Towards Anonymous, Unlinkable, and Confidential Transactions in Blockchain

Kalpana Singh, Nicolas Heulot, Elyes Ben Hamida

Towards Anonymous, Unlinkable, and Confidential Transactions in Blockchain

Kalpana Singh*, Nicolas Heulot*, Elyes Ben Hamida*
*IRT SystemX, Paris-Saclay, France,
Email: {Kalpana.Singh, Nicolas.Heulot, Elyes.Ben-Hamida}@irt-systemx.fr

***Abstract*—In this paper, we investigate the issues of data and users' privacy in decentralized environments. We propose a novel security and privacy-preserving protocol for the blockchain that addresses the limitations of existing approaches, mainly the anonymity and unlinkability of users' identities and the privacy of transactions. We highlight the benefits of our proposed protocols across various use cases and we theoretically analyze its efficiency and robustness.**

I. Introduction

The concept of blockchain was introduced in 2008 by Satoshi Nakamoto [1] which offered a practical model for anonymous monetary transactions. It is a fastest growing revolutionary technology that is expected to disrupt businesses in key domains, including finance, energy, mobility, logistics, and insurance. More recently, smart contracts have emerged as a new feature enabling the automation of business workflows and rules over the blockchains and whose proper execution is enforced by the consensus mechanism. A blockchain is secure by design and relies on well-known cryptographic algorithms to provide key properties, such as resistance to tampering, pseudo-anonymity, fault-tolerance, auditability, resilience, and non-repudiation. However, *privacy* remains a fundamental obstacle for individuals, organizations, and industries as features of the blockchain's transparency and permanency. The blockchain technologies have several major technical barriers that make them impractical for mainstream use today. Given that anyone in the world can create a new wallet anonymously and transact using it. On one hand, the great promise of blockchain is pseudonymity: transactions are recorded and stored in a public ledger, but they are linked to an account address. There is no real-world identity attached to this address. However, this advent of total security is ambiguous. On the blockchain, a person can preserve his or her privacy as long as the pseudonym is not linked to the individual, but as soon as someone makes the connection, the secret is revealed. One instance of such an occurrence was revealed when law enforcement agencies admitted that they were able to identify specific Bitcoin users during investigations, thus "de-anonymizing" them and breaking with the overall premise of a blockchain's total transactional invisibility [2].

A. Privacy on the Blockchain

In the blockchain platforms, like Ethereum [3], users are interacting with smart contracts that handle not just simple value transfers but also data sharing and process automation. By design, all details about these smart contracts are accessible by anyone through the blockchain ledger, including senders and recipients, transaction data itself, the bytecode executed, and the state of each variable stored inside the contract. Privacy and security issues continue to be major barriers in the blockchain, since revealing sensitive data over the blockchain can present a serious threat. Anonymity and unlinkability of individuals' identity and privacy of transaction are the crucial issues in a decentralized network. In this line of research, T. Okamoto and K. Ohta [4] described six criteria for privacy, which included the relationship between the user and his purchases must be fully anonymous by anyone. In the paper [5], authors developed a fully anonymous model must satisfy in order to comply with the requirements outlined by Okamoto and Ohta. In addition to this line of research work, we find the careful blockchain analysis may reveal a connection between the users of the Bitcoin network and their transactions. It means the existing solutions did not satisfy the fully anonymous requirements. Since all the transactions that take place between the network's participants are public, any transaction can be evidently linked to the individuals.

From the descriptions of these papers [4] and [5], we determine our privacy notion. By privacy, we mean the unlinkability and anonymity in the sense that even an infinitely powerful an adversary with access to an unbounded number of transactions (same sender) cannot guess his identity with an advantage and cannot link the transactions to the same sender. We also achieve confidentiality during the transaction on the blockchain which adds an extra layer of privacy in our protocol.

B. Contribution

This paper proposes a solution which efficiently reduces the overhead and maintains levels of security and privacy of sensitive information and individual's identity during transactions on the blockchain platform. The solution has simpler and more intuitive design and improved efficiency than existing linkable and traceable ring signatures. Our objective is to obtain three privacy requirements. First is 'anonymity': for each transaction senders are not identified. Second is 'unlinkability': for any outgoing transactions, it is impossible to prove they were sent by the same sender (for example. if a sender has a number of accounts then it is not possible to link the transactions of the same sender). The third is a 'confidential transaction' which means hiding transaction value. The solution also attains security from malicious individuals. We acknowledge our solution on different use cases of decentralized systems.

C. Paper Organization

The rest of the paper is organized as follows. Section II gives background on privacy techniques for the blockchain.

Section III presents the basis of our solution, assumptions, definitions and notations for subsequent parts of the paper. Section IV describes the use cases and our contribution which provides a solution for the anonymous and confidential transaction in the Blockchain by accomplishing anonymity, unlinkability, and confidentiality features. Section V presents privacy and security proofs, and the comparison with analogous existing solutions in terms of privacy, security and computation costs. Finally, section VI concludes the paper.

II. Related Work

As discussed in Subsection I-A privacy is a hurdle in the blockchain applications. Several studies analyzing the privacy implications of blockchain technology and indicate that built-in privacy guarantees are not satisfactory. The solutions to the privacy difficulties are studied in the literature.

1) Mixing Schemes: In the direction of privacy solution in the blockchain, Mixcoin [6] allows to hold mixing services accountable in a reactive manner; however, the mixing services still remain single points of failure. Mixing schemes suffer from two drawbacks: First, the mix might just steal the money and never return it to the users. Second, the mix learns the permutation of the output addresses. To solve the problem Maxwell proposes CoinJoin [7]. With CoinJoin, it is similarly possible to hide the originator of a given transaction, however, these techniques in practice need a centralized server. If the trusted party is compromised, the anonymity of the transaction is also compromised. Another example of a mixing solution is CoinShuffle [8]. CoinShuffle tried to improve upon CoinJoin by not requiring a trusted third party to assemble the mixing transactions. CoinShuffle relies on the interaction between the users in the mixing to achieve unlinkability against malicious servers, verifiability, robustness, and cost-effectiveness.

2) Cryptographic Privacy Schemes: These mixing solutions did not provide complete privacy. In respect to these privacy drawbacks, there are cryptographic privacy solutions such as zerocoin. Zerocoin [9], an extension to Bitcoin, was among the first proposals to provide unlinkability between individual Bitcoin transactions without introducing a trusted party. Zerocoin [9] and its successors [10] provide strong anonymity without any third party, but lack compatibility with the current Bitcoin system. Another attempt to provide privacy in cryptocurrency is ZCash [11], which uses the zero-knowledge succinct non-interactive argument of knowledge (zk-SNARKs) [11]. There are few problems are examined in the literature. The first problem is the efficiency which is much worse than the normal bitcoin transaction (takes a few minutes to perform a spent computation). Another problem with ZCash is the "trusted setup". Zcash has some serious issues related to the "toxic waste" during the "trusted setup" and has many additional risks due to the very new cryptography that is being applied. However, we can not find the malicious sender/receiver due to complete anonymity feature.

3) Recent Studies on Cryptographic Privacy Schemes: The anonymous addresses on the blockchain have been described before in [12] and implemented in [13] for Monero cryptocurrency. The original Monero protocol was based on CryptoNote, which uses ring signatures and one-time keys to hide the transactions. Recently the technique of using a commitment scheme is proposed to hide the amount of a transaction [14]. Monero further improves the protocol by using a variant of linkable ring signature, which is called Ring Confidential Transactions (RingCT) [14]. The RingCT protocol is studied to protect the anonymity of an individual as well as the privacy of transactions. In the paper [15], a new efficient RingCT protocol is proposed known as RingCT 2.0. In comparison with the original RingCT protocol, RingCT 2.0 protocol presents a significant space saving, namely, the transaction size is independent of the number of groups of input accounts included in the generalized ring while the original RingCT suffers a linear growth with the number of groups, which would allow each block to process more transactions.

We observe the possible research direction for our work from the identified shortcomings on anonymity, unlinkability, scalability, and security from the Section II. We find the Zero-knowledge proofs (ZKP) based solutions [9], [10], [11], are best in privacy. However, these schemes are based on a trusted group. Ring signature based solution [14] has no trusted group. However, ring signature scheme has not achieved the privacy as ZKP based schemes [9], [10], [11]. In addition to this, ZKP protocols suffered from the malicious activities due to complete anonymous property. This paper presents a solution to solve these drawbacks by achieving the transactional privacy (confidentiality) and user's anonymity and unlinkability features in an efficient manner.

III. Preliminaries

Our scheme is based on the cryptographic assumptions and on cryptographic privacy solutions. We give the background details in this section.

A. Cryptographic Assumptions

An elliptic curve $E(\mathbb{F}_q)$ is defined as the total number of points $(x, y) \in E(\mathbb{F}_q)$. Consider a cyclic group $\mathbb{G}$ of prime order q with a generator $g \in E(\mathbb{F}_q)$, we have $g^n = 1$ where $n = \sharp E(\mathbb{F}_q)$. We define our assumptions with respect to an adversary $\breve{A}$ in Definitions 1, 2, and 3.

Definition 1: Elliptic Curve Discrete Logarithm Problem (ECDLP): $\breve{A}$ has no advantage in solving following: Given Y, $\mathbb{G} \in E(\mathbb{F}_q)$, find $x \in \mathbb{Z}_q$ such that $Y = x \cdot \mathbb{G}$.

Definition 2: Computational Diffie-Hellman Assumption (EC-CDH): $\breve{A}$ has no advantage in the following: Given $a \cdot \mathbb{G}$, $b \cdot \mathbb{G}$, $\in E(\mathbb{F}_q)$, $a, b \in \mathbb{Z}_q$, compute $ab \cdot \mathbb{G}$.

Definition 3: Decisional Diffie-Hellman Assumption (EC-DDH): $\breve{A}$ has no advantage in the following: Given $a \cdot \mathbb{G}$, $b \cdot \mathbb{G}$, $c \cdot \mathbb{G}$, $\in E(\mathbb{F}_q)$, $a, b, c \in \mathbb{Z}_q$, decide whether $c \cdot \mathbb{G} = ab \cdot \mathbb{G}$.

B. Cryptographic Building Blocks

1) Zero-Knowledge Proof: The purpose of the ZKP is for a party to prove to a verifier that they know some secret information x without revealing anything about the secret in the process. Goldwasser et al. introduced zero-knowledge proof of knowledge schemes in 1985 [16].

Goldwassers Scheme: Prover and verifier both know (g, h, y_1, y_2), with $g, h \neq 1$, $g, h \in \mathbb{G}$, $y_1 = g^x$, $y_2 = h^x$, for exponent $x \in \mathbb{Z}_q$. The scheme runs as follows [16]:

1) Prover chooses $r \leftarrow_R \mathbb{Z}_q$, sends $a \leftarrow g^r$, $b \leftarrow h^r$ to the verifier.
2) Verifier responds with challenge $c \leftarrow_R \mathbb{Z}_q$ for the prover.
3) Prover responds, sending $t \leftarrow r - c\, x\, mod q$ to the verifier.
4) Verifier accepts if and only if $a = g^t y_1^c$ and $b = h^t y_2^c$

For g and h in a cyclic group $\mathbb{G}$ where the discrete logarithm problem is assumed hard, the scheme above is both sound and honest-verifier zero-knowledge.

We use non-interactive zero-knowledge proof (NIZKP) version of Goldwasser et al. [16] scheme (NI-Goldwasser) in our proof of signature step of the protocol. This is known as NI proof of membership (NI-Goldwassers PoM).

2) Non-Interactive Schnorr proof of knowledge: We also use Fiat-Shamir [17] to transform $\sum$-protocols such as the Schnorr proof of knowledge (Schnorr PoK) of a discrete logarithm [18] into non-interactive-Schnorr PoK (NI-Schnorr PoK). Let $H(\cdot)$ be an ideal hash function. Here, we consider a single column v instead of columns v_j of a data matrix D, where $j = 1, 2, 3, ...n$ and n is the number of columns. We choose the single column to simplify the NI-Schnorr PoK using the Fiat-Shamir heuristic is defined below:
Statement: Prover (P) knows a v such that $y = g^v$ Public information: y, g, where g is a generator of a group $\mathbb{G}$ prime order q. Private information: v

1) $P \rightarrow V$**:** P Chooses random $r \in \mathbb{Z}_q$. Calculates $t = g^r$. Calculates $c = H(t)$. Calculates $s = c.v + r$. Sends the tuple (t, s).
2) V**:** V Calculates $c = H(t)$. Checks if $g^s = y^c.t$. If true, accepts proof. The proof π is the tuple (t, s).

We use this scheme during the commitment and proof of correctness (PoC) on committed data steps on a blockchain platform.

3) Verifiable Random Function: A verifiable random function (VRF) [19] is a pseudo-random function (PRF) that provides a non-interactively verifiable proof of correctness of its output. A PRF is denoted as an $F(\cdot)$ in our solution during the URS scheme. Consider a random public group element $y = g^x$, the function $F(m) = H(m)^x$ is a PRF if we model the hash function $H(\cdot)$ as a random oracle. Given the above PRF and NIZK proof system (defined in Goldwasser et al. [16]), we apply the BG paradigm [19] to obtain a VRF in our solution as the URS scheme. This is an important function in our ring signature step where we use PRF on the identity attributes.

4) Elliptic curve based Pedersen Commitment: In Pedersen commitments [20], the public parameters are a group $\mathbb{G}$ of prime order q, and generators $(g_0, ..., g_n)$. In order to commit to the values $(v_1, ..., v_n) \in \mathbb{Z}_q$, pick a random $r \in \mathbb{Z}_q$ and set $C = PedCom(v_1, ..., v_n; r) = (g_0^r \prod_{i=1}^{n} (g_i^{v_i}))$. We use a variant of the Pedersen commitment scheme based on the elliptic curve (EC) to reduce computation cost [20] during the blockchain transactions. We use "*ec-Com*" notation to represent this terminology in subsequent parts of the paper. Let $\mathbb{G}$ of large prime order q be the group of elliptic curve points. We choose n the number of values to commit to, and pick $n + 1$ random points $P_1, ..., P_n$, Q in $\mathbb{G}_q$. The commitment key is the tuple $(P_1, ..., P_n, Q)$. To commit to the values $(v_1, ..., v_n) \in \mathbb{Z}_q$, where $v_1, ..., v_n$ are numbers of column values. We pick a random integer $r \in Z_q$ and calculate the commitment as: $Com(v_1, ..., v_n, r) = v_1 \cdot P_1, ..., v_n \cdot P_n + r.Q$. To open the commitment we simply reveal the values $v_1, ..., v_n$ and r. The EC-Pedersen commitment scheme is additively homomorphic, so it has the following properties:
$Com(v_1 + y_1, ..., v_n + y_n, r_v + r_y) =$
$Com(v_1, ..., v_n, r_v) \cdot Com(y_1, ..., y_n, r_y)$
$Com(k \cdot v_1, ..., k \cdot v_n, k \cdot r) = Com(v_1, ..., v_n, r)^k$, Where, $y_1, ..., y_n$, r_v, r_y, $k \in Z_q$.

5) Ring Signature: To produce a ring signature, the actual signer declares an arbitrary set of possible signers that includes himself and computes the signature entirely by himself using only his secret key and the others public keys. A ring signature RS scheme consists of a quadruple of probabilistic polynomial time (PPT) algorithms $(Setup, RingKGen, RingSig, RingVfy)$. We refer to a ring $R = (pk_1, pk_2, ..., pk_n)$ of public keys as a ring. We utilize these quadruple $(Setup, RingKGen, RingSign, RingVfy)$ in our ring signature scheme with the privacy and security features if it is correct, unforgeable and anonymous in the Algorithm 1.

IV. Our Solution

The protocol we present in this section is designed to maintain the level of privacy as discussed in the above Section II, and to improve on existing efficiency by using elliptic curve based schemes. For the most part, our protocol is based on the Franklin and Zhangs URS [19]. Our protocol offers the usual expected features of anonymity and unlinkability of the sender but can reveal the identity of a sender who maliciously tries to use the system. Additionally, we provide the confidential transaction between the users on the blockchain. In addition to the privacy, our protocol provides unforgeability, undeniable sender identity, satisfies 'once concealed, twice revealed', defined and proved in Subsection V-A.

A. Use Cases Overview

We consider an architecture based on the Ethereum technology (Ethereum Virtual Machine 'EVM') [3]. This architecture can be either a public or consortium blockchain [21]. We use this technology because it implements a smart contract. So, this technology can make sure that only honestly generated transactions will be finally immutably stored in the blockchain ledger. We consider a process of information exchange based on the blockchain using smart contract with the following participants and delineated in the Figure (Figure 1):

- **Sender** (S): The sender information/data requires privacy. For example, an individual who wants to buy tickets for a sport's event in a stadium.
- **Receiver** (R): A receiver information/data which is supposed to be publicly known. For example, a ticket seller who is in charge of promoting an event.
- **Verifier** (V): The blockchain with its Smart contract in charge of verifying and validating the protocol.
- **Authority** (A): The authorities legally in charge of controlling the interactions between the different users. For example, a registration authority in charge

of controlling the identities of people entering a stadium.

As we mentioned previously, our protocol provides the anonymity and unlinkability of a sender. However, it does not address the same properties for the recipient during data exchange on the blockchain. So, we consider only use cases where the recipient is public such as a marketplace on the blockchain. In particular, the following use cases can be demonstrated in our protocol:

Ticket selling: the sale of tickets for a sporting event is publicly announced. However, each buyer of a ticket has to protect his privacy. So, each buyer will keep his identity private. Nevertheless, he has to share his identity information to the security authorities who are able to refuse the sell to the blacklisted individuals (for example, to refuse an access from dangerous hooligans to a stadium).

Bond market: This is a financial market where participants can issue a new debt, known as the primary market. This is usually in the form of bonds. When a participant issues bonds to raise money, all the details of his issuance project are publicly shared on a marketplace in order to bring investors. However, these investors who place an order have to protect both their identity and the amount of their investment. In this example, regulation authority appoints an investor's identity which is shared during a legal process named "Know Your Customer" (KYC).

The anonymity and unlinkability of each user and confidentiality of data during transactions should be preserved in these three use cases. These are the main focus of our protocol. In these use cases, legal authorities often impose to be able to ask to reveal the identity of users to prevent the malicious activity. Additionally, each sender can have more than one account in all use cases. However, senders can not use same data for the different transaction using distinct accounts.

In our protocol, we use I denoted as the identity attributes, such as ID card number, social security number or contact address. We use D denoted as the exchanged information which requires privacy according to the use cases (defined above) such as technical details, cost proposal or investment proposal. The proofs are presented in SubsectionV-A. We consider the same participants (as described in the above SectionIV-A) in following parts of the paper.

B. Protocol Description

We assume the data to be sent contains sensitive information about individuals. We maintain the privacy of these individuals by ensuring that an attacker who obtains the data cannot connect this sensitive information with the individual it describes. Thus, in the setup, a sender S wants to send data D to a receiver R in a secure and an efficient manner in such a way that the privacy of the data and sender's identity are preserved. S first determines which portion of data should be considered to be the 'identity attributes' (social security number) and 'other sensitive attributes' (cost proposal or investment proposal) in need of privacy maintenance. We use I to refer to 'identity attributes' data, as determined by S, and D to refer to the 'other sensitive attributes' data. This is an essential assumption in our use case. The privacy of the complete data is very crucial.

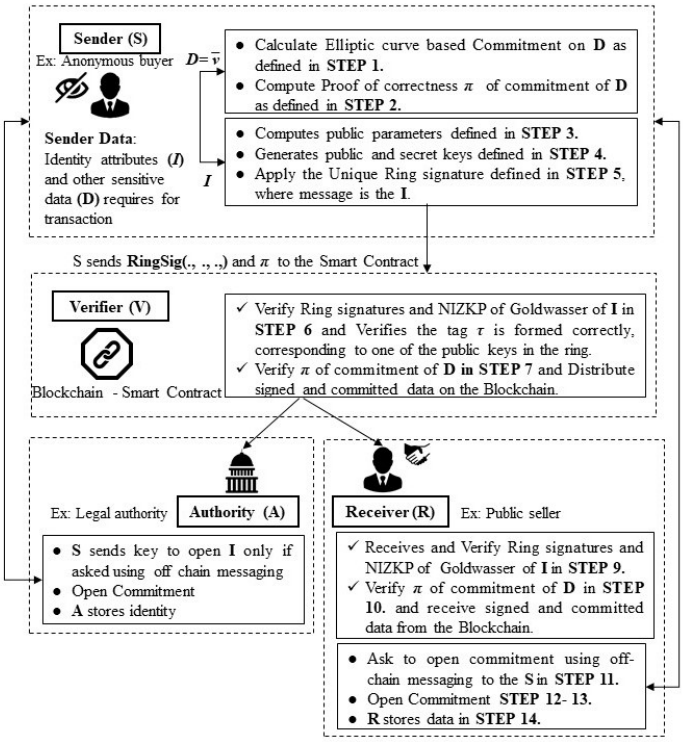

Fig. 1: The Protocol Overview

1) The Data Setup: Each sender S accounts data in a matrix format, which we refer to as D. Over time, the dimensions of D will vary; however, to simplify our explanation, we suppose that at this time, D has m rows, numbered 1 to m, and n columns, numbered 1 to n. Thus, in the setup, a sender S wants to send data D to another user (receiver) R. Let D = $[v_{ij}]$, $i = 1; m$, $j = 1; n$ be a matrix with values v_{ij}, where $i = 1, 2, 3, ..., m$ and $j = 1, 2, 3, ..., n$. We use the notation v_j to refer to a column of D. The S uses a commitment on each column data v_j to hide the data values. In addition to this data S has the *identity attributes* I. The I attributes are generated by an authority depending on the targeted case scenarios defined in Subsection IV-A.

2) The Protocol Setup: Sender S is the i^{th} user in the ring. We use the notation $\leftarrow_R$, to indicate choosing an element at random from a set, for example, $t_j \leftarrow_R \mathbb{Z}_q$ shows t_j chosen at random from $\mathbb{Z}_q$. We begin by recalling the definition of a ring signature scheme $RS = (setup, RingKGen, RingSig, RingVfy)$ that consists of four algorithms. This signature is based on the Franklin and Zhangs URS in the random oracle [19]. We use this signature scheme to get user's anonymity. The NI-Goldwassers PoM scheme is used for the proof of signature in our protocol. These schemes are used for anonymity of a sender in the paper [19]. We use $ec - Com$ commitment scheme (defined in Section III-B) to hide data exchanged, and generate the NI-Schnorr PoK to prove the correctness (PoC) of committed data. The in-depth security proof shows that our proposed scheme is provably secure under the random oracle model [19]. This $ec - Com$ protocol is utilized on data D and generates the committed data. D is first processed on the sender side by

the privacy-enhancing protocols of $ec-Com$ with NI-Schnorr PoK. The Franklin and Zhangs URS with NI-Goldwassers

Data: A data matrix $D = [v_{ij}]_{i=1,m}^{j=1,n}$ with columns $\mathbf{v_j}$, where $j = 1, 2, 3, ..., n$, Private information: $v_1, v_2, ..., v_n, r$ and r is a commitment key, S is in a ring with the public keys $R = (pk_1, pk_2, ..., pk_n)$. Identity attributes I.
Result: **RingVfy** $\parallel$ Proof of correctness (PoC) of committed data π
STEP-1: S calculates committed data using EC-commitment *ec-Com* $((v_1, v_2, ..., v_n), r)$ of data $\mathbf{v_j}$, define in Algorithm-phase 2.
STEP-2: S calculates PoC of committed data π using NI-Schnorr PoK scheme, define in Algorithm-phase 2.
; `// The URS scheme on identity attributes` I
STEP-3: S generates URS setup *Setup* (1^λ), define in Algorithm-phase 3.
STEP-4: S generates a Key generation of URS $(1^\lambda, \boldsymbol{pp})$, define in Algorithm-phase 3.
STEP-5: S calculates signature on I using URS *RingSig*$(sk_i, R = (pk_1, pk_2, ..., pk_n), I)$, define in Algorithm-phase 3.
; `//` S `sends committed and signed data to the` V `(smart contract).`
STEP-6: V Verifies signature *RingVfy*(R, I, σ), define in Algorithm-phase 4.
STEP-7: V verifies PoC of committed data pi, define in Algorithm-phase 4.
STEP-8, 9, 10 R receives signed data and verifies (if needed) *RingVfy*(R, I, σ), define in Algorithm-phase 4, and PoC also define in Algorithm-phase 4.
STEP-11 R asks to open commitment on the blockchain.
STEP-12 - 13: Sender i receives queries and sends private information via the off-chain communication channel to R.
STEP-14: R opens the commitment and stores on its private cloud.
STEP-15 I is only seen by the authority if asked.

Algorithm 1: Our Protocol: A Solution for Anonymous, Unlinkable and Confidential Transactions in Blockchain

PoM schemes in the random oracle are used for sender's anonymity on the identity attributes I. The privacy of the scheme relies on ECDDH, which itself is closely related to ECDLP, which has been described as "the hardest math problem ever" [22]. In the protocol, we consider four users: sender S, verifier V known as 'smart contract', receiver R, and authority A, defined in detail in Subsection IV-A. Our protocol is now described in Figure 1. The sensitive data D is passed by the S into the $ec-Com$ privacy-enhancement process box in Step-1 of Figure 1 and the EC-commitment protocol is applied to it. In addition to validate this commitment, S generates proof of correctness which is based on NI-Schnorr PoK in Step-2.

Data: A data matrix $D = [v_{ij}]_{i=1,m}^{j=1,n}$ with columns $\mathbf{v_j}$, where $j = 1, 2, 3, ..., n$, Private information: $v_1, v_2, ..., v_n, r$ and r is a commitment key.
Result: committed data $\tilde{w}$, and $s^{'}$ $\parallel$ PoC of committed data π
STEP-1: S Calculates *ec-Com* $((v_1, v_2, ..., v_n), r)$
; `//` $\tilde{v} = (v_1, v_2, ..., v_n)$ `To simplify notation we use`
S chooses a r is a commitment key $r \in \mathbb{Z}_q$.
S picks random values $u_1, u_2, ..., u_n \in Z_q$, $s_u \in Z_q$.
; `//` $\tilde{u} = (u_1, u_2, ..., u_n)$ `To simplify notation we use`
Calculates: $Com(\tilde{u}, s_u)$
(I) Calculates $H(Com(\tilde{u}, s_u))$.
(II) Transform the output into an integer $z \in Z_q^*$.
foreach $j \in \{1, ..., n\}$ **do**
  $w_j = z.v_j + u_j$;
$s^{'} = z.r + s_u$;
Output: $\{w_1, w_2, ..., w_n\}$, $s^{'}$.; `//` $\tilde{w} = (w_1, w_2, ..., w_n)$ `To simplify notation we use`
; `// Calculates proof of correctness (PoC)` π
STEP-2: Generates PoC π, $\pi(Com(\tilde{u}, s_u), \tilde{w},$ and $s^{'})$.
S sends $Com(\tilde{u}, s_u)$, $\tilde{w}, s^{'}$ to the V for verifying PoC π in Step 7 of Algorithm-phase 4.

Algorithm-Phase 2: STEPS 1 and 2: Calculation of Committed data and generation of PoC for verification in Step 7 of Algorithm-phase 4.

Now S uses URS protocol on attributes I from Steps 3-5. The two results are then sent to verifier V (smart contract) for verification and validation. V verifies the signature and PoC of committed data in Steps 6 and 7. Then, V publishes this validated information on the blockchain. We use a smart contract as a verifier V.

Data: R, Identity attributes I
Result: Ring signature $RingSig(R, I, H(I \parallel R)^{x_i}, c_1 t_1, ..., c_n t_n)$
STEP-3: Ring Setup:*Setup* (1^λ)*:*
For each User i (is a sender), $1 \leq i \leq n$ choose λ the security parameter, choose multiplicative group $\mathbb{G}$ with prime order q, and randomly chosen generator g of $\mathbb{G}$. Choose also two hash functions H and $H^{'}$ such that:
(I) $H: \{0,1\}_* \rightarrow \mathbb{G}$.
(II) $H^{'}\{0,1\}_* \rightarrow \mathbb{Z}_q$.
Output public parameters **pp** = $(1^\lambda, q, \mathbb{G}, H, H^{'})$
STEP-4: Ring Key Generation: *RingKGen*$(1^\lambda, \boldsymbol{pp})$
(I) Sender i generates secret key sk_i and public key pk_i.
(II) $x_i \leftarrow_R \mathbb{Z}_q$.
(III) $y_i \leftarrow g^{x_i}$.
Output public key $pk_i = (pp, y_i)$, secret key $sk_i = (pp, x_i)$.
; `//` I `The URS with NI-Goldwassers PoM schemes to sign on identity attribute`.
STEP-5: Ring Signature: *RingSig*$(sk_i, R = (pk_1, pk_2, ..., pk_n), I)$
foreach $j \in [n]$, $j \neq i$ **do**
  $t_j, c_j \leftarrow_R \mathbb{Z}_q$;
  $a_j \leftarrow g^{t_j} y_j^{c_j}$;
  $b_j \leftarrow H(I \parallel R)^{t_j} (H(I \parallel R)^{x_j})^{c_j}$;
foreach $j \in [n]$, $j = i$ **do**
  $r_i \leftarrow_R \mathbb{Z}_q$;
  $a_i \leftarrow g^{r_i}$;
  $b_i \leftarrow H(I \parallel R)^{r_i}$;
Calculate $c_i \leftarrow H^{'}(I, R, H(I \parallel R)^{x_i}, \{a_j, b_j\}_1^n) - \sum_{j \neq i} c_j] \bmod q$.
$t_i \leftarrow r_i - c_i x_i \bmod q$.
Return $(R, I, H(I \parallel R)^{x_i}, c_1 t_1, ..., c_n t_n)$.
S sends $(R, I, H(I \parallel R)^{x_i}, c_1 t_1, ..., c_n t_n)$ for the ring verification in STEP-6 of Algorithm-phase 4.

Algorithm-Phase 3: STEPS 3, 4, 5: Signed data with PoM $(R, I, H(I \parallel R)^{x_i}, c_1 t_1, ..., c_n t_n)$ for the V in STEP 6 of Algorithm-phase 4.

Data: $\pi((R, I, H(I \parallel R)^{x_i}, c_1 t_1, ..., c_n t_n) \parallel Com(\tilde{u}, s_u), \tilde{w}, s^{'}$
Result: **RingVfy** $\parallel$ Verify the PoC of committed data π
; `//` $\pi(Com(\tilde{u}, s_u),\ \tilde{w},$ `and` $s^{'})$ `from STEP 2 of Algorithm-phase 2 to verify the PoC using NI-Schnorr PoK.`
; `//` $(R, I, H(I \parallel R)^{x_i}, c_1 t_1, ..., c_n t_n)$ `from STEP 5 Algorithm-phase 3 to verifying signature anonymously.`
STEP-6:*RingVfy*(R, I, σ) and $Com(\tilde{u}, s_u)$, $\tilde{w}, s^{'}$.
(I) Parsing the output of **RingSig**, and using the notation $H(I \parallel R)^{x_i} = \tau$.
(II) Perform the comparison: $\sum_1^n c_j = H^{'}(I, R, \{g^{t_j} y_j^{c_j}, H(I \parallel R)^{t_j} \tau^{c_j}\}_1^n)$.
If satisfies the comparison, a signature is verified. Otherwise, an unauthorized user.
STEP-7: V verifies proof of correctness of committed data.
(I) Calculates: $Com(\tilde{w}, s^{'})$.
(II) Calculates: $H(Com(\tilde{u}, s_u))$ and the output z into Z_q.
ForEach$j \in \{1, ..., n\}$ $Com(w_j, s^{'}) = Com(v_j, r).z + Com(u_j, s_u)$;
If satisfies the comparison, PoC of data is verified. Otherwise, Unvalidated committed data.
If both STEPS 6 and 7 are validated by the V then it will publish on the blockchain.

Algorithm-Phase 4: STEPS-6 and 7: Verifying Signature and PoC

So, the receiver's address is mentioned on the transaction (receiver address is visible to this work as described in Subsection IV-A. R receives data and signature on the data. R can verify both the signature and committed data using ring verification and proof of correctness in Steps 9 and 10. Then, R asks to open commitment from S. R receives this secret information via off-chain communication channels, it reconstitutes the data D by opening the commitment. Thus, data D is finally in R's hands. The identification tag is $H(I \parallel R)^{x_i}$, which corresponds to $x_i.H(I \parallel R)$ when defined over an elliptic curve group (only

seen by an authority in our scenario in Subsection IV-A). These steps are designed in detail in our proposed Algorithm in 1. We use the Figure 1 to simplify the operations which are utilized in the Algorithm 1. The Algorithm 1 has three algorithm-phases to commit and generate proofs of committed data, to sign on the identity and proof of membership, to verify the signature and correctness of committed data in the Algorithm-phases 2, 3, and 4 respectively.

C. Communication between Sender S, Verifier V (Smart Contract), Receiver R and Authority A

The flow of data in our protocol is shown in Figure 2. The protocol starts with commitment and PoC of committed data using EC-commitment protocol and NI-Schnorr PoK define in detail in Steps 1 and 2 of Algorithm 1. To authenticate anonymously S uses URS with NI-Goldwassers PoM schemes define in Steps 3, 4, and 5 of Algorithm 1. The V verifies the ring signature and PoC of committed data in Steps 6 and 7 respectively of Algorithm 1. If a valid sender's signature and the correct proof of committed data are received, V accepts the data request and publish on the blockchain. In our protocol, we assume that the receiver is looking for some important data on the blockchain. When receiver will see the transaction has its public address. R gets the transactions and again validates the signature and PoC of committed data as Steps 6 and 7 Algorithm 1. When both verification steps are done successfully. R needs to open committed data D. R sends a message using $\textit{RingSig}(sk_i, R = (pk_1, pk_2, ..., pk_n), I)$ along with its public address on the blockchain. S is actively available on the blockchain (our assumption). S receives a message and sends the private information $\tilde{u}, s_u, r$ using off-chain communication channels (secure channel) to the R. R opens the commitment and stores on the private cloud. I is only visible to the authority if asked to the S. S can not send secret key sk_i at any cost to the R (assumption). This communication is delineated in Figure 2.

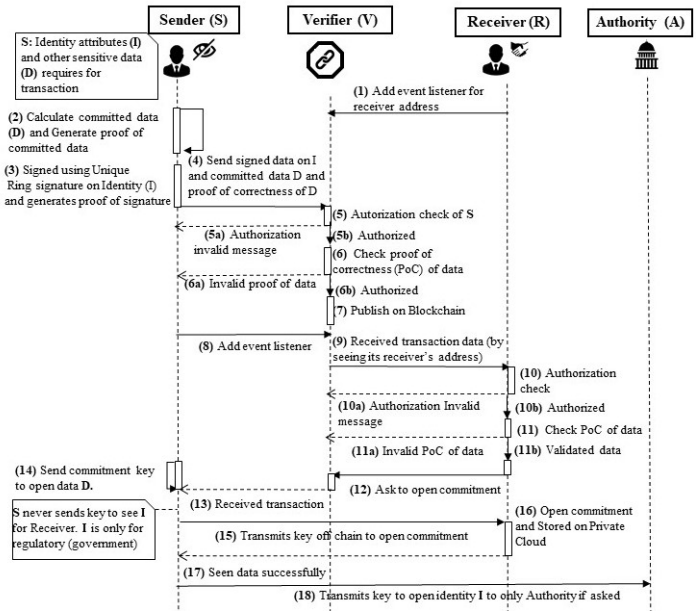

Fig. 2: Flow Diagram of the Proposed Protocol

V. Privacy, Security and Performance Analysis

This section analyses the privacy and security of our protocol. The privacy of the protocol is defined by conditional sender anonymity, confidential transaction, and unlinkability. The security is defined by unforgeability, satisfies "once concealed, twice revealed" and undeniable signer identity. We define privacy and security features by means of a game between a challenger C and an adversary $\breve{A}$, the latter who may be a sender S, a receiver R. We define the advantage of an adversary $\breve{A}$ in breaking the $sender\ anonymity$ of the protocol as the probability that the adversary $\breve{A}$ guesses private information correctly.

[*Definition*]Adversary with an Advantage The protocol has the sender anonymity if the advantage of any Probabilistic Polynomial Time (PPT) adversary is not more than $\frac{1}{2}$ plus any non-negligible value.

A. Privacy and Security Proofs

Theorem 1: Assume that the Franklin and Zhangs URS has the sender anonymity and commitment protocol also has the unconditionally hiding (i.e., a receiver does not learn anything about the values $(v_1, v_2, ..., v_n, r$ during the commit step. In this case, our protocol achieves sender anonymity and data confidentiality during the transaction.

Proof: In Section IV, Algorithm 1, S uses EC-commitment scheme and generates NI-Schnorr PoK for PoC of committed data. S sends committed data *ec-Com* $((v_1, v_2, ..., v_n), r)$ with PoC π to V for verification the correctness of committed data.

We consider any PPT adversary $\breve{A}$ generates a committed data by considering $u_1^{'}, u_2^{'}, ..., u_n^{'} \in Z_q$, $s_u^{'} \in Z_q$ and calculates $w_1^{'} = z^{'}.v_1^{'} + u_1^{'}$ for each data values $u_1^{'}, u_2^{'}, ..., u_n^{'}$, where $z^{'} \in Z_q$. Committed key r is a secret key define in Algorithm-phase 2. This is a useful step to calculate $s^{'} = z.r + s_u$. This step is a required in a verification step (Step 7 of an Algorithm-phase 4). $\breve{A}$ calculates $\pi^{'}$ as: $\pi^{'} = ec - Com(u_1^{'}, u_2^{'}, ..., u_n^{'}, s_u^{'}), w_1^{'}, w_2^{'}, ..., w_n^{'}$. However, r is a secret key. $\breve{A}$ can not calculate $s^{'}$. Hence, $\breve{A}$ can not generate the parameters for PoC of committed data. V never accepts a bad PoC of committed data, and also commitments reveal no information whatsoever about the committed values. Thus, we achieve perfectly hiding commitments. Sender anonymity proof is illustrated in detail in the Appendix F, [23]. In our case, the identity of S can reveal to Authority A only. This proof is applicable in our sender anonymity because we utilize the Franklin and Zhangs URS and EC-commitment schemes. ■

Theorem 2: Assume that the Franklin and Zhangs URS scheme has sender anonymity in random oracle, the NI-Schnorr PoK of committed data satisfies 'Honest verifier zero-knowledge'. Then our protocol satisfies the unlinkability.

Proof: As defined in the use case section IV-A, given two distinct bonds/tickets managed by the same sender S, an adversary $\breve{A}$, who may be the verifier V, or the receiver R, has two ways to link the transaction by the same sender S.

1) $\breve{A}$ links the two tickets/bonds according to their identity attributes I. In our protocol, each identity (i.e. $H(I \parallel$

$R)^{x_i} = \tau$) is used by a S only once. $\breve{A}$ first has to identify the S from the identity attributes and then link the two tickets/bonds with the identity. Our signature protocol is built on the Franklin and Zhangs URS protocol in random oracle to get a VRF (illustrated in the Subsection III-B) i.e truly random value. This URS is based on the DDH assumption (defined in Theorem 2 of the Paper [19]) which is the hardest math problem ever [22]. So, $\breve{A}$ can not guess/ calculate the identity values due to hardness solution of DDH and BG paradigm. If the Franklin and Zhangs signature provides sender anonymity. Therefore, this link attack can be prevented.

2) $\breve{A}$ traces the source of the S who sends the Bond/ticket, as defined in the Ethereum address of the sender (explain in the EVM [3]). If two bonds come from the same source, they are most likely transact by the same S. Because S sends the bond to V using EC-commitment protocol and PoC of committed bond using NI-Schnorr PoK, this link attack can be prevented.

■

Theorem 3: Assume that the Franklin and Zhangs URS scheme has unforgeability; then our protocol has unforgeability.

Proof: In Step 5 of Algorithm-phase 3, S sends signature $(R, I, H(I \parallel R)^{x_i}$ with proof of signature $c_1 t_1, ..., c_n t_n)$ to the V. Additionally, S sends proof of correctness of data $Com(\tilde{u}, s_u)$, $\tilde{w}, s^{'}$ (stated in Step 2 of Algorithm-phase 2) to V. We assumed that An adversary should be unable to verify the signature unless either a PPT adversary $\breve{A}$ has chosen one of the public keys in a ring or a sender whose public key is in ring explicitly signed the message previously (with respect to the same ring). S is controlled by $\breve{A}$. Therefore, $\breve{A}$ can construct any signature by considering the URS in random oracle. If $\breve{A}$ can produce a signature with the same ring, same identity for the same bond/ticket, he is able to forge a signature of S which contradicts the assumption that the Franklin and Zhangs URS scheme has unforgeability. Therefore, Theorem V-A holds. ■

Theorem 4: Assume that the Franklin and Zhangs URS scheme satisfies completeness and uniqueness, then our protocol detects malicious signer identity by the Authority which satisfies the feature "once concealed, twice revealed".

Proof: Based on the Theorem 1 and our assumptions III-A, we know that our protocol has sender anonymity and data confidentiality. Our protocol identifies the malicious sender in the same way as the Franklin and Zhangs URS scheme.

Here, S is controlled by $\breve{A}$ as stated in Theorem . S has more than one accounts (our assumptions defined in Subsection IV-A). If S can performs a transaction for the same bond/ticket using different accounts by using identity $H(I \parallel R)^{x_i} = \tau$ then S is able to cheat the protocol which contradicts the assumption that the Franklin and Zhangs URS scheme has completeness and uniqueness. Therefore, Theorem 4 holds. ■

Theorem 5: Assume that the Franklin and Zhangs URS scheme has NI-Goldwassers PoM of identity I, and PoC of committed data is verified by the NI-Schnorr PoK. Then our protocol satisfies the 'undeniable signer identity' feature.

Proof: Based on the properties of Franklin and Zhangs URS, NI-Goldwassers PoM, EC-commitment and NI-Schnorr PoK, Theorem security:undeniable can be proved with ease. Franklin and Zhangs URS has utilized the NI-Goldwassers PoM scheme which is used to prove that the S knows the identity I. We also commit our data using EC-commitment scheme. Now, S can not refuse on the committed data. S also generates the PoC of committed data. If S is controlled by an adversary and denies that he has not committed to the data which contradicts the protocols of Franklin and Zhangs URS, EC commitment, and NI-Schnorr PoK. Therefore, Theorem 5 holds. ■

Table I specifies the main differences between our scheme, and that of Franklin and Zhangs URS [19]. However, our scheme and Franklin and Zhangs URS have achieved Sender anonymity, Unforgeability, and Scalable features.

TABLE I: A comparison of our scheme with the Franklin and Zhang URS scheme

Feature	Malicious sender is detected	Unlink-ability	Proof of Correctness of Data	Confidential data	Undeniable sender identity
Franklin and Zhangs URS[19]	Not proved	Not Proved	NO	NO	NO
Our Protocol	YES	YES	YES	YES	YES

B. Computational Cost Comparison

As a measure of the practicality of our protocol, we calculate the computational costs of a sender, verifier/receiver in our protocol. We also compare our protocol with existing schemes on the blockchain. We follow the method of measurement used by [24]; thus, we assume that H is the computational time of one hashing operation, M is the computational time of one modular multiplication, E is the computational time of one modular exponential operation. EC_M, EC_P, and EC_A are the computation time of the multiplication, bilinear pairing, and addition operations respectively over an elliptic curve. Again following [24], we assume that $E \approx 240M$, $E \approx 3.2EC_P$, $H \approx \frac{2}{5}M$, and $E \approx 8.24EC_M$. The cost of inverting an element was calculated as for multiplication.

TABLE II: Comparative Evaluation Of Schemes on the Blockchain

Properties	**RingCT** [14]	**RingCT 2.0** [15]	**Our Protocol**
Computational cost to the sender	$17 \cdot n \cdot EC_M + n \cdot H$	$3 \cdot (n+1) \cdot EC_M + EC_P + 8 \cdot EC_M \approx 3 \cdot (n+1) \cdot EC_M + 10 \cdot EC_M$	$2 \cdot n \cdot EC_A + 8 \cdot n \cdot EC_M + 5 \cdot n \cdot H$
Computational cost to the verifier	$18 \cdot n \cdot EC_M + n \cdot H$	$9 \cdot E + 3 \cdot EC_P + 3 \cdot n \cdot EC_M \approx 3 \cdot n \cdot EC_M + 81 \cdot EC_M$	$7 \cdot n \cdot EC_A + 6 \cdot n \cdot EC_M$

Note: n: the number of users of input accounts

Table II shows the features of selected schemes and computes the cost of the protocols needed in the blockchain communication: signature and commitment by the S, transactions and verification by the V and R. We consider n number of users in the ring and assume that each user has only one account for computation cost only. Our protocol is applicable to a number of accounts of each user. The

computation costs of each scheme are presented in the Table II. We perform comparisons with these two schemes: RingCT [14] and RingCT 2.0 [15]. These only two solutions are close to our protocol (according to our knowledge) which addressed privacy of a user, the privacy of transaction and security on the blockchain. The maximum number of EC_M is in the RingCT 2.0, which is the costliest operation in the protocol according to the analysis in the paper [24]. The sequence success of the blockchain schemes in the computation costs are: RingCT 2.0 [15] $<$ RingCT [14] $<$ Our protocol. In this study, we achieve the low computation cost in our protocol better than the both RingCT 2.0 and RingCT schemes.

VI. Conclusion and Future work

This paper has examined the existing solutions to the lack of inherent privacy for individuals and transaction. The work is designed for the number of general use cases on the blockchain such as ticket selling, and bond market. We present a new solution with the ability to provide anonymity and unlinkability of a sender, the privacy of transaction (confidentiality), and security from the malicious behavior of entities on the blockchain. Our protocol also eliminates the relatively inefficient one-time signature. The security of the proposed system is ensured by a series of theorems. We have illustrated our solution with real uses cases such as ticket selling, and bond market. We compare our solutions with Franklin and Zhangs unique ring signatures, RingCT, and RingCT 2.0 in terms of privacy, security, and computation costs.

Future work: Our solution has some limitations such as lack of receivers anonymity and complete implementation. At present, our solution requires only basic implementations of elliptic curve arithmetic. Our future work will thus to implement our complete protocol using Ethereum Virtual Machine and address receivers anonymity.

Acknowledgment

This research work has been carried out under the leadership of the Institute for Technological Research SystemX, and therefore granted with public funds within the scope of the French Program Investissements d'Avenir.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.

[2] P. Kasireddy, "Fundamental challenges with public blockchains," https://medium.com/@preethikasireddy/fundamental-challenges-with-public-blockchains-253c800e9428, 2017.

[3] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," http://gavwood.com/Paper.pdf, 2014.

[4] T. Okamoto and K. Ohta, "Universal electronic cash," in *CRYPTO*, 1991, p. 324337.

[5] N. van Saberhagen, "Cryptonote v 2.0," Cryptonote, 2013, https://cryptonote.org/whitepaper.pdf.

[6] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," Cryptology ePrint Archive: Report 2014/077, 2014, https://eprint.iacr.org/2014/077.

[7] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," Post on Bitcoin Forum, 2013, https://bitcointalk.org/index.php?topic=279249.

[8] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "CoinShuffle: Practical decentralized coin mixing for bitcoin," in *ESORICS 2014*, 2014, pp. 345–364.

[9] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *IEEE Symposium on Security and Privacy*, 2013, pp. 397–411.

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

[11] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification, version 2016.0-alpha-3.1," github, 2016, https://github.com/zcash/zips/blob/master/protocol/protocol.pdf.

[12] BitcoinWiki, "Zero knowledge contingent payment," 2016, https://en.bitcoin.it/wiki/Zero_Knowledge_Contingent_Payment.

[13] G. Maxwell, "The first successful zero-knowledge contingent payment," BitcoinCore, 2016, https://bitcoincore.org/en/2016/02/26/zero-knowledge-contingent-payments-announcement/.

[14] S. Noether, "Ring signature confidential transactions for Monero," Cryptology ePrint Archive, Report 2015/1098, 2015, https://eprint.iacr.org/2015/1098.

[15] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero," in *ESORICS 2017*, 2017, pp. 456–474.

[16] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge of interactive proof systems," in *ACM symposium on theory of computing*, 1985, pp. 291–304.

[17] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO '86*, 1986, pp. 186–194.

[18] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptography*, vol. 4, no. 3, pp. 161–174, 1991.

[19] M. Franklin and H. Zhang, "A framework for unique ring signatures," Cryptology ePrint Archive, Report 2012/577, 2012, https://eprint.iacr.org/2012/577.

[20] B. França, "Privacy and pruning in the mini-blockchain," http://cryptonite.info/, 2014, http://cryptonite.info/files/Anonymity_account_tree.pdf.

[21] E. B. Hamida, K. L. Brousmiche, H. Levard, and E. Thea, "Blockchain for enterprise: Overview, opportunities and challenges," in *ICWMC 2017*, 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01591859

[22] A. M. Christophe Petit, Michiel Kosters, "Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields," in *In IACR International Workshop on Public Key Cryptography*, 2016, pp. 3–18.

[23] R. Mercer, "Privacy on the blockchain: Unique ring signatures," Cornell University, 2016, https://arxiv.org/abs/1612.01188.

[24] W.-S. Juang, "Ro-cash: An efficient and practical recoverable pre-paid offline e-cash scheme using bilinear pairings," *Journal of Systems and Software*, vol. 83, no. 4, pp. 638–645, 2010.