Advanced zkSNARKs with no Trusted setup

CSE 539: Applied Cryptography

Krishna Sree Gottumukkala
MSCS 2nd Year

Viraj Ravindra Sonaje
MSCS 2nd Year

Bhavani Mahalakshmi Gowri Sankar
MSCS 2nd Year

Abstract

The project report includes reviews of previously published Zero Knowledge Proofs (ZKP) that operate under the MPC paradigm. We implemented HyraxZK [15], which has comparatively smaller size for the proofs, increased performance, and most crucially, eliminated the need for a trusted setup. In addition to these characteristics, the proposed solution also produces proofs that are shorter and quicker than three out of five standard baseline systems that are currently widely used in practice.

1. Introduction

As zero knowledge proofs are computationally heavy to use these techniques in applications like IOT, Blockchain, or AI pipelines these proofs need to be designed in such a way that they will consume less time and resources. In late 1980s, Zk-SNARK was proposed which is an early succinct non-interactive zero knowledge proof that requires a trusted setup initially. A trusted setup has been one big flaw of the system since anyone with access to the private key then has access to the parameters used for the proof. Zk-SNARK is constructed based on multiple mathematical proofs such as Probabilistically Checkable Proofs (PCP), Quadratic Arithmetic Programs (QAP), Linear Interactive Proof (LIP), Polynomial Interactive Oracle Proofs (PIOP). The Zk-SNARK has been widely used to solve the major problems such as the fair-exchange problem [1], device registration and access to the Blockchain Network [2], verifiable provenance for decentralized AI pipelines [3], and decentralized blockchain based payment systems [4]. The presence of a trusted setup that runs an expensive one-time computation is essential to verify the validity of a proof. Hyrax relies on a standard discrete log cryptographic assumption and thus does not require a trusted setup. Hyrax is a short, crisp, doubly efficient SNARK. In this project, we will briefly go through how hyrax is implemented and compare the results with other baseline models. Current general-purpose ZK protocols have issues like having the proof size is to be linear or super-linear in the size of the computation, the prover or verifier is expected to evaluate super-linearly as compared to the time of verifying a witness, dependence on a trusted party to setup complex parameters and relying on non-standard assumptions. These concerns have restricted the usage of Zero Knowledge proof systems in several applications. We review protocols that overcome one or more of these challenges in the further sections and also see the current applications of such protocols.

2. Problem Setup and Threat Model

We intend to transform any computation for verifying an NP statement into a zero-knowledge proof of the statement's validity, which would assist us in addressing issues with existing ZK proofs and arguments. For an input x, witness w and an Arithmetic Circuit C of width G and depth d, and a design parameter ı ≥ 2 which will limit the tradeoff between the length of the proof and the time taken by the verifier.

- the proofs should be short, that is, sub-linear in the statement's size |C| and the witness |w| to the statement's validity; they require closer to $10d\ logG\ +\ |w|^{1/\iota}$ group elements
- the verifier should run in time linear in input plus proof size, ie, $O(|x|\ +\ d\ log\ G\ +\ |w|^{\iota-1/\iota})$, assuming C has sufficient parallelism.
- the prover, should run in time linear in the cost of the cryptographic operations for NP verification procedure, ie, $O(d\ log\ G\ +\ |w|)$, assuming C has sufficient parallelism
- the scheme should not require a trusted setup and is secured under the discrete log assumption in the random oracle model.

In addition, we have considered the advantages and disadvantages of using zero-knowledge proofs in practical settings, as well as the difficulties that arise when attempting to implement such a solution. Having effective implementations of the underlying ZKPs will make arriving at solutions easier. The applications we assessed focus on the protection of identity of the involved parties, confidentiality during data exchange, and security when the transactions involve dishonest provers. The threat models and security of the applications are discussed properly in their respective reviews.

3. Literature Review

- Ligero: Lightweight Sublinear Arguments Without a Trusted Setup [5]:

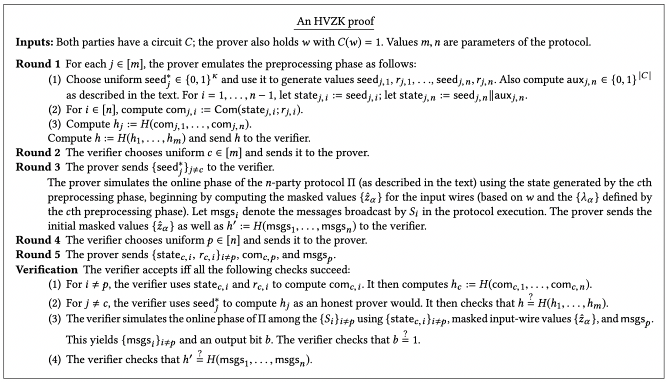
Ligero from Scot et.al.[6] is a NIZK (Non- Interactive Zero Knowledge) proof that is inspired by the existing approaches which aim to create practical ZK proofs namely, PCPs (Probabilistically checkable proofs), Linear PCPs, and Multiparty Computation. They have tried to combine all the salient features of these methods as much as possible and come up with an argument protocol for NP with communication complexity that runs in square-root of the verification circuit size. The protocol can be used with any collision-resistant hash which sounds like a flexibility but can be exploited when implemented using a poor hash function. They have relied on the random-oracle model which helps them to make their protocol a concretely efficient zk-SNARK without a trusted setup. The protocol uses "MPC-in-the-head" paradigm and takes the reader from implementing the general case to a ZKIPCP (ZK interactive PCP) using interleaved reed-solomon codes which are responsible for reducing the overall complexity of the algorithm. The paper shows how the overall algorithm works and the proofs for linear constraints and quadratic constraints. The paper first discusses how to create an interactive proof and then with the help of Fiat-Shamir Transform converts it to a non-interactive proof.

The most highlighted result the paper has described is the comparison with ZKB++ on the basis of communication size vs number of gates required. If we see the plot, Ligero performs significantly better than ZKB++. Although this paper puts forward an approach which has a superior efficiency and running time, it still depends on interleaved reed-solomon codes that basically require a lot of FFT computations which in turn require GPUs to calculate the results faster. The implementation of Ligero published relies on prime fields where the witness includes 2 bits per gate (AND and XOR) but instead if the algorithm uses characteristic 2 fields then the witness will need 3 bits for AND and none for XOR. The paper mentions that there are faster implementations available for characteristic 2 fields [7] [8]. The author mentions that the implementation of the protocol is production ready and can be used to make any cryptocurrency anonymous and secure. This paper is one of the state of the art techniques in the field of Zero knowledge proofs and also requires no trusted setup for its usage. The author also reveals that there can be more efficient implementations possible and there is work going on in that direction, tighter analysis of the algorithm is possible, and the proofs can be reduced in size with the use of recursion.

- Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures [9]:

As interesting the zero knowledge proofs are because of their ability to demonstrate proof of knowledge without actually sharing the important information the work of Jonathan et.al. [10] has improved it further to create a NIZKPoK (Non-Interactive Zero Knowledge Proof of Knowledge) that has comparable computations and shorter proofs than the prior works[11] [12] [13] [14] surveyed by them. The paper proposes a new way to instantiate the "MPC-in-the-head" due to which they are able to achieve these results. They have put forward a 5-round HVZK(Honest-Verifier Zero Knowledge) and also a compressed way to implement the 5-round approach that boils down to just 3-rounds. They use only symmetric key primitives like block ciphers and hashing techniques to implement their proposed system. The most significant contribution of this paper is that their approach claims to provide post quantum security. They have taken further steps to show the usefulness of their scheme by building standard signatures that are almost 3.2 times shorter than some leading hash-based signature approaches. In addition to this they also implemented group/ring signatures which asymptotically matches the state-of-the-art techniques. The algorithm discussed in this paper also improves the soundness as compared to ZKBoo and ZKB++ from 2/n to 1/n where n is the number of parties.

The paper although has linear complexity which is slower as compared to most of the SNARKs that are published and implemented but the most important feature here to observe is the post-quantum security and the length of the proofs are significantly smaller. We can also see that the results are compared to ZKB++ and Ligero and it outperforms Ligero up to 100,000 gates. This gives us enough gates to build applications like group and ring signatures. The only assumption used here when compressing to a 3-round model is that it relies on a random-oracle. Although the paper advertises about the post quantum security of its approach, they have only been tested against the known algorithms. As we know quantum computing is still a field that has very few real world implementations, the real quantum security remains a question. The potential extensions that we can look for from this paper are extensions of the "MPC-in-the-head" to build shorter NIZK proofs and there are still some aspects that can be realized from the theoretical world to implementations.



An HVZK proof

Inputs: Both parties have a circuit C; the prover also holds w with $C(w) = 1$. Values m, n are parameters of the protocol.

```
Round 1  For each $j \in [m]$, the prover emulates the preprocessing phase as follows:
    (1) Choose uniform $\mathsf{seed}^*_j \in \{0,1\}^\kappa$ and use it to generate values $\mathsf{seed}_{j,1}, r_{j,1}, \ldots, \mathsf{seed}_{j,n}, r_{j,n}$. Also compute $\mathsf{aux}_{j,n} \in \{0,1\}^{|C|}$ as described in the text. For $i = 1, \ldots, n-1$, let $\mathsf{state}_{j,i} := \mathsf{seed}_{j,i}$; let $\mathsf{state}_{j,n} := \mathsf{seed}_{j,n} \| \mathsf{aux}_{j,n}$.
    (2) For $i \in [n]$, compute $\mathsf{com}_{j,i} := \mathsf{Com}(\mathsf{state}_{j,i}; r_{j,i})$.
    (3) Compute $h_j := H(\mathsf{com}_{j,1}, \ldots, \mathsf{com}_{j,n})$.
    Compute $h := H(h_1, \ldots, h_m)$ and send $h$ to the verifier.
Round 2  The verifier chooses uniform $c \in [m]$ and sends it to the prover.
Round 3  The prover sends $\{\mathsf{seed}^*_j\}_{j \neq c}$ to the verifier.
    The prover simulates the online phase of the $n$-party protocol $\Pi$ (as described in the text) using the state generated by the $c$th preprocessing phase, beginning by computing the masked values $\{\hat{z}_\alpha\}$ for the input wires (based on $w$ and the $\{\lambda_\alpha\}$ defined by the $c$th preprocessing phase). Let $\mathsf{msgs}_i$ denote the messages broadcast by $S_i$ in the protocol execution. The prover sends the initial masked values $\{\hat{z}_\alpha\}$ as well as $h' := H(\mathsf{msgs}_1, \ldots, \mathsf{msgs}_n)$ to the verifier.
Round 4  The verifier chooses uniform $p \in [n]$ and sends it to the prover.
Round 5  The prover sends $\{\mathsf{state}_{c,i}, r_{c,i}\}_{i \neq p}$, $\mathsf{com}_{c,p}$, and $\mathsf{msgs}_p$.
Verification  The verifier accepts iff all the following checks succeed:
    (1) For $i \neq p$, the verifier uses $\mathsf{state}_{c,i}$ and $r_{c,i}$ to compute $\mathsf{com}_{c,i}$. It then computes $h_c := H(\mathsf{com}_{c,1}, \ldots, \mathsf{com}_{c,n})$.
    (2) For $j \neq c$, the verifier uses $\mathsf{seed}^*_j$ to compute $h_j$ as an honest prover would. It then checks that $h \overset{?}{=} H(h_1, \ldots, h_m)$.
    (3) The verifier simulates the online phase of $\Pi$ among the $\{S_i\}_{i \neq p}$ using $\{\mathsf{state}_{c,i}\}_{i \neq p}$, masked input-wire values $\{\hat{z}_\alpha\}$, and $\mathsf{msgs}_p$. This yields $\{\mathsf{msgs}_i\}_{i \neq p}$ and an output bit $b$. The verifier checks that $b \overset{?}{=} 1$.
    (4) The verifier checks that $h' \overset{?}{=} H(\mathsf{msgs}_1, \ldots, \mathsf{msgs}_n)$.
```

Shows a basic HVZK proof using 5 rounds

- Doubly-efficient zkSNARKs without trusted setup [15]

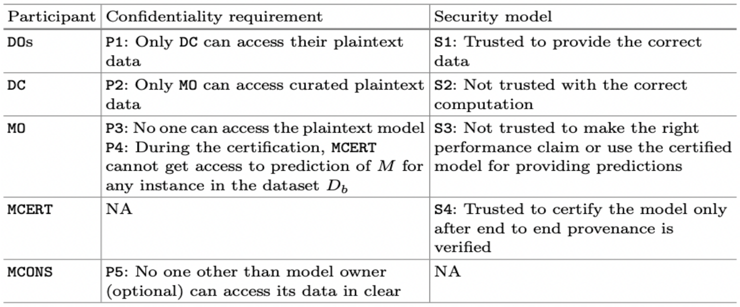
zkSNARK is a brief, non-interactive, and publicly verifiable zero-knowledge argument. The verifier examines the proof produced by the prover. Hyrax is an interactive zero-knowledge proof that establishes the verifiability of generalized boolean arithmetic circuits. It is based on the discrete log cryptographic assumption and does not need a once-expensive trusted setup. Hyrax's proof substitutes the messages of the prover with promises and transmits them to the verifier. Since the verifier is unable to discern the message from the commitment, Zero Knowledge is certainly applicable. Parallelism is extracted from serial calculations via a redistribution layer in Hyrax. Hyrax is adaptable to either smaller proofs with slower speed or larger proofs with quicker performance.

This scheme is practical because it tightly integrates three components: a state-of-the-art interactive proof (IP), which the paper modifies to reduce communication complexity; a highly optimized transformation from IPs to zero-knowledge arguments following the approach of Ben-Or et al. [16] and Cramer and Damgrd [18]; and a new cryptographic commitment scheme tailored to multilinear polynomials that adapts prior work [17, 19] to allow a sender to commit to a log G-variate multilinear polynomial and later to open it at one point, with O($G^{1/\iota}$) total communication and O($G^{(\iota-1)/\iota}$) receiver runtime for any ı ≥ 2.

The paper explored the efficacy of Hyrax when computations are sufficiently parallel or susceptible to batching. However, this suggests that the performance of the Hyrax relies on the level of parallelism, suggesting that the Hyrax would not perform well in situations where parallelism is very restricted. Given that libSTARK, an alternative to Hyrax, might have been used as a benchmark for multi-threaded implementation, the study did not include benchmarking and performance assessment for multi-threading. One of the most significant potential extensions to the paper would be to reduce proof size further without increasing verifier runtime.

- Zero Knowledge Proofs Towards Verifiable Decentralized AI Pipelines [1]

The work of Nitin et.al. [20] is one of the first steps towards employing zero knowledge proofs in the field of artificial intelligence. In an decentralized AI pipeline setup all the components of the pipeline are contributed from various sources which for instance can be independent organizations. So, inherently the individual contributors need trust between them to know if the data that is being used or the model that is working in the back-end is actually the same as what the owner claims. The scenario used in the paper consists of Data Owners(DO), Data Curators(DC), Model Owners(MO), Model Consumers(MCONS) and Model Certifiers(MCERTS). The paper helps to build trust between each of the entities moreover with the use of zero knowledge protects privacy of the overall structure. The underlying algorithm uses CP-SNARKs (commit and prove SNARKS) which works on the publicly available hashes of the constituents of the pipeline to establish a proof of knowledge. Below are the privacy requirements and the security assumptions of the constituents.



Participant	Confidentiality requirement	Security model
DOs	P1: Only DC can access their plaintext data	S1: Trusted to provide the correct data
DC	P2: Only MO can access curated plaintext data	S2: Not trusted with the correct computation
MO	P3: No one can access the plaintext model P4: During the certification, MCERT cannot get access to prediction of M for any instance in the dataset D_b	S3: Not trusted to make the right performance claim or use the certified model for providing predictions
MCERT	NA	S4: Trusted to certify the model only after end to end provenance is verified
MCONS	P5: No one other than model owner (optional) can access its data in clear	NA

The paper has used semi-honest verifiers to prove the completeness, soundness and zero knowledge of the proposed algorithm. They have internally used Adaptive Pinocchio[21] as the CP-SNARK and achieved innovation in the model certification, provenance architecture for AI artifacts and model inference with preserved confidentiality. For Data operations verification the system takes about 40 seconds to verify datasets of 100K rows and multiple columns although this sounds time consuming but the time is independent of the number of attributes of the dataset. This potentially unlocks a way to look into data having a large set of attributes with only limitation of number of rows which is a point that can be marked as a potential future work. The paper also compares with the existing state of the art zkDT[22] for decision tree inference and claims to be 1.5 - 4 times efficient. In my opinion the datasets used are only 100K rows and have 32-bit values which seems to be smaller for the actual production databases in organizations is a limitation of the paper. According to me, the applications of this line of work still are limited to the DCs, MOs, MCERTs because it is only practical for infrequent usage. If efficient implementations are made for consumers (MCONS) to use, the organizations can leverage it to build trust between the MOs and MCONS and report useful information like the fairness of the model.

- Demonstration of Blockchain-based IoT Devices Anonymous Access Network Using Zero-Knowledge Proof [4]

In this paper, they have used the Zero Knowledge proof for anonymous access in the network without exposing the confidential data to the several IOT devices. The whole system is designed to have two main processes: 1. For the new device to be registered in the block to be a member of a blockchain network. Each member in the network will verify if the identity hash of the device matches with the existing ones and they determine the new device is duplicate. If the hash value doesn't match in the block then all the members in the network determine the new device to be legal based on some threshold requirements and consensus algorithm. 2. For the legal device to access the blockchain network they have implemented a ZKP. For this to work, the device splits its original identity into two independent blocks for the participants to use that information to determine that the device is legal without exposing the complete identity information of the device.

It's necessary to come up with a novel approach for the access authentication process without exposing the original information of the user as it has been the quickly compromised approach in this process. In this paper, they have come up with a ZKP algorithm that ensures the privacy of users' private information while allowing access to the network.
Based on the algorithm, it splits the digital identity into two parts,

$$a = g^{k} \bmod p \tag{1}$$

$$m = \left(xa + kb\right) \bmod (p-1) \tag{2}$$

The operator (3) and device (4) calculates the,

$$X = y^{a} a^{b} \bmod p \tag{3}$$

$$Y = g^{m} \bmod p \tag{4}$$

If both sides return the same values then the device is legal and is registered to the block and can access the network. Otherwise, it fails to get the authentication.

However, in this paper they haven't discussed what different ZKP protocols can be used for the identity verification and haven't compared the results using multiple zero-knowledge proofs. As zero-knowledge proofs are costly, it's required to make sure that ZKP's don't add much more complexity as verification should be done in fraction of seconds.

- Exploiting Zero Knowledge Proof and Blockchains Towards the Enforcement of Anonymity, Data Integrity and Privacy (ADIP) in the IoT [23]

Using an interactive identification mechanism based on the Zero Knowledge Proof (ZKP) method, this paper discusses the anonymity of Internet of Things (IoT) devices. Each IoT device in the network is preloaded with two sets of keying information during the key generation and key distribution phase, and the IoT trusted key server produces and loads a set of unique private keys into each IoT device. The set comprises m-keys, k1,..., km, and the quadratic residues w1,..., wm for each of the m private keys are computed and broadcast across the IoT network. Prior to IoT device deployment, IoT devices are divided into clusters or groups, with each cluster Gi receiving a set of group-based secrets [24]. IoT devices inside group Gi, for example, are provided with an assortment of m-group-based keys, K1;Gi;...; Km;Gi. The IoT network then makes available the quadratic residues W1;Gi;...Wm;Gi for each group-based secret. ZKP proofs are computed and validated throughout the IoT network during anonymous authentication using group-based secrets. Identities of IoT devices are concealed by generating ZKP proofs and verification requests utilizing group-based shared secrets [25]. IoT devices employ their unique ZKP secrets to establish secure connections and enable identification by the IoT network in exceptional safety-critical situations when device traceability is more vital than maintaining its privacy. The illustration displays the proposed multifunction ZKP system with two cryptographic operation modes: (i) anonymous authentication mode and (ii) IoT device traceability mode. Our methodology generates two distinct blockchains.

Each IoT device and a public IoT blockchain participate in a ZKP authentication session which generates and exchanges x ZKP-blocks with the IoT ledger [26]. A ZKP-blockchain is then formed by hashing and encoding x ZKP-blocks into a Merkle tree [27]. A ZKP-blockchain is utilized to create the authentication-block, a bigger block. According to the report, each block needs only 2 MB of internal storage. Since IoT devices are often manufactured to be smaller and cheaper, this severely restricts the implementation of the suggested technique. Consequently, they have limited internal storage space that will be utilized to hold the authentication block. Inasmuch as the authentication protocol is implemented utilizing a blockchain to allow assessments of authentication session integrity, it would continue to be computationally demanding, difficult to scale, and performance-degraded. In addition, the ZKP operation can cause frequent bottlenecks in the blockchain's core network.

The paper addressed protection against IoT spoofing attack for the IoT ledger servers. However, it failed to address the sybil attack where malicious IoT nodes are introduced into the system, which could lead to potential take over of the IoT network by carrying out 51% attack.

The size of the authentication block can be reduced significantly by replacing Merkle trees with Verkle trees for storing the authentication blockchain. For comparison, the most fundamental characteristic of Verkle trees is their proof-size efficiency. A Verkle tree would require less than 150 bytes to produce a proof for a tree with a billion data points, as opposed to the approximately 1 kilobyte required by a typical binary Merkle tree.

- LiteZKP: Lightening Zero-Knowledge Proof-Based Blockchains for IoT and Edge Platforms [2]

The paper presents a decentralized ZKP-based P2P blockchain-based payment system with privacy protection for IoT and edge devices. By permitting decentralized financial services, devices with limited resources may exchange payments anonymously and without human or third-party involvement.[30] It offers a mini-merkle tree structure that decreases hash computations for merkle tree-based anonymity support, an integration of ZKP with off-chain payment channels to reduce repeated ZKP operations, and an optimization strategy to reduce proof creation and verification delay. Applying the zero-knowledge proof (ZKP) is recognized to be useful for achieving complete anonymity in blockchain activities[28]. LiteZKP aims to conduct a single ZKP operation to cover successive payments, as opposed to doing many ZKP procedures each payment. This enables LiteZKP to decrease the processing overhead required to facilitate blockchain-based payments, which reduces both latency and energy consumption. A note is used in LiteZKP to conceal the recipient's address and the money being sent. The paper employs notes as leaf nodes of a Merkle tree inside a smart contract to validate existing notes. Given that the number of hash computations for merkle tree route verification rises proportionately with tree depth, it is essential to reduce this depth for its implementation on systems with restricted resources. This is accomplished by dividing a single (deep) merkle tree into many tiny (shallow) merkle tree structures.

Given that the paper's use case is a payments system, the number of transactions per unit of time will be quite large. In addition to the blockchain processes that demand vast quantities of compute, this also requires large amounts of processing power. In IoT applications involving frequent data transmission, the operational costs of blockchain methods may outweigh their advantages. IoT nodes that communicate with edge platforms to exchange data at a predetermined fee per data point incur additional costs[29]. In terms of practical applications, the suggested method might prove to be a viable alternative to large blockchain-based currencies such as Etherium. However, the article did not devote sufficient attention to the LiteZKP's security. As it utilizes blockchain technology, it is not immune to the inherent security flaws of blockchain due to its reflecting characteristic. A malicious buyer sends an anonymous transfer transaction with the same transaction cost to an IoT device they control and a legitimate IoT device. Having control over one IoT device, an attacker may deceive other users by rejecting the first transaction, prioritizing the second transaction, and broadcasting the second transaction to the rest of the network.

The research makes a crucial assumption that the note parameter used to guarantee anonymity with ZKP is a hash, however it does not discuss which hashing method would be the most successful in preventing collisions. If the employed hashing method is not very resistant to collisions, the whole premise of employing ZKP for anonymity would be rendered worthless.

- Zero-Knowledge Contingent Payments Revisited: Attacks and Payments for Services [3]

In this paper, drawbacks of current ZKCP protocols for fair exchange problems have been brought up and also addressed how to solve these problems. Since exchanging digital goods between two parties with both of them cannot get misled due to another party. This has been studied for decades and it has been determined that without the use of a trusted third party, this can't be achieved. Fair-exchange of digital goods between parties have been studied for decades and have been determined that without the use of a trusted third party, this can't be achieved. The new innovations of Blockchain and Bitcoin have brought the key concept where it can be used as a trusted third party in a trustless manner.

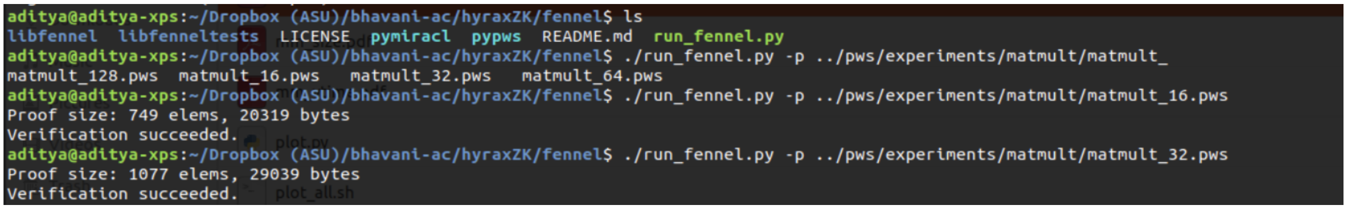
In a fair-exchange problem, bitcoin blockchain can be used. However, the scripting language of Bitcoin is not supportive to write any conditions that are required for money transactions. Here comes the ZKCP protocol for a fair-exchange utilizing the scripting feature of Bitcoin. In the earlier ZKCP protocol, common reference strings are used instead of trusted third parties. The assumptions made for solving the fair-exchange problem are to convince the buyer, so the buyer would act as a trusted third party and the buyer will generate CRS honestly. If the assumptions were broken then the fraud buyer can craft the CRS such that the buyer can find out some bits of information from the seller which breaks the zero-knowledge property. In the earlier ZKCP, no checks are performed even from the seller end which allows for the buyer to generate malignant CRS to capture the bits of information the seller has.

In a problem, Proof-of-Retrievability(PoR) which is similar to the sudoku example, the previous ZKCP protocol fails as PoR is the PoR and once either party receives what they require then they can abort the protocol. Zero-Knowledge Contingent Service Payments (ZKCSP) is designed in this paper to address these types of problems. The brief overview of the proposed ZKCSP protocol is, for the prover to prove it has m i.e v(m) = 1;
The prover generates a string y and provides a proof with zero knowledge proof that verifies the following:

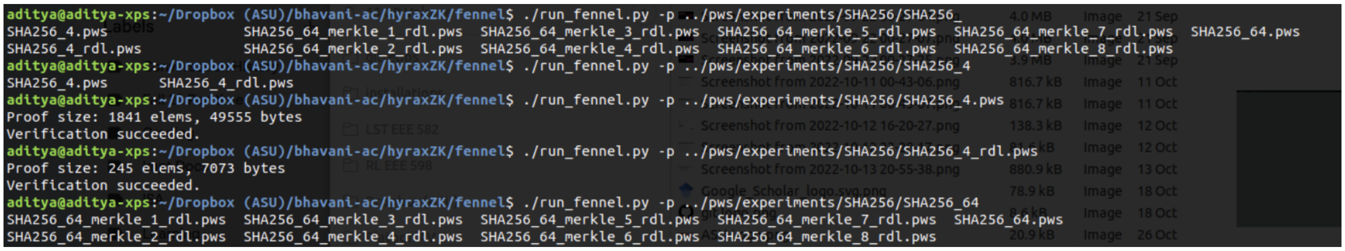
If v(m) = 1, then I know the preimage of y under SHA256.
But, if v(m) = 0, then the probability that I know a SHA256 preimage of y is negligible
There are many real-time applications which require this ZKCSP protocol such as BugBounty, GoodCode Inc, are the popular ones.

4. Evaluation

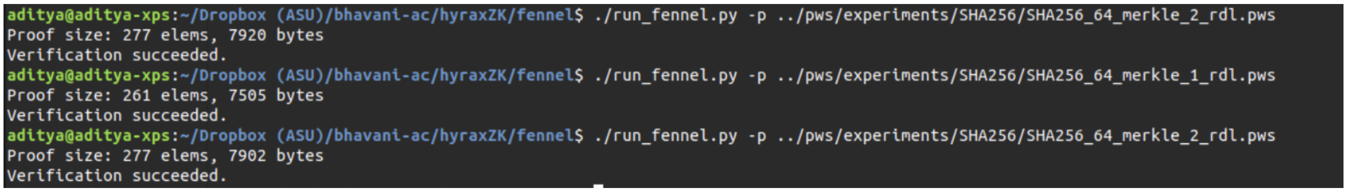
We built the reference implementation of Hyrax along with all of its dependencies. Fennel is the primary Hyrax codebase, pws contains scripts for producing prover worksheet (PWS) files and sample runscripts for fennel, and libpws is a library used by fennel for parsing PWS files. We execute fennel for two experiments: matrix multiplication and SHA256 on prover worksheets with various sizes. We have implemented this code on 12th Gen Intel® Core™ i5-1230U.



```
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ls
libfennel  libfenneltests  LICENSE  pymiracl  pypws  README.md  run_fennel.py
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/matmult/matmult_
matmult_128.pws  matmult_16.pws   matmult_32.pws   matmult_64.pws
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/matmult/matmult_16.pws
Proof size: 749 elems, 20319 bytes
Verification succeeded.
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/matmult/matmult_32.pws
Proof size: 1077 elems, 29039 bytes
Verification succeeded.
```

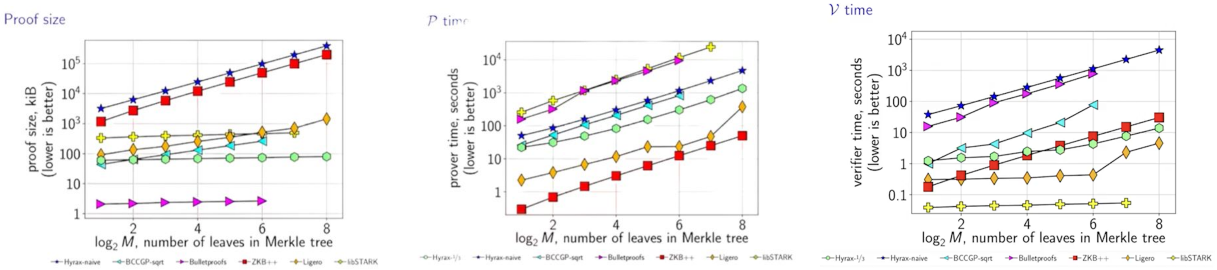


```
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_
SHA256_4.pws              SHA256_64_merkle_1_rdl.pws  SHA256_64_merkle_3_rdl.pws  SHA256_64_merkle_5_rdl.pws  SHA256_64_merkle_7_rdl.pws  SHA256_64.pws
SHA256_4_rdl.pws          SHA256_64_merkle_2_rdl.pws  SHA256_64_merkle_4_rdl.pws  SHA256_64_merkle_6_rdl.pws  SHA256_64_merkle_8_rdl.pws
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_4
SHA256_4.pws      SHA256_4_rdl.pws
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_4.pws
Proof size: 1841 elems, 49555 bytes
Verification succeeded.
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_4_rdl.pws
Proof size: 245 elems, 7073 bytes
Verification succeeded.
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_64
SHA256_64_merkle_1_rdl.pws  SHA256_64_merkle_3_rdl.pws  SHA256_64_merkle_5_rdl.pws  SHA256_64_merkle_7_rdl.pws  SHA256_64.pws
SHA256_64_merkle_2_rdl.pws  SHA256_64_merkle_4_rdl.pws  SHA256_64_merkle_6_rdl.pws  SHA256_64_merkle_8_rdl.pws
```



```
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_64_merkle_2_rdl.pws
Proof size: 277 elems, 7920 bytes
Verification succeeded.
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_64_merkle_1_rdl.pws
Proof size: 261 elems, 7505 bytes
Verification succeeded.
aditya@aditya-xps:~/Dropbox (ASU)/bhavani-ac/hyraxZK/fennel$ ./run_fennel.py -p ../pws/experiments/SHA256/SHA256_64_merkle_2_rdl.pws
Proof size: 277 elems, 7902 bytes
Verification succeeded.
```

We have given a selection of the graphs that were created to evaluate the implementation. The graphs provide a cost comparison between the two Hyrax variations addressed in the study and the other baseline systems presently available. By analyzing the graphs we were able to verify the results claimed by the Hyrax paper and we can see it performs better than 3 out of 5 the baseline protocols used in the comparison.



From the three graphs given above, we can see that Hyrax has much larger proofs than Bulletproofs and libSTARK, but smaller than BCCGP-sqrt, both asymptotically and concretely. Also, Hyrax requires a lesser number of cryptographic operations than BCCGP-sqrt and Bulletproofs and hence lesser Prover time. However, the Prover time for Hyrax is higher than ZKB++ or Ligero because those systems do not use any public-key cryptography. Hyrax's Verfier time is much lesser than Bulletproofs and BCCGP-sqrt. libSTARK's Verifier time has the best asymptotics among all systems and extremely low concrete costs. For this class of issue sizes, Ligero's V has sublinear scalability and a practically rapid verification time because of the amortization of its bottleneck calculation over several SHA-256 instances.

5. **Conclusion**

We have evaluated the new zk-SNARK protocols that have brought the theoretical concepts like “MPC-in-the-head” to real world scenarios and achieved no-trusted party setup. The protocols achieve sublinear complexity in the size of the verifier. These protocols also claim to be post-quantum secure which is a feature that researchers would need to focus on in the coming years due to the extensive research and development happening in quantum computing. We also implemented HyraxZK and compared it with other baseline protocols and verified the results claimed by the paper. In addition to these we also reviewed current applications where these protocols are employed. In cases of IoT infrastructures that would comprise of resource-constrained edge devices, a better version of ZKP with smaller proofs and faster performance would be indispensable. In the field of financial applications, the existing protocols were built on assumptions of avoiding the presence of a trusted third party, making way for newer ZKP protocols without trusted setup an ideal fit. We also reviewed the application in the domain of AI pipelines where the aim was to protect privacy of the users, data and the models used and certifying the fairness and building trust between the constituent parties. The main bottleneck for all these applications was the computation cost of the zero knowledge proofs used. After weighing the pros and disadvantages of each of these use cases, we conclude that developments such as shooter-proofs, improved implementations resulting in faster performance, and no-trusted setups will pave the way for more secure applications with comparatively lower latencies for end users.

6. **Description of My Contribution and Acknowledgements:**

We are a group of 3 students and we divided our work equally among us keeping the course workload in our mind. We reviewed the recently published papers on Zero-Knowledge Proofs to know about the on-going trends in the fields and have a clear understanding of what we needed to focus on during our project. We came across many protocols and applications out of which we selected 8 most recent and influential papers. I reviewed and gave my contributions in the report about 3 papers Ligero[6], Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures[9] and Zero Knowledge Proofs Towards Verifiable

Decentralized AI Pipelines[1]. I chose these because I felt they made great contributions in the advancement of ZKPs and using them to solve real world problems. I also made sure I brought my teammates up to speed on the concepts I understood from reviewing the above selected papers. I also helped my teammates with the implementation of HyraxZK[15] and understood how the project works. I was able to pick up easily because [6], [9] have contributions in the same line of work. Moreover, HyraxZK compares its results with Ligero[6] so it was particularly interesting for me. In the end, I am very thankful to my teammates Krishna and Bhavani who proactively helped me understand their work and finished their own tasks. They helped me understand concepts from their papers on IOT and Blockchain that employed zero-knowledge proofs.

References

[1] "Zero Knowledge Proofs Towards Verifiable Decentralized AI Pipelines," International Conference on Financial Cryptography and Data Security, 2022.

[2] "LiteZKP: Lightening Zero-Knowledge Proof-Based Blockchains for IoT and Edge Platforms," *IEEE Systems Journal,* 2022.

[3] "Zero-Knowledge Contingent Payments Revisited: Attacks and Payments for Services," Cryptology ePrint Archive, Paper 2017/566.

[4] "Demonstration of Blockchain-based IoT Devices Anonymous Access Network Using Zero-knowledge Proof," 2020 International Wireless Communications and Mobile Computing (IWCMC).

[5] "CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications SecurityOctober 2017 Pages 2087–2104https://doi.org/10.1145/3133956.3134104"

[6] "Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. 2017. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 2087–2104. https://doi.org/10.1145/3133956.3134104"

[7] Eli Ben-Sasson, Matan Hamilis, Mark Silberstein, and Eran Tromer. 2016. Fast Multiplication in Binary Fields on GPUs via Register Cache. In Proceedings of the 2016 International Conference on Supercomputing, ICS 2016, Istanbul, Turkey, June 1-3, 2016. 35:1–35:12.

[8] Shuhong Gao and Todd Mateer. 2010. Additive Fast Fourier Transforms over Finite Fields. IEEE Trans. Inf. Theor. 56, 12 (Dec. 2010), 6265–6272.

[9] CCS '18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications SecurityOctober 2018 Pages 525–537https://doi.org/10.1145/3243734.3243805

[10] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. 2018. Improved Non- Interactive Zero Knowledge with Applications, to Post-Quantum Signatures. In 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), October 15–19, 2018, Toronto, ON, Canada. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3243734.3243805

[11] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasub- ramaniam. 2017. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In ACM CCS 17: 24th Conference on Computer and Communications Security. ACM Press, 2087–2104.

[12] MelissaChase,DavidDerler,StevenGoldfeder,ClaudioOrlandi,SebastianRa- macher, Christian Rechberger, and Greg Zaverucha. 2017. Post- Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In ACM CCS 17: 24th Conference on Computer and Communications Security. ACM Press, 1825–1842.

[13] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. 2016. ZKBoo: Faster Zero-Knowledge for Boolean Circuits. In USENIX Security Symposium.

[14] Samuel Ranellucci, Alain Tapp, and Rasmus Winther Zakarias. 2016. Efficient Generic Zero-Knowledge Proofs from Commitments. In ICITS 16: 9th International Conference on Information Theoretic Security (Lecture Notes in Computer Science). Springer, Heidelberg, 190–212.

[15] R. Wahby, I. Tzialla, A. Shelat, J. Thaler and M. Walfish, "Doubly-Efficient zkSNARKs Without Trusted Setup," in 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2018 pp. 926-943.

doi: 10.1109/SP.2018.00060

[16] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway. Everything provable is provable in zero-knowledge. In CRYPTO, Aug. 1990.

[17] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Efficient range proofs for confidential transactions. In IEEE S&P, May 2018.

[18] R. Cramer and I. Damgård. Zero-knowledge proofs for finite field arithmetic, or: Can zero-knowledge be for free? In CRYPTO, Aug. 1998

[19] J. Groth. Linear algebra with sub-linear zero-knowledge arguments. In CRYPTO, Aug. 2009.

[20] Singh, N., Dayama, P., Pandit, V. (2022). Zero Knowledge Proofs Towards Verifiable Decentralized AI Pipelines. In: Eyal, I., Garay, J. (eds) Financial Cryptography and Data Security. FC 2022. Lecture Notes in Computer Science, vol 13411. Springer, Cham. https://doi.org/10.1007/978-3-031-18283-9_12

[21] Veeningen, M. (2017). Pinocchio-Based Adaptive zk-SNARKs and Secure/Correct Adaptive Function Evaluation. In: Joye, M., Nitaj, A. (eds) Progress in Cryptology - AFRICACRYPT 2017. AFRICACRYPT 2017. Lecture Notes in Computer Science(), vol 10239. Springer, Cham. https://doi.org/10.1007/978-3-319-57339-7_2

[22] Zhang, J., Fang, Z., Zhang, Y., Song, D.: Zero knowledge proofs for decision tree predictions and accuracy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 2039–2053 (2020)

[23] A. Rasheed, R. N. Mahapatra, C. Varol and K. Narashimha, "Exploiting Zero Knowledge Proof and Blockchains Towards the Enforcement of Anonymity, Data Integrity and Privacy (ADIP) in the IoT," in IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 3, pp. 1476-1491, 1 July-Sept. 2022, doi: 10.1109/TETC.2021.3099701.

[24] A. A. Rasheed, R. N. Mahapatra, and F. G. Hamza-Lup, “Adaptive group- based zero knowledge proof-authentication protocol in vehicular ad hoc networks,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 2, pp. 867–881, Feb. 2020.

[25] A. Fiat and A. Shamir, “How to prove yourself: Practical solutions to identification and signature problems,” in Advances in Cryptol. — CRYPTO’ 86, A. M. Odlyzko, Ed. Berlin, Germany: Springer, 1986, pp. 186–194.

[26] A. Rasheed, R. R. Hashemi, A. Bagabas, J. Young, C. Badri, and K. Patel, “Configurable anonymous authentication schemes for the Internet of Things (IoT),” in Proc. IEEE Int. Conf. RFID, 2019, pp. 1–8.

[27] H. Tohidi and V. T. Vakili, “Lightweight authentication scheme for smart grid using merkle hash tree and lossless compression hybrid method,” IET Commun, vol. 12, no. 19, pp. 2478–2484, Oct. 2018, doi: 10.1049/iet-com.2018.5698.G.

[28] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, “A survey on privacy protection in blockchain system,” J. Netw. Comput. Appl., vol. 126, pp. 45–58, 2019.

[29] M. Walshe, G. Epiphaniou, H. Al-Khateeb, M. Hammoudeh, V. Katos, and A. Dehghantanha, “Non-interactive zero knowledge proofs for the authentication of IoT devices in reduced connectivity environments,” Ad Hoc Netw., vol. 95, 2019, Art. no. 101988.

[30] D. Gabay, K. Akkaya, and M. Cebe, “Privacy-preserving authentication scheme for connected electric vehicles using blockchain and zero knowl- edge proofs,” IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 5760–5772, Jun. 2020.