Computers and Electrical Engineering 83 (2020) 106586

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

A novel credential protocol for protecting personal attributes in blockchain☆

Kalpana Singh[a,*], Omar Dib[a], Clément Huyart[b], Khalifa Toumi[a]

[a] IRT SystemX, 8 Avenue de la Vauve, Palaiseau 91120, Ille de france, France
[b] ERCOM, Vélizy-Villacoublay, France

ARTICLE INFO

Article history:
Received 28 June 2019
Revised 10 January 2020
Accepted 12 February 2020

Keywords:
Blockchain
Online trading System
Hyperledger fabric
User-centric system
Privacy-preserving credential
Short signature
Pairing
Self-blind
Security proofs

ABSTRACT

This paper proposes a novel user-centric and privacy-preserving credential scheme over the blockchain. The proposed protocol allows users to access services without revealing sensitive attributes. This new paradigm is based on an efficient short signature, which uses pairing and self-blindable credentials that are verifiable on the blockchain. Our scheme achieves the advanced features of anonymity, unlinkability, and untraceability of users. Moreover, confidentiality of users' attributes and unforgeability of their credentials are met. We provide security proofs and a real-world use-case where the protocol can be applied. To empirically assess the performance of our solution, the cryptographic components and communications between the various involved actors are implemented using GO and Java. In addition, an implementation for an online trading use case based on Hyperledger Fabric is provided. Finally, we prove the efficiency of our work by presenting some experimental results and exhibiting comparisons with known traditional credentials schemes.

1. Introduction

Nowadays, users are digitally connected for social engagements, businesses, education, e-banking and e-government services. In order to take opportunity of these shared services, users have firstly to be identified by showing their proof of identity. This proof is called a 'credential' such as a passport, driver's licenses, etc. This is an essential system we use today to identify each user. The identification process is an important step for a service provider in order to respect the KYC (Know Your Customer) conditions. It is also a very critical step for a user since he is asked to share his sensitive attribute values with a service provider in order to request a service. A significant component of existing identification approaches is to focus on the use of a certificate authority that serves as a trusted third party, collects and verifies users' attributes in order to deliver their credentials. This classical authentication system easily permits a service provider to check an identity and verify the provided list of attributes. However, these approaches are faced with several issues such as privacy challenges, poor security practices, and single point of failure because of their centralized features [1]. To overcome those shortcomings, recent studies have focused on the use of blockchain technologies. Essentially the blockchain is a technology that promises to shift control over digital activities from *central parties* to *users* [2,3]. A blockchain relies on well-known cryptographic algorithms

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Prof. Khaled Salah.

* Corresponding author.
E-mail addresses: kalpana.singh@irt-systemx.fr (K. Singh), omardib@irt-systemx.fr (O. Dib), clement.huyart@ercom.fr (C. Huyart), khalifa.toumi@irt-systemx.fr (K. Toumi).

https://doi.org/10.1016/j.compeleceng.2020.106586
0045-7906/

to provide key properties such as resistance to tampering, pseudo-anonymity, fault-tolerance, auditability, resilience, data encryption and operational resilience. Due to the high importance of the identity management topic [4,5] and its impact on the privacy of users, several companies and governments are trying to check whether the blockchain can deal with the limitations of traditional systems or not. For instance, in [6], the Estonia government has been working on a universal service based on the blockchain to authenticate users' identities and documents. Despite of these well-known advantages, applications based on the blockchain are still facing diverse challenges related to privacy, confidentiality and efficiency. This paper primarily deals with these issues. Here, by privacy, we mean the 'inability to identify an individual given a set of attribute values and inadequacy to determine by whom, and to what extent data is communicated to others'. We sub categorized privacy definitions into three terms: unlinkability, untraceability and anonymity. These features are defined as, a computationally powerful of an adversary with access to an unbounded number of transactions (same sender) cannot link to the same sender, cannot trace the patterns to sender's identity, and can not guess/know an identity with an advantage while in case of security we focus on confidentiality and unforgeability. By an efficiency term, we refer to the computational time to achieve the verification of users' credential over blockchain.

Ensuring privacy is critical for blockchain based solutions. Moreover, with the arrival of the General Data Protection Regulation (GDPR) privacy has become vital. Certainly, the right to oblivion is extremely complicated in a blockchain due to pseudo anonymity, on-chain ownership features. The ownership and provenance of assets can be easily tracked and particular users may be identified via abstract public addresses [2].

Anonymous credentials [7,8] provide a powerful tool for making assertions about an identity while maintaining privacy. Two well-known privacy-oriented attribute based credential schemes are Idemix [7] and U-Prove [9]. U-Prove is more efficient but does not provide unlinkability; in addition, its security is not fully proven. We detail these schemes in Section 2. However, to date, there is no provably secure scheme that is sufficiently efficient to allow secure implementations on the blockchain, while also providing privacy features (unlinkability, untraceability and anonymity). Therefore, in order to enable all these features, this paper proposes a novel privacy-preserving credential scheme in blockchain. In addition to this, we achieve a publicly verifiable feature on the anonymized/blinded credential of a user. We enable an efficient and user-centric feature by means of pairing-based short signature of modified-ZSS (Zhang, Safavi-Naini and Susilo) scheme [10] and self-blindable scheme by Verheul [11]. Our scheme is mainly proposed to enhance the existing blockchain-based solutions while dealing with users' personal data privacy.

1.1. Contributions of this paper

In this paper, we propose a novel credential protocol for protecting personal attributes in blockchain which achieves an efficient, user-centric and privacy-preserve features. Furthermore, we provide an essential blockchain-based use-case where our scheme can be useful. Summary of our contribution: Our scheme

- provides unlinkability and untraceability for each user, and maintains the confidentiality of users' data
- enables users' anonymity, while preserving an identity verification publicly over the blockchain.
- is the credential scheme providing security proofs with achieving privacy features over the blockchain in an efficient manner compared to the well-known existing schemes.
- retains unforgeability: only blockchain members or legally authorized parties can issue verifiable signatures.
- achieves user-centric: User self-blinds his credential. This credential is still verifiable publicly over the blockchain.
- preserves the correctness of signed and anonymized values, both are generated in a correct way and must be accepted by service providers on the blockchain.

We additionally provide protection in our solution from the following types of attacks:

- *Credential forgery:* Malicious users could interact with other organizations including a certificate provider, in order to forge users' credentials.
- *Compromise user's identity:* Malicious service providers can form an alliance to try to obtain information about a user's identity, by identifying information about the user's different pseudonyms.

1.2. Paper organization

Subsequent sections are arranged as follows. In Section 2, we review the recent works on anonymity, attribute based credential and self-blindable credential schemes. Section 3 provides a technical background for our solution. Section 4 details the components of the proposed scheme, and presents a use case to illustrate our solution. Section 5 describes our scheme. Sections 6 and 7 deal with the security proofs of our protocol and their performance analysis. Finally, conclusions are drawn in the last section and future works are discussed.

2. Related work

This section discusses well known credential management systems. We distinguish between classical approaches that use standard cryptographic techniques to achieve users' privacy and novel approaches based on the blockchain.

2.1. Privacy-preserving attribute-based credentials

Anonymous credentials [8] have proved to be a centrally important building block in privacy-minded identity management systems. The IBM identity mixer (Idemix) credential scheme [7], is probably the most well-known unlinkable attribute-based credential scheme. This scheme can be used for credential issuance and zero-knowledge proofs for attributes verification. This technology also provides an unlinkability. This makes it possible to use a credential multiple times without becoming traceable. However, the performance of this implementation is not better than the well-known U-Prove scheme [9].

The U-Prove issuance and verification protocols are based on Schnorr's blind signature scheme and zero-knowledge proofs respectively. This technology offers the fastest implementation for attributes verification. However, in U-Prove two transactions executed with the same credential are always linkable. Moreover, there is no unforgeability proof for U-Prove credentials.

In [12], an unlinkable scheme based on proof of knowledge of Boneh–Boyen signature was proposed, achieving an efficient scheme with short signatures like Idemix and involving pairings only on the verifier's side. Verheul's self-blindable credentials [11] has proposed a special cryptographic technique enabling the credential structures to be randomised using blinding factors while preserving their verifiability. The use of such credentials is untraceable. To achieve this, users can blind their credentials before they are verified, such that two occurrences of the same credential cannot be recognised. Verheul [11] proposes a variant of the Chaum–Pedersen signature scheme which allows these values to be randomised or blinded [13]. Boneh, Lynn and Shacham (BLS) short signature needs a special hash function. This hash function is probabilistic and generally inefficient. In 2017, S. Ringers, E. Verheul and J. Hoepman (RVH) presented a new self-blindable attribute-based credential scheme [14] and provided a security proof by showing that it is unforgeable as well as unlinkable. This scheme [14] uses elliptic curves and bilinear pairings. They provide a comparison of exponentiation counts and a comparison of run times with the IRMA Idemix implementation and the scheme [15], achieving the same security goals at less cost (CPU time in computing and verifying proofs). However, the confidentiality is a problem in this scheme. The attributes of a user are visible in communication with an issuer and parts of the attributes are disclosed with the verifier in the credential verification step. This is a serious concern if we implement this scheme over blockchain. The interactions between users, issuers and verifiers are interactive, which increase the communication costs.

2.2. Blockchain technology based credentials

In 2015 MIT media lab introduced its digital certificates project using the Bitcoin blockchain,[1] with the goal of making "certificates transferable and more easily verifiable". In [16], a decentralized scheme to issue credentials in the absence of a trusted third party is proposed using Bitcoin. Although those approaches succeeded at proposing decentralized solutions, they remain inefficient due to privacy and scalability issues. Other public blockchain were also used for the same propose. For instance, the uPort [17] is a web identity management system that links an Ethereum address with a name, profile picture and other information like an email address or Twitter account. In [18], a system is also proposed to store user information such as the GPS data from their phone in a distributed hash table and then store pointers to these data and permissions on how they may be used or retrieved on the blockchain. These existing schemes lack the privacy features (defined in anonymous credential schemes) and user-centric. Recently, in [19] a selective disclosure credential scheme and integrated with blockchains is proposed. This scheme has achieved privacy and security features as the paper [14] and also been implemented on Chainspace and Ethereum blockchains. Confidentiality, anonymity and untraceability proofs are not given. Additionally, unforgeability, blindness and unlinkability security proofs are not in detail. To deal with those limitations, we believe that there is a seamless need to propose a novel credential management scheme in the blockchain. Indeed, given the trade-offs of the above systems, our design goal is to generalize the blockchain based credential to permit user-centric, privacy, security features and efficiency during the verification steps while interacting with diverse service providers in the blockchain. Our construction is blockchain agnostic but Hyperledger Fabric [20] blockchain has been used for demonstration purposes.

3. Preliminaries

Our scheme is built on pairings, short signatures, commitment and zero knowledge proof cryptographic schemes. These schemes are built on the elliptic curve cryptography. We give the background details in this section.

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 $P \in E(\mathbb{F}_q)$. Elliptic curves offer efficiently computable bilinear pairings, explain in Section 3.1. We use this setup and notation in subsequent parts of the paper.

[1] https://certificates.media.mit.edu/.

3.1. Bilinear pairing and cryptographic problems

We use the pairing definition from the paper [10] and utilize it in our scheme. Let $\mathbb{G}_1$ be a cyclic additive group generated by P, whose order is a prime $q \in E(\mathbb{F}_q)$ and $\mathbb{G}_2$ be a cyclic multiplicative group with the same order $q \in E(\mathbb{F}_q)$. Let e: $\mathbb{G}_1 \times \mathbb{G}_2$ be a map with the following properties:

1. **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_q$.
2. **Non-degeneracy:** There exists $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$, in other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$;
3. **Computability:** There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

The **Non-degeneracy** is equivalent to $e(P, Q) \neq 1$ for all $P, Q \in \mathbb{G}_1$. So, when P is a generator of $\mathbb{G}_1$, $e(P, P)$ is a generator of $\mathbb{G}_2$. Such a bilinear map is called a bilinear pairing. The bilinear pairing is given by $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. We consider the following problems in the additive group $(\mathbb{G}_1; +)$ with respect to an adversary $\breve{A}$ in Definitions 1–6.

Definition 1. Discrete Logarithm Problem (DLP): $\breve{A}$ has no advantage in solving following: Given two group elements P and Q, find $n \in \mathbb{Z}_q^*$ such that $Q = nP$.

Definition 2. Computational Diffie–Hellman Problem (CDHP): $\breve{A}$ has no advantage in the following: Given P, aP, bP, for $a, b \in \mathbb{Z}_q^*$, compute abP.

Definition 3. Decisional Diffie–Hellman Problem (DDHP): $\breve{A}$ has no advantage in the following: Given P, aP, bP, cP, for $a, b, c \in \mathbb{Z}_q^*$, decide whether $c \equiv ab \ mod \ q$.

Definition 4. Inverse Computational Diffie–Hellman Problem (Inv-CDHP): For $a \in \mathbb{Z}_q^*$, given P, aP, to compute $a^{-1}P$.

Definition 5. Square Computational Diffie–Hellman Problem (Squ-CDHP): For $a \in \mathbb{Z}_q^*$, given P, aP, to compute a^2P.

Definition 6. The Bilinear Diffie–Hellman (BDHP) in $(\mathbb{G}_1, \mathbb{G}_2, e)$ is defined as follows: given (P, aP, bP, cP) for some $a, b, c \in \mathbb{Z}_q^*$, compute $v \in \mathbb{G}_2$ such that $v = e(P, P)^{abc}$.

We use bilinear inverse and bilinear square Diffie–Hellman problems from the paper [10]. Additionally, we use the defined Theorems 1 and 2 from the paper [10].

Theorem 1. *CDHP, Inv-CDHP* and *Squ-CDHP are polynomial time equivalent.*

Theorem 2. *BDHP, BSDHP* and *BIDHP are polynomial time equivalent.*

Our Assumptions *We assume that DLP, CDHP, Inv-CDHP, Squ-CDHP, BDHP, BIDHP and BSDHP are hard, which means there is no polynomial time algorithm to solve any of them with non-negligible probability.*

A Gap Diffie-Hellman (GDH) group is a group which a DDHP is easy but a CDHP is hard in it. From bilinear pairing, we can obtain a GDH group. We use the pairing based short signature scheme of ZSS [10] for self-blindable credential in this paper, which can work on any GDH group. Define a cryptographic hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \{0, 1\}^{\lambda}$, where $mod q mod \geq \lambda \geq 160$.

3.2. Cryptographic building blocks

This section exhibits short signature, non-interactive zero knowledge proofs and commitment cryptographic protocols. Our scheme utilizes these protocols.

3.2.1. A short signature scheme from pairings

This is a well-known signature scheme proposed by the authors F. Zhang, R. Safavi-Naini and W. Susilo (ZSS) [10]. According to our best of knowledge, the ZSS [10] is the most efficient short signature scheme exists in the literature, also validates in the report [21]. The ZSS signature scheme is described as follows:

1. ***ParamGen.*** The system parameters are $\{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H\}$.
2. ***KeyGen.*** Randomly selects $x \in_R \mathbb{Z}_q^*$, and computes $P_{pub} = xP$. The public key is P_{pub}. The secret key is x.
3. ***Sign.*** Given a secret key x, and a message m, computes $S = (H(m) + x)^{-1}P$. The signature is S.
4. ***Verify.*** Given a public key P_{pub}, a message m, and a signature S, verify if $e(H(m)P + P_{pub}, S) = e(P, P)$.

For our scheme, we modify ZSS signature protocol and use in our protocol to construct a self-blindable signature, which allows a user to randomise his key pair, corresponding credential and still be verified using verification equation.

3.2.2. Non-interactive zero knowledge proofs

Zero knowledge proofs (ZKP) [22] allows a prover to convince a verifier that a statement is true without revealing any further information. We use Fiat–Shamir transformation function to convert into non-interactive and then we use non-interactive-Schnorr proof of knowledge (NI-Schnorr PoK) to prove knowledge. S is called the prover and R is called the verifier. The protocol (S, R) must satisfy the properties of 'completeness' and 'soundness'. Let $\mathbb{G}$ be a group of prime order q with generator P, and let $\mathbb{Z}_q$ denote the field of integers modulo q.

Statement Prover S knows a v (as secret key), and calculates public key: $y = P^v$.

Public information: y, P. Private information: v

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

We use this scheme in elliptic curve setting in proof of correctness on committed data over the blockchain.

3.2.3. Commitment scheme

A commitment protocol [23] involves two parties: the committer and the receiver. It consists of two stages, a 'commitment phase' and a 'reveal phase'. The public parameters are a group $\mathbb{G}$ of prime order q, P be a generator of G_q, the unique order-q subgroup of $\mathbb{Z}_p^*$. We use $x \in_R \mathbb{Z}_q$ to denote that x is uniformly randomly chosen from $\mathbb{Z}_q$. User picks $x \in_R \mathbb{Z}_q$ and computes $h = (P^x mod p)$. User keeps the value x secret and makes the values p, q, P, h public.

- *Commit* $C = commit(a, r)$: In order to commit a value $a \epsilon \mathbb{Z}_q$, the user chooses $r \in_R \mathbb{Z}_q$ and computes the commitment $C = (P^a \cdot h^r mod p)$.
- *Reveal:* To open a commitment C, the user reveals a and r, and a verifier verifies whether $C = (P^a \cdot h^r mod p)$.

The commitment scheme is **unconditionally hiding:** Even with unlimited computational power it is impossible for an adversary to learn any information about the value a from C, because the commitments of any two numbers in $\mathbb{Z}_q$ have exactly the same distribution. This scheme is **computationally binding:** Under the DL assumption, it is computationally infeasible for an adversarial committer to open a value a' other than a in the open phase of the commitment scheme. We also use elliptic curve setting in the commitment scheme.

4. Our system architecture

In this section, we present our credential scheme architecture from a high-level point of view. Additionally, we provide a use case where the usage of the proposed solution is relevant. Technical explications related to our proposed scheme are provided later in Section 5.

4.1. System architecture description

As already mentioned, the objective of this paper is to allow a user for protecting his personal attributes and activities in a blockchain-based technology along with efficiency. To begin with, our system architecture is built on a blockchain technology and involves four entities: Identity Validator $\mathcal{IV}$, Certificate Provider $\mathcal{CP}$, User $\mathcal{U}$ (also called as credential holder) and Service Provider $\mathcal{SP}$ (also called as credential verifier).

(i) **User** $\mathcal{U}$**:** A person who wants to benefit from services proposed by service providers in the blockchain. A user has sensitive identity attributes (e.g. name, age, proof of address, etc.). So, he demands for minimizing their exchange and better controlling their usage in the blockchain. In addition, a user may ask for more privacy so that his activities (e.g. transactions) in the blockchain cannot be traced and linked to his real identity regardless of the public nature of the blockchain.

(ii) **Identity validator** $\mathcal{IV}$**:** This entity is responsible for validating the identity attributes of a user with the help of physical documents. It is a legal entity (i.e. a government office, a bank, etc.), which has the right to verify if the person is the one who is claiming to be. Here, we assume the list of identity validators is known for everyone in the blockchain. These validators are supposed to be certified and trusted entities.

(iii) **Certificate provider** $\mathcal{CP}$**:** This entity provides a credential to a user for each service after verifying a signature of an identity validator on user's anonymized attributes and proof of anonymized attributes. This entity also handles the security and privacy aspects. Using this credential, a user can access blockchain services in order to perform some actions such as online trading.

(iv) **Service provider** $\mathcal{SP}$**:** This entity provides services to each certified and interested user on the blockchain. $\mathcal{SP}$ verifies each user's credential who sends a request to access services on-chain. $\mathcal{SP}$ verifies each credential without seeing the attribute values and without knowing the identity of a user. Usually, this entity requires that each user fulfills some constraints (age $>$ 18, living address in a specific city, etc.).

Our solution has two different communications: Off-chain and On-chain. The communication that is executed without storing any information in the blockchain ledger is called off-chain. In an adverse, on-chain communication usually requires

an interaction with the blockchain ledger in order to read or write data. In the architecture, we categorize the communications between parties into three steps: (1) Signature issuing on committed attribute values by $\mathcal{IV}$ to $\mathcal{U}$, (2) Credential issuance to $\mathcal{U}$ from $\mathcal{CP}$, and (3) Credential presentation by $\mathcal{U}$ without identification and verification publicly by $\mathcal{SP}$. The first and second steps are off-chain and third is on-chain.

1. **Signature issuance on commitment:** Firstly, $\mathcal{IV}$ checks the physical document of $\mathcal{U}$. $\mathcal{U}$ commits these values and gets a signature of $\mathcal{IV}$ to authenticate on the blockchain.
2. **Credential issuance:** $\mathcal{U}$ sends a signed-committed attribute values and NI-Schnorr PoK of commitment to $\mathcal{CP}$. $\mathcal{CP}$ verifies the signature of $\mathcal{IV}$ and the proof of knowledge of commitment. Then, $\mathcal{CP}$ signs using modified-ZSS on the committed values and sends it to $\mathcal{U}$.
3. **Credential presentation:** $\mathcal{U}$ self-blinds the credential (issued from $\mathcal{CP}$), then authenticates to $\mathcal{SP}$ on-chain. $\mathcal{SP}$ verifies whether $\mathcal{U}$ has a credential from $\mathcal{CP}$ or not. If the verification is correct then $\mathcal{U}$ gets a permission to access a corresponding service, otherwise $\mathcal{SP}$ aborts the connection.

4.2. Signature issuance on commitment: $\mathcal{U} \Longleftrightarrow \mathcal{IV}$

In order to allow $\mathcal{U}$ interacting with the blockchain (i.e. send read and write transactions), firstly, we propose that $\mathcal{U}$ communicates with a trusted $\mathcal{IV}$. As already mentioned, this is done off-chain. Indeed, this step is primal in order to validate the identity attributes of $\mathcal{U}$ and to verify some general claims related to him such as if he is greater than 18 years old or if he is living in a specified city etc. This phase generally includes a physical meeting in order to allow $\mathbb{IV}$ to check the authenticity of user's attributes. To perform this step, $\mathcal{IV}$ makes a comparison between $\mathcal{U}$'s identity attributes, claims, and some official documents provided by $\mathcal{U}$ such as identity card, proof of address, driver licence, etc. Once the verification is done, $\mathcal{U}$ uses cryptographic algorithms: (**1**) $\mathcal{U}$ commits on his verified claims and attributes, and (**2**) $\mathcal{IV}$ signs on $\mathcal{U}$'s commitment using the classical *ECDSA* (see in Fig. 2) signature.

4.3. Credential issuance: $\mathcal{U} \Longleftrightarrow \mathcal{CP}$

To access services in a blockchain, $\mathcal{U}$ communicates with $\mathcal{CP}$ in order to check if he is eligible and to get a valid credential. Traditionally speaking, $\mathcal{CP}$ checks the identity attributes of each $\mathcal{U}$ in order to issue a credential. Obviously, this does not match the identity protection feature required by $\mathcal{U}$. To overcome this, $\mathcal{U}$ shows his signed and valid commitment provided by $\mathcal{IV}$ (Fig. 3) and sends NI-Schnorr PoK of all values to compute this commitment to avoid its usage by anyone in the network. By doing so, $\mathcal{CP}$ checks the signature of a trusted $\mathcal{IV}$ and NI-Schnorr PoK of commitment on attribute values. Additionally, $\mathcal{CP}$ can verify a specific type of claim. For instance, $\mathcal{CP}$ can check if $\mathcal{U}$ is greater than 18 years old by verifying that his claim is signed by $\mathcal{IV}$. After verification, a credential is issued to $\mathcal{U}$ whereby he can communicate with various service providers on the blockchain.

4.4. Credential presentation: $\mathcal{U} \Longleftrightarrow \mathcal{SP}$

The proposed protocol provides an additional privacy feature to protect $\mathcal{U}$'s activities on the blockchain by using self-blinding scheme (see in Fig. 4). For instance, in a trading system, users may require anonymized and untraced transactions (i.e. private buying, selling transactions etc). To achieve this, we use a modified ZSS short signature [10] and self credential Verheul's [11] schemes in order to blind the credential and $\mathcal{U}$'s keys from the original $\mathcal{U}$'s credential that is provided by $\mathcal{CP}$. This blind credential is still verifiable and holds the right signature of $\mathcal{CP}$. $\mathcal{U}$ uses a blind credential in communication over the blockchain. Thus, no one can trace the actions of $\mathcal{U}$ over the blockchain, and his identity attributes remain protected.

In case of mitigation, $\mathcal{SP}$ can ask to $\mathcal{CP}$ to reveal the identity attributes of $\mathcal{U}$. $\mathcal{CP}$ sends a request to $\mathcal{IV}$ in order to reveal the real identity of $\mathcal{U}$. In this way, we manage the control of $\mathcal{U}$'s identity attributes from by both $\mathcal{U}$ and the trusted $\mathcal{IV}$. The processing steps of each entity is depicted in Fig. 1.

4.5. Relevant use case

This section presents a use case where the application of our proposed scheme is relevant.

Use case: Online trading. In this use case, we assume that we have an online trading platform based on a consortium blockchain with multiple $\mathcal{SP}$s to propose the trading services. A blockchain can be used for such use case in order to allow multiple trading providers which do not trust each other to coexist in the same platform, for example, cryptocurrency trading system (CTS). Additionally, a blockchain can securely trace all users' transactions in a decentralized manner. Let us assume that our trading system is a CTS. So, If you wish to invest your fiat currency (such as US Dollars) into a crypto asset, you will need to set up an online account through an online platform. Once you have set up an account on a cryptocurrency exchange, you will need to send coins to this account to start trading. Cryptocurrency Trading platforms (CTPs) are market places that bring together different cryptocurrency users that are either looking to buy or sell coins, providing them with a platform on which they can directly trade with each other. In this trading platform users offer exchange services to cryptocurrency users. They allow cryptocurrency users to sell their coins for fiat currency or buy new coins with fiat currency.

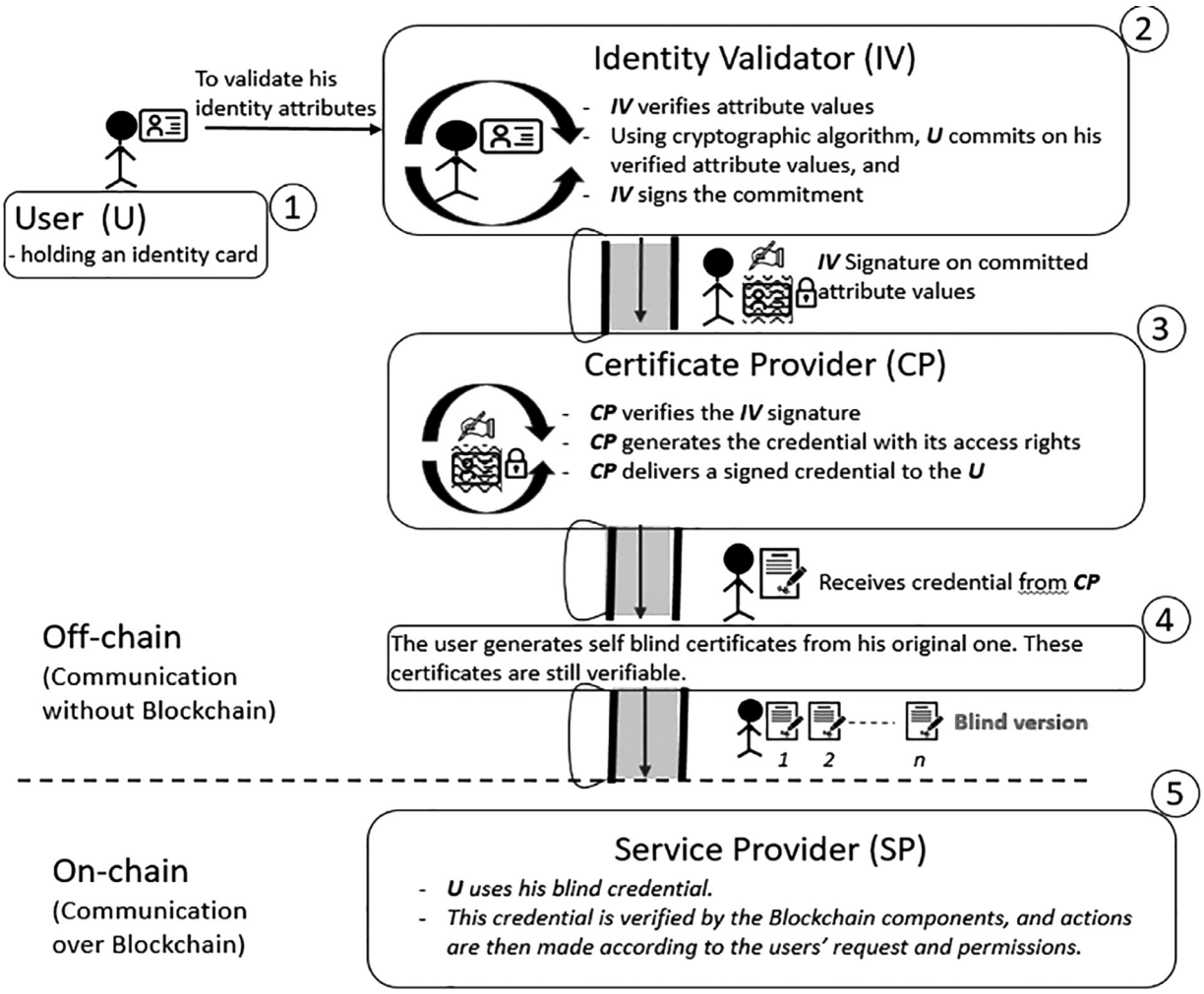
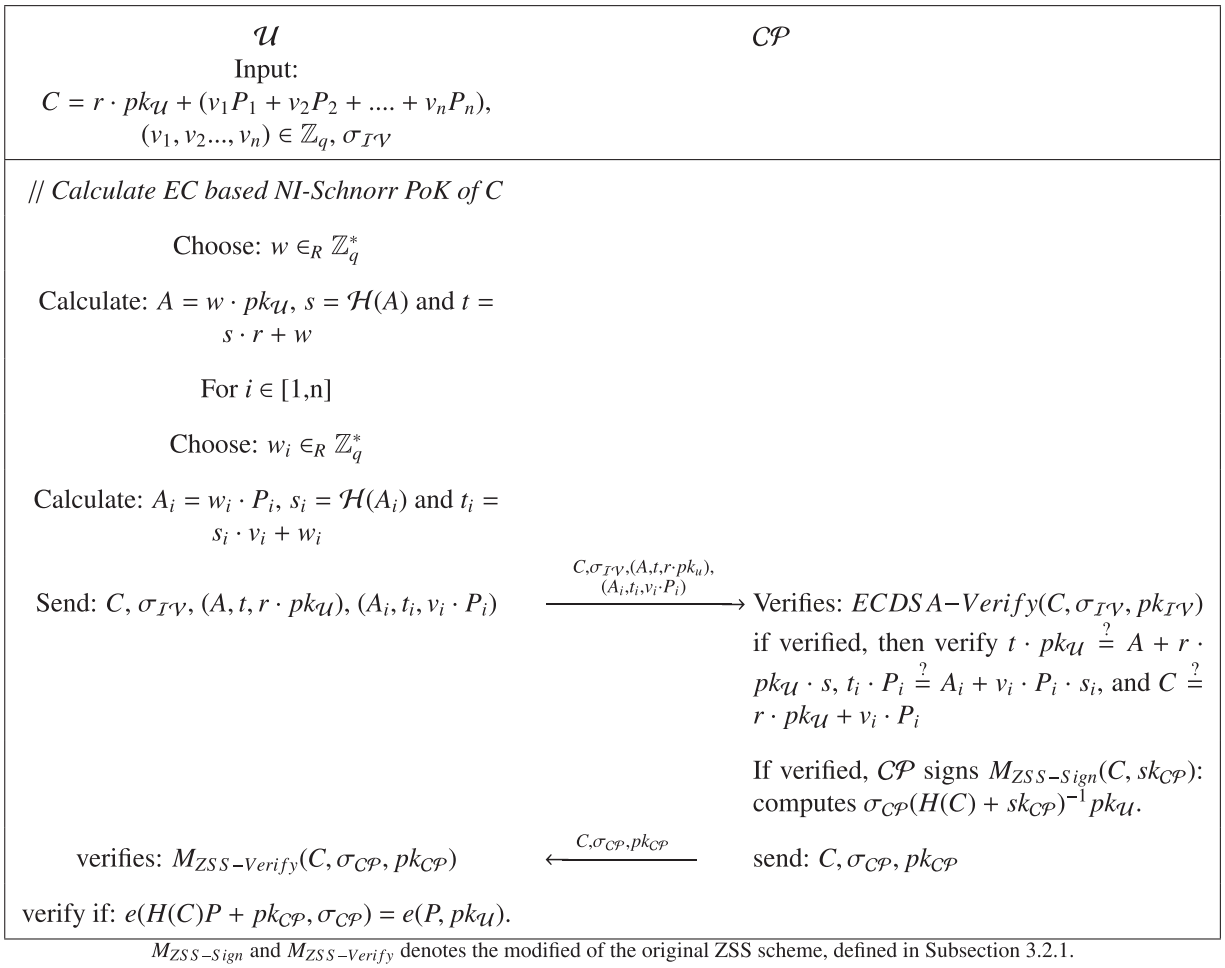


Fig. 1. Oursystem architecture.

$\mathcal{U}$		$\mathcal{IV}$
$(v_1, v_2..., v_m) \in \mathbb{Z}_q$, $r \in_R \mathbb{Z}_q$		
$C = r \cdot pk_{\mathcal{U}} + (v_1P_1 + v_2P_2 + ... + v_nP_n)$		
commitment: C, r	$\xrightarrow{C,r}$	$\sigma_{\mathcal{IV}} \leftarrow ECDSA - Sign(C, sk_{\mathcal{IV}})$
verifies: $ECDSA - Verify(C, \sigma_{\mathcal{IV}}, pk_{\mathcal{IV}})$ and stores $\sigma_{\mathcal{IV}}, C$ for $\mathcal{CP}$	$\xleftarrow{\sigma_{\mathcal{IV}}, pk_{\mathcal{IV}}}$	$\sigma_{\mathcal{IV}}, pk_{\mathcal{IV}}$

Fig. 2. Signature issuance on commitment protocol.

To entail our system architecture with this trading platform, user U who wants to trade with this platform. $\mathcal{SP}$ is an entity who provide this exchange platform. A $\mathcal{CP}$ is another $\mathcal{SP}$ who deal with credentials of each user who participate in this trading platform. To start trading inside the platform, we assume that it is required for any new $\mathcal{U}$ to prove that he is greater than 18 years old. Unfortunately, by using traditional verification mechanisms, the online verification of the identity attributes of $\mathcal{U}$ becomes a very tedious and inefficient task. Indeed, each new $\mathcal{U}$ provides a copy of his identity card. Thus, the CTP will have a full copy of $\mathcal{U}$s' identity attributes, which adds additional effort to protect these data and meet the GDPR requirements. Additionally, a malicious $\mathcal{U}$ can forge the attributes and identity card itself, provide a wrong identity



$M_{ZSS-Sign}$ and $M_{ZSS-Verify}$ denotes the modified of the original ZSS scheme, defined in Subsection 3.2.1.

Fig. 3. Credential issuance protocol.

card, falsify his identity card and modify his age, and use the identity card of another person, etc. This classical credential paradigm is therefore inefficient for both users who are seeking privacy and easiness, and service providers who require not to compromise users' privacy.

The usage of our proposed scheme in the use case: To overcome the drawbacks of the above mentioned credential system, our protocol does the following steps. Firstly, each $\mathcal{U}$ physically goes to a certified and trusted $\mathcal{IV}$ in order to get his age proof. $\mathcal{IV}$ verifies the identity card corresponding to $\mathcal{U}$ and then checks if $\mathcal{U}$ is greater than 18. In this use case $\mathcal{IV}$ is a bank. If so, $\mathcal{U}$ commits on his age attribute value and $\mathcal{IV}$ signs his commitment. $\mathcal{U}$ receives a signature on committed attribute values by $\mathcal{IV}$. Now, $\mathcal{U}$ connects to a trading platform and requests a credential from $\mathcal{CP}$. In cryptocurrency trading system, a $\mathcal{SP}$ could be a $\mathcal{CP}$. $\mathcal{U}$ sends his signed committed age value to a $\mathcal{CP}$. Here, $\mathcal{CP}$ is a one of the $\mathcal{SP}$ in cryptocurrency trading system. $\mathcal{CP}$ checks the signature of $\mathcal{IV}$. If the signature is authorized, $\mathcal{CP}$ signs on age committed value using the protocol (see in Fig. 3) and sends to $\mathcal{U}$. However, $\mathcal{CP}$ does not know the real age of $\mathcal{U}$. $\mathcal{U}$ self-blinds this credential using our protocol (see Fig. 4) and authorizes on the blockchain. This credential does not contain any personal information except a signed proof that he is greater than 18 and still verifies the credential as a legitimate $\mathcal{U}$. In case of a successful verification, $\mathcal{U}$ can start his trades over the blockchain. Due to the use of blinded credential, $\mathcal{U}$ achieves the privacy features and $\mathcal{SP}$ can not trace/link $\mathcal{U}$. The proofs of security are available in Section 6. In this use case, we identify that $\mathcal{U}$'s attributes are protected along the whole $\mathcal{U}$'s story and his trading actions are also anonymized. Along with, the trading $\mathcal{SP}$s are also satisfied since they do not store any attribute value in the verification step. Our solution may also be relevant to other use cases such as Blockchain-based distributed marketplace and data access management system.

5. Our protocols

We are now ready to present our construction based on the building blocks and system architecture introduced in Sections 3.2 and 4 respectively. Our full architecture includes several users, all accessing any nodes on the blockchain. To simplify the explanations, we present only one user, depict in Fig. 1.

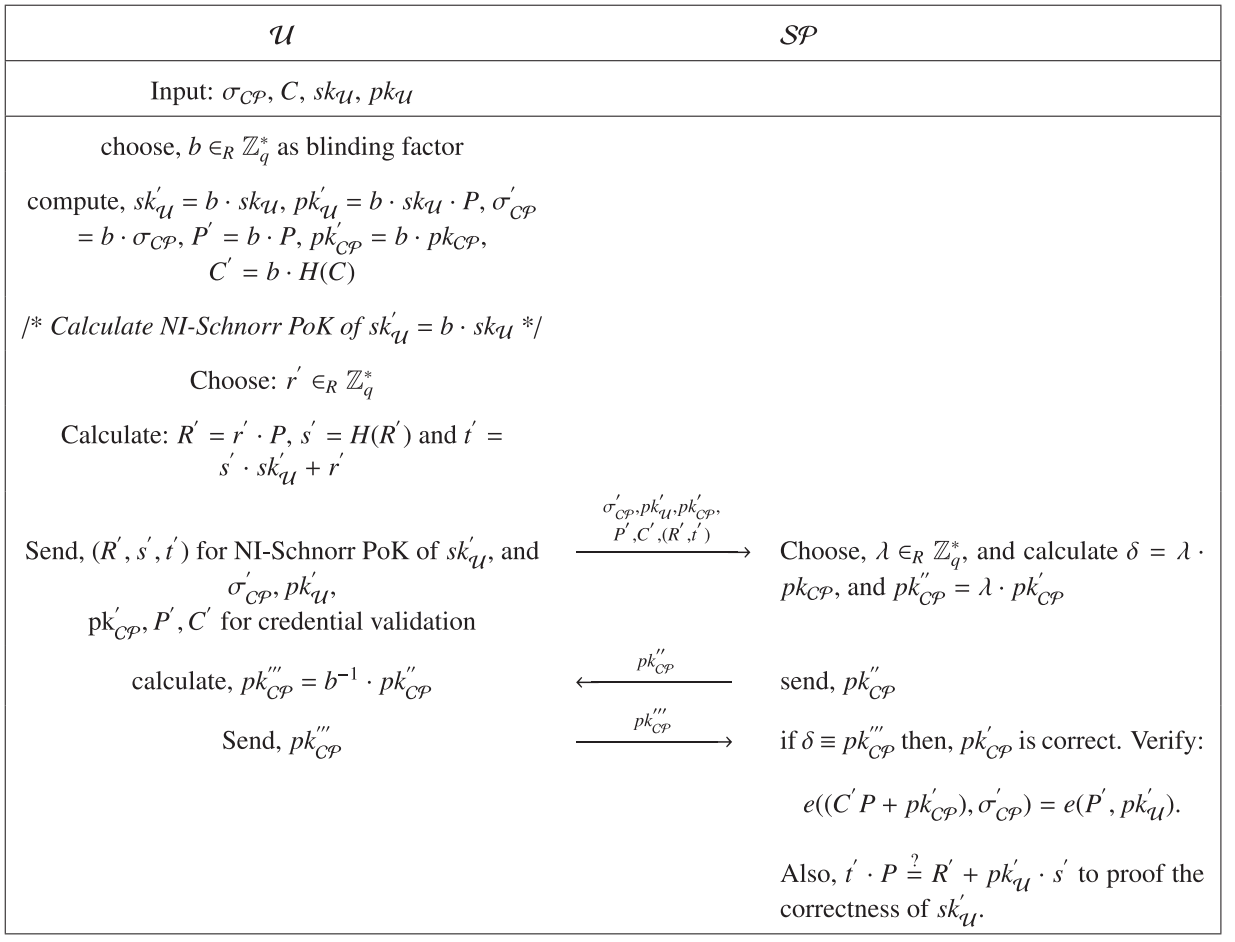


Fig. 4. Credential presentation protocol.

In our case, each user $\mathcal{U}$ has verified accounts with $\mathcal{IV}$ such as Bank (demonstrates in Section 4), where $\mathcal{IV}$ has made verification of each essential document. The document is the set of attributes $attrs = a_1, a_2, ..., a_m$ and v_1, v_2, ..., v_m to denote the corresponding m attribute values. These attributes are the essential lists for us in *Credential Issuance* step. In *Credential Issuance* step, $\mathcal{U}$ receives a credential from a $\mathcal{CP}$. $\mathcal{CP}$ is an untrusted entity for $\mathcal{U}$. Therefore, $\mathcal{U}$ does not reveal any attributes to $\mathcal{CP}$. We use elliptic curve based commitment protocol (defined in Section 3) to hide the attribute values v_1, v_2, ..., v_m from $\mathcal{CP}$. The obvious questions is if $\mathcal{U}$ does not reveal any v_1, v_2, ..., v_m to $\mathcal{CP}$ even $\mathcal{U}$ sends a cryptographic proof of committed attribute values to $\mathcal{CP}$. How $\mathcal{CP}$ checks the values are authorized or not? For example: If $\mathcal{U}$ wants to participate in an online game. He registers himself by providing attributes such as profession 'student', age '21', gender 'male', city 'London'. During *credential issuance* step, $\mathcal{U}$ provides his malicious information by providing wrong values of each attribute such as profession 'employee', age '29', gender 'female', city 'Luxembourg', and still $\mathcal{CP}$ is able to verify the correctness and issues a credential to $\mathcal{U}$. To prevent these types of fraud. $\mathcal{U}$ has verified attribute values and signature from an authorized $\mathcal{IV}$ on the $v_1, v_2, \ldots, v_m$ to get a credential in our protocol. We use the notation from Section 3 in subsequent parts of the paper.

Key Generation Set-up for $\mathcal{IV}$, $\mathcal{U}$, $\mathcal{CP}$, ***and*** $\mathcal{SP}$. We use the elliptic-curve setup and notation from Section 3. $\mathcal{U}$ generates a secret random number $sk_{\mathcal{U}} \in_R \mathbb{Z}_q$ and then performs an elliptic curve scalar multiplication to get the corresponding public key $pk_{\mathcal{U}} = sk_{\mathcal{U}} \cdot P$. Similarly, $\mathcal{IV}$, $\mathcal{CP}$, $\mathcal{SP}$ generates the key pairs $(sk_{\mathcal{IV}}, pk_{\mathcal{IV}})$, $(sk_{\mathcal{CP}}, pk_{\mathcal{CP}})$, and $(sk_{\mathcal{SP}}, pk_{\mathcal{SP}})$ respectively. Now $sk_{\mathcal{U}}$, $sk_{\mathcal{IP}}$, $sk_{\mathcal{CP}}$, and $sk_{\mathcal{SP}}$ are secret keys for $\mathcal{U}$, $\mathcal{IV}$, $\mathcal{CP}$, and $\mathcal{SP}$ respectively.

5.1. Signature issuance on commitment protocol: $\mathcal{U} \Longleftrightarrow \mathcal{IV}$

This is an off-chain communication protocol. In this step, $\mathcal{IV}$ checks all attributes $attrs = a_1, a_2, ..., a_m$ and corresponding values v_1, v_2, ..., v_m of $\mathcal{U}$, and registers $\mathcal{U}$ as an authorized member. $\mathcal{U}$ requests a signature from $\mathcal{IV}$ on the verified but committed attribute values to authenticate anonymously by $\mathcal{CP}$ to join the blockchain. $\mathcal{U}$**'s Setup:** (From the Section 3), Let a cyclic group $\mathbb{G}$ of prime order q with generators P_0, P_1, P_2, ..., $P_n \in E(\mathbb{F}_{11})$. From above $\mathcal{U}$'s Key Generation Set-up: secret

Table 1
Comparison of privacy-preserving credential schemes.

Parameters → Schemes ↓	credent-ial presentation	Unlink-able	Unforge-able	Untrace-able	Confiden-tial	Block-chain	User-Centric	Self-blindable Scheme
U-Prove [9]	$\mathcal{ECC}$	✗	Not proved	Not proved	✗	✗	√	✗
[14]	$\mathcal{ECC}$	√	√	Not proved	✗	✗	√	√
Idemix[25]	$\mathcal{RSA}$	√	√	√	√	√	✗	✗
[19]	$\mathcal{ECC}$	√	√	Not proved	Not proved	√	Not proved	✗
Our scheme	$\mathcal{ECC}$	√	√	√	√	√	√	√

Credential presentation protocol (CPP) of several attribute-based credential schemes. $\mathcal{ECC}$, $\mathcal{BP}$ and $\mathcal{RSA}$ show the protocol based in elliptic curves, target group of a bilinear pairing and RSA groups respectively. **confidentiality** in U-Prove does not achieve due to disclosing attributes in the CPP. **confidentiality** in [14] paper: No confidentiality with issuer and conditional confidentiality with verifier. Column **blockchain** delineates scheme has/has not been implemented over blockchain.

key $sk_{\mathcal{U}}$ and public key $pk_{\mathcal{U}} = sk_{\mathcal{U}} \cdot P_0$. Output $(E(\mathbb{F}_{\prime\prime}), q, (P_0, P_1, P_2, ..., P_n), pk_{\mathcal{U}}))$. We use ECDSA algorithm [24] for signature of $\mathcal{IV}$. As the ECDSA digital signature algorithm is a standard cryptographic algorithm in the blockchain, it has been our first choice for implementing $\mathcal{IV}$ signature. We denote key generation, signature and verification of ECDSA algorithms for $\mathcal{IV}$: $ECDSA - KeyGen$, $ECDSA - Sign$ and $ECDSA - Verify$ respectively, and use these notations for Fig. 2. $\mathcal{U}$ receives $ECDSA - Sign$ on committed attribute values C from $(\mathcal{IV})$ to anonymously authenticate in blockchain.

5.2. Credential issuance protocol: $\mathcal{U} \Longleftrightarrow \mathcal{CP}$

This communication step is on off-chain. We can also perform this protocol over the blockchain. In this protocol, $\mathcal{U}$ authenticates anonymously to $\mathcal{CP}$ as $\mathcal{U}$ has $\mathcal{IV}$ signature on committed values C (defined in Fig. 2). $\mathcal{U}$ sends the NI-Schnorr PoK of committed values C using elliptic curve based NI-Schnorr PoK (defined in Section 3.2). $\mathcal{CP}$ receives a signed committed value and NI-Schnorr PoK of committed attributes values. $\mathcal{CP}$ verifies the signature and proof of correctness of committed values. The protocol is delineated in Fig. 3. **Setup:** We use the elliptic curve (EC) setup defined in Section 3 to generate elliptic curve based NI-Schnorr PoK. Let a cyclic group $\mathbb{G}$ of prime order q with a generator $P \in E(\mathbb{F}_{\prime\prime})$. Recall the setup of EC based commitment from Section 5.1 and Key Generation Set-up of $\mathcal{U}$. $\mathcal{U}'$ssecret key: $sk_{\mathcal{U}} \in_R \mathbb{Z}_q$, and public key: $pk_{\mathcal{U}} = sk_{\mathcal{U}} \cdot P$. The verification works because of the following equations:

$$\begin{aligned} e(H(C)P + pk_{\mathcal{CP}}, \sigma_{\mathcal{CP}}) &= e((H(C)P + sk_{\mathcal{CP}}P), (H(C) + sk_{\mathcal{CP}})^{-1} pk_{\mathcal{U}}) = e((H(C) + sk_{\mathcal{CP}})P, (H(C) + sk_{\mathcal{CP}})^{-1} pk_{\mathcal{U}}) \\ &= e(P, pk_{\mathcal{U}})^{(H(C)+sk_{\mathcal{CP}})\cdot(H(C)+sk_{\mathcal{CP}})^{-1}} = e(P, pk_{\mathcal{U}}) \end{aligned}$$

5.3. Credential presentation protocol: $\mathcal{U} \Longleftrightarrow \mathcal{SP}$

This protocol is implemented over the blockchain. In this protocol, we modify the version of *pairing based short signature scheme of ZSS [10]* to blind credential. We utilize this scheme as this is the most efficient scheme in the literature. From the literature, we find the elliptic curve based signature scheme can be used to implement a credential scheme which allows the users themselves to blind their credentials in order to prevent linkability and traceability. We self-blinds this signature scheme using Verheul [11] approach to get self-blind credential. Since $\mathcal{U}$ can perform this blinding all by itself, Verheul [11] calls these signatures self-blindable. Firstly, $\mathcal{U}$ blinds its secret and public keys, and credential, which is received from $\mathcal{CP}$. $\mathcal{U}$ sends the results from this blinding operation to $\mathcal{SP}$ and proves knowledge of the (blinded) private key using elliptic curve based NI-Schnorr PoK. The resulting blind values are still be verified using the verification equation by $\mathcal{SP}$. Hence the signature remains valid. The protocol is presented in the Fig. 4. The verification works because of the following equations:

$$\begin{aligned} e(C^{'}P + pk^{'}_{\mathcal{CP}}), \sigma_{\mathcal{CP}'}) &= e(b \cdot (H(C)P + b \cdot pk_{\mathcal{CP}}), \sigma_{\mathcal{CP}'}) = e(b \cdot (H(C)P + b \cdot sk_{\mathcal{CP}}P), b \cdot (H(C) + sk_{\mathcal{CP}})^{-1} pk_{\mathcal{U}}) \\ &= e(b \cdot P(H(C) + sk_{\mathcal{CP}}), b \cdot pk_{\mathcal{U}}(H(C) + sk_{\mathcal{CP}})^{-1}) \\ &= e(P^{'}(H(C) + sk_{\mathcal{CP}}), pk^{'}_{\mathcal{U}}(H(C) + sk_{\mathcal{CP}})^{-1}) \\ &= e(P^{'}, pk^{'}_{\mathcal{U}})^{(H(C)+sk_{\mathcal{CP}})\cdot(H(C)+sk_{\mathcal{CP}})^{-1}} = e(P^{'}, pk^{'}_{\mathcal{U}}) \end{aligned}$$

We compare our scheme with well-known papers in terms of privacy, security, and efficiency in Table 1.

6. Proof of security

This section analyses the privacy and security of our protocol. The privacy of the protocol is determined by the user's anonymity, unlinkability, and untraceability. The security is characterized by unforgeability and confidentiality. In addition to this, we verify: (1) user is anonymous but verifiable identities (2) self-blinded but verifiable credential of a user. We define privacy and security features by means of a game between a challenger $\tilde{C}$ and an adversary $\tilde{A}$, the latter who may be a user $\mathcal{U}$, a certificate provider $\mathcal{CP}$/service provider $\mathcal{SP}$. We define the advantage of $\tilde{A}$ in breaking the user anonymity of the protocol as the probability that $\tilde{A}$ guesses private information correctly.

Definition 7 (Adversary with an Advantage)**.** The protocol has the user anonymity if the advantage of any probabilistic polynomial time (PPT) adversary is not more than $\frac{1}{2}$ plus any non-negligible value.

6.1. Privacy and security lemmas and proofs

The privacy lemmas are illustrated in Lemmas 1 and 2. The security lemmas are explained by these Lemmas 3 and 4. These lemmas are essential to prove the security proofs of our protocol.

Lemma 1 (User Anonymity)**.** *A PPT adversary $\check{A}$ with access to an unbounded number of chosen-message signatures/credentials produced by $\mathcal{CP}$/accessed by $\mathcal{SP}$ to the same $\mathcal{U}$ cannot guess his identity with any advantage.*

Lemma 2 (Unlinkability and Untraceability)**.** *A PPT adversary $\check{A}$ with access to an unbounded number of chosen-message credentials accessed by $\mathcal{SP}$ for the same $\mathcal{U}$ cannot link and trace the credentials to $\mathcal{U}$ with any advantage.*

Lemma 3 (Confidential transaction)**.** *A PPT adversary $\check{A}$ with access to the committed data C (generated by $\mathcal{U}$) can not reveal (almost) no information about C with any advantage.*

Lemma 4 (Unforgeability)**.** *Define this security notion by considering $\mathcal{U}$ is a PPT adversary $\check{A}$. If $\mathcal{U}$ sends self generated credential rather than self-blindable of received credential from $\mathcal{CP}$. Then credential verification would never work and $\mathcal{SP}$ would abort a request.*

Next, we present security proofs of our protocols.

Theorem 3. *Assume that a commitment protocol has unconditionally hiding of attribute values i.e., a $\mathcal{CP}$ does not learn anything about values and $sk_{\mathcal{U}}$ during a commit step. In this case, our protocol achieves user anonymity from $\mathcal{CP}$ and confidentiality during a transaction. Assume that a self-blinding credential has the anonymity feature. At $\mathcal{SP}$ end, a user sends blind credential and blind keys. Hence, user achieves anonymity from $\mathcal{SP}$.*

Proof. The security proof is divided into two parts:

case#1: User achieves anonymity and confidentiality from $\mathcal{CP}$

In Section 5.2, Fig. 3, $\mathcal{U}$ uses EC-commitment scheme and generates EC based NI-Schnorr PoK of committed data. $\mathcal{U}$ sends committed value $C = r \cdot pk_{\mathcal{U}} + (v_1P_1 + v_2P_2 + ... + v_nP_n)$, NI-Schnorr PoK of C, and $\sigma_{\mathcal{IV}} \leftarrow ECDSA - Sign(C, sk_{\mathcal{IV}})$ to $\mathcal{CP}$ for user anonymous authentication and proof of correctness of C. The $\sigma_{\mathcal{IV}}$ is used for anonymous authentication. Because, $\mathcal{U}$ does not include original attribute values in this step. During verification step, $\mathcal{CP}$ verifies the signature of $\mathcal{IV}$ without seeing $\mathcal{U}$'s attribute values and proof of correctness of committed values C. We consider any PPT adversary $\check{A}$ considers a cyclic group $\mathbb{G}'$ of prime order q' with generators $P_0', P_1', P_2', ..., P_n' \in E(\mathbb{F}_{\mu'})$. $\check{A}$ generates a committed attribute values by considering $v_1', v_2', ..., v_n' \in \mathbb{Z}_q$, $r' \in \mathbb{Z}_q$. $\check{A}$ Calculates EC based NI-Schnorr PoK of C'. Firstly, choose $w' \in_R \mathbb{Z}_q^*$, and calculates $\tilde{A} = r' \cdot pk_{\mathcal{U}'}$, $\tilde{s} = \mathcal{H}(\tilde{A})$, and $\tilde{t} = \tilde{s} \cdot r' + w'$ Then, $\check{A}$ calculates: For $i \in$ [1,n], Choose $w_i' \in_R \mathbb{Z}_q^*$, and calculates $\tilde{A}_i = w_i' \cdot P_i'$, $\tilde{s}_i = \mathcal{H}(\tilde{A}_i)$, and $\tilde{t}_i = \tilde{s}_i \cdot v_i' + w_i'$. The tuples: $(\tilde{A}, \tilde{t})$ and $(\tilde{A}_i, \tilde{t}_i)$ are required to verify NI-Schnorr PoK of C' by $\mathcal{CP}$. $\mathcal{CP}$ does verification steps as in the Fig. 3:

$$(t \cdot pk_{\mathcal{U}} \stackrel{?}{=} A + r \cdot pk_{\mathcal{U}} \cdot s) \neq (\tilde{t} \cdot pk_{\mathcal{U}'} \stackrel{?}{=} \tilde{A} + r' \cdot pk_{\mathcal{U}'} \cdot \tilde{s}), (t_i \cdot P_i \stackrel{?}{=} A_i + v_i \cdot P_i \cdot s_i) \neq (\tilde{t}_i \cdot P_i' \stackrel{?}{=} \tilde{A}_i + v_i' \cdot P_i' \cdot \tilde{s}_i), and$$
$$(C \stackrel{?}{=} r \cdot pk_{\mathcal{U}} + v_i \cdot P_i) \neq (C' \stackrel{?}{=} r' \cdot pk_{\mathcal{U}'} + v_i' \cdot P_i')$$

We achieve that the $\check{A}$'s calculated equations do not validate the authorized verification steps, which needs an original committed and secret key values r and $sk_{\mathcal{U}}$ as delineated in the Figs. 2 and 3. Hence, $\check{A}$ can not generate the authorized parameters for NI-Schnorr PoK of C. $\mathcal{CP}$ never accepts a bad NI-Schnorr PoK of C and commitments reveal no information whatsoever about the committed values. Thus, we achieve perfectly hiding commitments. $\mathcal{U}$'s anonymity is also maintained as $\mathcal{CP}$ only sees the signature on committed values. We just proved that commitment achieves perfectly hiding.

case#2: user achieves anonymity from $\mathcal{SP}$

In Fig. 4, Section 5.3, $\mathcal{U}$ sends $\sigma_{\mathcal{CP}}'$, $pk_{\mathcal{U}}'$, $pk_{\mathcal{CP}}'$, $P', C', (R', t')$ to $\mathcal{SP}$ for verifying the blinded credential.

We consider any PPT adversary $\check{A}$ with the same setup as above exhibited for user anonymity and confidentiality with $\mathcal{CP}$ communication. $\check{A}$ chooses $\psi \in_R \mathbb{Z}_q^*$, and calculate $\Psi = \psi \cdot pk_{\mathcal{CP}}$ and $p\tilde{k}_{\mathcal{CP}} = \psi \cdot pk_{\mathcal{CP}}'$. $\check{A}$ sends $p\tilde{k}_{\mathcal{CP}}$ to $\mathcal{U}$. $\mathcal{U}$ calculates $pk_{\mathcal{CP}}''' = b^{-1} \cdot p\tilde{k}_{\mathcal{CP}}$ and sends to the $\check{A}$ checks. $\check{A}$ checks if $\Psi = pk_{\mathcal{CP}}'''$ then, $pk_{\mathcal{CP}}'$ is correct. Verifies:

$$e((C'P + pk_{\mathcal{CP}}'), \sigma_{\mathcal{CP}}') = e(P', pk_{\mathcal{U}}').$$

Also, $t' \cdot P \stackrel{?}{=} R' + pk_{\mathcal{U}}' \cdot s'$ to proof the correctness of $sk_{\mathcal{U}}'$. $\check{A}$ can verify the verification equation. However, $\check{A}$ can not get any secret information about $\mathcal{U}$. Thus, we achieve user anonymity at $\mathcal{SP}$ end. □

Theorem 4. *Assume that the ZSS short signature and self credential Verheul's schemes provide user anonymity, the NI-Schnorr PoK of data satisfies 'zero-knowledge'. Then our protocol satisfies the unlinkability and untraceability from $\mathcal{SP}$.*

Proof. As defined in Section 5, given two distinct transactions are managed by the same user $\mathcal{U}$. An adversary $\check{A}$, who may be a $\mathcal{SP}$ or a *CP*, has ways to link or trace the transactions by the same $\mathcal{U}$.

1. If $\mathcal{CP}$ is an adversary $\check{A}$: links the two transactions with the same $\mathcal{U}$ according to their attribute values v_1, v_2, ..., v_m and $\sigma_{\mathcal{IV}}$.

In our protocol, the attribute values v_1, v_2, ..., v_m are never used in a plain text form with both communicating parties $\mathcal{IV}$ and $\mathcal{CP}$. $\mathcal{CP}$ receives $\sigma_{\mathcal{IV}}$ and NI-Schnorr PoK of the $C = r \cdot pk_{\mathcal{U}} + (v_1P_1 + v_2P_2 + ... + v_nP_n)$, which is $(A,\ t,\ r \cdot pk_u)$, $(A_i,\ t_i,\ v_i \cdot P_i)$ (see the Fig. 3). $\mathcal{CP}$ verifies $ECDSA - Verify(C, \sigma_{\mathcal{IV}}, pk_{\mathcal{IV}})$, and $t \cdot pk_{\mathcal{U}} \stackrel{?}{=} A + r \cdot pk_{\mathcal{U}} \cdot s,\ t_i \cdot P_i \stackrel{?}{=} A_i + v_i \cdot P_i \cdot s_i$, $C \stackrel{?}{=} r \cdot pk_{\mathcal{U}} + v_i \cdot P_i$ to check that $\mathcal{U}$ is an authorised user and the NI-Schnorr PoK of *C*. In the first verification step, $ECDSA - Verify(C, \sigma_{\mathcal{IV}}, pk_{\mathcal{IV}})$, $\mathcal{CP}$ needs output of $\sigma_{\mathcal{IV}}$, *C* and $pk_{\mathcal{IV}}$ to verify that $\mathcal{U}$ is an authorized user or not. *C* is a commitment of attribute values v_1, v_2, ..., v_m, which achieves perfectly hiding property (see Proof of Theorem 3). In our protocol, we assume the attributes and access policies are fixed for the similar type of services (see the detail in Section 4). So, $\mathcal{U}$ requests only one credential, which has similar attributes and policies. For another credential, the access policies and attributes are different. All these attributes achieve perfectly hiding property. Thus, by seeing two transactions from the same $\mathcal{U}$, $\mathcal{CP}$ can not link the signature.

The second verification step is for the NI-Schnorr PoK of *C* (see in the Fig. 3):

$$(t \cdot pk_{\mathcal{U}} \stackrel{?}{=} A + r \cdot pk_{\mathcal{U}} \cdot s), (t_i \cdot P_i \stackrel{?}{=} A_i + v_i \cdot P_i \cdot s_i), and (C \stackrel{?}{=} r \cdot pk_{\mathcal{U}} + v_i \cdot P_i)$$

$\check{A}$ requires secret values to calculate $(A,\ s,\ t)$ and $(A_i,\ s_i,\ t_i)$ (see in the Fig. 3). These secret values are: $A = r \cdot P$, where $r \in_R \mathbb{Z}_q^*$, $s = \mathcal{H}(A)$, $t = s \cdot r + w$, $A_i = w_i \cdot P_i$, $s_i = \mathcal{H}(A_i)$ and $t_i = s_i \cdot v_i + w_i$.

The security is based on the DLP and Inv-CDHP hardness problems, defined in the Definitions 1 and 4, and hash function. So, $\check{A}$ can not guess/calculate the attribute values.

2. If $\mathcal{SP}$ is an adversary $\check{A}$: links the two transactions with the same $\mathcal{U}$ by seeing the credential, describe in Fig. 4.

$\mathcal{SP}$ receives $\sigma'_{\mathcal{CP}}$, $pk'_{\mathcal{U}}$, $pk'_{\mathcal{CP}}$, P', C', R', t' to verify the credential of $\mathcal{U}$. All these values are blinded by $\mathcal{U}$, the blind factor *b* is randomly chosen. The security is based on the hardness solution of DLP, CDHP and Inv-CDHP defined in Section 3.1. If $\mathcal{SP}$ receives another credential from the same $\mathcal{U}$. $\mathcal{SP}$ is not able to calculate blind factor due to the security hardness. So, $\check{A}$ can not link and trace two credential is from the same $\mathcal{U}$. □

Theorem 5. *Assume that the ECDSA and ZSS short signature schemes have the unforgeability feature; then our protocol has unforgeability.*

Proof. Here, we consider that $\mathcal{U}$ is unforgeable in both protocols; credential issuance and credential presentation.

Credential Issuance: In Fig. 3 of the Section 5.2, $\mathcal{U}$ sends $\sigma_{\mathcal{IV}}$ on *C* and NI-Schnorr PoK of *C* to $\mathcal{CP}$. As assumed in Lemma 4, $\mathcal{U}$ is a PPT adversary $\check{A}$. Therefore, $\check{A}$ constructs any signature on *C* and generates a NI-Schnorr PoK of *C*. If $\check{A}$ can produce signature similar to $\mathcal{IV}$ using same secret key for the same *C*, he is able to forge a signature of $\mathcal{U}$ and committed data values. Due to hardness of discrete logarithm problem 1, $\check{A}$ can not calculates the signature on *C*. Also, due to unforgeability property of ECDSA, $\check{A}$ can not produce the same signature. The unforgeability of ECDSA is illustrated in the paper [24].

Credential presentation: In Fig. 4, Section 5.3, $\mathcal{U}$ sends blind version of credential and NI-schnorr PoK of blinded secret key. $\mathcal{U}$ chooses $b \in_R \mathbb{Z}_q^*$ as blinding factor, calculates blinded secret and public keys: $sk'_{\mathcal{U}} = b \cdot sk_{\mathcal{U}}$, $pk'_{\mathcal{U}} = b \cdot sk_{\mathcal{U}} \cdot P$, blinds credential and associated parameters $\sigma'_{\mathcal{CP}} = b \cdot \sigma_{\mathcal{CP}}$, $P' = b \cdot P$, $pk'_{\mathcal{CP}} = b \cdot pk_{\mathcal{CP}}$, $C' = b \cdot H(C)$. In addition to this, $\mathcal{U}$ calculates R', s', t' values (see in the Section 5.3) for NI PoK of $sk'_{\mathcal{U}}$. Then, sends these values to $\mathcal{SP}$, $\sigma'_{\mathcal{CP}}$, $pk'_{\mathcal{U}}$, $pk'_{\mathcal{CP}}$, P', C', (R', t') for blinded credential verification. If $\mathcal{U}$ is an adversary. Then, the $\check{A}$ blinds all he parameters as above steps. However, $pk_{\mathcal{CP}}$ is the important factor for blinded credential verification. $\check{A}$ can not cheat with the value of $pk_{\mathcal{CP}}$, as this value is calculated at $\mathcal{SP}$ end (see in the Fig. 4). If, $\check{A}$ modifies the $pk_{\mathcal{CP}}$ then credential verification would not work and $\mathcal{SP}$ aborts $\mathcal{U}$'s request. The verification equation is:

$$e((C'P + pk'_{\mathcal{CP}}), \sigma'_{\mathcal{CP}}) = e(P', pk'_{\mathcal{U}}).$$

Therefore, Theorem 5 holds. □

6.2. Comparison of privacy-Preserving credential schemes

Table 1 compares our scheme with those of several authors on the basis of privacy and security features, as well as blockchain based implementation and cost efficient cryptographic algorithms. The privacy feature includes anonymity, unlinkability and untraceability. The security builds on confidentiality and unforgeability features. Idemix [25] is based on the strong RSA problem, one would want the key size to be at least 2048 bits and preferably even 4096 bits; On the other hand, U-Prove [9] is more efficient but does not provide unlinkability; in addition, its security is not fully proven. This paper [14] has achieved a good level of privacy and provide security proofs. However, confidentiality is compromised in communication with issuer and verifier. This is a serious concern if we implement this scheme over blockchain. Additionally, this scheme has more than one round of communication with an issuer, this is an addition cost and overhead on the blockchain. The paper [19] has proposed a selective disclosure credential scheme and integrated with blockchains. This paper has been

Table 2
Experimental results.

Our protocols	Action	~	>	<
Section 5.1, Fig. 2	Generate Keys	10 ms	9 ms	11 ms
$\mathcal{U} \leftrightarrow \mathcal{IV}$	Generate commitment	975 µs	960 µs	980 µs
	SignCommitment	334 µs	328 µs	340 µs
	Verify Signature	543 µs	537 µs	550 µs
Section 5.2, Fig. 3	Generate ZKP Age	746 µs	732 µs	760 µs
$\mathcal{U} \leftrightarrow \mathcal{CP}$	Verify Proof Age	785 µs	772 µs	792 µs
	Generate Pairing Key	23 ms	21 ms	23 ms
	Generate Credential	4 ms	3 ms	5 ms
	Verify Credential	40 ms	37 ms	47 ms
Section 5.3, Fig. 4	Blind Certificate	19 ms	16 ms	25 ms
$\mathcal{U} \leftrightarrow \mathcal{SP}$	Verify Blind Credential	27 ms	23 ms	31 ms

~ : average computational time | > : minimum computational time | < : maximum computational time | ms: milliseconds | µs: microseconds.

Table 3
Comparison of privacy-preserving credential schemes.

Parameters → Schemes ↓	Computation cost for Credential presentation step in milliseconds (ms)
[19]	120.10 ms
Idemix[25]	71.72 ms
Our scheme	27 ms

we use the results of Idemix [12] and paper [19]. The computation cost for one attribute over Ethereum blockchain for Paper [19] is 120 ms. The computation cost of Idemix is mentioned for two attributes. For our protocol, the number of attributes does not impact the computation cost.

implemented on Chainspace and Ethereum blockchains. This paper does not provide confidentiality, anonymity and untraceability security proofs. However, unforgeability, blindness and unlinkability security proofs are not in detail.

Our scheme is implemented on the blockchain with achieving strong privacy features and providing security proofs. We attain an acceptable level of efficiency (see in Table 2). The efficiency is based on the usage of elliptic curves and ZSS-pairing based short signature, which is more efficient than [14] scheme, as calculated in report [21]. In addition, we use Inv-CDHP based on bilinear pairing (see Definition 4) and does not require any special hash function as the BLS scheme [13], which adds extra layer of efficiency in our protocol. This paper [14] presents a comparative analysis of Idemix computational cost and concludes that [14] is more efficient than Idemix. The part of Idemix implementation over blockchain is available.[2] However, this implementation is not fully working over blockchain. Idemix is a well-known studied scheme in the literature and also has implementation over the blockchain. This paper [19] also has been implemented on blockchains. Thus, we choose Idemix and the paper [19] schemes to compare with our scheme in terms of computational cost. The results are shown in Table 3.

7. Experimental results

This section presents the experimental setup, computational results and comparative analysis of our experimental work. To assess the performance of the proposed protocol, we consider our use case "Online Trading" (see in Section 4.5) for the implementation. More precisely, a user wants to join a blockchain-based trading platform without revealing his real age. Additionally, the user asks for a maximum of privacy and confidentiality while he performs his trading actions.

7.1. Experimental setup

From an implementation point of view, each actor involved in the use case has an interface coded in GO language in order to perform the required operations. For instance, the api "/iv/signCommitment" is one of the various GO services included in the identity validator interface. All cryptographic operations (i.e. keys generation, commitments, verification, pairing handling) are coded in GO Language inside their interface environments. All technical components have been implemented on the same machine having the following properties: Ubuntu 16.04.5 LTS, 16GB of RAM. The GO and Java version are respectively: go1.11 linux/amd64 and java 9.0.4. Go cryptographic packages are imported from the main Go tree except

[2] https://github.com/hyperledger/fabric/commit/238668d393a6186ec56ca6c150a8599dda9d3fa4.

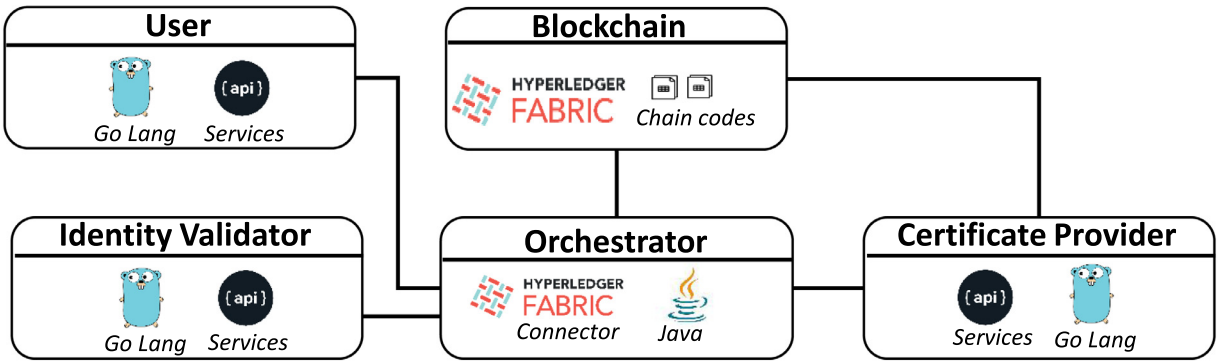


Fig. 5. Experimental setup.

for the (golang.org/x/crypto/bn256) package used for the pairing operation which is imported from the Go sub-repository. The source code of this additional package can be found here: (https://godoc.org/-/subrepo).

In order to simulate the use case, a Java program has been developed. This program can be seen as the orchestrator between all the GO interfaces and their corresponding services. At each step, the program prepares the right input, selects the right interface to be contacted, indicates the right service to be executed, and gets the output. This latter might be then used as an input for the following steps.

For the on-chain part, the consortium blockchain Hyperledger Fabric V1.1 has been used. It is important to emphasize again on the fact that the proposed scheme is a blockchain agnostic solution. Hyperledger Fabric is used only for illustration purposes. In contrast to the use case description, we assume that a user is already a member of the blockchain. This means that a user can send transactions. However, access rights have been added to a special type of transactions. Notably, each transaction of a user is considered valid if it contains a proof that the user is greater than 18; otherwise, the transaction will be rejected.

From a blockchain point of view, the transaction consists of executing an invoke function called “startTrading()” inside a test chaincode. This latter is also coded in GO. The access right verification is implemented inside the “startTrading()” function. This function is mimicking the role of a service provider in the proposed protocol. Thus, a service provider verifies that a user is greater than 18 without knowing the user’s real age value. Using this approach, the verification part is executed on-chain and indicators related to the performance of such cryptographic operations are then computed.

To populate the “startTrading()” function with the right parameters, the Java orchestrator program first calls the different entities involved in the protocol. Since the call of this function also involves a communication part with the blockchain in order to send a transaction, the Hyperledger Fabric Java SDK client has been integrated within the Java orchestrator module.

The experimental setup and sequence diagram are delineated in Figs. 5 and 6 respectively. Our source code is here (https://github.com/odib/privacy-preserving-credential-scheme-over-blockchain). In the first scenario of implementation, a user who is 20 years old is considered. As mentioned above, the Java orchestrator communicates with the right services in order to simulate the protocol. The last step is to send a transaction with the blinded credential protocol of the user. The experimental results showed that the on-chain verification inside the function “startTrading()” was successful. We repeat the same process with an age attribute of value 16 and we get the unsuccessful result because the on-chain verification did not succeed. These two scenarios have validated our protocol as well as, reflects the fact that the proposed protocol is practical and relevant and use cases can be easily dealt with.

7.2. Results

In the following, we provide the computational time related to the various functions used in our protocol. These numbers are the average of 100 of simulations. At each time, the age attribute value is changed and time indicators are computed. The results are shown in Table 2. We compare our scheme with Idemix and the paper [19] for the credential presentation step because this step works on the blockchain in Table 3.

The result in Table 2 demonstrates the time which is required to perform all cryptographic functions related to the off-chain part of the protocol. These steps are performed once at the beginning of the protocol. For the on-chain process that corresponds to the verification of the blind credential of a user, the average time is 27 ms and this can increase the maximum to 31 ms. This step is required to be performed for each transaction to communicate over blockchain.

The results above indicate that the online verification step does not exceed 31 ms in the worst case. This proves the efficiency of our protocol to deal with real-world applications where “real-time” answers are usually required to handle users’ requests.

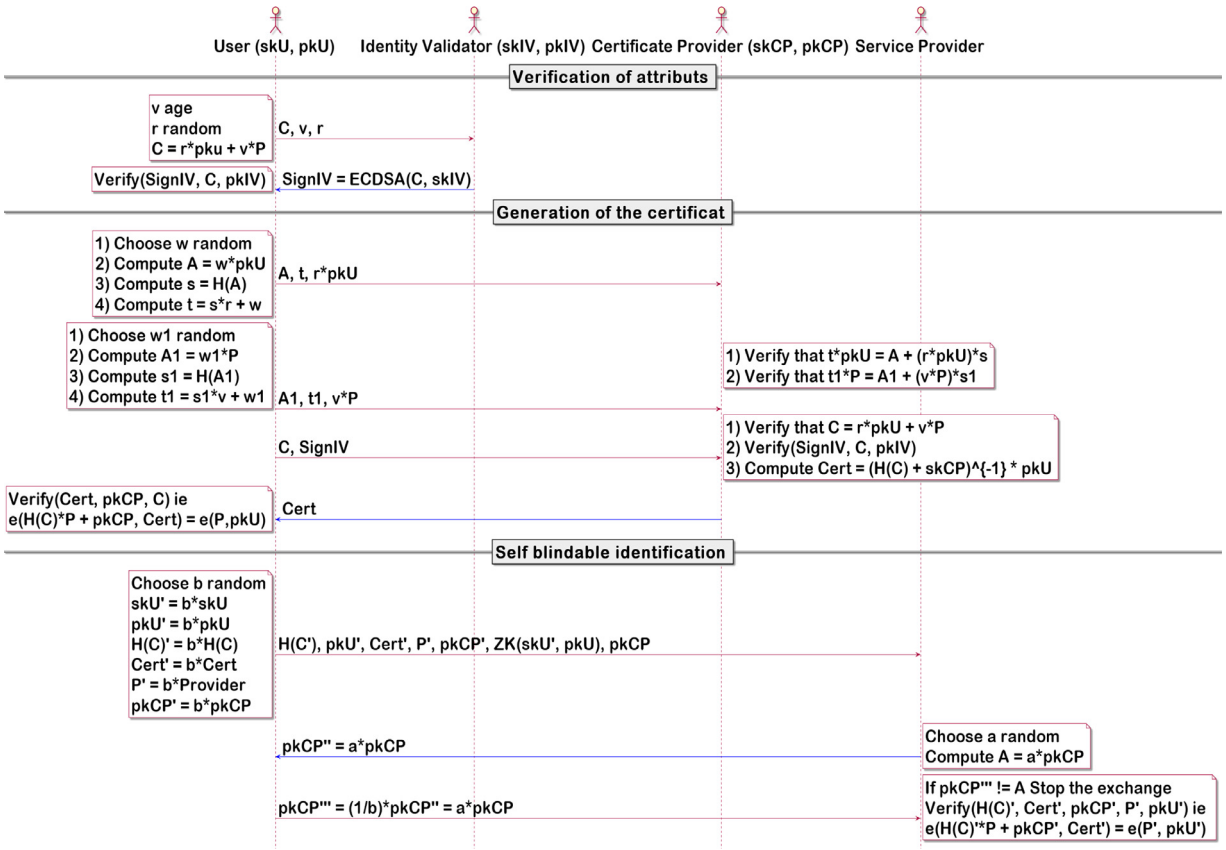


Fig. 6. Sequence Diagram of User and Identity Validator exchange

The comparison with the existing implemented methods of Idemix and Paper [19] shown in Table 3. The solutions of Idemix and [19] are significantly slower than our solution. In our protocol, the cost of verification attributes on-chain remains constant regardless the number of committed attributes.

8. Summary and conclusions

Our paper presents a novel privacy-preserving credential scheme over blockchain, allowing users to anonymize their attribute values and non-interactively communicate with certificate and service providers. We anonymize attribute values in both off-chain and on-chain communications to achieve anonymity. We utilize a modified version of Zhang, Safavi–Naini and Susilo short signature based on bilinear pairing to sign on committed attribute values. In order to access services on the blockchain, a user self-blinds his credential using Verheul approach. This credential is still verifiable by a service provider on the blockchain. We minimize the communication cost by using elliptic curve based non-interactive-Schnorr proof of knowledge for the correctness of commitment and the blind version of users secret key. In addition, we add security proofs to formally validate the advanced features of anonymity, unlinkability and untraceability of a user, as well as confidentiality and unforgeability of data. Furthermore, we detail an online trading use-case where the application of our scheme is relevant. Finally, we implement our solution using Go and Java with Hyperledger Fabric. We exhibit a comparison analysis of our scheme with well-known schemes Idemix, U-Prove, self-blindable and selective disclosure credential schemes on the basis of privacy and security features, along with blockchain implementation and cost efficient cryptographic algorithms. Our experimental analysis delineates that our credential scheme enables more advanced features in comparison with well-known existing schemes. Moreover, the computational time for the complete protocol is low enough to use our scheme in real-world applications.

In future work, we will investigate the possibility to minimize the communication cost between entities and further achieve smaller proof sizes. Moreover, we will focus on studying the behaviour of our protocol with other blockchain technologies such as Ethereum.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Kalpana Singh: Conceptualization, Methodology, Writing - original draft, Supervision, Project administration, Investigation. **Omar Dib:** Software, Validation, Investigation, Writing - review & editing. **Clément Huyart:** Software, Validation, Investigation, Writing - review & editing. **Khalifa Toumi:** Writing - review & editing.

Acknowledgement

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”.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compeleceng.2020.106586.

References

[1] Berkowsky JA, Hayajneh T. Security issues with certificate authorities. In: 8th Annual conference on ubiquitous computing, electronics and mobile communication conference, IEEE; 2017. p. 449–55.
[2] Khan MA, Salah K. Iot security: Review, blockchain solutions, and open challenges. In: Future Generation Computer Systems, Elseiver; 2018. p. 395–411.
[3] Dib OO, Huyart C, Toumi K. A novel data exploitation framework based on blockchain. Pervasive Mobile Comput 2019:101104.
[4] Lee J, Hwang J, Choi J, Oh H, Kim J. Sims: Self sovereign identity management system with preserving privacy in blockchain. 2019. Cryptology ePrint Archive: Report 2019/1241; https://eprint.iacr.org/2019/1241.
[5] Singh K, Heulot N, Hamida EB. Towards anonymous, unlinkable, and confidential transactions in blockchain. IEEE; 2018. [Available] https://ieeexplore.ieee.org/document/8726589.
[6] Sullivan C, Burger E. E-residency and blockchain. In Computer Law & Security, Elseiver 2017;33(4):470–81.
[7] Camenisch J, Lysyanskaya A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: International Conference on the Theory and Applications of Cryptographic Techniques, Springer; 2001. p. 93–118.
[8] Bogatov D, Caro AD, Elkhiyaoui K, Tackmann B. Anonymous transactions with revocation and auditing in hyperledger fabric. 2019. Cryptology ePrint Archive: Report 2019/1097; https://eprint.iacr.org/2019/1097.
[9] Paquin C, Zaverucha G. U-prove cryptographic specification v1.1. Technical Report, Microsoft Corporation. U-prove cryptographic specification v1.1.; 2011. [Available] http://research.microsoft.com/pubs/166969/U-ProveCryptographicSpecificationV1.1Revision2.pdf.
[10] Zhang F, Safavi-Naini R, Susilo W. An efficient signature scheme from bilinear pairings and its applications. In: international workshop on public key cryptography, Springer; 2004. p. 277–90.
[11] Verheul E. Self-blindable credential certificates from the weil pairing. In: International Conference on the Theory and Application of Cryptology and Information Security, Springer; 2001. p. 533–51.
[12] Barki A, Brunet S, Desmoulins N, Traoré J. Improved algebraic macs and practical keyed-verification anonymous credentials. In: International Conference on Selected Areas in Cryptography, Springer; 2016. p. 360–80.
[13] Boneh D, Lynn B, Shacham H. Short signatures from the weil pairing. In: Journal of Cryptology, Springer, 17(4); 2004. p. 514–32.
[14] Ringers S, Verheul E, Hoepman J. An efficient self-blindable attribute-based credential scheme. In: International Conference on Financial Cryptography and Data Security, Springer; 2017. p. 3–20.
[15] Camenisch J, Lysyanskaya A. Signature schemes and anonymous credentials from bilinear maps. In: International Cryptology Conference, Springer; 2004. p. 56–72.
[16] Garman C, Green M, Miers I. Accountable privacy for decentralized anonymous payments. In: International Conference on Financial Cryptography and Data Security, Springer; 2016. p. 81–98.
[17] Lundkvist C, Heck R, Torstensson J, Mitton Z, Sena M. uport: A platform for self-sovereign identity. u-port whitepaper; 2017. [Available] http://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf.
[18] Zyskind G, Nathan O, Pentland A. Decentralizing privacy: Using blockchain to protect personal data. In: Symposium on Security and Privacy Workshops, IEEE; 2015. p. 180–4.
[19] Sonnino A, Al-Bassam M, Bano S, Meiklejohn S, Danezis G. Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. 10.14722/ndss.2018.23xxx; 2019.
[20] Cachin C. Architecture of the hyperledger blockchain fabric. In: Workshop on distributed cryptocurrencies and consensus ledgers; 2016. p. 1–4.
[21] Markel A, Nemirovskiy L, Ziv B. Pairing-based short signatures. 2014 [available] http://markel.co/projects/ecc/2/article.pdf.
[22] Schnorr CP. Efficient signature generation by smart cards. J Cryptogr 1991;4(3):161–74.
[23] Pedersen TP. Non-interactive and information-theoretic secure veriable secret sharing. In: International Cryptology Conference, Springer, Berlin; 1991. p. 129–40.
[24] Fersch M, Kiltz E, Poettering B. On the provable security of (ec) dsa signatures. In: the SIGSAC Conference on Computer and Communications Security, ACM; 2016. p. 1651–62.
[25] IBM Research. Specification of the identity mixer cryptographic library, version 2.3.0. Tech. Rep.; 2010. [available] https://domino.research.ibm.com/library/cyberdig.nsf/papers/EEB54FF3B91C1D648525759B004FBBB1/File/rz3730_revised.pdf.

Kalpana Singh is working at IRT SystemX since 2017. She did her Ph.D at Deakin University, Australia in 2015. Her academic record is laden with First class throughout. She has been teaching successfully at GLA University, India. She has a number of publications to her credit in reputed journals and conferences in the area of Cryptography, Information Security and Blockchain.[1]

Omar Dib received his M.Sc. and Ph.D. in Computer Science from the University of Technology of Belfort-Montbéliard in 2014 and 2017, respectively. He joined the Research Institute of Technology SystemX in 2017, where he works as a research engineer. His current fields of interests are Blockchain, Cryptography, and Combinatorial Optimization.

Clément Huyart works since 2017 at ERCOM, a french company specialized in mobility security solutions. Before that he obtained a master degree at University Paris Diderot in Mathematics and Computer Science applied to Cryptography. He is passionate in cryptography and works also in partnership with IRT SystemX to explore blockchain technologies.

Khalifa Toumi has received his Ph.D. in Computer Science from Telecom Sud Paris (TSP) in 2014. He worked in SAGEMCOM as a software engineer during one year. He was then a technical responsible for different European and French projects in TSP. Actually, he is working as a research engineer for blockchain technology in IRT SystemX.