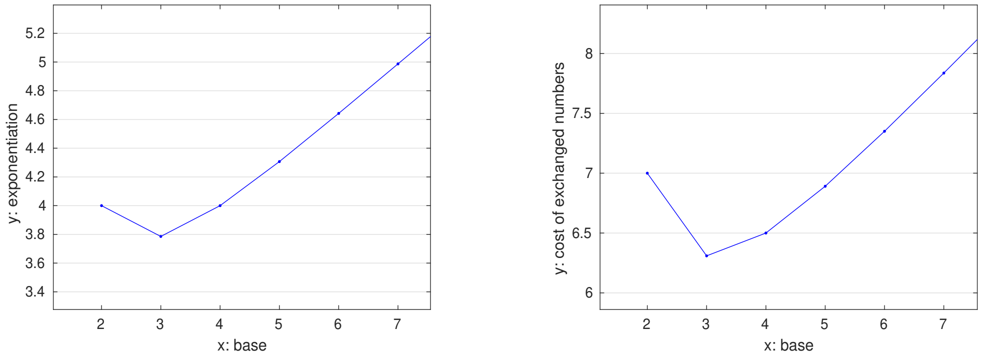
as we mentioned before.

$$\frac{0,63k(10k^{'})}{k(7k')} = \frac{6.309\mathscr{E}}{7\mathscr{E}}$$

So that we have approximately 7% efficiency in the base-3 approach comparing with the base-2 approach. If we generalize this to base-u, it can be formalized as $\frac{\log 2}{\log u}(3u+1)$, and the most efficient computations come in base-3.

We also sketch the total required exponentiations among different bases as seen in the first of the following graphs. You can check that maximum efficiency can be observed on base-3.

Also, in the second graph one can see the comparison of the total number of exchange bits again has maximum efficiency on base-3.



(a) Required exponentiation

(b) bits exchange numbers

Figure 5.2: Caption for this figure with two images

In the first graph 5.2a, you may find the number of required exponentiations depending on different bases. The graph has its minimum value in $(3, 3.78)$, which is our most efficient point.

In the second graph 5.2b, we can see the number of bits exchanged. Although in folklore bit-representation it equals 7, in base-3, base-4 and base-5 it has better results. Still, it has its minimum value in $(3, 6.31)$, which is our most efficient point. As a result of both comparisons, construction has its most efficiency when using base-3 with our base-3 OR-composition.

CHAPTER 6

APPLICATIONS ON DECENTRALIZED CONSTRUCTIONS

Many privacy concerns may easily be solved on up-to-date decentralized systems by using variants of ZKPs[27][4][35]. In these constantly improving systems, ZKRP's have a significant role in achieving privacy. We have mentioned various areas to use ZKRP's in electronic voting, electronic cash, multi-coupon, and anonymous credential systems. Many of these systems are non-decentralized applications. However, in the context of decentralized systems, we will not consider non-decentralized applications so that we will explain some financial decentralized distributed ledgers and how they use ZKRP's. In this chapter, we will mention some of the substantial applications that used ZKRP's.

Financial institutions might use Distributed Ledgers (DL) to construct an effective and efficient compromise transaction between organizations. Current DL systems have many drawbacks in terms of privacy and auditing. They are either public to all participants which is not okay because some sensitive strategy and trading information should have kept as secret, or they ensure privacy, but on the other hand, block auditing. Again this is very problematic for financial institutions need to be audited to proving they follow the regulations and financial oversights.

Usually, privacy in distributed ledgers can be provided in two different ways:

1. Using a hash function, hashing the transactions and keeping them as committing hashes. A TTP later can verify the transactions [37].

2. Using a cryptographic commitment scheme to commit transactions.

6.1 Confidential Assets

Confidential transactions are those transferred amounts hidden/encrypted. While privacy is holding, it is still possible to prove that there is no money coming from thin air or destroyed anywise. Its verification can be proven by showing the difference of input amount, and output amount equal to the fee.

On the other hand, *confidential assets* are supporting to track multiple asset types in a single blockchain transaction with keeping privacy [30]. In those schemes, both amount and the asset type are kept secret. It enables users to use their assets on privacy-related blockchain systems.

For example, Poelstra et al. [40] proposed a scheme in which multiple asset types can be tracked with a single distributed ledger. The hardness of this scheme relies on the elliptic curve discrete logarithm (ECDL). Due to its additively homomorphic property, the Pederson commitment scheme is used to blind each UTXO's amount. This results in an enhancement of privacy. Using a variant of *Borromean Ring Signature* [32], They constructed a *Back-Maxwell range proof* to deal with the attacks comes overflow. Similar to Schoenmaker's construction, OR-proofs are used to prove each digit lies in the intended interval, and the amounts of to each UTXO's are blinded.

6.2 Monero

Monero is an open-source, one-dimensional distributed acyclic graph, privacy-oriented cryptocurrency on the blockchain. It uses ring signatures and stealth addresses to ensure privacy [2]. Due to its ring signature basis, it allows hiding identity from other participants in a group.

In Monero, operations work on elliptic curves, specifically curve Ed25519. It uses Pedersen Commitments and *Schnorr-type Borremean signatures* described teborremean to hide the output amounts while input amounts are hidden with *multilayered linkable spontaneous anonymous group (MLSAG) signatures* [29].

In each transaction of Monero, we use range proofs to show validation of amounts in the outputs. It is sufficient to show each output committed value, say C_i, must be equal to the amount commitment C_A. So it should behold that $\sum_{i=0}^{k-1} = C_A$. It uses folklore bit decomposition and proves that each committed value equals weather 0 or 1 with OR proof.

6.3 zkLedger

We explained that ensuring auditing while preserving privacy has the utmost importance for financial institutions. zkLedger[37] offers privacy with fast and provable auditing with Schnorr-type NIZK's. While transaction graph and linkages between transactions kept secret, only transaction time and transferred asset type are published. Still, it allows auditors to use primitives such as sums, moving averages, variance, standard deviation, and ratios. The system does not need any trusted setup and provides completeness.

We work on elliptic curves, points of the group $\mathbb{G}$ are on curve secp256k1. It uses Pedersen Commitments because of its *additive homomorphic* property. A group of banks might store the transactions as commitments instead of plaintext, later combining them homomorphically.

Every bank in the system generates a secret key sk, and corresponding public key $pk = h^{sk}$, together (sk, pk) called as *Schnorr signature keypair*. The public keys are distributed to the participants in the system.

Suppose n be the order of the group $\mathbb{G}$. Then it is sufficient to check if the committed value is in the range $[0, n-1]$. Otherwise both $Com(x, r)$ and $Com(x+n, r)$ have the same result. This leads to the ability of a malicious bank to generate assets that cannot be detected. In this point, Back-Maxwell Range proof [40] under *Borremean ring signatures* [39], which are not explained in context of this thesis, is used. zkLadger construction needs two range proofs: to prove the committed value and prove the column's sum of assets. These two range proves can be down to one proof as using auxiliary commitment. This auxiliary commitment might be a commitment of , or might be sum of the values of first m values $\sum_{i=1}^{m} x_i$.

6.4 Zether

Zether is a fully decentralized, confidential payment mechanism implemented as *Ethereum Smart Contract*. Unlike Bitcoin or Monero with their UTXO model, Zether has *account-based model* [9].

Since Bulletproofs has short proofs and does not need any trusted setup, it is used as an underlying proof system. Instead of Pedersen commitments, it uses El-Gamal encryptions which also have homomorphic property. It combines Bulletproofs with $\Sigma-$protocol and results Σ-Bullets. To do so, they can efficiently prove that a value encrypted with El-Gamal is within a range. Moreover combining range proofs with ring signatures, they provide anonymous transfers.

CHAPTER 7

CONCLUSION

One particular kind of ZKP is zero-knowledge range proof (ZKRP). This thesis has examined and summarized the existing range proof methods into 3 main headlines based on their underlying cryptographic hardness assumption: , strong RSA problem based, DHP based, and DLP based. Since we focus on the cryptographic structure of the schemes, we explained primitives such as commitments, digital signatures, proof systems, and required sub-protocols in detail.

For each headline, we explain the workflow of one or two constructions. In strong RSA problem-based constructions, we choose to explain Camenish's scheme and Tsai et al. scheme. Camenish's scheme has importance for us due to its base-u representation, which we will use in Mao's scheme. In this scheme, it is stated that using the base-u approach does not add improvement, and this is why they use other techniques to achieve efficiency. Tsai et al. scheme draws attention because it focuses directly on DApps. In DLP-based constructions, although there exist many up-to-date works, we chose Mao's scheme since it is seen as classical proof in the context of range proofs and has classical OR-composition as a sub-protocol. Since our aim is to analyze the efficiency of Mao's classical range proof on different bases, we modified with OR-composition of Shnorr protocol to base-3. We compute the overall complexity in the context of required exponentiations and the cost of numbers to exchange. Since the cost grows in a pattern, we generalize and formalize the proof for different bases. At the end of these comparisons, we observed that comparing the number of computations in modulo exponentiations with other base approaches, the base-3 approach is 5.5% more efficient. On the other hand, comparing the cost of numbers exchanged between prover and verifier, the base-3 approach is 7% more efficient with respect to other base approaches.

To conclude, we have seen that in the base-3 approach, we get maximum efficiency in both costs of required exponentiations and the cost of numbers to exchange. We have tabulated and graphed the results to have a better look at overall comparisons.

$$\text{DL(P)} > \text{DDH(P)} > \text{SDH(P)} > \text{sRSA(P)}$$

It is obvious that anyone who can solve DL(P) can also solve DDH(P), and sRSA(P) is the easiest one among them. However, this comparison itself is not enough to choose the best range proof construction. Each of them has many advantages and drawbacks in them. For a further comparison on existing range proofs, implementation analysis should be done.

Table 7.1: ZKRP Methods based on their hardness assumptions

Method	Hardness Assumption	Commitment	Range form
Boudot's Scheme	Strong RSA(P)	Fujisaki Okomato C.	integer
Tsai's Scheme	Strong RSA(P)	Fujisaki Okomato C.	integer
Camenish's Scheme	DH(P)	Pedersen C.	u-base
Mao's Scheme	DL(P)	Pedersen C.	bit-wise
Improved Mao's S.	DL(P)	Pedersen C.	u-base

The security of these schemes depends on the different variants. We can say strong RSA assumption is weaker than others so that in general, the square decomposition method is less secure compared with other decompositions. To compare signature base and multi-base decompositions, one can also consider the range we were working on. To choose the best range proof construction, one should consider the range size and the security desire.

Since the development of decentralized systems is of broad and current interest, range proofs are also subject to intense study. So that, they are widely used in terms of privacy-preserving constructions to develop and improve current systems. Still, these methods have many drawbacks and challenges, so that they seem to receive a close review in future research. As future research, instead of finite field DL(P) based OR-proofs, other cryptographic primitives based range proofs can be investigated on different bases in the near future together with a comprehensive comparison to increase efficiency. The construction of multi-base quantum-safe range proofs can also be investigated.

REFERENCES

[1] Hsu university-faculty of computer science-lecture notes on theoretical computer science-interactive proof systems, `https://www.hse.ru/mirror/pubs/share/177376658`.

[2] K. M. Alonso and J. Herrera-Joancomartí, Monero - privacy in the blockchain, IACR Cryptol. ePrint Arch., 2018, p. 535, 2018.

[3] O. Baudron, P.-A. Fouque, D. Pointcheval, G. Poupard, and J. Stern, Practical multi-candidate election system, in *In PODC*, pp. 274–283, ACM Press, 2001.

[4] E. Ben-sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, Zerocash: Decentralized anonymous payments from bitcoin, pp. 459–474, 05 2014.

[5] J. Bethancourt, 2015, `https://people.csail.mit.edu/alinush/6.857-spring-2015/papers/bilinear-maps.pdf`.

[6] D. Boneh and X. Boyen, Short signatures without random oracles, in C. Cachin and J. L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pp. 56–73, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, ISBN 978-3-540-24676-3.

[7] G. Brassard, D. Chaum, and C. Crépeau, Minimum disclosure proofs of knowledge, Journal of Computer and System Sciences, 37(2), pp. 156 – 189, 1988, ISSN 0022-0000.

[8] E. F. Brickell, D. Chaum, I. B. Damgård, and J. van de Graaf, Gradual and verifiable release of a secret (extended abstract), in C. Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, pp. 156–166, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988, ISBN 978-3-540-48184-3.

[9] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, Bulletproofs: Short proofs for confidential transactions and more, in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 315–334, 2018.

[10] J. Camenisch, R. Chaabouni, and A. Shelat, Efficient protocols for set membership and range proofs, volume 5350, 12 2008, ISBN 978-3-540-89254-0.

[11] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, Compact e-cash, in *Proceedings of the 24th Annual International Conference on Theory and Applica-*

tions of Cryptographic Techniques, EUROCRYPT'05, p. 302–321, Springer-Verlag, Berlin, Heidelberg, 2005, ISBN 3540259104.

[12] J. Camenisch and A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pp. 56–72, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, ISBN 978-3-540-28628-8.

[13] J. Camenisch and M. Michels, A group signature scheme based on an rsa-variant, BRICS Report Series, 5(27), Jan. 1998.

[14] S. Canard, I. Coisel, A. Jambert, and J. Traoré, New results for the practical use of range proofs, in S. Katsikas and I. Agudo, editors, *Public Key Infrastructures, Services and Applications*, pp. 47–64, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, ISBN 978-3-642-53997-8.

[15] R. Chaabouni, Enhancing privacy protection set membership, range proofs, and the extended access control, p. 239, 2017.

[16] R. Cramer, I. Damgård, and B. Schoenmakers, Proofs of partial knowledge and simplified design of witness hiding protocols, in Y. G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pp. 174–187, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994, ISBN 978-3-540-48658-9.

[17] I. Damgård and E. Fujisaki, A statistically-hiding integer commitment scheme based on groups with hidden order, in Y. Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002*, pp. 125–142, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, ISBN 978-3-540-36178-7.

[18] I. Damgård, On the existence of bit commitment schemes and zero-knowledge proofs, volume 435, pp. 17–27, 01 1995.

[19] C. Deng, J. Fan, Z. Wang, Y. Luo, Y. Zheng, Y. Li, and J. Ding, A survey on range proof and its applications on blockchain, pp. 1–8, 10 2019.

[20] A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, in A. M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pp. 186–194, Springer Berlin Heidelberg, Berlin, Heidelberg, 1987, ISBN 978-3-540-47721-1.

[21] E. Fujisaki and T. Okamoto, Statistical zero knowledge protocols to prove modular polynomial relations., volume E82A, pp. 16–30, 08 1997, ISBN 978-3-540-63384-6.

[22] O. Goldreich, S. Micali, and A. Wigderson, How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design, volume 263, pp. 171–185, 08 1986, ISBN 978-3-540-18047-0.

[23] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof-systems, in *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, p. 291–304, Association for Computing Machinery, New York, NY, USA, 1985, ISBN 0897911512.

[24] D. Jao and K. Yoshida, Boneh-boyen signatures and the strong diffie-hellman problem, in H. Shacham and B. Waters, editors, *Pairing-Based Cryptography – Pairing 2009*, pp. 1–16, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ISBN 978-3-642-03298-1.

[25] A. Joux, The weil and tate pairings as building blocks for public key cryptosystems, in C. Fieker and D. R. Kohel, editors, *Algorithmic Number Theory*, pp. 20–32, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, ISBN 978-3-540-45455-7.

[26] M. Kim and H. Lee, Experimenting with non-interactive range proofs based on the strong rsa assumption, IEEE Access, PP, pp. 1–1, 08 2019.

[27] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 839–858, 2016.

[28] J. Li, N. Li, and R. Xue, Universal accumulators with efficient nonmembership proofs, volume 4521, pp. 253–269, 01 2007, ISBN 978-3-540-72737-8.

[29] J. Liu, V. Wei, and D. Wong, Linkable and anonymous signature for ad hoc groups, LNCS, 01 2004.

[30] G. M. M. Friedenbach and P. Wuille, Confidential assets main report, tari labs university (tlu).

[31] W. Mao, Guaranteed correct sharing of integer factorization with off-line shareholders, in *Public Key Cryptography, First International Workshop on Practice and Theory in Public Key Cryptography, PKC '98, Pacifico Yokohama, Japan, February 5-6, 1998, Proceedings*, volume 1431 of *Lecture Notes in Computer Science*, pp. 60–71, Springer, 1998.

[32] G. Maxwell and A. Poelstra, Borromean ring signatures , 2015.

[33] R. Metere and C. Dong, Automated cryptographic analysis of the pedersen commitment scheme, pp. 275–287, 05 2017, ISBN 978-3-319-65126-2.

[34] S. Micali, M. Rabin, and J. Kilian, Zero-knowledge sets, in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 80–91, 2003.

[35] I. Miers, C. Garman, M. Green, and A. D. Rubin, Zerocoin: Anonymous distributed e-cash from bitcoin, in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, 2013.

[36] E. Morais, T. Koens, C. Wijk, and A. Koren, A survey on zero knowledge range proofs and applications, 07 2019.

[37] N. Narula, W. Vasquez, and M. Virza, zkledger: Privacy-preserving auditing for distributed ledgers, in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pp. 65–80, USENIX Association, Renton, WA, April 2018, ISBN 978-1-939133-01-4.

[38] L. Nguyen, Accumulators from bilinear pairings and applications, in *Proceedings of the 2005 International Conference on Topics in Cryptology*, CT-RSA'05, p. 275–292, Springer-Verlag, Berlin, Heidelberg, 2005, ISBN 3540243992.

[39] S. Noether, A. Mackenzie, and T. Lab, Ring confidential transactions, Ledger, 1, pp. 1–18, 12 2016.

[40] A. Poelstra, A. Back, M. Friedenbach, G. Maxwell, and P. Wuille, Confidential assets, in A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, and M. Sala, editors, *Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers*, volume 10958 of *Lecture Notes in Computer Science*, pp. 43–63, Springer, 2018.

[41] A. Shamir, Ip = pspace, J. ACM, 39(4), p. 869–877, October 1992, ISSN 0004-5411.

[42] N. Tanaka and T. Saito, On the q-strong diffie-hellman problem., IACR Cryptology ePrint Archive, 2010, p. 215, 01 2010.

[43] Y. C. Tsai, R. Tso, Z. Liu, and K. Chen, An improved non-interactive zero-knowledge range proof for decentralized applications, in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pp. 129–134, 2019.

[44] C. Yun, Building on bulletproofs, 2019, `https://medium.com/@cathieyun/building-on-bulletproofs-2faa58af0ba8`.