Computers and Electrical Engineering 103 (2022) 108290

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Privacy-preserving ledger for blockchain and Internet of Things-enabled cyber-physical systems☆

Rajani Singh [a], Ashutosh Dhar Dwivedi [a,*], Raghava Rao Mukkamala [a,b], Waleed S. Alnumay [c]

[a] *Centre for Business Data Analytics, Department of Digitalization, Copenhagen Business School, Frederiksberg, Denmark*
[b] *Department of Technology, Kristiania University College, Oslo, Norway*
[c] *Riyadh Community College, Computer Science Department, King Saud University, Riyadh, Saudi Arabia*

ARTICLE INFO

Keywords:
Blockchain
Distributed Ledger Technologies
Zero knowledge proofs
Privacy-preserving
Cybersecurity

ABSTRACT

In recent years, decentralized applications such as Distributed Ledger Technologies and blockchain have evolved as suitable applications for secure sharing of information in a decentralized fashion using privacy preserving techniques like zero-knowledge protocols. However, the biggest issue with the traditional zero-knowledge protocols on a blockchain ledger is their slow performance on big data. This paper presents the advance zero-knowledge ledger by replacing their range-proof technique with the most efficient range-proof technique based on the improved inner product based zero-knowledge proofs. Moreover, this technique allows the aggregation of multiple range-proofs into a single range-proof, which makes the current zero-knowledge ledger system more efficient than the existing one.

1. Introduction

The first notion of the Zero-Knowledge Proof (ZKP) technique was originally proposed by Goldwass et al. [1], as a cryptographic mechanism and a theorem-proving procedure using interactive proof systems that limit the amount of information that is required for a sender to be able to prove knowledge about particular data. A ZKP allows the holder of data to reveal knowledge (e.g., an attribute) about the data by providing a suitable proof that the knowledge about the data is correct (e.g., the value of the attribute is within a certain range), but without the need to reveal the data itself. The most important utility of ZKP concept is that a prover can prove the correctness of an assertion or a fact to the verifier without leaking any extra information [2]. Even though the ZKP technique was initially developed with the application purpose to prove the correctness of the cryptographic protocols in a modular way, the ZKP technique received a lot of attention especially after the evolution of Decentralized Ledger Technologies (DLT) and Blockchain as a way of privacy-preserving technique to share knowledge in a secure way. The notations used throughout the paper are given in the Table 1.

In order to illustrate the core concept of ZKP, we offer a simple analogy [3] that uses a thought experiment. Consider two participants: Alice and Bob. Let us assume that Alice (the verifier) is a colorblind person who cannot distinguish between red and blue. Bob (the prover) would like to prove to Alice that the two balls that he is presenting to her, belong to two different colors. Note that Alice cannot difference the colors, which is the same as Alice not seeing the colors (the data), but she is interested in

☆ This paper is for special section VSI-biot. Reviews were processed by Guest Editor Dr. Uttam Ghosh and recommended for publication.

* Corresponding author.

E-mail addresses: rs.digi@cbs.dk (R. Singh), add.digi@cbs.dk, ashudhar7@gmail.com (A.D. Dwivedi), rrm.digi@cbs.dk (R.R. Mukkamala), wnumay@ksu.edu.sa (W.S. Alnumay).

https://doi.org/10.1016/j.compeleceng.2022.108290
Received 27 April 2022; Received in revised form 25 July 2022; Accepted 1 August 2022
Available online 17 August 2022

Table 1
Notations used throughout the paper and their description.

Symbol	Description
p, q	Prime number
$\mathbb{G}$	Cyclic group of prime order
$\mathbb{Z}_p$	Ring of integers modulo p
$\mathbb{G}^n$	Vector space of dimension n over $\mathbb{G}$
$\mathbb{Z}_p^n$	Vector space of dimension n over $\mathbb{Z}_p$
$\mathbb{Z}_p^*$	$\mathbb{Z}_p$ without 0
g, h, u	Generators of $\mathbb{G}$
$\mathcal{H} : \mathbb{Z}_p^{2n+1} \rightarrow \mathbb{G}$	Hash function
α	Blinding factor in the commitment
$x, y, z \in \mathbb{Z}_p^*$	Random challenge
$\vec{a} = (a_0, a_1, \dots, a_n) \in \mathbb{F}^n$	Vector with scalar elements $a_i \in \mathbb{F}$
$\overrightarrow{La} = (a_1, \dots a_k)$	Left slice of vector $\vec{a}$ for $0 \leq k \leq n$
$\overrightarrow{Ra} = (a_{k+1}, \dots, a_n)$	Right slice of vector $\vec{a}$ for $0 \leq k \leq n$
$\vec{e}^n = (1, e, e^2, \dots, e^{n-1})$	Vector containing the first n powers of k for $k \in \mathbb{Z}_p^*$
$\vec{p}(X) \in \mathbb{Z}_p^n[X]$	Vector Polynomials with vector coefficients
$\mathbf{A} \in \mathbb{F}^{m \times n}$	Matrix with m rows and n column
$a_{i,j}$	Elements of matrix $\mathbf{A}$
$\vec{a} \circ \vec{b} = (a_0 b_0, a_1 b_1, \dots a_n b_n)$	Hadamard Product of vectors $\vec{a}$ and $\vec{b}$
$v := \langle \vec{a}, \vec{b} \rangle = \sum_{i=0}^{n} a_i \cdot b_i$	Inner product of vectors $\vec{a}$ and $\vec{b}$
$\langle \vec{l}(X), \vec{r}(X) \rangle = \sum_{i=0}^{n} \sum_{j=0}^{i} \langle \vec{l_i}, \vec{r_j} \rangle \cdot X^{i+j}$	Inner product of vector polynomials $\vec{l}(X)$ and $\vec{r}(X)$
$C := \vec{g}^{\vec{a}} = \prod_{i=1}^{n} g_i^{a_i}$	Binding but not hiding commitment to vector $\vec{a}$

knowing if the colors are different from each other (the information). In this context, a ZKP mechanism by which Bob can prove to Alice that both balls are of the same color, works as follows: Alice puts both balls behind her back, such that she can switch them between her hands without Bob noticing. Then, she brings both balls forward and shows them to Bob, such that Bob knows which hand holds the red ball and which hand holds the blue ball. Alice will at this stage hide the balls again behind her back, which gives her the opportunity, but not the obligation, to switch them between hands. After changing (or not changing) the balls from hand, Alice will show them to Bob again and ask him if she switched the balls between her hands. If both balls are of the same color, Bob is forced to guess, whereas if both balls have different colors, Bob will always know what Alice did behind her back and, therefore always rightfully guess. After performing the experiment for several times and observing the constant rightfulness of Bob's answers, Alice comes to the conclusion that the balls belong to two different colors. As a result of this experiment, Alice knows that both balls are of different colors (i.e., the information), but has never seen the colors (i.e., the data) of the balls. Let us now assume that Bob is dishonest and tries to convince Alice that both balls belong to different colors, whereas in reality, both balls are of the same color (either red or blue). In doing so, Bob will have to randomly guess if Alice changed the balls behind her back. This gives Bob a 50% probability of successfully guessing each time. If Alice performs these experiments several (n) times, the probability of Bob correctly guessing in each run will decrease exponentially (such that the success probability of correctly guessing all the consecutive trials will be equal to 0.5^n). Hence, if Alice performs this experiment 10 times, then the probability of Bob guessing correctly in each run will be reduced to $0.5^{10} = 0.00098$. In this case, Bob will not be able to demonstrate that both balls are of the same color (since they are not). In this case, Alice has not seen the color of the balls (i.e., the data) but has gained information about them. Similar to this analogy, ZKP is an extremely powerful technique that allows sharing essential knowledge about information without actually revealing the information itself, but at the same time providing proof that the knowledge about information is correct. This technique can be quite instrumental in privately sharing personal data-driven information (without the need for sharing data) at an individual level and in a trustworthy manner.

The rest of the paper is as follows. In Section 1.1, we will introduce to basics of Zero-knowledge Ledger and then we will describe related research in the area of ZKP in Section 2. Preliminary concepts will be introduced in Section 3. Sections 4 and 5 described the Range proof used in zkLedger and range proof using bulletproof where as our proposed range-proof technique is described in Section 6 and finally we conclude in Section 7.

1.1. Zero-knowledge ledger

Zero-knowledge ledger (zkLedger) is the first zero-knowledge proof [1] based distributed ledger system introduced by Narula et al. [4] for the bank auditing purpose. zkLedger provides strong transaction privacy, public verifiability, complete, fast and provably correct auditing. The ledger has a table-like structure and consists of rows and columns. Each column in the table represents a participant of the system, and each row in the table represents a transaction among the participants. A transaction entry consists of some payment amount for each and every participant whether the participant was involved or not in that transaction. If the participant was not involved in any particular transaction, then the row entry value in the table for that participant will be 0. Every participant keeps the commitment caches, which are rolling products of every participants' column in the ledger. zkLedger has the following properties:

1. zkLedger is built for non-trusted setup, meaning that the participants do not trust each other. So, in the zkLedger system, participants can be either honest or dishonest or malicious.
2. It uses the Schnorr-type non-interactive zero-knowledge (NIZK) proofs which substantially reduces the cost of message communication among the system participants.
3. It guarantees strong transaction privacy by hiding the transaction amount, sender and receiver information, transaction graph, or linkages between transactions. Only the timestamp or time of transactions and the type of asset being transferred are known publicly. To hide the secret value or information, zkLedger uses Pedersen commitments scheme which can be homomorphically combined.
4. It guarantees completeness, meaning that no participant can hide the transactions from the verifier or auditor. This is possible due to the row-column construction of the table. Moreover, participants use rolling caches to produce and verify answers to the queries quickly.
5. It allows a wide variety of auditing queries such as sums, moving averages, ratios, standard deviations and variances. To compute measurements beyond sums like variance, skew, and outliers, it uses an interactive map or reduces paradigm over the ledger with NIZK proofs.
6. zkLedger based system is fast. For a ledger with 100,000 transactions, it produces provably correct answers to the verifier or auditor queries in less than 10 milliseconds. However, its efficiency can further be improved.
7. Since zero-knowledge proofs in zkLedger is defined on an elliptic curve over a prime field which is a cyclic group of prime order and uses modular arithmetic. It is required from the prover side to provide a range-proof that all the commitment values are in this cyclic group.

Range-proof is a zero-knowledge proof in which a prover proves to a verifier that a number say v lies in a certain range or interval say (A, B) without revealing the number itself. So v is hidden from the verifier, called a secret value while the range (A, B) is known to both prover and verifier. Consider a simple example of a bank loan. To apply for a bank loan, an applicant need to show that his salary satisfies the minimum criteria meaning that is above or at least equal to the minimum salary band. With zero-knowledge range-proof, an applicant can prove to the bank that his salary is good enough to apply for a bank loan while keeping his salary secret. Apart from improving the zkLedger performance, our proposed zero knowledge range proof is efficient and has wide variety of application. Some of them are listed below.

- Membership proof of a club or group.
- Banking to prove that the customer belongs to a certain range group and has a salary in a certain range group which helps in approving the loan.
- Healthcare to prove the patient’s reading is in certain range.
- Supply chain to prove that the temperature based medicines or products are in a certain range throughout the supply chain process.
- Know Your Customer (KYC) to validate that a specific piece of private information belongs to a certain range.
- Electronic voting.
- Electronic auction.

2. Related work

First of all, Wu et al. [5] provided a detailed survey on how zero-knowledge proofs gradually improved in the last 20 years. Authors studied the basic principles, application and efficiency improvement of the non-interactive zero-knowledge proof system. They summarize the research progress achieved by the non-interactive zero-knowledge proof system on the following aspects: a non-interactive zero-knowledge proof system of NP problems, the definition and related models of the non-interactive zero-knowledge proof system, non-interactive statistical and perfect zero-knowledge, the connection between interactive zero-knowledge proof system, non-interactive zero-knowledge proof system.

One of the first research works that adopted ZKP technique in DLT and blockchain is Zerocoin. Miers et al. [6] proposed cryptographic extension to Bitcoin named Zerocoin. The protocol used in Zerocoin allows for fully anonymous currency transactions without requiring a trusted setup. Further, they explained Zerocoin’s cryptographic construction, its integration into Bitcoin, and examine its performance both in terms of computation and impact on the Bitcoin protocol. Some of the recent work on privacy issue can be found in Iwendi et al. [7] and Patel et al. [8] and others [9–11]. Furthermore, Boudot et al. [12] provided the efficient (less than 20 exponentiation to perform and less than 2 Kilobytes to transmit) and exactly zero knowledge range-proofs to show that a committed number belongs to an interval without revealing the number itself. Their proofs has potential application in different areas such as electronic cash, group signatures, publicly verifiable secret encryption, etc.

Similarly, Camenisch et al. [13] proposed two different way of building set-membership zero-knowledge proofs. The first membership proof is based on bilinear group assumptions while, the second is based on a strong RSA assumption. Depending on the application, for example, when membership set is a published set of values such as frequent flyer clubs, cities, etc., these alternative proofs provides privacy-preserving solutions.

In terms of range-proof techniques, Peng et al. [14] also proposed a range-proof technique that needs a constant cost and is more efficient than the Boudot et al. [12] range-proof schemes. Therefore their technique improves the efficiency of range-proof without further compromising security. Koens et al. [15] proposed a more efficient zero-knowledge range-proof and compared it with the

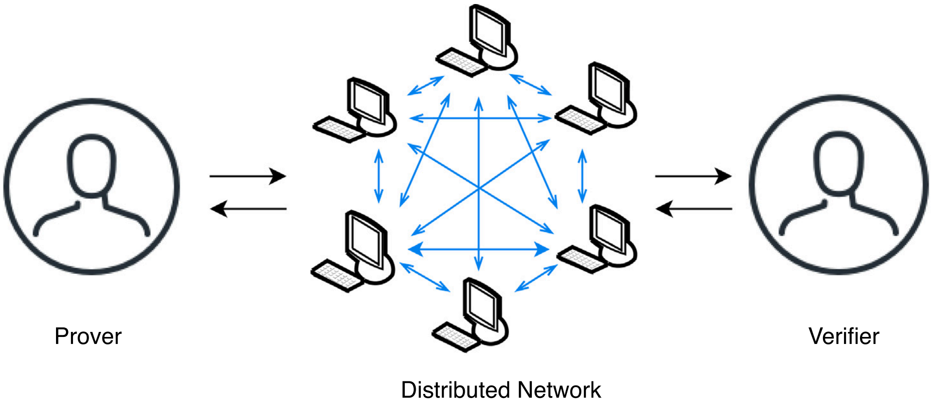

Fig. 1. Zero knowledge for distributed systems.

current zero-knowledge range-proof used in Ethereum. Surprisingly, their zero-knowledge range-proof is 10 times more efficient than the current zero-knowledge range-proof used in Ethereum. They modified the work of Peng [14], by making it non-interactive. Some of the recent works on Blockchain can found at [16].

Based on the Fiat–Shamir heuristic transformation [17], Yuen et al. [18] proposed the efficient (reduced proof with only constant size) non-interactive range-proof. Chaabouni et al. [19] showed that the range-proof proposed by Yuen et al. [18] is insecure. Moreover, they build a secure non-interactive range-proof. Their proofs required either very short communication or very efficient prover's computation. Bünz–Bootle et al. [20] proposed a new non-interactive zero-knowledge proof protocol named Bulletproofs, with very short proofs. Moreover, the proofs do not require a trusted setup. Their zero-knowledge proofs are based on improved inner product space. The efficiency of Bulletproofs is very well suited for the distributed and trustless nature of blockchains. Thus it has applications in the area of cryptocurrencies such as confidential transaction [21], non-interactive zero-knowledge proofs for Smart Contracts like Zether [20] etc. Finally, Morais et al. [2] provided an extensive survey on the range-proofs by explaining and comparing the three different schemes given by: Boudot et al. [12], Camenisch et al. [13] and Bünz–Bootle et al. [20]. Some of the recent works on Blockchain and privacy preserving techniques can be found at [22–24].

2.1. Our contribution

Range-proof in zkLedger is used to verify that the certain asset value lies in a predefined range, moreover, it helps to prevent the system from cheating user to create a new asset into the system which is non-detectable by the system users. However, the current zkLedger suffers from the slower transaction creation and validation process because of the range-proof scheme chosen from the confidential asset paper by Poelstra et al. [21] that uses the Borromean ring signature technique. These range-proofs are 10 times larger than the size of other zero-knowledge proof used and are computationally expensive which are the major drawback of the system, For example, it takes 5 times more than the other zero-knowledge proofs to prove and validate. So, it is important to address this issue in order to make the system efficient. To address this important issue, we use the range-proof technique as described in Bulletproofs paper by Bünz et al. which drastically reduce the proving and verification size. Moreover, this range-proof technique allows to aggregate multiple-proof into a single range-proof which has a significant advantage and can further improve the system performance. For example, to prove that a number is in range $(0, 2^{10})$ and if there are 20 such range-proofs then the zkLedger takes $(0.63 \cdot m \cdot n) = 126$ elements from cyclic group $\mathbb{G}$ which is an elliptic curve over prime field and $(1.26 \cdot m \cdot n + 1) = 153$ elements from the group of prime integer $\mathbb{Z}_p$. While, Bulletproof only take $2(\log_2\ m + \log_2 n) + 4 = 19.2876$ elements from group $\mathbb{G}$ and 5 elements from the group $\mathbb{Z}_p$. Bulletproofs techniques itself inspired by the work of Bootle et al. [25].

3. Preliminaries

The most challenging issue with financial systems is to share information about data and transactions while keeping sensitive information confidential and secure. Nowadays, blockchain is used as a decentralized and distributed system where information such as transaction details etc. are stored in public distributed ledger transparently, at the same time with suitable privacy mechanisms build into it using asymmetric cryptography and other cryptographic mechanisms. Similarly, IoT devices are much popular nowadays that are connected with blockchain and produce a lot of data. These devices are resource-constrained and are mainly used in smart applications such as smart home, smart city etc. Privacy is a major issue with IoT and blockchain ledger where Zero-Knowledge can play an important role (see Figs. 1 and 2). For example, Zero-knowledge allows any prover to prove that he/she is the owner of any data, and he/she can access that data without revealing his/her identity. This is similar to prove that Alice (Prover) is the account holder of a bank account without revealing her name to Bob (Verifier). The zero-knowledge protocol has three main phases namely: *commitment, challenge and verify:*

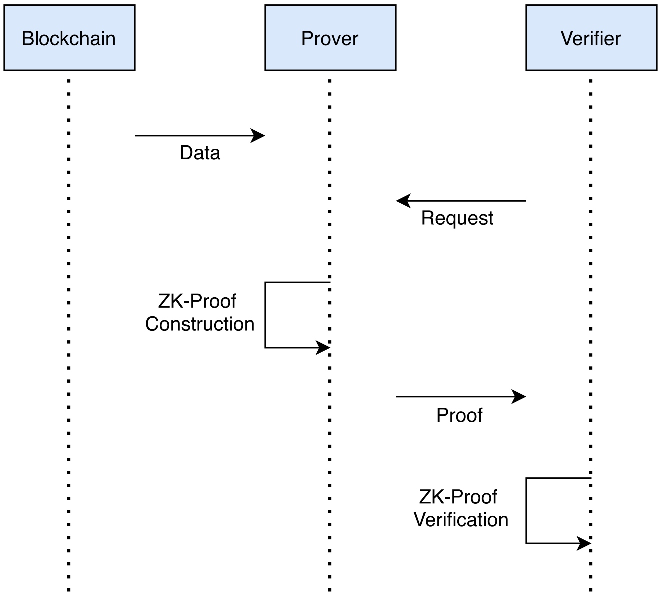

Fig. 2. Zero-knowledge based ledger.

1 **Commitment:** During the commitment phase, prover commits the secret to the verifier, and it is just an encrypted value that ensures verifier that later prover cannot change the secret value. Therefore, Alice (Prover) generate the commitment and send it to Bob (verifier).
2 **Challenge:** When once Bob receives the commitment, he sends a query to Alice, and this query is generally in the form of checkpoint called a challenge. For example, if Alice declares that she knows the solution of a Sudoku puzzle, then in such case, Bob can ask for the sum of the first row.
3 **Verify:** The last face is verification where Bob verifies the solution sent by Alice. By verifying the solution, Bob can now make an assurance that Alice is not cheating.

A probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ is a probabilistic interactive Turing Machine that runs in polynomial time in the security parameter λ. Throughout the paper, $\overset{\$}{\longleftarrow}$ symbol represents that the element is chosen uniformly at random from either the group or space. Now we define the discrete log assumption which is required to ensure the provable security in our zero knowledge range proof. Next we define the commitment scheme that is used by the prover to convince the verifier. This is one of the important tool to write any zero knowledge protocol.

Definition 1 (***Discrete Log Assumption***).

Let $(\mathbb{G}, \cdot)$ be an abelian group and g, h be generator of the group. Given $g, h \in \mathbb{G}$, Discrete Logarithm Problem (DLP) is defined by Eq. (1)

$$\text{Find } x \in \{0, 1, \dots, q-1\}, \text{ so that } h = g^x \tag{1}$$

Note that if $\mathbb{G}$ is a cyclic group of prime order q with generator g, solving DLP (1) is computationally hard or considered as infeasible. Moreover, the order q of the group used is implicitly dependent on the security parameter λ, which guarantees that the DLP in these groups is intractable for PPT adversaries $\mathcal{A}$. So we will leave the λ notation when it is implicit.

The commitment scheme defined in Definition 3 has some properties as specified below.

Definition 2 (***Binding Commitment Scheme***).

Consider public parameter $pp \in \mathcal{PP}$, message $m, m' \in \mathcal{M}$ and randomness $r, r' \in \mathcal{R}$. A commitment scheme is binding if for all PPT adversaries $\mathcal{A}$, the probability of $\mathcal{A}$ generating Commit(pp, m', r') with $m \neq m'$ such that Commit(pp, m', r') = Commit(pp, m, r) is negligible. Commitment scheme is *perfectly binding* if this probability is 0.

Definition 3 (***Commitment Scheme***).

A commitment scheme consists of two probabilistic polynomial-time algorithms (as shown in algorithm 1): Setup and Commit.

Algorithm 1:

```
1. Setup : $\cdot \rightarrow \mathcal{PP}$
   INPUT: Security parameter $\lambda$;
   COMPUTE: $pp = \mathrm{Setup}(1^{\lambda})$
   OUTPUT: Public parameter $pp$ in parameter space $\mathcal{PP}$
2. Commit : $\mathcal{PP} \times \mathcal{M} \times \mathcal{R} \rightarrow C$
   INPUT: A message $m$ from message space $\mathcal{M}$ i.e., $m \overset{\$}{\longleftarrow} \mathcal{M}$,
   A random number $r$ draws uniformly from randomness space $\mathcal{R}$ i.e., $r \overset{\$}{\longleftarrow} \mathcal{R}$,
   COMPUTE: $c = \mathrm{Commit}(pp, m, r)$
   OUTPUT: Committed value $c$ in a commitment space $C$;
```

Definition 4 (***Hiding Commitment Scheme*****).**

Consider public parameter $pp \in \mathcal{PP}$, message $m, m' \in \mathcal{M}$ with $m \neq m'$ and randomness $r \in \mathcal{R}$. A commitment scheme is hiding if the probability distributions $D = \{c : \ c = \mathrm{Commit}(pp, m, r)\}$ and $D' = \{c : \ c = \mathrm{Commit}(pp, m', r)\}$ are computationally indistinguishable. Commitment scheme is *perfectly hiding* if both distributions are equal.

Definition 5 (*Homomorphic Commitment*). Consider public parameter $pp \in \mathcal{PP}$, message $m, m' \in \mathcal{M}$ with $m \neq m'$, randomness $r, r' \in \mathcal{R}$ and two distinct commitments $c = \mathrm{Commit}(pp, m, r)$ and $c' = \mathrm{Commit}(pp, m', r')$. A commitment $\bar{c}$ is called homomorphic if $\bar{c} = c + c'$ i.e., $\mathrm{Commit}(pp, m + m', r + r')$ equals to $\mathrm{Commit}(pp, m, r) + \mathrm{Commit}(pp, m', r')$. Now, we define an specific commitment scheme that is based on discrete log assumption.

Definition 6 (*Pedersen Commitment*).

The Pedersen Commitment scheme consists of two probabilistic polynomial-time algorithms: Setup and Commit as shown in algorithm 2.

Algorithm 2:

```
1. Setup :
   Take $\mathcal{M} = \mathcal{R} = \mathbb{Z}_q$ for $q$ to be prime number.
   Take $C$ to be an isomorphic elliptic curve group.
   Choose two generators $g, h \overset{\$}{\longleftarrow} \mathbb{Z}_q$ such that no one knows $\log_g h$
2. Commit : $\mathcal{PP} \times \mathbb{Z}_q \times \mathbb{Z}_q \rightarrow C$
   INPUT: $m \overset{\$}{\longleftarrow} \mathbb{Z}_q$ and $r \overset{\$}{\longleftarrow} \mathbb{Z}_q$
   COMPUTE: $\mathrm{Commit}(pp, m, r) = g^m h^r$
   OUTPUT: $c = \mathrm{Commit}(pp, m, r)$
```

We use the Pedersen commitment scheme defined in Definition 6 throughout the paper to develop the zero knowledge range proof. Our proposed version of zero knowledge proof uses the vector spaces, therefore we now define the Pedersen commitment scheme for the vectors which we call it as Pedersen vector commitment.

Definition 7 (*Pedersen Vector Commitment*).

Similar to the original Pedersen Commitment (Definition 6), our proposed Pedersen vector commitment scheme also contains two probabilistic polynomial-time algorithms: Setup and Commit as shown in algorithm 3.

Pedersen commitment and Pedersen vector commitment, both schemes are computationally binding and perfectly hiding. However, if we put $r = 0$ then both commitment scheme is binding but not hiding. Next we define the range proof for the homomorphic commitment scheme.

Definition 8 (*Range-proof*).

Consider a homomorphic commitment scheme with lower bound L and upper bound U that is $0 \leq L \leq U \leq q$. Range-proof of the interval or range $\mathbb{I} = [L, U]$ consists of two algorithms as shown in algorithm 4.

Algorithm 3:

1. **Setup** :
   Take $\mathcal{M} = \mathbb{Z}_q^n$ and $\mathcal{R} = \mathbb{Z}_q$ for q to be prime number.
   Take $\mathcal{C}$ to be an isomorphic *elliptic curve group*.
   Choose $\vec{g} = (g_1, \dots, g_n) \overset{\$}{\longleftarrow} \mathbb{Z}_q^n$ and $h \overset{\$}{\longleftarrow} \mathbb{Z}_q$ such that no one knows $\log_{\vec{g}} h$
2. **Commit** : $\mathcal{PP} \times \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathcal{C}$
   INPUT: $\vec{m} = (m_1, \dots, m_n) \overset{\$}{\longleftarrow} \mathbb{Z}_q^n$ and $r \overset{\$}{\longleftarrow} \mathbb{Z}_q$
   COMPUTE: $\text{Commit}(pp, \vec{m}, r) = \vec{g}^{\vec{m}} h^r$
   OUTPUT: $c = \text{Commit}(pp, \vec{m}, r)$

Algorithm 4:

1. **Prove$_{\mathbb{I}}$** : $\mathcal{PP} \times \mathcal{M} \to \mathcal{C} \times \mathcal{M} \times \mathcal{I}$
   INPUT: $m \leftarrow \mathcal{M}$
   GENERATE: Commitment c to value m.
   OUTPUT: $c \in \mathcal{C}$ with opening information,
   An associated range-proof π from space of possible range-proofs $\mathcal{I}$.
2. **Verify$_{\mathbb{I}}$** : $\mathcal{PP} \times \mathcal{C} \times \mathcal{I} \to \{true, false\}$
   INPUT: $c \leftarrow \mathcal{C}$ and $\pi \leftarrow \mathcal{I}$
   OUTPUT: Accept if true and Reject if false.

Algorithm 5: zkLedger: Prover Algorithm

```
INPUT: secret integer $v$.
OUTPUT: Range proof $R = [c_0, (\text{Com}_0, s_0), \dots, (\text{Com}_{n-1}, s_{n-1})]$. Publish the range proof $R$.
Rewrite $v$ into base 2 form as $v = \sum_{i=0}^{n-1} 2^i \cdot b_i$ for $i = 0, 1, \dots, n-1$ do
    if $b_i = 0$ then
        For $\mathbb{Z}_p$ be a prime field and $g$ be a generator of a cyclic group
        Choose a random integer $r_i \in \mathbb{Z}_p$, Compute the commitment $P_i = g^{r_i}$,
    end
    if $b_i = 1$ then
        Choose random integer $\delta_i \in \mathbb{Z}_p$ and $r_i \in \mathbb{Z}_p$
        Compute the commitment $\text{Com}_i = h^{2^i \cdot b_i} + g^{\delta_i}$. Compute the challenge $c_i = \text{Hash}(g^{r_i})$.
        Compute $P_i = (\text{Com}_i)^{c_i}$.
    end
end
Calculate the initial challenge $c_0$ by simply concatenating the all $P_i$ and then taking hash: $c_0 = \text{Hash}(P_0 \parallel P_1 \parallel \dots \parallel P_{n-1})$.
for $i = 0, 1, \dots, n-1$ do
    if $b_i = 0$ then
        Choose a random integer $r'_i \in \mathbb{Z}_p$,
        Calculate the challenge $c'_i = \text{Hash}\left(r'_i + 2^i \cdot c_0 H\right)$
        Rewrite the commitment as $\text{Com}_i = \frac{P_i}{c'_i} \equiv \frac{r_i G}{c'_i}$
        Calculate a random point on a line along the secret bit as $s_i = r'_i + \frac{r_i c_0}{c'_i}$
    end
    if $b_i = 1$ then
        Choose random integer $s_i \in \mathbb{Z}_p$,
        Compute a random point on a line along the secret bit as $s_i = r_i + \delta_i c_0$.
    end
end
Calculate the sum of all the commitments as $\text{Com} = \sum_{i=0}^{n-1} \text{Com}_i$ and denote by $\text{C}_\text{m}$.
```

4. Range-Proof used in zkLedger

The range-proof consists of two algorithms, **Prove** and **Verify** as further explained in the following paragraphs. The Prover has a secret number v and a random integer δ as his input. He wants to prove that v lies in the interval $(0, 2^{64})$ without revealing the actual value of v. For brevity of notation, we denote the bit form of v as b_i. For example, if $v = 7$, then its bit form is 0111. So, the prover will follow the algorithm 5 as explained below to generate a range proof for his secret number v. The zkLedger uses the non-interactive range-proof technique used in Confidential Asset paper by Poelstra et al. [21]. The authors used the bit decomposition method and the Borromean ring signature-based OR proofs. Moreover, this range-proof can be used to prove that a secret number v lies in between a range $(0, k^n)$ without revealing the v. Here k represents the base in which we want to write the number v, while n represents the length of base k. Although, the proof is given for any base, we only need the base $k = 2$ and length $n = 64$ in zkLedger.

After generating the range proof R, the sender will send it to the verifier. Since the verifier has the range proof R, he knows c_0, and a series of commitments with their random points on a line along the secret bit $(\text{Com}_i,\ s_i)$ for all i such that $i = 0, 1, \ldots n-1$. To verify this range proof, the verifier needs to follow steps defined in the algorithm 6 as explained below.

Algorithm 6: zkLedger: Verifier Algorithm

```
for $i = 0, 1, \ldots, n-1$ do
    Check if the challenge is calculated correctly by computing $c'_i = \text{Hash}\left(s_i G - c_0 \text{Com}_i - c_0 H 2^i\right)$
    Check if the $P_i$ is calculated correctly by computing $P_i = \text{Com}_i \cdot c'_i$
    end
Recomputes the initial value of the challenge as $\tilde{c}_0 = \text{Hash}(P_0 \parallel P_1 \parallel \ldots \parallel P_{n-1})$.
Recomputes the sum of all the commitments as $\overline{\text{Com}} = \sum_{i=0}^{n-1} \text{Com}_i$.
if $\tilde{c}_0 = c_0$ and $\overline{Com} = Com$ then
    accept the proof
else
    otherwise reject.
    end
```

Algorithm 7: Improved Inner Product: Prover Algorithm

```
if $n = 1$ then
    Compute the commitment value as $P := \text{Com}\left((\vec{a}, \vec{b}), 0\right) = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}} \cdot u^c$.
    Publish the vectors $\vec{a}, \vec{b}$ and commitment value Com to the verifier.
    end
if $n > 1$ then
    while $n > 1$ do
        Halves the length of each vector as: $n' = \frac{n}{2}$
        Define a Hash function $\mathcal{H}$ as $\mathcal{H}(\overrightarrow{La}, \overrightarrow{Ra}, \overrightarrow{Lb}, \overrightarrow{Rb}, c) = \overrightarrow{Lg}^{\overrightarrow{La}} \cdot \overrightarrow{Rg}^{\overrightarrow{Ra}} \cdot \overrightarrow{Lh}^{\overrightarrow{Lb}} \cdot \overrightarrow{Rh}^{\overrightarrow{Rb}} \cdot u^c$.
        Using the function from Eq. ??, calculate the values of $L\mathcal{H}, R\mathcal{H} \in \mathbb{G}$ as
        $L\mathcal{H} = \mathcal{H}\left(\vec{0}^{\frac{n}{2}}, \overrightarrow{La}, \overrightarrow{Rb}, \vec{0}^{\frac{n}{2}}, \langle \overrightarrow{La}, \overrightarrow{Rb} \rangle\right)$
        $R\mathcal{H} = \mathcal{H}\left(\overrightarrow{Ra}, \vec{0}^{\frac{n}{2}}, \vec{0}^{\frac{n}{2}}, \overrightarrow{Lb}, \langle \overrightarrow{Ra}, \overrightarrow{Lb} \rangle\right)$
        Given a random challenge $r \in \mathbb{Z}_p$, compute new vectors $\vec{a}', \vec{b}', \vec{g}'$ and $\vec{h}'$ as:
        $\vec{a}' = r \cdot \overrightarrow{La} + \frac{\overrightarrow{Ra}}{r}$
        $\vec{b}' = \frac{\overrightarrow{Lb}}{r} + r \cdot \overrightarrow{Rb}$
        $\vec{g}' = (\overrightarrow{Lg})^{r^{-1}} \circ (\overrightarrow{Rg})^r$
        $\vec{h}' = (\overrightarrow{Lh})^r \circ (\overrightarrow{Rh})^{r^{-1}}$
        Compute the new commitment value as $P' = (L\mathcal{H})^{r^2} \cdot \text{Com} \cdot (R\mathcal{H})^{r^{-2}}$.
        Publish $L\mathcal{H}, R\mathcal{H}, \vec{a}', \vec{b}', \vec{g}', \vec{h}'$ and P′, so the verifier will know all these values.
        end
    end
```

5. Range-proof from bulletproofs

In this subsection, we define the zero knowledge range proof that is based on the concept of inner product space of linear algebra. Note that, the Improved inner product proofs requires an additional step by proving an equivalent statement to the original inner product proofs. Although, proving the equivalent statement is equally complex to solve or prove, but it is relatively shorter in size.

Definition 9 (***Improved Inner Product Proofs***).

Prover convince the verifier that he knows two vectors $\vec{a}, \vec{b}$ such that

$$\text{Com} = \vec{g}^{\,\vec{a}} \cdot \vec{h}^{\,\vec{b}} \text{ and } v = \langle \vec{a}, \vec{b} \rangle \tag{2}$$

for vector $\vec{g}, \vec{h} \in \mathbb{G}^n$, commitment Com $\in \mathbb{G}$, inner product value $v \in \mathbb{Z}_p$, and vectors $\vec{a}, \vec{b} \in \mathbb{Z}_p^n$. Note that the commitment Com is blinding but not hiding vector commitment to vector $\vec{a}, \vec{b}$. An equivalent statement to (2), a prover can also prove that he knows two vectors $\vec{a}, \vec{b}$ such that

$$\text{Com} = \vec{g}^{\,\vec{a}} \cdot \vec{h}^{\,\vec{b}} \cdot u^v \text{ for } v = \langle \vec{a}, \vec{b} \rangle \tag{3}$$

To prove the statement in (3), the prover follows the steps defined in Algorithm 7. In order to verify the statement in (3), the verifier follows the steps shown in Algorithm 8. To develop an efficient range proof, we need a zero knowledge proof that is shorter in size. To do so, we use the zero knowledge proofs based on the vector polynomials. In order to achieve that, we first define the commitment to vector polynomials.

Algorithm 8: Improved Inner Product: Verifier Algorithm

```
if $n = 1$ then
    Check if the inner product value is correct by computing $c = \vec{a} \cdot \vec{b}$.
    Check if the commitment value is correct by recomputing $\overline{P} = \vec{g}^{\,\vec{a}} \cdot \vec{h}^{\,\vec{b}} \cdot u^c$.
    if $\overline{P} = P$ then
        Accept the proof
    else
        Reject the proof
        end
    end
if $n > 1$ then
    while $n > 1$ do
        Check if the new commitment value is correct by recomputing $\overline{P'} = (L\mathcal{H})^{r^2} \cdot P \cdot (R\mathcal{H})^{r^{-2}}$.
        Compute the new hash value by using new $\vec{g}', \vec{h}'$ as $\mathcal{H}' = \mathcal{H}\left(\frac{\vec{a}'}{r}, r\vec{a}', r\vec{b}', \frac{\vec{b}'}{r}, \langle \vec{a}', \vec{b}' \rangle\right)$.
        if $\overline{P'} = P' = \mathcal{H}'$ then
            Accept the proof
        else
            Reject the proof
            end
        end
    end
```

Algorithm 9: New Range proof: Verifier Algorithm

```
Check if $x = \text{Hash}\left(\text{Com}(t_1, \delta), \text{Com}(t_2, \rho)\right)$ holds
Check if $y = \text{Hash}\left(\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta), \text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma)\right)$ holds
Check if $z = \text{Hash}\left(\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta), \text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma), y\right)$ holds
Compute the new generator vector $\vec{h}'$ by computing its component values as: $h_i' = h_i^{y^{1-i}}$, for $i = 1, 2, \dots, n$.
For $f(y,z) = (z - z^2) \cdot \langle \vec{1}^n, \vec{y}^n \rangle - z^3 \cdot \langle \vec{1}^n, \vec{2}^n \rangle$ and $t_0 = f(y,z) + z^2 \cdot v$
Check if $\hat{t}(x) = t_0 + t_1 x + t_2 x^2$ holds by checking that the following holds:
$g^{\hat{t}(x)} \cdot h^{\tau} ? = \text{COM}^{z^2}(v, \alpha) \cdot g^{f(y,z)} \cdot \text{COM}^{x}(t_1, \delta) \cdot \text{COM}^{x^2}(t_2, \rho)$
Check the commitments to $\vec{l}(X)$ and $\vec{r}(X)$ by recomputing them as
$\text{Com}_{\text{new}} = \text{COM}(v, \alpha) \cdot \text{COM}^{x}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta) \cdot \vec{g}^{-z} \cdot \vec{h}'^{\,z \cdot \vec{y}^n + z^2 \cdot \vec{2}^n}$.
Check if $\text{Com}_{\text{new}} = h^{\mu} \cdot \vec{g}^{\,\vec{l}(x)} \cdot \vec{h}'^{\,\vec{r}(x)}$ holds.
Check if the inner product is calculated correctly by checking:
$\hat{t}(X) = \langle \vec{l}(X), \vec{r}(X) \rangle$.
If all the above equality holds then accept the proof, otherwise reject.
```

Definition 10 (***Commitment to Polynomials***).

Note that a polynomial $p(X) = p_0 + p_1 X + p_2 X^2 + \cdots + p_k X^k$ can also be represented as a column vector of their coefficients p_i in vector space,

$$p(X) := \sum_{i=0}^{k} p_i \cdot X^i = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \cdot \begin{bmatrix} 1 & X^1 & \dots & X^k \end{bmatrix} \tag{4}$$

In commitment to vector polynomials scheme, prover commit to a polynomial $p(X)$ by committing to each of its non-zero coefficients p_i for $i = 0, 1, \dots, k$. For this, he uses the Pedersen commitment scheme which has the homomorphic property. In the range-proof scheme, we have a quadratic polynomial so, $p(X) = p_0 + p_1 \cdot X + p_2 \cdot X^2$. Prover chooses δ, ρ uniformly at random from $\mathbb{Z}_p$ to calculate the commitments P_1 and P_2 to non-zero coefficients p_1 and p_2 as

$$P_1 := \mathrm{Com}(p_1, \delta) = g^{p_1} \cdot h^{\delta} \tag{5}$$

$$P_2 := \mathrm{Com}(p_2, \rho) = g^{p_2} \cdot h^{\rho} \tag{6}$$

Algorithm 10: New Range proof: Prover Algorithm

```
choose random $\alpha \in \mathbb{Z}_p$
compute $V := \mathrm{Com}(v, \alpha) = g^v h^\alpha$ for $V \in \mathbb{G}$
choose $\overrightarrow{Lv} \in \{0,1\}^n$ such that $\left\langle v_{bits}, \vec{2}^n \right\rangle = v$
define $\overrightarrow{Rv} = \overrightarrow{Lv} - \vec{1}^n$ such that $\overrightarrow{Lv} \circ \overrightarrow{Rv} = \vec{0}^n$ for $\overrightarrow{Rv} \in \mathbb{Z}_p^n$
choose random $\beta \in \mathbb{Z}_p$
compute $A := \mathrm{Com}\left((\overrightarrow{Lv}, \overrightarrow{Rv}), \beta\right) = \vec{g}^{\overrightarrow{Lv}} \cdot \vec{h}^{\overrightarrow{Rv}} \cdot h^\beta$ for $A \in \mathbb{G}$
choose random $\overrightarrow{Ls}, \overrightarrow{Rs} \in \mathbb{Z}_p^n$ and random $\gamma \in \mathbb{Z}_p$
compute $S := \mathrm{Com}\left((\overrightarrow{Ls}, \overrightarrow{Rs}), \gamma\right) = \vec{g}^{\overrightarrow{Ls}} \cdot \vec{h}^{\overrightarrow{Rs}} \cdot h^\gamma$ for $S \in \mathbb{G}$
compute challenge $y = \mathrm{Hash}(A, S)$ for $y \in \mathbb{Z}_p^*$
compute challenge $z = \mathrm{Hash}(A, S, y)$ for $z \in \mathbb{Z}_p^*$
choose random $\delta, \rho \in \mathbb{Z}_p$ and
compute commitment to polynomial coefficients $t_1, t_2$:
$T_1 := \mathrm{Com}(t_1, \delta) = g^{t_1} h^\delta$
$T_2 := \mathrm{Com}(t_2, \rho) = g^{t_2} h^\rho$
compute challenge $x = \mathrm{Hash}(T_1, T_2)$ for $x \in \mathbb{Z}_p^*$
calculate vector polynomial $\vec{L}, \vec{R}$ as:
$\vec{a} = \vec{l}(X) = \overrightarrow{Lv} - z \cdot \vec{1}^n + \overrightarrow{Ls} \cdot X$
$\vec{b} = \vec{r}(X) = \vec{y}^n \circ \left(\overrightarrow{Rv} + z \cdot \vec{1}^n + \overrightarrow{Rs} \cdot X\right) + z^2 2^n$
calculate $\hat{t} := \langle \vec{l}(X), \vec{r}(X) \rangle = \sum_{i=0}^{n} \sum_{j=0}^{i} \langle \vec{l_i}, \vec{r_j} \rangle \cdot X^{i+j}$
calculate $\mu = \beta + \gamma \cdot x$
calculate $\tau = \rho \cdot x^2 + \delta \cdot x + z^2 \cdot \alpha$
calculate commitment to $\vec{a}, \vec{b}$
$P := \mathrm{Com}((\vec{a}, \vec{b}), 0) = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}}$
compute challenge $w = \mathrm{Hash}(\vec{g}, \vec{h}, P, \hat{t})$
calculate $P' = P \cdot u^{w \cdot \hat{t}}$
INPUT: $(\vec{g}, \vec{h}, P', u^w, \vec{a}, \vec{b})$
if $n = 1$ then
    Publish $a$ and $b$.
end
if $n > 1$ then
    while $n > 1$ do
        $n' = \frac{n}{2}$
        $\overrightarrow{Lc} = \langle \overrightarrow{La}, \overrightarrow{Rb} \rangle$
        $\overrightarrow{Rc} = \langle \overrightarrow{Ra}, \overrightarrow{Lb} \rangle$
        $L\mathcal{H} := \mathcal{H}\left(\vec{0}^{\frac{n}{2}}, \overrightarrow{La}, \overrightarrow{Rb}, \vec{0}^{\frac{n}{2}}, \overrightarrow{Lc}\right) = \overrightarrow{Rg}^{\overrightarrow{La}} \cdot \overrightarrow{Lh}^{\overrightarrow{Rb}} \cdot u^{w \cdot \overrightarrow{Lc}}$
        $R\mathcal{H} := \mathcal{H}\left(\overrightarrow{Ra}, \vec{0}^{\frac{n}{2}}, \vec{0}^{\frac{n}{2}}, \overrightarrow{Lb}, \overrightarrow{Rc}\right) = \overrightarrow{Lg}^{\overrightarrow{Ra}} \cdot \overrightarrow{Rh}^{\overrightarrow{Lb}} \cdot u^{w \cdot \overrightarrow{Rc}}$
        compute challenge $r = \mathrm{Hash}(L\mathcal{H}, R\mathcal{H})$
        compute $\vec{a}' = r \cdot \overrightarrow{La} + r^{-1} \cdot \overrightarrow{Ra}$
        compute $\vec{b}' = r^{-1} \cdot \overrightarrow{Lb} + r \cdot \overrightarrow{Rb}$
        compute $\vec{g}' = (\overrightarrow{Lg})^{r^{-1}} \circ (\overrightarrow{Rg})^r$
        compute $\vec{h}' = (\overrightarrow{Lh})^r \circ (\overrightarrow{Rh})^{r^{-1}}$
        compute $P'' = (L\mathcal{H})^{x^2} \cdot P' \cdot (R\mathcal{H})^{x^{-2}}$
        New INPUT: $\left(\vec{g}', \vec{h}', P'', u^w, \vec{a}', \vec{b}'\right)$
    end
end
```

In our case, coefficients of the polynomials are not scalars but vectors. A naive approach is that verifier sends an evaluation point x as a random challenge to prover. In reply the prover sends him the commitments P_1, P_2, and the value of polynomial at x that is $P(x)$. Finally, the verifier checks if p_0 is equal to $p(x) - (P_1)^x \cdot (P_2)^{x^2}$ or not. In the affirmative case, the verifier will get convinced by the prover's proof and accept it while in the other case, he rejects the proof. However, using this approach leaks the partial information about the coefficients p_1, p_2 of polynomial $p(X)$. Solution to this information leak problem is to blind the coefficients by a blinding value s, calculate one more commitment to this blinding value and denote it by S. Commitment to s is defined as $S := \text{Com}(s, \omega) = g^s \cdot h^\omega$. A quadratic polynomial in vector form with blinded value can be written as:

$$\ddot{p}(X) := p(X) + s \cdot X = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ s \end{bmatrix} \cdot \begin{bmatrix} 1 & X & X^2 & X \end{bmatrix} \tag{7}$$

Now, the verifier get convinced by checking if $p_0 = p(x) - (P_1)^x \cdot (P_2)^{x^2} \cdot (S)^x$ is true or not. He accept the prover's proof if and only if the above equality holds, otherwise reject.

6. Proposed efficient range-proof

Consider a simple statement to prove that a secret number v is in range $(0, 2^n)$. To prove it, the prover needs to provide a range of proof to the verifier that will guarantee the verifier that he is telling true that the value is, in fact, in the range without revealing the value itself. Several range-proof techniques exist in literature; however, in all of them, the bit decomposition technique used in Bulletproof paper named "proof of knowledge of vector" is the most efficient. Since any integer which is a scalar number v can be decomposed into bit form. To ensure that v lies in the interval $(0, 2^n)$, the decomposition need to be done using 2^{n-1}. For example to prove that $v = 5$ lies in interval $(0, 2^4)$, we can rewrite integer $v = 5$ as $5 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3$. We can also define the bit form of value $v = 5$ as one vector and call it $\vec{a}$ similarly, base 2 values as another vector and call it $\vec{b}$. So, vector $\vec{a} = (1, 0, 1, 0)$ and vector $\vec{b} = (2^0, 2^1, 2^2, 2^3)$ with $v = \vec{a} \cdot \vec{b}$. Notice that $\vec{a}. \cdot \vec{b}$ is nothing but the inner product value of two vectors. Therefore, in this scheme, for given v, the prover convinces the verifier that he knows two vectors $\vec{a}$ and $\vec{b}$ such that their inner product is equal to v that is $v = \langle \vec{a}, \vec{b} \rangle$. The prover uses the Pedersen commitment scheme but the randomness value as zero, making the commitment scheme binding but not hiding.

In the proposed range proof, prover follows the procedure given in Algorithm 10. In order to use our new efficient range proof, the verifier needs to follow the procedure given in algorithm 9.

7. Discussion and conclusion

This work improved the current zkLedger-based auditing system by replacing their range of proof with the most efficient range proof technique based on the improved inner product based zero-knowledge proofs. Replacement of range proof substantially improves the system efficiency by reducing the proof size. To prove that a number lies in the range $(0, 2^n)$ and if there are m such range-proofs then the current zero-knowledge ledger range-proof takes $(0.63 \cdot m \cdot n)$ elements from a cyclic group $\mathbb{G}$(an elliptic curve over the prime field) and $(1.26 \cdot m \cdot n + 1)$ elements from the group of a prime integer $\mathbb{Z}_p$. While, the new range-proof technique takes only $2(\log_2 \ m + \log_2 n) + 4$ elements from $\mathbb{G}$ and 5 elements from $\mathbb{Z}_p$. Due to this reduction, the overall requirement in terms of computational power and memory for the new range-proof will be significantly decreased. This aspect is quite useful, especially for resource-constrained devices such as IoT sensors and mobile devices and thereby helps to run zero-knowledge proofs on these resource-constrained devices. Moreover, the idea is not only theoretical but also realistic and practically implementable. Therefore, as part of future work, we would like to implement our range-proof technique in the current zkLedger prototype using Golang language to demonstrate the efficiency of our technique and also do a suitable evaluation by comparing it with the other range-proof implementations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research is part of the Youth-Community for CyberSkills project supported by Industriens Fond (The Danish Industry Foundation). The work of Waleed Alnumay is funded by Researchers Supporting Project number (RSP-2021/250), King Saud University, Riyadh, Saudi Arabia.

Appendix

Definition 11 (***Recursive Inner Product Argument***).

In this scheme, forgiven $c \in \mathbb{Z}_p$, prover convince the verifier that he knows two vectors $\vec{a}$ and $\vec{b}$ such that their inner product is equal to c that is $c = \langle \vec{a}, \vec{b} \rangle$. Prover uses the Pedersen commitment scheme but using the randomness value as zero since the zero-knowledge is not really required here. He computes the binding but not hiding commitments denoted by A and B, to the vector $\vec{a}$ and $\vec{b}$ as

$$A := \text{Com}(\vec{a}, 0) = \vec{g}^{\vec{a}} = \prod_{i=1}^{n} (g_i)^{a_i} \tag{8}$$

$$B := \text{Com}(\vec{b}, 0) = \vec{h}^{\vec{b}} = \prod_{i=1}^{n} (h_i)^{b_i} \tag{9}$$

Binding but not hiding vector commitment to inner product of $\vec{a}$ and $\vec{b}$ which is denote by I, can be calculated as

$$I := \text{Com}\left((\vec{a}, \vec{b}), 0\right) = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}} = \prod_{i=1}^{n} (g_i)^{a_i} \cdot (h_i)^{b_i} = A \cdot B \tag{10}$$

As the name suggests, prover recursively computes the new commitments A', B' to shorter vectors $\vec{a}', \vec{b}'$ and replacing them with previous A, B until he reaches to the base case which reduce the inner product of two vectors into a simple multiplication of two scalars. For this, he computes the recursive value of A_k, B_k and c_k for $k = (1-n), (2-n), \ldots, (n-1)$ as

$$A_k = \prod_{i=\max\{1,1-k\}}^{\min\{n,n-k\}} (g_i)^{a_{i+k}}, \qquad B_k = \prod_{i=\max\{1,1-k\}}^{\min\{n,n-k\}} (h_i)^{b_{i+k}}, \qquad c_k = \sum_{i=\max\{1,1-k\}}^{\min\{n,n-k\}} a_i \cdot b_{i+k}$$

Note that, $A_0 = A$, $B_0 = B$ and $c_0 = c$.

Given a random challenge $x \in \mathbb{Z}_p^*$ by the verifier, prover computes the new shorter vectors $\vec{a}', \vec{b}'$, new generator values $\vec{g}', \vec{h}'$, new commitment values A', B' and new inner product value c' as

$$\vec{a}' = \sum_{i=1}^{n} a_i \cdot x^i, \qquad \vec{b}' = \sum_{i=1}^{n} b_i \cdot x^{-i}, \qquad c' = \sum_{k=1-n}^{n-1} c_k \cdot x^{-k} \tag{11}$$

$$\vec{g}' = \prod_{i=1}^{n} g_i \cdot x^{-i}, \qquad \vec{h}' = \prod_{i=1}^{n} h_i \cdot x^{i}, \qquad A' = \prod_{k=1-n}^{n-1} (A_k)^{x^k}, \qquad B' = \prod_{k=1-n}^{n-1} (B_k)^{x^{-k}}$$

If the prover is honest then the following should hold

$$A' = (g')^{\vec{a}'}, \qquad A' = (g')^{\vec{a}'},$$

References

[1] Goldwasser Shafi, Micali Silvio, Rackoff Charles. The knowledge complexity of interactive proof systems. SIAM J Comput 1989;18(1):186–208.

[2] Morais Eduardo, Koens Tommy, Van Wijk Cees, Koren Aleksei. A survey on zero knowledge range proofs and applications. SN Appl Sci 2019;1(8):946.

[3] Koens Tommy, Ramaekers Coen, Van Wijk Cees. Efficient zero-knowledge range proofs in ethereum. 2018, https://www.ingwb.com/media/2667860/zero-knowledge-range-proofs.pdf.

[4] Narula Neha, Vasquez Willy, Virza Madars. zkledger: Privacy-preserving auditing for distributed ledgers. In: 15th {USENIX} symposium on networked systems design and implementation ({NSDI} 18), 2018, p. 65–80.

[5] Wu Huixin, Wang Feng. A survey of noninteractive zero knowledge proof system and its applications. Sci World J 2014;2014:560484, 1–7.

[6] Miers Ian, Garman Christina, Green Matthew, Rubin Aviel D. Zerocoin: Anonymous distributed e-cash from bitcoin. In: 2013 IEEE symposium on security and privacy. IEEE; 2013, p. 397–411.

[7] Iwendi Celestine, Jalil Zunera, Javed Abdul Rehman, Reddy Thippa, Kaluri Rajesh, Srivastava Gautam, et al. Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks. IEEE Access 2020;8:72650–60.

[8] Patel Harshita, Singh Rajput Dharmendra, Thippa Reddy G, Iwendi Celestine, Kashif Bashir Ali, Jo Ohyun. A review on classification of imbalanced data for wireless sensor networks. Int J Distrib Sens Netw 2020;16(4):1550147720916404.

[9] Dwivedi Ashutosh Dhar, Singh Rajani, Ghosh Uttam, Mukkamala Raghava Rao, Tolba Amr, Said Omar. Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things. J Ambient Intell Humaniz Comput 2021.

[10] Kaushik Keshav, Dahiya Susheela, Singh Rajani, Dwivedi Ashutosh Dhar. Role of blockchain in forestalling pandemics. In: 2020 IEEE 17th international conference on mobile ad hoc and sensor systems (MASS). 2020, p. 32–7.

[11] Makkar Aaisha, Ghosh Uttam, Rawat Danda B, Abawajy Jemal H. Fedlearnsp: Preserving privacy and security using federated learning and edge computing. IEEE Consumer Electron Mag 2022;11(2):21–7.

[12] Boudot Fabrice. Efficient proofs that a committed number Lies in an interval. In: Preneel Bart, editor. Advances in cryptology — EUROCRYPT 2000. Berlin, Heidelberg: Springer Berlin Heidelberg; 2000, p. 431–44.

[13] Camenisch Jan, Chaabouni Rafik, shelat abhi. Efficient protocols for set membership and range proofs. In: Pieprzyk Josef, editor. Advances in cryptology - ASIACRYPT 2008. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008, p. 234–52.

[14] Peng K, Bao F. An efficient range proof scheme. In: 2010 IEEE second international conference on social computing, 2010. p. 826–33.

[15] Koens Tommy, Ramaekers Coen. Efficient zero-knowledge range proofs in ethereum. 2017.

[16] Dwivedi Ashutosh Dhar, Singh Rajani, Dhall Sakshi, Srivastava Gautam, Pal Saibal K. Tracing the source of fake news using a scalable blockchain distributed network. In: 2020 IEEE 17th international conference on mobile ad hoc and sensor systems (MASS). 2020, p. 38–43.
[17] Feige Uriel, Fiat Amos, Shamir Adi. Zero-knowledge proofs of identity. J Cryptol 1988;1(2):77–94.
[18] Yuen Tsz Hon, Huang Qiong, Mu Yi, Susilo Willy, Wong Duncan S, Yang Guomin. Efficient non-interactive range proof. In: International computing and combinatorics conference. Springer; 2009, p. 138–47.
[19] Chaabouni Rafik, Lipmaa Helger, Zhang Bingsheng. A non-interactive range proof with constant communication. In: International conference on financial cryptography and data security. Springer; 2012, p. 179–99.
[20] Bünz Benedikt, Bootle Jonathan, Boneh Dan, Poelstra Andrew, Wuille Pieter, Maxwell Greg. Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE symposium on security and privacy (SP). IEEE; 2018, p. 315–34.
[21] Poelstra Andrew, Back Adam, Friedenbach Mark, Maxwell Gregory, Wuille Pieter. Confidential assets. In: International conference on financial cryptography and data security. Springer; 2018, p. 43–63.
[22] Al-Balasmeh Hani, Singh Maninder, Singh Raman. Framework of data privacy preservation and location obfuscation in vehicular cloud networks. Concurr Comput Pract Exp 2022;34(5).
[23] Dhall Sakshi, Dwivedi Ashutosh Dhar, Pal Saibal K, Srivastava Gautam. Blockchain-based framework for reducing fake or vicious news spread on social media/messaging platforms. ACM Trans Asian Low-Resour Lang Inf Process 2021;21(1).
[24] Choo Kim-Kwang Raymond, Ghosh Uttam, Tosh Deepak K, Parizi Reza M, Dehghantanha Ali. Introduction to the special issue on decentralized blockchain applications and infrastructures for next generation cyber-physical systems. ACM Trans Internet Tech 2021;21(2):38e:1–3.
[25] Bootle Jonathan, Cerulli Andrea, Chaidos Pyrros, Groth Jens, Petit Christophe. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Annual international conference on the theory and applications of cryptographic techniques. Springer; 2016, p. 327–57.

Rajani Singh is an Assistant Professor at the Department of Digitalization at the Copenhagen Business School (CBS). Her research field includes Blockchain, Game theory, Mathematical Economics, Optimization Theory and Cryptography.

Ashutosh Dhar Dwivedi is a Postdoctoral Researcher at the Department of Digitalization at the Copenhagen Business School (CBS). His research field includes CyberSecurity, Machine Learning, Cryptography (Symmetric Key, Lightweight, Post Quantum, Public Key) and Blockchain.

Raghava Rao Mukkamala is the director of the Centre for Business Data Analytics, an associate professor at the Department of Digitalization, Copenhagen Business School. Raghava's current research focus is on the interdisciplinary approach to big data analytics.

Waleed S. Alnumay is currently working as an Associate Professor of Mobile Networking with Riyadh Community College, Computer Science Department, King Saud University, Riyadh, Saudi Arabia.