DECENTRALIZED SECURE MULTIPARTY COMPUTATION

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

BY

BUSE TAŞCI

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

SEPTEMBER 2019

Approval of the thesis:

DECENTRALIZED SECURE MULTIPARTY COMPUTATION

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

Prof. Dr. Ömür Uğur
Director, Graduate School of **Applied Mathematics** _____

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

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

Assoc. Prof. Dr. Oğuz Yayla
Co-supervisor, **Mathematics, Hacettepe University** _____

Examining Committee Members:

Assoc. Prof. Dr. Ertan Onur
Computer Engineering, METU _____

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

Assist. Prof. Dr. Eda Tekin
Business Administration, Karabük University _____

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: BUSE TAŞCI

Signature :

ABSTRACT

DECENTRALIZED SECURE MULTIPARTY COMPUTATION

Taşcı, Buse
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Murat Cenk
Co-Supervisor : Assoc. Prof. Dr. Oğuz Yayla

September 2019, 44 pages

Advances in blockchain technology have led to new applications that aim to eliminate central systems, to improve transparency and user control in transactions while dealing with the privacy and security challenges. In this thesis, a system which enables users to control their private data and to share chosen data to other parties using secure computation techniques is reviewed. Then, we present a blockchain-based digital identity system depending on this architecture. This system ensures that identity information is shared and transmitted confidentially in accordance with their own consent.

Keywords: blockchain, secure multiparty computation, digital identity

ÖZ

MERKEZİ OLMAYAN GÜVENLİ ÇOK PARTİLİ HESAPLAMA

Taşcı, Buse
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Murat Cenk
Ortak Tez Yöneticisi : Doç. Dr. Oğuz Yayla

Eylül 2019, 44 sayfa

Blokzincir teknolojisindeki gelişmeler, bir yandan gizlilik ve güvenlik zorluklarıyla uğraşırken diğer yandan yapılan işlemlerde merkezi sistemleri ortadan kaldırmayı, şeffaflık ve kullanıcı kontrolünü geliştirmeyi amaçlayan uygulamaları beraberinde getirmiştir. Bu tezde, kullanıcıların kişisel verilerini kontrol etmelerini ve seçtikleri bilgileri güvenli çok partili hesaplama yöntemlerini kullanarak diğer taraflarla paylaşmasını sağlayan bir sistem anlatılmaktadır. Bu sistemi temel alan blokzincir tabanlı bir dijital kimlik sisteminin nasıl geliştirileceğine dair bir yöntem sunulmaktadır. Bununla birlikte, kimlik bilgilerinin, kişilerin kendi iznine uygun olarak gizli bir şekilde paylaşılmasını ve iletilmesini sağlayan bir sistem tasarlanmıştır.

Anahtar Kelimeler: blokzincir, çok partili hesaplama, dijital kimlik

To my family

ACKNOWLEDGMENTS

I would like to express my special thanks to my supervisor Assoc. Prof. Dr. Murat CENK and my co-supervisor Assoc. Prof. Dr. Oğuz YAYLA for their positive attitude, trusting and encouraging me during my studies. Their valuable guidance helped me all the time while doing the research and writing this thesis.

Also, I am grateful to all my friends. Especially, I would like to express my appreciation to Şeyma Fetvacı for her endless support and understanding not only during my thesis but also during my master education.

I would like to thank my dear colleagues and valuable managers at STM, specifically Seda Okutan, for their contributions while choosing my thesis subject, for being thoughtful at work during my thesis process.

Above all, I would like to express my profound gratitude to my family for their endless support and everything else in my life. My endless thanks go to my father for his support and trust in me through all of my life until his last breath.

TABLE OF CONTENTS

ABSTRACT vii

ÖZ ix

ACKNOWLEDGMENTS xiii

TABLE OF CONTENTS xv

LIST OF FIGURES xix

LIST OF ABBREVIATIONS xxi

CHAPTERS

1 INTRODUCTION 1

2 PRELIMINARIES 3

2.1 Cryptosystems 3

2.1.1 Symmetric Cryptography 4

2.1.2 Asymmetric Cryptography 5

2.1.3 Digital Signatures 5

2.2 Secret Sharing 6

2.2.1 Shamir's Secret Sharing Scheme 7

2.2.2 Verifiable Secret Sharing(VSS) 8

2.3 Secure Multiparty Computation 9
2.3.1 The Multi-Party Case of Secure Multiparty Computations 10
2.3.1.1 Addition Gates 11
2.3.1.2 Multiplication Gates 11
2.3.2 The Multi-Party Case of Secure Multiparty Computations for Malicious Adversaries 12
2.4 Randomness Beacons 13
3 BLOCKCHAIN 15
3.1 Technical Details 15
3.1.1 Transactions and Blocks 16
3.1.1.1 Smart Contracts 17
3.1.1.2 Cryptographical Identity 17
3.1.2 Distributed Ledger Technology 18
3.1.3 Types of Blockchain 18
3.1.4 Consensus Mechanisms 20
3.2 Use Cases 21
3.2.1 Financial Systems 21
3.2.2 Digital Identity 21
3.2.3 Supply Chain 21
3.2.4 Healthcare 22
4 DECENTRALIZED SECURE COMPUTATION 23

4.1 Computation 26
4.1.1 Quorum Selection 27
4.1.2 Multiparty Computation(MPC) 27
4.2 Enigma 31
4.2.1 Design 31
4.3 Security Analysis 34
5 DIGITAL IDENTITY 35
5.1 History of Digital Identity 35
5.2 Blockchain Based Digital Identity System 37
6 CONCLUSION 41
6.1 Future Work 41
REFERENCES 43

LIST OF FIGURES

Figure 2.1 A graphic of an arithmetic circuit 11

Figure 3.1 Content of a Block 16

Figure 3.2 Diagram of Blocks 18

Figure 4.1 Registration of Parties 24

Figure 4.2 Data Storage 25

Figure 4.3 Giving Access to Parties 26

Figure 4.4 Start of Computation 26

Figure 4.5 Computation Protocol 28

Figure 4.6 Design of Enigma 31

Figure 5.1 Self Sovereign Identity 38

LIST OF ABBREVIATIONS

ACL	Access Control List
AES	Advanced Encryption Standard
BFT	Byzantine Fault Tolerance
CA	Certificate Authority
DES	Data Encryption Standard
DSA	Digital Sgnature Algorithm
DHT	Distributed Hash Table
ECDSA	Elliptic Curve Digital Signature Algorithm
HTTP	HyperText Transfer Protocol
HTTPS	Secure HyperText Transfer Protocol
LFSR	Linear Feedback Shift Register
MPC	Multiparty Computation
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PoA	Proof of Authority
PoW	Proof of Work
POS	Proof of Stake
RC4	Rivest Cipher 4
RSA	Rivest Shamir Algorithm
VSS	Verifiable Secret Sharing
PVSS	Publicly Verifiable Secret Sharing

CHAPTER 1

INTRODUCTION

The Earth has been in constant change and development since its existence. People also exist in this process and keep up with this change. As people change, demands change, and technology progresses day by day. New technologies are radically transforming our working lives as well as the tools and ways of life we use in daily life. As the digitization moves accelerate, businesses are revolutionizing the way they work by integrating advanced technologies into their workflows. Blockchain is one of the these revolutionary transformation as emerged.

In 2008, Bitcoin has come into our lives with a white-paper published by Nakamoto [12] introducing a decentralized peer-to-peer electronic money system and thus we have met with blockchain technology. Although almost all of the people think that Bitcoin and blockchain are the same, Bitcoin is just one of the platforms of blockchain technology. Blockchain technology is a distributed ledger technology allowing parties to transmit their data without a central authority.

In recent years, Blockchain technology has led up to many applications. Companies/individuals have developed many platforms according to their needs and created their own blockchain infrastructures, starting with Bitcoin and evolving into many platforms such as Ethereum [24], Hyperledger [23], Monero [16].

The most important problem of Bitcoin is privacy that companies/individuals deal with. In addition, since blockchain technology does not depend on a central authority, the needs of privacy and security should be managed by the system itself. Thus, it is required to use cryptographical techniques in their platforms. Companies/indi-

viduals who wish to design privacy priority applications, they must use one of the privacy protocols on their platforms such as zero knowledge protocols, secure multiparty computation algorithms, fully homomorphic encryption etc.

In this thesis, we investigate the usage of secure multiparty computation on blockchain technology to provide data privacy in the applications. In Chapter 2, we give a summary for cryptographical background of traditional cryptosystems and details of some algorithms which are used in the next chapters. In Chapter 3, we focus on blockchain technology. We give a formal definition of blockchain, technical details and some popular use cases of this technology. In Chapter 4, we move on decentralized secure computation protocol which is the main subject of this thesis. We explain how multiparty computation implemented and used with blockchain technology in detail. Also we represent Enigma protocol, an application of decentralized secure computation protocol, in this chapter. In Chapter 5, we introduce a blockchain-based digital identity system. The most private data that people have is their credentials. So, for countries or individuals, it is crucial to have privacy enhanced identity systems allowing users to control their credentials themselves. Finally, in chapter 6, we conclude our studies and give some future works.

CHAPTER 2

PRELIMINARIES

2.1 Cryptosystems

Cryptosystems are applications of cryptographic methods to provide security for information systems. Some basic terms of cryptosystems are:

- *Plaintext:* It is the original message or data which is used as an input for encryption algorithm and output for decryption algorithm.
- *Ciphertext:* It is the coded form of plaintext produced as an output of encryption algorithm and used as input for decryption algorithm.
- *Encryption:* It is the process used to convert plaintexts to ciphertexts by doing several substitutions and permutations on the plaintext.
  Encryption is one of the most important subject of electronic communication, but it does not imply that a message remains entirely secure. The usage of digital signatures and hash functions support authenticity and integrity of the encrypted message.
- *Decryption:* It the process used to reconstruct plaintexts from ciphertexts.
- *Encryption key:* It's a value that the sender knows. To calculate the ciphertext, the sender uses this key as an input to the encryption algorithm together with plaintext.
- *Decryption key:* It's a value that the receiver knows. Similar with encryption

key, to calculate the plaintext, the receiver uses this key as an input to the decryption algorithm together with ciphertext.

There exists two types of cryptosystems, symmetric and asymmetric cryptosystems, designed to provide main goals of cryptography which are confidentiality, data integrity, availability and non-repudiation. Confidentiality can be provided by encryption and decryption schemes. Data integrity can be provided by using hash functions. Non-repudiation and authentication can be provided by asymmetric cryptosystems.

2.1.1 Symmetric Cryptography

Symmetric key cryptosystems are cryptosystems transforming plaintexts to ciphertexts by using and encryption algorithm and a secret key. Encryption and decryption keys are known by communication parties. These keys are usually same, or they are derived from each other.

For a secure symmetric system it is important to have a strong encryption algorithm. An adversary knowing the encryption algorithm and some ciphertexts should not be able to decrypt the ciphertext or reveal the key. The key should be distributed to communicating parties in a secure channel. If anyone reveals the key, then all communication will be readable [21].

In this type of systems, algorithms are usually fast and they can be used easily in hardware and software.However, key distribution among parties, achieving authentication and data integrity are difficult.Symmetric key cryptosystems consists of two types: stream ciphers and block ciphers.

Stream ciphers: Each digit in plaintexts is encrypted with a randomly generated key string. Each encryption step depends on the previous one. LFSRs are well known examples to generate random key strings. The most known stream ciphers are RC4 and A5. ChaCha is now becoming the most commonly used software stream cipher [22].

Block ciphers: Plaintexts are encrypted in blocks. They divide the text into blocks of a certain length (64-bit, 128-bit, etc.) and convert it into a new block by processing it

by using the generated keys. They are designed to prevent the direct relationship of letters to each other. The most known block ciphers are DES and AES.

2.1.2 Asymmetric Cryptography

Asymmetric cryptosystems base on some mathematical hard problems such as integer factorization problem, discrete logarithm problem, short vector problem while symmetric systems do not have an exact mathematical explanations.

These systems are designed to eliminate key distribution and non-repudiation problems holding on symmetric systems.

In this type of systems, for each party involved in the communication, there exists a key pair, public key and private key. Encryption and decryption processes is done through these keys. For this reason asymmetric cryptography is also known as public key cryptography. Since the public key is transmitted to everyone, the problem of key distribution in a secure way is eliminated. The relationship between the keys should not be established. Achieving authentication and data integrity is easy. Algorithms run slowly compared to symmetric cryptosystems because of the key length and heavy computations [10].

These systems can be used for encryption, key establishment and identification by using digital signatures. To encrypt messages, RSA,Elgamal algorithms can be used. For establishing keys over an insecure channels Diffie Hellman Key Exchange algorithm and RSA key transport protocols can be used. Digital signature algorithms like RSA, DSA or ECDSA can be used to provide non-repudiation and identification [13]. However, these systems run slowly since they include intensive mathematical operations, they are commonly used for digital signatures instead of encryption schemes.

2.1.3 Digital Signatures

With the advance of technology, the need to use electronic signatures instead of ink-signatures arises. Digital signature algorithms have been developed in order to enable the parties to authenticate each other and not to deny that s/he sends the message.

The fundamental concept is that the person signs the document by using his/her private key, and the receiver uses the corresponding public key to check who sent the document.

Hash Functions:

Cryptographic hash functions are functions using arbitrary length bit string as an input to generate a fixed length bit string as an output called *hash value* [20].

A hash function is a one-way function and satisfying following conditions:

- *Pre-image resistant:* Given a hash value $h(m)$ of a message m, it is computationally infeasible to find the message m.
- *Second pre-image resistant:* Given a message m_1, it is computationally infeasible to find another message $m_2 \neq m_1$ satisfying $h(m_1) = h(m_2)$.
- *Collision resistant:* It is computationally infeasible to find the same hash value for different messages. In other words, $h(m_1) \neq h(m_2)$ where $m_2 \neq m_1$.

In addition, hash functions cannot be used for encryption since it is also infeasible to decrypt. They can be used to check data integrity and verification of data. In digital signatures, hash of the data is signed and private key is used to encrypt this hash.

2.2 Secret Sharing

Secret sharing systems in cryptography are the systems distributing shares of a specified secret between a set of trusted parties. This secret may be a very valuable piece of data that may need to be kept private and secure. Each share is a meaningless piece of data on its own until the parties reconstructing the secret when they come together and making it obvious. Secret sharing mechanisms is used to access keys to accounts including extremely sensitive data.

A commonly used secret sharing scheme is (n, t)-threshold scheme. In such a scheme, a secret message m is divided among n parties, giving each party P_i its own unique piece (s_i). The shares are splited and distributed by a dealer. For the number of shares

$k \geq t$ where t is the threshold, the secret is reconstructed by k parties. The dealer realises this by providing each player a share so that any group of T or more parties can reconstruct the secret together but no group of less than T players can [19].

2.2.1 Shamir's Secret Sharing Scheme

Shamir's scheme is the first secret sharing scheme proposed in 1979 [18]. It depends on the idea of polynomial interpolation using Lagrange coefficients the interpolation of a given dataset by the smallest possible degree polynomial that goes through the points. This system builds an $t-1$-degree polynomial f, where t is the enough number of players for reconstruction of m. The algorithm works in following steps:

- A random $t-1$-degree polynomial and the number t is selected by the dealer such that $f(0) = m$, in which m is the message that must be kept secret and t is the number of parties to reconstruct the secret.
- The pairs $(x_i, f(x_i))$ where $x_i \neq 0$ is distributed for each party P_i where $i = 1, 2, ..., n$.
- At least t parties can reconstruct the polynomial f with these pairs. Any subset with $t-1$ parties can not reconstruct.
- For reconstruction, these parties use the Lagrange Interpolation method.
- To achieve the secret, $f(0)$ can be computed [7].

Example:
Assuming these $m = 1234,\ n = 6$ and $t = 3$, the polynomial will be a second degree polynomial such that $f(x) = ax^2 + bx + m$. Then the dealer selects a and b randomly, saying $a = 94$, $b = 166$. So,

$$f(x) = 94x^2 + 166x + 1234.$$

The dealer distributes the pairs to each party:
$P_1 : (1, 1494)$ where $f(1) = 1494$.
$P_2 : (2, 1942)$ where $f(2) = 1942$.

$P_3 : (3, 2578)$ where $f(3) = 2578$.
$P_4 : (4, 3402)$ where $f(4) = 3402$.
$P_5 : (5, 4414)$ where $f(5) = 4414$.
$P_6 : (6, 5614)$ where $f(6) = 5614$.

Let's choose any three of them: say; P_2, P_4, P_5. Then, compute Lagrange polynomials such as:

$$l_0 = \frac{x - x_1}{x_0 - x_1} \times \frac{x - x_2}{x_0 - x_2} = \frac{x - 4}{2 - 4} \times \frac{x - 5}{2 - 5} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}$$

$$l_1 = \frac{x - x_0}{x_1 - x_0} \times \frac{x - x_2}{x_1 - x_2} = \frac{x - 2}{4 - 2} \times \frac{x - 5}{4 - 5} = -\frac{1}{2}x^2 + \frac{7}{2}x - 5$$

$$l_2 = \frac{x - x_0}{x_2 - x_0} \times \frac{x - x_1}{x_2 - x_1} = \frac{x - 2}{5 - 2} \times \frac{x - 4}{5 - 4} = \frac{1}{3}x^2 - 2x + \frac{8}{3}$$

Then, by using Lagrange interpolation, each party have:

$$f(x) = \sum_{i=0}^{2} y_i \times l_i(x) = 94x^2 + 166x + 1234.$$

Now, all computing parties can evaluate $f(0) = 1234$, thus they get the secret correctly.

2.2.2 Verifiable Secret Sharing(VSS)

Verifiable secret sharing was first proposed in 1985 by Chor,Goldwasser, Micali and Awerbuch as a cryptographic protocol that enables to split a secret into n shares and distribute these shares to n parties so that the secret can be reconstructed by only enough parties. The difference from Shamir's scheme is that this scheme allows to verification of the shares of the secret without obtaining any information about what the secret is [4].

In publicly verifiable secret sharing(PVSS), which is more suitable for blockchain applications, parties can verify not only their own shares but also the parties who received correct shares. Each party release proof of correctness for each share. The

security of communication is provided by public key cryptography [2]. The system works in two phases: distribution and reconstruction.

Distribution:

- A random $t-1$-degree polynomial in $\mathbb{Z}_p$ is chosen by the dealer.

$$p(x) = s + \alpha_1 x + \alpha_2 x^2 + ... + \alpha_{t-1} x^{t-1}$$

  where the secret $s = \alpha_0$.

- The dealer sends shares $s_i = p(i)$ to each party P_i for $i = 1, ..., n$ by enrcrypting each share with the public key of related party.

- The dealer also broadcasts the commitments of shares $C_j = g^{\alpha_j}$ for $j = 1, ..., t-1$ where g is the generator of a cyclic group of order p.

- When s_i's are received, each party verifies validity of the share by computing

$$g^{s_i} = \prod_{J=0}^{t-1} C_j^{i^j}.$$

Reconstruction:

- Each party decrypts and finds s_i its share by using its private key.

- Similar with Shamir's scheme, using t valid shares, the secret $p(0) = s$ is obtained [2, 17].

For secure multiparty computation protocols, VSS schemes play an important role. Multiparty computation is typically done by creating secret input shares and manipulating the shares to compute some function [15].

2.3 Secure Multiparty Computation

Secure multi-party computation is a cryptographic field that wishes to produce methods that allow different parties to compute a function by using their private inputs. Just the result of this function is published and the parties learn nothing more than

their own input apart from anything that can be learned from the result.
In the general case we have n parties $P_1, P_2, ..., P_n$ having their private inputs $x_1, x_2, ..., x_n$. The aim is to compute a function $f(x_1, x_2, ..., x_n)$ where any of x_i's has no meaning for the party P_i. After the computation, the result is returned to all parties.

For secure computation protocols there exist two needs [7]:
Privacy: All parties should not learn anything except from the result.
Correctness: Every party should compute the correct result and an adversary should not be able to change the result.

Secure multi-party computation protocols generally depends on two different approaches. The first one is garbled circuits which is the idea of Yao which is generally used for two-party cases. The second approach is arithmetic circuits generally used for multi-party cases using secret sharing schemes which our works based on [19].

2.3.1 The Multi-Party Case of Secure Multiparty Computations

The multi-party case depends on arithmetic circuits which are evaluated by using secret sharing schemes. An arithmetic circuit combines a finite field and a function which can be computed by execution of addition and multiplication gates over this finite field [19].

Shamir's secret sharing scheme is used for sharing inputs. The value of share x_i is distributed among all parties $P_i's$ with each j^{th} party receiving a $x_i^{(j)}$ share. Clearly, if enough parties come together, then the features of the secret sharing system can determine the value of x_i.

At the beginning of the protocol, each player can create shares of their own input values and send a share to others.

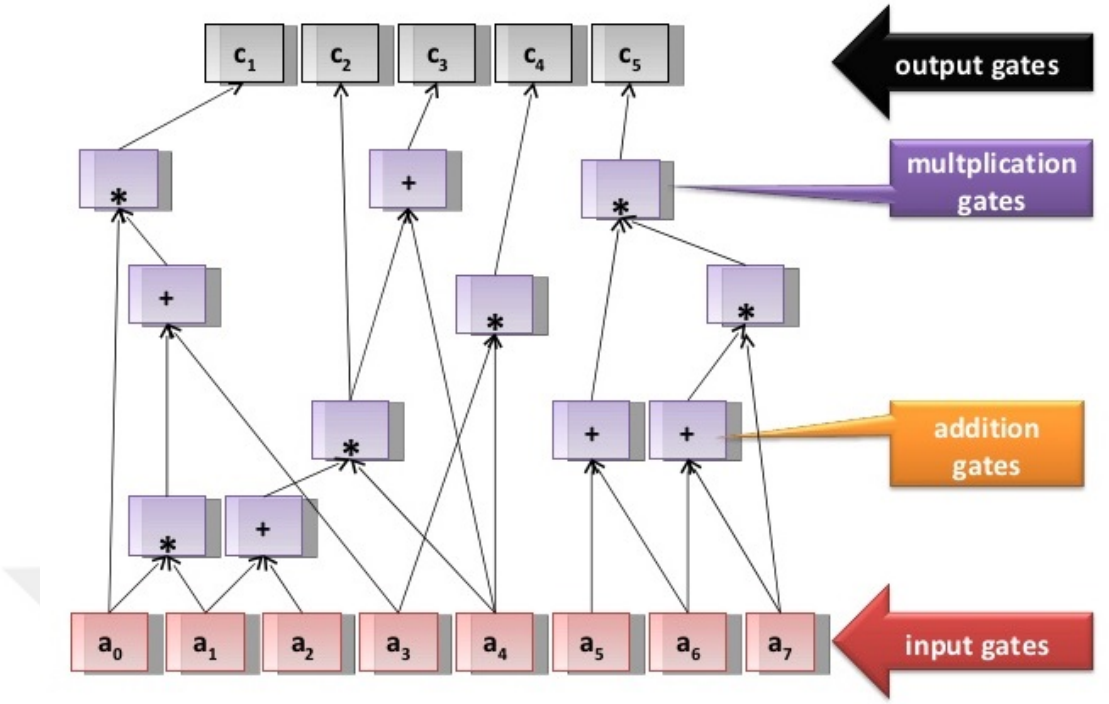

Figure 2.1: A graphic of an arithmetic circuit [5]

2.3.1.1 Addition Gates

Suppose there exist two secrets a and b shared using these polynomials:

$$f(X) = a + f_1X + f_2X + ... + f_tX^t$$

and

$$g(X) = b + g_1X + g_2X + ... + g_tX^t.$$

Now, each i^{th} party has $f(i) = a^{(i)}$ and $g(i) = b^{(i)}$, then

$$c^{(i)} = h(i) = f(i) + g(i) = a^{(i)} + b^{(i)}.$$

Thus, the parties can calculate a share of an addition gate without any communicating with others.

2.3.1.2 Multiplication Gates

Computation of an output of a multiplication gate is based on Lagrange Interpolation: if $f(X)$ is a polynomial and $f(i)$ values are distributed then there is a recombination

vector $(r_1, r_2, ..., r_n)$ such that

$$f(0) = \sum_{i=1}^{n} r_i \cdot f(i).$$

Similar to addition computations, each party should compute $c^{(i)} = h(i)$ such that $h(0) = c = a \cdot b$ for the secrets $a = f(0)$ and $b = g(0)$ where $a^{(i)} = f(i)$ and $b^{(i)} = g(i)$. To get $h(0)$, all parties realizes the following steps [19]:

- $d^{(i)} = a^{(i)} \cdot b^{(i)}$ is computed.
- At most t-degree polynomial $\sigma_i(X)$ such that $\sigma_i(0) = d^{(i)}$ is produced.
- Each i^{th} party sends to j^{th} party $d_{i,j} = \sigma_i(j)$.
- Each j^{th} party calculates $c^{(j)} = \sum_{i=1}^{n} r_i \cdot d_{i,j}$.

2.3.2 The Multi-Party Case of Secure Multiparty Computations for Malicious Adversaries

Against malicious adversaries, the above protocol is not secure because of the ability of an adversary to produce an invalid output of multiplication protocol. In order to achieve a secure protocol against malicious adversaries, malicious parties should be identified from errors. The protocol works in following steps [19]:

The protocol begins with pre-processing phase:

- Similar with the above protocol, $a^{(i)}$ and $b^{(i)}$ of degree t is generated.
- Also, a random sharing $r^{(i)}$ of degree t and $r^{(i)}$ of degree $2t$ of zero are generated.
- Then, each party calculates $s^{(i)} = a^{(i)} \cdot b^{(i)} - r^{(i)} + z^{(i)}$.
- The values $s^{(i)}$ are broadcast by the parties and they try to get s.
- Then, Reed-Solomon error detection can be used to detect malicious parties if the number of them limited by $t < n/3$. When an error is found, the protocol is aborted.

- The shares $c^{(i)}$ from $c^{(i)} = s + r^{(i)}$.

Now, by assuming the inputs $x^{(i)}$ and $y^{(i)}$ of multiplication gate, parties try to calculate $z^{(i)}$ of $z = x \cdot y$. The multiplication phase:

- $d^{(i)} = x^{(i)} - a^{(i)}$, $e^{(i)} = y^{(i)} - b^{(i)}$ are calculated locally and broadcast by the parties.
- $d = x - a$ and $e = y - b$ are reconstructed.
- Finally $z^{(i)} = d \cdot e + d \cdot b^{(i)} + e \cdot a^{(i)} + c^{(i)}$ and so $z = x \cdot y$ is computed.

2.4 Randomness Beacons

A cryptographic beacon is a service ensuring a random public source. A new random data is produced regularly and if everyone accepts that there is no way of determining the next beacon output, they can rely on it as a reasonable random value supplier [9]. Some security properties should be achieved by these beacons:

- *Unpredictability:* It is difficult to estimate any information about the beacon before it is published. It can not be altered.
- *Objectivity:* The record should be mathematically similar to a random string.
- *Sampleability:* All parties have free access to it when a beacon record has been released.
- *Verifiability:* Before the time the beacon record is released, the source of randomness, from which a beacon record is sampled, can be verified as unknown to any party.

CHAPTER 3

BLOCKCHAIN

Although the blockchain technology is known thanks to Bitcoin implemented by Nakamoto in 2008 [12], its initial point traces to Chaum's work [3]. From Chaum to Nakamoto, their aim is to perform financial transactions through a system without third parties. In 2008, Nakamoto designed and developed the first blockchain application from which the technology evolved and found its way into many applications beyond digital currencies. Blockchain technology that is a distributed ledger technology where central trust is distributed over the Internet enables the transfer of valuable assets or data by removing a central and trusted authority.

3.1 Technical Details

In this technology, transactions are collected and announced individually in different data structures called blocks, which are cryptographically connected together, copied and distributed in a peer-to-peer network to avoid manipulation of previously announced transactions. The transactions are embedded into blocks validated by the nodes that are running the blockchains peer-to-peer client. All parties in the system stores the copy of the ledger; validation and immutability are provided thanks to ledger technology.

The main components of blockchain technology are transactions embedded into the blocks, distributed ledgers and consensus mechanisms.

3.1.1 Transactions and Blocks

A blockchain system is composed of transactions and blocks. Transactions are updates of the blockchain including the signature of its sender. For each transaction, changes are committed to the system. They are queued in the transaction pools till they are approved and when they are approved, they are broadcast and then embedded to blocks by miners who produce the blocks by solving a hash puzzle.

Transactions can contain executable codes which are scripts to spent outputs, smart contracts to set permissions related with outputs and some timestamped random data [11].

To transfer a transaction, sender shares his/her blockchain address and specifies the address of receiver and the amount of currency s/he wants to sent.

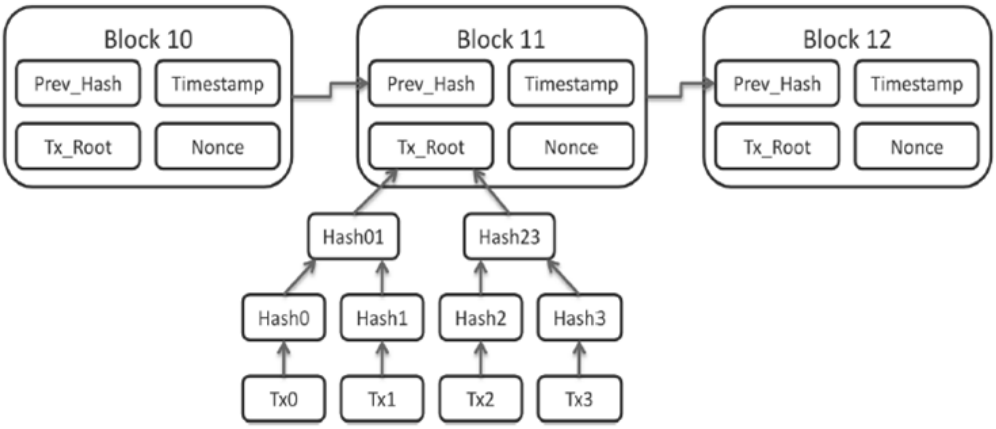

Figure 3.1: Content of a Block [12]

Blocks on the other hand, contains an ordered branch of transactions stored with their hash values as shown in Figure 3.1. They are produced by processing of transactions by the miners. Blocks are immutable by means of time stamps. Each block can be considered as a paper of the ledger. Blocks are added to the chain according to hash value of the previous one. Depending on the used/designed platform, dimensions and contents of blocks may vary. For example, Bitcoin has 1MB block-size including 4-7 transactions while Ethereum does not have a strict block-size, it includes 15 transactions in average.

3.1.1.1 Smart Contracts

The idea of smart contracts was proposed by Nick Szabo, a cryptographer, in 1994 to use computer programs that provide secure and coherent interactions between various parties. Afterwards, in 2014, by Vitalik Butterin, this idea was applied to Ethereum because of the need of Bitcoin transactions to be accelerated and programmable [24]. A smart contract can be defined as an immutable computer program running on a blockchain network and a set of rules agreed by the parties involved [14]. In other words, smart contracts are small computer programs that include predefined logical codes stored,replicated and validated on a distributed, decentralized platform and lead to updates on its located platform. They are written after the related parties came to the agreement(set the conditions) and then they are signed and uploaded to the blockchain network. Uploaded contracts can have some interactions such as a start of a transaction, a transfer of any data with all other components. When included conditions are arose, they execute the agreements itself automatically.

3.1.1.2 Cryptographical Identity

User accounts depend on asymmetric-key cryptography and enable other nodes to sign transactions. All parties has a key pair to sign their transactions. Private key provides that each party is responsible for its own account. After a transaction is started, the signature and whether there is enough money in the account or not are checked by the other parties in the validation processes. Keys are produced by the wallets/accounts:

- Each wallet gathers information about its owner and uses these information to generate an-ECDSA private key.
- Then, the random coordinates on the selected elliptic curve is collected to produce a public key related with this private key.
- Finally, by using this public key, the blockchain address is generated with chosen hash algorithms.

The whole system works as in the following figure 3.2:

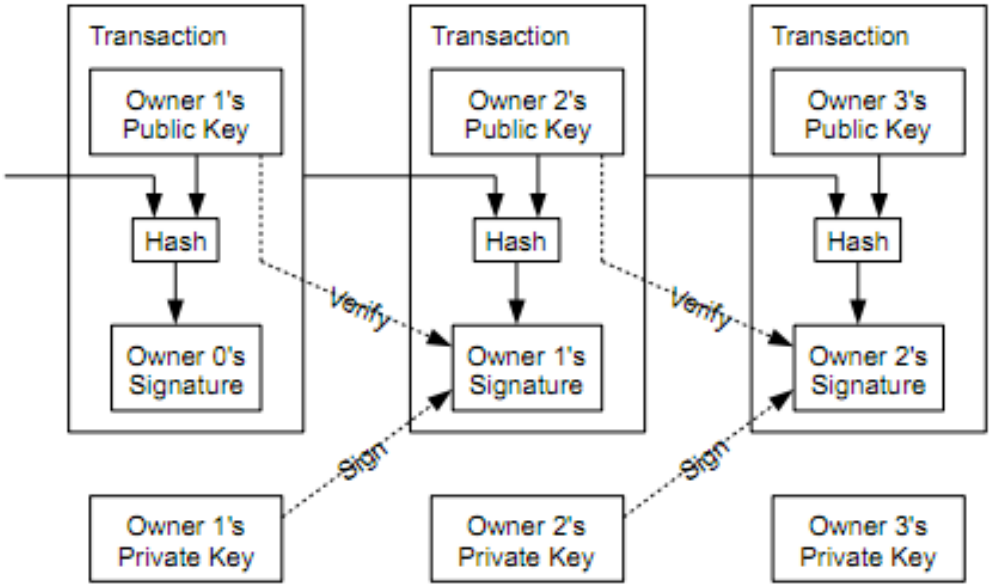

Figure 3.2: Diagram of Blocks [12]

3.1.2 Distributed Ledger Technology

A decentralized database distributed through various nodes/parties can be considered as a distributed ledger. This database is not managed by a central authority or server unlike all nodes have identical authority status. Each node will sustain the ledger and if there are any data modifications, the ledger will be updated. The update exists at each node directly.

Blockchain is a form of distributed ledger where each node is provided its own ledger copy. In each time a new transaction is transmitted, all of the copies are updated. Before recording transactions to the ledger, they are encrypted and signed. Blockchain coordinates the data according to the blocks where these blocks are connected to each other.

3.1.3 Types of Blockchain

Public Blockchain: In this type of blockchain network is accessible by all nodes where a central management does not exist. Every node can download and start to execute the code a on their local device, validate and read transactions, and thus they

take place in the consensus process such as Proof of Work (PoW) or Proof of Stake (PoS). Also, they will be able to send transactions over the network and hope to see these transactions included in blocks if they are valid.

Despite the network being accessible by all of the nodes, the identity of the parties is pseudonymous. The anonymity of parties is a mixed blessing issue, as malicious transactions might threaten to the reputation of the network. Bitcoin, Ethereum, Monero and almost any other most known digital currencies are examples of public ones.

Private Blockchain: This type of blockchain is a permissioned, closed distributed ledger system controlled by an individual or an organization. Only a custodian on the network can read, write, and audit the blockchain. The custodian can also enable permissioned access to determined nodes only for transactions to be validated and read. These networks are usually applied to internal systems of private companies. Because of centralized and exclusive use of this type, the aim of the initial blockchain technology where decentralization is its key offering is defeated. However, these networks are centralized with regards that permissions are given to a central reliable entity. In this type of system, it is difficult to manipulate data and easy to validate transactions which makes system faster and more cost-effective. Ripple, Monax, and Hyperledger are some examples of private ones.

Federated Blockchain: In this type of system, instead of one custodian,there exists a group enabling access to pre-selected nodes to read, write, and audit the blockchain. Only consortium members can perform, validate and review transactions. Consensus is reached via an algorithm of voting or multi-party consensus whose rules focus on the approvals of the parties. The same advantages with private ones such as cost-effectiveness and privacy are also provided in this type of blockchains. R3s Corda and EWF are examples of this type.

Hybrid Blockchain: This type of blockchain can be described as the blockchain which tries to use advantages both private and public blockchain alternatives. In other words, a hybrid blockchain will imply regulated access and independency simultaneously. The hybrid blockchain is distinguishable from not being accessible by everyone, but it still provides blockchain characteristics such as integrity, transparency, and security. Who will be able to participate in the system and which transactions will be

made public are determined by the people involved in the hybrid blockchain.

3.1.4 Consensus Mechanisms

A consensus mechanism is a system of rules determining the contributions by the parties of the blockchain. There are different types of consensus mechanisms working on different principles. [6]

Proof of Work (PoW): In this type of consensus mechanism, miners, who produces the blocks, solve a hard problem to add the block to the system and the others validate this block. The problem is to find an acceptable nonce value that results desired hash value of the current block, e.g. at least 32 bits of leading zeros in the hash value. Each block contains a difficulty-degree and the difficulty increases for next blocks. An attacker who wants to capture the system should have more computational power than total computational power of half of the nodes in the system to overcome this challenge.

Longest Chain Rule: This rule is designed to keep the system running on a single chain and to avoid forks. To validate an added new block, it is required to wait to next six blocks to guarantee the longest chain. After validation of six blocks, the transactions are realised.

Proof of Stake (PoS): The proof of stake (PoS) algorithm is designed to an alternative for PoW algorithm to reduce the energy consumption and high costs. Within this approach, the block production and validation approval mechanism is associated with the stake of the party who produces the next block on the network. In such systems, all the amount of cryptocurrencies is produced at beginning, and then the parties in the system take their stakes in terms of their investments. The amount of stakes is calculated according to the amount of owned cryptocurrencies.

Byzantine Fault Tolerance (BFT): In this mechanisms, there exist validators to validate transactions. Each party checks the incoming transactions by using the data structure holding on itself and distributes it with the network by signing an approved transaction. If a transaction is approved by a certain number of parties, it is considered to be valid. In the scope of this mechanism, all validators included in the network

should be aware of others and for any other party that wants to become a validator, should be accepted by a central structure. This contradicts with the decentralization of public blockchain. Therefore, this consensus approach is more suitable for private blockchains like Hyperledger.

Proof of Authority (PoA): In this type, pre-defined parties, considered as admins, validate transactions. The other parties trust these authorized nodes and believe their reliability. This mechanism is also designed for private/permissioned blockchains.

3.2 Use Cases

3.2.1 Financial Systems

Blockchain technology attracts the attention of the banking and financial system more than the businesses and consumers allowing them to create value, to exchange and to trade in an environment where the system itself guarantees security, without the need of any central institution.

3.2.2 Digital Identity

Identity solutions used today generally depend on storage of identity information in a centralized structure and controlled presentation of it to the external services. However, this approach does not meet the total needs of the digital world and the people. With the usage of blockchain technology which behaves according to requirements of the person and controls data flows according to these behaviours can be created. We will investigate a blockchain-based identity system in Chapter 5.

3.2.3 Supply Chain

Supply chain scenarios may be the most applicable and convenient applications for blockchain. The problem of transparency is one of the concerns in the management of supply chains. Blockchain technology helps to improve transparency, enables to prevent fake products and theft, increases legal compliance, decreases documentation

and considerably decreases costs. In addition, blockchain technology encourages the documentation of a product in real time moving from its initial point and all its touch points. From the point of view of an end user, it can empower consumers to figure out exactly whether products are not changed.

3.2.4 Healthcare

Another popular use case for blockchain technology is healthcare scenarios. In current schemes, information about patients is held across different institutions in old databases in different formats and standards, sharing of the data unsuitable for the expectations of patients. By using blockchain technology, information of patient can be recorded on a distributed ledger allowing conditional access to distinct parties. Thus, any activity with the medical data of the patient can be reported as a transaction with a time-stamped electronic record on a ledger. This can remove bottlenecks with existing data management methods, and provides patients with greater control over their own health data.

CHAPTER 4

DECENTRALIZED SECURE COMPUTATION

For any cryptographic system, privacy and integrity of data are the most crucial points. Decentralized secure computation system, behaves like a distributed cloud, allows data control and to produce correct result without viewing real data by using multi-party computation techniques. This ensures privacy and data integrity. In this chapter we review some existing mechanisms for decentralized secure computation, for details see [26] and references therein.

There are some entities with multiple roles in the system:

- *Owners:* These nodes are the owners of data and they are responsible for sharing data to the system and control who can query it. They can be considered as input parties.
- *Services:* These nodes are responsible querying data without learning anything except from the result. Also, they can be considered as output parties or clients.
- *Parties:* These nodes as constructors of the cloud, are responsible for providing computational and storage resources. They can be considered as general-purpose miners, who are prepared to do computational work.

In addition to these roles, blockchain is the key component of the inner cloud as a single party that offers correctness, incentives and synchronized time. It is also responsible for book-keeping, identifying behaviours of parties and achieving consensus. The system based on returning reward or punishment in terms of the actions of parties. Payments for good behaviour or penalizing for malicious one is also a

mission of blockchain.

Registration

Registration is the starting point of the system in which all computation parties register to the cloud by paying the deposit and link to others to create the distributed cloud, a blockchain network.

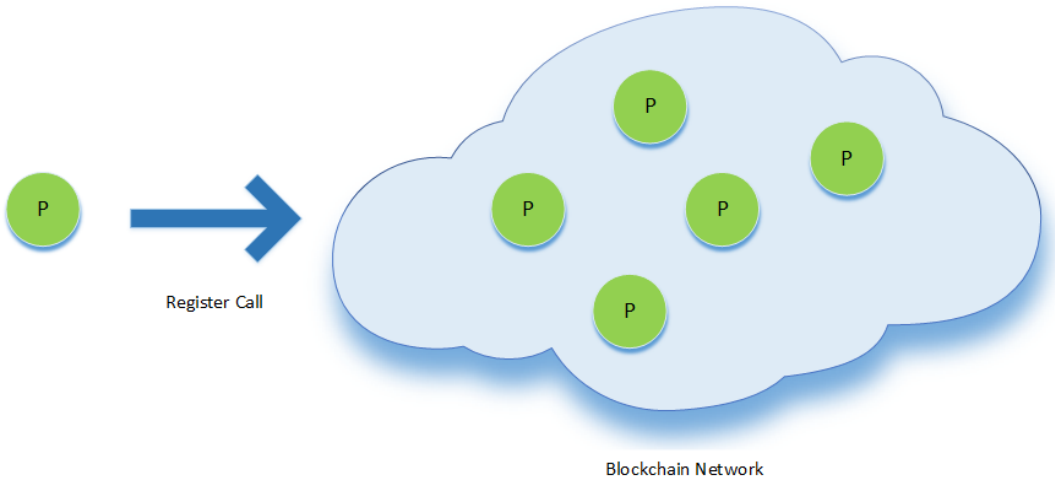

Figure 4.1: Registration of Parties

The security and efficiency of system depend on the security deposit account which is used to penalize a malicious party for its cheating. Only parties who have at least total amount of deposit can register to the system to do computations.

- In the initialization, global timing and payment parameters are specified.
- Then, a register call is sent by the party who wants to be a part of computation process.
- It is checked that whether the party has total amount of deposit or not.
- If this party does not exist in the system already, it is added to the system.
- The amount of security deposit at the account of this party is locked.

Share and Access

Owners store data and set permissions in this part of the system. Blockchain is used to log policies for access-control.

- Approved list of public keys of services is sent to blockchain by the owner. An access call received from the owner can be done for change the list if it is required. The evaluation function can only be sent by an approved service.
- Storage:
  - Owner sends store request to the blockchain.
  - A contract executed by the blockchain randomly selects a quorum, a set of registered parties, to store data.
  - Each share is encrypted by using the public key of related party.
  - The set of these encrypted shares and the commitments of these are sent to the blockchain.
  - In some period of time, a re-sharing protocol is triggered.

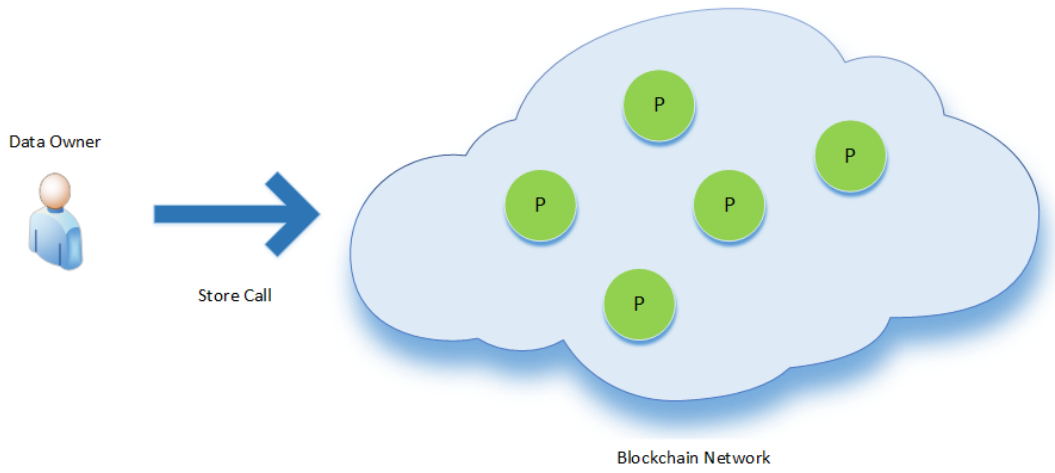

Figure 4.2: Data Storage

- Re-sharing: It is used to recover data in the event that data is lost. With help of re-sharing, data does not stand too long in one quorum which has the possibility of collapse.
  - In order to store the data, a replacement quorum is selected.
  - Shares are transferred to this new quorum.
- Access:
  - The owner gives information to the parties about who can query or compute the data.

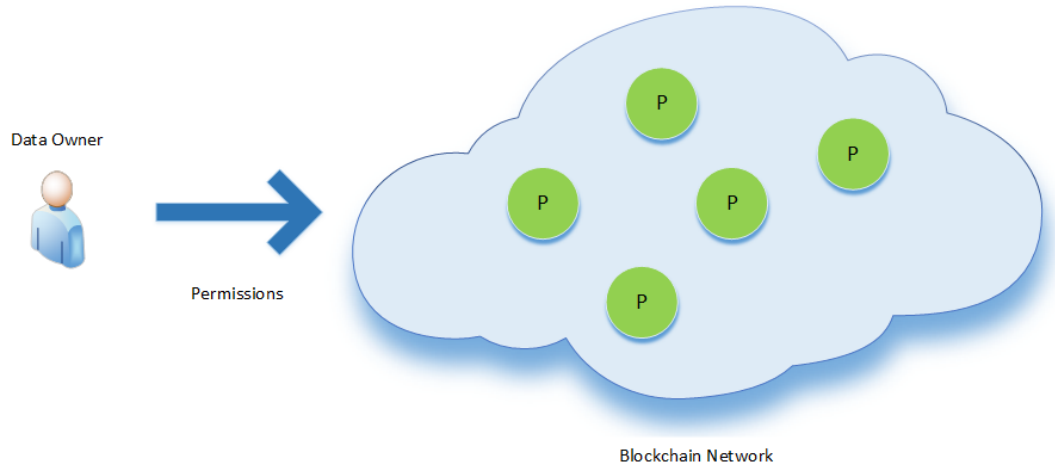

Figure 4.3: Giving Access to Parties

– The owner has the authority to change the set of permissions, to add new services and to revoke the access.

4.1 Computation

The most important part of this thesis is the computation part since the most of the work takes place in this part. Arbitrary functions are evaluated in the network in a secure manner by permitted parties. At the end of the computation, correct output of the result or payment refund is provided to the service and privacy is provided to the data owner.

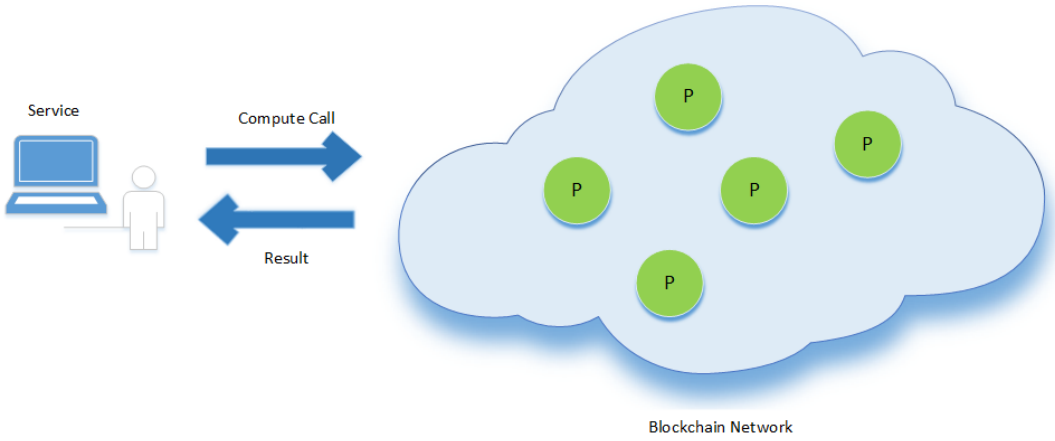

Figure 4.4: Start of Computation

- A compute request including a function, a reference to related data and the

deposit is sent to the blockchain by the service.

- Blockchain executes a contract verifying that this service has been authorized by data owners.
- Upon verification of the service, the contract randomly chooses a one-time computation quorum and queries storage quorums carrying the reference data for this computation to re-share the new computation quorum in parallel.

The computation consists of two parts: quorum selection and multiparty computation.

4.1.1 Quorum Selection

Quorum selection has an important role in the system since the fast quorum selection provides scalability which allows to increase performance by using load-balancing and doing similar works in parallel.

Whenever a secret is shared, a storage quorum is chosen to keep it and in a period of time this quorum is refreshed. In a similar manner, whenever a compute request is sent, a computation quorum is chosen to execute the secret in a secure way.

Instead of Byzantine Agreement protocol, quorums are used to reach consensus on the identities in the network by using blockchain as a source of cryptographic beacons.

When all the inputs are distributed, computation quorum involves the function in a secure multiparty computation. If the result is computed correctly which is checked by the service, then the payment is divided among the honest parties. If not, the payment is paid back to the service. In all cases, rewarded or penalized parties are identified by blockchain. Honest ones are paid by the service or the deposits of the malicious ones.

4.1.2 Multiparty Computation(MPC)

Generally, multiparty computation described in this thesis allows computing parties to access blockchain and be encouraged to behave honestly. MPC protocol concentrates on doing computations by using addition and multiplication gates.

Together with others, parties build a transcript to supply sufficient data to blockchain so that the blockchain recognizes cheaters even if the computation corrupts. There is an observer in the system to reveal cheaters without considering the number of collapses to guarantee that following the protocol is incentive-compatible where s/he can reward or penalize parties, in the game-theoretic sense.

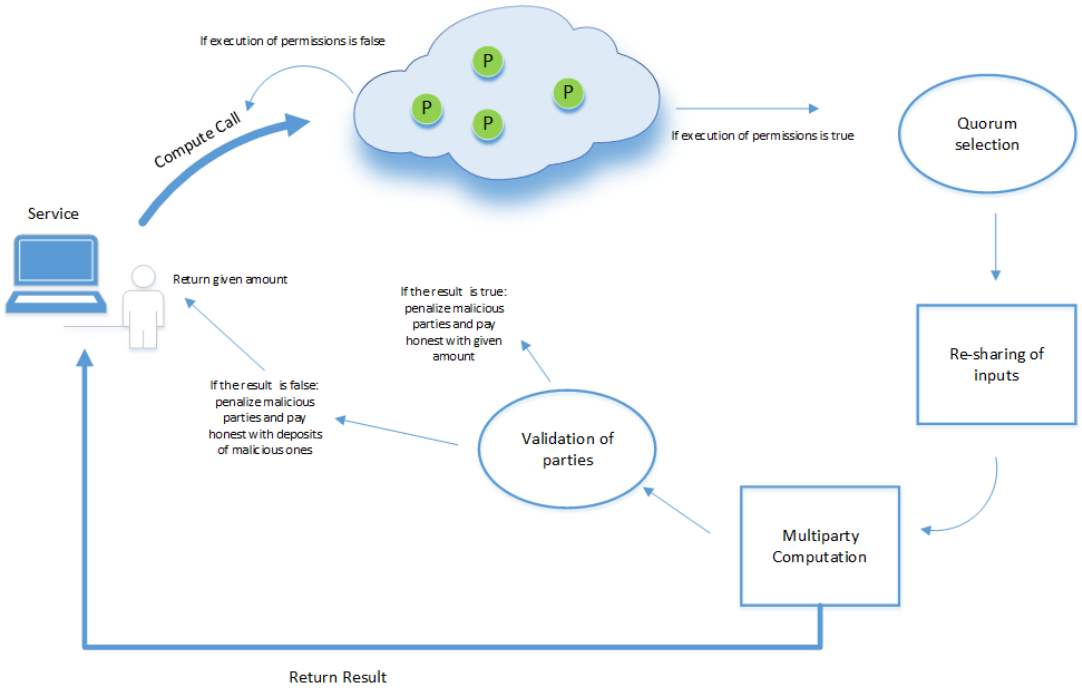

Figure 4.5: Computation Protocol

- Service:
  - Compute request including computation function, amount of deposit and references related to data is sent to blockchain.
  - Service can query all references authorized by blockchain.
- Blockchain:
  - A random computation quorum consisting of computing parties is chosen by using cryptographic beacons.
  - For each reference of data, re-share protocol is called.
    - A new random quorum is chosen.
    - References of data are renewed at the new quorum and old ones are deleted.

  - Then the computation is started.

- Computing parties:
  - In each round, the computation function is locally performed reach up to a multiplication gate.
  - After achieving a multiplication gate, they randomize their shares and transmit them to the blockchain.
  - For each such a gate, the shares are masked with a random mask.
  - This operation continues until reaching an output gate.
  - After achieving an output gate, each party encrypts the share of random masking output by using public key of the service and broadcast it.
  - Round information, masked shares and encrypted shares are sent to the contract executed by blockchain.
  - This completes one round and the number of rounds are increased by one.
  - All these computation steps up to here work asynchronously from the blockchain.

- Service:
  - For each party in for each round, encrypted share and masked share are queried from the blockchain.
  - Encrypted share is decrypted by using private key of the service to reach random mask.
  - Then, by using random mask and masked share, output is reconstructed.
  - If the reconstruction is successful, the state is designated as dispute.
  - The party is added to the list of parties who have failed to encrypt right share in each round.
  - This list is sent to the blockchain.
  - If the reconstruction fails, computation ends early.

- Blockchain:
  - The result is taken from the service.

  - If the result is true:
    * A party is randomly selected from the parties out of corrupted list.
    * To verify this chosen party, a simulator is executed by using computation function and related party's reference data.
    * If the result of simulator execution is true, then state is set as dispute.
    * If not, it seems that all honest parties are cheated.
      · All current parties excluding aborters are deleted from the corrupt list.
      · The parties seems as honest are added to list.
  - If the result is false, i.e. computation fails:
    * The simulator is executed again for the parties who are not in the corrupt list.
  - The payment is made by the blockchain in two ways:
    * If computation runs correctly, deposits are collected from corrupted parties and a certain amount of money paid for honest ones by the service.
    * If computation fails in spite of honest parties, deposits are collected from corrupted parties by the service and a certain amount of money paid for honest ones by the corrupted ones. In other words, the service is not expected to pay and the penalties are transferred to honest parties.

- All parties terminate their activation and delete their shares and any other related data at the end.

To summarize, the whole computation and storage process takes place in three phase:

Pre-processing Phase: In this step, input sharing and randomization occur. Data owner participates in this step only to distribute data and to set permissions.

Online Phase: Multiparty computation is executed in this phase. Although input sharing is pushed to the offline phase because of its being synchronously, the online phase can occur asynchronously.

Post-processing Phase: Identifying, rewarding and penalizing of parties is done in this phase. Also, payments are distributed to parties. This phase occurs synchronously by the blockchain decoupling from the online phase [26].

4.2 Enigma

Enigma is designed to handle privacy and scalability issues holding on some of the blockchain platforms by using secret contracts which are used to execute computations. In other words, Enigma, as a decentralized application, provides end-to-end data privacy and strong correctness for blockchain applications [25].
Privacy is provided by using secure multiparty computation techniques where private data is divided among nodes and, without leaking data to other nodes, they compute functions together as mentioned earlier.
On the account the fact that the data is not stored on each node and computations are made on specific nodes, this scheme is scalable.

4.2.1 Design

In the design, Enigma and blockchain collaborates in separate parts as it is shown in Figure 4.6. Enigma works as a computation and storage layer while blockchain works as a consensus layer.

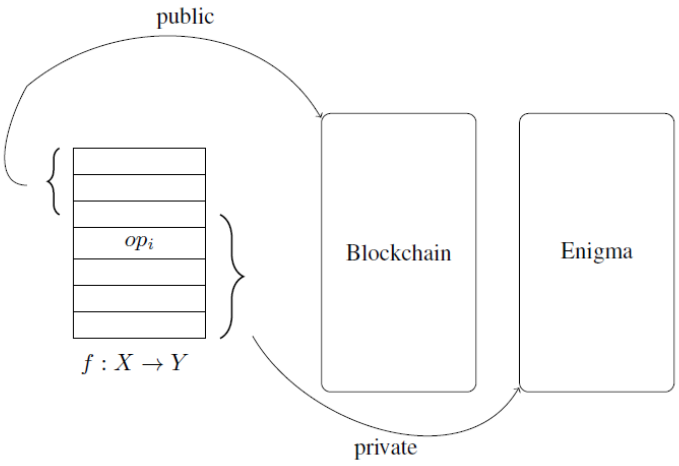

Figure 4.6: Design of Enigma [25]

The blockchain enables all transactions, providing access control based on digital signatures and programmable permissions. Besides that, Enigma guarantees privacy and correctness by executing intensive computations. The difference between blockchain and Enigma is that in every node in blockchain executes the same code and gets the same result in a decentralized way while in Enigma the work of computation is distributed to whole network. Ethereum [24] is used as the blockchain platform since Enigma uses smart contracts to set permissions related with public and private data.

- Users generate private computation tasks via Enigma Network.
- To verify the selected quorum, a cryptographic proof is used before generating the task.
- Secret contract address, signature of computation function, input parameters and computation fee is encrypted by the data owner while generating the task.
- An Ethereum transaction which includes a record of the task, containing hash of all inputs and some metadata, is created.
- This record is used for verification of inputs and the order of tasks.
- Computation:
  - The generated task data is broadcast to the network by the data owner when the record transaction is presented.
  - Soon after receiving the data, the inputs are verified.
  - Selected quorum executes the computation task.
  - Parties in the quorum stores a current copy of the state.
  - Updates of state are spread through the parties in the form of encrypted state changes.
  - The state changes stored in each party in an ordered list to reconstruct the complete state.
  - The computation results are encrypted and broadcast to the network.
  - The verification of computation is made by blokchain and the computation task is approved.

  - Quorum commits a receipt of task related with the record.

- Only hash values are stored on the Ethereum. Task results, encrypted state changes are not stored.

- Secret Contract:

  - The contract is stored on Enigma with a hash on Ethereum.
  - It is used to execute general conditions of private data related with the computation.

- Key Management:

  - All types of data, at rest or in motion, should be enrcrypted in every time apart from computation process realised on quorums.
  - Parties work on encrypted data using the related keys. Various types of keys are used in terms of their functions:
    * User keys:
      · Signing key: It is an Ethereum key used for identification of the user.
      · Encryption key: It is used to obtain input and output keys.
    * Quorum keys:
      · Signing key: It is an Ethereum key used for identification of the owner.
      · Encryption key: It is used to obtain input and output keys.
      · Master key: It is used to obtain state keys in the form of hash values.
      · State key: It is a shared key for each contract(AES256).if it is derived from the master key, it does not require to store, it can be used for data in transit.
    * Shared keys:
      · Input key: This key, derived from the public key of the computing party and private key of the user, is used to generate a unique key to encrypt the argument by the user.

· Output key: This key, derived from the private key of the computing party and public key of the user, is used to generate a new symmetric key by the related party.

- Enigma provides different independent runtimes communicating via a main controller.
- Private data safely encrypted.
- For each task, a record is generated on the Ethereum ledger.
- An authoritative cryptographic proof is obtained that the target worker is securely running trusted hardware prior to sending data to it and paying the corresponding fees.

4.3 Security Analysis

Secret sharing and MPC algorithms guarantee the privacy of data of the owners. Usage of secret contracts and random quorums allow to correct executions on data. Although output of the execution seems like accessible by everyone, the owners approve and limit services to query and access the output. It is assumed that owners or services do not serve malicious inputs. Also, it is ensured that services achieve correct outputs.

By using commitments for inputs and transcripts of executions, verification of computations and identification of malicious parties can be done without leakage of any information.

By using blockchain, it is provided to parties acting on the current state of the network, consensus and broadcast. PKI is used on private channels and public keys of parties stored on-chain. To allow fast executions, parties communicate asynchronously, but the time of blocks define the synchronisation time with an upper bound(round times). Since the computations are done asynchronously, the parties could not delay the protocol.

CHAPTER 5

DIGITAL IDENTITY

People share their personal details, credentials with other parties for authentication or admission purposes. In order to prevent security weaknesses that may arise in this context, it is important that authentication and identity sharing processes is to be managed by the person oneself which is called self-sovereign identity. In fact, this kind of self-sovereign identity management systems has a strong visionary alignment with the general data protection regulations that is common privacy protection nowadays [8].

Traditionally, centralized identity systems have some security risks such as single point of failure, unauthorized access of data and hijacking. In general, an identity system providing data ownership and full security does not exist. In this context, it is aimed to develop a blockchain-based architecture instead of traditional central structure in order to meet the requirements of security and protection of personal data regulations. In addition, since identity credentials contains the most private data about a person, privacy is the most important issue.

5.1 History of Digital Identity

People and companies continue to look for secure ways of identification, privacy protection, access control mechanisms since their existence. The improvements of identity systems have started in 1960s with usage of databases [1].

Institutional Databases (mid-1960s): Institutions and organizations enhance and control digital identity databases to access and handle stored data on people. Digital identity systems were first used in the world in such a way that.

Access Control Lists (early-1970s): The concepts of identity and access have been incorporated into computer systems managing databases as technology developed. Many of the techniques for access control lists (ACLs) were developed in the 60s and 70s and they are still used today. The thinking behind current username and password schemes is ACLs.

Public Key Cryptography-PKI (late-1970s): One of the most important cryptographic discoveries from 1977 to today is public key cryptography. In PKI, it is important to know who owns the key pairs. By this means, the key pairs must be connected to the identities. PKI systems enable publication and storage of digital certificates to identify that a public key belongs to a specific identity.
From the long-standing practice of PKI and CAs, to the trust experiment internet of PGP, to the blockchain technology, public key cryptography has been the foundation of identity on the Internet.

World Wide Web: In order to generate HTTPS, PKI and CA techniques have been introduced to the internet to allow websites that users can trust their identities and they can share data over secure channels with these websites.

Shared identities did not exist on websites in the early days of world wide web. Microsoft took the first step to present an easier way of authentication to websites, Passport, in 1999. Its aim is to provide users a trusted online identity and to provide organizations more registered users by simplifying the login processes.

Passport was the first to be dangerous about the privacy of users as in the first attempt to make shared identities. With Passport, Microsoft also owns and controls the user's identity and each transaction of user performed through the identity management system of Microsoft.

Unlike the centralized scheme of Microsoft, the federated systems has been developed. In federated identity systems, mechanisms send only bits and some pieces of the stored profile of the user, so authentication processes does not trust central author-

ity, but a group of trusted ones.

The Social Networks: Even CA model dominates many of the identity schemes after 1980s, social networking sites precipitated the next significant change in digital identity. The most populated one of this sites is Facebook. It works by enabling users to put online their social graph, matching their relationships with others, organizations, ideologies, interests, etc. While Facebook describes itself as a network, it is actually a huge database centrally managed that stores and executes algorithms about information gathered by a third of the world's population.

Social network identification has become highly common with billions of identities of users available, while users also enjoy their usability for single-sign-on and reusage of identity.

Biometric Identity: Back in the nineteenth century, the technique of identifying a person depends on digital storage of physical characteristics, such as fingerprints. On the other hand, modern biometrics utilize digital abstractions to define people with their physiological and even behavioral characteristics.

The most known biometric identification scheme ever introduced in India holds the personally identifiable information of more than 1.19 billion people in a centralized database. For each Indian resident, a 12-digit unique ID number corresponds to a database record that contains biometric data including a photograph, iris-scans, fingerprints etc.

Blockchain and a New Federated Model for Identity: Destroying the existing identity model, combined with traditional username and password, single-sign-on, biometrics, new identity schemes are arising, based on a new approach to create federated identities: using blockchain. They attempt to restore transparency, privacy and self-control of users.

5.2 Blockchain Based Digital Identity System

Especially on the internet, when we consider social media sites, there are lots of identities for only one person . Also, in real world, people have to prove their identity in

many times and in many places. However, every person has only one identity together with different credentials. On the other hand, while the people manage these personal credentials, every related information must be kept and transferred in a secret manner. Taking into account all of above technologies covered in the previous chapters, it is possible to develop an identity management system by using a secure multiparty computation.

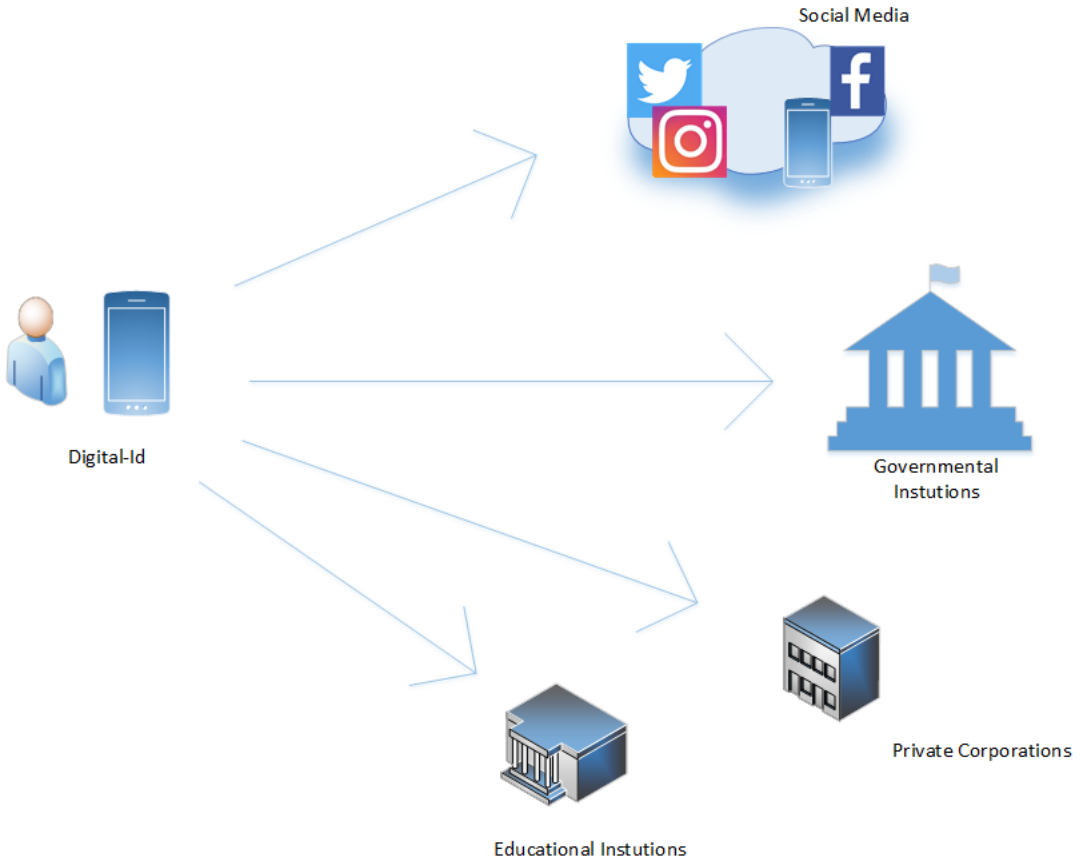

Figure 5.1: Self Sovereign Identity

To do this, as shown in the figure, first we want to develop a mobile application in which credentials are managed and their references are stored as sub-identites. These sub-identities arranged in terms of their contents and intended purposes. Then these credentials are distributed via secret sharing methods to the parties who need this related identity information.

This system can work as in the following:

- Credentials are created by the related institutions. For example, a graduation

certificate is created by the university.

- These credentials are divided into some meaningless pieces by using Shamir's scheme and these pieces are sent to computing parties.
- A reference data about these credentials can be stored on the mobile phone of the owner and permissions are controlled via this reference data.
- When the compute permission is sent to creator of the credential, the computing parties come together, compute this credential by using secure multiparty computation techniques on the blockchain and send this to the party who needs this credential.

Such an identity system will be a solution for emerging and increasing sensitivity to the protection of personal data. Execution of operations through the blockchain system eliminating the need for paperwork and other conventional registration procedures will decrease waste of time and will be cost-effective.

CHAPTER 6

CONCLUSION

As in previous systems, central authorities ensure privacy and data control; with the spread of blockchain technology, trust of centralized systems has decreased and this accompanies privacy concerns. Nowadays, share of data is an irrecoverable; when it is sent, there is not any way to return it and usage of this data is limitless. Usage of secure computation protocols during sharing data provides control of data and makes data reversible.

In the scope of this thesis, firstly, most known cryptographic algorithms, functions and some privacy protocols like secure multiparty computation, secret sharing were examined. Secondly, details and building blocks of blockchain technology were given. Thirdly, we investigated a privacy enhancement protocol on blockchain technology by using secure multiparty computation protocols. And then we give one application of this protocol, Enigma which is designed to enable control of personal data without leaking the original data to other parties. In the next chapter, we focus on a daily-life applications of this protocol, a blockchain based identity management system enabling people to self-control of their credentials.

6.1 Future Work

Secure multiparty computations has become a current issue due to the privacy protocols. However, there are many sub-algorithms to be examined behind this protocols such as secure comparison protocols. Thus, secure comparison protocols on secure multiparty computation can be examined in detail and their complexity can be de-

creased. Also, comparison of algorithms using secure multiparty computations and their effectiveness can be done in the near future together with the other privacy enhancement protocols using zero knowledge proofs, fully homomorphic encryption schemes. Lastly, new use cases can be implemented and applied for user-friendly applications.

REFERENCES

[1] Apocrita, A brief history of identity management, https://apocritaaccess.com/a-brief-history-of-digital-identity/.

[2] A. Chandrayan, Secret sharing and its application to electronic-voting.

[3] D. Chaum, Blind signatures for untraceable payments (1983), in *Advances in Cryptology*, 1982.

[4] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, in *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pp. 383–395, IEEE, 1985.

[5] S. Dziembowski, Multiparty computation protocols, 2009, https://www.slideshare.net/sdziembowski/lecture-10-multiparty-computation-protocols-1988894.

[6] J. Frankenfield, Consensus mechanism, 2019, https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp.

[7] A. G. Giannopoulos, D. I. Mouris, and M. N. Garofalakis, Privacy preserving medical data analytics using secure multi party computation. an end-to-end use case., 2018.

[8] G. B. A. D. I. W. Group, Self-sovereign identity, 2018, https://jolocom.io/wp-content/uploads/2018/10/Self-sovereign-Identity.

[9] C. Interpretation, Cryptographic beacons (a.k.a. randomness beacons), http://www.copenhagen-interpretation.com/home/cryptography/cryptographic-beacons.

[10] J. Katz and Y. Lindell, *Introduction to modern cryptography*, Chapman and Hall/CRC, 2014.

[11] A. D. G. Market, Block chain 101, 2017, https://cfaemirates.com/wp-content/uploads/2017/11/blockchain-technology-101.pdf.

[12] S. Nakamoto et al., Bitcoin: A peer-to-peer electronic cash system, 2008.

[13] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*, Springer Science & Business Media, 2009.

[14] M. Pratap, Everything you need to know about smart contracts: A beginners guide, 2019, https://medium.com/hackernoon/everything-you-need-to-know-about-smart-contracts-a-beginners-guide-c13cc138378a.

[15] T. Rabin and M. Ben-Or, Verifiable secret sharing and multiparty protocols with honest majority, in *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pp. 73–85, ACM, 1989.

[16] N. v. Saberhagen, Cryptonote v 2.0, CryptoNote. org.[Online], 17(10), 2013.

[17] B. Schoenmakers, A simple publicly verifiable secret sharing scheme and its application to electronic voting, in *Annual International Cryptology Conference*, pp. 148–164, Springer, 1999.

[18] A. Shamir, How to share a secret, Communications of the ACM, 22(11), pp. 612–613, 1979.

[19] N. P. Smart, *Cryptography made simple*, volume 481, Springer, 2016.

[20] N. P. Smart et al., *Cryptography: an introduction*, volume 3, McGraw-Hill New York, 2003.

[21] W. Stallings, *Cryptography and network security: principles and practice*, Pearson Upper Saddle River, 2017.

[22] N. Sullivan, Do the chacha: better mobile performance with cryptography, 2015, https://blog.cloudflare.com/do-the-chacha-better-mobile-performance-with-cryptography/.

[23] D. Voell, F. L.-N. Gaski, R. Jagadeesan, R. Khasanshyn, H. Montgomery, S. Teis, T. Blummer, M. K. Katipalli, and M. Bowman, Hyperledger whitepaper, Published: https://wiki. hyperledger. org/groups/whitepaper/whitepaper-wg, 2016.

[24] F. Vogelsteller, V. Buterin, et al., Ethereum whitepaper, 2014.

[25] G. Zyskind, O. Nathan, and A. Pentland, Enigma: Decentralized computation platform with guaranteed privacy, arXiv preprint arXiv:1506.03471, 2015.

[26] G. Zyskind et al., *Efficient secure computation enabled by blockchain technology*, Ph.D. thesis, Massachusetts Institute of Technology, 2016.