AATM: An Anonymous Authentication Protocol for Time Span of Membership with Self-blindness and Accountability

Qiuyun Lyu, Xiwen Liang, Shaopeng Cheng, Fu Li, Yizhi Ren, Chengli Xu, Weizhi Meng, *Member, IEEE*, and Duohe Ma

***Abstract*—Internet of Things (IoT) devices using subscription services (e.g. connected vehicles accessing entertainment programs) often purchase membership credentials from service providers with limited usage counts or validity periods, we call them pay-per-use or time span of membership services. However, users' access records, usage preferences, and habits are collected by network adversarys or membership providers for creating users' profiles, targeted advertising, and even for being sold maliciously. To deal with these problems, lots of anonymous authentication protocols are proposed to provide users with pseudonyms to conceal their real identities. Although these protocols effectively prevent network adversarys from compromising users' privacy, membership service providers can still gather users' behavioral privacy via their membership credentials. Therefore, several scholars proposed k-times anonymous authentication protocols and self-blind credentials to enhance users' privacy protection, but the k-times anonymous authentication protocols are only for pay-per-use membership services and the schemes of self-blind credentials are lack of regulating malicious users. To address these issues, this article proposes an anonymous authentication protocol for time span of membership (AATM) with self-blindness and accountability. Specifically, we utilize Structure Preserving Signatures on Equivalence Classes (SPS-EQ) and Signatures with Flexible Public Key (SFPK) to build accountable, self-blinding credentials that ensure that every time a user visits a member, he or she can create a brand new identity on their own, which not only prevents users from being linked by service providers, but also supports conditional fair regulation. Security and performance analyses show that AATM is better than the state-of-the-art schemes in terms of security and privacy-preserving capabilities, and its computation cost also meets the practical application requirements.**

***Index Terms*—Authentication, Anonymous, Accountability, Self-blind credentials, Conditional privacy-preserving.**

I. Introduction

SERVICE providers often offer personalized membership services to the subscribed Internet of Things (IoT) devices (e.g. vehicles requesting personalized entertainment programs from the service provider). A user is able to purchase a membership credential with a fixed number of uses or limited duration from the service provider, which is called pay-per-use or time span membership services. This credential can then be used to authenticate the user's membership status with the service provider during subsequent accesses to the service.

Traditional authentication protocols, such as Kerberos [1], Open ID [2] and Oauth [3], are first proposed to provide support for users to authenticate their membership identity to service providers. However, users' access records, usage behaviors, and personal privacy data are collected by service providers and adversarys for the purposes of user profiling, targeted advertising, and even for profit-making.

In order to alleviate privacy concerns associated with traditional authentication, numerous researchers have conducted extensive research on privacy-preserving anonymous authentication protocols, especially after the implementation of GDPR[1] and CCPA[2] regulations. In general, anonymous authentication protocols can be categorized into three types: traditional anonymous authentication protocols, conditional anonymous authentication protocols, and membership anonymous authentication protocols. Traditional anonymous authentication protocols typically employ pseudonyms to indicate a user's identity and can be further classified into two categories: static pseudonym-based authentication protocols [4], [5], [6], [7], [8], [9] and dynamic pseudonym-based authentication protocols [10], [11], [12], [13], [14], depending on whether the pseudonyms remain unchanged or not during each authentication process. Although traditional anonymous authentication protocols hide the user's identity through pseudonyms, they fail to provide accountability for malicious users. To mitigate this risk, many scholars have proposed conditional anonymous authentication protocols [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], where a trusted third party is adopted as a regulator to uncover the real identity of adversaries. Unfortunately, the only one regulator in the schemes is vulnerable to the single point of failure.

In addition, both traditional anonymous authentication protocols and conditional anonymous authentication protocols achieve only the anonymization of a member's identity. That is, they only prevent adversaries from identifying a user by sniffing data streams, service providers can still link users via

Qiuyun Lyu, Xiwen Liang, Shaopeng Cheng, Yizhi Ren and ChengLi Xu are with the School of Cyberspace, Fu Li is with the School of Student Affairs, Hangzhou Dianzi University, Hangzhou 310018, China, Qiuyun Lyu is also with Pinghu Research Institute of Hangzhou Dianzi University, Jiaxing 314213, China, and Yizhi Ren is also with the Department of New Networks, Peng Cheng Laboratory, Shenzhen 518000, China. E-mail: {laqyzj, chengshaopeng, lifu, renyz, wchenglixu}@hdu.edu.cn, xiwenliang8@163.com

Weizhi Meng, School of Computing and Communications, Lancaster University, United Kingdom. Email: weizhi.meng@ieee.org.

Duohe Ma, Institute of Information Engineering Chinese Academy of Sciences. Email: maduohe@iie.ac.cn

(corresponding author: Yizhi Ren and Weizhi Meng)

[1]https://www.gdpreu.org

[2]https://oag.ca.gov/privacy/ccpa

his membership credential. Moreover, conditional anonymous authentication protocols only focus on regulating malicious users, lacking of consideration for the security of service providers and membership credentials. Therefore, these anonymous authentication protocols cannot meet a user's need for the privacy and security requirements of membership services. 1) $Privacy\ requirements$. While enjoying membership services, a user expects his private information such as usage preferences, browsing history, and personal characteristics to be protected. Anyone including a service provider cannot usurp in an unauthorized way. 2) $Security\ requirements$. When a dishonest membership provider deny the validity of a user's rightful membership credential, the user looks forward to safeguard his rights.

In order to meet privacy requirements, the k-times anonymous authentication protocols [27], [28], [29], [30], one type of membership anonymous authentication protocol, provide privacy support for pay-per-use membership services, where a user accesss services anonymously up to k times. However, k-times anonymous authentication protocols not only are lack of support for time span membership services but also fail to prevent the user from being linked by the service provider. To regulate dishonest membership provider, Huang et al. [30] introduced identity list and application log into k-times anonymous authentication protocols, but they still failed to solve the problem of user linkage. Even though the self-blind credentials [31], [32], [33], [34] prevent the linkage of a user from membership service providers, they do not explicitly indicate the credential's validity period or the number of times for use. Furthermore, the above schemes are unable to provide accountability for malicious sharing of a user's membership credential.

Although these schemes provide privacy protection for users when accessing membership services, they do not enable users to access time span membership services without being linked by a service provider, meanwhile lacking regulation of malicious users and service providers. Therefore, we focused on the privacy and security requirements of a user's membership services, providing the user with a "zero-knowledge membership identity" that allows him anonymously access to time span membership services without being linked by the service provider. Additionally, using Structure-Preserving Signatures on Equivalence Classes with Signatures with Flexible Public Key to bind the membership credentials of users to the identity, combined with threshold secret sharing and blockchain technology, we achieved distributed accountability of users and service providers to effectively regulate the misuse of anonymous membership credentials.

A. Contribution

The major contributions of this paper can be summarized as:

1) Our work achieves anonymous access in time span membership authentication on the scenarios where a service provider provides a personalized information service to a IOT device. Specifically, our protocol allows a user to anonymously access the service with an unlimited number of times in the valid period of their membership credentials.
2) Accountable self-blind credentials are constructed to allow a user to update his membership credentials locally to an equivalent and unlinkable new credential, which mitigates the risk of being linked by the service provider. At the same time, both malicious users and service providers are regulated in a distributed way to guarantee open and fair audit.
3) Security and performance analysis results show that AATM achieves better privacy protection than state-of-the-art schemes, and the computational overhead meets the practical requirements.

The structure of this article is as follows. Section II describes the relate work. Section III introduces the preknowledge. Section IV defines the model of our planned system, the security model, and the security objectives. Section V describes the details of the proposed protocol. In Sections VI and VII, we analyze the security of the proposed protocol and compare it with the state-of-the-art privacy-protecting authentication protocols in terms of computation cost as well security and privacy features. Section VIII concludes the article.

II. Related Work

A. Traditional anonymous authentication protocol

Traditional anonymous authentication protocols are first designed to implement anonymous access services for users by pseudonyms, which can be further classified into static pseudonym-based anonymous authentication protocols [4], [5], [6], [7], [8], [9] and dynamic pseudonym-based anonymous authentication protocols [10], [11], [12], [13], [14] depending on whether the pseudonym changes or not with each access.

Static pseudonym-based anonymous authentication protocols use a static identifier to represent the user's identity. For example, to avoid the leakage of user privacy, the public key was used to represent user's identity in the Bitcoin [4] system, and the use of a static string to represent a car pressure sensor [5]. Similarly, dummy identities are employed to provide anonymity for mutual authentication among vehicles in vehicular ad hoc networks [6] and authentication between vehicles and roadside unit (RSU) in VANETs communications [7]. Although anonymity authentication protocols utilize a permanently fixed static pseudonym to hide a users identity, adversaries can still track his behaviors by tracing the static pseudonym.

In order to address the problem of linking a users identity through a fixed pseudonym, dynamic pseudonym-based anonymous authentication protocols are proposed. Gope et al. [11] designed a lightweight authentication protocol for IoT devices, in which after each authentication, the device obtains a new pseudonym from the server for the next authentication. Li et al. [12] proposed a protocol for Industrial Internet of Things, that a random number was selected to calculate an authentication factor in order to prevent users' behaviors from being linked. However, the use of dynamic pseudonyms [11], [12], that is, one pseudonym for one session, also makes it

difficult to identify users and reduces the cost of evil for malicious users.

B. Conditional anonymous authentication protocol

Conditional anonymous authentication protocols [15], [16], [17], [18], [20], [21], [22], [23], [24], [25], [26] introduces a trusted third party as a regulator who can reveal the identity of a malicious anonymous user in the system. Xue et al. [16] introduced a central authority to generate secret keys for legitimate users accessing the cloud public data store, which can be traced to malicious authenticators when malicious authentication occurs in the system. Cui et al. [18] proposed a conditional anonymous authentication protocol for vehicular networks in multi-cloud environment, where vehicles register with a Trusted Authority (TA) for proxy access to cloud services. However, the single TA results in excessive centralization of power, which poses a risk to user privacy in the event of TA compromise or malfeasance. In order to decentralize regulatory power, Li et al. [25] proposed a privacy authentication scheme for abuse-resistant tracking, where the secret keys for regulation were generated and stored in a distributed manner, avoiding the problem of a single regulator revealing the true identity of a vehicle.

In addition, group signature technology [35] is used to achieve conditional anonymous authentication in [20], [36], [17], [21], [22], [23]. Concretely, users are hided in a group, and they can anonymously sign messages on behalf of the whole group. Once malicious behavior occurs, the group manager can open the signature to determine the identity of the signer for regulation. Although these conditional anonymous authentication protocols can regulate user's behavior, they do not provide adaptation for the membership services of the Internet. Because only external adversarys are prevented from identifying users by sniffing data streams, service providers can still link users' identities with their membership credentials.

C. Membership anonymous authentication protocol

To mitigate the privacy risks posed by service providers to membership users, several researchers have put forward protocols [27], [28], [29], [30], [34], [37], [38] for membership anonymous authentication.

k-times anonymous authentication protocols [27], [28], [29] allow users to access pay-per-use membership services anonymously up to k times, while being able to track dishonest users who exceed the k access limit. In 2022, Huang et al. [30] applied the concept of k-time anonymous authentication to pay-as-you-go cloud computing. Concretely, the group manager and the cloud provider maintain an identity list and an application log list, respectively, and the user can request an anonymous identity from the group manager before accessing the service. Then, the user can purchase a k-times anonymous credential from the cloud provider with the anonymous identity, and the valid count of credential is recorded by the provider in the application log for subsequent access. Unfortunately, these protocols [27], [28], [30], [29] do not support time span membership services. In addition, during each authentication, the service provider can link the user by the validity count of the user's membership credential.

Self-blind credentials was proposed by Verheul [31] to address the issue of service providers linking users by membership credentials. With self-blind credentials, users can update their old valid credential to a new equivalent and unlinkable credential locally. In 2008, Kiyomoto et al. [34] utilized self-blind certificates to construct an identity authentication protocol where users can modify their own self-blind certificates for each access, avoiding identity recognition and linkage by service providers. In 2014, Hanser et al. [39] proposed Structure-Preserving Signatures on Equivalence Classes (SPS-EQ), a signature scheme with self-blinding properties, which allows users to transform the acquired signature into an indistinguishable representation on the same equivalence class. Similarly, Backes et al. [33] proposed Signatures with Flexible Public Key (SFPK), which imparted self-blindness to the key pair, and constructed a self-blind credential in combination with SPS-EQ. Specifically, due to the characteristics of SFPK and SPS-EQ, this self-blind credential binds the user's identity to the credential, and SFPK provides an alternative trapdoor key generation scheme for the system, which allows the association of two SFPK public keys via a trapdoor, thus breaking its unlinkability. Nevertheless, empowering only a single regulator to hold the trapdoor and reveal the identity of the user also comes with a single point of failure and regulatory corruption.

Recently, an anonymous authentication protocol suitable for time span membership services was proposed by Xu et al. [37]. To access the Private Key Generator (PKG), each device (DE) in the protocol must first request a time-sensitive membership credential from the PKG. And, the device is able to access the PKG anonymously during the validity period of the credential. However, the anonymous credential is invariant at each access in scheme [37], and the service provider can link the user by tracking the credential. Furthermore, Xu et al. [37] does not introduce a regulatory mechanism, which leads to the misuse of anonymous credential.

III. PRELIMINARIES

In this section, we introduce the cryptographic primitives used in this paper, including Structure-Preserving Signatures on Equivalence Classes (SPS-EQ), Signatures with Flexible Public Key (SFPK) and threshold secret sharing algorithm.

A. Structure-Preserving Signatures on Equivalence Classes (SPS-EQ)

SPS-EQ was proposed by Hanser et al. [39] in 2014. In an equivalence relation R, the signature was considered as a representative on an equivalence class which can be transformed into multiple unlinkable representatives on the same equivalence class. In other words, the signature of a message can be changed without re-signing it, and the newly generated signature is indistinguishable from the previous one.

The SPS-EQ consists of the following five algorithms:

$\boldsymbol{BGGen(\chi)}$: Input a security parameter χ and return a bilinear group $BG := \{p, G_1, G_2, G_T, e, P, P^{'}\}$, where the

common group order p of the groups G_1, G_2 and G_T is a prime of bitlength , e is a pairing and P as well as $P^{'}$ are generators of G_1 and G_2, respectively.

$\boldsymbol{KeyGen_{SPS}(BG, l)}$: Given the bilinear group BG and vector length $l > 1$, select $x \xleftarrow{R} Z_p^*$ and $(x_i)_{i=1}^l \xleftarrow{R} (Z_p^*)^l$, set the private key as $SK_{SPS} \longleftarrow (x, (x_i)_{i=1}^l)$ and calculate the public key $PK_{SPS} \longleftarrow (xP^{'}, (x_i x P^{'})_{i=1}^l) = (X^{'}, (X_i^{'})_{i=1}^l)$, and finally output the secret key pair (PK_{SPS}, SK_{SPS}).

$\boldsymbol{Sign_{SPS}(M, SK_{SPS})}$: Input $M = (M_i)_{i=1}^l \in (G_1^*)^l$ which is representative of the equivalence class $[M]_R$, the private key $SK_{SPS} = (x, (x_i)_{i=1}^l)$ and select $y \xleftarrow{R} Z_p^*$ to compute:

$$Z_1 \leftarrow x \sum_{i=1}^{l} x_i M_i \tag{1}$$

$$Z_2 \leftarrow y \sum_{i=1}^{l} x_i M_i \tag{2}$$

$$(Y, Y^{'}) \leftarrow y \cdot (P, P^{'}) \tag{3}$$

Then, output $\sigma = (Z_1, Z_2, Y, Y^{'})$ as the signature of the equivalence class $[M]_R$.

$\boldsymbol{ChgRep(M, \sigma, \rho, PK_{SPS})}$: Input a representative $M = (M_i)_{i=1}^l \in (G_1^*)^l$ of the equivalence class $[M]_R$, a signature $\sigma = (Z_1, Z_2, Y, Y^{'})$, $\rho \in Z_p^*$, a SPS-EQ public key PK_{SPS}. Select $y^{'} \xleftarrow{R} Z_p^*$ to compute $\sigma^{'} \leftarrow (\rho Z_1, y^{'} \rho Z_2, y^{'} Y, y^{'} Y^{'})$ and $M^{'} \leftarrow \rho \cdot (M_i)_{i=1}^l$, then return the new representative and the new representative's corresponding signature $(\sigma^{'}, M^{'})$.

$\boldsymbol{Verify_{SPS}(M, \sigma, PK_{SPS})}$: Input a representative $M = (M_i)_{i=1}^l \in (G_1^*)^l$ of the equivalence class $[M]_R$ and the corresponding signature $\sigma = (Z_1, Z_2, Y, Y^{'})$, and input the public key $PK_{SPS} = (X^{'}, (X_i^{'})_{i=1}^l)$ to check whether both of the following holds:

$$\prod_{i=1}^{l} e(M_i, X_i^{'}) \stackrel{?}{=} e(Z_1, P^{'}) \tag{4}$$

$$e(Z_1, Y^{'}) \stackrel{?}{=} e(Z_2, X^{'}) \tag{5}$$

$$e(P, Y^{'}) \stackrel{?}{=} e(Y, P^{'}) \tag{6}$$

The signature will be accepted if the output is true, otherwise it is rejected.

B. Signatures with Flexible Public Key (SFPK)

Inspired by SPS-EQ, Backes et al. [33] proposed Signatures with Flexible Public Key (SFPK) in 2018, which extended the key pairs to equivalence classes. In scheme [33], signers are able to convert the signed key pairs into different representatives of the same equivalence class, and cannot distinguish whether two secret keys belong to the same equivalence class without a trapdoor τ.

A more efficient SFPK algorithm was designed by Hanzlik et al. [40], which sacrifices a small amount of security but brings efficiency gains compared to scheme of Backes et al., which consists of the following eight algorithms:

$\boldsymbol{CRSGen(1^{\lambda})}$: Input a security parameter λ, generate a description $BG := \{p, G_1, G_2, G_T, e, g_1, g_2\}$ of a bilinear group by $BG \leftarrow BGGen(\lambda)$, choose a $y \xleftarrow{R} Z_p^*$ and calculate $Y_1 = g_1^y$ and $Y_2 = g_2^y$, set $\rho = (BG, Y_1, Y_2)$. g_1, g_2 are generators of respectively G_1 and G_2.

$\boldsymbol{KeyGen(1^{\lambda})}$: Choose $x \xleftarrow{R} Z_p^*$, set $PK_{SFPK} = (g_1, g_1^x)$ and $SK_{SFPK} = (Y_1^x, PK_{SFPK})$.

$\boldsymbol{TKeyGen(1^{\lambda})}$: Choose $x \xleftarrow{R} Z_p^*$, set $PK_{SFPK} = (g_1, g_1^x)$, $SK_{SFPK} = (Y_1^x, PK_{SFPK})$, and $\tau = (g_2^x)$.

$\boldsymbol{Sign_{SFPK}(m, SK_{SFPK})}$: Input a message $m \in \{0,1\}^{\lambda}$, choose $r \xleftarrow{R} Z_p^*$ and return the signature $\sigma = (Y_1^x H(m)^r, g_1^r, g_2^r)$.

$\boldsymbol{ChgPK_{SFPK}(PK_{SFPK}, r)}$: For a public key $PK_{SFPK} = (A, B)$, compute $PK_{SFPK}^{'} = (A^r, B^r)$, and return the new public key $PK_{SFPK}^{'}$.

$\boldsymbol{ChgSK_{SFPK}(SK_{SFPK}, r)}$: For a private key $SK_{SFPK} = (Y_1^x, PK_{SFPK})$, compute $PK_{SFPK}^{'} = (A^r, B^r)$, and return the new private key $SK_{SFPK}^{'} = ((Y_1^x)^r, PK_{SFPK}^{'})$.

$\boldsymbol{ChkRep_{SFPK}(\tau, PK_{SFPK})}$: Input a public key $PK_{SFPK} = (A, B)$. Return 1 iff $e(A, \tau) = e(B, g_2)$.

$\boldsymbol{Verify_{SFPK}(m, \sigma, PK_{SFPK})}$: Parse σ as $(Sig_{SFPK}^1, Sig_{SFPK}^2, Sig_{SFPK}^3)$, parse PK_{SPFK} as (A, B). Return 1 iff $e(Sig_{SFPK}^2, g_2) = e(g_1, Sig_{SFPK}^3)$ and $e(Sig_{SFPK}^1, g_2) = e(B, Y_2) \cdot e(H(m), Sig_{SFPK}^3)$.

C. Threshold Secret Sharing

Threshold secret sharing was proposed by Shamir [41], which shared secrets in a secure way. In this scheme, a master secret D can be split into n fractions, which are distributed to n secret conservators through a trusted manager. Any t pieces less than the threshold value cannot be reconstructed out of the master secret D. It consists of the following steps:

Firstly, pick a random polynomial $t-1$ degree $f(x) = a_0 + a_1 x + ... a_{t-1} x^{t-1} \mod p$, in which $f(0) = a_0 = D$, $a_1, a_2, ..., a_{t-1}$ are random coefficients.

Then, randomly choose n different points $(x_1, f(x_1)), ..., (x_n, f(x_n))$ in the two-dimensional plane, where $x \in Z_p \backslash \{0\}$, and evaluate:

$$D_1 = (x_1, f(x_1)), ..., D_n = (x_n, f(x_n)) \tag{7}$$

Subsequently, $(D_1, ..., D_n)$ are distributed to n secret conservators.

Finally, given a subset consisting of any t sub-secrets, it is possible to find the coefficients of $f(x)$ and eventually obtain $D = f(0)$ by interpolation.

$$f(x) = \sum_{i=1}^{t} f(x_i) \cdot \prod_{j=1, j \neq i}^{t} \frac{x - x_j}{x_i - x_j} \mod p \tag{8}$$

Furthermore, no subset with any number of elements less than t can be recovered from D.

IV. PROBLEM STATEMENT

A. System model

The system proposed in this paper consists of five entities: Identity Manager, User (IoT device, hereinafter referred to as the user), Service Provider, Blockchain, and Regulators, as shown in Fig. 1.

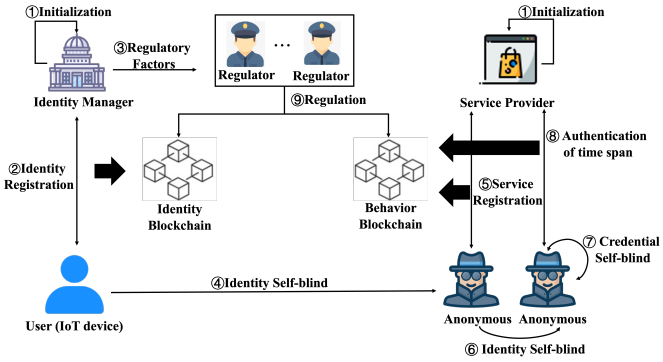


Fig. 1. AATM system model

- Identity Manager is a semi-honest entity responsible for issuing identities. When a user joins, it assigns a dynamic anonymous identity and a legal identity credential to the user, and generates regulatory factors, which are sent to multiple regulators in order to regulate malicious users. For better reliability, user's registration information will be recorded on the identity blockchain.
- User can obtain a self-blinded, anonymous identity from the identity manager. With the identity, the user can anonymously obtain a time span self-blinded membership credential from the service provider and have anonymous, unlinked access to the membership service for the duration of the credential. In addition, the user can request regulation to protect his rights.
- Service provider is an entity that provide various Internet membership services, receive requests from users, issue self-blinded membership credentials for users, or provide anonymous authentication for membership services. Similar to an identity manager, after processing a user's request, it records the request information on the behavior blockchain. When attacked by a malicious user, it can request regulation from the regulator to reveal the user's identity and pursue liability.
- Blockchain is the entity responsible for recording various actions in the system and consists of various entity nodes in the system. Identity managers and service providers act as nodes of the identity and behavior blockchain, respectively, with write access. In addition, any entity can read the data in the identity and behavior blockchain.
- Regulator is responsible for processing regulatory requests sent by users or service providers. Based on the request, the regulator looks for relevant transactions in the blockchain and restores the regulatory factor with other regulators to pursue malicious behaviors.

B. Security assumption

Security assumptions in the proposed scheme are as follows:

- Identity manager is a semi-honest entity that follows the protocol strictly, but is still curious about the information or it is likely to be controlled by an adversary.
- A user or service provider may be malicious. Specifically, a user may use anonymous indentities to launch attacks against the service provider, or purchased anonymous membership credentials may be shared with unauthorized users to the detriment of the service provider. Similarly, a service provider violate users' privacy by linking their behavior based on their credentials when offering membership services to them.
- The blockchain is a consortium chain maintained by multiple entities in the system, where only identity managers and service providers can write, and other entities can only read the records on the blockchain. The standard encryption algorithm is assumed to be secure and unbreakable.
- The regulatory body consists of credible government departments (Public Security Bureau, Audit Department, etc.) that are honest and do not collude with other adversaries.
- In order to prevent a service provider from linking user identities through the time credentials of member users, n member users are registered for membership within the same time window (e.g. one month or one year).

C. Security Model

The identity and anonymous access behaviour of member users are stored in the identity blockchain and behavioural blockchain, an adversary (other than the regulators) may try to obtain the identity of a member-user from two blockchains and attempt to link and track the member-user's identity.

Oracles. All entities within the system can access the publicly available system parameters BG and can anonymously retrieve the identity and behavior stored on the blockchain. Only the managers and service providers are permitted to write to the blockchain. The trapdoor algorithm oracle $\boldsymbol{OChkRep_{SFPK}}$ is compromised. It accepts two identity public keys and a trapdoor as input and outputs whether these two public keys belong to the same equivalence class, thereby indicating whether the two identity public keys correspond to the same user. The compromised identity update oracles $\boldsymbol{OChgPK_{SFPK}}$ obtain a user's identity key PK_i, and transform it into a new identity key $PK_i^{'}$. The compromised signing oracle $\boldsymbol{OSign_{SFPK}}$ accepts an SFPK signing private key SK_i and a message m, and returns a signature σ on the message. The registration oracle $\boldsymbol{OTKeyGen}$ is honest. It generates a user's identity key pair (PK_i, SK_i) and a trapdoor τ. The trapdoor verification algorithm $\boldsymbol{OChkRep}$ is corrupted,it accepts a trapdoor τ and the identity public key PK_i of the member user and determines whether the public key PK_i matches the trapdoor τ. Shamir threshold secret sharing algorithm is honest,it accepts sub-secrets from multiple users and outputs the original secret. The credential issuance oracle $\boldsymbol{OSign_{SPS-EQ}}$ is honest and issues legitimate identity and

membership credentials to users. The signature verification algorithms $\boldsymbol{OVerify_{SPS-EQ}}$ and $\boldsymbol{OVerify_{SFPK}}$ are honest. They return the verification result for an input SPS-EQ or SFPK signature.

The complete anonymity game and the conditional traceability game are defined as follows:

Membership Anonymous Game

Setup phase: A malicious service provider A_{anony}, attempting to obtain user identities, acquires the public parameters BG of the identity blockchain as well as the user's identity public key PK_i. Additionally, A obtains the membership credential σ_u and the legitimate identity credential σ_i for each user accessing the membership service within polynomial time. Furthermore, A_{anony} is capable of repeatedly accessing the oracles $\boldsymbol{OChgPK}$ and $\boldsymbol{OChgRep}$ within polynomial time.

Query phase: Upon receiving an authentication request from a membership user, the malicious service provider A_{anony} attempts to link the membership user's identity by utilizing their identity public key PK_i, membership credential σ_u, identity credential σ_i, and service access records on the behavior blockchain. Due to the self-blinding properties of SFPK identities and SPS-EQ credentials, the malicious service provider A_{anony} further attempts to iteratively link the membership user's identity through oracles $\boldsymbol{OChgPK}$ and $\boldsymbol{OChgRep}$.

Identity query: Malicious service provider A_{anony} obtains the member user's identity public key PK_i and inputs it into $\boldsymbol{OChgPK}$ to obtain a new public key $PK_i^{'}$.

Credential query: Malicious service provider A_{anony} obtains the member user's identity credentials σ_i and membership credentials σ_u, inputs them into $\boldsymbol{OChgRep}$, and obtains new identity credentials $\sigma_i^{'}$ and membership credentials $\sigma_u^{'}$.

Challenge phase: Malicious service provider A_{anony} is able to iteratively output the user's identity public key $PK_i^{'}$, identity credentials $\sigma_i^{'}$, and membership credentials $\sigma_u^{'}$ over and over again. If any of the following conditions occurs, the adversary A_{anony} successfully carries out the attack.

Case 1: The malicious service provider A_{anony} keeps iterating to acquire the transformed public key $PK_i^{'}$, which is found on the behavioral blockchain among the access records identified with that $PK_i^{'}$.

Case 2: The malicious service provider A_{anony} maliciously records locally the access membership credentials and identity credentials information of each member user, and queries the membership credentials $\sigma_i^{'}$ and identity credentials $\sigma_u^{'}$ by iteratively transforming them over and over again to appear in the local access records.

Definition 1 (Membership Anonymity). We say that AATM satisfies Membership anonymity without PPT adversary A_{anony} has a non-negligible advantage in the Membership anonymity game. That is, the adversary A_{anony} is

$$Adv_{A_{anony}}^{M-anony}(k) \leq negl(k)$$

Conditional Traceability Game

Setup: malicious attacker A_{trace} obtains the public parameters BG from the identity blockchain, as well as identity information stored on both the identity blockchain and the behavior blockchain. Additionally, A_{trace} is capable of accessing the oracle $\boldsymbol{OChkRep}$ within polynomial time.

Query phase: Attacker A_{trace} has the ability to execute $\boldsymbol{OChkRep}$ in polynomial time.

Tracking queries: Attacker A_{trace} picks a member user's identity public key PK_i in the identity or behavior blockchain and generates a trapdoor τ, which is fed into $\boldsymbol{OChkRep}$ to get the return result from $\boldsymbol{OChkRep}$.

Challenge phase: Attacker A_{trace} fixes the member user identity public key PK_i, iterates iteratively, keeps generating trapdoor τ by Shamir, and inputs it into $\boldsymbol{OChkRep}$ with the member user identity public key PK_i. If $\boldsymbol{OChkRep(PK_i, \tau) = 1}$, attacker A_{trace} wins the challenge.

Definition 2 (Conditional Traceability). We say that AATM satisfies conditional traceability if there is no polynomial adversary who does not have the advantage of being neglected to win the conditional anonymity game. Thus, the adversary A_{trace} is

$$Adv_{A_{trace}}^{C-trace}(k) \leq negl(k)$$

D. Security Objectives

In order to safeguard the privacy and security of users utilizing time span membership services, the AATM 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***: Only the regulator has the ability to uncover the true identity of a user by analyzing the access records writed on the blockchain.
- ***Conditional traceability***: Nobody except regulators can link a user's identity through their membership credentials.
- ***Complete protection of content privacy***: Nobody else, even the regulator, can decrypt the detailed content exchanged between the user and the corresponding service provider after each authentication.
- ***Forward/Backward Confidentiality***: The leakage of the current session key should not affect the security of the protocol's future and previous session keys.
- ***Membership privacy***: During the validity of the membership credentials, the user can access the membership services with a new identity each time. In this process, the service provider can only verify the validity of the membership and cannot track the user based on the user's identity credential each time.

V. The Proposed AATM Protocol

The details of the proposed AATM protocol is shown in Fig. 1. Firstly, the identity manager and service providers perform initialization to generate and public the system parameters, see step ①. Secondly, before accessing the service, the user needs to apply for a legal identity from identity manager. After receiving a registration request from the user, the identity manager generates a self-blindable anonymous identity for the user

TABLE I
NOTATIONS AND DESCRIPTIONS

Notation	Description
ΔT	Timeout time for timestamps
τ_i	SFPK trapdoor for user i
$E_{AES}(x,y), D_{AES}(x,y)$	Encrypt/decrypt x using symmetric secret key y
$E(x,pk), D(x,sk)$	The RSA algorithm uses the public key pk to encrypt x and the private key sk to decrypt x
$U\|\|V$	Concatenate operation
s_1, s_2	Communication secret key for symmetric encryption algorithm
$\sigma_i, \sigma_i^{'}, \sigma_i^{''}$	User i's initial identity credentials and self-blinded identity credentials
$(\sigma_T, \sigma_u, t_e), (\sigma_T, \sigma_u^{'}, t_e)$	User i's membership credentials and self-blinded membership credentials
$\sigma_{is}^{'}, \sigma_{is}^{''}$	SFPK signature of user i for identity credentials $\sigma_i^{'}$,$\sigma_i^{''}$
$(PK_i, SK_i), (PK_i^{'}, SK_i^{'}), (PK_i^{''}, SK_i^{''})$	The user's initial SFPK key pair and the SFPK key pair after self-blinding
(PK_{SPS}^e, SK_{SPS}^e)	SPS-EQ key pair of entity e
(PK_{RSA}^e, PK_{RSA}^e)	RSA key pair of entity e
$Hash()$	Secure Hash Function $Hash: \{0,1\}^* \rightarrow \{0,1\}^n$
$Shamir(s,t,n)$	Dividing the secret s into n copies with a threshold t
$Sign_{RSA}(M, SK)$	RSA signing of message M with private key SK
$Verify_{RSA}(M, \sigma, PK)$	Verify message M and signature σ using RSA public key PK, output 1 if correct otherwise output 0.

and records the user's registration information on the identity blockchain. To ensure the equity of regulation, regulatory factors are distributed to multiple regulators by the identity manager, as shown in steps ② - ③. Thirdly, once a user obtains an identity, he can self-blindly generate a new identity, which is used to purchase time span membership credentials from a service provider. After checking the legitimacy of the user's identity, the service provider issues a time span membership credential to the user and writes the user's purchase record to the behavioral blockchain, see steps ④ - ⑤. Fourthly, the user authenticates to the service provider after self-blinding identity and membership credentials. When receiving the authentication request, the service provider authenticates the legitimacy of the user's credentials and records the user's access behavior on the blockchain, see steps ⑥ - ⑧. Finally, in order to ensure the fairness of regulation, multiple regulators will ultimately restore regulatory factors and pursue malicious behavior accountable by records on the identity and behavior blockchain, see step ⑨. To illustrate our scheme more clearly, we give the notations and important terms used in Table I. Afterwards, we describe the details of the protocol, which consists of five phases: initialization, user identity registration, membership service registration, anonymous authentication for time span membership and regulation.

A. Initialization

In this phase, the identity manager and the service provider are initialized.

Firstly, the identity manager picks a security parameter χ to construct a description $BG := \{p, G_1, G_2, G_T, e, g_1, g_2\}$ of a bilinear group $BG \leftarrow BGGen(\chi)$, chooses a $y \xleftarrow{R} Z_p^*$ and compute $Y_1 = g_1^y, Y_2 = g_2^y$. Then, the description BG, Y_1, Y_2 is published to the genesis block of the identity blockchain.

Secondly, the identity manager chooses a vector of length $l > 1$ which is used with BG as input to $KeyGen_{SPS}(BG, l)$ to create the SPS-EQ key pair (PK_{SPS}^I, SK_{SPS}^I), and publishes the PK_{SPS}^I.

The service provider obtains the description BG from the identity blockchain and selects a vector length $l^{'} > 1$ to generate the SPS-EQ key pairs $(PK_{SPS}^S, SK_{SPS}^S) \leftarrow KeyGen_{SPS}(BG, l^{'})$. Then, the service provider generates the RSA key pairs (PK_{RSA}^S, SK_{RSA}^S) and publishes (PK_{SPS}^S,PK_{RSA}^S).

B. User Identity Registration

In order to access the membership service, a user must first register a self-blinding, conditionally traceable, anonymous identity with the identity manager. The user identity registration is shown in Fig. 2.

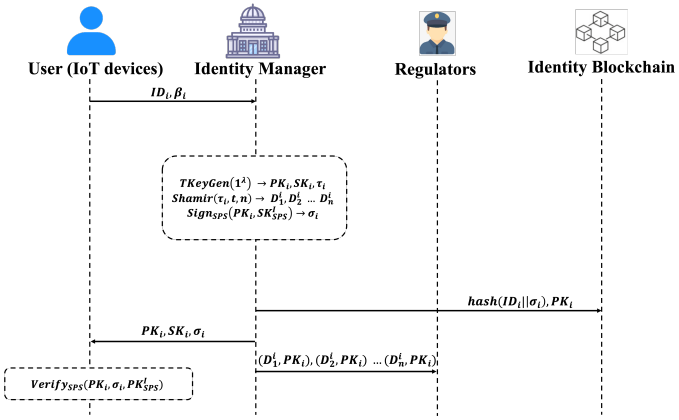


Fig. 2. User identity registration phase

STEP UR1 To ensure the authenticity of a user's identity, the user i must send his ID_i number and biometrics β_i to the identity manager for verification.

STEP UR2 The identity manager verifies whether β_i as well as ID_i is legitimate, selects a security parameter λ and a threshold value t. Then, it uses the algorithm $TKeyGen(1^\lambda)$ to generate the key pair (PK_i, SK_i) of user i and the trapdoor information τ_i which is divided into n copies (n is the number of regulators) $D_1^i, D_2^i ... D_n^i \leftarrow Shamir(\tau_i, t, n)$.

STEP UR3 The identity manager signs PK_i with the $Sign_{SPS}(PK_i, SK_{SPS}^I)$ algorithm to generate σ_i as the legitimate credentials of user i. Then, it calculates the hash of ID_i and the initial credentials σ_i, and writes $(hash(ID_i||\sigma_i), PK_i)$ to the identity blockchain. Finally, the identity manager sends (PK_i, SK_i, σ_i) to user i and the secret value (D_j^i, PK_i) of threshold τ_i is sent to each regulator j.

STEP UR4 After receiving a response from the identity manager, user i verifies the σ_i by $Verify_{SPS}(PK_i, \sigma_i, PK_{SPS}^I)$. If correct, the key pair

will be accepted and the PK_i is use to represent user's identity. In addition, each regulator stores the received threshold secret value $(D_1^i, PK_i), (D_2^i, PK_i)...(D_n^i, PK_i)$ locally.

C. Membership Service Registration

After the anonymous identity is generated, the user need to apply for a self-blinded, unlinkable membership credential from the service provider to request membership service. This process is shown in Fig. 3 and is designed as follows:

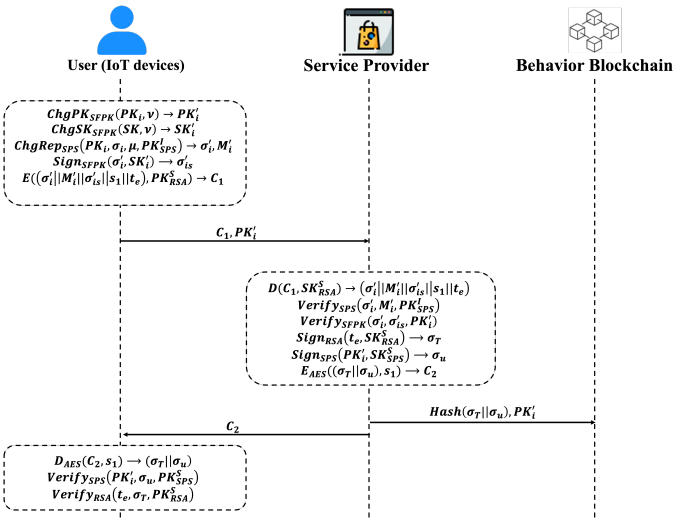


Fig. 3. Membership service registration phase

STEP MR1 The user randomly selects a $\nu \xleftarrow{R} Z_p^*$ and treats it as the input to the $ChgPK_{SFPK}(PK_i, \nu)$ and $ChgSK_{SFPK}(SK_i, \nu)$ algorithms to generate a brand new key pair $(PK_i^{'} = (PK_i)^{\nu}, SK_i^{'} = (SK_i)^{\nu})$. Then, the user randomly picks a $\mu \in Z_p^*$ to change his identity credentials $(\sigma_i^{'}, M_i^{'}) \leftarrow ChgRep_{SPS}(PK_i, \sigma_i, \mu, PK_{SPS}^I)$, and signs the changed credential $\sigma_{is}^{'} \leftarrow Sign_{SFPK}(\sigma_i^{'}, SK_i^{'})$ with his new private key $SK_i^{'}$.

STEP MR2 The user randomly selects a 128-bit AES key as the session key s_1 and he selects a credential expiration time t_e according to the himself requirements, and encrypts $C_1 \leftarrow E((M_i^{'}||\sigma_i^{'}||\sigma_{is}^{'}||s_1||t_e), PK_{RSA}^S)$ using the service provider's public key PK_{RSA}^S. Afterwards, the ciphertext C_1 and the transformed identity $PK_i^{'}$ are sent to the service provider.

STEP MR3 The service provider firstly decrypts the received ciphertext C_1 using its private key SK_{RSA}^S by $(M_i^{'}||\sigma_i^{'}||\sigma_{is}^{'}||s_1||t_e) \leftarrow D(C_1, SK_{RSA}^S)$. Then, the service provider checks whether the user's credentials are legitimate by $Verify_{SPS}(M_i^{'}, \sigma_i^{'}, PK_{SPS}^I)$ and $Verify_{SFPK}(\sigma_i^{'}, \sigma_{is}^{'}, PK_i^{'})$. If the verification is passed, its signs the t_e by $\sigma_T \leftarrow Sign_{RSA}(t_e, SK_{RSA}^S)$ to generate a time credential σ_T. Next, the service provider uses SPS-EQ private key SK_{SPS}^S to sign a self-blindable, legitimate membership credential for user $\sigma_u \leftarrow Sign_{SPS}(PK_i^{'}, SK_{SPS}^S)$. With the session key s_1, σ_T and σ_u are encrypted $C_2 \leftarrow E_{AES}((\sigma_T||\sigma_u), s_1)$ and C_2 is sent to the user. Finally, the service provider calculates $H_1 = hash(\sigma_T||\sigma_u)$ and records H_1, along with the anonymous identity $PK_i^{'}$, on the behavior blockchain.

SETP MR4 Decrypting the received C_2 to get (σ_u, σ_T) by $D_{AES}(C_2, s_1)$, the user verifies the membership credentials through $Verify_{RSA}(t_e, \sigma_T, PK_{RSA}^S)$ and $Verify_{SPS}(PK_i^{'}, \sigma_u, PK_{SPS}^S)$. If they are correct, these credentials are stored locally.

D. Anonymous Authentication for Time Span of Membership

When a user holds a membership credential $(\sigma_T, \sigma_u, t_e)$, he can self-blind the identity $PK_i^{'}$ as well as the credentials $\sigma_i^{'}$ to access the service. This stage achieves anonymous authentication of member users for the valid period and is shown in Fig. 4.

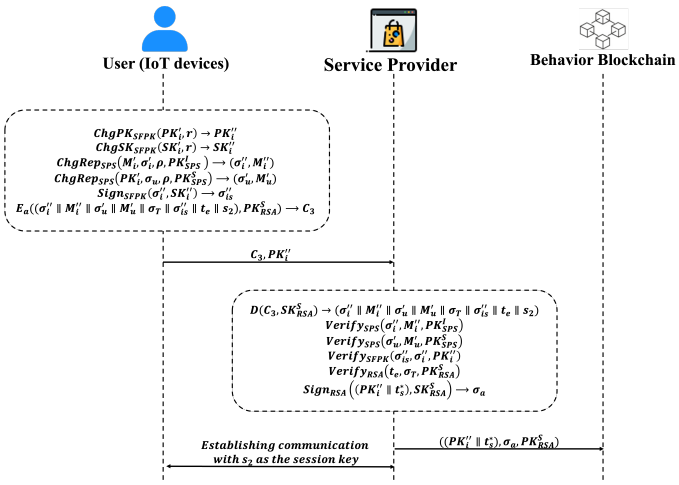


Fig. 4. Anonymous authentication for time span of membership phase

STEP AA1 The user needs to choose two random parameters $\rho \in Z_p^*$ and $r \xleftarrow{R} Z_p^*$ for self-blinded legal identity credentials and membership credentials for authentication, obtain the current time as timestamp t_s, and select a session secret key s_2 for subsequent communication.

STEP AA2 After the initial parameters are selected, the user can locally self-blind the public key $PK_i^{''} \leftarrow ChgPK_{SFPK}(PK_i^{'}, r)$, private key $SK_i^{''} \leftarrow ChgSK_{SFPK}(SK_i^{'}, r)$, the identity credentials $(\sigma_i^{''}, M_i^{''}) \leftarrow ChgRep_{SPS}(M_i^{'}, \sigma_i^{'}, \rho, PK_{SPS}^I)$, and the membership credentials $ChgRep_{SPS}(PK_i^{'}, \sigma_u, \rho, PK_{SPS}^S)$. Then the brand new private key $SK_i^{''}$ is used to sign the self-blinded identity credentials $Sign_{SFPK}(\sigma_i^{''}, SK_i^{''}) \rightarrow \sigma_{is}^{''}$. Finally, the user calculates $E((\sigma_i^{''}||M_i^{''}||\sigma_u^{'}||M_u^{'}||\sigma_T||\sigma_{is}^{''}||t_e||s_2), PK_{RSA}^S) \rightarrow C_3$ with the public key of the service provider, and send his brand new identity $PK_i^{''}$, timestamp t_s, and ciphertext C_3 to the service provider.

STEP AA3 Once receiving $C_3, PK_i^{''}, t_s$ from the user, the service provider firstly gets the current time t_s^*. If $|t_s^* - t_s| < \Delta T$, the service provider will decrypt C_3 with its own private key $(\sigma_i^{''}||M_i^{''}||\sigma_u^{'}||M_u^{'}||\sigma_T||\sigma_{is}^{''}||t_e||s_2) \leftarrow D(C_3, SK_{RSA}^S)$. Afterwards, it verifies the user's identity credentials $Verify_{SPS}(M_i^{''}, \sigma_i^{''}, PK_{SPS}^I)$, legal mem-

bership credentials $Verify_{SPS}(M_u^{'}, \sigma_u^{'}, PK_{SPS}^S)$, time attributes $Verify_{RSA}(t_e, \sigma_T, PK_{RSA}^S)$ and the user's signature $Verify_{SFPK}(\sigma_i^{''}, \sigma_{is}^{''}, PK_i^{''})$. If the verification is passed, the service provider will receive the session secret key s_2, establish a session with the user and provide the membership service.

STEP AA4 In order to prevent future disputations and to regulate malicious behavior, the service provider uses its own private key SK_{RSA}^S to sign the user's anonymous identity and access time $\sigma_a \leftarrow Sign_{RSA}((PK_i^{''}||t_s^*), SK_{RSA}^S)$ and write $((PK_i^{''}||t_s^*), \sigma_a, PK_{RSA}^S)$ to the behavior blockchain.

E. Regulation

In order to prevent malicious behavior, users and service providers can apply to the regulators for arbitration. Additionally, to promote regulatory fairness, multiple regulators are involved in each regulation for malicious service providers or malicious users.

1) Regulation for malicious service providers: Users may face a tricky situation where the certificate may be revoked by the service provider before it expires or the service provider denies the validity of the credential, where the user can apply for regulation.

STEP RS1 The user provides his initial identity PK_i, membership credential $(\sigma_T, \sigma_u, t_e)$, and the service provider's public key (PK_{SPS}^S, PK_{RSA}^S) to the regulators.

STEP RS2 The first regulator to receive the request for regulation will be reguarded as the master regulator, and the other regulators involved in this regulation will be called sub regulators. First, the master regulate obtains the transaction including $Hash(\sigma_u||\sigma_T)$ and $PK_i^{'}$ on the behavior blockchain. Then, the master regulator restores the trapdoor τ_i in conjunction with the other $k-1$ sub regulators. Based on the recovered trapdoor τ_i, the master regulator can calculate $ChkRep_{SFPK}(\tau_i, PK_i^{'})$. With the output results, the veracity of the membership credential can be determined by the master regulator. If it is true, the master regulator then uses the service provider's public key (PK_{SPS}^S, PK_{RSA}^S) to verify whether the membership credential is signed by it. And, if it is true, the master regulator holds the service provider accountable for any malicious conduct.

2) Regulation for malicious users: Because of the anonymity and untraceability, malicious users may attack the system and misuse credential. In this case, the service provider can apply for regulation of these behaviors.

STEP RU1 Based on the suspicious behavior, the service provider extracts malicious user's identity PK_m from the transaction on the behavior blockchain and sends it to the regulators for accountability.

STEP RU2 Upon receipt of a regulation request, the master regulator joints other sub-regulators to recover all trapdoors τ_i $(i = 1, 2, 3..., j)$ corresponding to public keys. By iteratively computing $ChkRep_{SFPK}(\tau_i, PK_m)$, the original public key PK_i of the malicious user can be recovered when $ChkRep_{SFPK}(\tau_i, PK_m) = 1$.

Since the identity manager writes the hash of the user's information and his initial identity to the identity blockchain at the time of registration, after obtaining the original identity PK_i of the malicious user, the master regulator can determine his true identity and pursue his responsibility by querying the transactions in the blockchain.

VI. Security Analysis

In this section, we first prove that the proposed AATM achieves all the security objectives presented in Section 4.3. Secondly, we demonstrate that the AATM scheme is provably secure.

A. Analysis of Security Objectives

1) Mutual Authentication: User encrypts his credentials and session key using the public key of the service provider by $E((\sigma_i^{''}||M_i^{''}||\sigma_u^{'}||M_u^{'}||\sigma_T||\sigma_{is}^{''}||t_e||s_2), PK_{RSA}^S)$, which only the service provider can extract by decrypting the private key SK_{RSA}^S. Similarly, the service provider authenticates the user by verifying the user's identity with $Verify_{SPS}(M_i^{''}, \sigma_i^{''}, PK_{SPS}^I)$, $Verify_{SPS}(M_u^{'}, \sigma_u^{'}, PK_{SPS}^S)$, $Verify_{RSA}(t_e, \sigma_T, PK_{RSA}^S)$ and $Verify_{SFPK}(\sigma_i^{''}, \sigma_{is}^{''}, PK_i^{''})$. Therefore, mutual authentication is achieved between legitimate user and service provider in our protocol.

2) Conditional anonymity: In our protocol, the user's identity PK is hidden in an equivalence class. The service provider is not able to recover the user's identity PK by the user's new identity $PK^{'}$ during the access. In order to audit the user's behavior, the regulator are able to recover the user's trapdoor τ and use it to reveal the identity of user $PK^{'}$ through $ChkRep_{SFPK}(\tau, PK^{'})$. Thus, our scheme achieves conditional anonymity where no one except regulators can reveal user's identity.

3) Conditional unlinkability: Our system deploys both the identity blockchain and behavior blockchain as consortium chains where only trusted identity managers and service providers have write permissions. The identity blockchain stores users' ID_i, hashed identity credentials σ_i, and their initial identity public keys PK_i while the behavior blockchain records membership purchase information and service access behaviors including hashed membership credentials (σ_T, σ_u), identity public key $PK_i^{'}, PK_i^{''}$, and access timestamps t_S^*. Through the publicly readable behavior blockchain ledger or a security vulnerability in the identity blockchain, an attacker can obtain all the users' identity public keys, but cannot recover the tuple $(PK_i, PK_i^{'}, PK_i^{''})$. Specifically, each user access employs a one-time public key, making identity linkage impossible without the trapdoor τ_i. During identity registration, the identity manager splits this trapdoor τ_i into n subsecrets using Shamir's threshold secret sharing algorithm, distributing them among multiple trusted regulatory authorities. The attacker faces negligible probability of simultaneously corrupting more than t authorities to reconstruct the trapdoor τ_i. Therefore, our scheme achieves conditional unlinkability.

4) Complete protection of content privacy: For each new session between the user and the service provider, the user generates a brand new session key s and encrypts it by the public key of service provider PK_{RSA}^S, which is then

sent to the service provider. With the new session key s, all communication content is encrypted to prevent theft. Furthermore, without the private key SK^S_{RSA}, the session key s are not available to anyone but the user. Therefore, our scheme possesses complete protection of content privacy.

5) Forward/Backward Confidentiality: In our scheme, each PK is generated and self-blinded with randomly selected numbers, thus, an adversary cannot infer them even if they are cracked. In addition, the communication between the user and the service provider is also encrypted by a randomly chosen session key s. Therefore, an adversary cannot infer the other ss with the cracked ones. In summary, our scheme achieves forward/backward confidentiality.

6) Resistance to Replay Attacks: In the anonymous authentication with time span membership phase, the timestamp t_s is introduced, when receiving the authentication message, the service provider picks the current time t^*_s for verification. If $|t^*_s - t_s| < \Delta T$ holds, it is a small probability that a replay attack occurs. Therefore, our scheme is able to resist the replay attack.

7) Membership privacy: After obtaining a identity credential σ_i and a membership credential σ_u, the user is able to locally self-blind the credential to a brand new unlinkable credential $(\sigma_i^{'}, \sigma_u^{'})$ before accessing to the services. The service provider is only able to verify the validity of the user's identity and cannot track the user's identity according to the membership credential. Thus, our scheme realizes member privacy protection.

B. Provable Security

The scheme is based on Structure-Preserving Signatures on Equivalence Classes (SPS-EQ), Signatures with Flexible Public Key (SFPK), and Shamir threshold secret sharing (denote as SHAMIR) construction. Based on the security features of each module, we demonstrate that our protocol achieves membership anonymity and conditional traceability.

Definition 3 (Class-hiding with Key Corruption). For SFPK whit relation R and adversary $\mathcal{A}$ we define the following experiment:

$$
\begin{aligned}
&\underline{CH^{\mathcal{A}}_{SFPK,R}(k)} \\
&\omega_0, \omega_1 \xleftarrow{R} coin \\
&(PK_i, SK_i) \xleftarrow{R} KeyGen(1^\lambda) \text{ for } i \in \{0,1\} \\
&b \xleftarrow{R} \{0,1\}; r \xleftarrow{R} \text{coin} \\
&(PK^{'}) \leftarrow ChgPK(PK_i, r) \\
&(SK^{'}) \leftarrow ChgSK(SK_i, r) \\
&b^* \xleftarrow{R} \mathcal{A}^{Sign_{SFPK}(SK_i^{'},\cdot)}(\omega_0, \omega_1, PK^{'}) \\
&\text{return } b = b^*
\end{aligned}
$$

A SFPK is class-hiding with key corruption if for all PPT adversaries $\mathcal{A}$ their advantage is negligible:

$$Adv^{ch}_{\mathcal{A},SFPK}(k) = |Pr[CH^{\mathcal{A}}_{SFPK,R}(k) = 1] - \frac{1}{2}| \leq negl(k)$$

Definition 4 (Class-hiding with Signatures Corruption). An SPS-EQ-$\mathcal{A}$ scheme is called class hiding, we define the following experiment:

$$
\begin{aligned}
&\underline{CH^{\mathcal{A}}_{SPS-EQ,R}(k)} \\
&BG \leftarrow BGGen_R(k), b \xleftarrow{R} 0,1 \\
&(state, PK^i_{SPS-EQ}, SK^i_{SPS-EQ}) \leftarrow \mathcal{A}(BG, l) \\
&\mathcal{O} \leftarrow \mathcal{O}^{RM}(l), \mathcal{O}^{RoR}(PK^i_{SPS-EQ}, SK^i_{SPS-EQ}, b, \cdot) \\
&b^* \leftarrow \mathcal{A}^{\mathcal{O}}(state, PK^i_{SPS-EQ}, SK^i_{SPS-EQ}) \\
&\text{return } b = b^*
\end{aligned}
$$

If for every PPT adversary $\mathcal{A}$ with oracle access to $\mathcal{O}^{RM}$ and $\mathcal{O}^{RoR}$, their advantage is negligible:

$$Adv^{ch}_{\mathcal{A},SPS-EQ}(k) = |Pr[CH^{\mathcal{A}}_{SPS-EQ,R}(k) = 1] - \frac{1}{2}| \leq negl(k)$$

Theorem 1. If the SPS-EQ and SFPK algorithms satisfy the class hiding of keys and signatures in Definitions 3 and 4, then PK_i of user i is anonymous to everyone except regulators. Therefore, this scheme satisfies the membership anonymity.

Proof 1. Defining A_{anony} as an anonymous simulation game in which an adversary (except regulators) attacks our scheme, A_{SFPK} as an opponent attacking the anonymity of SFPK signature algorithm, A_{SPS-EQ} as an opponent attacking the anonymity of SPS-EQ signature algorithm. Assuming that A_{anony} successfully attacks the anonymity of the scheme, a polynomial-time algorithm $A_\theta \in (A_{SFPK}, A_{SPS-EQ})$ is defined which has the ability to attack our SFPK and SPS-EQ signature algorithms. That is, if A_{anony} attacks the anonymity of our scheme, it means that A_θ successfully attacks the anonymity of the SFPK and SPS-EQ algorithms in the scheme with certain probability.

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

STEP 1: Initialization phase: Adversary A_{anony} obtains the system public parameters from the identity blockchain BG.

STEP 2: Inquiry phase: The adversary A_{anony} can execute algorithm A_θ in polynomial time:

Identity query: The adversary A_{anony} A_{anony} obtains the member user's identity public key PK_i and inputs it into A_{SFPK} to obtain a new public key $PK_i^{'}$.

Credential query: The adversary A_{anony} obtains the member user's identity public key PK_i and inputs it into A_{SFPK} to obtain a new public key $PK_i^{'}$.

STEP 3: Challenge phase: Eventually, the adversary A_{anony} is able to iteratively output the user's identity public key $PK_i^{'}$, identity credentials $\sigma_i^{'}$, and membership credentials $\sigma_u^{'}$ over and over again by A_{SPS-EQ}. If any of the following conditions occurs, the adversary A_{anony} successfully carries out the attack.

Case 1: The adversary A_{anony} keeps iterating to acquire the transformed public key $PK_i^{'}$, which is found on the behavioral blockchain among the access records identified with that $PK_i^{'}$.

Case 2: The adversary A_{anony} maliciously records locally the access membership credentials and identity credentials information of each member user, and queries the membership credentials $\sigma_i^{'}$ and identity credentials $\sigma_u^{'}$ by iteratively

transforming them over and over again to appear in the local access records.

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

$$\begin{aligned}
&Adv_{A_{anony}}(k)\\
&= Pr[Exp_{A_{anony}}(k) = 1]\\
&= Pr[\sigma_u \sim \sigma_u^{'}] + Pr[PK_i \sim PK_i^{'}] + Pr[\sigma_i \sim \sigma_i^{'}]\\
&= 2Pr[Exp_{A_{SFPK}}(k) = 1] + Pr[Exp_{A_{SPS-EQ}}(k) = 1]\\
&= 2Adv_{A_{SPFK}}(k) + Adv_{A_{SPS-EQ}}(k)
\end{aligned}$$

Therefore, if an adversary A_{SPFK} successfully attacks SFPK algorithm or an adversary A_{SPS-EQ} successfully attacks SPS-EQ algorithm, adversary A_{anony} wins the game. However, under the assumption that the SFPK algorithm and the SPS-EQ algorithm satisfy the class hiding of keys and signatures in Definitions 3 and 4, the probability of a successful attack by adversary A_{anony} is negligible. Thus, the scheme satisfies anonymity without regard to regulators. In other words, our scheme achieves membership anonymity.

Definition 5 (Key Recovery). A SFPK has recoverable signing keys if there exists an efficient algorithm $Recover$ such that for all security parameters $\lambda \in \mathbb{N}$, random coins r and all $(PK_i, SK_i, \tau) \xleftarrow{R} TKeyGen(1^\lambda)$ and $PK_i^{'} \xleftarrow{R} ChgPK(PK_i, r)$ we have $ChgSK(SK_i, r) = Recover(PK_i^{'}, SK_i, \tau)$.

Theorem 2. If SFPK prevents Definition 5 key recovery and the participants in the shamir algorithm are honest, then the proposed AATM protocol satisfies the conditional traceability.

Proof 2. Define A_{trace} (except regulators) to be an adversary that attacks the traceability of our scheme, A_{SFPK} as an opponent attacking the anonymity of SFPK signature algorithm and A_{Shamir} as a opponent attacking confidentiality of shamir threshold secret share algorithm. Assuming that A_{trace} successfully attacks our scheme, a polynomial-time algorithm $A_\tau \in (A_{SFPK}, A_{Shamir})$ is defined which has the ability to attack the SFPK, Shamir algorithm. Through the query of A_{trace} and the A_τ's interaction in the traceability simulated attack game, the A_τ algorithm is iterated over and over until it successfully attacks SFPK, Shamir algorithm. That is, if A_{trace} successfully attacks the traceability of the protocol, it means A_τ successfully attacks our SPFK and shamir algorithms with a certain probability.

According to the steps defined above, the interaction process between A_{trace} and A_τ is as follows:

STEP 1: Initialization phase: The adversary A_{trace} obtains the public parameters BG from the identity blockchain, as well as identity information stored on both the identity blockchain and the behavior blockchain. Additionally, A_{trace} is capable of accessing the A_τ within polynomial time.

STEP 2: Query phase: The adversary A_{trace} has the ability to execute A_τ in polynomial time.

Tracking queries: The adversary A_{trace} picks a member user's identity public key PK_i in the identity or behavior blockchain and generates a trapdoor τ, which is fed into A_{SFPK} to get the return result from A_{SFPK}.

STEP 3: Challenge phase: The adversary A_{trace} fixes the member user identity public key PK_i, iterates iteratively, keeps generating trapdoor τ by Shamir, and inputs it into A_{Shamir} with the member user identity public key PK_i. If A_SFPK of $ChkRep(PK_i, \tau) = 1$, attacker A_{trace} wins the challenge.

According to the interaction process between Algorithm A_{trace} and A_τ, the probability of an adversary successfully attacking the conditional traceability of the algorithm is as follows:

$$\begin{aligned}
&Adv_{A_{trace}}(k)\\
&= Pr[Exp_{A_{trace}}(k) = 1]\\
&= Pr[ChkRep_(PK_i, \tau) = 1]\\
&= Pr[Exp_{A_{Shamir}}(k) = 1] + Pr[Exp_{A_{SFPK}}(k) = 1]\\
&= Adv_{A_{SFPK}}(k) + Adv_{A_{Shamir}}(k)
\end{aligned}$$

Thus, if the adversary A_{SFPK} successfully attacks SFPK algorithm and the adversary A_{Shamir} successfully attacks shamir algorithm, adversary A_{trace} successfully attacks the conditional traceability of the protocol and wins the game. However, according to the security assumptions, SFPK satisfies the key class hiding in Definition 3, the security of key recovery in Definition 5 satisfies the assumptions of Backes et al. [33] and the participants of the Shamir parties are credible governmental agencies guaranteeing and the success of adversary A_{trace}'s attack is negligible. As a result, the AATM achieves conditional traceability.

VII. Performance Analysis

In this section, we compare and analyze the performance of our scheme with that of advanced authentication protocols [24], [25], [28], [30], [38] from two perspectives: security and privacy features and computation cost.

A. Security and Privacy Features Comparison

Table II represents the security characteristics of our scheme compared to the representative schemes.

In Table II, all the schemes implement mutual authentication, anonymous access and conditional traceability. And, timestamps are introduced to resist replay attacks, except for scheme [25]. Compared to the above schemes, only our AATM achieves protection of member privacy, allowing users to perform "one identity for one access" and update their credentials locally, which prevents service providers from linking users' identities through users' fixed membership credentials and authentication history.

B. Computation Cost

To evaluate the computation cost of our scheme versus advance schemes, we simulated cryptography operations running

TABLE II
A COMPARISON IN SECURITY AND PRIVACY FEATURES

	Mutual Authentication	Anonymity	Conditional Traceability	Replay Attack	Membership Privacy
[24]	✓	✓	✓	✓	×
[25]	✓	✓	✓	×	×
[28]	✓	✓	✓	✓	×
[30]	✓	✓	✓	✓	×
[38]	✓	✓	✓	✓	×
Our Scheme	✓	✓	✓	✓	✓

TABLE III
DIFFERENT CRYPTOGRAPHY TIME COSTS

Notation	**Time spent**	**Description**
T_{bp}	$3.25ms$	Time spent performing bilinear pairing
T_{bpa}	$0.006ms$	Time spent performing bilinear point addition
T_{bsm}	$3.22ms$	Time spent on scalar multiplication of bilinear pairs
T_{es}	$2.05ms$	Time spent performing scalar multiplication of elliptic curves
T_{esm}	$0.541ms$	Time spent performing small scalar multiplication of elliptic curves
T_{ea}	$0.038ms$	Time spent performing elliptic curve point addition operations
T_H	$0.71ms$	Time spent executing the mapped-to-point hash function
T_h	$0.006ms$	Time spent executing general hash functions
T_{Ge}	$3.4ms$	Time spent performing exponential operations on the group
T_{PUF}	$0.28ms$	Time spent performing physical unclonable functions
T_{RFE}	$265.01ms$	Time spent performing Reverse Fuzzy Extractor
T_{FE}	$131.58ms$	Time spent performing Fuzzy Extractor
T_{ss}	$0.057ms$	Time spent performing shamir threshold secret sharing
T_{rs}	$2.31ms$	Time spent performing RSA signature operations
T_{rv}	$0.61ms$	Time spent performing RSA signature verification operations
T_{sym}	$0.003ms$	Time cost of symmetric encryption or decryption operations
T_{sp}	$10.91ms$	Time spent performing the generation of schnnorr proof
T_{sv}	$7.51ms$	Time spent performing verification of schnorr proof
T_{re}	$0.76ms$	Time spent on RSA encryption algorithm
T_{rd}	$2.38ms$	Time spent on RSA decryption algorithm

on Ubuntu 16.04.7 LTS with dual cores and 4G RAM. The computation cost was simulated using the average time to perform 1000 relevant cryptographic operations in the pypbc 0.2, pypuf 2.2, fuzzy-extractor 0.3, pycryptodemo 3.16 and petlib 0.0.39 libraries in Python 3.5. In addition, in order to simplify the process, we ignore the less expensive operations such as XOR, while AES, RSA signature and SHA-256 algorithms are choosen to evaluate computation cost. The variables that affect the overhead in the system are taken to be the smallest within the legal range (for example, the supervisory behavior overhead comparison assumes only one user in the system). A symbolic representation of the time cost of different cryptographic operations is defined in Table III.

Our scheme includes five phases: initialization, user identity registration, membership service registration, anonymous authentication for time span of membership, and regulation. For a user, the system initialization phase and the identity registration phase often occur only once, and their comparison is not given in this section. Furthermore, in our scheme, the blockchain only records the individual behaviors within the system for storage, so its computation cost is not considered. Therefore, the computation cost is focused on the member service registration, anonymous authentication, and regulation phases under the similar setup compared to other schemes.

Based on the results of the simulation experiments for each cryptographic operation, we give a comparison of the computation cost in Table IV. In the scheme [24], the registration phase needs to perform two operations of the reverse fuzzy extractor, two fuzzy extraction operations and four exponential operations on the group with a total cost of $2T_{RFE}+2T_{FE}+4T_{Ge} \approx 806.78ms$. The authentication phase requires two fuzzy extractors, two physical unclonable functions and six hash operations for both the user and the service provider, with a total cost of $2T_{FE}+2T_{PUF}+6T_h \approx 263.756ms$. In order to regulate the malicious behavior, a total of two reverse fuzzy extraction operations and five hash operations are performed in the regulation phase and the cost is $5T_h+2T_{RFE} \approx 530.05ms$.

Li et al. [25] performs one Schnnor proof generation and verification operation, three bilinear pairings and five exponential operations on the group in registration phase, and the cost is $T_{sp}+T_{sv}+3T_{bp}+5T_{Ge} \approx 45.17ms$. In authentication phase, the user needs to perform two hash operations mapped to points and seven exponential operations on groups, costing $2T_H+7T_{Ge} \approx 25.22ms$. The service provider executes two bilinear pairings, two hashes mapped to points and six exponential operations on groups, with an overhead of $2T_{bp}+2T_H+6T_{Ge} \approx 28.32ms$. In addition, it is necessary to perform an exponential operation on the group, taking $T_{Ge} \approx 3.4ms$ to pursue the malicious user.

Liu et al. [28] requires one hash operation and two exponential operations with a time overhead of $T_h+2T_{Ge} \approx 6.81ms$,The authentication process requires one hash operation as well as six exponential operations, taking time of $T_h+6T_{Ge} \approx 20.41ms$. Finally, in order to expose malicious users, the one hash operation and eight exponential operations are needed, with an overhead of $T_h+8T_{Ge} \approx 27.21ms$.

In the scheme [30], the process of requesting credentials from the cloud provider requires the user to compute four hash-to-point, eight bilinear pairings, seventeen scalar multiplications based on bilinear groups, and thirty-eight exponential operations on groups with a computation cost of $4T_H+8T_{bp}+17T_{bsm}+38T_{Ge} \approx 212.78ms$. In the authentication process, the user needs to compute one hash-

TABLE IV
COMPUTATION COST (MS)

Schemes	Service Registration	Authentication			Traceability	Total
		User	Service Provider	Authentication Expense	Regulator	
[24]	$2T_{RFE}+2T_{FE}+4T_{Ge}$ $\approx 806.78ms$	$2T_{FE}+2T_{PUF}+6T_h \approx$ $263.756ms$	$2T_{FE}+2T_{PUF}+6T_h \approx$ $263.756ms$	$527.512ms$	$5T_h+2T_{RFE}$ $\approx 530.05ms$	$1864.342ms$
[25]	$T_{sp}+T_{sv}+3T_{bp}+5T_{Ge}$ $\approx 45.17ms$	$2T_H+7T_{Ge}$ $\approx 25.22ms$	$2T_{bp}+2T_H+6T_{Ge}$ $\approx 28.32ms$	$53.54ms$	$T_{Ge} \approx 3.4ms$	$102.11ms$
[28]	$T_h+2T_{Ge} \approx 6.8ms$	Null	$T_h+6T_{Ge} \approx 20.41ms$	$20.41ms$	$T_h+8T_{Ge} \approx 27.21ms$	$54.43ms$
[30]	$4T_H+8T_{bp}+17T_{bsm}$ $+38T_{Ge} \approx 212.78ms$	T_H+4T_{Ge} $\approx 14.31ms$	$T_H+2Tbsm+4T_{Ge}$ $\approx 20.75ms$	$35.06ms$	$T_{bsm} \approx 3.22ms$	$251.06ms$
[38]	$11T_{Ge} \approx 37.4ms$	$4T_{Ge} \approx 16.85ms$	$6T_{Ge} \approx 20.40ms$	$37.25ms$	$8T_{Ge} \approx 27.2ms$	$101.85ms$
Our Scheme	$T_{rs}+T_{rv}+T_{re}$ $+T_{rd}+2T_{sym}+3T_h$ $+6T_{Ge}+9T_{bsm}+19T_{bp}$ $\approx 116.614ms$	$T_h+T_{re}+6T_{Ge}$ $+10T_{bsm} \approx 53.366ms$	$T_h+T_{rv}+T_{rd}$ $+19T_{bp} \approx 64.746ms$	$118.112ms$	$T_{ss}+2T_{bp}$ $\approx 6.557ms$	$241.283ms$

to-point and four exponential operations on the group and the computation cost is $T_H+4T_{Ge} \approx 14.31ms$. In addition, the service provider authenticates the user by computing one hash-to-point, two scalar multiplications based on bilinear pairing and four exponential operations on the group, which costs $T_H+2Tbsm+4T_{Ge} \approx 20.75ms$. Finally, dishonest entities can be tracked by computing once a scalar multiplication based on bilinear pairings and the computation cost is $T_{bsm} \approx 3.22ms$.

In Yang et al.'s scheme [38], eleven exponential operations are required with a time overhead of $11T_{Ge} \approx 37.4ms$, the authentication process requires the user to perform four exponential operations as well as one bilinear pairwise registration, which takes $T_{bp}+4T_{Ge} \approx 16.85ms$, and the service provider needs to perform six exponential operations to verify the user's identity, which takes $6T_{Ge} \approx 20.40ms$. Finally, in order to reveal the malicious user, eight exponential operations are required with an overhead of $8T_{Ge} \approx 27.20ms$.

In the service registration phase, after self-blinding the key pair and identity credentials, a user applies for credentials from a service provider, which requires a total of one signature, one signature verification, one asymmetric cryptographic decryption operation, one symmetric cryptographic decryption operation, three hash functions, six exponential operations on group elements, nine bilinear-based scalar multiplications and nineteen bilinear pairing operations with the computation result of $T_{rs}+T_{rv}+T_{re}+T_{rd}+2T_{sym}+3T_h+6T_{Ge}+9T_{bsm}+19T_{bp} \approx 116.614ms$. During accessing the service, the user self-blinds their key pair and credentials in the same way, where one hash function, one asymmetric cryptographic encryption, six exponential operations on the group and ten scalar multiplication operations based on bilinear pairings are performed and the cost is $T_h+T_{re}+6T_{Ge}+10T_{bsm} \approx 53.366ms$. The service provider needs to verify whether the user's identity is legitimate and store the access record on the blockchain by performing one hash function, one asymmetric cryptographic algorithm decryption, one signature verification and nineteen bilinear pairing operations, which costs $T_h+T_{rv}+T_{rd}+19T_{bp} \approx 64.746ms$. Malicious users' identities can be revealed by the regulators through one shamir threshold secret recovery and two bilinear pairing operations, requiring $T_{ss}+2T_{bp} \approx 6.557ms$. In terms of regulatory overhead, our scheme is superior to schemes [24], [28], [38]. During the authentication phase, our scheme trades marginal performance degradation for enhanced privacy protection. To prevent identity linkage by service providers, users must perform multiple scalar multiplication operations to self-blind both identity credentials and membership credentials. Additionally, service providers need to execute nineteen bilinear pairing operations to validate the self-blinded credentials' authenticity. These cryptographic operations result in slightly higher authentication overhead compared to schemes [25], [28], [38].

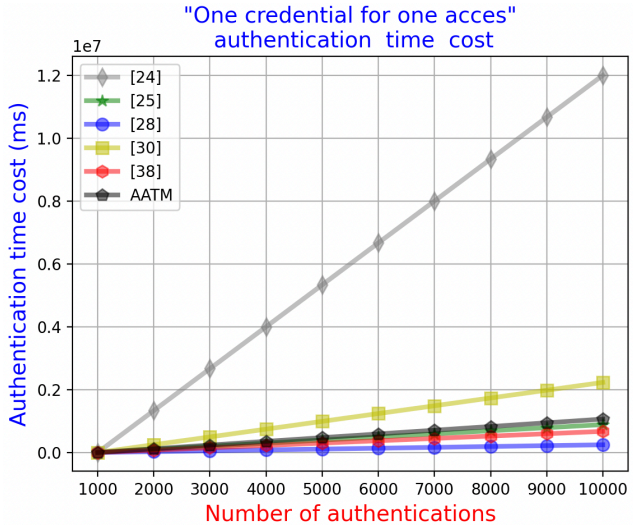


Fig. 5. "One credential for one access" authentication time cost

To prevent members from being linked by the service provider, schemes [24], [25], [28], [30], [38] must re-register with the service provider after each access to obtain a new membership credential. This process is referred to as "one credential for one access". Due to the self-blinding nature of our scheme, users can update their membership credentials locally at each authentication after registering for the

membership service. As shown in Fig. 5, with the number of authentications increasing, the time overhead of our scheme is lower than the others ([24], [30]). And Li et al. [25] has a similar time overhead as our AATM. Although Lie et al. [28] and Yang et al. [38] outperforms our protocol, they require the user to provide the old credentials to the service provider to obtain new credentials, still resulting in the user being linked by the service provider, which is avoided by the self-blinding nature of our protocol.

VIII. Conclusion

To protect users privacy on the scenarios where a service provider provides a personalized information service to a IOT device. An anonymous authentication protocol for time span of membership (AATM) with self-blindness and accountability is proposed in this paper. The protocol guarantees that users' behavioral and privacy information is not collected by service providers, and users can maintain both authenticity and anonymity in their interactions with the service providers by self-blind credentials. In addition, utilizating the SFPK trapdoor key generation algorithm's linkability, in combination with blockchain and threshold secret sharing algorithm, our AATM decentralizes regulatory authority to achieve impartial accountability for malicious users and service providers. Likewise, the abusive sharing behavior of anonymous membership credentials is effectively regulated. The analysis of security and performance indicates that the AATM has a superior capability to protect privacy and meets the requirements of practical applications.

Acknowledgments

This work was supported by the Natural Science Foundation of Zhejiang Province (Grant no. LY23F020017).

References

[1] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[2] D. Recordon and D. Reed, "Openid 2.0: a platform for user-centric identity management," in *Proceedings of the second ACM workshop on Digital identity management*, 2006, pp. 11–16.

[3] D. Hardt, "The oauth 2.0 authorization framework," Tech. Rep., 2012.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[5] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of {In-Car} wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium (USENIX Security 10)*, 2010.

[6] P. Vijayakumar, V. Chang, L. J. Deborah, B. Balusamy, and P. Shynu, "Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks," *Future generation computer systems*, vol. 78, pp. 943–955, 2018.

[7] J. Zhou, Z. Cao, Z. Qin, X. Dong, and K. Ren, "Lppa: Lightweight privacy-preserving authentication from efficient multi-key secure outsourced computation for location-based services in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 420–434, 2019.

[8] C. Lin, X. Huang, and D. He, "Ebcpa: Efficient blockchain-based conditional privacy-preserving authentication for vanets," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 1818–1832, 2022.

[9] M. A. R. Baee, L. Simpson, X. Boyen, E. Foo, and J. Pieprzyk, "Ali: Anonymous lightweight inter-vehicle broadcast authentication with encryption," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 1799–1817, 2022.

[10] F. Chen, Z. Xiao, T. Xiang, J. Fan, and H.-L. Truong, "A full lifecycle authentication scheme for large-scale smart iot applications," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2221–2237, 2022.

[11] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[12] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ecc-based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2017.

[13] J. Chen, Z. Zhan, K. He, R. Du, D. Wang, and F. Liu, "Xauth: Efficient privacy-preserving cross-domain authentication," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3301–3311, 2021.

[14] L. Bai, C. Hsu, L. Harn, J. Cui, and Z. Zhao, "A practical lightweight anonymous authentication and key establishment scheme for resource-asymmetric smart environments," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 3535–3545, 2022.

[15] Q. Lyu, Y. Qi, X. Zhang, H. Liu, Q. Wang, and N. Zheng, "Sbac: A secure blockchain-based access control framework for information-centric networking," *Journal of Network and Computer Applications*, vol. 149, p. 102444, 2020.

[16] K. Xue, Y. Xue, J. Hong, W. Li, H. Yue, D. S. Wei, and P. Hong, "Raac: Robust and auditable access control with multiple attribute authorities for public cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 953–967, 2017.

[17] Y. Li, W. Susilo, G. Yang, Y. Yu, X. Du, D. Liu, and N. Guizani, "Toward privacy and regulation in blockchain-based cryptocurrencies," *IEEE Network*, vol. 33, no. 5, pp. 111–117, 2019.

[18] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1654–1667, 2019.

[19] Q. Lyu, H. Li, Z. Deng, J. Wang, Y. Ren, N. Zheng, J. Liu, H. Liu, and K.-K. R. Choo, "A2ua: An auditable anonymous user authentication protocol based on blockchain for cloud services," *IEEE Transactions on Cloud Computing*, no. 01, pp. 1–16, 2022.

[20] Z. Wang, J. Fan, L. Cheng, H. An, H. Zheng, and J. Niu, "Supervised anonymous authentication scheme," *Journal of Software*, vol. 30, no. 6, pp. 1705–1720, 2019.

[21] L. E. Funderburg and I.-Y. Lee, "Efficient short group signatures for conditional privacy in vehicular ad hoc networks via id caching and timed revocation," *IEEE Access*, vol. 9, pp. 118 065–118 076, 2021.

[22] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for vanets," *IEEE Transactions on vehicular technology*, vol. 65, no. 3, pp. 1711–1720, 2015.

[23] K. Lim, K. M. Tuladhar, X. Wang, and W. Liu, "A scalable and secure key distribution scheme for group signature based authentication in vanet," in *2017 IEEE 8th annual ubiquitous computing, electronics and mobile communication conference (UEMCON)*. IEEE, 2017, pp. 478–483.

[24] J. Subramani, A. Maria, A. S. Rajasekaran, and F. Al-Turjman, "Lightweight privacy and confidentiality preserving anonymous authentication scheme for wbans," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3484–3491, 2021.

[25] J. Li, Y. Li, C. Cao, and K.-Y. Lam, "Conditional anonymous authentication with abuse-resistant tracing and distributed trust for internet of vehicles," *IEEE Internet of Things Journal*, 2021.

[26] C. Özbay and A. Levi, "Blacklisting based anonymous authentication scheme for sharing economy," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 828–846, 2023.

[27] I. Teranishi, J. Furukawa, and K. Sako, "K-times anonymous authentication," in *Advances in Cryptology-ASIACRYPT 2004: 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004. Proceedings 10*. Springer, 2004, pp. 308–322.

[28] Y. Liu, D. He, Q. Feng, M. Luo, and K.-K. R. Choo, "Perce: A permissioned redactable credentials scheme for a period of membership," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3132–3142, 2023.

[29] B. Lian, G. Chen, M. Ma, and J. Li, "Periodic k-times anonymous authentication with efficient revocation of violators credential," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 543–557, 2014.

[30] J. Huang, W. Susilo, F. Guo, G. Wu, Z. Zhao, and Q. Huang, "An anonymous authentication system for pay-as-you-go cloud computing," *IEEE Transactions on Dependable and Secure Computing*, 2020.

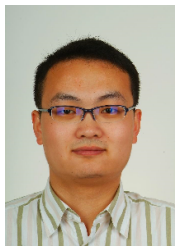
[31] E. R. Verheul, "Self-blindable credential certificates from the weil pairing," in *Advances in Cryptology ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*. Springer, 2001, pp. 533–551.
[32] S. Ringers, E. Verheul, and J.-H. Hoepman, "An efficient self-blindable attribute-based credential scheme," in *Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers 21*. Springer, 2017, pp. 3–20.
[33] M. Backes, L. Hanzlik, K. Kluczniak, and J. Schneider, "Signatures with flexible public key: Introducing equivalence classes for public keys," in *Advances in Cryptology ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part II*. Springer, 2018, pp. 405–434.
[34] S. Kiyomoto and T. Tanaka, "Anonymous attribute authentication scheme using self-blindable certificates," in *2008 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2008, pp. 215–217.
[35] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptology EUROCRYPT91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*. Springer, 1991, pp. 257–265.
[36] B. Gong, C. Cui, M. Hu, C. Guo, X. Li, and Y. Ren, "Anonymous traceability protocol based on group signature for blockchain," *Future Generation Computer Systems*, vol. 127, pp. 160–167, 2022.
[37] Z. Xu, W. Liang, K.-C. Li, J. Xu, A. Y. Zomaya, and J. Zhang, "A time-sensitive token-based anonymous authentication and dynamic group key agreement scheme for industry 5.0," *IEEE Transactions on Industrial Informatics*, 2021.
[38] Y. Yang, W. Xue, Y. Zhan, M. Huang, Y. Li, and R. H. Deng, "Anopay: Anonymous payment for vehicle parking with updatable credential," *IEEE Transactions on Dependable and Secure Computing*, 2023.
[39] C. Hanser and D. Slamanig, "Structure-preserving signatures on equivalence classes and their application to anonymous credentials," in *Advances in Cryptology ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014. Proceedings, Part I 20*. Springer, 2014, pp. 491–511.
[40] L. Hanzlik and D. Slamanig, "With a little help from my friends: constructing practical anonymous credentials," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2004–2023.
[41] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

Qiuyun Lyu received her Ph.D. degree with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China in 2021. She received the bachelors and masters degrees from Changan 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.

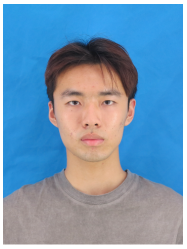
Xiwen Liang received his M.S. degree in the School of Cyberspace, Hangzhou DianZi University. His research interests include identity authentication, privacy protection and data security.

Shaopeng Cheng received his M.S. degree in the School of Cyberspace, Hangzhou DianZi University. His research interests include digital identity management and blockchain.

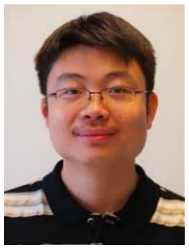
Fu Li is an internet ideological and political teacher of Hangzhou Dianzi University. He received his degree of Master of Science(M.Sc.) Software Development from University of Glasgow, UK. He got the certification of Project Management Professional(PMP) in 2019. His current research interests include anonymous authentication and big data.



Yizhi Ren received his PhD in Computer software and theory from Dalian University of Technology, China in 2011. He is currently an professor with School of Cyberspace, Hangzhou Dianzi University, China. From 2008 to 2010, he was a research fel- low 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.



Chengli Xu is currently pursuing his master's degree in the School of Cyberspace, Hangzhou DianZi University. His research interests include identity authentication, privacy protection and data security.



Weizhi Meng is a Full Professor in the School of Computing and Communications, Lancaster University, United Kingdom. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong. He was a recipient of the Hong Kong Institution of Engineers (HKIE) Outstanding Paper Award for Young Engineers/Researchers in both 2014 and 2017. He also received the IEEE ComSoc Best Young Researcher Award for Europe, Middle East, & Africa Region (EMEA) in 2020. His primary research interests are blockchain technology, cyber security and artificial intelligence in security including intrusion detection, blockchain applications, smartphone security, biometric authentication, and IoT security. He serves as associate editors / editorial board members for many reputed journals such as IEEE TDSC and IEEE TIFS, as well as general chair for various international conferences such as ACM CCS 2023 and ESORICS 2022. He is a senior member of IEEE.

Duohe Ma born in Anhui, China, in 1982. He received the B.S. and M.S. degrees from the Harbin Institute of Technology in 2007, and the Ph.D. degree in network security from the State Key Laboratory of Information Security, Institute of Information Engineering in 2015. His main research interest includes application security, moving target defense, and cloud security. Since 2016, he has been a Research Assistant with the State Key Laboratory of Information Security, Institute of Information Engineering. He is a Peer Reviewer of the Chinese Journal of Computers. Mr. Ma received the first prize of Beijing Technology in 2011, the Excellent Engineering Prize of the Institute of Information Engineering in 2014.