Shorter, Privacy-Preserving Proof of Reserves Protocols for Cryptocurrency Exchanges

A Seminar Report
Submitted in partial fulfillment of
the requirements for the degree of
Dual Degree (B.Tech & M.Tech) in Electrical Engineering with Specialization in Communication & Signal Processing
by

Suyash Bagad
(Roll No. 15D070007)

Supervisor:
Prof. Saravanan Vijayakumaran

Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai 400076 (India)

20 June 2020

Dedicated to ...

Acceptance Certificate

Department of Electrical Engineering
Indian Institute of Technology, Bombay

The dissertation entitled "Shorter, Privacy-Preserving Proof of Reserves Protocols for Cryptocurrency Exchanges" submitted by Suyash Bagad (Roll No. 15D070007) may be accepted for being evaluated.

Date: 20 June 2020

Prof. Saravanan Vijayakumaran

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this report. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Suyash Bagad

Date: 20 June 2020 (Roll No. 15D070007)

Abstract

The rise of cryptocurrencies began with the inception of Bitcoin in 2009. Since then, several cryptocurrencies with better privacy and security guarantees are being developed. Cryptocurrencies gained popularity among general masses with the establishment of cryptocurrency exchanges. Also known as digital currency exchanges or crypto exchanges, they are essentially businesses that allow customers to trade cryptocurrencies or digital currencies for other assets including conventional fiat money or different digital currencies. From a customer point of view, exchanges not only made owning cryptocurrencies possible to non-miners but also provided them with fast trading platforms for transactions within cryptocurrencies and fiat. Customers were also provided with custodial wallets freeing them from the hassle of storing and remembering private keys. In the early days of cryptocurrency, crypto exchanges were very few and less-known, but not too long ago their number increased dramatically and they became an integral part of the cryptoeconomic ecosystem. They were responsible for the boost in the transaction volumes of the vast majority of the cryptocurrency sales and liquidity.

The downside of such cryptocurrency exchanges is that they are required to store sensitive information of customers like the private keys and account balances. If in case an exchange is hacked, it might result in loss of customer-owned cryptocurrency assets. There have been many high-profile hacks over the years, many of which went unnoticed for some time [1]. Although having a fool-proof method to avoid such hacks might be a difficult task, proof of reserves is one way to uphold the trust of customers. A proof of reserves is the guarantee by the exchange that it owns reserves at least as much as its total liabilities towards customers. In this way, even after cases of hacking, the exchange could repay its liabilities to the customers.

The simplest way to publish a proof of reserves for an exchange is to reveal all the addresses or account details it owns so that the customers are convinced about the assets owned by the exchange. Another way could be to send all the reserves it owns from all its addresses to a single addresses it owns. If amounts involved in

a transaction are public as in the case of Bitcoin, such a self-transaction would be a proof of the exchange's reserves. For example, in 2011, Mt. Gox cryptocurrency exchange transferred 424,242 bitcoins from its wallets to a previously revealed Bitcoin address [2]. However, such proofs of reserves do not preserve the privacy of the exchanges. Information of an exchange's addresses or accounts and the total assets it owns are crucial for aspects of its business. Exchanges naturally would not be in a position to compromise such critical information as a part of proofs of reserves. The main challenge in design of proofs of reserves is to preserve privacy and confidentiality of exchanges but at the same time convince customers about an exchange's actual asset ownership. Regaining *trust* of the customers without compromising exchanges' *privacy* is the primary motivation behind the design of better proofs of reserves. Advanced cryptographic techniques make it possible to design proofs of reserves which reveal *nothing* beyond an assertation of the form:

Exchange X owns ? amount of the cryptocurrency Y.

Note that here we do not intend to reveal even the total amount. A publicly verifiable proof backing up such a claim is a cryptographic tool known as a *Non-Interactive Zero-Knowledge* proof.

In this work, we study the existing proof of reserves protocols for privacy-centric cryptocurrencies Grin, Beam and Monero. The existing proof of reserves protocols possess some shortcomings which becomes a hurdle in their practical deployment. With an aim to alleviate limitations of previously designed proofs of reserves, we design novel proofs of reserves for crypto exchanges supporting the above cryptocurrencies. Our protocols are shorter and privacy enhancing in comparison to the existing state-of-the-art proofs of reserves. Previous state-of-the-art proofs of reserves protocols provided some privacy to exchanges by hiding the exchange-owned addresses (or outputs) in a larger anonymity set. The proof sizes for these protocols scaled linearly with the anonymity set size. Since the level of privacy in a proof of reserves directly depends on how large the size of the anonymity set size is as compared to the number of exchange-owned addresses, larger anonymity sets imply stronger privacy. However, as the previous protocols proof sizes grew linearly as the anonymity set grows, it bought practical limitations (with regards to that of proof storage and broadcast) on what level of privacy could be attained. With an aim to improve scalability of the previously design proof of reserves protocols, we design a strategy based on Bulletproofs technique [3] to design proofs of reserves scaling logarithmically in the anonymity set. This brings flexibility in the choice of the

size of anonymity set and therefore enhances the attainable level of privacy. Along with this, we also use the concept of *key images* to ensure that different exchanges cannot share their addresses. The key images subsequently also helps in detecting double-spending of addresses by an exchange. Going a step further, we also devise a cryptographic technique to enforce different exchanges to publish proofs of reserves corresponding to a same blockchain state, which is absent in previous work. This ensures that exchanges can publish proofs of reserves only at particular time instances (for example, after each block is mined), preventing them from cheating customers with ambiguous choices of anonymity sets. We also implement the protocols we design as well as the previous state-of-the-art protocols for comparison and show feasibility in practical deployment of our protocols. We believe that our work on proof of reserves could be well adopted by crypto exchanges and benefit several of their customers. Furthermore, our work also opens up avenues of exploration of establishing trust without compromising privacy or anonymity in a more general setting. We are hopeful that this work acts as a small but crucial contribution towards the goal of establishing a *trustless, decentralized economy.*

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rise of cryptocurrencies have openend up unending possibilities of how minimal or no trust based systems could be established owing to decentralization. The concept of blockchain was introduced with the founding of Bitcoin by Satoshi Nakamoto [4]. Satoshi Nakamoto proposed and implemented idea of a consensus based, trustless, truly peer-to-peer system for financial transactions. Notwithstanding an intrumental step towards establishing a decentralized and a trustless system, Bitcoin has several practical limitations with regards to privacy, security and scalability [5]. This led to the development of more privacy and anonymity focussed cryptocurrencies like Monero [6] and Zcash [7]. Grin [8] and Beam [9] are two relatively new projects which are backed by the MimbleWimble protocol [10] and claim to promise scalability, anonymity and fungibility all at once. The rise in privacy-centric cryptocurrencies further led to growth in popularity of cryptocurrencies not only among investors but also common people.

1.1 What is a Blockchain?

1.1.1 Decentralized Ledger

1.2 Notion of Privacy on a Blockchain

1.3 Cryptocurrency Exchanges & Security

1.4 Proof of Solvency

1.4.1 Proof of Reserves

1.4.2 Proof of Liabilities

Chapter 2

Cryptographic Preliminaries

Elliptic-curve cryptography (ECC) is essentially a public-key cryptography system, design of which is based on the algebraic structure of elliptic curves over finite fields [11]. ECC enables significant reduction in memory usage as the public-key length in ECC framework is much smaller than other public-key cryptography schemes like RSA for the same security level. It is widely used for many applications like encryption, digital signatures, pseudo-random generators and other tasks.

All cryptocurrencies are built on the Elliptic-curve cryptography framework. The security of such systems depend on the security guarantees provided by the ECC framework. We will see a couple of cryptographic assumptions which promise us the security guarantees for cryptocurrency systems. Thus, before delving into the details of proof of reserves protocols, it is worthwhile spending some time learning about the background math of cryptocurrencies. Note that we assume familiarity of the reader with basic concepts of group theory and modular arithmetic. A short but sufficient primer on both of these topics in present in [12].

2.1 Notation

Let $\mathcal{G} = \{\mathbb{G}, q, g\}$ be the description of a cyclic group $\mathbb{G}$ of prime order q with generator g of $\mathbb{G}$. Let $h \in \mathbb{G}$ be another random generator of $\mathbb{G}$ such that the discrete logarithm relation between g and h is not known. Let $\mathbb{G}^n$ and $\mathbb{Z}_q^n$ be the n-ary Cartesian powers of sets $\mathbb{G}$ and $\mathbb{Z}_q$ respectively. Group elements which are Pedersen commitments are denoted by uppercase letters and randomly chosen group elements are denoted by lowercase letters. Bold font denotes vectors. Inner product of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^n$ is defined as $\langle \mathbf{a}, \mathbf{b} \rangle \coloneqq \sum_{i=1}^{n} a_i \cdot b_i$ where $\mathbf{a} = (a_1, \ldots, a_n), \mathbf{b} = (b_1, \ldots, b_n)$. Further, Hadamard and Kronecker products are defined respectively

as, $\mathbf{a} \circ \mathbf{b} \coloneqq (a_1 \cdot b_1, \ldots, a_n \cdot b_n) \in \mathbb{Z}_q^n$, $\mathbf{a} \otimes \mathbf{c} \coloneqq (a_1\mathbf{c}, \ldots, a_n\mathbf{c}) \in \mathbb{Z}_q^{nm}$ where $\mathbf{c} \in \mathbb{Z}_q^m$. For a base vector $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{G}^n$, vector exponentiation is defined as $\mathbf{g}^{\mathbf{a}} = \prod_{i=1}^{n} g_i^{a_i} \in \mathbb{G}$. For a scalar $u \in \mathbb{Z}_q^*$, we denote its consecutive powers in the form of a vector $\boldsymbol{u}^n \coloneqq (1, u, u^2, \ldots, u^{n-1})$. To represent the exponentiation of all components of a vector $\mathbf{a}$ by the same scalar $k \in \mathbb{Z}_q$, we use $\mathbf{a}^{\circ k}$ to mean $(a_1^k, a_2^k, \ldots, a_n^k)$. If an element a is chosen uniformly from a set A, such a choice is denoted by $a \xleftarrow{\$} A$. We denote the relation *Relation* using the specified input and witness as $\{(Public\ Input;\ Witness) : Relation\}$. We refer to $\mathcal{A}$ as a PPT adversary which is a probabilistic Turing Machine that runs in polynomial time in the security parameter λ. An *interactive proof* for the decision problem π is described as follows:

1. There are two participants, a prover $\mathcal{P}$ and a verifier $\mathcal{V}$.
2. The proof consists of a specified number of rounds.
3. In the beginning, both participants get the same input.
4. In each round, the verifier challenges the prover, and the prover responds to the challenge.
5. Both the verifier and the prover can perform some private computation.
6. At the end, the verifier states whether he was convinced or not.

2.2 Basics of Elliptic Curves

Let $a, b \in \mathbb{R}$ such that $4a^3 + 27b^2 \neq 0$. Let E be the set of solutions $(x, y) \in \mathbb{R}$ to the equation

$$y^2 = x^3 + ax + b. \tag{2.1}$$

An elliptic curve over $\mathbb{R}$ is given by the set $E \cup \{\mathcal{O}\}$ where $\mathcal{O}$ is known as the *point at infinity*. An example of an elliptic curve over $\mathbb{R}$ is given in Figure 2.1. Note that the condition $4a^3 + 27b^2 \neq 0$ ensures that the curve does not have repeating roots. This condition is necessary in the discussion of elliptic curve groups.

2.2.1 Point Addition in Elliptic Curves

The set $E \cup \{\mathcal{O}\}$ needs to be an algebraic group for us to be able to define operations on it. Thus, we define a group operation over $E \cup \{\mathcal{O}\}$ known as *point addition*. Suppose we have two points $P = (x_1, y_1), Q = (x_2, y_2) \in \mathbb{R} \times \mathbb{R}$ such that $x_1 \neq x_2$. We show them by blue and yellow coloured points in Figure 2.2(a). We

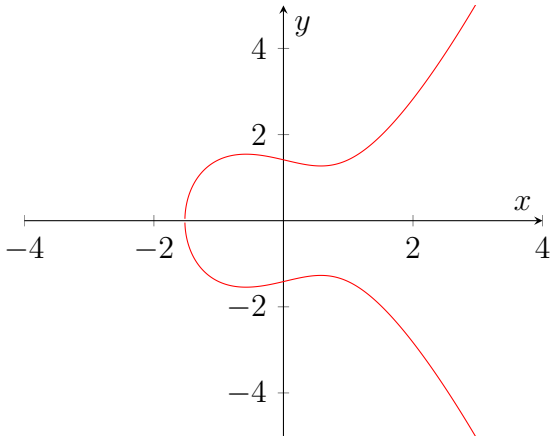


Figure 2.1: An elliptic curve given by the equation $y^2 = x^3 - x + 2$ over $\mathbb{R}$

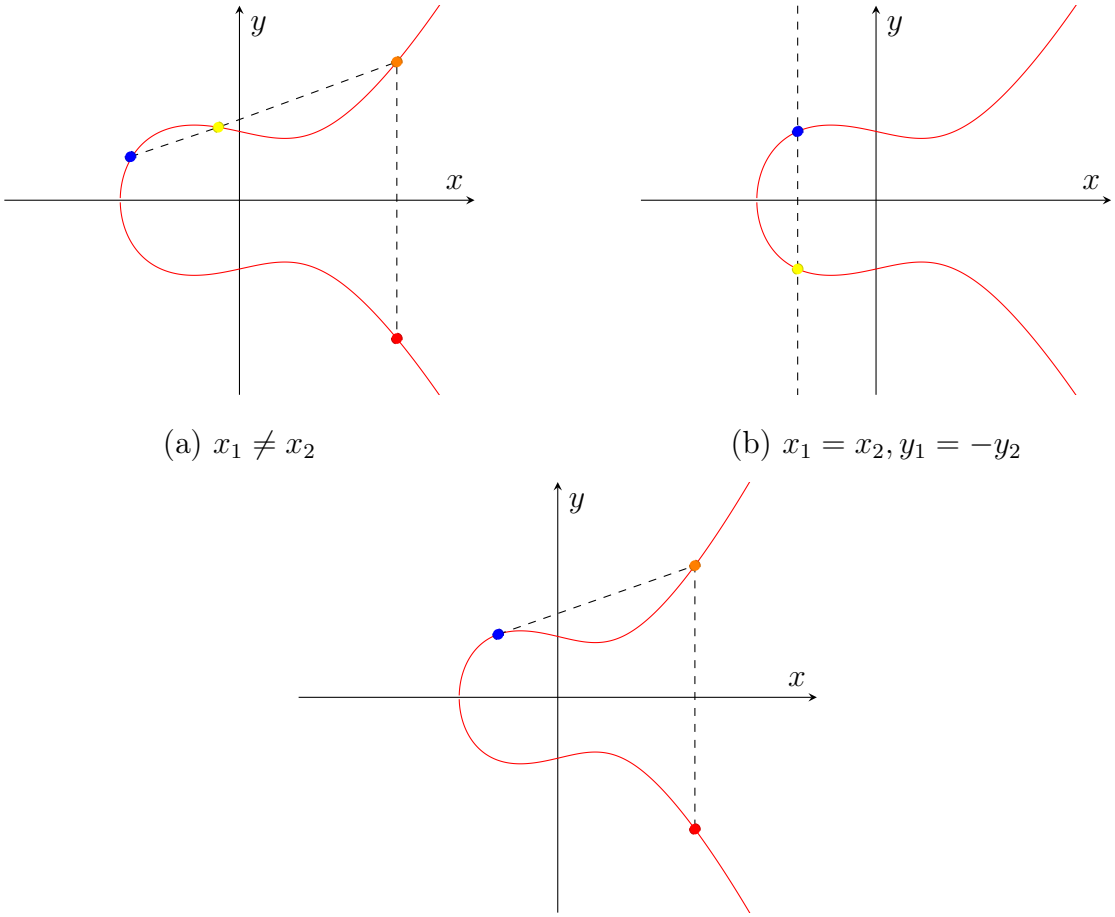
draw a line passing through P and Q. As the degree of an elliptic curve equation is 3, any non-tangent line must intersect the curve in 3 distinct points. Suppose the line passing through P and Q intersects the curve at point $R' = (x_3, -y_3) \in \mathbb{R} \times \mathbb{R}$, shown in orange colour. We define the result of point addition of points P and Q to be the point R which is the mirror reflection of R'. Therefore, we have $R = (x_3, y_3)$ shown in red colour. Simple calculation shows that given $P = (x_1, y_1), Q = (x_2, y_2)$ and $x_1 \neq x_2$, point addition gives us $P + Q = (x_3, y_3)$ such that

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2, \;\; y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \tag{2.2}$$

Now if P and Q are such that $x_1 = x_2$ but $y_1 = -y_2$, the line passing through P and Q is vertical and possibly intersects the curve at infinity. In this case, we define the point addition operation as $P + Q = \mathcal{O}$. Therefore, the special point $\mathcal{O}$ acts as an identity point in the group $E \cup \{\mathcal{O}\}$. Further, if we have $P = Q$, i.e. $x_1 = x_2, y_1 = y_2 \neq 0$, we draw tangent to the curve at point P. As the degree of the curve is 3, any tangent will intersect the curve at only and only one point, say point R'. The result R in this case is again the mirror reflection of point R' about the x-axis. The addition of a point to itself is known as *point doubling*. The explicit formula for point doubling of point $P = (x_1, y_1)$ can be written as $P + P = 2P = (x_2, y_2)$ such that

$$x_2 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1, \;\; y_2 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_2) - y_1. \tag{2.3}$$

The point addition operation is closed in the set $E \cup \{\mathcal{O}\}$ by construction. Further, $\mathcal{O}$ acts as an identity element in the set $E \cup \{\mathcal{O}\}$. For each point $P \in E$, we can find its *inverse* $Q \in E$ as its reflection about the x-axis as for such cases, we have $P + Q = \mathcal{O}$. Therefore, the set $E \cup \{\mathcal{O}\}$ is a group under point addition.



(a) $x_1 \neq x_2$

(b) $x_1 = x_2, y_1 = -y_2$

(c) $x_1 = x_2, y_1 = y_2 \neq 0$

Figure 2.2: Point addition in elliptic curves over $\mathbb{R}$

In practice, we use elliptic curves over finite fields instead of real numbers. The group operations are now defined over an underlying finite field F. The division by a non-zero field element $x \in F$ is interpreted as multiplication by its multiplicative inverse x^{-1}. Similarly, subtraction of a field element $x \in F$ is interpreted as addition by its additive inverse $-x$. This is the basis of the elliptic curve cryptography used in modern day systems except that of For example, Bitcoin and Grin cryptocurrency systems use the prime-ordered elliptic curve `secp256k1` [13]. Note that from hereon we use multiplicative notation to denote group operation on elliptic curves $\mathbb{G}$ on finite fields $\mathbb{F}_q$ for a large prime q. Lastly, addition of a point $P \in \mathbb{G}$ for k times is shown as scalar multiplication in additive notation and exponentiation in multiplicative respectively.

$$\underbrace{P + P + \cdots + P}_{k \text{ times}} = kP \in \mathbb{G}.$$

Similarly, in multiplicative notation, such an operation is called as exponentiation and is shown below

$$\underbrace{P \cdot P \cdot \ldots \cdot P}_{k \text{ times}} = P^k \in \mathbb{G}.$$

2.3 Cryptographic Assumptions

Each practical cryptographic systems is built on certain hardness assumptions. For example, the widely popular RSA digital signature algorithm was based on the assumption that it computationally hard to factorize big primes [14]. Elliptic curve cryptography is similarly based on the assumption that the *Discrete Log Problem* is difficult to be solved by a computationally bounded adversary.

Definition 2.3.1 (Discrete Log Relation) *For all PPT adversaries $\mathcal{A}$ and for all $n \geq 2$, $\exists$ a negligible function $\mu(\lambda)$ s.t*

$$\Pr\left[\begin{array}{ll} \mathbb{G} = Setup(1^\lambda), g_1, \ldots, g_n \leftarrow \mathbb{G} \ ; & \\ a_1, \ldots, a_n \in \mathbb{Z}_p \leftarrow \mathcal{A}(\mathbb{G}, g_1, \ldots, g_n) & \therefore \exists a_i \neq 0 \wedge \prod_{i=1}^n g_i^{a_i} = 1 \end{array}\right] \leq \mu(\lambda)$$

We say $\prod_{i=1}^n g_i^{a_i} = 1$ is a non trivial discrete log relation between $g_1, \ldots, g_n$. If the Discrete Log Relation assumption stands, it implies that no PPT adversary can find a non-trivial relation between randomly chosen group elements. This is known as the Discrete Log Problem.

We use additional cryptographic assumptions such as Decisional Diffie-Hellman and its variants as described in [15].

2.4 Cryptographic Commitments

Cryptographic commitments are an important preliminary widely used to anonymise data like amounts. We also briefly discuss some key properties of commitments of our interest.

Definition 2.4.1 (Commitments) *A non-interactive commitment consists of two PPT algorithms* (Setup, Com)*. For a message $x \in \mathbf{M}_{pp}$ (message space), the algorithm proceeds as follows:*

1. *public parameters $pp \leftarrow Setup(1^\lambda)$ for security paramter λ*
2. *$\mathrm{Com}_{pp} : \mathbf{M}_{pp} \times \mathbf{R}_{pp} \to \mathbf{C}_{pp}$, where $\mathbf{R}_{pp}$ is randomness space*
3. *$r \leftarrow \mathbf{R}_{pp}$ and compute $\mathbf{com} = \mathrm{Com}_{pp}(x; r)$*

Definition 2.4.2 (Homomorphic Commitments) *A homomorphic commitment is a non-interactive commitment such that $\mathbf{M}_{pp}$, $\mathbf{R}_{pp}$, $\mathbf{C}_{pp}$ are all abelian groups, and $\forall\ x_1, x_2 \in \mathbf{M}_{pp}$, $r_1, r_2 \in \mathbf{R}_{pp}$, we have*

$$\mathrm{Com}(x_1; r_1) + \mathrm{Com}(x_2; r_2) = \mathrm{Com}(x_1 + x_2; r_1 + r_2)$$

Definition 2.4.3 (Hiding Commitment) *A commitment scheme is said to be hiding if for all PPT adversaries $\mathcal{A}$, $\exists \mu(\lambda)$, a negligible function such that,*

$$\left| \Pr\left[b\text{'}{=}b \;\middle|\; \begin{array}{l} pp \leftarrow \mathrm{Setup}(1^\lambda); \\ (x_0, x_1) \in \mathbf{M}_{pp}^2 \leftarrow \mathcal{A}(pp),\ b \leftarrow \{0,1\},\ r \leftarrow \mathbf{R}_{pp}, \\ \mathbf{com} = Com(x_b; r),\ b' \leftarrow \mathcal{A}(pp,\ \mathbf{com}) \end{array} \right] - \frac{1}{2} \right| \leq \mu(\lambda)$$

where the probability is over $b', r, Setup$ and $\mathcal{A}$. For perfectly hiding schemes, $\mu(\lambda) = 0$.

In simple words, a commitment scheme is *hiding* if it is impossible for a computationally bounded adversary to find what the message is hidden in a commitment or what randomness was used in computing the commitment.

Definition 2.4.4 (Binding Commitment) *A commitment scheme is said to be binding if for all PPT adversaries $\mathcal{A}$, $\exists \mu(\lambda)$, a negligible function such that,*

$$\Pr\left[\mathrm{Com}(x_0; r_0) = \mathrm{Com}(x_1; r_1) \wedge x_0 \neq x_1 \;\middle|\; \begin{array}{c} pp \leftarrow \mathrm{Setup}(1^\lambda), \\ x_0, x_1, r_0, r_1 \leftarrow \mathcal{A}(pp) \end{array} \right] \leq \mu(\lambda)$$

where the probability is over Setup and $\mathcal{A}$. Again, if $\mu(\lambda) = 0$ then we say the scheme is perfectly binding.

A commitment scheme is known as *binding* if it is impossible for a computationally bounded adversary to change the message a commitment commits to once it has published the commitment to the original message.

Definition 2.4.5 (Pedersen Commitment) $\mathbf{M}_{pp}, \mathbf{R}_{pp} = \mathbb{Z}_p$, $\mathbf{C}_{pp} = \mathbb{G}$ *of order* p.

1. Setup*:* $g, h \leftarrow \mathbb{G}$
2. $\mathrm{Com}(x; r) = (g^x h^r)$

Definition 2.4.6 (Pedersen Vector Commitment) $\mathbf{M}_{pp} = \mathbb{Z}_p^n$, $\mathbf{R}_{pp} = \mathbb{Z}_p$, $\mathbf{C}_{pp} = \mathbb{G}$ *of order* p.

1. Setup*:* $\boldsymbol{g} = (g_1, \ldots, g_n), h \leftarrow \mathbb{G}$
2. $\mathrm{Com}(\boldsymbol{x} = (x_1, \ldots, x_n); r) = (h^r \boldsymbol{g}^{\boldsymbol{x}})$

The Pedersen vector commitment is *perfectly hiding* and *computationally binding* under the discrete logarithm assumption. This means that no matter how much computational power an adversary possesses, he cannot find the message hidden in a Pedersen commitment. Further, for a computationally bounded adversary, it is infeasible to find another opening to a Pedersen commitment once he has committed it to original message.

2.5 Zero-Knowledge Arguments of Knowledge

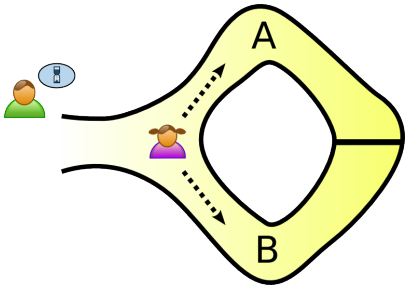
2.5.1 Zero-Knowledge Arguments

A protocol in which a prover convinces a verifier that a statement is true *without* revealing any information about why it holds is known as a Zero-knowledge argument. An argument is a proof only if the prover is computationally bounded and some computational hardness holds. Hereafter, we use the terms *proof* and *argument* interchangeably.

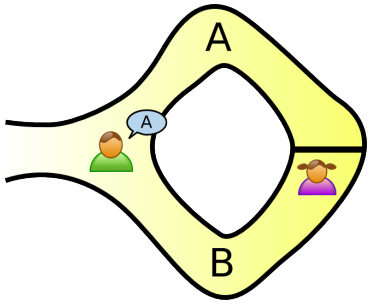
We illustrate the idea of zero-knowledge arguments of proof using the example of *Ali-Baba's secret cave* [16]. *

In the above example, Peggy knows the secret word used to open a mysterious door in a cave. The cave is shaped like a horse-hoe. The entrance is on one side and the magic door blocking the opposite side. Victor wants to know whether Peggy

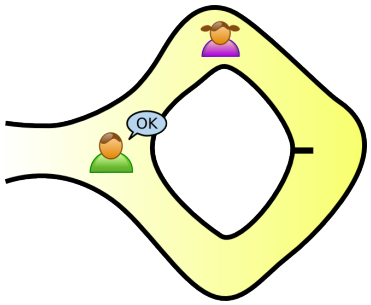
*Figure courtesy: `https://en.wikipedia.org/wiki/Zero-knowledge_proof`.



(a) Peggy chooses a path uniformly from A, B without Victor knowing.



(b) Victor asks her to come out of the cave from path A.



(c) If Peggy had entered from path A, she returns trivially. Otherwise, she could open the door using the secret key and return from path A.

Figure 2.3: Example of a zero knowledge proof

knows the secret word; but Peggy, does not want to reveal her knowledge (the secret word) to Victor or to reveal the fact of her knowledge to anyone in the world.

Peggy and Victor run the protocol described in figure 2.3. Provided she really does know the magic word, and the path she enters and path Victor asks her to come from are same, then it's trivial for Peggy to succeed and Victor to believe that she actually knows the secret key. Further, if the chosen path by Peggy and asked by Victor doesn't match, even then she could open the door and return from a desired path. If they were to repeat this protocol many times, say 15 times in a row, her chance of successfully "guessing" all of Victor's requests would become exponentially small (about three in a lakh).

For zero-knowledge arguments presented in this report, we will consider arguments consisting of three interactive probabilistic polynomial time algorithms $(\text{Setup}, \mathcal{P}, \mathcal{V})$. These algorithms are described by:

1. Setup: $\sigma \leftarrow \text{Setup}(1^\lambda)$, σ is common reference string
2. $\mathcal{P}$: prover, $\mathcal{V}$: verifier
3. Transcript $tr \leftarrow \langle \mathcal{P}, \mathcal{V} \rangle$
4. $\langle \mathcal{P}, \mathcal{V} \rangle = b$, $b = 0$ if the verifier rejects or $b = 1$ accepts

Further, we define the relation $\mathcal{R}$ and the CRS-dependent language as:

$$\mathcal{R} := \{(\sigma, u, w) \in \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* : \ w \text{ is a witness for } u \mid \sigma\}$$
$$\mathcal{L}_\sigma := \{x \mid \exists w \in (\sigma, u, w) \in \mathcal{R}\}$$

So, $\mathcal{L}_\sigma$ is essentially the set of statements x that have a witness w in the relation $\mathcal{R}$.

2.5.2 Defining Zero-Knowledge Arguments of Knowledge

To mathematically define the notion of zero-knowledge and zero-knowledge arguments, we will provide the necessary definitions below.

Definition 2.5.1 (Argument of Knowledge) *The triple $(\mathit{Setup}, \mathcal{P}, \mathcal{V})$ is called an argument of knowledge for relation $\mathcal{R}$ if it is perfectly complete and has computational witness-extended emulation.*

Definition 2.5.2 (Perfect completeness) *$(\mathit{Setup}, \mathcal{P}, \mathcal{V})$ has perfect completeness if for all non-uniform polynomial time adversaries $\mathcal{A}$*

$$\Pr\left[\ (\sigma, u, w) \notin \mathcal{R} \text{ or } \langle \mathcal{P}(\sigma, u, w), \mathcal{V}(\sigma, u) \rangle = 1 \ \middle| \ \begin{array}{l} \sigma \leftarrow \mathit{Setup}(1^\lambda) \\ (u, w) \leftarrow \mathcal{A}(\sigma) \end{array} \right] = 1$$

Perfect completeness implies that if a statement is actually true, then an honest verifier is convinced with probability 1 about the truth of the statement by an honest prover.

Definition 2.5.3 (Computational Witness-Extended Emulation) *$(\mathit{Setup}, \mathcal{P}, \mathcal{V})$ has witness-extended emulation if for all deterministic polynomial time P^* there exists an expected polynomial time emulator $\mathcal{E}$ such that for all*

pairs of interactive adversaries $\mathcal{A}_1, \mathcal{A}_2$ *there exists a negligible function* $\mu(\lambda)$ *such that*

$$\left| \Pr\left[\mathcal{A}_1(tr) = 1 \;\middle|\; \begin{array}{l} \sigma \leftarrow \mathit{Setup}(1^\lambda,) \\ (u, s) \leftarrow \mathcal{A}_2(\sigma), \\ tr \leftarrow \langle (\mathcal{P}^*(\sigma, u, s), V(u, s) \rangle \end{array} \right] - \Pr\left[\begin{array}{l} \mathcal{A}_1(tr) = 1 \wedge \\ \\ (tr \text{ accepted} \implies (\sigma, u, w) \in \mathcal{R}) \end{array} \;\middle|\; \begin{array}{l} \sigma \leftarrow \mathit{Setup}(1^\lambda,) \\ (u, s) \leftarrow \mathcal{A}_2(\sigma), \\ (tr, w) \leftarrow \mathcal{E}^{\mathcal{O}}(\sigma, u) \end{array} \right] \right| \leq \mu(\lambda)$$

where the oracle is given by $\mathcal{O} = \langle (\mathcal{P}^*(\sigma, u, s), V(u, s) \rangle$*, and permits rewinding to a specific point and resuming with fresh randomness for the verifier from this point onwards. We can also define com- putational witness-extended emulation by restricting to non-uniform polynomial time adversaries* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*.*

Computational witness-extended emulation implies that when an adversary produces an argument to convince the verifier with some probability, then we have a corresponding emulator producing identically distributed argument with same probability, but also a witness.

Definition 2.5.4 (Public coin) *An argument of knowledge* $(\mathit{Setup}, \mathcal{P}, \mathcal{V})$ *is called public coin if all messages sent from the verifier to the prover are chosen uniformly at random and independent of the prover's messages, i.e., the challenges correspond to the verifier's randomness* ρ*.*

Definition 2.5.5 (Zero Knowledge Argument of Knowledge) *An argument of knowledge* $(\mathit{Setup}, \mathcal{P}, \mathcal{V})$ *is zero knowledge if it reveals no information about* w *apart from what could be deduced from the fact that* $(\sigma, u, w) \in \mathcal{R}$*.*

An argument of knowledge is zero knowledge if it does not leak information about w apart from what can be deduced from the fact that $(\sigma, u, w) \in \mathcal{R}$. More explicitly, we note that, a zero knowledge argument of knowledge ensures that no PPT adversary (or verifier) can ever recover w given it's relation with σ, u.

Definition 2.5.6 (Perfect Special Honest-Verifier Zero-Knowledge) *A public coin argument of knowledge* $(\mathit{Setup}, \mathcal{P}, \mathcal{V})$ *is a perfect special honest verifier zero knowledge (SHVZK) argument of knowledge for* $\mathcal{R}$ *if there exists a probabilistic polynomial time simulator* $\mathcal{S}$ *such that for all pairs of interactive adversaries* $\mathcal{A}_1, \mathcal{A}_2$

$$\Pr\left[\ (\sigma,u,w)\in\mathcal{R}\wedge\mathcal{A}_1(tr)=1 \;\middle|\; \begin{array}{l}\sigma\leftarrow \mathit{Setup}(1^\lambda,)\\ (u,w,\rho)\leftarrow\mathcal{A}_2(\sigma),\\ tr\leftarrow\langle(\mathcal{P}(\sigma,u,s),V(\sigma,u;\rho)\rangle\end{array}\right]$$

$$=\Pr\left[\ (\sigma,u,w)\in\mathcal{R}\wedge\mathcal{A}_1(tr)=1 \;\middle|\; \begin{array}{l}\sigma\leftarrow \mathit{Setup}(1^\lambda,)\\ (u,w,\rho)\leftarrow\mathcal{A}_2(\sigma),\\ tr\leftarrow\mathcal{S}(u,\rho)\end{array}\right]$$

PSHVZK AoK implies that even if an adversary chooses a distribution over statements and witnesses, it isn't able to distinguish between simulated transcript and honestly generated transcript for $u\in\mathcal{L}_\sigma$.

We will be using these definitions of Zero knowledge argument and its properties in the discussion further without redefining them, unless explicitly stated.

Chapter 3

Literature Survey

Chapter 4

Materials and Methods

4.1 Including Figures

Chapter 5

Results and Discussions

5.1 Including Tables

Tables are to be used in a special environment so that they have a Number, caption and appear in the list of tables. Table 5.1 is a sample table. In the case of tables, it is a convention to write the caption above the table. Note that in the case of figures the caption appears below the figure.

Table 5.1: Physical properties of the materials used.

Property	Value
Particle Density, ρ_{p}	2500 kg/m^3
Viscosity, η_{s}	1 $\times 10^{-3}$ Pa-s

Appendix A

Supporting Material

References

[1] IDEX blog. A complete list of cryptocurrency exchange hacks [updated]. [Accessed 27-MAY-2020]. [Online]. Available: https://blog.idex.io/all-posts/a-complete-list-of-cryptocurrency-exchange-hacks-updated

[2] Wikipedia contributors. Mt. Gox — Wikipedia, the free encyclopedia. [Accessed 10-JUNE-2020]. [Online]. Available: https://en.bitcoin.it/wiki/Mt._Gox

[3] 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)*, May 2018, pp. 315–334.

[4] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[5] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, pp. 3416–3452, 2018.

[6] N. v. Saberhagen, "CryptoNote v 2.0," White paper, 2013. [Online]. Available: https://cryptonote.org/whitepaper.pdf

[7] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, 2014, pp. 459–474.

[8] "Grin project website." [Online]. Available: https://grin-tech.org/

[9] "Beam project website." [Online]. Available: https://www.beam.mw/

[10] A. Poelstra, "Mimblewimble," 2016. [Online]. Available: https://download.wpsoftware.net/bitcoin/wizardry/mimblewimble.pdf

[11] A. H. Koblitz, N. Koblitz, and A. Menezes, "Elliptic curve cryptography: The serpentine course of a paradigm shift," *Journal of Number Theory*, vol. 131,

no. 5, pp. 781 – 814, 2011, elliptic Curve Cryptography. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022314X09000481

[12] S. Vijayakumaran. (2017) An introduction to Bitcoin, 2017. [Online]. Available: https://www.ee.iitb.ac.in/~sarva/bitcoin/bitcoin-notes-v0.1.pdf

[13] P. Hess, "Sec 2: Recommended elliptic curve domain parameters," 2000.

[14] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, p. 120âĂŞ126, Feb. 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[15] R. W. F. Lai, V. Ronge, T. Ruffing, D. Schröder, S. A. K. Thyagarajan, and J. Wang, "Omniring: Scaling private payments without trusted setup," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, November 2019, p. 31âĂŞ48.

[16] J.-J. Quisquater, M. Quisquater, M. Quisquater, M. Quisquater, L. Guillou, M. A. Guillou, G. Guillou, A. Guillou, G. Guillou, and S. Guillou, "How to explain zero-knowledge protocols to your children," in *Advances in Cryptology — CRYPTO' 89 Proceedings*, G. Brassard, Ed. New York, NY: Springer New York, 1990, pp. 628–631.

List of Publications

[1] S. Bagad and S. Vijayakumaran, "On the confidentiality of amounts in grin," in *Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2020.

[2] S. Bagad and S. Vijayakumaran, "Performance trade-offs in design of mimblewimble proofs of reserves," in *IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2020.

Acknowledgements

I would like to express my deepest appreciation and gratitude to my guide Prof. Saravanan Vijayakumaran for his guidance and constant supervision and help throughout the last 16 months. His insightful suggestions and comments have time and again helped me get a more in-depth understanding of the field. His most valuable lesson for me, from amongst many others, was to be persistent especially in times when things are not going as planned. Not only did he offer consistent assistance with regards to the academic work but also gave invaluable suggestions in helping me carve out my career path. Thanks are also due to Arijit Dutta of IIT, Bombay for his many useful remarks and discussions about the project and otherwise. I also acknowledge the help offered by my brother Piyush Bagad (Wadhwani AI) in understanding Python as well as in running simulations. Finally, I would also like to thank my dear friend Madhura Pawar for her constant encouragement which helped me work towards my goal even when things weren't working out.

Suyash Bagad
IIT Bombay
20 June 2020