Application of Zero-Knowledge Proof for Decentralized Bayesian Games with Complex Rule

u7469758

April 2023

Word Count: 3067 words (excluding References)

Introduction

For the last decade, computational approaches with zero-knowledge proof (hereafter, ZKP) have been gaining global attention in many fields, namely cyber security and cryptocurrency [Gupta et al., 2020]. It is quite evident that ZKP's unique property of allowing one party to prove a statement to another party without revealing any other additional information can fundamentally change the paradigm of the centralized system towards decentralization. It means that initially-untrusted people can directly interact with each other to somehow gain trust, all while not leaking their secret information. For real-world examples, one can prove to a verifier that they are a valid user without giving away their passwords or people can make a financial transaction without having to go through the third-party/centralized system. In fact, such usages of ZKP are deeply involved in Privacy-enhancing technologies (PETs), which will become more important in this modern era since the number and cost due to cyber threats are greater than ever before (36 billion data breaches in the first half of 2020). [Dilmegani, 2022].

However, the extent to which ZKP can be applied to enhance security is far broader than just the fields mentioned above. Hence, this research aims to implement ZKP in online Bayesian games by making the core game system decentralized. Unlike the common cases where ZKP is used, players of the game with very complex rules are prone to unintentionally make invalid moves, which would correlate with one's experience level. Thus, this research will also explore whether introducing an additional parameter 'experience level' of players can improve the efficiency of the decentralized system along with the typical ZKP approach.

The ZKP can essentially be portrayed as a logically flawless and honest interaction between a prover and a verifier. In another word, it needs to satisfy all three criteria - completeness, soundness and zero-knowledge [Ethereum, 2023]. Suppose that S indicates the statement that the prover wants to prove is true, and P indicates that the verifier is convinced by the prover. Each criterion can be summarized as follows:

- Completeness: $S \rightarrow P$
- Soundness: $\neg S \rightarrow \neg P$
- Zero-knowledge: The verifier cannot deduce anything other than the truth of the statement.

It should be noted that the implication of the soundness is vulnerable for the verifier to be convinced by the false statement since ZKP is a probabilistic proof, not a deterministic proof like a typical math proof. However, the chance of a false positive is still extremely low and can be adjusted to as arbitrarily low as possible [Labs, 2019].

Research Problem

As the title of the proposal indicates, the research problem is "How can ZKP be used for decentralized Bayesian games". The types of ZKP and which one will be focused on for this research should be clarified. There are two main categories of ZKP, which are interactive and non-interactive proof [Kotecha, 2021].

The interactive ZKP requires a prover to perform a series of actions to convince the verifier. A typical example of this is 'colourblind friend and two balls'. Assume the prover, Peggy is not colourblind and tries to prove that she can distinguish the colours of red and blue balls to the verifier, Victor, who is colourblind. Hence Vicor is sceptical that the balls are distinguishable by colours and Peggy does the following to convince him. Peggy gives two balls to each hand of Victor and gives him the option to either switch or remain unchanged behind his back. Since Peggy indeed can distinguish the colours, she will be able to respond whether he has switched the balls or not. Of course, Peggy could have guessed with a chance of 50% even if she was colourblind as well (violation of soundness). However, as they repeat the process many times, the soundness error decreases to $\frac{1}{2^n}$, theoretically becoming negligible. Victor should eventually be convinced that Peggy's statement is correct (completeness), and yet he still doesn't know the colour of each ball or anything else other than the fact that Peggy can distinguish the colours of balls (zero-knowledge).

Meanwhile, the non-interactive proof allows a prover to send proof that any verifier can verify for themselves and be convinced. It essentially relies on the verifier picking a random problem for the prover to solve [Ray, 2019]. This type of ZKP eliminates the necessity of repeated interaction, which generally results in lower time complexity but higher space complexity to handle all of the random assignments of a problem with a hash function and the complexity of the problems in nature.

Out of these types of ZKP, the research attempts to utilize non-interactive ZKP, since the space complexity is not so much of a concern nowadays and this type is known to prove better for non-trivial and a number of statements (many rules in a game) [Gong et al., 2022]. Specifically, zk-SNRAKs will be used, which stands for Zero-Knowledge Succinct Non-interactive Argument of Knowledge. The zk-SNARKs require a trusted setup ceremony, where the keys that are used to create the proofs and the verification of those proofs are created. Having this proponent makes it more efficient and faster compared to other non-interactive ZKPs (see Figure 1), such as zk-STARKs and Bulletproofs, where a trusted setup is not required. But at the same time, the security of zk-SNARKs is heavily based on the security of the initialization of the trusted setup and it is also not quantum resistant, unlike the other non-interactive ones [Chainlink, 2023].

	SNARKs	STARKs	Bulletproofs
Algorithmic complexity: prover	O(N * log(N))	O(N * poly-log(N))	O(N * log(N))
Algorithmic complexity: verifier	O(1)	O(poly-log(N))	O(N)
Communication complexity (proof size)	O(1)	O(poly-log(N))	O(log(N))
size estimate for 1 TX	Tx: 200 bytes, Key: 50 MB	45 kB	1.5 kb
size estimate for 10.000 TX	Tx: 200 bytes, Key: 500 GB	135 kb	2.5 kb
Ethereum/EVM verification gas cost	600k (Groth16)	2.5M (estimate, no impl.)	N/A
Trusted setup required?	YES	NO	NO
Post-quantum secure	NO	YES	NO
Crypto assumptions	DLP + secure bilinear pairing	Collision resistant hashes	Discrete log

Figure 1: Comparison of the most popular non-interactive ZKPs [Gluchowski et al., 2023]

Before discussing how zk-SNARKs can be implemented in a game, a specific type of game should first be deconstructed. Firstly, the 'Bayesian' game refers to the element of incomplete information, so each player does not directly have access to all information of other players. Popular games, including poker and battleship, are considered Bayesian games [Greenwald, 2018]. Having the 'complex' rule is another attribute that the game has, as the rule should be intricate enough such that the parameter 'experience level' is able to significantly distinguish players' likelihood of making unintentional invalid moves. The parameter will be designed to manipulate the degree of rigorousness or complexity of the problems a player has to prove to other players. This slight adjustment hopes to make some contribution to the advancement of the implementation of ZKP in decentralized games.

In fact, many scholars have expressed their interest in this subject matter through their literature. There is research about how zk-SNARK was implemented for online battleship [Gupta et al., 2020]. The authors designed the game consisting of three components:

- Frontend using ReactJS
- Smart contract (trusted setup) using Ethereum virtual machine which functions as a backend storing the game state of every game as a list
- Proof generation and verification system using zk-SNARK integrated into the frontend and backend.

With this model, they were able to demonstrate the feasibility of ZKP for developing decentralized games. They further made a positive comment about how the application of ZKP can "open up interesting new design paradigms, especially for playing games online".

Similar to the research above, there is another literature investigating the technical progress and practice of zk-SNARKs, but in more depth, particularly with the Pinocchio protocol [Chen et al., 2021]. It also mentions the zk-STARKs and recursive zk-SNARKs as the future of ZKP for the games. Another insightful suggestion was that NFT could potentially be used as players' cards to play various games of incomplete information. Furthermore, it points out the difficulty in simulating the Bayesian games on ZKP-based chain, as preventing cheating and ensuring privacy need to be controlled simultaneously.

Consequently, it is evident that many literature works corroborate the usefulness of the ZKPs on Bayesian games and acknowledge its increasing significance in a variety of expertise. This research aims to contribute to this fascinating progress by examining the research problem.

Methodology

The research will be mainly divided into two parts, implementation of the zk-SNARKs for a specific game and simulation with participants to see how well it works. Without the loss of generality, the implementation will deal with a 2-player poker game, where the theory should be still applicable to other multiplayer Bayesian games or other similar fields.

As mentioned before, the trusted setup needs to be fulfilled so that proof and verification keys can be safely generated during the process of validating each other's moves. Following a common way for this to happen, a smart contract deployable on the Ethereum blockchain will be designed to run the trusted setup ceremony. The process of initiating a smart contract starts with the initial caller, who is represented by the green figure in Figure 2. The caller pays a small amount of Ethereum in the form of gas to encourage miners to execute the contract. They also initialize the contract by assigning certain variables, such as their balance, proof, or other data. The miners then run the contract and record the result on the blockchain. After multiple miners reach a consensus, the hash is added to the blockchain and can be accessed by any party represented by the orange figure [Yanai, 2019]. The hash informs all parties that an action has been executed. This process only needs to occur only once to start every game.

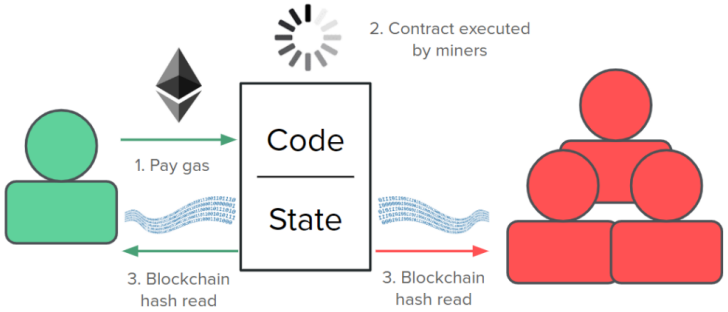

Figure 2: Execution of a sample smart contract [Gupta et al., 2020]

It should also be noted that this manual approach of the trusted setup is, of course, not as useful as the proof-of-work blockchains like Ethereum 1.0 or 2.0 due to the few minutes of execution time for the code to commit to the blockchain. Hence, this setup should be treated as a proof of concept, rather than the most optimized application.

With this trusted setup, zk-SNARKs can now be used to verify the validity of each player's moves by another player. More specifically, the Pinocchio Protocol will be implemented, which is the first practical implementation of zk-SNARKs [Parno et al., 2013], and its workflows can be categorized into four steps as follows [PPIO, 2019]:

- Transformation of the computational problem into the arithmetic circuits
- Conversion of the arithmetic circuits to Rank-1 Constraint System (hereafter, R1CS)
- Conversion of the R1CS to Quadratic Arithmetic Program (hereafter, QAP)
- Implementation of zk-SNARK algorithm based on the QAP

The research will almost follow this workflow as usual, which is definitely feasible as many researchers have successfully done so. In addition to this fundamental approach, the parameter 'experience level' (ϵ) will be adopted. It will be measured by the period of inactivity of the game, which was thought to be a more reliable and objective way rather than simply asking a player to express their own experience level. However, this is based on an assumption that a player is more 'experienced' as the period of inactivity is shorter at the current moment. The following equation to calculate

one's ϵ was motivated from the rating deviation of the Glicko rating system used in chess rating [Glickman, 2022], where ϵ is initially 350, t represents the period of inactivity in months and the constant 34.6 plays a role of determining the rate of change of ϵ, which can be adjusted.

$$\epsilon_{updated} = \sqrt{\epsilon^2 + 34.6^2 t}$$

The higher value of this parameter implies that a player has less experience with the game recently (higher deviation $\rightarrow$ lower reliability), and therefore more likely to unintentionally make invalid moves, meaning the rigorousness of the zk-SNARKs can be slightly adjusted higher compared to players with higher ϵ. Conventionally, there is a stage where a prover needs to prove that $P(x)$ generated from the QAP is divisible by the target polynomial $t(x)$ by checking only one random point since the probability of even having one identical point after going through all the processes with false statement is already sufficiently low [PPIO, 2019]. The manipulation of its rigorousness can be achieved by changing the number of points in the polynomial to be checked. One way this number can be calculated is the position of the first significant point of $\frac{10}{\epsilon}$ expressed as decimal. For example, the ϵ from 10 (non-inclusive) and 100 (inclusive) will have the first significant point of $\frac{10}{\epsilon}$ at tenth place, so only one point needs to be checked, while if the first significant point appears at the hundredth place, then two points will be checked and so on.

After all of the necessary implementations, the recruited participants with a variety of experience levels for playing poker will play the game. Since each verification of a move is independent of past verification and also only the experience level of a prover is used, the distribution and pair-matching of different experience levels do not need to be considered. To minimize the effect of individual differences, participants will play two or three games, depending on the sample size. Since this is only a simulation, no real monetary value will, of course, be used to bet in the game. Participants will not get any remuneration, but will sincerely be appreciated by the author.

In terms of ethical considerations, the research will follow the Ethics process at ANU and get its approval in advance. It is expected to receive E1 (Low Risk) since there is virtually no undue harm involved both physically and mentally in playing a few poker games with fake money. The right to withdraw will also be constantly reminded to all participants. In addition, the purpose of the simulation and raw data collected from it will be debriefed to the participants and only the consented data will be used. Finally, the source of the data will be confidential, as the author will likely be exposed to their information as an investigator of the simulation.

The timetable in Figure 3 was designed to guide and monitor the progress of the research.

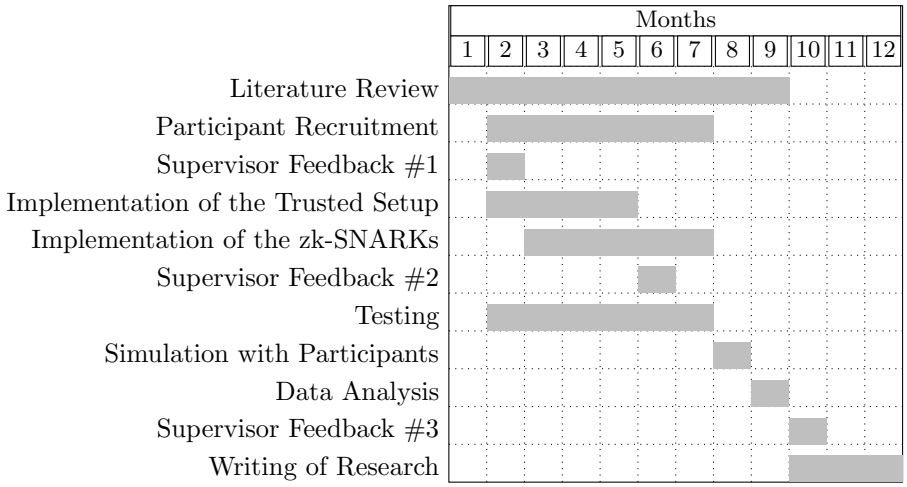

Figure 3: Research Timetable

Evaluation Criteria

Through the internal testing and simulation with participants, many useful data will be collected to assess the result of this research. White box tests can mainly be used as unit tests to validate the specification of all the sub-parts of the zk-SNARKs Pinocchio protocol. Black box tests can be used to examine the overall degree of completeness and soundness. If the input move is valid, then the output verification should accept it, while if the move was invalid, the verification should reject it. The soundness error gathered from this test along with the time taken to execute each move's verification in the simulation will be the main data to analyze. Their means will be used to represent their central tendency unless the prevalence of an outlier is at a concerning level. The zero-knowledge-ness of the zk-SNARKs should also be present in theory since the only adjustment made to the classical Pinocchio protocol only increases the rigorousness of the proof and reduces soundness error, which does not have any impact on the innate zero-knowledge property in the protocol. Examining whether participants cannot see other players' cards or any other private information other than being able to tell whether the opponent has made a valid move or not can further corroborate the last criterion of the ZKP.

The mean execution time of proof generation and verification can be compared with the typical Pinocchio's verification time, which is 10ms, using the one-sample T-test. Statistical significance in the test indicates that the population means of the execution time is not different from the hypothesized time. If the significance is not present, a percentage increase or decrease can be calculated to examine how much longer or shorter it takes to execute relative to the 10ms [Parno et al., 2013]. The soundness error should still be indistinguishable from 0, so another one-sample t-test can be used between the mean soundness error and 0 as a literature value. The statistical insignificance of this test would represent the soundness error is critical and hence the implementation should be revised. The standard deviation of the mean values will be used to determine how reliable and consistent the implementation is.

All of this data evaluation will be calculated in Python using the SciPy library and visualized appropriately using Panda and Matplotlib libraries. Overall, the result of the research can be assessed by the above numeric results with satisfying the three criteria of the ZKP as the most prioritized attributes. This is because the aspect of efficiency (the execution time of proof) can only be meaningful when the proof indeed is the ZKP.

Conclusion

To conclude, the objective of the research is to investigate the application of the ZKP, in particular with the zk-SNARKs, for Bayesian games with complex rules by decentralizing the verification system. This problem will contribute to the development of cheating-related security, although some aspects of cheating, such as collusion, ghosting, and the use of bots [Peckaitis, 2020] would need to be dealt with through other approaches. The additional parameter 'experience level' will be employed with the intention to improve efficiency by counterbalancing the given information about the likelihood of making an unintentional invalid move and the rigorousness of the verification in the process of the Pinocchio protocol. This implementation of zk-SNARKs has the potential to be applied in other Bayesian areas where gaining specific information can undoubtedly infer something important that allows the modification of the original protocol. Ultimately, the insights and results from this research anticipate contributing to a wider application of the ZKP. This discovery and development include, but are not limited to, scopes as personal as whistle-blowers exchanging information securely [Jie, 2019] to as global as verifying nuclear disarmament without leaking any other top secrets [Philippe et al., 2016].

References

[Chainlink, 2023] Chainlink (2023). Understanding the difference between zk-snarks and zk-starks.

[Chen et al., 2021] Chen, T., Lu, A., Kunpittaya, J., and Luo, A. (2021). A review of zero knowledge proofs.

[Dilmegani, 2022] Dilmegani, C. (2022). Zero-knowledge proofs: How it works & use cases in 2023.

[Ethereum, 2023] Ethereum (2023). What are zero-knowledge proofs?

[Glickman, 2022] Glickman, P. M. E. (2022). Example of the glicko-2 system.

[Gluchowski et al., 2023] Gluchowski, A., Korolev, P., and Scaffino, G. (2023). Awesome zero knowledge proofs (zkp).

[Gong et al., 2022] Gong, Y., Jin, Y., Li, Y., Liu, Z., and Zhu, Z. (2022). Analysis and comparison of the main zero-knowledge proof scheme.

[Greenwald, 2018] Greenwald, P. (2018). Bayesian games.

[Gupta et al., 2020] Gupta, A., Kaashoek, N., Wang, B., and Zhao, J. (2020). Zero-knowledge battleship.

[Jie, 2019] Jie, K. W. (2019). Private voting and whistleblowing on ethereum using semaphore.

[Kotecha, 2021] Kotecha, D. (2021). Zero knowledge proof.

[Labs, 2019] Labs, O. (2019). Zero-knowledge proofs: An intuitive explanation.

[Parno et al., 2013] Parno, B., Howell, J., Research, M., Gentry, C., Raykova, M., and Research, I. (2013). Pinocchio: Nearly practical verifiable computation.

[Peckaitis, 2020] Peckaitis, T. (2020). The most common poker cheats and how to avoid them.

[Philippe et al., 2016] Philippe, S., Goldston, R. J., Glaser, A., and d'Errico, F. (2016). A physical zero-knowledge object-comparison system for nuclear warhead verification.

[PPIO, 2019] PPIO (2019). Code talks: The a-to-z on zksnarks and zero-knowledge proof.

[Ray, 2019] Ray, S. (2019). What are zero knowledge proofs?

[Yanai, 2019] Yanai, A. (2019). The trusted setup phase.

Literature Review: Application of Zero-Knowledge Proof for Decentralized Bayesian Games with Complex Rules

u7469758

May 2023

Word Count: 4671 words

Contents

1 Introduction

As there is a saying that "knowledge is a power" and knowledge requires information, it is a natural desire for humans to gain information as much as possible. Having said that, protecting one's own information from those who are trying to fetch information is also a very intuitive behaviour. These traits are significant, especially in the modern digital era, where there are interactions of information more than ever before. This means that it sometimes becomes inevitable for one to share part of their information with others to achieve a certain goal, such as presenting an identification card to buy alcohol or providing a password to log in. Luckily, the Zero-Knowledge Proof (hereafter, ZKP) can accommodate the desires mentioned above such that a verifier has gained just enough information to be convinced the prover is indeed over the age of drinking or a valid user without leaking additional information.

Inspired by this pondering, this research endeavours to implement ZKP for decentralized Bayesian games with complex rules. In short, the research will perform the Pinocchio protocol to decentralize the online poker game with some adjustments. By implementing such a method, the players should be convinced that other players are not cheating without knowing their cards or other additional information. It will introduce the additional parameter 'experience level' intending to improve efficiency by counterbalancing the given information about the likelihood of making an unintentional invalid move against the rigorousness of the verification process. Despite the attempt to make a valid cheat-proof system, the ZKP by nature cannot catch some aspects of cheating, such as collusion, ghosting, and the use of bots [1]. Nevertheless, the implementation of the new parameter has the potential to be applied in other Bayesian areas where gaining specific information can undoubtedly infer something important that allows the modification of the original protocol. Ultimately, the insights and results from this research anticipate contributing to a wider application of the ZKP.

Consequently, this literature review aims to provide the foundation of relevant knowledge and portrays the prior and fairly recent state of the research topic. Most of the review will be about the ZKP, beginning with the background and history of ZKP and narrowing the focus to the specific ZKP that this research will implement, namely the Pinocchio protocol in Zero-Knowledge Succinct Non-interactive Argument of Knowledge (hereafter, zk-SNARKs), along with its other notable applications. The review will then end with briefly providing an overview of the Bayesian games and rating deviation in the chess rating system to convey their relevance to the research.

2 Literature Review

2.1 Background and History of ZKPs

A holistic view of the ZKP will be dealt with in this section to ensure readers become knowledgeable about this field to some extent, hoping to make the rest of the review more comprehensible.

2.1.1 Foundation of ZKPs Theory

In 1989, the concept of ZKP was first proposed by Goldwasser et al as they were first fascinated by the idea of a zero-knowledge interactive proof system, and then were motivated to make it applicable in the area of cryptographic protocols [2]. The research defines the zero-knowledge proof to be an interactive protocol that has biconditional implications between the truthness of the claim and the outcome of the verification (completeness and soundness). The third property, zero knowledgeness (indistinguishability of random variables), should be discussed more in detail, as this is what distinguishes ZKP from other types of proofs.

But first, notice that the interactive protocol was denoted as an ordered pair of an interactive Turing machine (ITM)'s A (prover) and B (verifier) where Machine A is not computationally bounded, while Machine B's computation time is bounded by a polynomial in the length of the common input. In other words, the overall problem is a language of NP class, where the details that the prover has is exceptionally difficult to be found out by the verifier, but the answer to the problem can be verified in polynomial time. In this respect, the computational zero knowledge in an interactive protocol (A,B) is said to be achieved when the following happens. For every polynomial time of B, the distribution that it 'sees' (gain information) on all its tapes, when interacting with A on input x, is 'statistically indistinguishable' from a distribution that can be computed from x in polynomial time. More formally speaking, given two families of random variables, $U(x)$ and $V(x)$, they are statistically indistinguishable, if for all constants $\epsilon > 0$ and all sufficiently long input x,

$$\sum_{\alpha \in \{0,1\}^*} |P(U(x) = \alpha) - P(V(x) = \alpha)| < |x|^{-c}$$

Overall, this pioneering research set a milestone in the field of ZKP by proving a theorem that every language in NP has a zero-knowledge interactive proof system. It also commented that "these proof systems for NP languages appear to have application in just about every protocol problem", foreshadowing its great potential in applicability.

2.1.2 Interactive ZKPs

Two years later, the scholars who were deeply involved in the proposal of the ZKP, once again, published two papers about the ZKP, one for interactive ZKP [3] and another one for non-interactive ZKP [4]. This subsection will first discuss the interactive ZKP research.

It demonstrates the relevancy of ZKP in not just the cryptographic protocols, but also in graph non-isomorphism. It is interesting how graph non-isomorphism can be proved via interactive ZKP since the problem is not known to be in NP. Despite the uncertainty of the existence of an efficient algorithm that demonstrates two graphs are not isomorphic, the probabilistic property of the ZKP allows the likelihood of the verifier accepting isomorphic graphs (false positive case leading to soundness error) is at most 2^{-m}, where the m is the number of repetition the proof went through the verification stage. Nevertheless, the research accentuates the ZKPs still heavily rely on the quadratic residue problem, which is known to be an NP-completeness as follows. An integer q is called a quadratic residue (QR) modulo n, if there exists an integer x such that $x^2 \equiv q \ (mod \ n)$ (otherwise, quadratic non-residue (QNR)). Most situations where ZKP is required can be processed using the QNR.

1. Given an common input (x, y), $n = |x|$, the length of the binary representation of x.
2. The verifier B flips coins to obtain n random bits $b_1, b_2, \ldots, b_n$ and another n times to get random string $z_1 z_2 \ldots z_n$.
3. For each digit in the string B computes $w_1, w_2, \ldots, w_n$, where if $b_i = 0$, then $w_i = z_i^2 \ mod \ x$ and if $b_i = 1$, then $w_i = (z_i^2 y) \ mod \ x$. These $w_1, w_2, \ldots, w_n$ are sent to the prover A.
4. For each i, A sends bit c, where $c_i = 0$ if and only if w_i is a QR mod x. Note that A will have the secret information along with the common input (x, y) that allows one to be confident whether w_i is QR mod x.
5. B checks that $b_i = c_i$ for every i, and is "convinced" that $(x, y) \in$ QNR and furthermore the prover's claim is true.

This was thought to be one of the classic approaches to make ZKP usable in many cryptographic scenarios, and is still a foundational concept of the specific ZKP that the author will be using. Along with these technical insights, the author was also intrigued by the research's conclusion, which depicts the ZKP as a positive use of NP-completeness. Most NP-completeness results apparently had a negative utility, exposing the intractability of a problem. However, the fact that the problem is very hard to solve underlies the 'zero-knowledge' property.

2.1.3 Non-interactive ZKPs

As mentioned before, there is another big branch of ZKP called the non-interactive ZKP, which was designed to be a "new, simpler scenario for zero-knowledge"[4]. The research mentions that three main components differentiate standard ZKP from more traditional ones: interaction between the prover and verifier, hidden randomization (a verifier tossing coins and generating random string) and computational difficulty the prover embeds. It aimed to advance the efficiency and applicability of ZKP by achieving the same result as interactive ZKP with fewer components. The non-interactive one essentially simplifies the first two components by making the prover and verifier communicate only once to send proof and receive the verification result and making the coins and random string public to both parties.

Loosening criteria as such first theoretically changed the completeness rate and soundness error dramatically to at least $\frac{2}{3}$ and up to $\frac{1}{3}$, respectively, which are definitely too inaccurate to be used in real life. Hence, the researchers improved the procedure by embedding a 3-satisfiability (3 SAT)problem consisting of QR-related components as its triplets in the proof. Note that 3 SAT is another NP-complete problem that determines whether assigning truth values appropriately will make a given boolean expression in a conjunctive normal form true that consists of triplets as each clause $((..\vee..\vee..)\wedge(..\vee..\vee..)\wedge...)$. The soundness error was then capped at $2^{2n}7n(\frac{7}{8})^{11n}$, where n is the number of clauses in the 3 SAT. Although this error will still approach zero as the n approaches infinity, it requires at least 80 clauses to have a reasonable bound, as can be seen in Figure 1. In fact, this highlights how non-interactive ZKP, in general, would require an immense amount of memory compared to interactive ZKP, while the time complexity becomes significantly more efficient.

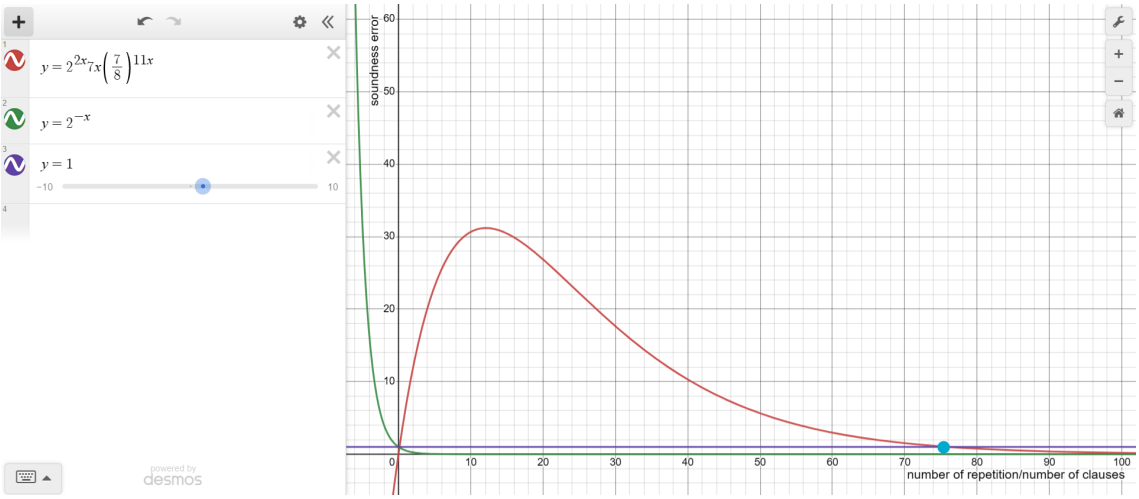

Figure 1: Soudness error comparison between interactive and non-interactive ZKPs

In this respect, the author hopes to improve the soundness error of certain non-interactive ZKPs with the new parameter 'experience level' in the context of games, as this will adjust the rigorousness of the proof.

2.1.4 Different Types of ZKPs

Thanks to these fundamental ZKP systems that have been developed a few decades ago, there are now various state-of-art ZKP systems, where practically all of them are non-interactive ZKP [5]. This trend depicts the outstanding efficiency of non-interactive ZKPs compared to the interactive ones, disputing the usability of the latter proofs mentioned in section 2.1.3, although it has undoubtedly set up a foundation for the more advanced ones. Figure 2 illustrates this wide range of systems in almost chronological order from Pinocchio to Zilch invented in 2013 and 2021, respectively. It is simply phenomenal to see how much progress has been made in this field as this latest system Zilch includes all the beneficial aspects in the figure while being easy to be programmed by users with an object-oriented language. Notice how some of the state-of-art ones are even resilient to attacks with quantum computational power. This may not be so meaningful in the context of this research but may be a desirable attribute as the value of the secret information becomes very significant as exemplified in section 2.2.2. Nevertheless, this incredible optimization indicates intricate logic is involved. Hence, the author decided to use the Pinnochio protocol to investigate the impact of the new parameter without loss of generality with a still relatively user-friendly programmability (procedural programming).

COMPARISON OF EXISTING ZKP SYSTEMS BASED ON THEIR CRYPTOGRAPHIC ASSUMPTIONS, THE NEED FOR A TRUSTED SETUP, THEIR UNIVERSALITY, AND RESILIENCE AGAINST KNOWN ATTACKS FROM QUANTUM COMPUTERS. REGARDING EASE OF PROGRAMMABILITY, EACH BAR INDICATES SUPPORT FOR DEVELOPING ZKPS USING ARITHMETIC CIRCUITS, ASSEMBLY LANGUAGE, PROCEDURAL AND OBJECT-ORIENTED PROGRAMMING, RESPECTIVELY. AMONG FRAMEWORKS THAT SUPPORT HIGH-LEVEL PROGRAMMING (I.E., THOSE WITH THREE OR FOUR BARS), ONLY ZILCH SUPPORTS THE OBJECT-ORIENTED PARADIGM

ZKP System	Protocol*	Cryptographic Assumptions¶	Transparent	Universal	Post-Quantum Resilient	Ease of Programmability ACs < ASM < PP < OOP‡	Compiler Available
Pinocchio [21]	zk-SNARK	KoE	○	○	○	▂▄▆▁	○
Geppetto [24]	zk-SNARK	KoE	○	○	○	▂▄▆▁	○
TinyRAM [22]	zk-SNARK	KoE	○	○	○	▂▄▆▁	○
Buffet† [18]	zk-SNARK	KoE	○	○	○	▂▄▆▁	●
ZoKrates† [63]	zk-SNARK	KoE	○	○	○	▂▄▆▁	●
xJsnark† [64]	zk-SNARK	KoE	○	○	○	▂▄▆▁	●
vRAM [65]	zk-SNARG	KoE	○	●	○	▂▄▁▁	N/A
vnTinyRAM [23]	zk-SNARK	KoE	○	●	○	▂▄▆▁	○
MIRAGE [31]	zk-SNARK	GGM	○	●	○	▂▁▁▁	N/A
Sonic [35]	zk-SNARK	AGM	○	●	○	▂▁▁▁	N/A
Marlin [66]	zk-SNARK	KoE, AGM	○	●	○	▂▁▁▁	N/A
PLONK [67]	zk-SNARK	AGM	○	●	○	▂▁▁▁	N/A
SuperSonic [34]	zk-SNARK	ARA	●	●	○	▂▁▁▁	N/A
Bulletproofs [30]	zk-ShNARK§	DL	●	●	○	▂▁▁▁	N/A
Hyrax [32]	zk-SNARK	DL	●	●	○	▂▁▁▁	N/A
Halo [68]	zk-SNARK	DL	●	●	○	▂▁▁▁	N/A
Virgo [33]	zk-VPD	CRHF	●	●	●	▂▁▁▁	N/A
Ligero [27]	zk-SNARK	CRHF	●	●	●	▂▁▁▁	N/A
Aurora [28]	zk-SNARK	CRHF	●	●	●	▂▁▁▁	N/A
zk-STARK [29]	zk-STARK	CRHF	●	●	●	▂▄▁▁	N/A
Zilch† (this work)	zk-STARK	CRHF	●	●	●	▂▄▆▆	●

† These zero-knowledge proof systems focus on front-end optimizations and offer comprehensive programming interfaces.
* SNARK stands for Succinct Non-Interactive ARgument of Knowledge, STARK stands for Scalable Transparent ARguments of Knowledge, SNARG stands for Succinct Non-interactive ARGuments, and VPD stands for Verifiable Polynomial Delegation.
¶ KoE stands for Knowledge of Exponent, AGM stands for Algebraic Group Model, GGM stands for Generic Group Model, ARA stands for Adaptive Root Assumption, DL stands for Discrete Logarithm, and CRHF stands for Collision-Resistant Hash Functions.
‡ ACs stands for Arithmetic Circuits, ASM is Assembly language, PP is Procedural Programming, and OOP is Object-Oriented Programming.
§ Bulletproofs is not considered a zk-SNARK because it is not succinct (i.e., has linear verification time). "Sh" stands for *short* instead of *succinct*.

Figure 2: Comparison of existing ZKP systems [5]

2.2 Application of ZKPs

As the overview of the background of ZKP might have already signaled enough, the extent to which it can be applied in different fields is very broad and continuously growing, so the author listed a few below.

2.2.1 Zero Knowledge Password Proofs

In 1992, one of the early applications of ZKP was a zero-knowledge authentication system, as it allows users to prove that they have a valid credential without having to give away the credential details [6]. This essentially happens via Encrypted Key Exchange (EKE) using Rivest–Shamir–Adleman (RSA) cryptosystem. In this system, the user's password is what ensures their confidence in determining a given value from a verifier is QR or QNR to the modulo n (public key) in the verification stage (see step 4 in section 2.1.2). It is also worthwhile noting that the EKE is designed to protect users with weak passwords, which one might argue that protocol should focus under the idealized circumstance, but the researchers rebut by saying "empirically, weak passwords are fact of life". In fact, such implementation revolving around the unavoidable flaw motivated the author to consider introducing a new parameter in the context of games with 'complex rules', since the protocol should ideally only focus on the cheaters. However, in reality, it is also inevitable that humans are susceptible to making mistakes and this could be interpreted as unintentional cheating, which is more likely to occur when the rules get complex enough. This conveys the importance of revising the context in which the theory is being applied, as identical theory or concept may need to be adjusted or can be improved depending on the context.

2.2.2 Nuclear Warhead Verification

The situation when ZKP is being used sometimes can be far more globally influential than when an individual tries to log in to their favourite website without leaking their password. Verifying nuclear warheads using ZKP, proposed in 2014, is such instance, where it is expected to be a critical feature in future rounds of arms-control negotiations [7]. This protocol is based on "non-electronic differential measurements of transmitted and emitted neutrons that can detect small diversions of heavy metal from a representative test object". It is an interactive ZKP system where both parties can be convinced that their potential warheads exhibit the same number of neutron transmissions and emissions, as simply illustrated in Figure 3. Of course, other protocols to ensure that both parties are detecting each others' real warheads are also considered, which involve some complex chemical theories.

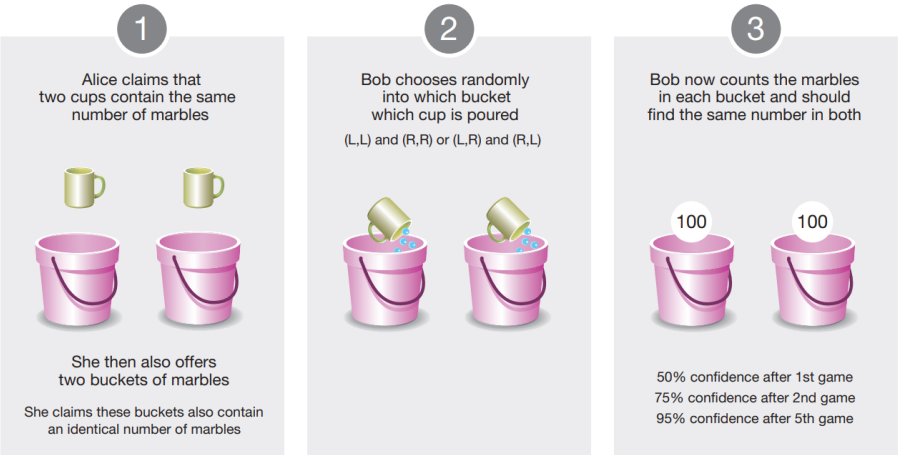

Figure 3: Nuclear warhead verification analogy using marbles as neutrons [7]

2.2.3 Decentralized Applications

The final notable application of ZKP is decentralized systems, which is more related to the application this research utilizes [8]. The research proposes zero-knowledge range proof (ZKRP), which is non-interactive and range-flexible. This proof brings advancement to the decentralized applications, namely blockchain that "operates on a peer-to-peer network with no central authority yet secured by cryptographic techniques". The 'range' in ZKRP means that the protocol can convince others that secret information/value is in between an interval without revealing any information from its witness. Additionally, the ZKRP is known to be range-flexible, implying there is no limitation on the lower bound and upper bound of the range as long as they are natural numbers. This allows such proof to be used beyond payment, including e-voting and e-auction systems. The superiority of non-interactive ZKP compared to interactive ZKP in decentralized applications has also been noted, which led the author to consider using interactive ZKP as the method for this research. Another keyword that the research discusses is the smart contract, which has apparently brought advancement of blockchain recently by providing general executable scripts. This terminology will further be decomposed in section 2.3.4.

2.3 Details of zk-SNARKs

This section will provide some more insights about the type of ZKP that is to be implemented in this research.

2.3.1 Pinocchio Protocol

In 2013, a year after the proposal of zk-SNARKs, the Pinocchio protocol was invented, which is known to be the first practical zk-SNARKS in a real-life context [9]. Pinocchio implements Verifiable Computation (VC) to allow clients to delegate a computation to a more powerful server while retaining the ability to verify the correctness of the results without performing the entire computation themselves. This makes the protocol useful as computational power between parties is often asymmetric. Essentially in this rseaerch, the client will create a

public evaluation key to describe player's current status. The other player then evaluates the status on a particular input and uses the evaluation key to produce a proof of correctness. Anyone can use a public verification key to check the proof. The process will be based on the end-to-end tool chain that "compiles a subset of C into programs that implement the verifiable computation protocol" (see Figure 4).

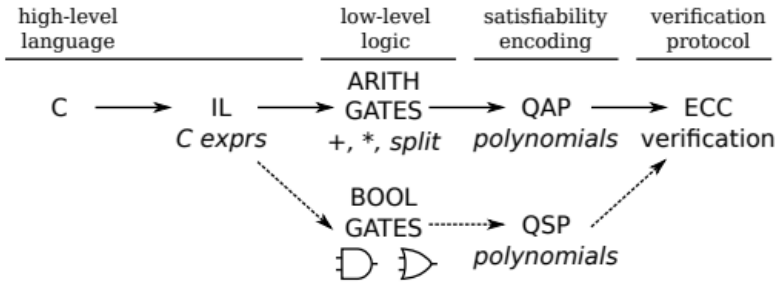

Figure 4: Overview of Pinnochio's toolchain [9]

Although the intermediate language can be converted into arithmetic gates or boolean gates (see Figure 4), the first option with Quadratic Arithmetic Program (QAP) polynomials were found to be more efficient as the size of Quadratic Span Program (QSP) was 38.6 times that of QAP and also QSP's key generation stage took 35.2 times and computing stage took 55.4 times that of the QAP. Consequently, this research will only follow the QAP path when utilizing the Pinocchio. The QAP has Q over field $\mathbb{F}$ contains three sets of $m+1$ polynomials $V = \{v_k(x)\}$, $W = \{w_k(x)\}$, $Y = \{y_k(x)\}$, for $k \in \{0...m\}$, and a target polynomial $t(x)$. Under this setting, it is said that Q computes F if and only if a given $(c_1, \ldots, c_n) \in \mathbb{F}^N$ makes the $t(x)$ divides $p(x)$, where

$$p(x) = \left(v_0(x) + \sum_{k=1}^{m} c_k \cdot v_k(x)\right) \cdot \left(w_0(x) + \sum_{k=1}^{m} c_k \cdot w_k(x)\right) - \left(y_0(x) + \sum_{k=1}^{m} c_k \cdot y_k(x)\right)$$

This is known to be NP-hard problem so finding the right set of c to satisfy the above criteria in polynomial time is intractable, while verifying the given set of c to determine whether $t(x)$ is divisible by $p(x)$ can be done in polynomial time. During this verification stage, the problem eventually gets reduced to checking x points to see whether $p(x)$ is divisible by $(x-\alpha), (x-\beta), \ldots$, where $\alpha, \beta, \cdots \in \mathbb{N}$. Normally, only one linear factor is checked at random, but this will be where this research adjusts the rigorousness of the verification by determining the number of points need to be checked based on the 'experience level' of players. Overall, the research well demonstrates the practicality and efficiency of the Pinocchio protocol, as it always produces 288-byte proofs, regardless of the size of the computation with a verification period of typically 10ms (when they experimented with Lenovo X201 ThinkPad). Nonetheless, the research acknowledges the gaps still exist, especially the possibility of parallel execution of the verification. The multi-party proof generation in QAP-based zk-SNARKs has actually been proposed in 2021 [10], but will not be dealt with in this research as unfortunately, its implementation is beyond the author's understanding of the field. The main drawback of the zk-SNARKs is known to be the mandatory aspect of the trusted setup, which is criticized in section 2.3.3.

2.3.2 Memory Optimization Method for zk-SNARKs

As much as zk-SNARKs including Pinocchio are practical and time efficient, they actually require huge memory to generate proofs (although the proof size is always constant), which leads to the extremely large size of arithmetic circuits even for a simple use case. In zk-SNARKs, it should be noted that a prover needs to first calculate every variable in the constraints, then generate a proof that depends on these variables. This unfortunately forces all the intermediate variables to be preserved. In addition to this inefficiency, the fact that the arithmetic circuit used in this process only supports addition and multiplication in $\mathbb{F}_p$, it does not natively support jumps, conditions, loops, and random accesses.

As a consequence, the research was conducted to optimize this memory issue by using a hashed-based method in 2023 [11]. The 'Split' was the initial optimization, which could "partition a zk-SNARK circuit into two components that can be processed sequentially" so that the obsolete variables are no longer preserved in the memory. Later in the research, they improved this idea to allow n-Split, reducing memory usage up to 50%. The circuit is said to have achieved 'Good n-Split' if it satisfies the following definition. A Good n-split is an n-split $S = (F_1, F_2, \ldots, F_n)$ such that $|m| << |F|$ and $max\{|F_1|, |F_2|, \ldots, |F_n|\}$ is relatively small, where F_i is a hash circuit, m is the intermediate variables and $|\ |$ is denoting the cardinality not the absolute value. In other words, a Good n-split should involve as few intermediate variables as possible. Although the research demonstrates memory can undoubtedly be minimized as long as a Good n-Split for a circuit exists, it also points out that an approximate algorithm to find a Good n-Split for arbitrary circuits has still not been identified. In fact, the author was first impressed by this optimization method and was considering utilizing it for the Pinocchio, but soon decided not to due to the gap above.

2.3.3 Protecting Data Feed to Smart Contract

Perhaps, one of the most criticized aspects of zk-SNARKs is the need to have an initial trusted setup ceremony. Such a setup initializes a set of public parameters that are required to generate ZKP. With this, a smart contract can be executed, which allows automation and enforcement of agreements without the need for intermediaries or third-party involvement. The issue is that the public parameters, almost serving a role of 'rules' are all dependent on this one-time setup. Hence, if the adversary succeeds in getting access to the randomness that generates the parameters, one becomes more than capable of creating false proofs that would be seen as valid by the verifier, which violates the soundness criteria of ZKP.

To mitigate this issue, zk-AuthFeed was invented in 2023 inspired by zk-DASNARK which extends the conventional zk-SNARK scheme with data authentication [12]. The new protocol was motivated as there were no satisfactory solutions to "attain privacy and authenticity at the same time". In this approach, the blockchain itself is leveraged as the data authenticator. A user of a decentralized application is required to consistently record their data on the blockchain, preventing any further modifications to the data. The protocol is known for its efficiency as key generation, proof generation and proof verification only takes about 10 seconds, less than 4 seconds and 40 ms, respectively. Such protocol is an incremental

contribution in the overall ZKP field, as there are already non-interactive ZKPs, namely zk-STARKs, where no trusted setup is required, but still convey a remarkable approach for the existing zk-SNARKs, the most common type of non-interactive ZKPs currently.

2.4 Overview of non-ZKP topics

The sections below will provide more detailed contexts of the new parameter 'experience level' and the games this research will be focusing on.

2.4.1 Overview of Rating Deviation in the Glicko Rating

The author was thrilled to use the concept of rating deviation used in the chess rating system for the new parameter in this research, since it was a fascinating instance where his past and current interests could assemble to create a new perspective. As a side note, the author has been interested in the chess rating system since high school, as evidenced by his IB Extended Essay about the reliability of Elo (original) and Glicko rating (improved) systems in reflecting one's performance in chess. The Elo rating system had a flaw in that a player may simply not play after reaching their highest rating [13]. The player's performance would be predicted as good as any other players with the same ratings, even if they may be a lot more active in reality.

Hence, the Glicko rating system was developed by Dr. Mark E. Glickman in 1995, as an extension of the Elo rating system, which takes the reliability of a player's rating into consideration. This system consists of another parameter called rating deviation (RD) on top of the variables in the initial system, which indicates the uncertainty of a player's rating. Unlike the Elo rating system, where the standard deviation of all players' ratings was fixed at 200, Glicko's RD fluctuates depending on the degree of the player's activity in chess, which increases as the period of inactivity increases. New players are given the maximum RD (350), as no game records make their rating as unreliable as it can be. Based on a Bayesian analysis, Glickman was able to derive the recursive equation for updating RD as follows, where t represents the appropriate rating period, which was a month in the proposed system, and c represents sensitivity in which the RD changes:

$$RD_{updated} = min\{\sqrt{RD_{old}^2 + c^2 t}, 350\}$$

With this RD, the reliability of one's performance ($g(RD)$) is measured by

$$g(RD) = \frac{1}{\sqrt{1 + \frac{3q^2(RD)^2}{\pi^2}}}, where\ q = \frac{ln(10)}{400}$$

In this research, the parameter RD will be denoted as 'experience level' as it is still measuring practically the equivalent value in a Bayesian game instead of chess, but will not be used as a standard deviation. Rather, it will indicate the likelihood of a player committing an unintentional mistake due to a sufficient complexity of the game's rule. The corresponding reliability value will then be categorized into 3 different ranges, $< 90\%$, $90\% \leq g(RD) \leq 99\%$

and $> 99\%$. Depending on which range the reliability falls under, the verification process mentioned in section 2.3.1 will check for three, two and one x point/s, respectively.

2.4.2 Overview of Bayesian Games

As the ZKP is being implemented to check players' status in Bayesian games in this research, the concept of the game should also be covered. It should be quoted that "Bayesian games (also known as Games with Incomplete Information) are models of interactive decision situations in which the decision makers (players) have only partial information about the data of the game and about the other players [14]." The paper further elaborates one of their properties as having unavoidable 'infinite hierarchies of beliefs'. This is due to the underlying assumption of the games that the decisions and beliefs of other players affect the decisions of the player. But after all, that player is also considered as an other player by another player, and this interactiveness results in a player needing to have beliefs on the others' beliefs on their beliefs and so on. Hence, it would be fatal in Bayesian games for one's secret information to be leaked, as it would turn the 'belief' into a guaranteed fact which then impacts the whole infinite chain. This is why ZKP comes in handy to convince everyone that no one is cheating, all while maintaining their secrets to themselves. Furthermore, the paper actually provides a touching insight into how the situation where people make a decision based on others' (based on others' and so on) with only partial information of others is all too common in real life. Such observation ultimately guides this research to find the usefulness of ZKP beyond just in Bayesian games and explore the insinuated real-life situation where untrusted people are ironically seeking for approval from each other.

3 Conclusion

To conclude, this literature review has provided detailed enough background information on the relevant topics and ultimately demonstrated the plausibility of the specific ZKP method, Pinocchio, on the specific type of field, Bayesian games with complex rules. It also indicates the finding in this research is still applicable in more advanced ZKP or zk-SNARKs by adjusting their corresponding verification stages with appropriate prior information, although the type of ZKP this research uses is not the most state-of-art one. Beyond the scope of this research, other opportunities for further improvement, including memory optimization and securing the data for smart contracts, have also been well identified.

References

[1] Thomas Chen, Abby Lu, Jern Kunpittaya, and Alan Luo. A review of zero knowledge proofs. *Semantic Scholar*, 2021.

[2] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[3] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.

[4] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991.

[5] Dimitris Mouris and Nektarios Georgios Tsoutsos. Zilch: A framework for deploying transparent zero-knowledge proofs. *IEEE Transactions on Information Forensics and Security*, 16:3269–3284, 2021.

[6] Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attack s. In *1992 IEEE Symposium on Security and Privacy*, Oakland, USA, 1992.

[7] Alexander Glaser, Boaz Barak, and Robert J. Goldston. A zero-knowledge protocol for nuclear warhead verification. *Nature*, 510(7506):497–502, 2014.

[8] Ya-Che Tsai, Raylin Tso, Zi-Yuan Liu, and Kung Chen. An improved non-interactive zero-knowledge range proof for decentralized applications. In *2019 IEEE International Conference on Decentralized Applications and Infrastructures*, San Francisco, USA, 2019.

[9] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, San Francisco, USA, 2013.

[10] Ali Rahimi and Mohammad Ali Maddah-Ali. Multi-party proof generation in qap-based zk-snarks. *IEEE Journal on Special Areas in Information Theory*, 2(3):931–941, 2021.

[11] Huayi Qi, Ye Cheng, Minghui Xu, Dongxiao Yu, Haipeng Wang, and Weifeng Lyu. Split: A hash-based memory optimization method for zero-knowledge succinct non-interactive argument of knowledge (zk-snark). *IEEE Transactions on Computers*, pages 1–14, 2023.

[12] Zhiguo Wan, Yan Zhou, and Kui Ren. zk-authfeed: Protecting data feed to smart contracts with authenticated zero knowledge proof. *IEEE Transactions on Dependable and Secure Computing*, 20(2):1335–1347, 2023.

[13] Mark E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Applied Statistic*, 48(3):377–394, 1999.

[14] Shmuel Zamir. *Bayesian Games: Games with Incomplete Information*, pages 426–441. Springer New York, New York, NY, 2009.