A Lightweight Decentralized Authentication Protocol with Anonymity and Traceability for Web3.0 Applications

Jing Lei
Xidian University, China
leijingxd@163.com

Ting Qin
Xidian University, China
15035997096@139.com

Yuepu Liu
Xidian University, China
liuyuepu233@163.com

Jie Feng
Shaanxi Key Laboratory of Blockchain and Secure Computing, China
jiefengcl@163.com

Lei Liu
Xidian University, China
leiliu@xidian.edu.cn

Qingqi Pei
Shaanxi Key Laboratory of Blockchain and Secure Computing, China
qqpei@mail.xidian.edu.cn

Mianxiong Dong
Muroran Insitute of Technology, Japan
mxdong@mmm.muroran-it.ac.jp

***Abstract*—With the development of Web 3.0, the application of distributed authentication systems has become increasingly widespread in sectors such as e-government and healthcare. However, existing decentralized solutions either suffer from single points of failure and trust assumptions or demonstrate inefficient batch verification, making it difficult to meet practical demands. This paper proposes a Decentralized Anonymous Traceable Authentication scheme DATA with batch verification through random masking. It also implements credential revocation verification using a double-layer mixed accumulator. The scheme supports users in selectively disclosing their attributes and achieving auditability without leaking sensitive information. We prove the scheme's security under standard analysis and experimentally evaluate its operating performance. The results exhibit superior performance in batch verification, effectively improving the practicality of distributed identity authentication systems.**

***Index Terms*—Distributed Authentication, Anonymous Credential, Batch Verification.**

I. Introduction

Web 3.0, as the infrastructure of metaverse, is characterized by decentralization, privacy protection, and timeliness. Its applications are increasingly extensive in sectors such as e-government and healthcare, where user identity attributes often rely on credentials from multiple issuers. Estonia's e-Residency program [1] manages citizens' digital identity through multiple Certification Authorities (CAs), with different certificates corresponding to various government services (such as taxation, social insurance, and driving license). Due to the ability for users to move across different platforms without the need for a centralized identity authority, the distributed authentication system is gaining increasing attention.

Unfortunately, the increase in participants heightens the risk of user privacy breaches. In 2023, European Cybersecurity Competence Centre (ECCC) highlighted that European health industries have experienced a considerable number of cybersecurity incidents, with 46% of cybersecurity events aimed to steal or disclose privacy data from health organizations [2]. The introduction of global privacy protection regulations such as General Data Protection Regulation (GDPR) mandates that organizations implement stringent measures when handling user privacy information. Users' needs for authentication extend beyond mere verification; they also seek to engage in various online activities without disclosing personal data. Therefore, achieving multi-credential authentication while safeguarding users' privacy presents a significant challenge in the current landscape.

Numerous efforts have been made to incorporate anonymity mechanisms into authentication systems. Other hand, if the centralized credential issuer is attacked, the attacker may generate valid credentials with arbitrary attributes based on the signed key he possesses. Therefore, multiple studies proposed distributed anonymous authentication systems. Tsang et al. [3] proposed a distributed anonymous authentication system based on group signature, while Brands [4] implements an anonymous identity system that allows for multi-party collaborative authentication. These schemes achieved the decentralization of systems, but when handling multiple credentials, they adopted a single verification approach. This leads to an exponential increase in system verification time as the scale grows and no longer meets the current timeliness required by Web 3.0.

To improve the efficiency of verification with multiple credentials, Boneh et al. [5], and Zhang et al. [6] proposed a batch-showing mechanism. Nevertheless, these schemes suffer from significant computational overhead. Besides, some research focuses on the revocation verification of users' credentials. Au et al. [7] introduced the first dynamic universal accumulator (DUA) that supports proving whether an element has been accumulated. Kate et al. [8] proposed a scheme enabling selective disclosure of credentials under constant-sized commitments. Conversely, ensuring the correct construction of polynomials and a valid disclosure introduces increased complexity when dealing with a large number of attributes.

In this paper, we propose a lightweight Decentralized Anonymous Traceable Authentication scheme DATA. The core contribution of this work lies in an efficient batch verification algorithm, which notably reduces the computational overhead during the verification process while ensuring the privacy of user attributes and issuers. The specific contributions of this paper are as follows:

- **Efficient Batch Verification:** To meet Web 3.0 requirements for timeliness, we propose a new batch-showing algorithm. By incorporating a random number κ, the batch verification expressions are transformed, allowing the more costly parts to be separated from the complicated zero-knowledge proof system.
- **Transparent regulation:** A transparent regulatory mechanism exists in our scheme to protect user attribute privacy, and only the regulator can decrypt the identity-escrowed information.
- **Revocation Checking Mechanism:** We propose a dual-layer hybrid accumulator structure, enabling users to efficiently prove that their credentials have not been revoked without revealing the details of the credentials.
- **Security Analysis and Experiments:** A rigorous security analysis is given to demonstrate that our scheme satisfies all the security properties specified in the security model. Experiments demonstrate that our batch verification scheme offers a significant efficiency improvement over the traditional single verification method.

II. Related Work

A. Distributed Anonymous Credential Issuance

David Chaum introduced the concept of Anonymous Credentials (ACs) [9], endowing credentials with unforgeability and combining them with zero-knowledge proof [10]. Camenisch and Lysyanskaya [11] effectively implemented efficient computational signatures that adhere to Chaum's principles, fulfilling non-transferability to the credentials.

In 2014, Garman et al. [12] first introduced the concept of decentralized anonymous credentials, supporting multiple private attributes and selective disclosure of credentials. Later, Sonnino et al. proposed the Coconut scheme [13]. By combining zero-knowledge proofs (ZKP) and distributed key generation (DKG), the system achieves threshold issuance and selective disclosure of credentials. However, the revocation mechanism may require regular updates and synchronization, which introduces additional complexity and calculated consumption. Jiang et al. [14] introduces a certificateless signature scheme based on pairing for distributed learning, ensuring the traceability of malicious actions through non-repudiation. The AAKA designed by Yu et al. [15] employs BBS signatures and ElGamal encryption, allowing regulators to reconstruct identities based on the identity escrow information submitted by users.

B. Anonymous Credentials with Hiding Issuer

Most existing research requires users to disclose the issuer's public key when validating credentials, which enables the verifier to infer user's private information based on their chosen issuer.

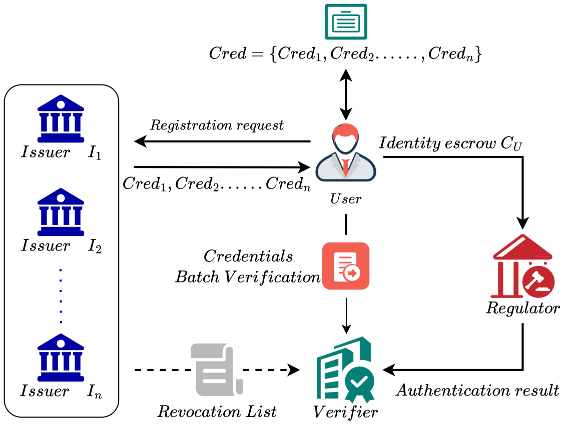


Fig. 1: System model

Conolly et al. [16] improve structure-preserving signatures on equivalence classes (SPS-EQ). This scheme randomizes the signed message and the public key used to verify the signature. Additionally, their construction requires an OR-Proof to demonstrate that the issuer's private key for the user's credential is among a set of keys accepted by verifier. This linear relationship results in high computational costs for authentication. Mir et al. [17] proposed an anonymous credential scheme with aggregate signatures and multi-issuers' identity hiding, allowing users' credentials to be re-randomized during the verification process.

C. Credential Revocation Management Based on Accumulators and Set Commitments

Au et al. [7] proposed the first Dynamic Universal Accumulator DUA, which allows for the accumulation of elements in a group $\mathbb{G}$ under the Decisional Diffie-Hellman(DDH) assumption and supports the dynamic changes of elements. This accumulator supports verifying whether a particular element has been accumulated through zero-knowledge proofs.

Thakur [18] introduced a series of batch membership and non-membership proof protocols based on bilinear accumulators, utilizing exponential proofs [19] to shift the computational overhead from verifiers to users. The core idea is to replace certain exponential operations with polynomial division and to use the Fiat-Shamir transformation to generate non-interactive proofs. Jia et al. [20] addressed the challenges of credential management in existing systems. Specifically, users can aggregate their credentials autonomously and selectively disclose their attributes to verifiers.

III. System Structure

A. System Model

The proposed scheme primarily consists of four types of participants: user $\mathcal{U}$, issuing authoritie $\mathcal{I}_i$, regulator $\mathcal{S}$, and verifier $\mathcal{V}$.The interactions among the participants are illustrated in Figure 1.

User $\mathcal{U}$ possesses multiple attributes m_i and interacts with multiple issuers $\mathcal{I}_i$ to obtain corresponding credentials. $\mathcal{U}$ encrypts their identifier ID_U using the regulator $\mathcal{S}$'s public key pk_S as an identity escrow ciphertext $C_U = Enc(pk_S, ID_U)$.

During authentication process, $\mathcal{U}$ first submits his anonymous credentials to verifier $\mathcal{V}$ for proving the credential's validity. Afterward, $\mathcal{U}$ sends C_U to $\mathcal{S}$ for authentication.

Issuer $\mathcal{I}_i$ possesses a public/private key pair for signing users' attributes and issuing the corresponding anonymous credentials. Additionally, each $\mathcal{I}_i$ manages a revocation list, regularly checking which credentials have been revoked and providing participants with revocation status information.

Regulator $\mathcal{S}$ possesses public/private key pair (pk_S, sk_S), where public key pk_S is used to escrow user's identity ID_U as C_U. If the user has malicious behaviour, $\mathcal{S}$ can use sk_S to check his true identity ID_U and expose him.

Verifier $\mathcal{V}$ **(i.e., service provider)** employs a batch verification mechanism to validate the anonymous credentials; $\mathcal{V}$ interacts with regulator $\mathcal{S}$ to validate the user's identity. If the validation passes, $\mathcal{V}$ receives the credentials from $\mathcal{U}$. $\mathcal{V}$ also queries the revocation list to ensure that the credentials have not been revoked.

B. DATA Overview

Our proposed scheme includes the following five algorithms.

System Initialization: The public parameters are agreed upon by the Issuer $\mathcal{I}_i(i \in [1, n])$, User $\mathcal{U}$, Regulator $\mathcal{S}$, and the verifier $\mathcal{V}$, with one party designated to execute the $\Pi_{CL}.SymSetup$ to generate system parameter.

Key Generation: Each issuer $\mathcal{I}_i$ generates its public/private key pair $(pk_i, sk_i) = (x_i, g^{x_i})$ for signature computation by running $\Pi_{CL}.KeyGen$.

Distributed Signature: $\Pi_{CL}.DisSign$ outputs the credential signed with different attributes m_i for User $\mathcal{U}$. $\mathcal{U}$ uses the public key pk_S to escrow his identity ID_U as C_U.

Batch Validity Verification: Verifier $\mathcal{V}$ randomly selects a vector $\Delta = (\delta_1, \ldots, \delta_n)$ where each element is a random number of ℓ_b bits, and then check whether the bilinear pairing equation $\mathbf{e}(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g) = \mathbf{e}(a, \prod_{i=1}^{n} X_i^{\delta_i}) \cdot \mathbf{e}(b, \prod_{i=1}^{n} X_i^{w_i \delta_i})$ holds. User $\mathcal{U}$ also provides a zero-knowledge proof to demonstrate that his identity ciphertext C_U is escrowed using the public key pk_S. Please note that we set $a = H_1(\phi)$, $b = H_2(\phi)$, and $w_i = H_3(ID_i||m_i||\phi)$ to generate the signature $\sigma_i = a^{x_i} b^{x_i w_i}$.

Revocation Verification: To ensure that users can successfully authenticate only when their credentials have not been revoked, we propose an efficient revocation scheme based on a dual-layer hybrid accumulator. This scheme combines polynomial accumulators [7] and operation accumulators to provide efficient revocation checking.

C. Threat Model

Issuers $\mathcal{I}_i$ and the regulator $\mathcal{S}$ lack any motivation for malicious behavior. Verifier $\mathcal{V}$ may be interested in users' private data. Our scheme accounts for the presence of a malicious user who may deliberately violate the protocol to exploit system resources or leak the privacy of other users. The following is a detailed description of the threat model:

Curious but honest verifier $\mathcal{V}$ will correctly execute the protocol but may attempt to analyze and infer the true identity or attribute information from the credentials provided by user $\mathcal{U}$.

Malicious user $\mathcal{U}$ may attempt to modify the attributes within the credentials. Users who are wanted or engaged in illegal activities may try to commit identity fraud by forging credentials to evade regulation.

Honest Issuer $\mathcal{I}_i$ **and regulator** $\mathcal{S}$ are subject to strict legal constraints and supervisory limitations, and will not attempt to compromise the privacy of users

D. Design Goals

We intend to design an efficient distributed anonymous credential system with efficient batch-showing. We implement revocation checking for anonymous credentials, effectively proving that user credentials have not been revoked during their validity period through a dual-layer hybrid accumulator structure. Specifically, we aim to achieve the following design goals:

- **Enhancing privacy Protection:** The scheme proposes a distributed anonymous credential mechanism for protecting users' attributes privacy and identity of the issuer.
- **Efficient Batch-showing Verification:** The excessive overhead resulting from using bilinear pairings can be effectively eliminated through random masking.
- **Transparent regulation:** Only Regulator $\mathcal{S}$ has the right to reveal users with malicious behavior.
- **Revocation verification**: The system implements revocation verification through a dual-layer hybrid accumulator. Users generate a "non-membership proof" to demonstrate that their credentials are not included in the revocation list.
- **Unlinkability:** If a user utilizes credentials for authentication multiple times, the records cannot be tracked or associated.
- **Unforgeability:** The credentials are unforgeable, and the attribute values within the credentials cannot be arbitrarily modified by users.

IV. Preliminaries

A. Π_{CL} signature scheme

We describe a new security signature scheme Π_{CL} proposed by Camenisch et al. [21], which is implemented based on CL signature [22].under the LRSW assumption [23].

(1) $\Pi_{CL}.SymSetup(1^\ell) \rightarrow \theta$: Upon inputting the security parameter 1^ℓ, randomly select a prime $q \in \Theta(2^\ell)$ and generate cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q,g is the generator of $\mathbb{G}_2$, and a bilinear mapping $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Randomly select $\phi \in \mathbb{Z}_q$ to control the validity period of the credentials. The message space is defined as $\mathcal{M} = \{0, 1\}^*$. Generate three hash functions: $H_1 : \phi \rightarrow \mathbb{G}_1$, $H_2 : \phi \rightarrow \mathbb{G}_1$, $H_3 : \mathcal{M} \times \phi \rightarrow \mathbb{Z}_q$. Output the system parameters $\theta = (q, g, g, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, \phi, H_1, H_2, H_3)$.

(2) $\Pi_{CL}.KeyGen(\theta) \rightarrow (sk, pk)$: Randomly select $x \in \mathbb{Z}_q$ and set $X = g^x$. Output the private/public key pair ($sk = x$,$pk = X$).

(3) $\Pi_{CL}.Sign(m, \theta, sk) \rightarrow \sigma$: For message $m \in \mathcal{M}$ to be signed, compute $a = H_1(\phi)$,$b = H_2(\phi)$, and $w = H_3(m||\phi)$, resulting in the signature $\sigma = a^x b^{xw}$ for message m.

(4) $\Pi_{CL}.Verify(m, \theta, \sigma) \rightarrow 1/0$: The verifier $\mathcal{V}$ must verify the signature σ on each message m. First, compute $a = H_1(\phi)$, $b = H_2(\phi)$, and $w = H_3(m||\phi)$. Then, check the equation $\mathbf{e}(\sigma, g) = \mathbf{e}(a, X) \cdot \mathbf{e}(b, X)^w$. If the equation holds, the verification succeeds and outputs 1; otherwise, the verification fails and outputs 0.

B. Non-interactive zero-knowledge proof

Non-interactive zero-knowledge proof(NIZK), satisfying Completeness, Soundness, and Zero-Knowledge, allows the prover to demonstrate that a statement x belongs to a language L defined by an NP relation R in zero-knowledge manner, that is, $L = \{x \mid \exists w \text{ such that } R(x, w) = 1\}$.

x is the statement, which is a candidate for membership in the language.w is the witness, which is a secret information held by prover used to demonstrate that $x \in L$; $R(x, w)$ is the NP relation, which is a relational decision function; if $R(x, w) = 1$, it indicates that w proves that x belongs to L

NIZK allows prover to demonstrate $x \in L$ to verifier without disclosing w. For the relation $\{R_\lambda\}_{\lambda \in \mathbb{N}}$, NIZK consists of three algorithms:

(1) $KeyGen(R \in R_\lambda) \rightarrow crs = (ek, vk)$: Output a public reference string that includes the evaluation and verification key pair (ek, vk).

(2) $Prove(ek, x, w) \rightarrow \pi$:Return a demonstrate that the proof $R(x, w)$ is true.

(3) $VerProof(vk, x, \pi) \rightarrow b \in \{0(\text{reject}), 1(\text{accept})\}$:Verify the proof.

C. Accumulator

Accumulator is a cryptographic data structure that allows a set of elements to be compressed into a fixed-size value. Users can prove whether a particular element is in the accumulator by providing a proof without revealing the entire set.

The proposed scheme utilizes the Dynamic Universal Accumulator (DUA) based on exponential polynomials, allowing for both dynamic management of elements and support for non-membership proofs. The core idea is to embed the polynomial representation of elements into the exponent to prevent the verifier from directly recovering the original elements. Please refer to [7] for a detailed discussion on this topic.

V. Distributed Anonymous Credential Scheme Supporting Efficient Batch Verification

We introduce two efficient identity authentication schemes under different privacy requirements, namely, Privacy protection for user credential attributes or both user credential attributes and issuer identity. Furthermore, an efficient revocation checking mechanism is implemented using a dual-layer hybrid accumulator structure.

A. Privacy Protection for User Credential Attributes

This section will detail the process of the scheme that protects user credential attribute privacy (i.e.,hiding m_i), which includes four algorithms:

1) $SymSetup(1^\ell) \rightarrow \theta$: Each participant generates the system parameter $\theta = (q, g, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, \phi, H_1, H_2, H_3)$ by running this algorithm. This includes multiplicative cyclic group $\mathbb{G}_1, \mathbb{G}_2$ of prime order $q \in \Theta(2^\ell)$ and a bilinear mapping $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, as well as a generator g of the cyclic group $\mathbb{G}_2$. The parameter $\phi \in \mathbb{Z}_q$ is used to control the validity period of credentials, along with three hash functions: $H_1 : \phi \rightarrow \mathbb{G}_1$, $H_2 : \phi \rightarrow \mathbb{G}_1$, and $H_3 : \mathcal{D} \times \mathcal{M} \times \phi \rightarrow \mathbb{Z}_q$. User's message space is $\mathcal{M} = \{0, 1\}^*$ and the identity space is $\mathcal{D} = \{0, 1\}^*$.Regulator $\mathcal{S}$ selects an ECC Elliptic Curve to generate its public/private key pair (pk_S, sk_S): the private key $sk_S \in \mathbb{Z}_q$ is a random value. The public key $pk_S = sk_S \cdot G$, where q is the order and G is the generator of elliptic curve.

2) $KeyGen(\theta) \rightarrow (sk_i, pk_i)$: Each issuer $\mathcal{I}_i$ generates its private/public key pair (sk_i, pk_i).Select random value $x_i \in \mathbb{Z}_q$ as its private key $sk_i = x_i$, and the public key is computed as $pk_i = X_i = g^{x_i}$.

3) $DisSign(ID_i, m_i, \theta, sk_i) \rightarrow \sigma_i$: Issuer $\mathcal{I}_i$ signs a attribute $\mathcal{I}_i$ and send the credential to User $\mathcal{U}$. For the attribute $m_i \in \mathcal{M}$ to be signed, compute $a = H_1(\phi)$, $b = H_2(\phi)$, and $w_i = H_3(ID_i||m_i||\phi)$, generating the signature $\sigma_i = a^{x_i} b^{x_i w_i}$. $\mathcal{U}$ selects a random value $r_U \in \mathbb{Z}_q$ and uses the public key pk_S of the regulator $\mathcal{S}$ to escrow their identity ID_U as $C_U = (C_1, C_2)$, where $C_1 = r_U \cdot G, C_2 = r_U \cdot pk_S + ID_U$.

4) $BatchVer_1(\theta, \sigma_i) \rightarrow 1/0$: The verifier $\mathcal{V}$ verifies the signature σ_i on each message m_i by jointly running Algorithm 1 with user $\mathcal{U}$.

To hide m_i, it is necessary to prove under zero-knowledge that $w_i = H_3(ID_i||m_i||\phi)$ and $\mathbf{e}(\prod_{i=1}^n \sigma_i^{\delta_i}, g) = \mathbf{e}(a, \prod_{i=1}^n X_i^{\delta_i}) \cdot \mathbf{e}(b, \prod_{i=1}^n X_i^{w_i \delta_i})$ hold. However, the presence of bilinear pairings leads to significant overhead. By adding a random number κ, the batch verification equation can be transformed, allowing the more costly parts to be verified in plaintext, thereby reducing the overhead.

Specifically, by adding the random number $\kappa = k_1 \cdot k_2$ in the exponent, we obtain the equation:

$$\mathbf{e}(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g)^\kappa = \mathbf{e}(a^\kappa, \prod_{i=1}^{n} X_i^{\delta_i}) \cdot \mathbf{e}(b^{k_1}, (\prod_{i=1}^{n} X_i^{w_i \delta_i})^{k_2})$$

Let $A = a^\kappa$, $B = b^{k_1}$, $\sigma = (\prod_{i=1}^n \sigma_i^{\delta_i})^\kappa$, $X' = (\prod_{i=1}^n X_i^{w_i \delta_i})^{k_2}$. The above equation can be transformed into:

$$\mathbf{e}(\sigma, g) = \mathbf{e}(A, \prod_{i=1}^{n} X_i^{\delta_i}) \cdot \mathbf{e}(B, X')$$

$\mathcal{U}$ proves under zero-knowledge: $w_i = H_3(ID_i||m_i||\phi)$, $X' = (\prod_{i=1}^n X_i^{w_i \delta_i})^{k_2}$, and will be $A = a^\kappa$, $B = b^{k_1}$, $\sigma = (\prod_{i=1}^n \sigma_i^{\delta_i})^\kappa$ send to $\mathcal{V}$. Upon receiving the commitment sent by $\mathcal{U}$, $\mathcal{V}$ check the bilinear pairing in plaintext space: $\mathbf{e}(\sigma, g) = \mathbf{e}(A, \prod_{i=1}^n X_i^{\delta_i}) \cdot \mathbf{e}(B, X')$.During this process, $\mathcal{V}$ may request the regulator $\mathcal{S}$ to check the legitimacy of the user's identity. $\mathcal{S}$ uses its private key sk_S to decrypt the user's

Algorithm 1 $BatchVer_1(\theta, \sigma_i) \rightarrow 1/0$

```
Require: System Parameter $\theta$,Signature$\sigma_i$.
Ensure: If the verification is successful, output 1; otherwise, output 0.
1: Calculate $a = H_1(\phi)$,$b = H_2(\phi)$;
2: Select a vector $\Delta = (\delta_1, \ldots, \delta_n)$, where each element is a random number of $\ell_b$ bits.
   Steps 3 and 4 are completed in the form of zero-knowledge proof:
3: Randomly select $k_1, k_2 \in \mathbb{Z}_q$, and set$\kappa = k_1 \cdot k_2$;
4: Prove : $w_i = H_3(ID_i||m_i||\phi)$,$X' = \left(\prod_{i=1}^{n} X_i^{w_i \delta_i}\right)^{k_2}$ hold.
5: Let $A = a^{\kappa}$,$B = b^{k_1}$,$\sigma = (\prod_{i=1}^{n} \sigma_i^{\delta_i})^{\kappa}$.
6: Under plaintext, check whether the bilinear pairing equation$\mathbf{e}(\sigma, g) = \mathbf{e}(A, \prod_{i=1}^{n} X_i^{\delta_i}) \cdot \mathbf{e}(B, X')$ holds. If it does, the verification is successful, and output 1; otherwise, the verification fails, and output 0.
```

identity ciphertext C_U: it computes $C_1 - sk_S \cdot C_1$ to obtain the corresponding plaintext identity for verification, and returns the verification result to $\mathcal{V}$.

B. Privacy Protection for User Credential Attributes and Issuer Identity

This section will propose a batch verification scheme that can simultaneously protect user credential attributes privacy and issuer identity. The protocol is detailed as follows.

1) $BatchVer_2(\theta, \sigma_i) \rightarrow 1/0$: To hide X_i and m_i, it is necessary to prove under zero-knowledge that $w_i = H_3(ID_i||m_i||\phi)$, $X_i = g^{x_i}$ and $\mathbf{e}(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g) = \mathbf{e}(a, \prod_{i=1}^{n} X_i^{\delta_i}) \cdot \mathbf{e}(b, \prod_{i=1}^{n} X_i^{w_i \delta_i})$ holds. As mentioned above, the presence of bilinear pairings leads to significant overhead,and the batch verification equation can be transformed to allow the more costly parts to be verified in plaintext by adding a random number κ.

Specifically, by adding the random number $\kappa = k_1 \cdot k_2$ in the exponent, we obtain the equation:

$$\mathbf{e}(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g)^{\kappa} = \mathbf{e}(a^{k_1}, (\prod_{i=1}^{n} X_i^{\delta_i})^{k_2}) \cdot \mathbf{e}(b^{k_1}, (\prod_{i=1}^{n} X_i^{w_i \delta_i})^{k_2})$$

Let $A = a^{k_1}$, $B = b^{k_1}$, $\sigma = (\prod_{i=1}^{n} \sigma_i^{\delta_i})^{\kappa}$, $X' = \left(\prod_{i=1}^{n} X_i^{w_i \delta_i}\right)^{k_2}$, $X'' = \left(\prod_{i=1}^{n} X_i^{\delta_i}\right)^{k_2}$.The above equation can be transformed into:

$$\mathbf{e}(\sigma, g) = \mathbf{e}(A, X'') \cdot \mathbf{e}(B, X')$$

$\mathcal{U}$ proves under zero-knowledge:$w_i = H_3(ID_i||m_i||\phi)$,$X' = \left(\prod_{i=1}^{n} X_i^{w_i \delta_i}\right)^{k_2}$,$X'' = \left(\prod_{i=1}^{n} X_i^{\delta_i}\right)^{k_2}$, and will be $A = a^{k_1}$,$B = b^{k_1}$,$\sigma = (\prod_{i=1}^{n} \sigma_i^{\delta_i})^{\kappa}$ send to $\mathcal{V}$. Upon receiving the commitment sent by $\mathcal{U}$, $\mathcal{V}$ check the bilinear pairing in plaintext space:$\mathbf{e}(\sigma, g) = \mathbf{e}(A, X'') \cdot \mathbf{e}(B, X')$.

C. Revocation Verification with Dual-Layer Hybrid Accumulator

This scheme operates in a cyclic group $\mathbb{E}$ with discrete logarithm security and employs a generator g to construct the revocation verification mechanism.

1) Lower Layer:Polynomial-Based Revocation List Accumulator: Each credential issuer $\mathcal{I}_i$ maintains a revocation list $RL_i = \{ID_1, ID_2, \ldots, ID_m\}$, based on a revocation polynomial accumulator. And users can verify their absence from the revocation list of the issuer.

2) Upper Layer: Conjunction Operation-Based Revocation List Accumulator: In a distributed anonymous credential system, the user may hold multiple credentials issued by different authorities $\{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_n\}$. Therefore, proving that the user has not been revoked by a single issuer is insufficient; it must further be ensured that the user has not been revoked by any credential issuer. To this end, we introduce a second-layer "AND" operation accumulator to aggregate revocation information from multiple credential issuers.

D. Security Analysis

The security of our proposed scheme is based on the security guarantees of Π_{CL} signatures, zero-knowledge proofs, random masking, and cryptographic obfuscated accumulators. We analyze security under the Universal Composability (UC) model [30], which ensures that the protocol we designed can maintain its security even when it is combined with other protocols.

Theorem 1 (Security of Π_{CL} Signature)**.** *Under the assumption of LRSW, Π_{CL} signature is unforgeable in the random oracle model [21].*

Theorem 2 (Security of Accumulators)**.** *If the underlying Universal Accumulator (UA) is secure, the Dynamic Universal Accumulator (DUA) is also secure [7].*

Theorem 3. *The zero-knowledge proof protocol in this scheme is a secure proof protocol that satisfies completeness, soundness and zero-knowledge.*

Proof. **Completeness.** First, we demonstrate that when

$$\text{Verify}(X_1, m_1, \sigma_1) = \cdots = \text{Verify}(X_n, m_n, \sigma_n) = 1,$$

it implies that:

$$\text{BatchVer}((X_1, m_1, \sigma_1), \ldots, (X_n, m_n, \sigma_n)) = 1.$$

This can be derived from the verification equation of the Π_{CL} signature scheme, which is given by

$$(\prod_{i=1}^{n} \mathbf{e}(\sigma_i, g)^{\delta_i})^{\kappa} = (\prod_{i=1}^{n} (\mathbf{e}(a, X_i) \cdot \mathbf{e}(b, X_i)^{w_i})^{\delta_i})^{\kappa}$$

$$= (\prod_{i=1}^{n} \mathbf{e}(a, X_i)^{\delta_i})^{\kappa} \cdot (\prod_{i=1}^{n} \mathbf{e}(b, X_i)^{w_i \delta_i})^{k_1 k_2}$$

TABLE I: Functional Comparison

Schemes	Distributed Issurance	Multi-attribute Protection	Issuer Hidding	Transparent Regulation	Revocation Verification	Batch Verification
[20]	✓	✓	✓	×	✓	✓
[24]	✓	✓	✓	×	×	✓
[25]	✓	×	×	×	×	✓
[26]	✓	✓	×	×	×	×
[27]	×	✓	×	×	×	×
[28]	✓	✓	×	✓	✓	×
[29]	×	✓	×	✓	✓	×
Ours	✓	✓	✓	✓	✓	✓

$$\Leftrightarrow \mathbf{e}(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g)^{\kappa} = \mathbf{e}(a^{\kappa}, \prod_{i=1}^{n} X_i^{\delta_i}) \cdot \mathbf{e}(b^{k_1}, (\prod_{i=1}^{n} X_i^{w_i \delta_i})^{k_2}).$$

Soundness. For any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the success probability under the security parameter λ is given by:

$$\Pr\left[\mathsf{Verify}(w_i^*, \pi^*) = 1 \mid w_i^* \neq H_3(ID_i \| m_i \| \phi)\right] \leq \mathsf{negl}(\lambda)$$

where $\mathsf{negl}(\lambda)$ is a negligible function, indicating that the system is secure. If $\mathcal{A}$ can successfully forge a valid pair (w_i^*, π^*) using random challenge values c_1 and c_2:

$$\pi_1 = \mathsf{Prove}(w_i, r_1, c_1), \quad \pi_2 = \mathsf{Prove}(w_i, r_2, c_2)$$

where r_1 and r_2 are distinct random values chosen by the adversary (commonly used in the Fiat-Shamir transformation), and the verifier accepts both proofs, then the difference can be calculated as follows:

$$r_1 - r_2 = (\alpha + c_1 \cdot x) - (\alpha + c_2 \cdot x) = (c_1 - c_2) \cdot x$$

Since $c_1 \neq c_2$, it follows that:

$$x = \frac{r_1 - r_2}{c_1 - c_2}$$

This implies that $\mathcal{A}$ can derive x, thereby breaking the collision resistance of the hash function $H_3(\cdot)$. This contradicts the collision resistance assumption of the hash function, hence $\mathcal{A}$ cannot successfully forge w_i^* and π^*.

Zero-knowledge. We construct a simulator $\mathcal{S}$ that can generate computationally indistinguishable proofs without any additional information.

In the real proof, w_i is computed by the hash function: $w_i = H_3(ID_i \parallel m_i \parallel \phi)$. Due to the unidirectionality of H_3, the verifier cannot deduce ID_i, m_i, and ϕ from w_i.

In the simulated proof, $\mathcal{S}$ relies solely on public information and randomly generates $w_i^* \leftarrow \{0,1\}^{\lambda}$.

Since H_3 outputs uniformly distributed pseudorandom values under the random oracle model, the real proof w_i is computationally indistinguishable from the simulated proof w_i^*:

$$\begin{aligned} &\{(\text{view}^{\pi}(ID_i, m_i, \phi), w_i)\}_{ID_i, m_i, \phi} \\ &\overset{c}{\equiv} \{(S(ID_i, \phi), w_i^*)\}_{ID_i, m_i, \phi} \end{aligned}$$

□

Theorem 4. *In batch verification process, the introduction of random values does not reveal the private attributes m_i. Specifically, for any computationally bounded adversary $\mathcal{A}$, it is infeasible to effectively distinguish the private attributes from the observed information.*

Proof. Let the message in batch verification be defined as:

$$X' = \left(\prod_{i=1}^{n} X_i^{w_i \delta_i}\right)^{k_2},$$

where w_i encompasses the private attributes m_i. To ensure privacy protection, we introduce a random value R and define $\tilde{X'} = R^{k_2}$. In a scenario where the adversary can only access X' and $\tilde{X'}$, the randomness of R, which is independent of the private attributes m_i, implies that the adversary cannot computationally distinguish between X' and $\tilde{X'}$:

$$X' \overset{c}{\equiv} \tilde{X'}.$$

Therefore, sending X' to the verifier does not leak private attributes m_i. Thus, the introduction of random values in batch verification process effectively enhances the security of system and prevents information leakage. □

Theorem 5. *For a multi-party protocol $\mathcal{P}$ and an ideal function $\mathcal{F}$, if there exists a simulator $\mathcal{S}$ that can simulate the behavior of the protocol $\mathcal{P}$ in the ideal world, so that no matter whether the protocol is executed in the real world or the ideal world, any polynomial-time adversary cannot distinguish between the two worlds, then the protocol $\mathcal{P}$ is UC secure [30].*

VI. Experiments

A. Functionality analysis and comparison

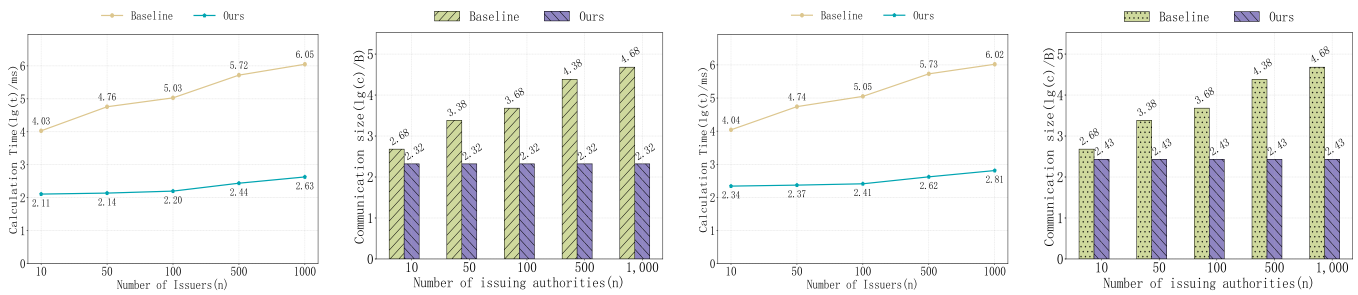
In this subsection, we compare the proposed program with related works. Table I shows the comparison of DATA with several related works in terms of functionality.

Distributed Issurance is a fundamental attribute for preventing single points of failure in a system, providing reliable guarantees for system stability. Among the related works, [27] and [29] do not support distributed issuance, while other works and DATA support this attribute.

Multi-attribute Protection The dynamic threshold issuance scheme DTACB [25] does not support multi-attribute protection. DATA and other schemes do support this functionality.

Issuer Hidding can prevent honest-but-curious verifiers from associating the corresponding user by analyzing the issuer's identity contained in credential. The schemes in [20] and [24], and DATA support hiding the issuer's identity during verification process.

Transparent Regulation ensures that users with malicious behavior can be traced by regulator. The schemes in [28], [29],



(a) Computation Times of Scheme 1 (b) Credential Sizes of Scheme 1 (c) Computation Times of Scheme 2 (d) Credential Sizes of Scheme 2

Fig. 2: Computational Time and Communication Overhead

TABLE II: Computation Time and Credential Sizes

Key Lengths (*bit*)	128	192	256
Comp.Time (μs)	284.14	288.99	326.59
Comm.Size (B)	120	120	120

and DATA support user identity accountability, while other schemes do not offer this functionality.

Revocation Verification ensures that a user can only be authenticated if their credential has not been revoked. Unfortunately, the schemes in [24], [25], [26], and [27] do not support this property.

Batch Verification addresses the issue of low authentication efficiency in distributed systems when dealing with multiple credentials. The DATA scheme proposes a batch-processing mechanism that supports selective disclosure of multiple attributes. However, the approaches in [26], [27], [28], and [29] do not support such a feature.

The proposed scheme ensures attribute privacy protection and supports transparent regulation. Additionally, an efficient batch verification mechanism is designed to enhance the computational efficiency of credential verification. Furthermore, it incorporates a credential revocation mechanism that effectively prevents the continued use of invalid credentials.

B. Experimental Analysis

Our experimental platform is constructed on a standard PC (Intel(R) Core(TM) i5-13600KF 3.50 GHz 14-core CPU and 32GB of RAM) running the 64-bit Ubuntu 20.04 operating system. Gnark offers a high-level API for designing zero-knowledge proof circuits and supports mainstream proof systems such as Groth16 and PlonK. This experiment primarily utilizes the BN254 hash function and the Groth16 proving system.

1) Credential Issurance: Table II demonstrates the generation time and size of credentials under varying key lengths.In the transition from a key length of 128 bits to 256 bits, the credential generation time increased from 284.14 µs to 326.59 µs, representing an approximate increase of 15%. Furthermore, the credential size remains constant at 120 bytes, indicating that increases in key length do not affect storage size.

2) Credential Verification: Fig 2 compares the computational and communication costs of two Batch Verification schemes (supporting attribute privacy or both attribute and issuer identity privacy) with Single Verification schemes across different numbers of issuers.Due to the significant discrepancy in experimental data between individual verification and batch verification, we applied the logarithm base 10 to the experimental results for analysis.

- Fig 2(a) illustrates the computation time of Scheme 1 (supporting attribute privacy) under different verification methods as the number n of issuers varies. The computational time for single verification increases almost linearly with n. When $n = 10$, the computation time for single verification is approximately 10,826.86ms. When $n = 100$ and $n = 1000$, the computation times increase to 107,326.31 ms and 1,119,719.17 ms, respectively. In contrast, batch verification can be completed in under 0.5 seconds for $n = 1000$, while single verification takes over 19 minutes. Batch verification demonstrates a significant computational efficiency advantage in large-scale verification scenarios.
- As shown in Fig 2(b), the communication overhead of Scheme 1 increases linearly with n for single verification, specifically 480 B (for $n = 10$) $\rightarrow$ 2400 B (for $n = 50$)$\rightarrow$ 4800 B (for $n = 100$) $\rightarrow$ 24000 B (for $n = 500$)$\rightarrow$ 48000 B (for $n = 1000$). Batch verification maintains a stable overhead of approximately 208 B, independent of the number n of issuers.
- Fig 2(c) compares the computation times of Scheme 2 (supporting both attribute and issuer identity privacy) for single verification versus batch verification under varying number n of issuers. The experimental data indicates that the computation time for single credential verification grows exponentially with the increase in the number of issuing authorities n. When $n = 10$, the computation time is approximately 10,991.94 ms, whereas at $n = 1000$, the time sharply escalates to 1,058,662.43 ms. In contrast, the computation time for batch verification is significantly reduced. When $n = 10$, the computation time is only 219.75 ms. Even in the case of $n = 1000$, the computation time remains only 642.43 ms.
- In Fig 2(d), the overhead for single verification increases linearly with n. When $n = 10$, the communication overhead is 480 B, while it grows to 48,000 B when

TABLE III: Comparison between Scheme 1 and Scheme 2 when $n = 100$.

$n = 100$	Scheme 1		Scheme 2	
	Single Verification	Batch Verification	Single Verification	Batch Verification
Comp.Time (ms)	107326.32	157.74	111239.67	255.97
Comm.Size (B)	4800	208	4800	272

$n = 1000$. Conversely, the communication overhead for batch verification remains constant at 272 B, independent of changes in n.

- Table III presents a comparison of computation time and communication overhead between Scheme 1 and Scheme 2 under single verification and batch verification (for $n = 100$). The computation time for both schemes under batch verification differs by less than 100 ms, while the communication overhead is less than 70 B.

The experimental results provide compelling evidence that batch verification can effectively reduce computation time while constraining communication overhead to a fixed value. This allows the system to achieve high efficiency and scalability while ensuring privacy protection.

VII. Conclusion

In this work, we introduce the DATA scheme to satisfy the authentication needs of Web3.0 applications while protecting user privacy. This scheme allows users to selectively disclose attributes, enabling the more costly verification to be conducted in plaintext by adding random numbers to improve the efficiency of batch verification, and the experimental results demonstrate its significant application potential. DATA achieves revocation verification through a dual-layer accumulator, requiring users to prove that their credentials are not on the revocation list. While this approach ensures secure and privacy-preserving revocation checks, it currently does not support high-frequency real-time updates to the revocation list. Overcoming this limitation by designing efficient dynamic update protocols will be the focus of our subsequent research.

References

[1] e Residency. (2023) e-residency 2.0 white paper. Republic of Estonia. [Online]. Available: https://s3.eu-central-1.amazonaws.com/ereswhitepaper/e-Residency+2.0+white+paper+English.pdf

[2] ECCC, "Checking-up on health: Ransomware accounts for 54 percent of cybersecurity threats," Website, 2023, https://ec.europa.eu/newsroom/ECCC/items/795638/en.

[3] P. K. Tsang, A. Kapadia, S. W. Smith, and P. D. McDaniel, "Short group signatures using strong diffie-hellman assumption," in *European Symposium on Research in Computer Security (ESORICS)*, 2005.

[4] S. Brands, *Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy*. MIT Press, 2000.

[5] D. Boneh, X. Boyen, and H. Shacham, "Short signatures from the weil pairing," *Journal of Cryptology*, 2011.

[6] F. Zhang, Y. Liu, and H. Li, "Efficient batch verification for distributed anonymous identity authentication," in *Proceedings of the 2020 IEEE International Conference on Distributed Computing Systems*, 2020.

[7] M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, "Dynamic universal accumulators for ddh groups and their application to attribute-based anonymous credential systems," in *Cryptographers' track at the RSA conference*. Springer, 2009, pp. 295–308.

[8] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *International conference on the theory and application of cryptology and information security*, 2010.

[9] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, 1985.

[10] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," Cryptology ePrint Archive, Paper 2001/019, 2001.

[11] Camenisch, Jan and Lysyanskaya, Anna, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Advances in Cryptology—EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck*. Springer, 2001.

[12] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," *Cryptology ePrint Archive*, 2013.

[13] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers," *arXiv preprint arXiv:1802.07344*, 2018.

[14] Y. Jiang, K. Zhang, Y. Qian, and L. Zhou, "Anonymous and efficient authentication scheme for privacy-preserving distributed learning," *IEEE Transactions on Information Forensics and Security*, 2022.

[15] H. Yu, C. Du, Y. Xiao, A. Keromytis, C. Wang, R. Gazda, Y. T. Hou, and W. Lou, "Aaka: An anti-tracking cellular authentication scheme leveraging anonymous credentials," in *Proceedings 2024 Network and Distributed System Security Symposium*, 2023.

[16] A. Connolly, P. Lafourcade, and O. Perez Kempner, "Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes," in *IACR International Conference on Public-Key Cryptography*. Springer, 2022, pp. 409–438.

[17] O. Mir, B. Bauer, S. Griffy, A. Lysyanskaya, and D. Slamanig, "Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[18] S. Thakur, "Batching non-membership proofs with bilinear accumulators," *Cryptology ePrint Archive*, 2019.

[19] D. Boneh, B. Bünz, and B. Fisch, "Batching techniques for accumulators with applications to iops and stateless blockchains," in *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019*.

[20] M. Jia, J. Chen, K. He, M. Shi, Y. Wang, and R. Du, "Generic construction of threshold credential management with user-autonomy aggregation," *IEEE Transactions on Information Forensics and Security*, 2023.

[21] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen, "Batch verification of short signatures," *Journal of cryptology*, 2012.

[22] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Annual international cryptology conference*. Springer, 2004, pp. 56–72.

[23] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Selected Areas in Cryptography: 6th Annual International Workshop, SAC'99 Kingston, Ontario, Canada*, 2000.

[24] J. Doerner, Y. Kondi, E. Lee, A. Shelat, and L. Tyner, "Threshold bbs+ signatures for distributed anonymous credential issuance," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 773–789.

[25] C. Li, J. Ning, S. Xu, C. Lin, J. Li, and J. Shen, "Dtacb: Dynamic threshold anonymous credentials with batch-showing," *IEEE Transactions on Information Forensics and Security*, 2024.

[26] W.-Z. Yeoh, M. Kepkowski, G. Heide, D. Kaafar, and L. Hanzlik, "Fast {IDentity} online with anonymous credentials ({FIDO-AC})," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023.

[27] J. Ma, S. Xu, J. Ning, X. Huang, and R. H. Deng, "Catch me if you can: A secure bilateral access control system with anonymous credentials," *IEEE Transactions on Services Computing*, 2023.

[28] M. Rosenberg, J. White, C. Garman, and I. Miers, "zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023.

[29] D. Rathee, G. V. Policharla, T. Xie, R. Cottone, and D. Song, "Zebra: Snark-based anonymous credentials for practical, private and accountable on-chain access control," *Cryptology ePrint Archive*, 2022.

[30] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of CRYPTOLOGY*, vol. 13, pp. 143–202, 2000.