A2UA: An Auditable Anonymous User Authentication Protocol Based on Blockchain

Qiuyun Lyu[a], Hao Li[a], Zhining Deng[b], Jingyu Wang[a], Yizhi Ren[a,*], Ning Zheng[a,**], Junliang Liu[c] and Huaping Liu[d]

[a]School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[b]Shanghai Shizhuang Information Technology Co., Ltd, Shanghai 200082, China
[c]Hangzhou Meichuang Technology Co., Ltd, Hangzhou 320011, China
[d]School of Electrical Engineering and Computer Science, Oregon State Univ OR 97331, Corvallis

ARTICLE INFO

Keywords:
Authentication
Anonymous
Accountability
Privacy
Credit building

ABSTRACT

Regulating the illegal activities to balance user privacy and cyberspace governance has been a non-trivial challenge for anonymous authentication. Several existing anonymous authentication protocols maintain a good accountability, but they either risk leaking users' private keys or incur too much overhead of accountability for each ongoing authentication. This paper proposes an auditable anonymous user authentication (A2UA) protocol based on blockchain, aiming to address these problems. The A2UA protocol mainly employs bilinear pairing, partial authentication factors, dynamic credits and fake-public keys (FPKs) to achieve anonymous mutual authentication, and applies ring signature and blockchain to accomplish two-level accountability while maintaining user privacy. We present analysis results, which show significant improvements of the A2UA protocol over existing schemes in terms of security, computation and communication costs as well as security and privacy features. Additionally, it has good feasibility in Ethereum Gas cost.

1. Introduction

Authentication protocols provide some guarantees for the communicating parties to know the true identity of the correspondent [1] to ensure Internet services are accessed by the authorized users. Widely used authentication protocols such as Kerberos [2], Open ID [3], OAuth [4], FIDO [5], etc., could provide a uniform and fast authentication and play a vital role in securing social order in cyberspace. However, the contents and user access history are often abused in these protocols, such as being collected, associated and sold for profit, due to a lack of privacy protection of user identities. For example, an adversary launches a tracking attack since a static user identity is used, and could identify a user in the physical world across different systems where OAuth or Open ID is applied. In short, users are 'naked' in cyberspace, in which any entity could obtain others' information he needs, since both the service providers and adversaries could recognize and record users.

Therefore, many anonymous identity authentication schemes are designed to protect the privacy of users' identities. These schemes may be classified into three categories: *static pseudonym based authentication, dynamic pseudonym based authentication*, and *auditable pseudonym based authentication*.

Static pseudonym based authentication schemes allow service providers to confirm users with a fixed pseudo-name to protect the users' real identities from adversaries [6–11]. For example, pseudo identity [6], dummy identity [7], or vehicular username [8] is introduced separately to hide the vehicle's real identity, and two static authentication parameters denote the identity of a sensor node in [9]. However, these schemes leave adversaries the option to trace a fixed target by sniffing the network flow.

Dynamic pseudonym based authentication schemes overcome the above problem by empowering users to transmit a masked pseudonym with random numbers to access a service each time [12–24], or putting a user in a set of valid users where neither the eavesdroppers nor the authenticators can make a distinction [25, 26]. Specifically, Gope *et al.* [12] and Aman *et al.* [13] developed one-time alias identity for each access, and the schemes in [14–17, 19–24] encrypt the pseudonyms with random numbers in the network flow. Although these schemes prevent adversaries from tracing attack, a verifier maps each authenticating messages to a prover, which jeopardizes his identity privacy. Abdallah *et al.* [25] proposed an aggregating based authentication scheme to achieve anonymity of each appliance by taking all the smart household appliances as a whole. Cassola *et al.* [26] introduced private information retrieval (PIR) to make the authenticating exchange between a user and the server unidentifiable by eavesdroppers, the server and access point (AP). However, this leaves the malicious users out of control since the specific requester and the authentication messages are indistinguishable [26].

Auditable pseudonym based authentication schemes [27–32] aim to solve such problems by regulating the users' activities while keeping the users anonymous. In [27, 28], a trusted third party (TTP) and group signature are introduced to build an anonymous authenticating procedure while the malicious behaviors can still be traced. But the TTP stores all the users' private keys, which may result in attacks,

*Corresponding author.
**Corresponding author.
ORCID(s):
E-mail addresses: laqyzj@hdu.edu.cn (Q. Lyu); l_h@hdu.edu.cn (H. Li); ibradypod@gmail.com, dengzhining@theduapp.com (Z. Deng); 1565704822@qq.com (J. Wang); renyz@hdu.edu.cn (Y. Ren); nzheng@hdu.edu.cn (N. Zheng); liujunliang@mchz.com.cn (J. Liu); huaping.liu@oregonstate.edu (H. Liu).

such as user impersonation and key tampering. Zhang *et al.* [29] designed a regulatable digital currency model based on consortium blockchain to achieve the anonymity of payment amount and object. However, this scheme is applicable to digital payment related cases where the currency can be merged or split arbitrarily and not to other scenarios. A conditional privacy protection authentication scheme was proposed by Cui *et al.* [30] for secure vehicular networks where a vehicle is kept anonymous from a cloud service provider. And the trusted authority (TA) is responsible for preventing malicious activities. But it introduces much overhead since it checks a vehicle's real identity for each authentication.

Taking the merits and problems of the above *auditable pseudonym based authentication* schemes into account, we propose an auditable anonymous user authentication protocol based on blockchain. The contributions of the proposed scheme are summarized as follows.

1. We propose a novel auditable anonymous user authentication (A2UA) protocol based on blockchain, which achieves both mutually anonymous authentication and two-level accountability simultaneously.
2. In A2UA, the risk of massive leakage from a third party is relieved because only the user has his private key. The dynamic credit of requester is introduced as additional authentication factor to mitigate the threat from leakage of fixed authentication factor in current schemes.
3. The procedure of rating each other after authenticating is adopted to improve the accountability. The privacy of users' contents are completely protected since the semi-honest third party (TA) has no way to reveal the session key through the user's recovered public key.

The rest of this article is organized as follows. Related works are discussed in Section II, Section III formulates the problem being addressed. Section IV describes the proposed scheme in detail. Security, computation and communication costs as well as security and privacy features comparison are analyzed in Section V and VI, respectively, followed by concluding remarks in Section VII.

2. Related Work

2.1. Static Pseudonym Based Authentication Schemes

Static pseudonyms are first introduced and widely used in authentication of the Internet service for the sake of administration. In order to achieve anonymity and lightweight of authentication in the resource-constrained scenarios, fixed pseudo-names are applied to replace the real identity during authentication schemes [6–9]. A pseudo identity [6] in terms of plain text is used in the whole procedure of authentication, and it is computed based on the real ID for each vehicle. Dummy identities were employed in [7] to achieve the anonymity of authentication between vehicles, while the vehicular username was directly broadcasted to the neighbor roadside unit (RSU) in [8]. Although Li *et al.* [9] concealed a user identity in a temporary identity, two constant authenticating parameters are still sent in each authentication phase. Therefore, adversaries could trace a user with a plain and static pseudonym or parameter through sniffing the network flow in [6–9].

2.2. Dynamic Pseudonym Based Authentication Schemes

Dynamic pseudonyms have been proposed to deal with eavesdroppers' trace attack. Gope *et al.* [12] and Aman *et al.* [13] introduced one-time alias identity where a server and the IoT device negotiate a new pseudonym for the latter one after each authentication. Dawoud *et al.* [14] substituted a static pseudonym with a set of randomly selected sequential combinations of sub-keys when authenticating. Cryptographic algorithms are implemented to randomize the pseudonyms before sending them to the network in [15–17, 19–24]. Specifically, Gope *et al.* [15] utilized hash function and XOR operation combined with a random number and a shared secret key to generate a masked pseudonym. He *et al.* [16], Odelu *et al.* [17], Arfaoui *et al.* [18] and Maria *et al* [24] introduced bilinear pairing algorithm to ameliorate the pseudonym-masking process and elliptic curves cryptography (ECC) algorithm is applied in [19–23].

Although the adversaries cannot identify the client from the network flow in [12–17, 19–24], the server still matches the authenticating messages with a particular client's pseudonym or identity. Once the sever is compromised or semi-trusted, the client faces the risk of being traced or being disclosed of identity in the physical world. Therefore, the algorithms described in [25, 26] target to achieve anonymity to a server. Abdallah *et al.* [25] helped the smart household appliances (APs) to achieve anonymity to the base station (BS) and control center (CC). The main idea is to let one of the APs aggregates their readings and forwards to the BS via the smart meter (SM). The scheme of Cassola *et al.* [26] applies PIR to make a visitor unknown to the access point (AP) and the service provider during the whole authentication. However, dealing with the malicious users remains a significant challenge since requesters are hidden in a set of valid users [26].

2.3. Auditable Pseudonym Based Authentication

To achieve both anonymity and compliance for all users, auditable pseudonym based authentication schemes [27–32] introduce a trusted third party (or parties) to recover the user real identity in need. However, the mutual authentication between a prover and a verifier is left alone. Based on the technology of group signature, Yang *et al.* [27] designed a group signature-based conditional anonymous authentication protocol for mobile users roaming to a foreign network. In this scheme, adversaries, the foreign low earth orbit (FLEO) satellite and foreign gateway station (FGS) cannot identify the mobile user, while the home network control center (HNCC) can reveal the user' real identity. But a HNCC stores all the users' private keys which could be subject to attack (e.g., user impersonation and key tampering). Moreover, the communicating cost for regulating is

substantial since the HNCC needs collect the authenticating messages from FLEO for each user. And if the messages cannot be collected, the HNCC loses the right of accountability. Wang *et al.* [28] combined group signature with non-interactive zero-knowledge proof to improve the scheme described in [33]. Wang's scheme allows a user to select limited attributes needed to generate a valid and pseudonym-based certification when he accesses network service, and provides the certificate authority (CA) to trace the real user of misconduct. However, the CA still maintains the private keys for all users which may suffer from the same attacks as in [27].

In the scheme by Zhang *et al.* [29], the transactions of consortium blockchain (CB) are shuffled and encrypted to achieve anonymity of payment amount and object, and malicious behaviors are monitored by secret sharing based on a threshold. Note that this scheme is limited to the scenarios similar to the digital payment where the operating unit, such as currency, can be merged or split arbitrarily. Cui *et al.* [30] presented a conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment. In this scheme, a vehicle remains anonymous as it accesses the cloud server provider, but the trusted authority (TA) checks its real identity for each authentication. As a result, the problem of non-compliant activity, happens occasionally, is resolved at the expenses of a significant computation and communication overhead.

3. Problem Statement

3.1. System Model

The system proposed in this paper consists of four entities: the semi-trusted authority (TA), a user (U), the service provider (SP), the miners (M) of blockchain (BC), as shown in **Fig**. 1.

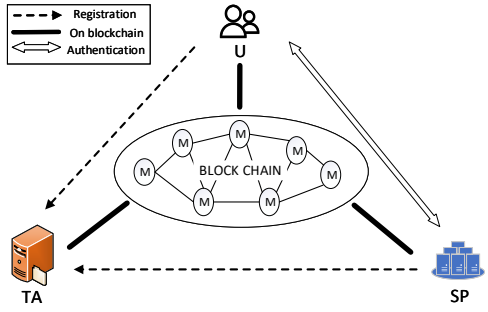


Figure 1: System model.

- **TA** is a semi-honest entity who provides partial authenticating factors for U and SP through registration service, and maintains the authentic scores for them with the support of the blockchain system. Although TA regulates the anonymous U through revealing his real identity, it cannot decrypt the detailed content exchanged between U and the corresponding SP.
- **U** generates a random pseudonym for each access with TA's public key, and issues an authentication request with the partial authenticating factor to SP. Meanwhile, U records the authenticating result and SP's authentic score into the blockchain system.
- **SP** is the entity of providing all kinds of services. SP receives the authentication request from U, performs mutual authenticating process, and similar to U, he records the U's authentic score into the blockchain system.
- **M** is a node who manages the blockchain system. M receives all the transactions from TA, U, and SP, verifies them and puts them on the blockchain system. The **BC** provides a decentralized ledger for A2UA and TA can effectively audit a malicious U or SP in the system through the transactions recorded in BC.

3.2. Security Model

Security assumptions for the proposed scheme are as follows.

- It is built upon the Canetti-Krawczyk (CK) threat model [34], in which any two parties could communicate over an insecure channel. Specifically, a polynomial time adversary tries to reveal, track or even imitate U through sniffing, tampering or launching the authenticating messages between U and SP.
- The TA is a semi-honest entity. It follows the protocol strictly but is still curious about the information, or it could be controlled by adversaries with a high possibility.
- Either U or SP may be malicious where U may behave illegally to gain his own personal interest or SP may gather the detailed access data of U for profit surreptitiously. But they are not malicious simultaneously.
- There is a public blockchain system securely maintained by the Miners. In particular, mainstream public blockchains, such as Ethereum that already have mature consensus algorithms and can resist common blockchain network attacks, could act as the BC. And it is assumed that the standard cryptographic algorithms are secure and unbreakable.
- Similar to the widely used cryptographic algorithms, the public key is assumed to be public to everyone, while the private key is assumed to be exposed only to the owner.

3.3. Security Objectives

To achieve an auditable anonymous authentication, the A2UA protocol aims to achieve the following security objectives.

- ***Mutual authentication***: It provides mutual authentication to ensure the validity of the peer communication parties.
- ***Conditional anonymity***: No one except TA can reveal U's real identity from the blockchain transactions.
- ***Conditional unlinkability***: No one except TA can link the authentication results to a U's real identity.

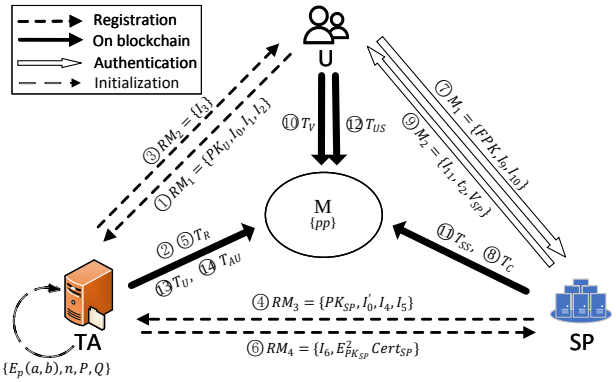


Figure 2: Overview of A2UA

- ***Complete protection of content privacy***: Nobody else, even TA, can decrypt the detailed content exchanged between U and the corresponding SP after each authentication.
- ***Forward/backward confidentiality***: A leakage of the current session key should not affect the security of the protocol's future and previous session keys.

4. The Proposed A2UA Protocol

This section describes the details of the proposed A2UA protocol based on blockchain, as shown in **Fig**. 2. Firstly, both U and SP register with TA to obtain partial authentication factors, and TA stores their identities and default credit, indicated by authentic score (AS), on blockchain, see steps ①-⑥. Secondly, U and SP perform anonymous mutual authentication by evaluating each other's credit and verifying his validity with his own partial authenticating factor, and then, both of them record the authenticated result in blockchain, see steps ⑦-⑩. Thirdly, for the sake of mutual accountability, both U and SP give each other an AS as soon as the session is completed and put it on blockchain, see steps ⑪⑫. At the same time, TA periodically gathers all the ASs from blockchain, computes the average ones for each U and SP, and updates them on blockchain, see step ⑬. Finally, TA regularly audits SP to ensure it is not compromised and reveals U's real identity if U is indeed malicious in step ⑭. To elaborate the A2UA protocol clearly, we first define the notations adopted (see **Table** 1), then several frequently used and important terms are provided. Afterwards, we give a detailed description of the protocol, which consists of five phases: *system initialization, registration, anonymous authentication, credit building, accountability*.

4.1. Definitions

4.1.1. Our Bilinear Pairing Cryptosystem on Elliptic Curves

Inspired by the schemes of Galindo [35] and Boneh and Franklin [36], we develop our bilinear pairing cryptosystem on elliptic curves, as follows:

Let $\mathbb{G}_1, \mathbb{G}_2$ be two cyclic additive groups of prime order p on elliptic curves, and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order p, we construct a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, \psi)$, where ψ is a computable isomorphism which satisfies $\mathbb{G}_2 \to \mathbb{G}_1$, and $\hat{e}$ is the Weil pairing: $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, in other words, $\forall P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$. Choose a generator $P_2 \leftarrow \mathbb{G}_2$, $H_2 : \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \to \{0,1\}^n$, $H_3 : \{0,1\}^* \to \mathbb{Z}_p^*$ and compute $P_1 = \psi(P_2)$, then we have the public parameters: $\{p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, H_2, H_3\}$.

Table 1
Notations and Descriptions.

Notations	Descriptions
$E_p(a,b), n, P, Q$; pp	Public parameters generated by TA; Public parameter generated by BC
x, Q	TA's private and public key
$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$	Cyclic additive groups and cyclic multiplicative group
h	$h : \{0,1\}^* \to \{0,1\}^n$
H_1	$H_1 : \{0,1\}^* \to \mathbb{G}_2^*$
H_2	$H_2 : \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \to \{0,1\}^n$
H_3	$H_3 : \{0,1\}^* \to \mathbb{Z}_p^*$
$PXor(Px, Py)$	Point Px XOR point Py
$U\|\|V$	Concatenate operation
$E_x^0(y), D_x^0(y)$	Encrypt/decrypt y with symmetric key x
$E_x^1(y), Sig_x^1(y)$	Encrypt and sign y with asymmetric key x based on ECC
$E_x^2(y), Sig_x^2(y)$	Encrypt and sign y with asymmetric key x based on bilinear pairings
$Sign_i$	The ring signature of U_i
AF_U, AF_{SP}	Partial authentication factors of U and SP issued by TA
FPK	User's fake-public key
m^U/m^{SP}, m^U_{lst}/m^{SP}_{lst}	U/SP's authentic score and the tolerable lowest score of U/SP

Afterwards, we randomly choose $s \leftarrow \mathbb{Z}_p^*, Q \leftarrow \mathbb{G}_2^*$ and calculate $Q_{pub} = sP_2 \in \mathbb{G}_2^*, P_{pub} = sP_1 \in \mathbb{G}_1^*$, where $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{O_1\}$, $\mathbb{G}_2^* = \mathbb{G}_2 \setminus \{O_2\}$, and O_1, O_2 are the identity elements in $\mathbb{G}_1, \mathbb{G}_2$ respectively. Thus, we have the private key is $sk = d = sQ \in \mathbb{G}_2^*$ and the public key $pk = \langle P_{pub}, Q_{pub}, Q \rangle$.

$\mathbf{E^2_x(y)}$. To encrypt $M \in \{0,1\}^n$, we choose $\mu \leftarrow \{0,1\}^*$, compute $r = H_3(M \parallel \mu) \in \mathbb{Z}_p^*$, and we have $E^2_{pk}(M)$:

$$C = E^2_{pk}(M) = \langle rP_1, (M \parallel \mu) \oplus H_2(g^r) \rangle \tag{1}$$

where $g = \hat{e}(P_{pub}, Q) \in \mathbb{G}_T$.

$\mathbf{D^2_x(y)}$. Let $C = \langle U, V \rangle \in C$ be a ciphertext under the public key pk, we have $D^2_{sk}(C) = V \oplus H_2(\hat{e}(U, d)) = M \parallel \mu$.

$\mathbf{Sig^2_x(y)}$. To sign $M \in \{0,1\}^n$, we choose $r \leftarrow \mathbb{Z}_p^*$, and we have $Sig^2_{sk}(M)$:

$$Sig = Sig^2_{sk}(M) = \langle r, M, H_2(f) \rangle \tag{2}$$

where $f = \hat{e}(rP_1, sk \cdot h(M)) \in \mathbb{G}_T$.

$\mathbf{Ver^2_x(y)}$. To verify Sig, we have $Ver^2_{pk}(M) = H_2(g^r)$, where $g = \hat{e}(P_{pub}, h(M) \cdot Q) \in \mathbb{G}_T$.

Table 2
Transaction Format.

Transaction ID:	Tid
Transaction Type:	$Type$
Transaction data:	$Content$
Input script φ:	ϕ or $Sig_{SK_i}(T_{pre})$
Output script ω $(body, \sigma)$:	$Ver_{PK_j}(body, \sigma)$

4.1.2. PXor

We define two formats of PXor, as follows:

Definition 1: ***PXor(PA, PB)***. $PXor(PA, PB)$ operation is used to perform the XOR operation of two points, as shown in Eq. 3.

$$PXor(PA, PB) = (PA_x \oplus PB_x, PA_y \oplus PB_y) \tag{3}$$

where PA, PB are points.

Definition 2: ***PXor(na, PB)***. $PXor(na, PB)$ operation is used to perform the XOR operation of a number na and a point PB, as shown in Eq. 4.

$$PXor(na, PB) = (na \oplus PB_x, na \oplus PB_y) \tag{4}$$

where na is a number and PB is a point.

4.1.3. Transaction Format

Generally, each transaction consists of three parts: transaction's identity Tid, an input array $Tin[\ ]$ and an output array $Tout[\ ]$. In our scheme, the identities, certified results, and authentic scores of Us and SPs are all put in blockchain as transactions. Therefore, in order to improve the retrieval efficiency of blockchain, transaction type $Type$ (including: $T_R, T_C, T_V, T_{SS}, T_{US}, T_U, T_{AU}$) is added as a new field. Similar to [37], the transaction format is defined in the following equation:

$$Tx = (Tid, Type, Tin[PK_i, T_{pre}, \varphi], Tout[PK_j, Content, \omega]) \tag{5}$$

In the Eq. 5, PK_i and PK_j represent the public key address of the issuer and recipient respectively; T_{pre} denotes the last transaction; φ is an input script for obtaining the previous transaction; ω is an output script which gives the condition for obtaining the $Content$ in the transaction. For a clearer explanation, the format of the transaction is shown in the **Table** 2. In **Table** 2, the transaction body includes T_{pre}, $Type$, *Transaction data* and PK_j. σ is the signature of an issuer on Tx, and if the transaction is the first one, its input is empty and can be set to ϕ.

4.2. System Initialization Phase

In this phase, both TA and the blockchain system perform initialization.

TA generates system parameters as follows. Firstly, it selects a large prime p and an elliptic curve $E_p(a, b)$ with order n under the finite field F_p. Secondly, it randomly chooses a number $x \in Z_q^*$, a base point P which is a generator of $E_p(a, b)$, and calculates the system public key $Q = x \cdot P$. At last, TA stores its private key $SK_{TA} = x$, and publishes system parameters $\{E_p(a, b), n, P, Q\}$.

The blockchain system prepares public parameters for all the system entities. At first, it constructs a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, \psi)$, where $\mathbb{G}_1, \mathbb{G}_2$ are two cyclic additive groups of prime order p on elliptic curves and $\mathbb{G}_T$ is a cyclic multiplicative group of the same order p, $\hat{e}$ is the Weil pairing: $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, in other words, $\forall P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, and ψ is a computable isomorphism which satisfies $\mathbb{G}_2 \to \mathbb{G}_1$.

Then it chooses $H_1 : \{0,1\}^* \to \mathbb{G}_2^*, H_2 : \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \to \{0,1\}^n, H_3 : \{0,1\}^* \to \mathbb{Z}_p^*$ and selects a generator $P_2 \leftarrow \mathbb{G}_2$, then computes $P_1 = \psi(P_2)$, where $\mathbb{G}_2^* = \mathbb{G}_2 \setminus \{O_2\}$, and O_2 is the identity element in $\mathbb{G}_2$. At last, it publishes the public parameter pp:

$$pp = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, H_1, H_2, H_3 \rangle \tag{6}$$

4.3. Registration Phase

For registration, U and SP need to obtain the public parameter pp from the blockchain in advance. There are two separate registration phases: user registration phase and service provider registration phase.

4.3.1. User Registration (UR) Phase

A user must provide his personal attributes (PA): identity (ID), biometrics, such as fingerprints (FP) or face images, etc., to register with TA. For simplicity, personal attributes can be described as $PA = \{ID, FP, ...\}$. And U's PA is supposed to be known by TA before registration.

STEP UR1. U generates his own public and private keys as follows: first, U generates public key based on ID: $PK_U = H_1(ID)$, then, U selects a random number a and generates the private key $SK_U = a \cdot PK_U$.

STEP UR2. U generates parameters for an authenticating factor and sends a request. In detail, U first chooses two random numbers r_{u1}, r_{u2}, calculates $rID = h(ID \parallel r_{u1})$, $mr_{u1} = r_{u1} \oplus H_2(PK_U)$, and $mr_{u2} = r_{u2} \oplus r_{u1}$. Next, U constructs a point $MP_U = (H_2(PK_U) \parallel mr_{u1}, rID \parallel mr_{u2})$ and chooses a random number r_{tmp}. Then, U calculates $I_0 = E^0_{mr_{u2}}(PA)$, $I_1 = PXor(MP_U, r_{tmp} \cdot Q)$ and $I_2 = r_{tmp} \cdot P$. At last, U sends $RM_1 = \{PK_U, I_0, I_1, I_2\}$ to the TA, see step ① in **Fig**. 2.

STEP UR3. After receiving RM_1, TA calculates $MP'_U = PXor(I_1, SK_{TA} \cdot I_2)$ and then recovers $H'_2(PK_U), rID, mr_{u1}$, and mr_{u2} from MP'_U. Next, TA checks if $H_2(PK_U) = H'_2(PK_U)$, if it is, then decrypts I_0 with mr_{u2} to get $PA' = \{ID', FP', ...\}$. At last, TA retrieves the corresponding PA from his own identity database with ID', and checks if $PA' = PA$ and $PK'_U = H_1(ID)$ is true. If it isn't, then the protocol aborts, else goes to next step.

STEP UR4. TA stores PK_U on blockchain, generates the authentication factor AF_U and responds to U. TA issues a *Registration (Reg)* type of transaction to store PK_U. According to Eq. 5, the detailed *Reg* type of transaction is as follows:

$$T_R = (Tid, Reg, Tin[PK_{TA}, \phi, \varphi], Tout[PK_U, PK_U, \omega]) \tag{7}$$

Then, TA computes $AF_U = rID \cdot h(x) \cdot P$, $I_3 = PXor(mr_{u2}, AF_U)$, and responds $RM_2 = \{I_3\}$ back to U, see steps ②③ in **Fig**. 2.

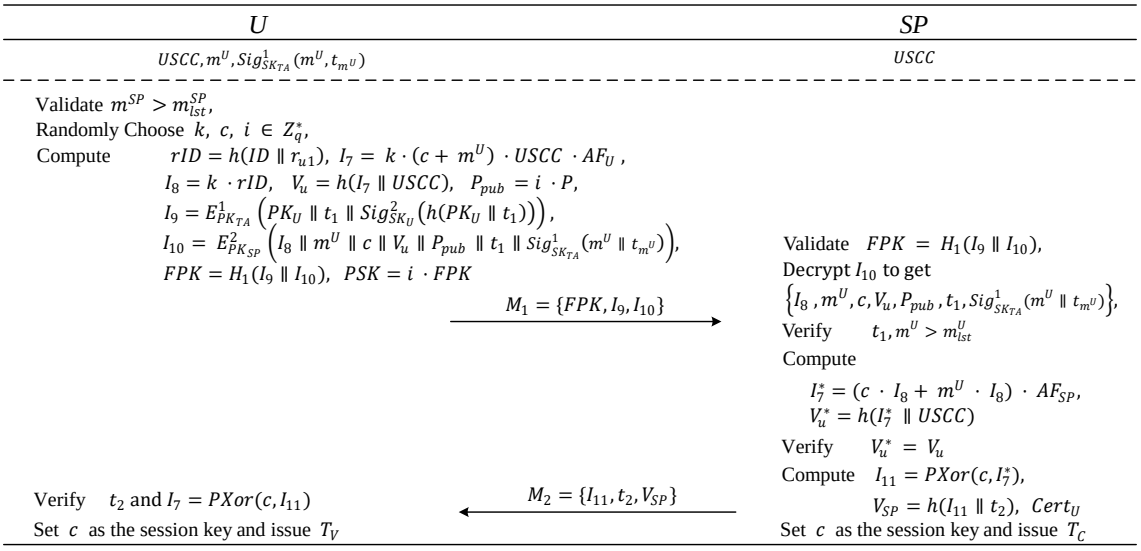


Figure 3: Anonymous authentication phase

STEP UR5. U restores $AF_U = PXor(mr_{u2}, I_3)$ and then stores AF_U and r_{u1} in his device.

4.3.2. *Service Provider Registration (SPR) Phase*

Similar to U, SP must provide his service provider attributes (SPA): Unified Social Credit Code ($USCC$), documents (business scope (BS), representative, address, etc.), to register with TA. For simplicity, service provider attributes can be described as $SPA = \{USCC, BS, ...\}$. And SP's SPA is supposed to be known by TA before registration.

STEP SPR1. SP generates his own public and private keys for registration: SP first generates public key based on $USCC$: $PK_{SP} = H_1(USCC)$, and then, SP selects a random number b and computes the private key $SK_{SP} = b \cdot PK_{SP}$.

STEP SPR2. SP generates parameters for an authenticating factor and sends a request. Specifically, SP first chooses two random numbers r_{s1}, r_{s2}, calculates $mr_{s1} = r_{s1} \oplus H_2(PK_{SP})$ and $mr_{s2} = r_{s2} \oplus r_{s1}$. Secondly, SP constructs a point $MP_{SP} = (H_2(PK_{SP}) \| mr_{s1}, USCC \| mr_{s2})$ and chooses a random number r'_{tmp}. Then, SP calculates $I'_0 = E^0_{mr_{s2}}(SPA)$, $I_4 = PXor(MP_{SP}, r'_{tmp} \cdot Q)$ and $I_5 = r'_{tmp} \cdot P$. At last, SP sends $RM_3 = \{PK_{SP}, I'_0, I_4, I_5\}$ to the TA, see step ④ in **Fig**. 2.

STEP SPR3. After receiving RM_3, TA calculates $MP'_{SP} = PXor(I_4, SK_{TA} \cdot I_5)$ and then restores $H'_2(PK_{SP}), USCC', mr_{s1}$, and mr_{s2} from MP'_{SP}. Further, TA checks whether $H_2(PK_{SP}) = H'_2(PK_{SP})$ holds, if it does, then decrypts I'_0 with mr_{s2} to obtain $SPA' = \{USCC', BS', ...\}$. At last, TA retrieves the corresponding SPA from his own identity database with $USCC'$, and check if $SPA' = SPA$ and $PK'_{SP} = h(USCC)$ is true. If it isn't, then the protocol aborts, else enters next step.

STEP SPR4. TA records PK_{SP} in to blockchain, computes the authentication factor AF_{SP} and sends it to SP. First, TA issues a *Registration (Reg)* type of transaction to store PK_{SP}. Specifically, the detailed transaction is as Eq. 8:

$$T_R = (Tid, Reg, Tin[PK_{TA}, \phi, \varphi], Tout[PK_{SP}, USCC, \omega]) \quad (8)$$

Then, TA computes $AF_{SP} = USCC \cdot h(x) \cdot P$, $I_6 = PXor(mr_{s2}, AF_{SP})$, and generates a certificate ($Cert_{SP} = (mr_{s1}, mr_{s2}, t, Sig^1_{SK_{TA}}(mr_{s1} \| mr_{s2} \| t))$) for SP, and t is the timestamp. At last, TA stores $PK_{SP}, Cert_{SP}$ and responds $RM_4 = \{I_6, E^2_{PK_{SP}}(Cert_{SP})\}$ back to SP, see steps ⑤⑥ in **Fig**. 2.

STEP SPR5. SP computes $PXor(mr_{s2}, I_6)$ to get AF_{SP}, and then stores AF_{SP} and $Cert_{SP}$.

4.4. Anonymous Authentication (AA) Phase

This phase achieves mutual anonymous authenticating, as showing in **Fig**. 3.

STEP AA1. At first, U searches blockchain system to fetch the $USCC$ of SP with expected credit, where SP's authentic score (m^{SP}) is higher than U's expectational m^{SP}_{lst}. Then, U prepares the materials for mutual authentication as follows: U obtains his $m^U, Sig^1_{SK_{TA}}(m^U \| t_{m^U})$, and selects two random values k and c, further calculates $rID = h(ID \| r_{u1}), I_7 = k \cdot (c + m^U) \cdot USCC \cdot AF_U, I_8 = k \cdot rID$ and $V_u = h(I_7 \| USCC)$.

STEP AA2. U generates a fake-public key (FPK) and the corresponding fake-private key (FSK) for the sake of anonymity and sends an authenticating request. Firstly, U selects a random number i and calculates $P_{pub} = i \cdot P$, $I_9 = E^1_{PK_{TA}}(PK_U \| t_1 \| Sig^2_{SK_U}(h(PK_U \| t_1)))$, $I_{10} = E^2_{PK_{SP}}(I_8 \| m^U \| c \| V_u \| P_{pub} \| t_1 \| Sig^1_{SK_{TA}}(m^U \| t_{m^U}))$, where t_1 is the timestamp. Then, U generates the FPK: $FPK = H_1(I_9 \| I_{10})$. Further, U generates a private key

$FSK = i \cdot FPK$. At last, U sends $M_1 = \{FPK, I_9, I_{10}\}$ to SP, see step ⑦ in **Fig**. 2.

STEP AA3. After receiving M_1, SP checks if $FPK = H_1(I_9 \parallel I_{10})$ holds, if it holds, SP decrypts I_{10} with SK_{SP} and get $\{I_8, m^U, c, V_u, P_{pub}, t_1, Sig^1_{SK_{TA}}(m^U \parallel t_{m^U})\}$. Then, it checks whether $t_{now} - t_1 \leq \triangle t_t$ holds, where t_{now} is the time of M_1 received, $\triangle t_t$ is the max tolerable transmission delay. And if not, the authentication is aborted, else SP retrieves m^{U*} and $t^*_{m^U}$ from $Sig^1_{SK_{TA}}(m^U \parallel t_{m^U})$ with PK_{TA}. Further, SP verifies if $m^{U*} = m^U$, $m^{U*} > m^U_{lst}$ and $t_{now} - t^*_{m^U} \leq \triangle t_h$ hold, where m^U_{lst} is the tolerable lowest score of U which is expected by SP, and $\triangle t_h$ is the longest updating intervals of score. If all fulfill, SP calculates $I^*_7 = (c \cdot I_8 + m^U \cdot I_8) \cdot AF_{SP}$ and $V^*_u = h(I^*_7 \parallel USCC)$. Afterwards, SP verifies whether $V^*_u = V_u$ is true. If it is true, FPK is from one of the valid Us. Next, SP computes $I_{11} = PXor(c, I^*_7), V_{SP} = h(I_{11} \parallel t_2), Cert_U = (x_U, y_U, I_9, t, Sig^2_{SK_{SP}}(FPK \parallel x_U \parallel y_U \parallel I_9 \parallel t))$, where $x_U = FPK \cdot mr_{s1}, y_U = FPK \cdot mr_{s2}$, and t is the timestamp of the certificate. At last, SP returns $M_2 = \{I_{11}, t_2, V_{SP}\}$ to the U. At the same time, SP takes c as the session key, and SP issues a *Cer* type of transaction to U to store $Cert_U$, see step ⑧⑨ in **Fig**. 2:

$$T_C = (Tid, Cer, Tin[PK_{SP}, \phi, \varphi], Tout[FPK, Cert_U, \omega]) \quad (9)$$

To prevent transaction-replay attack, miners first check if $t_{T_c} - t < \triangle t$, where t_{T_c} is the timestamp of transaction, and $\triangle t$ is the maximum tolerable delay. If it is, the T_C is verified and is put on blockchain.

STEP AA4. After receiving M_2, U checks whether $t_{now} - t_2 \leq \triangle t$ holds, where t_{now} is the time of M_1 received, $\triangle t$ is the max tolerable transmission delay. And if not, the authentication is aborted, else calculates whether $I_7 = PXor(c, I_{11})$ is true. If it is, SP is verified, U takes c as the session key and issues a *Verify (Ver)* type of transaction to record the successful result of authentication (denoted as *result*), see step ⑩ in **Fig**. 2. Specifically, the detailed transaction is as Eq. 10:

$$T_V = (Tid, Ver, Tin[FPK, T_C, \varphi], Tout[PK_{SP}, result, \omega]) \quad (10)$$

4.5. Credit Building Phase

There are two credit building phases: U's credit building phase and SP's credit building phase. In our scheme, the authentic score (AS) is used for indicating credit. After the registration, the initial ASs of U and SP are set to 60.

4.5.1. U's Credit (UC) Building Phase

STEP UC1: SP gives U's FPK an AS before the session is closed. In detail, the i-th SP issues the *Score (Sco)* type of transaction to store m^U_i, see step ⑪ in **Fig**. 2 as Eq. 11:

$$\begin{aligned} T_{SS} =& (Tid, Sco, Tin[PK_{SP}, T_V, \varphi], \\ & Tout[PK_{TA}, \{m^U_i, FPK, Cert_U\}, \omega]) \end{aligned} \quad (11)$$

STEP UC2: TA periodically computes the average AS for U through collecting the *Sco* type of transaction (T_{SS}). Specifically, TA extracts all the latest *Sco* transactions of SP on the blockchain, and then parses out the real public key of each U. After that, the scores are aggregated according to the same PK_U and the new AS of PK_U is calculated in the following Eq. 12:

$$m^U_{new} = \frac{k \cdot m^U_{old} + \sum_{i=1}^{n} m^U_i}{k + n} \quad (12)$$

where k is the number of historical *Sco* type of transactions for U, n is the number of latest U's *Sco* type of transactions.

Finally, TA issues an *Update (Upd)* type of transaction to store AS of U. The detailed *Upd* transaction is as Eq. 13:

$$\begin{aligned} T_U =& (Tid, Upd, Tin[PK_{TA}, T_{pre}, \varphi], Tout[PK_{TA}, PK_U, \\ & \{m^U_{new}, E^1_{PK_U}(Sig^1_{SK_{TA}}(m^U_{new} \parallel t_{m^U_{new}})), k\}, \omega]) \end{aligned} \quad (13)$$

where T_{pre} is the last Upd type of transaction of the PK_U, see step ⑬ in **Fig**. 2.

4.5.2. SP's Credit (SPC) Building Phase

STEP SPC1: U gives SP an AS before the session is closed. In detail, the j-th U issues the *Score (Sco)* type of transaction to store m^U_j, see step ⑫ in **Fig**. 2 as Eq. 14.

$$\begin{aligned} T_{US} =& (Tid, Sco, Tin[FPKs, \phi, \varphi], \\ & Tout[PK_{TA}, \{m^{SP}_j, PK_{SP}, Sign_j, t_r\}, \omega]) \end{aligned} \quad (14)$$

To hide the real FPK of himself, firstly, U sets the last transaction as empty (ϕ) and puts a set of FPKs who communicated with the SP into $Tin[]$, as Eq. 14 shows. In fact, each FPK owns one certificate ($Cert_U$) published by the SP in ***4.4. STEP AA3***, and all the certificates belong to the same equivalence class according to [38]. Secondly, U employs the ring signature to sign the authentic score (m^{SP}_j). In detail, the ring signature $Sign_j$ is calculated as algorithm 1, where t_r is timestamp.

Algorithm 1 Generate $Sign_j$

Input: m^{SP}_j, PK_{SP},
$FPK_1, FPK_2, FPK_j, FPK_4, FPK_5, ..., FPK_n$;

Output: $Sign_j, t_r$;

1: Set $v = h(t_r)$;
2: Select random numbers $x_1, x_2, x_4, x_5, ..., x_n$;
3: Set symmetric key k: $k = h(m^{SP}_j + PK_{SP})$;
4: Set $y_1 = E^2_{FPK_1}(x_1), y_2 = E^2_{FPK_2}(x_2), y_4 = E^2_{FPK_4}(x_4), y_5 = E^2_{FPK_5}(x_5), ..., y_n = E^2_{FPK_n}(x_n)$;
5: Computer y_j according to the equation: $E^0_k(y_n \oplus ...E^0_k(y_5 \oplus E^0_k(y_4 \oplus E^0_k(y_j \oplus E^0_k(y_2 \oplus E^0_k(y_1 + v)))))...) = v$;
6: Set $x_j = D^2_{FSK_j}(y_j)$;
7: Set $Sign_j = (v, (FPK_1 : x_1), (FPK_2 : x_2), (FPK_j : x_j), (FPK_4 : x_4), (FPK_5 : x_5), ..., (FPK_n : x_n))$;
8: **Return** $(Sign_j, t_r)$;

STEP SPC2: After receiving the Sco type of transaction (T_{US}), the miners check if t_r is earlier than current time, if it is, then search the previous T_{US}s which were generated after t_r. If it exists, they discard this T_{US}, else the miners examine whether the FPKs belong to the SP's equivalence class. Specifically, for each FPK, the miners search its T_C

type of transaction and retrieve its certificate. Then, for any two FPKs from T_{US}, denoted as FPK_{i1}, FPK_{i2}, the miners check whether $x_{U_{i1}} \cdot y_{U_{i2}} = x_{U_{i2}} \cdot y_{U_{i1}}$, if it is, they are in the same equivalence class.

Secondly, the miners check whether $Sign_j$ is correct as algorithm 2. If it is true, the miners record T_{US} into blockchain.

Algorithm 2 Verify $Sign_j$

Input: $(m_j^{SP}, PK_{SP}, Sign_j, t_r)$;
Output: **T** or **F**;
1: **if** $v = h(t_r)$ **then**
2: Set $k = h(m_j^{SP} + PK_{SP})$;
3: Set $y_1 = E^2_{FPK_1}(x_1), y_2 = E^2_{FPK_2}(x_2), y_4 = E^2_{FPK_4}(x_4), y_5 = E^2_{FPK_5}(x_5), ..., y_n = E^2_{FPK_n}(x_n)$;
4: Set $v' = E^0_k(y_n \oplus ...(E^0_k(y_5 \oplus E^0_k(y_4 \oplus E^0_k(y_i \oplus E^0_k(y_2 \oplus E^0_k(y_1 + v))))))$;
5: **if** $v' = v$ **then Return T**;
6: **end if**
7: **end if**
8: **Return F**;

STEP SPC3: TA periodically computes the average AS for SP through collecting the *Sco* type of transaction (T_{US}). TA computes the AS of PK_{SP} according to Eq. 15.

$$m_{new}^{SP} = \frac{k \cdot m_{old}^{SP} + \sum_{j=1}^{n} m_j^{SP}}{k+n} \tag{15}$$

where k is the number of historical *Sco* type of transactions for SP, n is the number of latest SP's *Sco* type of transactions.

At last, TA issues a *Update (Upd)* type of transaction to store SP's AS. Specifically, the transaction is as Eq. 16:

$$\begin{aligned} T_U =& (Tid, Upd, Tin[PK_{TA}, T_{pre}, \varphi], Tout[PK_{TA}, PK_{SP}, \\ & \{m_{new}^{SP}, Sig^1_{SK_{TA}}(m_{new}^{SP} \parallel t_{m_{new}^{SP}}), k\}, \omega]) \end{aligned} \tag{16}$$

where T_{pre} is the last Upd type of transaction of the PK_{SP}, see step ⑬ in **Fig**. 2.

4.6. Accountability Phase

In the accountability phase, each U is audited not only by TA but also by the SPs. Similarly, each SP is audited not only by TA but also by the Us.

4.6.1. TA audits U

Firstly, SP files a lawsuit including the U's FPK since he thinks his interest is damaged by U. Then, TA searches the *Ver* type transactions of the blockchain according to the FPK, and confirms that SP did authenticate the U. Further, TA extracts I_9 from $Cert_U$ of FPK in Eq. 9, decrypts it with SK_{TA} to reveal the PK_U. Then, TA handles this lawsuit offline. If U is confirmed to be malicious, TA issues an *Audit (AU)* type of transaction to subtract a fixed score of U's (denoted as *AUU_result*) , see step ⑭ in **Fig**. 2 as Eq. 17.

$$\begin{aligned} T_{AU} =& (Tid, AU, Tin[PK_{TA}, T_{AU}, \varphi], \\ & Tout[PK_{TA}, PK_U, AUU\_result, \omega]) \end{aligned} \tag{17}$$

4.6.2. TA audits SP

Before TA updates SP's AS, TA needs to check the SP is not be counterfeited. TA first extracts FPK, FPK's $Cert_U$ and PK_{SP} from T_C (in Eq. 9) and fetches mr_{s1}, mr_{s2} from its database according to PK_{SP}. Then, TA calculates $x'_U = FPK \cdot mr_{s1}$ and $y'_U = FPK \cdot mr_{s2}$. Further, TA checks if $x'_U, y'_U \in Cert_U$. If it holds, the SP is authentic, else, the $Cert_U$ comes from an illicit SP and he will be punished. In detail, TA revokes the SP through issuing an *AU* type of transaction to set his AS to zero (denoted as *AUSP_result*), see step ⑭ in **Fig**. 2 as Eq. 18:

$$\begin{aligned} T_{AU} =& (Tid, AU, Tin[PK_{TA}, T_{AU}, \varphi], \\ & Tout[PK_{TA}, PK_{SP}, AUSP\_result, \omega]) \end{aligned} \tag{18}$$

And TA initiates a batch of transactions to return SP's improper profit to the affected FPKs.

4.6.3. U and SP audit Mutually

U audits SP through issuing an authentic score (AS) transaction T_{US}, and SP audits U through similar type of transaction T_{SS}. The difference is, U employs ring signature to hide his FPK to relieve the pressure of scoring, while SP rates a FPK objectively since the FPK cannot be linked to the same person in the history or future.

5. Security Analysis

In this section, we first prove that the proposed A2UA scheme achieves all the security objectives presented in Section *3.3*. Secondly, we demonstrate that the A2UA scheme is provably secure.

5.1. Analysis of Security Objectives

5.1.1. Mutual Authentication

SP authenticates U by verifying whether U's authentic score $m^U > m^U_{lst}$ and $V_u = h(I_7^* \parallel USCC)$ where I_7^* is computed from I_8 with his authenticated factor AF_{SP}. Similar to SP, U authenticates SP by checking $m^{SP} > m^{SP}_{lst}$ and whether $I_7 = PXor(c, I_{11})$ is true, where c is random number sent to SP by U and only the one who owned a valid AF_{SP} can construct a valid I_{11}. Therefore, mutual authentication is achieved between legitimate Us and SPs in our scheme.

5.1.2. Conditional Anonymity

In our protocol, the U's identity (PK_U) is hidden in I_9 which is used to generate a disposable FPK and $Cert_U$. SP, miners or adversaries cannot recover (PK_U) from FPK, $Cert_U$ or I_9 since I_9 is revealed by using TA's private key. At the same time, by introducing timestamp, random numbers in each generating process of I_8, I_9, and combining $H_1(I_9 \parallel I_{10})$ to get FPK, our scheme allows FPK and I_9 not to be paraphrased by SP, miners or adversaries for each session. For the accountability sake, TA is allowed to recover the U's identity (PK_U) by decrypting I_9 from $Cert_U$ which stored on blockchain. Therefore, our scheme realizes conditional anonymity where no one except TA can reveal U's identity.

5.1.3. Conditional Unlinkability

For SP, he can get $FPK, I_9, I_{10}, m^U, I_8, c, P_{pub}$ from U for each authentication, however, he cannot link these information from history authentications to the same U since m^U is not unique for a U, c is a random chosen number, FPK is a hashed value and I_8, P_{pub} are random number embedded. Although miner has a lot of FPKs, I_9s, $Cert_U$s, he cannot link them to the same U since he cannot decode them. For an adversary, he cannot trace U even he has gathered all the information from both communication channel and blockchain. From his point of view, FPKs, $Cert_U$s, I_{10}s, I_{11}s, etc. are only random numbers since he cannot interpret without SP's and TA's private key. However, to audit Us, TA reveals each U's PK_U from blockchain transactions T_C, T_V and links multiple FPKs, I_9s, $Cert_U$s to the specific U. Therefore, the scheme achieves conditional unlinkability.

5.1.4. Complete Protection of Content Privacy

Although TA stores SP's private information (mr_{s1}, mr_{s2}) and also can recover U's identity through $FPK, Cert_U$, all the messages between U and SP are encrypted by their shared session key c after authentication. And c is hidden in I_{10} which is encrypted by a SP's public key. As a result, an adversary cannot decrypt it without the SP's private key. Further, c is not recorded in blockchain, therefore, both TA and miners cannot access the communication content between U and SP, not to mention adversaries. Accordingly, complete protection of content privacy is fulfilled in our scheme.

5.1.5. Forward/Backward Confidentiality

In our scheme, each FPK, I_9 or $Cert_U$ is generated with randomly chosen numbers, thus, they cannot be inferred by an adversary even if he has cracked some of them. In other words, with cracked FPKs, I_9s or $Cert_U$s, an adversary cannot crack the others in the history or future. Secondly, the session after each authentication is also encrypted by a random number c. As a result, an adversary cannot infer the other cs with the cracked ones. Consequently, our scheme possesses forward/backward confidentiality.

5.1.6. Resistance to Replay Attacks

In the *Anonymous Authentication Phase* (**4.4**), the timestamps t_1, t_2 are included in the sent authenticating messages M_1, M_2, and the verifier first checks them with the receiving time t_{now}. If $t_{now} - t_1 < \triangle t$ and $t_{now} - t_2 < \triangle t$ hold, then replay attack is a small probability event in the process of authenticating. Further, miners check if $t_{T_c} - t < \triangle t$ to prevent transaction-replay attack because of $T_{pre} = \phi$. Similarly, in the *Credit building Phase* (**4.5**), t_r is also introduced to make sure T_{US} is fresh. Therefore, our scheme is resistant to replay attacks.

5.2. Provable Security

The proposed scheme is based on bilinear pairing cryptosystem on elliptic curves (denoted as BPCEC), ring signature and ECC. According to the security characteristics of each module, we show that our protocol achieves conditional anonymity, conditional traceability and complete protection of content privacy.

Theorem 1. If the BPCEC satisfies the basic security properties, the ring signature algorithm meets complete anonymity and FPK is anonymous except TA, then, the scheme in this paper meets conditional anonymity.

Proof 1. Define A_{anony} as an adversary (except TA) attacking the anonymous simulation game of our scheme, A_{rs} as an opponent attacking the anonymity of our ring signature algorithm, A_{BPCEC} as an adversary who attacks the BPCEC, and A_{FPK} as an adversary (except TA) attacking the FPK generation algorithm. Assuming A_{anony} successfully attacks the anonymity of the scheme, a polynomial time algorithm $A_\theta \in (A_{rs}, A_{FPK})$ is defined, which has the ability to attack the algorithms of our ring signature and FPK generation. Through the query of A_{anony} and the A_θ's interaction in the anonymous simulated attack game, A_θ is optimized repeatedly to successfully attack the ring signature and FPK generation algorithms. That is, if the adversary A_{anony} successfully attacks the anonymity of the scheme, it means A_θ successfully attacks the anonymity of algorithms of our ring signature and FPK generation with a certain probability.

According to the steps defined by anonymity, the interaction between algorithm A_θ and adversary A_{anony} is as follows:

STEP 1: Generation phase: Through the public parameters generated by A_{BPCEC} in the system initialization phase, algorithm A_θ generates U's public key PK_U, which is a public parameter of the anonymous attack simulation game. At last, A_θ sends PK_U to A_{annoy};

STEP 2: Inquiry phase: The adversary A_{anony} can query the algorithm A_θ for polynomial time:

Registration query: The adversary A_{anony} requests the private key SK_U corresponding to a user U, and the algorithm A_θ simulates the registration query in the game by running A_{BPCEC}, and then the obtained registration inquiry result is returned to A_{anony};

Tracking query: The adversary A_{anony} chooses a signature to request a query, and the algorithm A_θ simulates the tracking query in the attack game by running A_{rs} in the ring signature scheme and A_{FPK} in the FPK generation algorithm, and returns the obtained query result to A_{anony};

STEP 3: Challenge phase: The adversary A_{anony} selects PK_{SP} and score m. Algorithm A_θ first randomly selects $b \in \{0, 1\}$, executes A_{FPK}, A_{rs} in the FPK generation algorithm and our ring signature protocol of the challenge phase in the game, and finally gets the ring signature $Sign$ and returns it to the opponent A_{anony};

STEP 4: Guessing phase: After the inquiry phase, the adversary A_{anony} outputs a bit b'. If $b' = b$ exists, it indicates that the adversary A_{anony} successfully carried out the attack.

The success probability of adversary A_{anony} is:

$$
\begin{aligned}
&Adv_{A_{anony}}(k) = Pr[Exp_{A_{anony}}(k) = 1] \\
&= Pr[A_{annoy}(guess) = 1 \mid b = 1] \cdot Pr[b = 1] \\
&\quad + Pr[A_{anony}(guess) = 0 \mid b = 0] \cdot Pr[b = 0] \\
&= \frac{1}{2} Pr[\begin{matrix} A_{rs}(guess = 1) \\ A_{FPK}(guess = 1) \end{matrix} \quad | \quad b = 1] \\
&\quad + \frac{1}{2} Pr[\begin{matrix} A_{rs}(guess = 0) \\ A_{FPK}(guess = 0) \end{matrix} \quad | \quad b = 0] \\
&< \frac{1}{2}(Pr[A_{rs}(guess) = 1 \mid b = 1] + Pr[A_{rs}(guess) = 0 \mid b = 0]) \\
&+ \frac{1}{2}(Pr[A_{FPK}(guess) = 1 \mid b = 1] + Pr[A_{FPK}(guess) = 0 \mid b = 0]) \\
&= Pr[Exp_{A_{rs}}(k) = 1] + Pr[Exp_{A_{FPK}}(k) = 1] \\
&= Adv_{A_{rs}}(k) + Adv_{A_{FPK}}(k)
\end{aligned}
$$

If an attacker A_{BPCEC} successfully attacks the BPCEC, the attacker A_{FPK} can successfully attack FPK generation algorithm. However, the probability of A_{rs} successfully attacking the anonymity of our ring signature algorithm is still $1/n$, then A_{anony} wins in the anonymous simulation attack game of A2UA scheme in a probability of $1/2 + 1/n$. However, according to the assumptions that BPCEC satisfies the basic security properties, we concluded that the probability of A_{anony} successfully attacking can be ignored, so the scheme meets anonymity without considering TA. In other words, our scheme achieves conditional privacy.

Theorem 2. If the BPCEC satisfies the basic security features, FPK cannot be traced except TA, then the proposed A2UA scheme satisfies conditional traceability.

Proof 2. Define A_{trace} as the adversary (except TA) attacking the traceability simulation attack game of our scheme, A_{rs} as an opponent attacking the anonymity of our ring signature algorithm, A_{BPCEC} as the adversary who attacks the BPCEC, and A_{FPK} as the adversary attacking the FPK generation algorithm. Assuming A_{trace} successfully attacks the traceability of the scheme, a polynomial time algorithm $A_\tau \in (A_{rs}, A_{FPK})$ is defined, which has the ability to attack our ring signature algorithm and FPK generation algorithm. Through the query of A_{trace} and the A_τ's interaction in the traceability simulated attack game, A_τ is optimized repeatedly to successfully attack the ring signature and FPK generation algorithms. That is, if the adversary A_{trace} successfully attacks the traceability of the scheme, it means A_τ successfully attacks our ring signature algorithm and the FPK generation algorithm with a certain probability.

According to the steps defined by conditional traceability, the interaction between algorithm A_τ and adversary A_{trace} is as follows:

STEP 1: Generation phase: Algorithm A_τ executes the system generation algorithm, and generates public parameters PK_U of the traceability attack simulation game through the public parameters generated by A_{BPCEC}. Then A_τ sends PK_U to A_{trace};

STEP 2: Collusion phase: The adversary A_{trace} selects the user's private key SK_U to make the request. The algorithm A_τ simulates the collusion query in the attack game by running the A_{rs}, A_{FPK}, returns the obtained query result to A_{trace}, and then adds corruption members of the ring signature scheme to the list Ω.

STEP 3: Inquiry phase: The adversary A_{trace} can query the algorithm A_τ for polynomial time:

Registration query: The adversary A_{trace} requests the private key SK_U corresponding to user U, and the algorithm A_τ simulates the registration query in the game by running A_{BPCEC}, and then the obtained registration inquiry result is returned to A_{trace};

Signature query: The adversary A_{trace} selects user private key SK_U and score m to request a query, and algorithm A_τ returns signature $Sign$ to A_{trace} by running A_{BPCEC}, A_{rs}, A_{FPK} in the simulation attack game;

Traceability query: The adversary A_{trace} selects a published signature $Sign$ to request the query, and the algorithm A_τ simulates the tracking query in the attack game by running A_{rs}, A_{FPK} in the ring signature scheme and the FPK generation algorithm, and returns the result to A_{trace};

STEP 4: Challenge phase: Eventually, the adversary A_{trace} outputs a signature $Sign^*$ about the m^*. If any of the following conditions occur, the adversary A_{trace} succeeds in the attack.

Case 1: The published signature $Sign^*$ is valid, but the recovered user is invalid, that is, $Trac(Sign^*, SK_{TA}) = \perp$;

Case 2: The published signature $Sign^*$ is valid and has not been queried, but the recovered user is not in the collusion list, that is to say $Trac(Sign^*, SK_{TA}) = PK_i \notin \Omega$.

According to the interaction between the algorithm A_τ and the adversary A_{trace}, the success probability of the adversary A_{trace} is:

$$
\begin{aligned}
&Avd_{A_{trace}}(k) \\
&= Pr[Exp_{A_{trace}}(k) = 1] \\
&= Pr\{[Ver(Sign^*, score^*, SK_{TA}) = 1, \\
&\qquad Trac(Sign^*, SK_{TA}) = \perp]\} \\
&\quad + Pr\{[Ver(Sign^*, score^*, SK_{TA}) = 1, \\
&\qquad Trac(Sign^*, SK_{TA}) = PK_i \notin \Omega]\} \\
&= Pr[Exp_{A_{rs}}(k) = 1] \cdot Pr[Exp_{A_{FPK}}(k) = 1] \\
&= Avd_{A_{rs}}(k) \cdot Avd_{A_{FPK}}(k)
\end{aligned}
$$

Therefore, if the attacker A_{BPCEC} successfully attacks the BPCEC, the attacker A_{rs} successfully attacks the complete anonymity of the ring signature scheme, and the attacker A_{FPK} successfully attacks the FPK generation algorithm, then A_{trace} wins in the traceability simulation attack game. However, according to the assumptions that BPCEC satisfies the basic security properties, we concluded that the probability of A_{trace} successfully attacking can be ignored, so the proposed scheme meets unlinkability without considering TA. In other words, our scheme achieves conditional traceability.

Theorem 3. If all the crypto-algorithms such as BPCEC, ECC, etc., satisfy the basic security features, then complete protection of content privacy can be achieved in our proposed A2UA scheme.

Proof 3. Define $A_{privacy}$ as the adversary attacking the content privacy simulation attack game of our scheme,

A_{BPCEC} as the adversary who attacks the BPCEC, A_{ECC} as the adversary who attacks the ECC scheme. During the authentication between U and SP, the two authenticating messages are secured by BPCEC, ECC. Therefore, for the adversary $A_{privacy}$, the probability of successfully attacking content privacy is:

$$\begin{aligned} & Avd_{A_{privacy}}(k) \\ &= Pr[Exp_{A_{privacy}}(k) = 1] \\ &= Pr[Ver(Sig^1_{ECC}) = 1] + Pr[Ver(Sig^2_{BPCEC}) = 1] \\ &= Pr[Exp_{A_{ECC}}(k) = 1] + Pr[Exp_{A_{BPCEC}}(k) = 1] \\ &= Avd_{A_{ECC}}(k) \cdot Avd_{A_{BPCEC}}(k) \end{aligned}$$

Therefore, if the attacker A_{BPCEC} successfully attacks the BPCEC, the attacker A_{ECC} successfully attacks the ECC algorithm, then $A_{privacy}$ wins in the content privacy simulation attack game. However, according to the assumptions about the security, the probability of $A_{privacy}$ successfully attacking can be ignored. As a result, the A2UA scheme accomplishes complete protection of content privacy.

6. Performance Analysis

In this section, we analyze the performance of our scheme in terms of computation, communication and Gas cost. Moreover, we make a comparison in security and privacy features between ours scheme and several representative schemes.

6.1. Computation Cost

In our scheme, system initialization phase, registration phase, anonymous authentication phase, credit building phase and accountability phase are involved. Because the first two phases happen rarely, their comparison is not be given in this section. Further, although most phases interact with blockchain, the blockchain system varies and updates swiftly, as a result, a miner's computation cost is considered only. Therefore, for comparison with other schemes in the similar settings, the computation cost is mainly explained as the time costs during the authentication phase and accountability phase. Lots of schemes focus on anonymous authentication, while the schemes in [27–31] include the anonymous authentication and accountability phases. The scheme of [27] adopt scale multiplication and modular exponentiation operations related to ECC to achieve authentication and accountability. Modular exponentiation operation and zero knowledge proof are involved in the scheme of [28]. Zhang *et al.* [29] employed the secret sharing based on threshold to realize inspection of evil behavior. Cui *et al.* [30] used XOR and one-way hash function operation to satisfy the need of authentication and accountability. In [31], Lin *et al.* introduced scale multiplication operation related to ECC and zero knowledge proof to realize authentication and accountability. From the perspective of the similarity of the authentication and accountability, we choose the schemes [17, 18, 27, 28, 30] to do performance analysis and comparison. To achieve persuasive expression of computation comparison, the same symbols and parameters in [30, 39] are introduced: T_{bp} denotes bilinear pairing operation, T_{bpa} means point addition operation related to the bilinear pairing, T_{mtp} is MapToPoint operation related to the bilinear pairing, T_{ecm} expresses scale multiplication operation related to ECC, T_n represents number multiplication operation, T_e indicates modular exponentiation operation, T_h signifies one-way hash function operation, T_{rng} expresses random number generator, T_s denotes symmetric encryption/decryption operations. And their time costs are as follows: T_{bp} = 5.086 ms, T_{ecm} = 0.3218 ms, T_{bpa} = 0.0018 ms, T_n = 0.1 ms, T_e = 6.014 ms, T_h = 0.001 ms, T_{rng} = 0.5 ms, T_{mtp} = 0.0992 ms, T_s = 0.276 ms. Further, compared with the above operations, the time for performing XOR operation is negligible, we do not take this into account in calculation of computation cost. The detailed computation costs for each entity of authentication and accountability phase for all the schemes are illustrated in **Table** 3 and **Fig**. 4.

Table 3
Computation Cost (ms).

	Authentication		Accountability	Total
	Requester	Receiver	TA	
[17]	$4T_n + 2T_{ecm} + T_{bpa}$	$2T_{bpa} + 3T_n + T_{bp} + 8T_h + 2T_s$	$Null$	7.00
[18]	$6T_{ecm} + 3T_{bp}$	$4T_{mtp} + T_e + 4T_{bp} + 2T_h$	$Null$	43.95
[27]	$14T_{ecm} + 6T_e + 2T_{bp}$	$10T_{ecm} + 5T_e + 6T_{bp}$	$2T_{ecm} = 0.64$	115.21
[28]	$26T_e + 2T_h + 3T_{bp}$	$22T_e + 2T_h + 7T_{bp}$	$2T_e = 12.03$	351.56
[30]	$3T_{ecm} + 8T_h$	$3T_{ecm} + 7T_h$	$2T_{ecm} + 10T_h = 0.65$	2.60
A2UA	$4T_h + 3T_{mtp} + 5T_{ecm} + 3T_n + 2T_{bp} + 3T_{rng}$	$2T_h + 4T_{ecm} + 2T_n + 2T_{mtp} + 4T_{bp} + 2T_{rng}$	$T_{ecm} = 0.32$	37.23

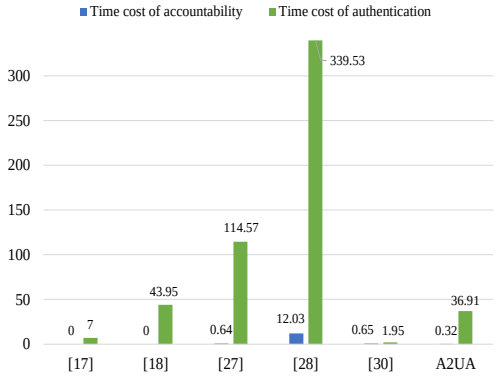


Figure 4: Comparison of computation cost (ms).

We analyze Cui *et al.*'s scheme [30] and our scheme in detail, and the specific computational cost in others can be implemented similarly. Cui *et al.* [30] embedded account-

ability in the authentication process, and three entities are involved in their scheme: requester (vehicle), authenticator (R/C/V) and TA. The scheme stipulates requester to execute three scale multiplication about the ECC and eight one-way hash function operations, that is, the computation time of requester is $3T_{ecm} + 8T_h = 0.9734$ ms. Meanwhile, the authenticator is required to perform three scale multiplication about the ECC and seven one-way hash function operations, consequently, the execution time of authenticator is $3T_{ecm} + 7T_h = 0.9724$ ms. For TA, it needs to carry out two scale multiplication about the ECC and ten times of one-way hash function operations; namely, the calculation time of TA is $2T_{ecm} + 10T_h = 0.6536$ ms. Therefore, the total execution time in [30] for authentication and accountability is about 2.5994 ms.

In our scheme, there are also three entities: requester (U), receiver (SP), and auditor (TA). The computation time needed in U is four one-way hash function operations, three random number generator operations, five scale multiplication operations related to ECC, three number multiplication operations, two bilinear pairing operations and three MapToPoint/PointToMap operations related to the bilinear pairings, accordingly, the execution time is $4T_h + 3T_{rng} + 5T_{ecm} + 3T_n + 2T_{bp} + 3T_{mtp} = 13.8826$ ms. The SP and a miner are totally required to perform two one-way hash function operations, two random number generator operations, four scale multiplication operations related to ECC, two number multiplication operations, four bilinear pairing operations and two MapToPoint/PointToMap operations related to the bilinear pairings, accordingly, the execution time is $2T_h + 2T_{rng} + 4T_{ecm} + 2T_n + 4T_{bp} + 2T_{mtp} = 23.0316$ ms. Therefore, the total running time required in our scheme during the entire authentication phase is approximately 36.912 ms. To reveal the real identity of a malicious user as Cui *et al.*'s scheme [30] does, TA is required to carry out one scale multiplication operation related to ECC in the accountability phase, the computation time is $T_{ecm} = 0.3218$ ms.

From **Table** 3 and **Fig**. 4, the time cost of accountability (reveal the real identity of a user) of our A2UA protocol is the lowest, and the one of authentication of our scheme is lower than the ones of [27, 28]. Although the time cost of authentication in the scheme [30] is lower than ours, five messages are sent over network which takes more communication time, since two messages are needed in our scheme. In addition, the problems (such as information leakage) from the single point of failure are existed inevitably in its scheme [30] since the centralized trusted authority is used.

6.2. Communication Cost for Authentication

For the sake of comparability, all the schemes assume that both p and the generated random number are 160 bits and the elements in the elliptic curve are also 160 bits. The output length for bilinear pairing one-way hash operation is 160 bits. And according to the current mainstream settings, the timestamp size is set to 32 bits, and the output size of the general hash function (h) is set to 256 bits. The detailed communication costs during the authentication phase are shown in the **Fig**. 5.

Table 4
Parameters Setting

Parameter	Type	Length(bits)
$Time, t$	Number	32
$p, r, i, c, k, I_8, USCC$	Number	160
$FPK, FSK,$ $PK_{SP}, SK_{SP},$ $P_{pub}, I_7, I_7^*, I_9, I_{11}$	Point	320
rID, V_u, V_u^*, V_{SP}	Number	256
I_{10}	Number	1,440

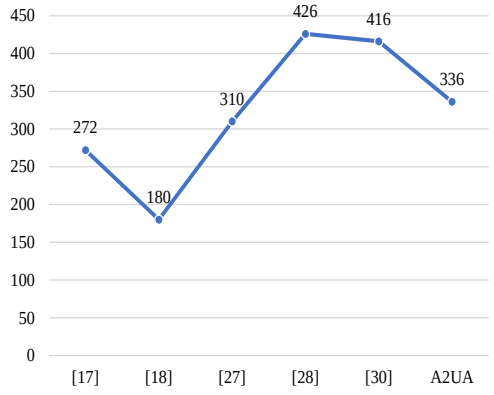


Figure 5: Comparison of communication cost (bytes).

Similar to ***6.1***, we also only analyze the cost of communication of Cui *et al.*'s scheme [30] and ours, and the communication cost in other schemes can be calculated in the same way. In Cui *et al.* [30] authentication process, the communication messages are from M_1 to M_5, where $M_1 = \{M_i, PID_i, X, \eta, tt_i\}$, $M_2 = \{PID_j, Y, \theta, tt_i\}$, $M_3 = \{CID_i, AID_j, \alpha, \beta, X, tt_i\}$, $M_4 = \{AID_j, Y, \beta, \gamma, tt_i\}$ and $M_5 = \{\lambda\}$. Due to $\langle X, Y\rangle \in G, \langle PID_i, PID_j, \eta, \theta, \alpha, \beta, \gamma, \lambda, CID_i, AID_j\rangle$ are the results of one-way hash operation, and tt_i denotes the latest timestamp, consequently, the total communication overhead is $40 \times 4 + 20 \times 12 + 4 \times 4 = 416$ bytes.

In our scheme, the detailed settings of parameters in use are shown in **Table** 4. And the authentication process involves two messages which are M_1 and M_2: $M_1 = \{FPK, I_9, I_{10}\}$, $M_2 = \{I_{11}, t_2, V_{SP}\}$. By the way, the transactions (T_C, T_V) involved in authentication phase are not computed in the communication cost since they are introduced only for later accountability.

According to **Table** 4, the length of $FPK, I_9, I_{10}, I_{11}, t_2, V_{SP}$ are 320 bits, 320 bits, 1,440 bits, 320 bits, 32 bits and 256 bits. Therefore, the total scheme communication cost is $320 + 320 + 1,440 + 320 + 32 + 256 = 2,688$ bits, that is 336 bytes.

As illustrated in **Fig**. 5, the communication costs of ours and the other three auditable anonymous authentication schemes [27, 28, 30] are more than the ones in the anonymous authentication schemes [17, 18], because they

Table 5
Blockchain-based performance evaluation of our scheme

Phase	Writer	Length of data to be written (bytes)	Gas cost
4.3: Registration	TA	$\{PK_U\} = 40$	22,940
		$\{USCC\} = 20$	22,488
4.4: Anonymous Authentication	SP	$\{Cert_U\} = 304$	27,817
	U	$\{result\} = 1$	22,344
4.5: Credit Building	SP	$\{m_i^U, FPK, Cert_U\} = 348$	28,557
	U	$\{m_j^{SP}, PK_{SP}, Sign_j, t_r\} = 368$	29,009
	TA	$\{m_{new}^U, E^1_{PK_U}(Sig^1_{SK_{TA}}(m_{new}^U \parallel t_{m_{new}^U})), k\}$ $+\{m_{new}^{SP}, Sig^1_{SK_{TA}}(m_{new}^{SP} \parallel t_{m_{new}^{SP}}), k\} = 164$	25,276
4.6: Accountability	TA	$\{AUU\_result\} + \{AUSP\_result\} = 32$	22,516

embed extra information for achieving accountability. And the communication cost of our scheme is less than the ones in [28, 30] and more than the one in [27].

6.3. Gas cost of our scheme

Blockchain is introduced in our scheme, and it is hardly for us to search for other current blockchain-based auditable anonymous authentication scheme, so we evaluate the Gas cost of our scheme based on a public Ethereum testnet (Rinkeby) to demonstrate the practical performance.

In our scheme, the system initialization is only performed once since the establishment of the system, therefore, the Gas cost of the initialization is not been counted. And the phases of registration, anonymous authentication, credit building and accountability are often executed, so we obtain the Gas cost of them to evaluate our scheme. The result is shown in the **Table** 5.

From the **Table** 5, in the registration phase, the TA records U's and SP's real identities $\{PK_U\}$ and $\{USCC\}$ in the blockchain, which are 40 and 20 bytes respectively, and the Gas cost are 22,940 and 22,488. Then, in the anonymous authentication phase, U and SP mutually authenticate each other's identities, and after the authentication, they record the authentication result in the blockchain. In detail, the SP first writes a $\{Cert_U\}$ with a length of 304 bytes into the blockchain, consuming 27,817 Gas, and then U consumes 22,344 Gas to record whether the authentication is passed, that is, a 1-byte *result*. Next, once authenticated, U and SP give authentic scores (AS) to each other's identity authenticity and service, and the TA updates their AS. Specifically, the SP sends 348 bytes of $\{m_i^U, FPK, Cert_U\}$ by the transaction T_{SS} to the blockchain, the U transmits 368 bytes of $\{m_j^{SP}, PK_{SP}, Sign_j, t_r\}$ by the transaction T_{US} to the blockchain, and the TA needs to record 164 bytes of $\{m_{new}^U, E^1_{PK_U}(Sig^1_{SK_{TA}}(m_{new}^U \parallel t_{m_{new}^U})), k\} + \{m_{new}^{SP}, Sig^1_{SK_{TA}}(m_{new}^{SP} \parallel t_{m_{new}^{SP}}), k\}$ with the two transaction T_Us to updates their AS, which use 28,557, 29,009 and 25,276 Gas respectively. Finally, when malicious behavior is audited, TA writes 32-byte $\{AUU\_result\} + \{AUSP\_result\}$ in the blockchain to record the audit result, consuming a total of 22,516 Gas.

Table 6
A comparison in security and privacy features.

Schemes	F1	F2	F3	F4	F5	F6
[17]	√	√	×	×	√	√
[18]	√	√	√	×	√	√
[27]	√	√	√	√	×	√
[28]	√	√	√	√	×	×
[30]	√	√	√	√	×	√
A2UA	√	√	√	√	√	√

[1] **F1**: Mutual authentication; **F2**: User's anonymity in the channel; **F3**: User's anonymity in the receiver; **F4**: Accountability; **F5**: Resistant impersonation attack; **F6**: Resistant replay attack.
[2] '√' means realized and '×' denotes unfulfilled.

In general, the Gas cost of recording the key information in the registration, anonymous authentication, credit building and accountability phases of our scheme is relatively low, so the scheme has good feasibility in practice.

6.4. Security and Privacy Features Comparison

The security and privacy features of our proposed authentication scheme and several representative schemes are compared in the **Table** 6.

As can be seen from the **Table** 6, these schemes all implement mutual authentication and user's anonymity in the communication channel. But the receiver in [17] can restore the user's identity, which the user may still be traced or disclosed in the physical world. In addition, as shown in **Table** 3, Odelu *et al.* [17] and Arfaoui *et al.* [18] ignore the accountability of malicious entities. Moreover, the schemes in [27], [28] and [30] could not resist impersonation attack since the user's private key or authentication information is public to the third party and the third party could use it to imitate user behavior. Lastly, the user's authentication certificate could be replayed by attackers to achieve user authentication in the scheme of [28]. Compared to the above schemes, our A2UA not only designs an auditable anonymous user authentication protocol, but also resists attacks.

7. Conclusions

To deal with the difficulty of tracing the illegal activities caused by anonymous authentication, we proposed an auditable anonymous user authentication (A2UA) protocol based on blockchain to achieve both anonymity and accountability in the procedure of authentication. In A2UA, a user (U) and a service provider (SP) perform mutually authentication where a disposable FPK of U is used in each session. And the FPK is generated with bilinear pairing cryptosystem, ECC algorithm, random numbers and U's partial authentication factor from TA. In addition, ring signature combined with bilinear pairing is adopted by U to rate SP. In this way, the legality, protection of content privacy, conditional anonymity, and conditional unlinkability of U are satisfied. Further, we not only introduce blockchain and TA to audit U and SP to prevent malicious activities, but also let U and SP score each other after each session to regulate each other's behavior. Security analysis and performance analysis show that A2UA achieves security objectives with lower computation cost and medium communication cost than the tiptop schemes, and A2UA has better security and privacy features.

For the sake of lowering the risk from the semi-honest TA, TA has no right to access the detailed content between U and SP in the scheme, as a result, the problem caused by the collusion of malicious U and rancorous SP is open. Therefore, we will employ secure multiparty computing to improve it in the future work.

Declaration of competing interest

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

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (No. 61872120), Discipline Construction of Zhejiang Provincial Key University-Cyberspace Security (No. GK198800299013) and Zhejiang Province Natural Science Foundation (No. LY19F020039).

References

[1] Whitfield Diffie, Paul C Van Oorschot, and Michael J Wiener. Authentication and authenticated key exchanges. *Design. Code. Cryptogr.*, 2(2):107–125, 1992.

[2] Jennifer G. Steiner, B. Clifford Neuman, and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In *Proc. of the USENIX Winter Conf.*, pages 191–202. USENIX Association, 1988.

[3] David Recordon and Drummond Reed. Open ID 2.0: A platform for user-centric identity management. In *Proc. 2nd. ACM Workshop Digit. Identity Manage.*, pages 11–16, 2006.

[4] Dick Hardt et al. The OAuth 2.0 authorization framework. Technical report, RFC 6749, October, 2012.

[5] FIDO Alliance. FIDO. https://fidoalliance.org/, 2012.

[6] Nai-Wei Lo and Jia-Lun Tsai. An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. *IEEE trans. Intell. Transp. Syst.*, 17(5):1319–1328, 2015.

[7] Pandi Vijayakumar, Victor Chang, L Jegatha Deborah, Balamurugan Balusamy, and PG Shynu. Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks. *Future Gener. Comput. Syst.*, 78:943–955, 2018.

[8] Jun Zhou, Zhenfu Cao, Zhan Qin, Xiaolei Dong, and Kui Ren. LPPA: Lightweight privacy-preserving authentication from efficient multi-key secure outsourced computation for location-based services in VANETs. *IEEE Trans. Inf. Forensics Secur.*, 15:420–434, 2019.

[9] Xiong Li, Maged Hamada Ibrahim, Saru Kumari, Arun Kumar Sangaiah, Vidushi Gupta, and Kim-Kwang Raymond Choo. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. *Comput. Netw.*, 129:429–443, 2017.

[10] Aiqing Zhang and Xiaodong Lin. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. *J. Med. Syst.*, 42(8):140, 2018.

[11] Thein Than Thwin and Sangsuree Vasupongayya. Blockchain-based access control model to preserve privacy for personal health record systems. *Secur. Commun. Netw.*, 2019, 2019.

[12] Prosanta Gope and Biplab Sikdar. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. *IEEE Internet Things J.*, 6(1):580–589, 2018.

[13] Muhammad Naveed Aman, Mohammed Haroon Basheer, and Biplab Sikdar. Data provenance for iot with light weight authentication and privacy preservation. *IEEE Internet Things J.*, 6(6):10441–10457, 2019.

[14] Mohanad Dawoud and D Turgay Altilar. HEADA: A low cost RFID authentication technique using homomorphic encryption for key generation. *Secur. Commun. Netw.*, 9(17):4182–4191, 2016.

[15] Prosanta Gope and Tzonelih Hwang. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Trans. Ind. Electron.*, 63(11):7124–7132, 2016.

[16] Debiao He, Sherali Zeadally, Neeraj Kumar, and Wei Wu. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Trans. Inf. Forensics Secur.*, 11(9):2052–2064, 2016.

[17] Vanga Odelu, Sourav Saha, Rajendra Prasath, Lakshminarayana Sadineni, Mauro Conti, and Minho Jo. Efficient privacy preserving device authentication in WBANs for industrial e-health applications. *Comput. Secur.*, 83:300–312, 2019.

[18] Amel Arfaoui, Omar Rafik Merad Boudia, Ali Kribeche, Sidi-Mohammed Senouci, and Mohamed Hamdi. Context-aware access control and anonymous authentication in WBAN. *Comput. Secur.*, 88:101496, 2020.

[19] Xiong Li, Jianwei Niu, Md Zakirul Alam Bhuiyan, Fan Wu, Marimuthu Karuppiah, and Saru Kumari. A robust ECC-based provable secure authentication protocol with privacy preserving for industrial internet of things. *IEEE Trans. Industr. Inform.*, 14(8):3599–3609, 2017.

[20] Libing Wu, Jing Wang, Kim-Kwang Raymond Choo, and Debiao He. Secure key agreement and key protection for mobile device user authentication. *IEEE Trans. Inf. Forensics Secur.*, 14(2):319–330, 2018.

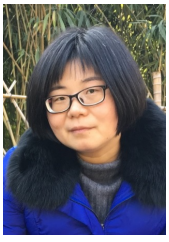
[21] Mengxia Shuai, Nenghai Yu, Hongxia Wang, and Ling Xiong. Anonymous authentication scheme for smart home environment with provable security. *Comput. Secur.*, 86:132–146, 2019.

[22] Qiuyun Lyu, Ning Zheng, Huaping Liu, Can Gao, Si Chen, and Junliang Liu. Remotely access 'my' smart home in private: An anti-tracking authentication and key agreement scheme. *IEEE Access*, 7:41835–41851, 2019.

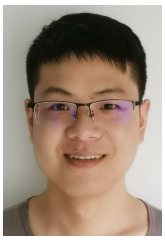
[23] Hamza Hammami, Sadok Ben Yahia, and Mohammad S Obaidat. A lightweight anonymous authentication scheme for secure cloud computing services. *The J. of Supercomput.*, 77:1693–1713, 2021.

[24] Azees Maria, Vijayakumar Pandi, Jeatha Deborah Lazarus, Marimuthu Karuppiah, and Mary Subaja Christo. Bbaas: Blockchain-based anonymous authentication scheme for providing secure communication in vanets. *Secur. and Commun. Netw.*, 2021, 2021.

[25] Asmaa Abdallah and Xuemin Sherman Shen. A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Trans. Smart Grid*, 9(1):396–405, 2016.
[26] Aldo Cassola, Erik-Oliver Blass, and Guevara Noubir. Authenticating privately over public Wi-Fi hotspots. In *Proc. ACM Conf. Computer Commun. Secur.*, pages 1346–1357, 2015.
[27] Qingyou Yang, Kaiping Xue, Jie Xu, Jiajie Wang, Fenghua Li, and Nenghai Yu. AnFRA: Anonymous and fast roaming authentication for space information network. *IEEE Trans. Inf. Forensics Secur.*, 14(2):486–497, 2018.
[28] Zhen Wang, Jia Fan, Lin Cheng, Hongzhang An, Haibin Zhang, and Junxiang Niu. Supervised anonymous authentication scheme. *Journal of Software*, 30(6):1705–1720, 7 2019.
[29] Jianyi Zhang, Zhiqiang Wang, Zhili Xu, Yafei Ouyang, and Tao Yang. A regulatable digital currency model based on blockchain. *J. of Comput. Res. and Develop.*, 55(10):2219, 2018.
[30] Jie Cui, Xiaoyu Zhang, Hong Zhong, Jing Zhang, and Lu Liu. Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment. *IEEE Trans. Inf. Forensics Secur.*, 15:1654–1667, 2019.
[31] Chao Lin, Debiao He, Xinyi Huang, Muhammad Khurram Khan, and Kim-Kwang Raymond Choo. DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain. *IEEE Trans. Inf. Forensics Secur.*, 15:2440–2452, 2020.
[32] Ping Yu, Wei Ni, Guangsheng Yu, Hua Zhang, Ren Ping Liu, and Qiaoyan Wen. Efficient anonymous data authentication for vehicular ad hoc networks. *Secur. and Commun. Netw.*, 2021, 2021.
[33] Jan Camenisch et al. Specification of the identity mixer cryptographic library. Technical report, Tech. rep, 2010.
[34] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *International conference on the theory and applications of cryptographic techniques*, pages 453–474. Springer, 2001.
[35] David Galindo. Boneh-franklin identity based encryption revisited. In *Int. Colloq. on Automata, Languages, and Programming, ICALP*, pages 791–802. Springer, 2005.
[36] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Proc. Annu. Int. Cryptol. Conf*, pages 213–229. Springer, 2001.
[37] Qiuyun Lyu, Yizhen Qi, Xiaochen Zhang, Huaping Liu, Qiuhua Wang, and Ning Zheng. SBAC: A secure blockchain-based access control framework for information-centric networking. *J. Netw. Comput. Appl.*, 149:102444, 2020.
[38] Chao Lin, Debiao He, Xinyi Huang, Muhammad Khurram Khan, and Kim-Kwang Raymond Choo. A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems. *IEEE Access*, 6:28203–28212, 2018.
[39] Shaoyong Guo, Xing Hu, Song Guo, Xuesong Qiu, and Feng Qi. Blockchain meets edge computing: A distributed and trusted authentication system. *IEEE Trans. Inf. Forensics Secur.*, 16(3):1972–1983, 2019.

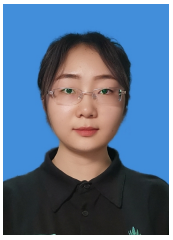


Qiuyun Lyu is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. She received the bachelor's and master's degrees from Chang'an University, in 2000 and 2003, respectively. She is an associate professor of the School of Cyberspace, Hangzhou Dianzi University. Her current research interests include anonymous authentication, privacy enhancing technology and blockchain.

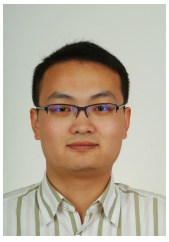


Hao Li received the B.S. degree from School of Cyberspace from Hangzhou Dianzi University, Hangzhou, China, in 2019, where he is currently pursuing the M.S. degree with the school of cyberspace. He is excited about identity authentication, privacy protection, and blockchain.

Zhining Deng received his B.S. degree in information security from Hangzhou Dianzi University, Hangzhou, China, in 2016. He is currently engaged in cybersecurity in Shanghai Shizhuang Information Technology Co., Ltd and obtained the certifications of CISSP, ISO27001, ITIL4, Prince2, etc. His research interests include network security, privacy enhancing technology, key management, and blockchain.



Jingyu Wang is currently pursuing the B.S. degree with the School of Cyberspace from Hangzhou Dianzi University, Hangzhou, China. Her research interests include identity authentication, privacy protection and blockchain.



Yizhi Ren received his PhD in Computer software and theory from Dalian University of Technology, China in 2011. He is currently an associate professor with School of Cyberspace, Hangzhou Dianzi University, China. From 2008 to 2010, he was a research fellow at Kyushu University, Japan. His current research interests include: network security, complex network, and trust management. Dr. REN has published over 60 research papers in refereed journals and conferences. He won IEEE Trustcom 2018 Best Paper Award, CSS2009 Student Paper Award and AINA2011 Best Student paper Award.

Ning Zheng is the vice president of Hangzhou Dianzi University, Hangzhou, China. He has authored over 70 referred journal and conference papers. His research interests are privacy enhancing information management system and network information security.

Junliang Liu used to serve Alibaba Co., Ltd, Hangzhou, China, and now is an expert in the security department of Hangzhou Meichuang Technology Co., Ltd, Hangzhou, China. His current research interests include protocol security analysis, privacy enhancing technology, authentication protocol design and Data Security Maturity Model.

Huaping Liu received the B.S. and M.S. degrees in electrical engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987 and 1990, respectively, and the Ph.D. degree in electrical engineering from New Jersey Institute of Technology, Newark. Since September 2001, he has been with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, where he is currently a professor. His research interests include privacy enhancing technology in communication systems and multiuser communications.