

ISTIC Informatique-Électronique

Faculty of Electrical Engineering,
Mathematics and
Computer Science

Master's Degree in
Cyber Security

Zero-Knowledge Proofs Applied to Finance

Supervisor
Pierre-Alain Fouque

Supervisor
Joris Cramwinckel

Student
Elvira Sanchez Ortiz

Academic Year 2019-2020

Synthesis

General Context

A Zero Knowledge Proof (ZKP) is a cryptographic tool that allows a party to prove knowledge of a secret to another party without revealing said secret. ZKPs have been around in the literature since the mid-80s [1]. But with the dawn of Distributed Ledger Technology (DLT) the development of new zero-knowledge proof protocols has opened the door for new possible applications.

Most economic transactions involve information asymmetry [2], where one party holds more information than the other. In a transparent world, a contract would guide both parties' behaviors. However, confidentiality plays a major role in the incentive of doing business. Therefore, in many financial activities there is a trade-off between transparency and confidentiality. An example of this trade-off is investment management, where there is an *asset owner* who wants their wealth to be managed by an *asset manager* in an investment fund. In this case, asset managers keep their portfolio private, because revealing it would trump their competitive advantage and give away their strategy. However, asset owners want to know that their money is being managed properly and according to the restrictions imposed by them (e.g. risk restrictions, exposure to a certain sector, etc.). Solving this problem nowadays usually involves a Trusted Third Party (TTP) to both, agent and principal, will have access to the information and reveal only what is necessary for regulatory purposes.

Applying ZKPs to cases where there is the aforementioned trade-off between confidentiality and transparency could provide a way to add more transparency in these activities without compromising the confidentiality.

Research Problem

The research question is *"What use cases in Finance can benefit from zero-knowledge proofs and is it feasible to implement them with the state of the art technology?"*

Financial regulation for the *transparency versus confidentiality* setting is complicated. Currently this is done through the role of a TTP, like a custodian or a notary. This method is expensive and time inefficient, since the data has to go through the TTP and they report to the client with a time-lag. The use of zero-knowledge proofs would mean instant automated proofs.

This is not a new problem. There are two works that stand out on zero-knowledge proofs applied to regulatory finance. One from 2005 [3] where they implement ZKPs for portfolio reporting, and one from 2018 [4] where they propose different applications to the

domain. However, the technology has greatly advanced since the implementation of [3] and in [4] there is just a mathematical description of the protocols. The aim of this thesis is to propose new applications and to research the implementation of, at least, one of the use cases with the most recent and mature technology available.

Contribution

In order to carry out the project, the first step was desk research on the literature. Once we had a few use cases, we interviewed people from the company who were familiar with the topic to validate the ideas. Three applications were proposed, namely *portfolio reporting*, *blind real estate auctions*, and *dark pools and blind bids in stock market*. The next logical step was to prototype one of the applications. We focused on the implementation of a risk reporting tool where an asset manager can prove their portfolio is within the risk constraints set by the asset owner, without revealing the composition of the portfolio.

Two iterations for the use case were carried out with two different ZKP technologies (i.e. Bulletproofs and zk-SNARKs). For the second iteration a prototype of an application was built in order to illustrate and show the feasibility of the application.

Results

The prototype is built with open source libraries that are used and reviewed by the community. This work brings awareness to the fact that these technologies are quickly evolving and becoming more mature. The proofs generated are only valid when all the constraints in the designed circuit are satisfied.

The review of the working assumptions and limitations for this application has been the main result of the project as well as the trade-offs between the different solutions. Out of these assumptions, the one that hinders its viability the most is the relationship between real world and virtual world. Financial assets are not digital native. Therefore, in order for the application to be fully trusted and to make sure the representation of the data is done correctly, there has to be the assumption that the prover is honest (i.e. introduces the right data into the proof).

Summary and Future Work

Different applications of zero-knowledge proofs to financial regulation are reviewed and proposed in order to tackle the trade-off between confidentiality and transparency. Moreover, a prototype for *private portfolio reporting* is built in order to prove its feasibility.

The next logical steps would be to try to challenge the assumptions and try to find solutions to them. For example, companies like Symbiont or Dusk are working on building a decentralized financial market infrastructure, and a collaboration with them could bring some answers to the relationship between real and virtual world.

Moreover, a real estate blind auction is proposed as an use case for zero-knowledge proofs. Another line of work would be to develop a protocol for this use case.

Acknowledgement

I would like to thank Joris Cramwinckel for his support throughout the internship. My parents for supporting me through my studies and always. Andreas Peter for his guidance. A big thanks to my Grandma and Francesco.

Rennes, France, August 2020 Elvira Sanchez Ortiz

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Between the months of February and August 2020, I carried through my internship at Ortec Finance, in Rotterdam, The Netherlands. Ortec Finance is a company that provides software solutions for risk and return management. During my internship, I was a part of the R&D Lab, which performs research within the company. Within the team, my task was to research different use cases for the application of zero-knowledge proofs in finance. I worked on my thesis three days a week and dedicated the other two days to the position of Student Assistant in the same company, where I could dedicate my time to other projects as a part-time employee.

A ZKP is a type of cryptographic tool that allows a party to prove knowledge of a secret to another party without revealing said secret. ZKPs have been around in the literature since the mid-80s. However, for years, its research has been mostly theoretical and not practical for real-life applications due to high communication and computational complexity. Nonetheless, in recent years, the development of new zero-knowledge proof protocols, linked to the dawn of the distributed ledger technology, has opened the door for new research on possible applications. Especially, in Ortec Finance, the interest is set on the financial sector, since it is their area of expertise.

The financial field is known for its lack of transparency and the constant contrast between tight regulations and secrecy. Within the field, there is information asymmetry, where some parties hold more information than others, and the parties that hold greater amount of information profit from that. This topic will be dealt with in more depth in following sections, but it is important to superficially present this characteristic of the financial sector in order to explain why zero-knowledge proofs might be useful tools to address the trade-off needed between transparency and privacy.

1.1 Research Question and Contribution

The main goal of this thesis is to research the state of the art of zero-knowledge proofs and its implementations and to look into the financial sector to find possible use cases where zero-knowledge proofs can provide a solution to information asymmetry, so characteristic to this domain. The research question can be formulated as follows:

"What use cases in Finance can benefit from zero-knowledge proofs and is it feasible to implement them with the state of the art technology?"

The contributions of this work are two-folded. First, the proposition of a number of different applications where zero-knowledge proofs can play a role in financial transactions. Second, the implementation of the zero-knowledge risk reporting application proposed by [4] making use of the state of the art protocols and available libraries and the review of the resulting trade-offs between technologies, limitations and assumptions.

1.2 Structure of the Thesis

The rest of the thesis is structured as follows. In the following section and Section 1.2, a brief description is given of the company where the internship was carried out and the topic of the internship, respectively. Chapter 2 contains a literature review on zero-knowledge proofs, including definition, history, cryptographic primitives, definition of the most popular technologies and a survey of some popular applications. Chapter 3 contains an introduction into the Financial sector and its characteristics, as well as an overview of different use cases of zero-knowledge proofs in the domain. Chapter 4 includes the description of the implementation of zkSNARKS and Bulletproofs for Portfolio Risk Reporting. Chapter 5 discusses the assumptions and limitations of the implementation built, and of zero-knowledge proofs in general. Finally, Chapter 6 is a conclusion and a review of future lines of work on the topic.

1.3 The Company and Business Sector

As mentioned above, Ortec Finance is a "leading provider of technology and solutions for risk and return management" [5]. The company delivers technology and solutions for investment decision-making to financial institutions all over the world. Among the solutions they provide are specialized prediction software for pension funds, housing associations in the Netherlands, banks and insurance companies. Ortec Finance has over 250 employees and is present in Rotterdam, Amsterdam, London, Toronto, Zurich, Melbourne and Hong Kong.

It is an innovative enterprise that has a research and development department, R&D Labs, where there is constant experimentation with new modelling approaches and IT techniques in order to find new applications to investment decision making or other topics within the financial domain, like is the case of this work. The company has a close relationship with the academic world, constantly collaborating with students and universities.

The business sector is mainly the risk and return management solutions. This entails the process of identification, analysis and acceptance or mitigation of uncertainty in investment decisions [6]. It occurs everywhere in the realm of finance, given that an inadequate risk management might result in severe consequences for companies, individuals and economies. Therefore, the use of accurate prediction models is necessary for any financial venture. With the technology advances and the increase of data gathering, new ways of predicting risk scenarios are being developed. In order to do this in a satisfactory way, a deep expertise in Finance and econometrics as well as in technology and software development is essential.

1.4 The Internship Topic

The core focus of the internship was put on the research of recent developments of zero-knowledge proofs and on finding applications where this technology would be suitable for the financial domain.

In order to carry out the research, the first months consisted mostly of literature review. Given the multidisciplinary character of the company, and thus the internship, the thesis required not only to achieve familiarity with zero-knowledge proofs, their mathematical bases and their implementations, but also with the financial sector, its players, and its needs. After acquiring the basic knowledge, the next steps focused on the implementation of a working prototype of an already known application (i.e. portfolio reporting). This took a thorough work of investigation of the technologies and the libraries available—and mature enough—at the time of writing. After selecting the most fitting technologies, the efforts were put in the implementation, which lead to reviewing some assumptions and limitations. Since the project goal was to find different viable applications for zero-knowledge proofs, the work did not stop with the risk reporting tool. In order to bring another approach to the project, another potential application (i.e. Real Estate Auctions) was proposed, along with the goal of creating a protocol that, using zero-knowledge proofs and other cryptographic primitives, could provide a secure and private way of performing real estate auctions. However, the thesis will explain the first application in depth, leaving only a brief description of the second application on Chapter 3 which can be left as a proposal for future work.

On the figure below, the reader can see the planning of the different tasks during the duration of the internship.

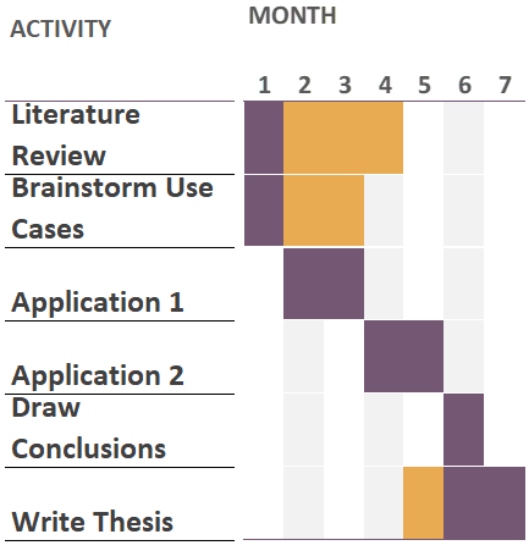


Figure 1.1: Gannt chart with the projected (purple) and actual (orange) duration of each planned task

The daily tasks varied between reviewing literature, to find the most interesting technologies as well as cryptograpic building blocks, and trying to apply the state of the art technology to already researched applications in finance, as well as trying to find new applications within the sector.

Chapter 2

Literature Review

2.1 Introduction

A zero-knowledge proof is a cryptographic tool that presents a way to prove knowledge that a statement is true without revealing anything other than the validity of said statement. These proofs were first proposed in the decade of the 1980's, but practical implementations have not come until recently, with the blooming of distributed ledger technology and cryptocurrencies. Thus, with the new protocols in place, the scope of possible use cases opens up. In this thesis, the main point of interest is the application of zero knowledge proofs to finance, especially regulatory finance, where parties have to demonstrate that they are playing by the rules. Therefore, the following section will give an overview of zero-knowledge proofs, including definition, background, history and different applications already proposed.

2.2 Zero-Knowledge Proofs

An intuitive definition of zero-knowledge proof was given by [7] in 1998, by making an analogy between zero-knowledge proofs and the Ali Baba cave. The article explains zero-knowledge proofs through the story of Ali Baba, who—after being mugged by thieves forty times—discovers they are escaping by going into a round cave (see figure 2.1) with a secret gate in the middle, that can only be opened with the words *Open Sesame*. Years after this event, someone (Mick Ali) discovers the cave and wants to prove to the journalists that he knows the secret password that opens the gate without revealing it. In order to do that, the person is first filmed going into the cave via both ways until the dead end where the gate is. Then the journalists go out of the cave and Mick goes back in choosing one of the both sides (A or B in 2.1). Once he gets to the dead end the journalist goes to the bifurcation of the paths and tells Mick to come out from one of the sides. The first time they do

this, the chance that Mick was just in the right side is 50%. However, by repeating it enough times, the probability of Mick not knowing the password decreases to be negligible.

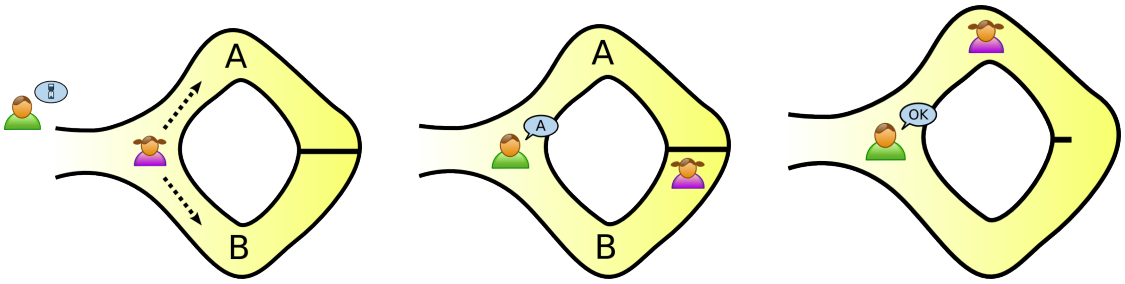


Figure 2.1: Ali Baba cave from [8]

2.2.1 Definition

Zero-knowledge proofs were first presented by Goldwasser, Micali and Rackoff in 1985 [1]. In these kinds of protocols, a prover, P, wants to convince a verifier, V, of the validity of a statement, without leaking any other information to the verifier. In order to define zero-knowledge proofs, we have to first define the view of V on x.

Definition 1. Zero Knowledge Proofs [9]. We say that an interactive proof system $\langle P, V \rangle$—with P, V Turing Machine (TM)—is (computational) zero-knowledge for a language L if for every Probabilistic Polynomial Time (PPT) verifier V^* there exists a PPT simulator S—also a Turing Machine—such that

$$\forall x \in L, z \in \{0,1\}^*, View_{V^*}[\, P(x) \Leftrightarrow V^*(x,z)] \, = S(x,z) \tag{2.1}$$

with $View_{V^*}[\, P(x) \Leftrightarrow V^*(x,z)]$ being all messages sent from P to V as well as random bits used by V during the execution of the protocol on x.

In other words, the system is zero knowledge if, for any verifier V^*, whatever they learned by interacting with the prover, can be learned by running an efficient simulator S with the same input. Therefore, since S does not know the solution, the verifier cannot have gained any additional information.

A zero knowledge proof must have three basic properties:

1. **Soundness [10]:** this property ensures that a cheating prover (who sends the prove $M(x)$ can successfully lie to an honest verifier with a negligible probability (negl).

$$if\ x \notin L \implies Prob(M(x) = ACCEPTED) \leq negl \tag{2.2}$$

2. **Completeness [10]:** this property ensures that an honest verifier will always accept a correct witness.

$$if\ x \in L \implies Prob(M(x) = REJECTED) \leq negl \tag{2.3}$$

3. **Zero Knowledge [11]:** this property ensures that V does not learn any aditional information from its interaction with P.

$$\forall\, V^*, \exists\, PPT\ Sim_{V^*}\ such\ that\ \forall x \in L$$

$$View_{V^*}[\, P(x) \Leftrightarrow V^*(x,z)] \approx S(x,z) \tag{2.4}$$

2.3 Cryptographic Preliminaries

In order to understand zero-knowledge proofs there are a series of cryptographic primitives that need to be understood. The definitions are taken from Nigel Smart's book [12].

2.3.1 Pedersen Commitments

"Commitment schemes arise out of the need for parties to commit to a choice or value and later communicate that value to the other parties involved in such a way that is fair to all the parties" [13]. The most common commitment scheme is called Pedersen Commitment.

Given g, h random generators of a finite abelian group G of prime order q, such that no user in the system knows the discrete logarithm of g ad h, we define $B_a(x)$ as the Pedersen Commitment of x

$$B_a(x) = h^x \cdot g^a \tag{2.5}$$

where a is called the blinding value because it blinds the value of x to a computationally unbounded adversary. When the commitments are revealed, the user must publish the pair (a, x).

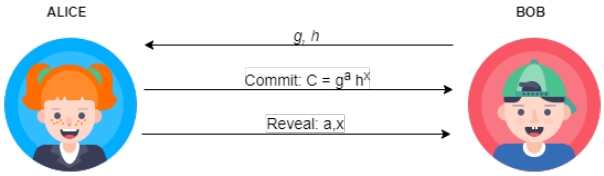


Figure 2.2: Pedersen Commitment Communication Protocol

Commitments must satisfy two properties:

- **Binding:** A commitment scheme is information-theoretically (resp. computationally) binding if no infinitely powerful (resp. computationally bounded) adversary can output two values (x', a') such that $x' \neq x$ and $C(x, r) = C(x', a')$.

- **Hiding:** A commitment scheme is information-theoretically (resp. computationally) hiding if no infinitely powerful (resp. computationally bounded) adversary can generate two messages x_0, x_1 of equal length such that she can distinguish between their corresponding commitments C_0, C_1.

The Pedersen commitment scheme is computationally binding and information-theoretically concealing.

2.3.2 Sigma-protocols

A Σ-protocol is a three-move protocol—hence its name: the prover goes first with a commitment phase, consequently the verifier responds with a challenge, and in the end, the prover sends a final response. This way the verifier can verify the proof. Σ-protocols involve the assumption of an honest verifier (i.e. the verifier follows the protocol correctly).

- **Schnorr's Identification Protocol:** Schnorr's protocol allows a prover to proof the knowledge of the discrete logarithm x of y with respect to g in some finite abelian group G of prime order q. The protocol goes as follows:

  $P \to V : r \leftarrow g^k \; for \; a \; random \; k \leftarrow \mathbb{Z}/q\mathbb{Z}$,

  $V \to P : e \leftarrow \mathbb{Z}/q\mathbb{Z}$,

  $P \to V : s \leftarrow k + x \cdot e \; (mod q)$.

  This way the verifier can check that the prover knows the discrete logarithm of y by verifying that $r = g^s \cdot y^{-e}$.

2.3.3 Fiat-Shamir Heuristic

The interactive nature of zero-knowledge proofs diminishes their applicability. Fortunately, it is possible and straight-forward to turn an interactive zero-knowledge proof into a non-interactive proof by replacing the verifier's challenge component with a hash of the commitment and the statement. This is known as the Fiat-Shamir Heuristic. For example, the Schnorr proof of knowledge of discrete logarithms can be made non-interactive by making the challenge $e \leftarrow H(r||g||y)$.

2.4 History of Zero-Knowledge Proofs

Zero knowledge proofs have been a part of cryptography research for almost forty years. However, the implementations for actual applications have only recently bloomed, as can be seen below on the zero-knowledge progress timeline, with

advances in the technology that make it more efficient and light-weight. The dawn of DLT and its need to protect data and to scale efficiently have been the principal causes of the research boost on zero-knowledge proofs.

- **1985**: Publication of "The Knowledge Complexity of Interactive Proof-Systems". It was the first paper on zero knowledge proofs. The document explains the mathematical basis of the proofs in an informal way. At the time, zero-knowledge proofs are "horribly inefficient" due to the size of the proofs and the needed interaction between prover and verifier [1].
- **1988**: The first publication proposing Non-Interactive Zero-Knowledge (NIZK) by Blum, Feldman, and Micali [14].
- **1992**: "A note on efficient zero-knowledge proofs and argument" [15], by Joe Kilian, is the first publication on succinct (i.e. small, compact) zero-knowledge proofs.
- **1994**: Goldreich and Oren publish a paper where they propose the first succinct *non-interactive* zero-knowledge proof.
- **2006**: Groth proposes the first linear size proofs [16]. Proof sizes start getting smaller.
- **2011**: Publication of first paper on Succinct Non-Interactive Adaptive Argument of Knowledge (SNARK) [17].
- **2016**: Groth proposes an algorithm that reduces significantly computational complexity of Zero-Knowledge Succinct Non-Interactive Adaptive Argument of Knowledge (zkSNARK) [18]. The fastest and smallest known zk-SNARK to the time. Used in Zcash.
- **2017**: The publication on Bulletproofs [19], "non-interactive zero-knowledge proof protocol with very short proofs and without a trusted setup".
- **2018**: Zero-Knowledge Scalable Transparent ARguments of Knowledge (zk-STARK) are proposed on [20]. zk-STARKs are Succinct Non-Interactive Adaptive Argument of Knowledges with a transparent setup, or in other words, wihtout a trusted setup.
- **2019**: Zero-Knowledge Succinct Non-interactive Oecumenical aRguments of Knowledge (zkSNORK) are proposed (e.g. Sonic [21], Plonk [22] ...). Sonic supports a universal and updatable common reference string. Proofs are constant size. However, verification is expensive. Many of newest constructs later this year are based on Sonic. Libra protocol was also proposed [23], which is a protocol that yields a ZKPs with linear prover time and succint proof size and verification time. However it needs a trusted setup. Its follow-up, Virgo [24] does not require a trusted setting.

Once again, it is interesting to remark the fast evolution of zero-knowledge proofs on the last ten years, as opposed as the first three previous decades.

2.5 Technology Description

2.5.1 zk-SNARKs

zk-SNARK is the best known type of zero-knowledge cryptography. As its name indicates it is succint, which means that it can be verified within milliseconds with a proof of length of a few hundred bytes, and non-interactive, which means that the prover can create the proof without the need to communicate with the verifier [25]. In the following lines a high level summary of how zk-SNARKs work is provided. zk-SNARKs work by transforming the statement that needs to be proven into algebraic equations. In short, the statement has to be transformed into an arithmetic circuit, and from it build a Rank 1 Constraint System (R1CS). A R1CS is a sequence of groups of three vectors (a, b, c) and the solution to the constraint system is a vector s that satisfies the equation $s.a * s.b - s.c = 0$, where . is the dot product [26].

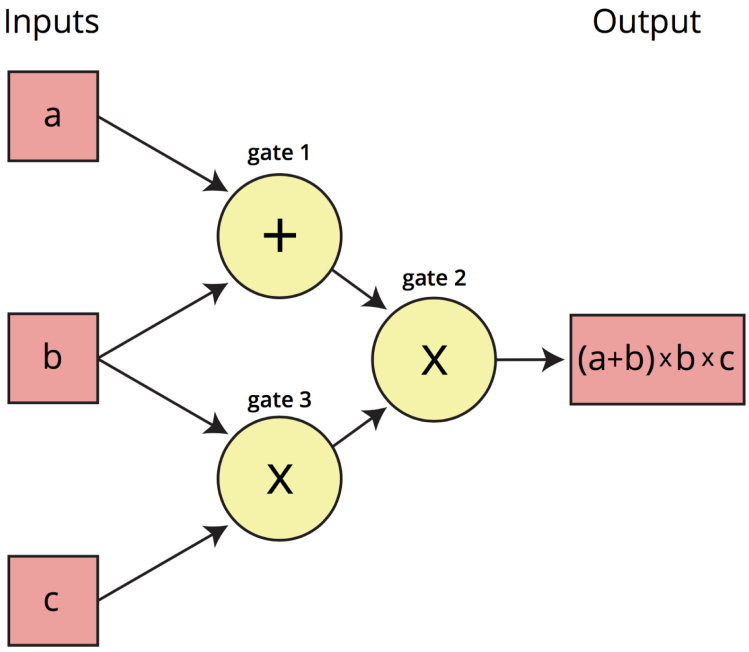


Figure 2.3: Example of arithmetic circuit [25]

With R1CS there are different constraints for almost every wire in the arithmetic circuit. Therefore, it is important for the *succinctness* of the proof to bundle them together, and transform the R1CS into a Quadratic Arithmetic Program (QAP), which follows the same logic as R1CS but using polynomials instead of dot products. This transformation is performed using *Lagrange interpolation*. This way, the verifier only needs to check that two polynomials match at one random point to correctly verify the proof with a high probability [25].

zk-SNARKs uses Homomorphic Encryption and pairings of elliptic curves to

blindly—without knowing what point is evaluated—evaluate polynomials, so the prover cannot create a fake proof that satisfies the identity at that point. In order to achieve zero-knowledge, the prover uses *random shifts* of the original polynomials that satisfy the identity.

The non-interactive nature of zk-SNARKs makes it necessary for the prover and the verifier to have a *Common Reference String (CRS)*, which provides a way of knowing that the proving and verification keys in the protocol were generated by the same set-up algorithm. This CRS prevents the prover from cheating. This requires the verifier to have a full representation of the statement being proven. However, with large statements this might result in slow verifications. In order to make zk-SNARKs efficient, a pre-processing step is added to create a *Structured Reference String (SRS)* that will be available to both parties. The problem with this pre-processing step is that it relies on non-public randomness (or toxic waste) which can be used to fake proofs if it is not properly deleted. Therefore, for setting up zk-SNARKs, there needs to be a *setup ceremony* where the probability of this toxic waste to not be discarded is almost negligible. More can be read about the setup ceremony of zcash in [27].

2.5.2 Bulletproofs

The second main protocol to implement zero-knowledge proofs is called Bulletproofs. As explained above, Bulletprooofs was published in 2017 by Bünz et al. with the goal of enabling efficient confidential transactions in Bitcoin and other cryptocurrencies by proving that a secret commited value lies within certain range [19]. It also supports aggregation of range proofs in a single proof. Apart from range proofs, Bulletproofs can provide zero-knowledge proofs for arithmetic circuit by relying only on the discrete logarithm assumption. Bulletproofs build on Pedersen Commitments and Inner Product Proofs. However, in this thesis we will not go into greater detail. In figure 2.4 an overview from [28].

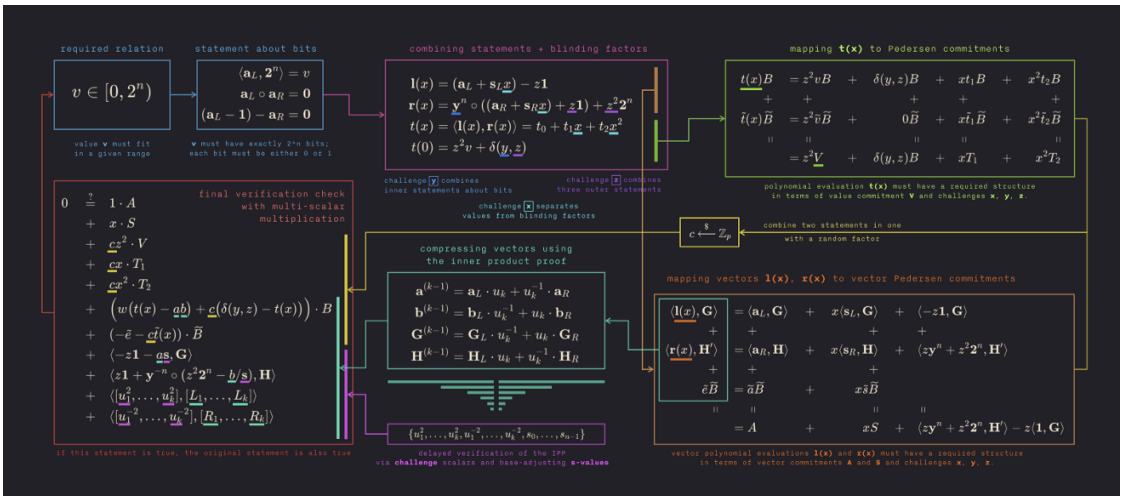


Figure 2.4: Stylized overview of Bulletproof protocol [28]

Unlike zk-SNARKs, Bulletproofs do not require a trusted setup. This feature makes them less controversial security-wise, but the toll is taken on the verifying time, longer than in zk-SNARKs.

2.5.3 Trade-offs in Current Solutions

Zero-Knowledge Proof protocols are hard to scale for large statements due to a high overhead on generating the proof. In this subsection, a brief discussion is given on the trade-offs in the different solutions. Vitalik Buterin talks about trade-off between proof size (in bytes) and security assumptions [29] (see Figure 2.5).

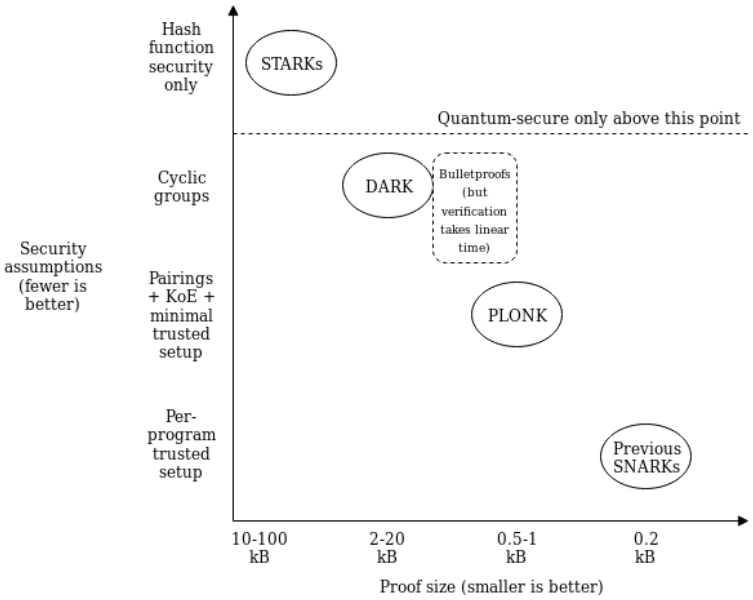


Figure 2.5: Situation of different ZKP protocols according to the axes *security assumption* (y) and *proof size*(x) [29]

However, we would also like to address the time complexity, which encompasses proving and verification time, as an average of both. Current solutions are situated in different planes formed by different values along these three axes, trading off some features for others. No solution has yet achieved a small proof size with transparent arguments and low time complexity. Table 2.1 shows different zero-knowldedge proof protocols and a high level classification of each on the three different parameters. Proof size refers to the amount of bytes the proof takes. Transparent arguments refers to the need of having a trusted setup and is equivalent to the security assumptions in the Figure 2.5. If the protocols have transparent arguments it means that no trusted setup (i.e. no toxic waste when creating public parameters) is needed and therefore there have weaker security assumptions, which translates in better security.

Algorithm	Proof Size	Transparent Arguments	Time Complexity
zkSNARK	smallest	no	low
zkSTARK	big	yes	low
Bulletproof	medium	yes	high
Aurora	big	yes	medium
Sonic	small	no	low
Super-sonic	medium	yes	medium
Fractal	big	yes	medium
Libra	medium	no	low

Table 2.1: Comparison of different algorithms according to the parameters proof size, transparent arguments and time complexity

The *low* time complexity in zk-STARKs is relative. It has a poly-logarithmic time-complexity distribution, as opposed to Bulletproofs that have a logarithmic distribution. This means that for small proofs, the poly-logarithmic complexity is faster, but as proofs grow in size, there will be a threshold where the logarithmic distribution will become more efficient. However, since the proof size for zk-STARKs is big, they will probably never be used for big proofs, maintaining for the small ones the fast prover and verifier times.

These trade-offs have to be considered when designing a zero-knowledge proof application. Depending on the purpose of the application a technology will be more or less suitable. For example, if the aim of the zero-knowledge proof is to scale a blockchain it will be important to have a small proof size, but if the proof always follows the same circuit, one reliable set-up ceremony will be accepted in order to get a smaller proof size. However, if the purpose of the application is to prove different things and the circuit or input size might change then a technology with transparent arguments or universal set-up would be ideal.

2.6 General Applications

- **E-voting**: This application was proposed by Groth in 2005. In electronic voting it is important to keep the vote private. However, when the voter encrypts their vote and sends it to the authority that tallies it, they can cheat if there is no way of knowing whether the encrypted value is correctly formed. Non-interactive zero-knowledge proofs can be used for proving that the vote was cast in a valid format [30] .

- **Nuclear Disarmament**: This might be one of the most creative applications for zero-knowledge proofs. Zero-knowledge proofs can be used to process

classified physical data, namely in the field of nuclear arms control. This application uses a non-electric fast neutron differential radiography technique that can confirm two objects are identical without revealing their geometry or composition. This way the authenticity of nuclear weapons could be confirmed without revealing any secret design information [31].

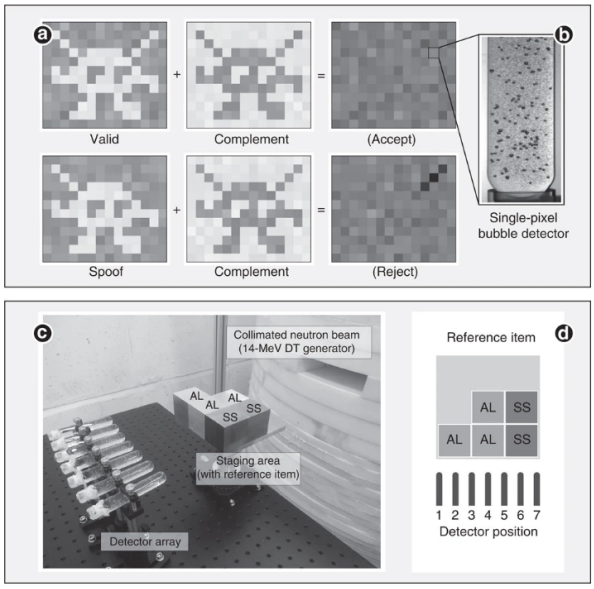


Figure 2.6: Image of the experiment to create a zero-knowledge object comparison system for nuclear disarmament by [31]

- **Cryptocurrencies**: zero-knowledge proofs on the distributed ledger might be the best known use case. The development of all the different distributed ledger technologies after the creation of Bitcoin, and the need to address its privacy and scalability issues, have been essential for the recent evolution in zero-knowledge proof technology. There are several cryptocurrencies that use zero-knowledge proofs, namely zk-SNARKs and Bulletproofs, to provide privacy and anonymity to their transactions. Among these cryptocurrencies are Zcash, Monero [32], and Grin [33].

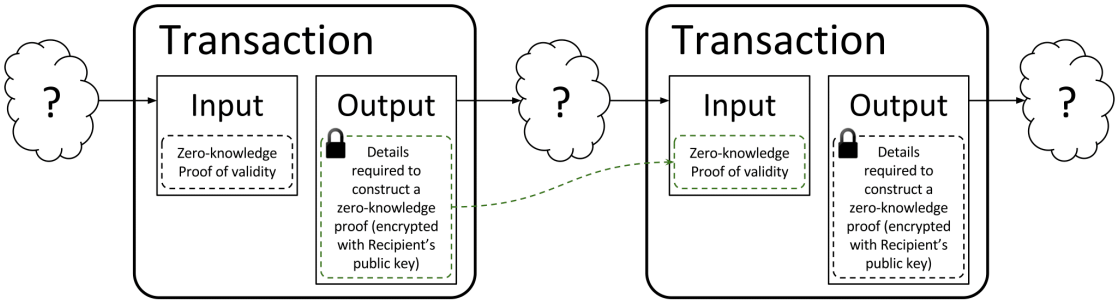


Figure 2.7: Shielded transaction in Zcash [34]

Zcash, for example, uses zk-SNARKs in order to construct a proof to validate that the input values sum to the output values, and that the sender has

the private spending keys of the input notes [25]. Moreover, there are some solutions to add a zero-knowledge layer in Ethereum [35].

- **WPA3**: The third generation Wifi Protected Access (WPA) was introduced in 2018 by the Wi-Fi Alliance. The new version aims to replace its faulty predecesor WPA2. WPA3 addresses the security issues by securing the password, and uses zero-knowledge proofs to avoid transmitting elements of the password over the network. Afterwards both parties pass their knowledge of the password, and both can prove that they know the secret password [36]. The protocol of zero-knowledge proofs used by WPA3 is called Dragonfly.

Chapter 3

Applications in Finance

3.1 Introduction

In economics there is a well known theory called the theory of asymmetric information, first proposed in 1970 by George Akerlof [2]. Information asymmetry happens in economy when a party to a transaction possesses more knowledge than the other party. Almost all economic transactions involve information asymmetry [37]. This information asymmetry between *buyers* and *sellers* leads to what is known as adverse selection and the principal-agent problems. Adverse selection describes the process where some parties are able to use their *private* knowledge of risk factors involved in a transaction to maximize their outcomes, at the expense of other parties. The most prominent example is usually the car market, where there are good cars—*peaches*—, which are worth more, and malfunctioning cars—*lemons*—, which are worth less. Since the buyer does not know whether the seller is selling a *peach* or a *lemon*, they will not be willing to pay as much as they would pay if they knew the car was a *peach*. Although at the price they are willing to pay, only sellers selling *lemons* will accept the offer. The principal-agent problem arises as a consequence of the information asymmetry as well; "how can a principal (e.g. the buyer) get an agent (e.g. the seller) to behave how they want, when they cannot monitor them all the time [38]?". In an ideal and transparent world, a contract would guide the agent's behavior, without need for incentivising them. However, in most settings in finance, *confidentiality* plays a major role in the competitive advantage or is the incentive for doing business.

In other words, there is a fine line between confidentiality and transparency in order to keep competitive incentives while being able to operate. Usually regulatory institutions look over and make sure that the actors play according to the rules in order to keep the trust in the system and the party with the greater information *signal* [39] that they are honest players by having credentials or a good reputation. However, in these kinds of scenarios where it is important to know that certain conditions are being met, but at the same time it is important to keep some aspects

confidential, zero-knowledge proofs could introduce more transparency without the need of disclosing certain sensible information.

The area of cryptography for finance is one studied thoroughly. Nonetheless, for the use of zero-knowledge proofs, there has not been a great deal of publications in recent years outside of the distributed ledger domain. Two papers worth mentioning are [3] and [4].

The following sections propose different applications of zero-knowledge proofs for cases where there is the aforementioned trade-off between confidentiality and transparency, namely portfolio reporting, real estate blind auctions, and dark pools and blind bids.

3.2 Portfolio Reporting

A normal scenario for individuals who want to invest their savings is to give it to a hedge fund or other sort of investment fund. In this setting, the individual who owns the money is called an *asset owner*, and the agent that manages it is called an *asset manager*. It can be the case that the asset owner has some preferences on where to invest their money. Some common restrictions might be to limit the risk of the investments or to invest in companies that practice Corporate Social Responsibility (CSR), among others. The issue at hand is that the composition of the portfolios held by asset managers is kept secret, thus making it challenging for the asset owners to verify whether their preferences or constraints are being followed. Investment funds usually prove they are doing their job right by making returns for the *asset owner*, but no (or hardly any) proof is given that they are doing it by following the constraints imposed by their clients.

The two cases of portfolio reporting we are going to focus on in the following subsections are aggregated risk reporting and temperature score reporting.

3.2.1 Risk Reporting

Different investors have different risk appetite, and when they invest in a fund, they usually specify their risk threshold. This risk can be measured, for example with the volatility. Volatility is a "statistical measure of the dispersion of returns for a given security or market index." [40]. Therefore, in general, a higher volatility means a riskier security. Asset volatility is a measure that is publicly available for every company in the stock market. Other measures that can indicate the risk of a security are the beta or the Sharpe ratio.

In the case of risk reporting, the current solution is for the asset managers to periodically send a calculation of the weighted average risk of their portfolio (R), by adding the individual risk (r_i) of the assets that make up the client's investment

multiplied by the relative weight of each asset in the portfolio (w_i). Hence, the client can verify it is within their risk limits (r_{min}, r_{max}).

$$r_{min} \leq R = \sum_{i=1}^{N} w_i r_i \leq r_{max} \tag{3.1}$$

This brings some concerns, since the asset manager could lie about these values or the asset owner could learn things about the composition of the portfolio, given the individual risk measures are publicly available.

This is an ideal case for applying zero-knowledge proof, since there is a clear trade-off between transparency and confidentiality. This application has been proposed already by [3], [4]. In this example, there is information asymmetry between the asset managers and the asset owners. The asset managers hold more power since they are the only ones who know the assets that compose the portfolio. On the other hands, the asset owners get the short end of the stick, having to trust that their wealth is being correctly managed.

The trade-off between transparency and confidentiality is unavoidable. If there were full transparency and the asset owners had access to the composition of their portfolio, it would be a risk for the asset owners, since other funds might find out what their investing strategy is. Moreover, if the asset managers are exploiting arbitrage opportunities, making their strategy public would eliminate their competitive advantage, since other funds could learn about it and copy the strategy.

However, total confidentiality—the closest to the current scenario—creates an ideal environment for theft, fraud and conflicts of interest, leaving the asset owner unaware of the risks their portfolio is exposed to.

Thus, by using zero-knowledge proofs, the asset manager could provide a proof that the weighted average risk of their client's portfolio is lower than a certain risk value without revealing it, using a range proof. Another solution would be to provide a proof of correctness of the calculation of said risk, without revealing the weights or the number of assets that make up the portfolio.

In the next chapter, an implementation of this use case is proposed.

3.2.2 Temperature Score Reporting

Another noteworthy case within portfolio reporting is that of proving that the portfolio securities correspond to companies that are making efforts towards following the goals of the Paris Agreement.

"The Paris Agreement is the first-ever universal, legally binding global climate change agreement, adopted at the Paris climate conference (COP21) in December

2015" [41]. It was signed by 195 governments and it sets out a goal to keep global warming below 2°C.

As the awareness about climate change penetrates into society, individuals start to realize that it is not only government's responsibility to impose the necessary changes. Climate change risk and opportunity assessment enables investors to identify the risks that could potentially affect the portfolios and the opportunities linked to transitioning to a lower carbon economy. According to *The Economist*, not considering the implications of climate change can be utterly harming to a portfolio. They estimate that climate change will cause the loss of $4.3 trillion in assets due to damage caused by events like droughts, floods and storms, directly related to climate change [42].

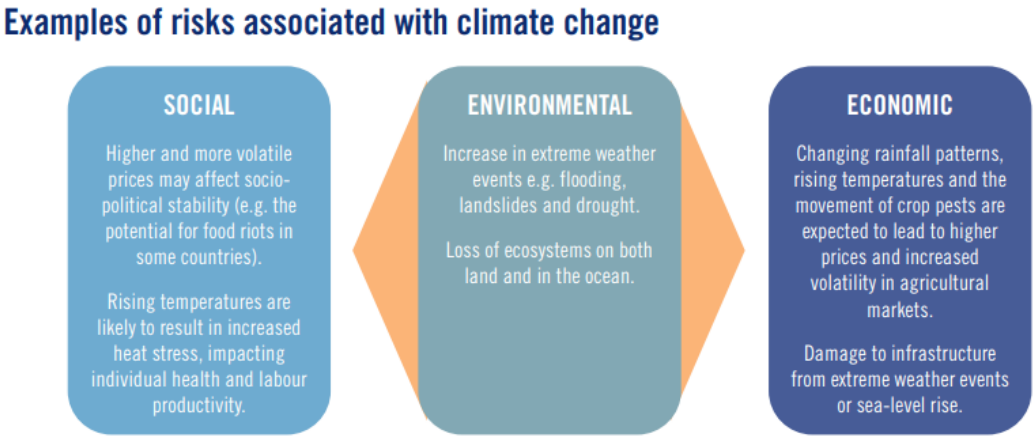


Figure 3.1: Risks associated with climate change [42]

Due to these risks, investors are increasingly looking for more information about climate risks in their portfolios, and want their asset managers to take action. This means that the importance of monitoring and reporting on climate risks is becoming of primary importance. Moreover, it is important that asset owners are aware of the power they possess in the investment chain. "As the powerhouse of long term global investment, they can and do influence the companies in which tey invest and their service providers—such as their investment managers" [43].

A graphic overview of asset owners within the financial system can be seen in figure 3.2.

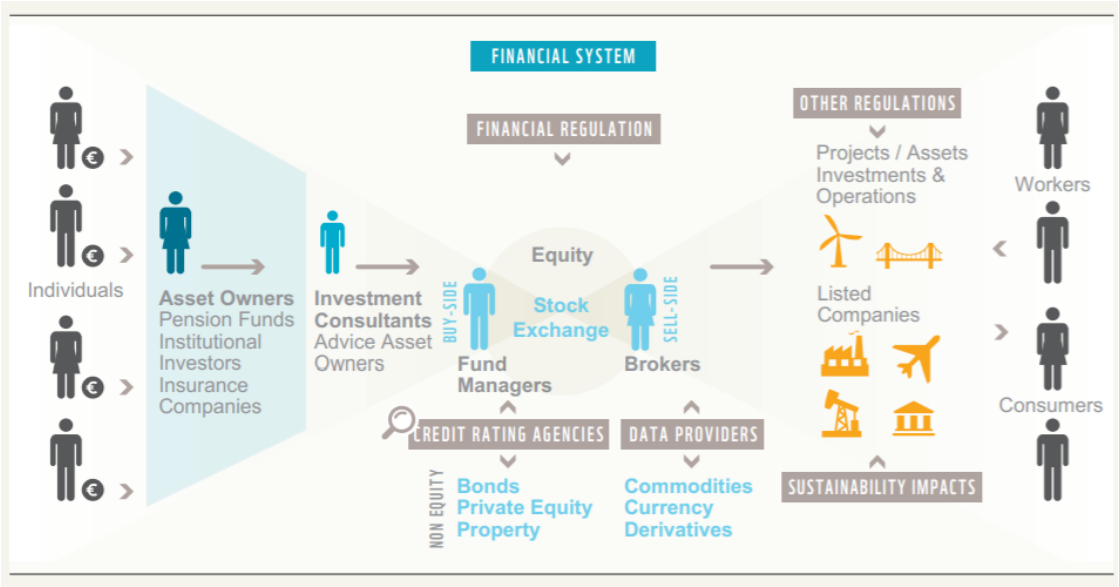


Figure 3.2: Asset owner's position in the financial system [43]

A concrete example of initiatives that are working toward portfolio climate reporting is Science Based Targets initiative (SBTi). SBTi is a colaboration between CDP, the United Nations Global Compact (UNGC), World Resources Institute (WRI) and the World Wide Fund for Nature (WWF). Their goal is to define and promote best practices in science-based target setting for companies. Science-based targets "provide companies with a clearly defined pathway to future-proof growth by specifying how much and how quickly they need to reduce their greenhouse gas emissions"[44]. In other words, they aim to guide the companies on how to get closer to the goals determined in the Paris Agreement, and to give them a score depending on how they are following the specific measures to get to this goals. This score is known as the *Temperature Score*. Temperature—or climate—scores provide qualitative scores to companies on climate issues like carbon footprint, green exposure, etc [43].

The SBTi along with Ortec Finance are working on releasing an open source tool to assess investment portfolios. This tool aims to support investors that seek science-based targets for all portfolio companies by 2050, and who want to assess the long-term emissions goals of the firms, and the portfolio-level emissions [45].

Even though at the moment of writing the main goal of the consortium is to get their open source platform up and running, it is a concern how to make sure that investors can know the temperature score of their portfolio.

This could be easily achieved with the same methodology as the risk reporting case, that is, by calculating the weighted average temperature score of the portfolio, thus, calculating the sum of the individual temperature scores, that would be public knowledge, multiplied by their weight in the portfolio.

3.3 Real Estate Blind Auctions

Another integral part of the financial sector is Real Estate. The housing market in The Netherlands is consistently getting more expensive since the demand—especially in major Dutch cities—is rapidly increasing, with no room for new developments due to lack of planning and building capacity [46]. This increase in prices is also aggravated by investors who can afford to pay higher prices for properties in order to rent them afterwards.

In this setting, real estate auctions have become a popular way of accessing the housing market. In particular blind auctions, where the bidders (potential home buyers) do not know who the other bidders are or how much they are bidding. This has promoted a trend where overbidding on a house has become the norm [47]. In 2019, 40% of the houses were sold above the asking price.

In order to motivate the need for a *private* blind real estate auction it is necessary to understand how they currently work. In The Netherlands, real estate auctions are organized by real estate agencies, and hosted by notaries, or the agencies themselves (which can be problematic). The rules may vary slightly, but bidders make their best offer in writing, by a specific date and time. These offers include, besides price, terms and conditions, settlement dates and finance. The real estate agent negotiates with the prospective buyers to get an offer close to the seller's price [48]. On this type of auction, also known as *expression of interest*, the seller is not bound to accept the best offer. This means that, if they are not satisfied with the offers, they can put the property up for another round of expressions of interest, or just put it back on the market. These blind auctions intend to solve some of the problems inherent to public auctions, namely the collusion between bidders. However, hiding the different bidders from each other only solves part of the problem. Collusion is still possible between seller and real estate agent to drive up the price, or even just on the real estate agent's behalf, in order to get a higher commission for the sale. This might take the form of the agent contacting bidders and giving them a second chance to outbid the current highest bid.

Only the winning bid and the eventual buyer are made public in these auctions. Therefore, there is a lack of transparency and a feeling of uncertainty in the buyers. It is virtually impossible to know if, when given a second opportunity, there is indeed a higher bid or if it is just a bluff in order to increase the purchase price.

Zero-knowledge proofs could be a solution to this problem by helping to create a fair auction model where the bidders are sure that their bid is treated equally to other bids, and the winner is calculated based on some predefined rules. This rules would take different parameters of the bid, like bidding price, financial dimension (whether the bidder has the money or needs a mortgage), timing (closing date) or type of bidder (investor or residential). This different dimensions are given a weight by the seller in order to calculate the "final bid" for each participant.

In order to help protect the interests of bidders and auctioneer and avoid collusion

between different parties, the ideal Real Estate Auction should have all the following properties.

- Every bidder should be able to verify whether they are the winner or not.
- There should be a ranking algorithm for the bids, where the number of outputs of the algorithm was parameterized.
- Privacy of the bidders should always be preserved, unless they are the winners of the auction.
- The bid should not be revealed to any party, even to the seller. The seller knowing the bid could introduced some favoritism.
- Weights are secret but committed upfront. If they change during the process favoritism could be introduced.

This solution could be accomplished through a combination of different cryptographic tools, like Pedersen commitments, Homomorphic Encryption, zero-knowledge proofs and Multi-Party Computation (MPC). There is a great deal of literature on securely implementing sealed bid auctions. However, the applications proposed are usually aimed to either general or internet auctions, a protocol specific to these kinds of blind real estate auctions is yet to be designed. In this protocol, zero-knowledge proofs could be used to prove that the bids are sent in the correct format and to prove that the computation of the comparison protocol—to calculate the ranking of bids—is performed correctly. However, the development of this protocol is out of the scope for this thesis, but it can be left as a future project.

3.4 Dark Pools and Blind Bids

Dark pools are private exchanges for trading securities that are not accessible by the investing public [49]. The reason to be of dark pools are mainly to allow block trading by institutional investors without affecting the market and to avoid front running. The way this is done is by hiding orders sent to the market until there is a match on the dark pool for that order, as opposed to the traditional stock exchange, where the orders show up on the exchange's trading book.

On the other hand, blind bids are another tool to trade high amount of stocks without impacting the market. However, blind bids achieve this by selling a high amount of stocks without revealing what kind of shares make up the book of securities [50]. The only information given is some general characteristics of the book, like the aggregated risk or the different sectors represented in the book of securities.

It is apparent that these two alternatives to the public stock exchange, although might help alleviate the front-running in trading, are really non-transparent methods.

In particular, in the case of dark pools, the opacity of the platforms, owned in many cases by big banks, is not unfunded. In the last decade, with the popularization of dark pools, have come also allegations that "dark pool allegedly promised its customers they would be protected against predatory High Frequency Trading (HFT) while at the same time allowing HFTs access to the pool and customers' order flow" [51]. This was famously documented by the book "*Flash Boys*" by Michael Lewis [52].

These transparency issues could be improved by encrypting the orders on the dark pool with Homomorphic Encryption, but providing zero-knowledge proofs of the encrypted values. Work on deploying a dark pool in the distributed ledger was done by [53].

For the blind bids, a solution similar to portfolio reporting could be given, where the seller of the bucket of securities can give a proof of the aggregated risk of the whole book of securities without revealing the composition of said bucket.

Chapter 4

Zero-Knowledge Portfolio Reporting

4.1 Introduction

As explained in the previous chapter, portfolio reporting within investment funds is an important practice for asset owners who want to monitor the investments made by asset managers.

The exploration of the utility of zero-knowledge proofs in this domain has been previously done. Notorious is the 2005 paper "Risk Assurance for Hedge Funds using Zero Knowledge Proofs" [3]. This paper explains the cryptographic tools and the protocol to be followed in order to allow an investor to verify the portfolio risk characteristics of a fund manager, without requiring the latter one to give information about what specific assets make up the portfolio. However, the state of the art of the technology is, at the time of writing, drastically different than it was at the time of publication of this paper, fifteen years ago.

The other related paper on the topic is a Harvard Thesis written in 2018 titled "Zero Knowledge Proofs and Applications to Financial Regulations"[4]. The thesis defines protocols for applying zero knowledge to three cases in financial regulation, including the investment fund case at hand. The other two cases where use of zero-knowledge proofs is proposed are to let employers verify that employees are not doing inside trading with a blacklisted company, and to verify aggregate information provided by a fund to a collection of investors.

All these cases have in common the need for a trade-off between transparency and confidentiality, as explained in chapter 3. Recapitulating, transparency is important for regulators and investors to know the health of the market and their investments respectively. However, the confidentiality kept by investment managers and brokers incentives participation and innovation. Thus, this conundrum is a constant in the financial sector.

Nowadays, the role of the custodian as a trusted third party is the one closest to

the role of a zero-knowledge proof. Nonetheless, custodians do not usually calculate measures or ratios. Through zero-knowledge proofs, the use case could be extended to other reporting measures such as the Sharpe Ratio, Treynor Ratio, beta, etc. fairly easily once the whole environment is set.

The main contribution on this chapter is to demonstrate that a tool utilizing ZKPs can be built in order to report the aggregated risk measure of a portfolio.

Ideally the algorithm to calculate aggregated risk should look like the following algorithm.

Algorithm 1: Aggregated Risk Calculation

Input: Private: P, W, Public: R, $maxRisk$ integer vectors of size N
for $i \leftarrow 1$ ***to*** n **do**
  $AggRisk = \sum_{i=1}^{N} w_i \times r_i$;
  if $AggRisk < maxRisk$ **then**
    return $True$;
  else
    return $False$;
  end
end

The rest of this chapter has the following structure. Section 4.2 reviews the current technologies and libraries available to implement zero-knowledge proofs. Section 4.3 explains the implementation of bulletproofs in Go for the risk reporting application, and Section 4.4 goes on to explain the same application implemented with zk-SNARKs in JavaScript.

4.2 Available Technologies

At the moment of writing, the state of the art in theoretical zero-knowledge proofs schemes is extensive. However, the available libraries or technology stacks that support these advancements are limited. One of the main challenges is to translate the requirements of the application and the zero knowledge theory into an functional implementation. The following libraries are the main open source resources available for zero-knowledge proofs implementations. In particular, for zk-SNARKs and Bulletproofs (see Chapter 2 for definition of both).

- **Libsnark:** C++ library for zk-SNARKs. It is the most mature the stacks reviewed. The library provides an implementation of proof systems, gadget libraries for constructing R1CS instances and examples of applications [54].
- **ZoKrates:** a high level language that aims to bring zero-knowledge proofs to the Ethereum network. It can be implemented in Solidity contracts.

- **Bellman:** zk-SNARK library for Rust. Used by companies like Z-cash for their zero-knowledge protocols. It aims to solve some security problems from Libsnark and to improve performance [55].
- **Xjsnark:** Java framework for zk-SNARKs. It is a high level framework that aims to help users who are not specialized in cryptography and to automate circuit minimization [56].
- **Circom and Snarkjs:** Circom is a language to write arithmetic circuits that works with snarkjs [57], a JavaScript and Pure Web Assembly library for implementing zk-SNARKs. It also includes support for a MPC set-up ceremony [58].
- **Bulletproofs implementation in Rust:** A Rust implementation of Bulletproofs. It supports single and aggregated range proofs, MPC, and an experimental constraint system API [28].
- **Bulletproofs implementation in Go by ING(ZKRP):** The repository contains implementations for Bulletproofs, Zero Knowldege Range Proofs (ZKRP) and Zero Knowledge Set Membership (ZKSM) [59].

For this work, two libraries were chosen in order to implement zero-knowledge proofs for risk reporting in the portfolio. The first library used was the Bulletproof in Go by ING. It was a comprehensive library, and it already supported the range proof calculation for any interval. The second library used, after realizing the shortcomings of using Bulletproofs on this application, were the Circom and Snarkjs libraries for JavaScript. The choice for these zk-SNARK libraries over other libraries was the fact that it was written in JavaScript, and it could later be integrated in a small application with relative ease as well as the availability of examples in other domains.

4.3 Description of the tool

Before implementing the privacy-preserving risk reporting tool some assumptions and definitions have to be made.

First of all, it is important to set the threat model we are going to be working with. In this case, it is honest but curious verifier. The verifier, or asset owner, will follow the correct verification protocol, but will try to get as much information as possible. On the side of the prover, we will start by defining an honest prover and later challenge that assumption.

The focus of this tool is to preserve the privacy of the contents of the portfolio. The assets and their weights in the portfolio have to be private. However, if the weights are private but the individual risks are not, having the individual risks can reveal the portfolio composition, or at least part of it.

In order to solve that problem, the implementation focuses on Exchange Traded Funds (ETF) instead of having a more general applicability to every type of fund (e.g. hedge funds). ETFs are investment funds that are traded in stock exchange and try to follow the performance for certain index (e.g. S&P500, MSCI World Index, etc).

Thus, for the implementation let us assume that the fund trades assets that belong to an index. Therefore, the calculation of the aggregated risk will be done over the stocks in that index (500 companies if we take S&P500), and only the weights of those stocks that are actually on the portfolio will be non zero.

4.4 Bulletproofs in Go

After reviewing the different implementations of the most recent developments in zero-knowledge proofs, the Bulletproofs implementation of ING called Zero-Knowledge Range Proofs (ZKRP) seemed to be the most suitable start point to begin prototyping. Not only was it comprehensive, and followed the steps of the bulletproof paper, but it was a standalone library as opposed to the implementation in Rust.

"Overall, Go is a simple, yet powerful, language and which produces robust, fast and powerful code. If you want any kind of library, it just connects to GitHub, and downloads the code in a simple to use form [60]." In order to create the prototype, an excel file was created with the top ten assets in the MSCI World Index, as well as their volatility at the time, and dummy weights for each one of the assets.

This data was read by the program and, from the weights and individual risk measures, the aggregated risk was calculated according to the formula in algorithm 1.

In the test prototype, $N = 10$ (from Algorithm 1), corresponding to the top ten assets listed in the MSCI World Index. However, ideally, the program would loop over all the assets in the index, and calculate the aggregated risk taking into account all the assets—some of them with null weight—in order to hide the size of the portfolio.

Once the manager has calculated the aggregated risk, they can create the proof that states that the overall portfolio risk is within a range agreed upon contract and they send it to the asset owner through a secure channel (i.e. TLS) and the asset owner can verify the proof sent by the asset manager, and confirm that the contract conditions are being met.

In Listing 4.1 a snippet of the code in Go used for the application is shown.

Listing 4.1: script for the implementation with Bulletproofs

```
func RangeProver(path string) (bulletproofs.ProofBPRP, error)
    {
        proof := bulletproofs.ProofBPRP{}

        assets := ReadAssetsFromExcel("weights")
        weights := ReadWeights("weights")
        risk := ReadVolatility("weights")

        //Calculate aggregated risk
        AggregatedRisk, errorCal := CalculateAggregatedRisk(
           assets, weights, risk)
        if errorCal != nil {
                fmt.Println(errorCal.Error())
                return proof, errorCal
        }

        //Create the proof
        proof, errorCal = ProveRiskWithinRange(200000,
           600000, AggregatedRisk)
        if errorCal != nil {
                fmt.Println(errorCal.Error())
                return proof, errorCal
        }

        return proof, nil
}
```

4.5 zk-SNARKS in JavaScript

The second implementation of the example was carried out with two JavaScript libraries developed by the consortium Iden3 [57], [58]. The decision to try another implementation after trying the Bulletproof library implemented by ING was two-folded. The first reason was the incapability of proving that the calculation of the aggregated risk had been done right, which led to a shift of perspective and to stop trying to hide the aggregated risk, and try to prove its right calculation instead. The second reason was the lack of community adoption and documentation on the ING stack. The decision to try the Iden3 libraries came about because of the existence of examples and the accessibility of the language, which is one of the most used computer languages, JavaScript. Moreover, JavaScript allows for a straightforward implementation of an application for the tool.

The two libraries used in this process were Circom and Snarkjs. Circom, as defined by its developers, is a language designed to write arithmetic circuits that can be used in zero-knowlede proofs [57]. Even though the goal of the new implementation was to prove the correctness of the calculations, instead of the belonging of a number to an interval, it soon became clear that the zk-SNARKS were powerful and could act as range proofs.

4.5.1 Circuit Definition

zkSNARKs are zero-knowledge proofs of an arithmetic circuit. Thus, the first step is to define the circuit. This circuit is comprised of certain inputs—some of which are private—, a number of operations, and an output. In the circuit some constraints are defined.

The implementation was done with Circom, as mentioned above. This language compiles the circuit to R1CS files that can be transformed to human readable (json) format.

zkSNARKS need a different trusted setup for every circuit. Therefore, this could be considered a crucial step in the implementation of a zkSNARK. It is essential to define this circuit right and to test it and think of all the implications of the circuit before setting up the proof. The reason of the importance of this step is that the trusted setup is an expensive and fundamental part of a successful zkSNARK. If the circuit is buggy or there needs to be a change in the code of the circuit, the ceremony needs to be repeated, and it is not desirable once in production.

4.5.2 Proof implementation

After the definition of the circuit, it has to be transformed into the format accepted by the JavaScript Snarkjs library. Once that is done with a functionality of the

Listing 4.2: Circom circuit for aggregated risk calculation

```
include "node_modules/circomlib/circuits/comparators.circom";
template Summation(n){
        signal private input weight[n]; //Each weight
        signal input risk[n];   //Individual risks
        signal input minRisk;   //int
        signal input maxRisk;   //int
    //Output
        signal output out; //1 or 0

        //intermediary variables
        signal sum;
        signal intermediary;

        //Constraint : The aggregated risk must be the
           weighted sum of the risks
        for (var i=0; i < n; i++) {
                intermediary <-- intermediary + weight[i] *
                   risk[i];
        }
        sum <== intermediary;

        //Constraint: the aggregated risk within the range
           minRisk <= AggregatedRisk <= maxRisk.
        //Max num bits is 20. 2^20 > 1000000. The magnitude
           is this big because numbers have to be integers.

        component lt1 = LessEqThan(20);
        lt1.in[0] <== sum;
        lt1.in[1] <== maxRisk;
        lt1.out === 1;

        component gt1 = GreaterEqThan(20);
        gt1.in[0] <== sum;
        gt1.in[1] <== minRisk;
        gt1.out === 1;

        out <-- (lt1.out * gt1.out)
        out === 1;
}

component main = Summation(10);
```

Circom library, the next steps can be implemented.

- *Set-up Phase*: This is the critical phase. In a production setting a ceremony like the one for Zcash [25] should be put in place in order to guarantee the public parameters are created and no toxic waste remains. It is important to note that the set-up is inherent to the circuit we want to create the proof for. It is important to note that, in our model where the fund follows an index, this sets a limit to the number of assets that can be in the portfolio (in our case it was set to 10 assets). If the number of maximum assets were to be increased, a new set-up should be put in place.
- *Proving Phase*: The prover, or asset manager, gives as input the weight and corresponding individual risk for each asset in the portfolio. In the prototype an excel sheet was used to pass the input. In a production setting, this input would come from a database—ideally a distributed data base. This is the input for the circuit and, as output the prover creates a (zero-knowledge) proof—with a proving key created during the set-up phase—of having calculated the aggregated risk right and to state that said risk is within the range specified by the asset owner.
- *Verifying Phase*: The asset owner receives the proof and checks that the proof is valid with the validation key—created in the set-up phase.

In order to illustrate the steps, a small application was created using Angular and Electron. Some pictures of the prototype can be seen in Figure 4.1 and in the appendix.

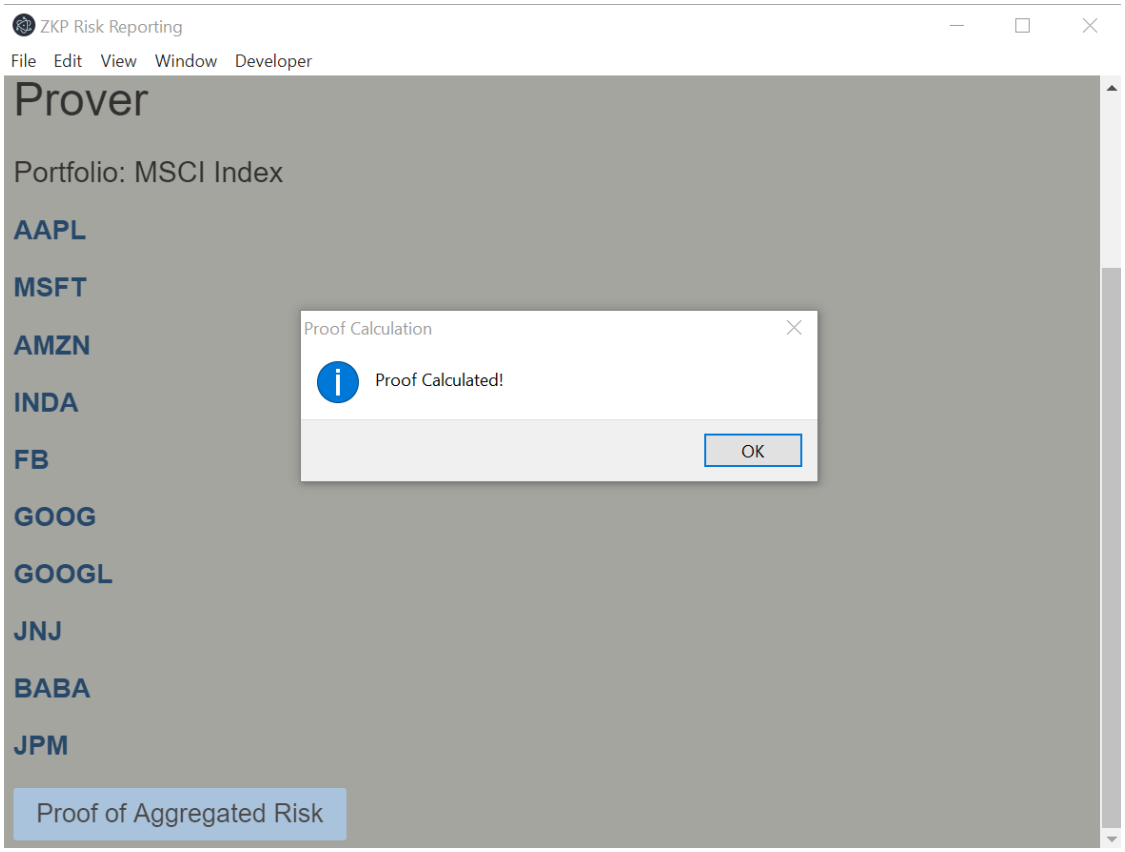


Figure 4.1: Snapshot of application using the snarkjs ZKP risk reporting tool

Chapter 5

Discussion

5.1 Introduction

In the previous chapter the feasibility of the use of zero-knowledge proofs for risk reporting has been demonstrated through the implementation of two different solutions. This chapter discusses the implications of the implementations explained in Chapter 4. The first section, Section 5.2, discusses the assumptions and limitations of each of the different technologies. Section 5.3 compares the use of Bulletproofs and zk-SNARKs for this specific application. Lastly, Section 5.4 discusses the limitations of zero-knowledge proofs in general.

5.2 Assumptions and Limitations

5.2.1 Bulletproofs

The implementation of Bulletproofs in Go provides a range proof that demonstrates that a number (aggregated risk measure) is within a range. It results in a rather simple prototype that relays on several strong security assumptions—the stronger the security assumptions, the weaker the model. However, it is a start point to understand what properties are essential for such an application to be successful in production.

- **Honest Prover:** with this model, the only way the verifier can be certain that the calculation of the aggregated risk was done right is if we work with an honest prover model. This is a strong security assumption. In this case, the verifier has to trust that the prover is doing everything as it is supposed to. However, historically, it has been demonstrated that hedge funds have not always been trustworthy. Therefore, for a use case like this one, the honest

prover would be too much of a stretch. This is one of the potential limitations listed in the Hardvard thesis [4]. As a solution to this, the paper proposes the introduction of a TTP that gets commitments of the information sent by the prover, so it can be a tiebreaker in case of conflict. However, ideally, the need for a TTP should be eliminated.

- **Technology stack:** Even though the ING implementation of Bulletproofs follows the mathematics in the homonym paper, this is not the most popular implementation and might not have enough peer reviews to consider it a safe implementation. For a more mature implementation it would be interesting to consider other stacks like the Bulletproof implementation in Rust [28].

5.2.2 zkSNARKS

The second implementation described in the previous chapter uses zk-SNARKs by creating an arithmetic circuit and proving that the calculation of the weighted average risk was done correctly and that the risk is within certain range. As for Bulletproofs, the zk-SNARKs implementation does not solve all the concerns. The following are some considerations that have to be made when deploying the zk-SNARKs with the Iden3 JavaScript library.

- **Trusted Set-Up:** zk-SNARKs are, at the time of writing, the most powerful family of zero-knowledge proofs. The proof size is relatively small and scales well and the verifier time is low, giving the entire computing burden to the prover. However, in order to achieve this, a different setup is needed for each circuit. This entails a security hazard, since there needs to be a secure setup (or trusted ceremony) where the private parameters that are used to create the public parameters are destroyed, or otherwise, the proofs would have no validity, since they could be tampered with. Hence, while building the prototype, no special measures are going to be taken when creating the public parameters. However, if the prototype were to be scaled into a production model and zkSNARKS are chosen as the preferred technology, an elaborate trusted setup would have to be put in place. As explained in the previous chapter, the library chosen for the implementation of zkSNARKS supports the creation of this *multi-party ceremony*.

- **Honest Prover:** Once again, like with Bulletproofs, one has to trust that the origin of the data is truthful. This means that a part of the "honest verifier" assumption stays in the model. Even though now it is possible to know that the calculation of the aggregated risk was done right without knowing the singular weights of each asset, it is still not possible to prove that the data used to create the proof is indeed the one corresponding to the real portfolio. Since we have seen this is a common problem not fully solved in any of the solutions tested, more about this will be said in the next section.We are, once again, faced with the need of a platform (i.e. a distributed ledger) or a

trusted third party who gets commitments on the data and can open those commitments in a time-lapsed manner in order to verify that the prover is being honest. Another choice might be to install cryptographically protected and tamper-proof hardware that has access to the portfolios of the asset manager, and that periodically creates and sends the proofs.

- **Scalability:** The current implementation is based in a circuit that takes 10 assets and calculates the aggregated risk for a portfolio composed of some of those 10 assets. A normal portfolio is more complex and is made up of significantly more than 10 assets. Maybe a universal zero-knowledge proof scheme or a transparent one would mean a more flexible and scalable circuit.

5.3 Technology Assessment

A glimpse of the differences between bulletproofs and zk-SNARKs has been given in the introduction and on Chapter 4. However, this section will summarize the main differences found for the application at hand.

It was noted during the process of implementing the solution with Bulletproofs that the technology provides a way of proving that the value of the aggregated risk is within a range without revealing said value. However, unlike zk-SNARKs, it does not prove that the aggregated risk calculation was done correctly. Therefore, Bulletproofs provide a weaker solution to the problem, where still a trusted third party might have to intervene and stronger security models are needed.

It is also important to discuss the effect of the "trusted set-up" needed in zk-SNARKs. While the trusted set-up allows for the computational burden to be shifted to the prover, making it lightweight for the verifier, it can have severe unwanted effects if the set-up is not done in such away that the toxic waste is discarded.

As explained in Chapter 2, there are trade-offs between ZKP solutions and there is no perfect solution that can achieve a small size proof, low communication complexity and a transparent setting. However, newer constructions (i.e. zk-SNORKs) offer a universal set-up common to every circuit, achieving sort of a middle ground when thinking of proof size and security assumptions.

5.4 Considerations of Zero-Knowledge Proofs

This section will talk about the assumptions and considerations taken when developing the implementation. Issues that are not solved can be included in future lines of research.

- **Honest Prover:** In both the implementations reviewed, we are faced with the need of a platform (i.e. a distributed ledger) or a trusted third party that gets commitments on the data and can open those commitments in a time-lagged manner in order to verify that the prover is being honest. Another choice might be to install cryptographically protected and tamper-proof hardware that has access to the portfolios of the asset manager, and that periodically creates and sends the proofs.

  An alternative to a trusted third party as a solution to the previous issue can be the use of a business ledger that both parties can access with different roles, and where the assets that make up the portfolio and all the needed information is kept by the prover in an encrypted manner. For this solution, the data (i.e. assets) should be native to the distributed ledger or some sort of tokenization (i.e. trusted transformation from physical asset o digital asset) should be performed to represent the data digitally. Using Distributed Ledger Technology, the prover computes the proof, the verifier, without learning anything from the data, can know that the calculations were done correctly with the data stored in the distributed ledger. This solution, even though it is not applicable right away due to the slow adoption of distributed ledgers in financial companies, it is a feasible solution given the state of the art of the technology. However slow, a trend toward adoption of the technology is still observable, and solutions like Symbiont.io or the Dusk Network are already working on and could offer the right environment where this application of zero-knowledge proofs could live.

- **Honest Verifier:** As in the case for honest prover, the implementations assume that the verifier is also honest (but curious).

- **Transmission Channel:** The channel must preserve the privacy and reliability of the data sent over. There have to be explicit assumptions on whether the adversary can tap all communication channels or not, whether the channel is reliable and authenticity guaranteed. Moreover, there must be a choice between a synchronous or asynchronous channel.

- **Formatting the values:** ZKPs are mostly built with arithmetic circuits and only support algorithms which are integers. In order to work with rational numbers, like was the case of this implementation, the values passed are input have to be scaled. This has to be taken into account when implementing the tool and the values have to be pre-processed before being passed as input for the arithmetic circuit.

- **Linear calculations:** None of the two implementations presented in this thesis support proving zero-knowledge non-linear algorithms. Therefore, no infinite (e.g. while) loops are supported and for loops need to have the number of iterations hard coded in the circuit. Our example works because calculates the weighted average risk (volatility). However, more advanced risk measures might not be supported.

- **Trust in the technology:** As stated in [61], it is important to consider that asset managers and asset owners would have to trust the organization

implementing the system and that the zero-knowledge proofs indeed work and there is no backdoor where private information is being sent. Thus, the reputation of the party who implements the system is, in reality, an important determinant in the success or failure of this platform.

Chapter 6

Conclusion

Throughout the duration of this thesis, the question trying to be answered has been: *"What use cases in Finance can benefit from zero-knowledge proofs and is it feasible to implement them with the state of the art technology?"*. In order to answer it, the first step was to do a thorough literature review, and afterwards try to find different scenarios in finance where there is opportunity for successfully applying zero-knowledge proofs. Three applications are proposed in this work. The first one, original to [3], is portfolio reporting. The second and third applications proposed are a contribution from this work, namely blind real estate auctions and dark pools and blind bids in stock exchange.

To answer the second part of the question, whether it is feasible to apply zero-knowledge proofs to technology, the first use case was selected (i.e. portfolio reporting, specifically risk reporting). Two prototypes were implemented. One with Bulletproofs in Go and a second one with zk-SNARKs in JavaScript along with an Electron and Angular application (for illustrative purposes). After the implementations, we gave a review of the assumptions made for the implementation to work, and of its limitations thus far.

Overall, the technology is improving and gaining maturity in the context of the Distributed Ledger Technology. As we can see in the 2019 Gartner Hype Cycle for Blockchain Business [62], blockchain, in general, is currently going through the *Through of Disillusionment*, and therefore it still has about 2 to 5 years to reach its maturity. Therefore, in order to make a reliable *private* risk reporting tool, there needs to be some progress in the adoption of the Distributed Ledger Technology in the financial sector.

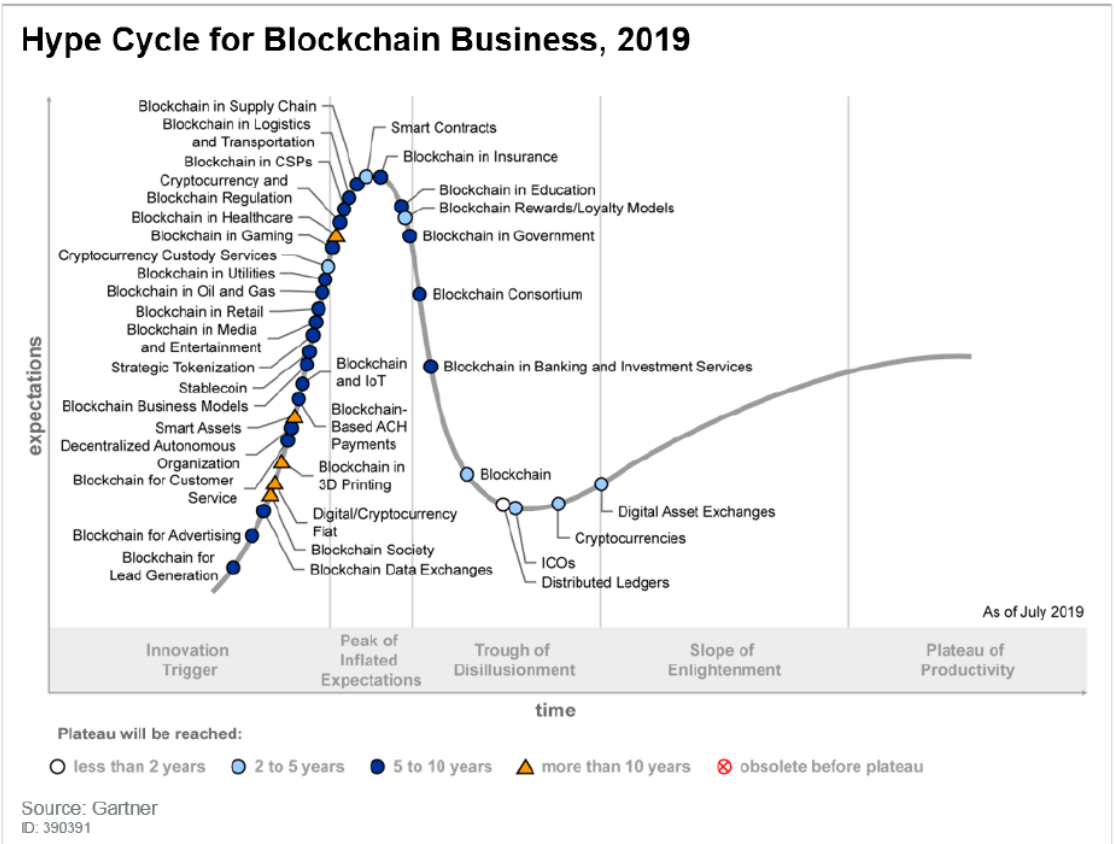


Figure 6.1: Gartner Hype Cycle for Blockchain [62]

6.1 Future Work

The topic of this thesis is one that can still be further researched, since the technology is still in its early stages of adoption and there are some promising developments of new zero-knowledge proof techniques. Moreover, new applications can come out of this work on the financial domain when generalizing the problem to the confidentiality-transparency trade-off. Some future work of research and development can include:

- Developing further the proposed proof of concept to get a working prototype.
- Developing a protocol for a secure and private real estate auction using different cryptographic tools, among which are ZKP, as the one proposed in chapter 3.
- Researching the implementation of new applications in the Finance Domain, like dark pools or the Paris Agreement reporting tool.
- Researching how to solve some of the limitations found in the current applications, some of which are generalized to all zero-knowledge proofs and, when solved, it would mean the possibility of using these technologies in production.

- Looking for partnerships with companies like symbiont.io or the Dusk Network to build a decentralized financial market infrastructure where the zero-knowledge portfolio reporting tool can run without the *honest prover* assumption.

Glossary

API A set of routines, protocols, and tools for building software applications. 27

Distributed Ledger Technology An asset database that can be shared across a network of multiple sites, geographies or institutions. All participants within a network can have their own identical copy of the ledger. Any changes to the ledger are eventually reflected in all copies [63]. 38, 41, 47

Homomorphic Encryption Encryption schemes that have the property of having equivalent operations between values in the plaintext domain and in the encrypted domain. HE makes it possible to analyze and manipulate encrypted data without revealing the underlying data. 10, 23, 24

NIZK Zero-Knowledge proof that does not require communication between prover and verifier during the proving process. 47

PPT Refers to a uniform algorithm, with a fixed program size independent of a security parameter n. 6, 7, 47

Succinct Non-Interactive Adaptive Argument of Knowledge See zkSNARK. 9, 47

TTP An entity that facilitates interactions between two parties who do not trust each other but both trust the third party; the Third Party reviews all critical transaction communications between the parties to avoid fraud [64] . iii, 36, 47

Turing Machine A mathematical model of computation that defines an abstract machine, which manipulates symbols on a strip of tape according to a table of rules [65]. 6, 47

ZKP Cryptographic tool that allows a party to prove knowledge of a secret to another party without revealing said secret. iii, iv, ix, 1, 9, 12, 26, 33, 37, 38, 42, 47

zk-SNARK Proof construction where one can prove possession of certain information, e.g. a secret key, without revealing that information, and without any interaction between the prover and verifier [25]. They are characterized by being small and fast-to-verify non-interactive proofs. iv, 9–12, 14, 26, 27, 35–37, 41, 47

zk-SNORK Zero-knowledge proofs with universal set-up that is common to all arithmetic circuits proven. 37, 47

zk-STARK Zero-knowledge proofs with transparent set-up, which means that all the randomness in the set-up is public (no toxic waste). 9, 13, 47

CSR Type of private business self-regulation that aims to contribute to societal goals of a philanthropic, activist, or charitable nature by engaging in or supporting volunteering or ethically-oriented practices [66]. 47

HFT A method of trading that uses powerful computer programs to transact a large number of orders in fractions of a second in order to frontrun other investors [67] . 24, 47

MPC A cryptographic protocol that distributes a computation across multiple parties where no individual party can see the other parties' data [68]. 27, 47

QAP Form used to group R1CS statements. Instead of usign dot products, it uses polynomials. 47

R1CS A list of three vectors $\overline{a_i}$,$\overline{b_i}$,$\overline{c_i}$ and a vector $\overline{s}$ that is a solution to the equation:

$$\langle \overline{a_i}, \overline{s} \rangle * \langle \overline{b_i}, \overline{s} \rangle - \langle \overline{c_i}, \overline{s} \rangle = 0 \ \forall i \tag{6.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product of two vectors [69]. R1CS are used on zkSNARKs . 10, 47

SBTi A collaboration between CDP, the United Nations Global Compact (UNGC), World Resources Institute (WRI), and the World Wide Fund for Nature (WWF) and one of the We Mean Business Coalition commitments. It champions science-based target setting as a powerful way of boosting companies' competitive advantage in the transition to the low-carbon economy [44]. 21, 47

Acronyms

CSR Corporate Social Responsibility. 18, 46

DLT Distributed Ledger Technology. iii, 9

HFT High Frequency Trading. 24, 46

MPC Multi-Party Computation. 23, 27, 46

NIZK Non-Interactive Zero-Knowledge. 9

PPT Probabilistic Polynomial Time. 6

QAP Quadratic Arithmetic Program. 10, 46

R1CS Rank 1 Constraint System. 10, 46

SBTi Science Based Targets initiative. 21, 46

SNARK Succinct Non-Interactive Adaptive Argument of Knowledge. 9

TM Turing Machine. 6

TTP Trusted Third Party. iii, 36

ZKP Zero-Knowledge Proof. iii, 1

zkSNARK Zero-Knowledge Succinct Non-Interactive Adaptive Argument of Knowledge. 9

zkSNORK Zero-Knowledge Succinct Non-interactive Oecumenical aRguments of Knowledge. 9

zkSTARK Zero-Knowledge Scalable Transparent ARguments of Knowledge. 9

Bibliography

[1] S. Goldwasser, S. Micali, and C. Rackoff, "Knowledge Complexity of Interactive Proof-Systems.", *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 291–304, 1985, ISSN: 07349025. DOI: 10.1145/3335741.3335750 (cit. on pp. iii, 6, 9).

[2] G. A. Akerlof, "The market for "lemons": Quality uncertainty and the market mechanism", *The Quarterly Journal of Economics*, vol. 84, no. 3, pp. 488–500, 1970, Publisher: Oxford University Press, ISSN: 0033-5533. DOI: 10.2307/1879431. [Online]. Available: https://www.jstor.org/stable/1879431 (visited on 07/21/2020) (cit. on pp. iii, 17).

[3] M. Szydlo, "Risk assurance for hedge funds using zero knowledge proofs", *Lecture Notes in Computer Science*, vol. 3570, pp. 156–171, 2005, ISSN: 03029743. DOI: 10.1007/11507840_16 (cit. on pp. iii, iv, 18, 19, 25, 41).

[4] N. R. Gowravaram, "Zero Knowledge Proofs and Applications to Financial Regulation", PhD thesis, Harvard College, 2018. [Online]. Available: http://nrs.harvard.edu/urn-3:HUL.InstRepos:38811528%7B%5C%%7D09 (cit. on pp. iii, iv, 2, 18, 19, 25, 36).

[5] O. Finance. (). About us. Library Catalog: www.ortecfinance.com, [Online]. Available: https://www.ortecfinance.com/en/about-us (visited on 07/13/2020) (cit. on p. 2).

[6] W. Kenton. (). Risk management in finance, Investopedia. Library Catalog: www.investopedia.com, [Online]. Available: https://www.investopedia.com/terms/r/riskmanagement.asp (visited on 07/13/2020) (cit. on p. 3).

[7] 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 *Conference on the Theory and Application of Cryptology*, Springer, 1989, pp. 628–631 (cit. on p. 5).

[8] *Zero-knowledge proof*, in *Wikipedia*, Page Version ID: 967721702, Jul. 14, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Zero-knowledge_proof&oldid=967721702 (visited on 07/21/2020) (cit. on p. 6).

[9] I. R. Pass, "Lecture 18 : Zero-Knowledge Proofs The formal definition Graph isomorphism", *ReCALL*, pp. 1–5, 2009 (cit. on p. 6).

[10] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems", *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994, ISSN: 09332790. DOI: 10.1007/BF00195207 (cit. on p. 6).

[11] I. H. Corrigan-gibbs, S. Kim, and D. J. Wu, "Lecture 1: interactive proofs and zero-knowledge proofs", pp. 9–10, 2018 (cit. on p. 7).

[12] N. P. Smart, "Cryptography Made Simple Information Security and Cryptography", pp. 197–223, 2016. DOI: 10.1007/978-3-319-21936-3 (cit. on p. 7).

[13] Yevgeniy Dodis, "Lecture 14. Commitment Schemes", *NYU*, no. 1, pp. 1–14, 2008. [Online]. Available: https://cs.nyu.edu/courses/fall08/G22.3210-001/lect/lecture14.pdf (cit. on p. 7).

[14] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications", in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC '88, Chicago, Illinois, USA: Association for Computing Machinery, 1988, pp. 103–112, ISBN: 0897912640. DOI: 10.1145/62212.62222. [Online]. Available: https://doi.org/10.1145/62212.62222 (cit. on p. 9).

[15] J. Kilian, "A note on efficient zero-knowledge proofs and arguments (extended abstract)", in *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, ser. STOC '92, Victoria, British Columbia, Canada: Association for Computing Machinery, 1992, pp. 723–732, ISBN: 0897915119. DOI: 10.1145/129712.129782. [Online]. Available: https://doi.org/10.1145/129712.129782 (cit. on p. 9).

[16] J. Groth, "Simulation-sound NIZK proofs for a practical language and constant size group signatures", in *Advances in Cryptology – ASIACRYPT 2006*, X. Lai and K. Chen, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 444–459, ISBN: 978-3-540-49476-8 (cit. on p. 9).

[17] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again", *ITCS 2012 - Innovations in Theoretical Computer Science Conference*, pp. 326–349, 2012. DOI: 10.1145/2090236.2090263 (cit. on p. 9).

[18] J. Groth, "On the size of pairing-based non-interactive arguments", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9666, pp. 305–326, 2016, ISSN: 16113349. DOI: 10.1007/978-3-662-49896-5_11 (cit. on p. 9).

[19] B. Bunz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short Proofs for Confidential Transactions and More", *Proceedings - IEEE Symposium on Security and Privacy*, vol. 2018-May, pp. 315–334, 2018, ISSN: 10816011. DOI: 10.1109/SP.2018.00020 (cit. on pp. 9, 11).

[20] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs", Advances in Cryptology

– EUROCRYPT 2019, vol. 11476, pp. 103–128, 2019. DOI: 10.1007/978-3-030-17653-2_4 (cit. on p. 9).

[21] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn, *Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference strings*, Cryptology ePrint Archive, Report 2019/099, https://eprint.iacr.org/2019/099, 2019 (cit. on p. 9).

[22] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, *Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge*, Cryptology ePrint Archive, Report 2019/953, https://eprint.iacr.org/2019/953, 2019 (cit. on p. 9).

[23] T. Xie, J. Zhang, Y. Zhang, C. Papamanthou, and D. Song, "Libra: Succinct Zero-Knowledge Proofs with Optimal Prover Computation", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11694 LNCS, pp. 733–764, 2019, ISSN: 16113349. DOI: 10.1007/978-3-030-26954-8_24 (cit. on p. 9).

[24] J. Zhang and T. Xie, "Virgo: Zero knowledge proofs system without trusted setup", p. 13, 2019 (cit. on p. 9).

[25] (). What are zk-SNARKs?, Zcash. Library Catalog: z.cash, [Online]. Available: https://z.cash/technology/zksnarks/ (visited on 07/16/2020) (cit. on pp. 10, 15, 32, 45).

[26] V. Buterin. (Dec. 13, 2016). Quadratic arithmetic programs: From zero to hero, Medium. Library Catalog: medium.com, [Online]. Available: https://medium.com/@VitalikButerin/quadratic-arithmetic-programs-from-zero-to-hero-f6d558cea649 (visited on 07/28/2020) (cit. on p. 10).

[27] D. Benarroch. (Feb. 14, 2019). Diving into the SNARKs setup phase, Medium. Library Catalog: medium.com, [Online]. Available: https://medium.com/qed-it/diving-into-the-snarks-setup-phase-b7660242a0d7 (visited on 07/28/2020) (cit. on p. 11).

[28] *Dalek-cryptography/bulletproofs*, original-date: 2018-02-02T01:35:49Z, Jul. 27, 2020. [Online]. Available: https://github.com/dalek-cryptography/bulletproofs (visited on 07/28/2020) (cit. on pp. 11, 27, 36).

[29] V. Buterin. (). Understanding PLONK, [Online]. Available: https://vitalik.ca/general/2019/09/22/plonk.html (visited on 07/28/2020) (cit. on p. 12).

[30] J. Groth, "Non-interactive zero-knowledge arguments for voting", in *Applied Cryptography and Network Security*, J. Ioannidis, A. Keromytis, and M. Yung, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 467–482, ISBN: 978-3-540-31542-1 (cit. on p. 13).

[31] S. Philippe, R. J. Goldston, A. Glaser, and F. D'Errico, "A physical zero-knowledge object-comparison system for nuclear warhead verification", *Nature*, pp. 1–7, 2016. DOI: 10.1038/ncomms12890 (cit. on p. 14).

[32] (). The monero project, getmonero.org, The Monero Project. Library Catalog: web.getmonero.org, [Online]. Available: https://getmonero.org/index.html (visited on 07/16/2020) (cit. on p. 14).

[33] (). Grin, [Online]. Available: https://grin.mw/ (visited on 07/16/2020) (cit. on p. 14).

[34] (Mar. 23, 2017). ZSL: Zk-SNARKs for the enterprise, Electric Coin Company. Library Catalog: electriccoin.co, [Online]. Available: https://electriccoin.co/blog/zsl/ (visited on 07/21/2020) (cit. on p. 14).

[35] P. DeFi. (Feb. 3, 2020). ZK-rollup: Scaling ethereum for the long-term, Medium. Library Catalog: medium.com, [Online]. Available: https://medium.com/plutusdefi/zk-rollup-scaling-ethereum-for-the-long-term-287aa95e3ba9 (visited on 07/16/2020) (cit. on p. 15).

[36] P. B. B. OBE. (Apr. 26, 2020). Goodbye to WPA-2 and hello to WPA-3, Medium. Library Catalog: medium.com, [Online]. Available: https://medium.com/asecuritysite-when-bob-met-alice/hello-to-wpa-3-ae8b9c365b95 (visited on 08/07/2020) (cit. on p. 15).

[37] A. Bloomenthal. (). An uneven playing field: Asymmetric information, Investopedia, [Online]. Available: https://www.investopedia.com/terms/a/asymmetricinformation.asp (visited on 07/06/2020) (cit. on p. 17).

[38] "What is information asymmetry?", *The Economist*, ISSN: 0013-0613. [Online]. Available: https://www.economist.com/the-economist-explains/2016/09/04/what-is-information-asymmetry (visited on 07/21/2020) (cit. on p. 17).

[39] M. Spense, "Job market signaling", *The Quarterly Journal of Economics*, vol. 87, no. 3, pp. 355–374, 1973 (cit. on p. 17).

[40] J. Kuepper. (). Volatility, Investopedia. Library Catalog: www.investopedia.com, [Online]. Available: https://www.investopedia.com/terms/v/volatility.asp (visited on 07/22/2020) (cit. on p. 18).

[41] Anonymous. (Nov. 23, 2016). Paris agreement, Climate Action - European Commission. Library Catalog: ec.europa.eu, [Online]. Available: https://ec.europa.eu/clima/policies/international/negotiations/paris_en (visited on 07/20/2020) (cit. on p. 20).

[42] PRI and IIGCC, "A guide on climate change for private equity", 2016 (cit. on p. 20).

[43] S. Godinot and J. Vandermosten, "WWF Climate Guide To Asset Owners: Aligning Investment Portfolios With the Paris Agreement", p. 72, 2017. [Online]. Available: https://www.wwf.org.uk/sites/default/files/publications/Dec17/WWF%20Climate%20Guide%20to%20Asset%20Owners%20Full%20version%20Dec17%7B%5C_%7D0.pdf (cit. on pp. 20, 21).

[44] (). What is a science-based target? | science based targets. Library Catalog: sciencebasedtargets.org, [Online]. Available: https://sciencebasedtargets.org/what-is-a-science-based-target/ (visited on 07/23/2020) (cit. on pp. 21, 46).

[45] (). OFBDABV/SBTi: This toolkit helps companies and financial institutions to assess the temperature alignment of current targets, commitments, and investment and lending portfolios, and to use this information to develop targets for official validation by the SBTi.', [Online]. Available: https://github.com/OFBDABV/SBTi/ (visited on 08/06/2020) (cit. on p. 21).

[46] R. Nijskens and M. Lohuis, "The Housing Market in Major Dutch Cities", *Hot Property*, vol. 15, pp. 23–35, 2019. DOI: 10.1007/978-3-030-11674-3_3 (cit. on p. 22).

[47] (). Housing market in the Netherlands in 2020 | Should you buy a house?, Hanno. Library Catalog: www.hanno.nl, [Online]. Available: https://www.hanno.nl/expat-mortgages/housing-market-in-the-netherlands/ (visited on 07/20/2020) (cit. on p. 22).

[48] (Nov. 8, 2018). Property buyers warned about being pressured into 'buying blind', Australian Financial Review. Library Catalog: www.afr.com Section: wealth, [Online]. Available: https://www.afr.com/wealth/property-buyers-warned-about-being-pressured-into-buying-blind-20181106-h17kuj (visited on 07/27/2020) (cit. on p. 22).

[49] E. Picardo. (). An introduction to dark pools, Investopedia. Library Catalog: www.investopedia.com, [Online]. Available: https://www.investopedia.com/articles/markets/050614/introduction-dark-pools.asp (visited on 07/27/2020) (cit. on p. 23).

[50] J. Chen. (). Blind bid definition, Investopedia. Library Catalog: www.investopedia.com, [Online]. Available: https://www.investopedia.com/terms/b/blind-bid.asp (visited on 07/27/2020) (cit. on p. 23).

[51] J. D. Jr. (Jan. 3, 2020). FLASH FRIDAY: Turning the spotlight on dark pools, Traders Magazine. Library Catalog: www.tradersmagazine.com Section: Flash Friday, [Online]. Available: https://www.tradersmagazine.com/flashback/flash-friday-turning-the-spotlight-on-dark-pools/ (visited on 07/27/2020) (cit. on p. 24).

[52] M. Lewis, *Flash boys: a Wall Street revolt*. WW Norton & Company, 2014 (cit. on p. 24).

[53] B. França, "Deep Ocean: A blockchain-agnostic dark pool protocol", pp. 1–4, 2019. arXiv: 1910.02359. [Online]. Available: http://arxiv.org/abs/1910.02359++ (cit. on p. 24).

[54] *Scipr-lab/libsnark*, original-date: 2014-06-02T20:27:34Z, Jul. 24, 2020. [Online]. Available: https://github.com/scipr-lab/libsnark (visited on 07/27/2020) (cit. on p. 26).

[55] (Apr. 4, 2017). Bellman: Zk-SNARKs in rust, Electric Coin Company. Library Catalog: electriccoin.co, [Online]. Available: https://electriccoin.co/blog/bellman-zksnarks-in-rust/ (visited on 08/01/2020) (cit. on p. 27).

[56] A. Kosba, C. Papamanthou, and E. Shi, "xJsnark: A framework for efficient verifiable computation", in *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA: IEEE, May 2018, pp. 944–961, ISBN: 978-1-5386-4353-2. DOI: 10.1109/SP.2018.00018. [Online]. Available: https://ieeexplore.ieee.org/document/8418647/ (visited on 08/01/2020) (cit. on p. 27).

[57] *Iden3/circom*, original-date: 2018-08-09T06:20:52Z, Aug. 2, 2020. [Online]. Available: https://github.com/iden3/circom (visited on 08/02/2020) (cit. on pp. 27, 30).

[58] *Iden3/snarkjs*, original-date: 2018-08-09T06:16:06Z, Jul. 30, 2020. [Online]. Available: https://github.com/iden3/snarkjs (visited on 08/02/2020) (cit. on pp. 27, 30).

[59] *Ing-bank/zkrp*, original-date: 2019-07-17T09:07:53Z, Jul. 27, 2020. [Online]. Available: https://github.com/ing-bank/zkrp (visited on 08/02/2020) (cit. on p. 27).

[60] P. B. B. OBE. (Jul. 12, 2019). The power of go and the threat of ransomware: Meet eCh0raix, Medium. Library Catalog: medium.com, [Online]. Available: https://medium.com/asecuritysite-when-bob-met-alice/the-power-of-go-and-the-threat-of-ransomware-meet-ech0raix-f663befee161 (visited on 07/29/2020) (cit. on p. 28).

[61] (Feb. 5, 2020). Using zero-knowledge proofs with fluree fluree, Fluree. Library Catalog: flur.ee, [Online]. Available: http://flur.ee/2020/02/05/using-zero-knowledge-proofs-with-fluree/ (visited on 08/03/2020) (cit. on p. 38).

[62] (). Gartner 2019 hype cycle for blockchain business shows blockchain will have a transformational impact across industries in five to 10 years, Gartner. Library Catalog: www.gartner.com, [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-09-12-gartner-2019-hype-cycle-for-blockchain-business-shows (visited on 08/03/2020) (cit. on pp. 41, 42).

[63] "Distributed ledger technology: Beyond block chain", p. 88, (cit. on p. 45).

[64] *Trusted third party*, in *Wikipedia*, Page Version ID: 937194896, Jan. 23, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Trusted_third_party&oldid=937194896 (visited on 08/04/2020) (cit. on p. 45).

[65] *Turing machine*, in *Wikipedia*, Page Version ID: 967654699, Jul. 14, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Turing_machine&oldid=967654699 (visited on 08/04/2020) (cit. on p. 45).

[66] *Corporate social responsibility*, in *Wikipedia*, Page Version ID: 966595548, Jul. 8, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Corporate_social_responsibility&oldid=966595548 (visited on 08/04/2020) (cit. on p. 46).

[67] J. Chen. (). High-frequency trading (HFT) definition, Investopedia. Library Catalog: www.investopedia.com, [Online]. Available: https://www.investopedia.com/terms/h/high-frequency-trading.asp (visited on 08/04/2020) (cit. on p. 46).

[68] (). What is secure multiparty computation, inpher. Library Catalog: www.inpher.io, [Online]. Available: https://www.inpher.io/technology/what-is-secure-multiparty-computation (visited on 08/04/2020) (cit. on p. 46).

[69] V. Ganev and S. Deml. (). Introduction to zk-SNARKs (part 1), [Online]. Available: https://blog.decentriq.ch/zk-snarks-primer-part-one/ (visited on 08/04/2020) (cit. on p. 46).

Appendix

Appendix A: Bulletproofs in Go

The following listing is the complete version of the code for the range proofs in Golang.

```
package hedgefund

import (
        "errors"
        "fmt"
        "math/big"

        "github.com/ing-bank/zkrp/bulletproofs"
)

//Here I should decide how to calculate f and the source of
   this data

func ProveRiskWithinRange(a, b, secret int64) (bulletproofs.
   ProofBPRP, error) {
        proof := bulletproofs.ProofBPRP{}
        params, errSetup := bulletproofs.SetupGeneric(a, b)
        if errSetup != nil {
                fmt.Println(errSetup.Error())
                return proof, errSetup
        }
        bigSecret := new(big.Int).SetInt64(secret)
        proof, errProve := bulletproofs.ProveGeneric(
           bigSecret, params)
        if errProve != nil {
                fmt.Println(errProve.Error())
                return proof, errProve
        }
        return proof, nil
}

func CalculateAggregatedRisk(A []string, W, F []int64) (int64
```

```
    , error) {
        /*Here the aggregated risk is calculated according to
            the formula f= sum(wi*fi)
        * where wi is the weight of each asset in the
           portfolio and fi are the individual risk measures.
        *
         */
        var aggregatedRisk int64
        lengthW := len(W)
        lengthF := len(F)
        if lengthW != lengthF {
                return aggregatedRisk, errors.New("The number
                    of elements in W must be equal to the
                   number of elements in F")
        }
        for i := 0; i < lengthW; i++ {
                aggregatedRisk += W[i] * F[i]
        }
        return aggregatedRisk, nil
}

func RangeProver(path string) (bulletproofs.ProofBPRP, error)
    {
        proof := bulletproofs.ProofBPRP{}

        assets := ReadAssetsFromExcel("C:/Users/elviras/go/
           src/hedgefund/documents/weights")
        weights := ReadWeights("C:/Users/elviras/go/src/
           hedgefund/documents/weights")
        risk := ReadVolatility("C:/Users/elviras/go/src/
           hedgefund/documents/weights")

        //Calculate aggregated risk
        AggregatedRisk, errorCal := CalculateAggregatedRisk(
           assets, weights, risk)

        if errorCal != nil {
                fmt.Println(errorCal.Error())
                return proof, errorCal
        }

        //Create the proof
        proof, errorCal = ProveRiskWithinRange(200000,
           600000, AggregatedRisk)
        if errorCal != nil {
                fmt.Println(errorCal.Error())
                return proof, errorCal
        }

        return proof, nil
```

```
}
```

Appendix B: zkSNARKS in JavaScript

Listing 1: Code for zkSNARKs implementation

```
//class proof that generates and verifies the proofs
class Proof {



  constructor(){
    this.setup;
    this.proofParameters;
    }



  async setupCircuit(){
    const circuit = await loadR1cs(path.join(__dirname, "
       circuit", "circuit.r1cs"),true);
    this.setup =  zkSnark.groth.setup(circuit);

    this.setup.toxic = '';


  }

  async genProof(){
     //1. Parse input
     const input = JSON.parse(fs.readFileSync(path.join(__
        dirname, "circuit/data/", "input.json"),"utf8"));
     //2. Generate witness

     const wasm = await fs.promises.readFile(path.join(__
        dirname, "circuit", "circuit.wasm"));
     const wc = await WitnessCalculatorBuilder(wasm);
     const witness =  await wc.calculateWitness(input);
     this.proofParameters =  zkSnark.groth.genProof(this.
        setup.vk_proof, witness);
     console.log(this.proofParameters);

    // fs.writeFileSync(path.join(__dirname,"circuit/params
       /","circuit.proof"), proof.toJSON(), "utf8");
    // fs.writeFileSync(path.join(__dirname,"circuit/params
       /","circuit.publicSignals"), publicSignals.toJSON(), "
       utf8");
```

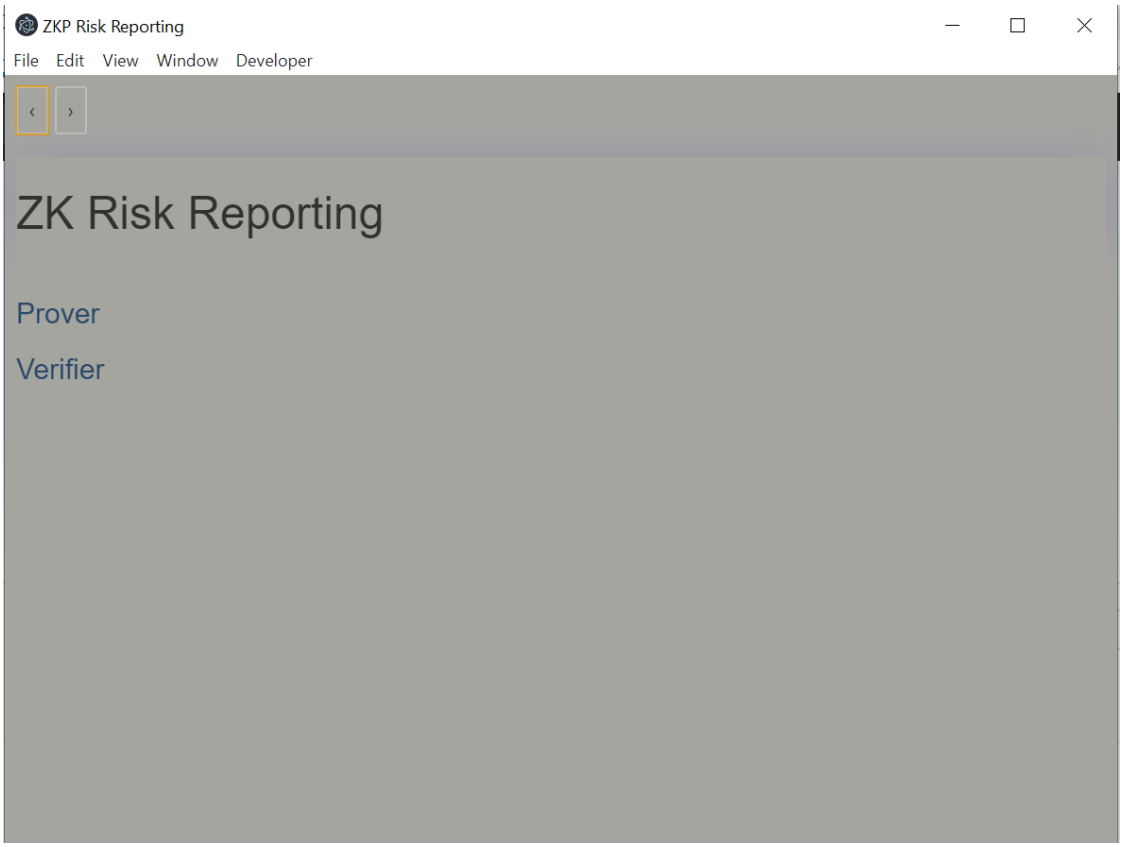
```
  }


  verify(){
      //const vk_verifier = JSON.parse(fs.readFileSync(path.
         join(__dirname, "circuit/params", "circuit.vk_
         verifier"), "utf8"));
      console.log(zkSnark.groth.isValid(this.setup.vk_
         verifier, this.proofParameters.proof, this.
         proofParameters.publicSignals))
      if (zkSnark.groth.isValid(this.setup.vk_verifier, this.
         proofParameters.proof, this.proofParameters.
         publicSignals)) {
          return true;
      } else {
          return false;
      }
  }
```

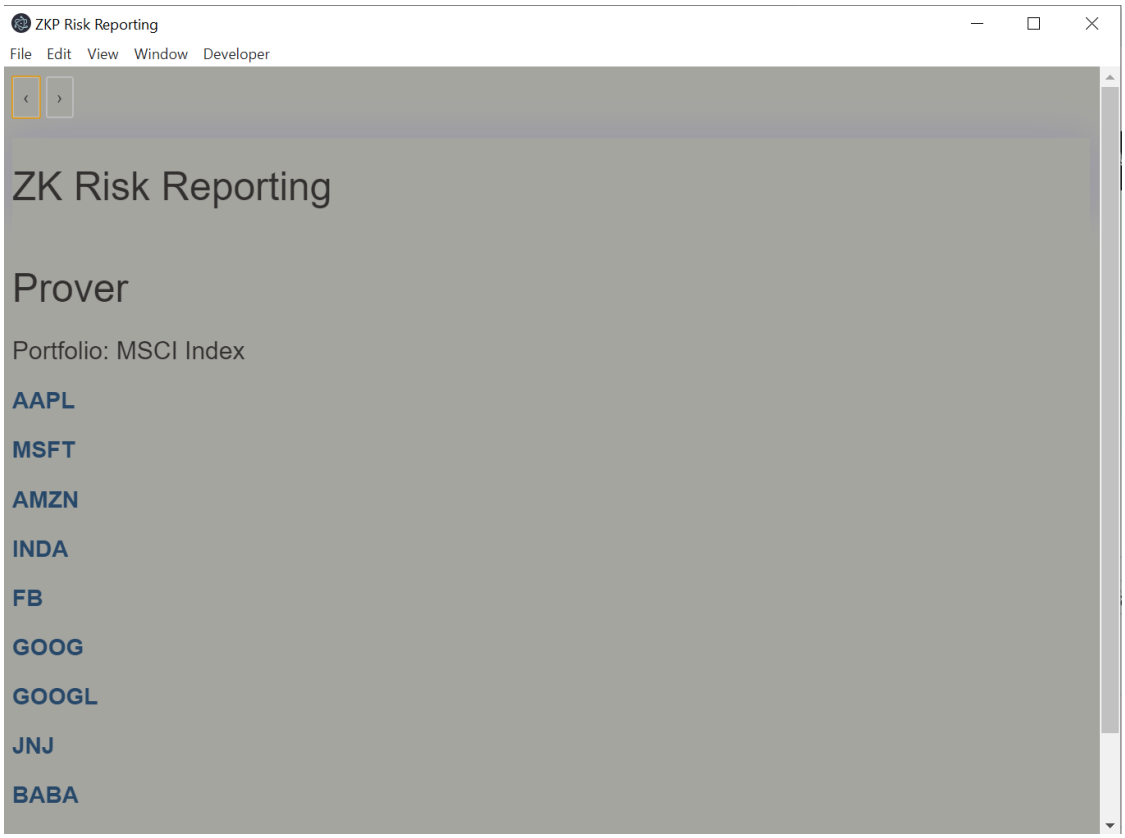
Appendix C: Proof of Concept: zkSNARKS

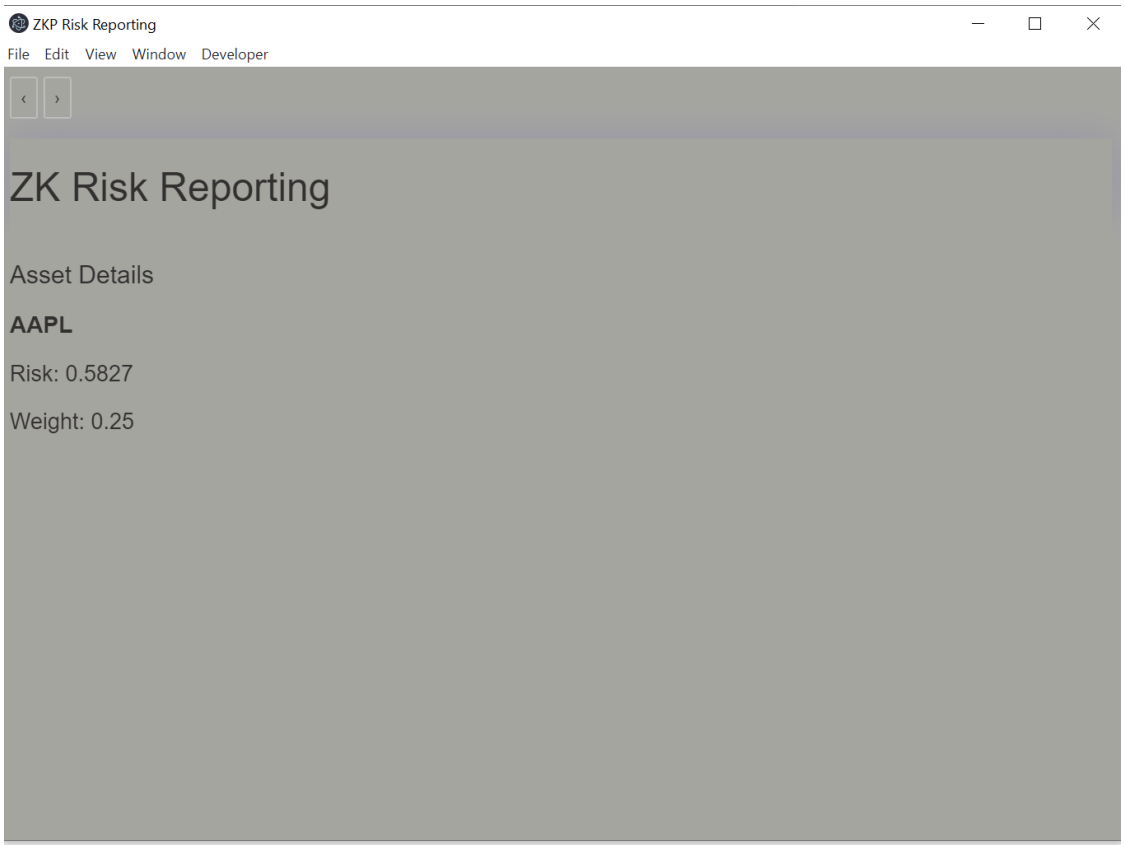
Listing 2: Proof generated by the zkSNARK in JSON

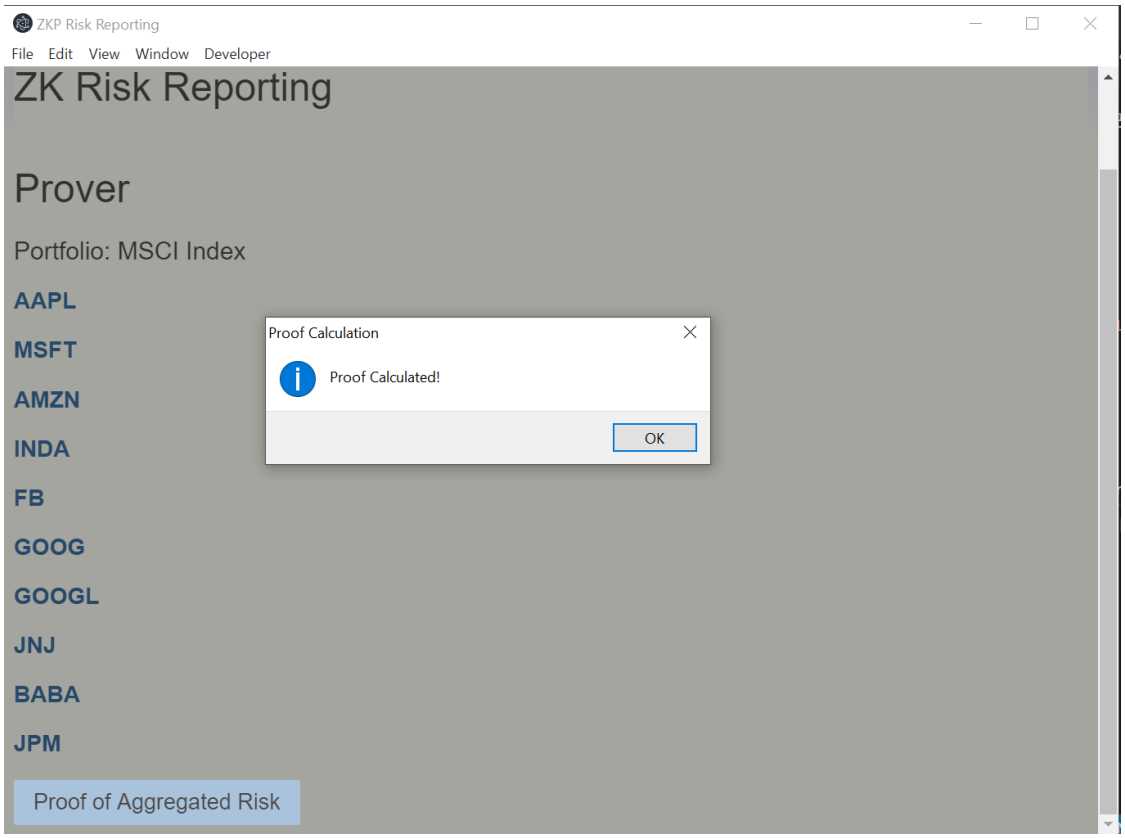
```
{
  proof: {
    pi_a: [
      3017243038822244589062790269530993024795071491304280031
      7559006770004762404 55n,
      2468133951422414197222960189817625056123176784833898969
      252742376406280129497n,
      1n
    ],
    pi_b: [ [Array], [Array], [Array] ],
    pi_c: [
      1052900094321766146874474821462889033214310573702628789
      3913520490243925276234n,
      4237209989879748434264013088259322523552671752978019760
      840559131822217736579n,
      1n
    ],
    protocol: 'groth'
  },
  publicSignals: [
         1n, 5827n,   5430n,
      3941n, 2584n,   4333n,
      4410n, 4418n,   3566n,
```

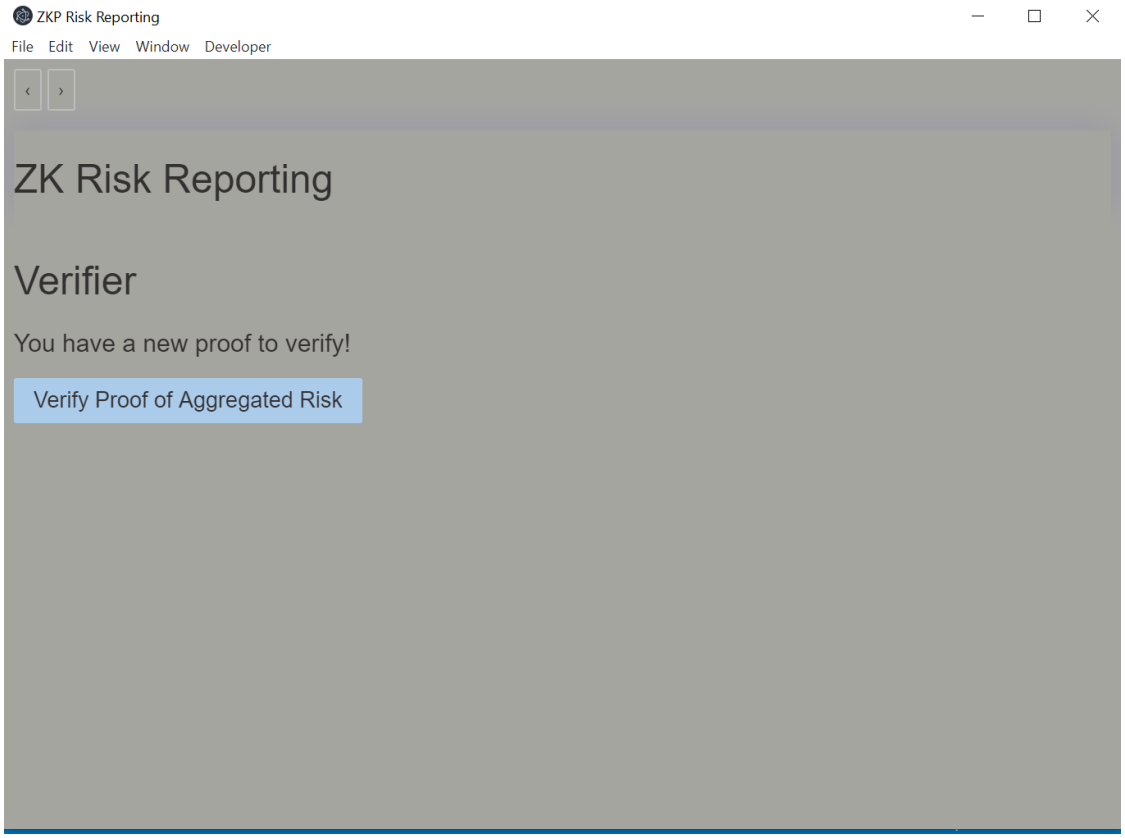
```
      3021n, 6218n, 200000n,
    600000n
  ]
```




ZKP Risk Reporting
File Edit View Window Developer
ZK Risk Reporting
Prover
Portfolio: MSCI Index
AAPL
MSFT
AMZN
INDA
FB
GOOG
GOOGL
JNJ
BABA


ZKP Risk Reporting
File Edit View Window Developer
ZK Risk Reporting
Asset Details
AAPL
Risk: 0.5827
Weight: 0.25


ZKP Risk Reporting
File Edit View Window Developer
ZK Risk Reporting
Prover
Portfolio: MSCI Index
AAPL
MSFT
AMZN
INDA
FB
GOOG
GOOGL
JNJ
BABA
JPM
Proof of Aggregated Risk
Proof Calculation
Proof Calculated!
OK


ZKP Risk Reporting
File Edit View Window Developer
ZK Risk Reporting
Verifier
You have a new proof to verify!
Verify Proof of Aggregated Risk

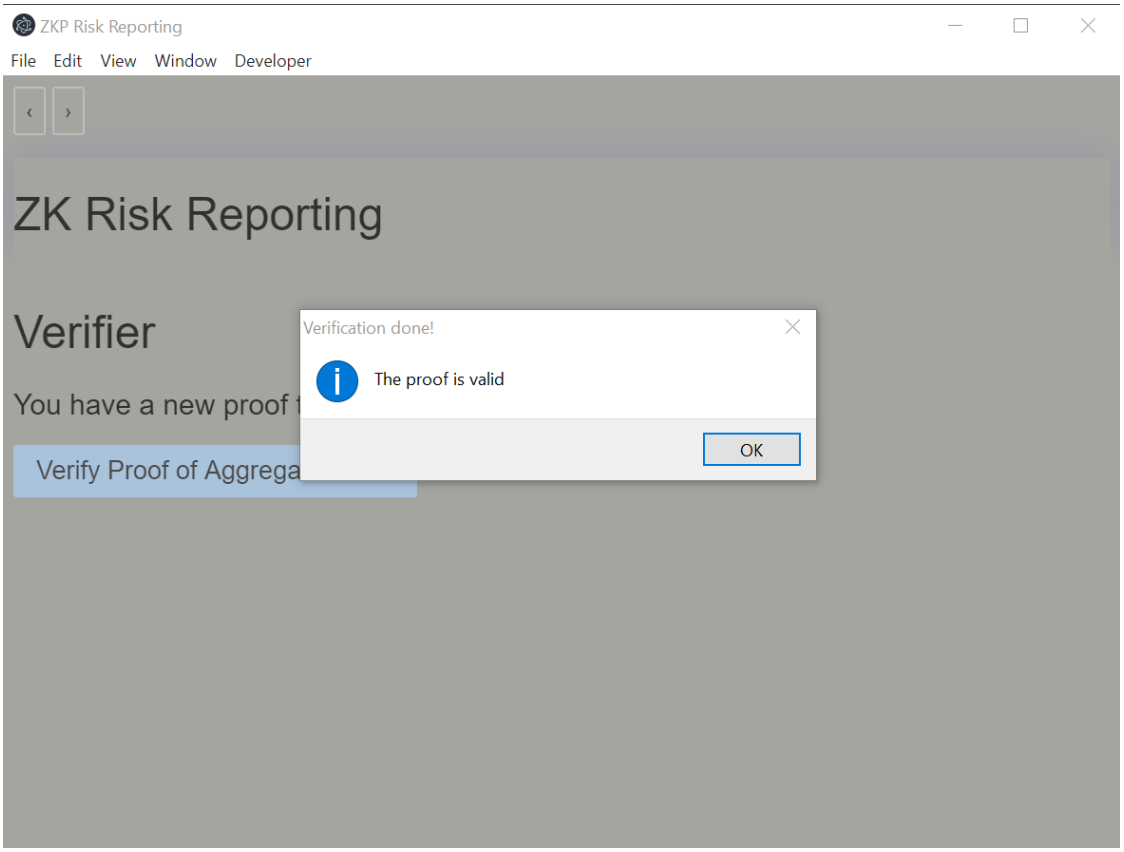


Figure A1: Screenshots of the prototype application for zero-knowledge risk-reporting