Computer Standards & Interfaces 97 (2026) 104097

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

Fully decentralized period k-times anonymous authentication with access criteria[☆], [☆☆]

Hongyan Di [a], Yinghui Zhang [a],[*], Ziqi Zhang [a], Yibo Pang [a], Rui Guo [a], Yangguang Tian [b]

[a] School of Cyberspace Security, Xi'an University of Posts & Telecommunications, 710121, Xi'an, China
[b] University of Surrey, GU2 7XH, Surrey, UK

ARTICLE INFO

Keywords:
Fully decentralized
Publicly auditable
Access criteria
Anonymous authentication
Signature proof of knowledge

ABSTRACT

The explosive growth of Internet user devices highlights the strong and urgent need for digital identity infrastructure. However, the existing decentralized identity schemes are still not fully decentralized, and there is still a contradiction between publicly auditable credentials and maintaining anonymity. Therefore, using advanced cryptographic techniques such as signature proof of knowledge, Pedersen commitment, and Merkle tree, this paper propose a fully decentralized period k-times anonymous authentication with access criteria. The scheme allows user credentials to be publicly audited, users can manage their identity independently, and the verifier can not only verify the user's identity, but also implement access control. The issuer does not need to hold a key or maintain a list, and it can still authenticate even after the trusted center is attacked, and only three zero-knowledge proofs are needed for registration and verification. The security analysis indicates that this scheme satisfies unforgeability, anonymity, unlinkability and attribute privacy. Performance evaluation shows significant improvements in both computational and communication efficiency over existing schemes.

1. Introduction

With the surge in digital services accessed through network connections, the number of digital identities has seen an unprecedented increase. Therefore, the vast majority of the global population has at least one digital identity, which becomes the key to unlocking a variety of online functions and services. However, the concept of digital identity goes far beyond human identity recognition [1]. With the wide adoption of IoT and the powerful functions of the 5th Generation Mobile Communication Technology (5G) network, as well as the upcoming 6th Generation Mobile Communication Technology (6G), the number of connected devices has increased significantly [2]. These devices require unique digital identities to enable their participation in digital ecosystems, such as establishing secure communications.

Authentication and authorization are crucial security-related core tasks in the digital world. Their purpose is to ensure the authenticity of the identities of the communicating parties and implement access control over digital resources such as services. The core of this system is the concept of digital identity. The evolution of digital identity has gone through multiple eras, during which digital identity recognition has gradually shifted from centralized to decentralized identity models [3]. In fact, the way entities prove the ownership of digital identities may be affected by various vulnerabilities [4]. The current Internet ecosystem generally adopts the centralized Identity Provider (IdP) model, with tech giants such as Google and Facebook (e.g., Meta) serving as the custodians of digital identities. Other services can directly rely on the identity information provided by IdP. This architecture simplifies the authentication process by achieving single sign-on through protocols such as OAuth, it has fundamental flaws when examined from the perspective of privacy protection, users lose control over their digital identities [5], and all their identity attributes are centrally stored in the IdP's servers. Users neither know the specific usage of these data nor can they effectively manage their flow. More seriously, this architecture has created a dangerous "data island" phenomenon—IdP can fully

[☆] This article is part of a Special issue entitled: 'Information Security and Privacy' published in Computer Standards & Interfaces.

[☆☆] This work is supported by the National Cryptologic Science Fund of China (2025NCSF02037), the National Natural Science Foundation of China (62072369), the Youth Innovation Team of Shaanxi Universities (23JP160), the Shaanxi Special Support Program Youth Top-notch Talent Program, the Technology Innovation Leading Program of Shaanxi (2023-YD-CGZH-31), the Technology Innovation Guidance Special Fund of Shaanxi Province (2024QY-SZX-17), the Graduate Innovation Fund of Xi 'an University of Posts and Telecommunications (CXJJBDL2024004).

[*] Corresponding author.
E-mail addresses: 15029659213@163.com (H. Di), yhzhaang@163.com (Y. Zhang), qiqizhang0408@163.com (Z. Zhang), ybpang1998@163.com (Y. Pang), guorui@xupt.edu.cn (R. Guo), yangguang.tian@surrey.ac.uk (Y. Tian).
URLs: https://www.xiyou.edu.cn/ (Y. Zhang), http://www.surrey.ac.uk (Y. Tian).

https://doi.org/10.1016/j.csi.2025.104097
Received 12 July 2025; Received in revised form 26 September 2025; Accepted 11 November 2025
Available online 19 November 2025

grasp the cross-platform service usage trajectory and behavioral characteristics of users, essentially constructing a panoramic user profile. IdP, on the other hand, can obtain information about all the network services used by users (and related usage data). When the server storing user data is invaded, sensitive personal information may be "obtained" by malicious attackers, causing significant loss of personal data and damaging the reputation of stakeholders [6]. In 2022 alone, there were over 1800 major data breaches worldwide, involving more than 400 million user records. The increasing number of data breach cases has raised significant concerns to data confidentiality and transparency in the field of digital identity management. In addition, centralized identity management systems rely on specific identity service nodes, making them vulnerable to single point of failure problem [7].

Therefore, the increasing popularity of online services, the growing trend of decentralization, and the rising awareness of the shortcomings of traditional methods are paving the way for more secure and privacy-protecting approaches. Under this trend, supported by current laws and regulations (such as the General Data Protection Regulation (GDPR) of the European Union) [8], the concept of Self-Sovereign Identity (SSI) [9] has attracted significant attention from both academia and industry. SSI is based on the idea that individuals should have full control over their information without being forced to outsource data to any centralized institution or third party. Such technologies play a crucial role in establishing trust among entities (including non-human entities such as humans and IoT devices) and ensuring communication security through digital identities. Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), as effective solutions for enhancing privacy and security, have been promoted in multiple application fields such as intelligent transportation and smart healthcare. These standards can be extended to anyone or anything, covering cloud, edge, and IoT resources. It is worth noting that several institutions, including industry giants such as Microsoft, have recently developed and released a variety of implementation plans to support these technologies. In addition, global government agencies are also actively promoting the widespread application of DIDs and VCs. For instance, the European union promulgated regulation 2024/1183 [10] in May 2024, establishing the European digital identity framework, aiming to provide European citizens with digital passes for cross-border access to public and private services through the SSI system. This represents a significant milestone in the development of digital identity solutions. However, current decentralized anonymous authentication schemes still face significant challenges. These include the inability to achieve full decentralization, a lack of mutual trust between users and issuers, and the persistent contradiction between public verifiability and true anonymity. Against this backdrop, AI-driven identity threat analysis has become a new focus of security research. Initiatives such as the Global Digital Identity Wallet (GDIW) have launched cross-border interoperability tests, while "Digital Identity Chain" has completed the integration of DIDs with the national government service platform—efforts that represent preliminary but critical explorations in addressing these underlying issues.

2. Relate work

2.1. Decentralized anonymous credential (DAC)

In the 1980s, David Chaum [11,12] introduced privacy-preserving cryptographic techniques, aiming to create a more privacy-focused and user-centered authentication and authorization solution. It enables users to prove their membership, identity, or any other arbitrary attribute in a group in a privacy-preserving manner. Such techniques are often referred to as anonymous credentials (ACs), and various methods for building AC systems have been widely studied in the academic community. However, since Camenish and Lysyanskaya [13] first proposed a completely anonymous credential scheme in 2001, a large number of anonymous credit construction schemes suitable for various scenarios have emerged. These include zero-knowledge credentials, lightweight anonymous credentials without heavy zero-knowledge proofs and other computationally intensive operations, self-blinding credentials, group signatures, AC schemes without unlinkability, and post-quantum AC schemes. In order to reduce the trust dependence of the credential issuance process on a central authority in traditional anonymous credential schemes, Garman et al. [14] proposed the concept of decentralized anonymous credential (DAC), which allows users to construct and manage credentials in a completely anonymous manner. Derler et al. [15] designed a new revocable multi-show attribute anonymous credential based on previous work, which has good scalability and constant operation of two roles. Bui and Aura [16] developed a distributed access control revocation framework to facilitate the manipulation of revocation methods. Subsequently, Sonnino et al. [17] proposed a special selective disclosure voucher solution based on blind signatures and bilinear pairing, which holds short and highly efficient vouchers. Inspired by Sonnino's work, Halpin [18] redesigned the tagging mechanism to improve scalability and support embedding arbitrary attributes. Cui et al. [19] constructed a Blockchain Digital Identity Management System (BDIdM) by extending the functional features of the DAC scheme [14], which enabled limited reusability of specific credentials on the premise of maintaining the security of the DAC scheme. In addition, decentralized anonymous credentials are widely integrated with other scenarios. Lin et al. [20] applied the DAC scheme to the smart grid scenario and enhanced the privacy protection mechanism. The solutions combined with the application scenarios of blockchain-based Internet of Vehicles include [21–25], Zeng et al. [26] also applied anonymous credentials to cross-domain authentication in IIoT.

2.2. k-Time anonymous authentication (k-TAA)

The k-period anonymous authentication allows users to be authenticated up to k-times within a certain time period while remaining anonymous. Teranishi et al. [27] introduced the first k-TAA scheme, allowing the identification of users who exceeded the authentication limit. Nguyen and Safavi-Naini [28] extended this concept to dynamic k-TAA, enabling each authenticator to independently grant or revoke access rights. Au et al. [29] proposed a fixed-size dynamic k-times. Chaterjee et al. [30] proposed a k-TAA scheme based on physically unclonable functions (PUFs), which is applicable to trusted platform modules (TPM). Huang et al. [31] designed an efficient k-TAA system tailored for pay-as-you-go pricing, facilitating multiple service accesses and related payments within each certification cycle. However, many existing k-TAA schemes fail to provide periodic anonymous authentication. Although the existing schemes [32,33] support periodic anonymous authentication, they have deficiencies in supporting the selective disclosure of credential attributes to achieve fine-grained authentication. In addition, they require a large number of pairing operations, resulting in significant verification delays. In contrast, scheme [34,35] supports periodic k-times anonymous authentication while reducing cumbersome pairing operations. However, scheme [34] does not support credential revocation. As shown in Table 1, our scheme, while meeting the above requirements, supports full decentralization and access control.

- *Research Contributions*

  Next, we list the main research contributions of this paper.

  The Proposed Scheme: We propose a fully decentralized k-times period anonymous authentication scheme with access control. The scheme enforces both access criteria and authentication during the verification process, while eliminating the need for issuers to hold keys or maintain lists, thus remaining secure even if the trusted center is compromised. Only three zero-knowledge proofs are required for registration and verification.

  Security Analysis: We conducted a correctness and theoretical security analysis based on the game definition of the proposed

Table 1
Function comparison.

Security features	[29]	[30]	[31]	[33]	[19]	[34]	[35]	Our Scheme
Anonymity	✓	✓	✓	✓	✓	✓	✓	✓
Unlinkability	✓	N.A	✓	N.A	✓	✓	✓	✓
k-times period anonymous authentication	×	×	×	✓	×	✓	N.A	✓
Publicly auditable	N.A	×	N.A	N.A	✓	✓	✓	✓
Select attribute disclosure	×	×	×	×	✓	✓	N.A	✓
Key forward and backward secure	✓	✓	✓	✓	✓	✓	✓	✓
Reveal violator's identity without TTP	✓	✓	×	✓	✓	✓	×	✓
Issuer not hold key and identity list	×	×	×	×	×	×	×	✓
Support credential revocation	✓	✓	✓	✓	✓	×	✓	✓

Note*: ✓: Support this feature; ×: Does not support this feature; N.A: No applicable; TTP: Trusted third party.

scheme. By simulating games and citing programmable random oracles and fork lemmas, among other techniques, we demonstrated that the scheme meets the requirements of unforgeability, anonymity, unlinkability, and attribute privacy. This analysis emphasizes that the plan has protected the integrity and validity of the data.

Performance Evaluation: We conducted a detailed analysis of this authentication scheme, demonstrating its efficiency advantages over existing authentication schemes. Tests were also carried out on secp256k1 and BLS12-381 curves, verifying that the proposed algorithm performs better on lightweight curves.

- *Structure of Paper*
  The remaining paper is structured as follows: Section 3 introduces the problem assumptions and fundamentals. Section 4 defines the syntax, security model, and detailed construction of the scheme. Section 5 analyzes its correctness and theoretical security. Section 6 evaluates performance in terms of computation and communication overhead, and Section 7 concludes the paper.

3. Preliminaries

3.1. Group description and hardness assumptions

A group generator $GGen(1^{\kappa}) \to (\mathbb{G}, q)$ inputs a security parameter κ and outputs a cyclic group $\mathbb{G}$ of prime order q. This scheme is based on the following hard problem assumption.

Definition 2.1 (*Discrete Logarithm Problem (DLP) Assumption*). Let g be a generator of a group $\mathbb{G}$. Given a tuple $(g, g^a) \in \mathbb{G}^2$, where $a \in \mathbb{Z}_q^*$, the Discrete Logarithm Problem is output a. The DLP assumption holds if for all PPT adversary $\mathcal{A}$, the advantage is negligible.

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DLP}}(\kappa) = |Pr[\mathcal{A}(g, g^a)| = a] \leq negl(\kappa).$$

Definition 2.2 (*Decisional Diffie–Hellman (DDH) Assumption*). Let $\mathbb{G}$ be a group of order a large prime q, g be the generator of $\mathbb{G}$. The input is a random quadruple $\mathcal{R} = (g, g^x, g^y, g^{xy}) \in \mathbb{G}^3$, and quadruple $\mathcal{D} = (g, g^x, g^y, g^z) \in \mathbb{G}^3$, where $x, y, z \leftarrow \mathbb{Z}_q^*$. It is computationally hard for adversary $\mathcal{A}$ to distinguish between two tuples, the advantage of PPT adversary $\mathcal{A}$ is negligible.

$$Adv_{\mathcal{A}}^{\mathrm{DDH}}(\kappa) = |Pr[\mathcal{A}(\mathcal{R}) = 1] - Pr[\mathcal{A}(\mathcal{D}) = 1]| \leq negl(\kappa).$$

Definition 2.3 (*Computing Diffie–Hellman (CDH) Assumption*). Let $\mathbb{G}$ be a cyclic group of order q with generator g. Given the tuple $\mathcal{I} = (g, g^a, g^b)$ where $a, b \leftarrow \mathbb{Z}_q^*$, computing g^{ab} is hard. For all probabilistic polynomial-time (PPT) algorithms $\mathcal{A}$, the advantage probability of successfully solving the CDH problem is negligible.

$$Adv_{\mathcal{A}}^{CDH}(\kappa) = \left| Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab}\right] \right| \leq negl(\kappa).$$

where κ is a security parameter, $negl(\kappa)$ denotes a negligible function.

3.2. Zero-knowledge proof

A signature proof of knowledge (SPK) is a non-interactive zero-knowledge proof (ZKP) technique that enables a prover to demonstrate knowledge of a secret value without revealing it, while also signing a message. We constructed a cyclic group $\mathbb{G}$ of prime order q and employed the Fiat–Shamir heuristic [36] to convert an interactive proof into a non-interactive one. These non-interactive constructs are precisely referred to as signature proofs of knowledge (SPK). All the signatures of knowledge are secure in the random oracle model. According to the symbols introduced by Camenisch and Stadler [37], $PoK\{(x) : y = g^x\}$ represents the zero-knowledge proof protocol between the prover and the verifier. Such prover knows $x \in \mathbb{Z}_p$ and $y = g^x \in \mathbb{G}$. The corresponding non-interactive signature knowledge proof on the message m should be expressed as $SPK\{(x) : y = g^x\}(m)$. It can be regarded as a signature on the message m, which is signed by a key pair (g^x, x) based on discrete logarithms.

3.3. Pedersen commitment

Literature [38] uses Poseidon to realize the hash of Merkle tree and commitment. Instantiate another method of using Pedersen hashing and perfectly hiding commitments in the scheme. The Pedersen commitment algorithm as follows:

- $Gen(1^{\kappa}) \to ck$: Select a finite group $\mathbb{G}$ with a large prime order q, and choose two generators g and h from the group $\mathbb{G}$. The parameters of this commitment scheme are $ck = (\mathbb{G}, q, g, h)$.
- $Commit(ck, u) \to c$: Generate a commitment c for a secret value u. The commitment party randomly selects a blind factor r and then calculates $c = g^u h^r$.
- $OpenCom(ck, c, u, r) \to 0/1$: The verifier checks whether c is equal to $g^u h^r$.

3.4. Merkle tree

In the proposed scheme, the Merkle tree T is used to represent the membership of the set. The root of the tree T is denoted T_{root}. The Merkle tree has the following functions:

- $T.Insert(v) \to T$: Inserts the value v into the next available leaf in T and returns the modified tree.
- $T.Remove(v) \to T'$: Removes v from the tree, if it exists, and returns the modified tree T'.
- $T.AuthPath(v) \to \theta$: Generate an authentication path θ that proves $v \in T$. The size of θ is proportional to the height of the tree, ensuring efficient verification in cryptographic protocols.

Table 2
Summary of notations.

Symbol	Description
$\mathcal{U}, \mathcal{I}, \mathcal{V}$	User, Issuer, Verifier
λ	Security parameter
h	The maximum height of the Merkle tree
m	The maximum number of attributes
n	The number of access criteria the verifier is allowed to define
ι_{pub}, ι_{zk}	Verify the access policy for ancillary information when the request is issued
$iaux_{zk}, iaux_{pub}$	Auxiliary information when requesting registration
ϕ_i	The verifier defines the ith access criterion
aux_i	Show proof of auxiliary information
$Attrs = \left\{attr_i\right\}_{i=1}^{m}$	The ith attribute of the user and the attribute set
w	Witness Collection
ctx	Context information
I, V	Collection of issuance criteria and access criteria
$\Pi_U^1, \Pi_V^1, \tilde{\Pi}$	Zero-knowledge proofs generated by the user and issuer
$s'' \leftarrow \mathbb{Z}_q^*$	A secret random number randomly selected by the issuer
θ	The authentication path generated by the Merkle tree
$T_{root}, T_\kappa, T'_\kappa$	Merkle tree root, Merkle tree, updated Merkle tree

Note*: $\iota, \phi : \mathcal{A} \rightarrow \{0, 1\}$ is a predicate over the user's attributes that needs to be satisfied in order to pass verification, i.e., verification only passes if $\iota_{pub}(iaux_{pub}) = 1$, $\phi(Attrs, aux) = 1$.

3.5. Pseudo-Random Function (PRF)

A Pseudo-Random Function (PRF) is a family of computational functions $\left\{F_k\right\}$, where k is a key and F_k is a function from the input space to the output space. For an ideal PRF, when the key k is unknown, its output is computationally indistinguishable from that of a true random function. We construct a PRF with efficient correctness proof. We adopt the specific PRF construction proposed by Dodis and Yampolskiy [39] (DY-PRF). The DY-PRF is defined by the tuple $(\mathbb{G}, q, g, s)$, where $\mathbb{G} = \langle g \rangle$ is a cyclic group of prime order q and $s \in \mathbb{Z}_q$. For an input k, $PRF_{g,s}(k)$ is defined as $PRF_{g,s}(k) : k \mapsto g^{-(s+k+1)}$. There exists an efficient proof of correct formation for the output, and as long as the q-DDHI assumption holds, the output $PRF_{g,s}(k)$ is indistinguishable from a random element in $\mathbb{G}_q$.

4. Proposed scheme

In this section, we describe in Table 2 all the symbolic definitions involved as well as the implications, followed by defining the syntax and designing the scheme.

4.1. Syntax and security model

4.1.1. Security definition

The security of the system is defined by the standard properties of anonymous credentials, including unforgeability, anonymity, unlinkability, and attribute privacy. In our model, the attacker is assumed to have only polynomial-time computational capability, and all communications occur over open channels.

Threat Model. Our model considers adversaries as external attackers intercepting or modifying communications without breaking hard cryptographic problems, internal attackers misusing valid credentials for forgery, transfer, or link attacks, semi-honest verifiers inferring user identities or attributes while following the protocol, and trusted-but-curious issuers complying with the protocol but attempting to snoop on user data.

4.1.2. Syntax definition

Referring to the ideal function $\mathcal{F}$ in [38], the zk-credit anonymous credential approach realizes $\mathcal{F}$ using *Groth16* [40], which is not suitable for authentication. In this work, $\mathcal{F}$ is instantiated using signatures of knowledge, resulting in an algorithm that meets the authentication requirements. The specific algorithm is as follows:

- $Setup(1^\lambda, 1^h, 1^m) \rightarrow pp$: The algorithm inputs the security parameter λ, the maximum height h of the Merkle tree, and the maximum number m of attributes in a credential. Generates the system parameters pp.
- $IssueSetup_I(pp) \rightarrow (I, \iota_{pub})$: The algorithm inputs the public parameter pp, outputs the issue criteria set I and the issue criteria for verifying public auxiliary information ι_{pub}.
- $ShowSetup_V(pp) \rightarrow V$: The verifier sets up n access criteria to define the user's access policy. This algorithm outputs a collection of access criteria $V = \{\phi_1, \phi_2, \dots, \phi_n\}$ where each ϕ_i represents an access criteria.
- $IssueReq_U(pp, I, Attrs, w, ctx, iaux_{zk}, iaux_{pub}) \rightarrow \left(Cm, \left(\Pi_U^1, iaux_{zk}\right), iaux_{pub}\right)$: The issue request algorithm inputs the public parameters pp, the issue criteria I, the set of attributes $Attrs$ of $\mathcal{U}$, the secret value w, the context ctx, and the auxiliary information $(iaux_{zk}, iaux_{pub})$. $\mathcal{U}$ generates the Π_U^1 associated with $iaux_{zk}$ and outputs $((\Pi_U^1, iaux_{zk}), iaux_{pub})$.
- $IssueGrant_I(pp, (I, \iota_{pub}), (\Pi_U^1, iaux_{zk}), iaux_{pub}) \rightarrow (s'', (\theta, T_{root}), k, T_\kappa)$: The algorithm inputs the zero-knowledge signature Π_U^1, and the auxiliary information $(iaux_{zk}, iaux_{pub})$. Then $\mathcal{I}$ return the random value s'', authentication path θ, number of times k to $\mathcal{U}$, and locally generated Merkle tree T_κ.
- $ShowCred_U(pp, V, T_{root}, cred, \theta, \left\{w_i, aux_i\right\}_{i=1}^{n}) \rightarrow (\tilde{\Pi}, \left\{aux_i\right\}_{i=1}^{n})$: $\mathcal{U}$ inputs the root T_{root} of the affiliated tree, the credential $cred$, and the authentication path θ. $\mathcal{U}$ shows that the sent credential satisfies the access criterion ϕ_i and proves that the displayed credential belongs to the tree T_κ. Then, the algorithm outputs $(\tilde{\Pi}, \left\{aux_i\right\}_{i=1}^{n})$.
- $VerifyShow_V(pp, V, (cred, T_{root}), (\tilde{\Pi}, \left\{aux_i\right\}_{i=1}^{n})) \rightarrow 0/1$: $\mathcal{V}$ verifies that the credentials $cred$ displayed by $\mathcal{U}$ meet the access criteria and that $cred$ belongs to the Merkle tree T_κ, $\mathcal{V}$ outputting 0/1.
- $RevokeCred_I(pp, T_\kappa, cred) \rightarrow T'_\kappa$: $\mathcal{I}$ revoke the $cred$ registered by dishonest users and update the Merkle tree T_κ to T'_κ.

4.1.3. Security requirements

The scheme is required to satisfy the following security requirements:

Unforgeability: Attackers cannot forge valid credentials and deceive validators into performing correct verification. This game is reduced to discrete logarithm or CDH problems.

Anonymity: Credentials are displayed without revealing the user's identity. This game specification is reduced to the DDH problem.

Unlinkability: Different displays of the same certificate cannot be linked, even if the merkle path remains identical across multiple authentications.

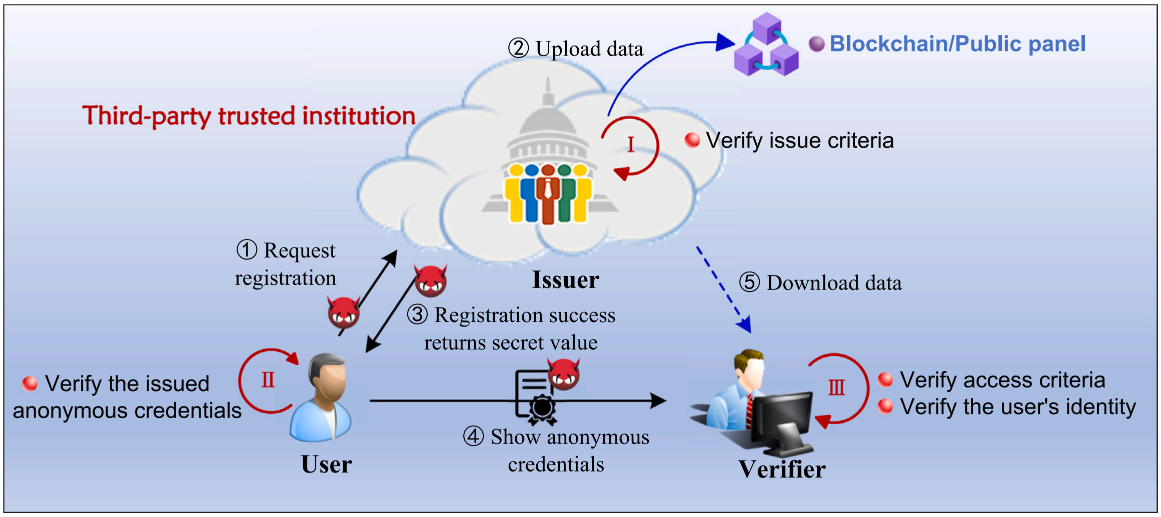

② Upload data
Blockchain/Public panel
Third-party trusted institution
I
Verify issue criteria
① Request registration
Issuer
⑤ Download data
③ Registration success returns secret value
Verify the issued anonymous credentials
II
III
Verify access criteria
Verify the user's identity
④ Show anonymous credentials
User
Verifier

Fig. 1. System Model.

Attribute Privacy: Hides attributes when displaying credentials unless the access policy requires them to be displayed.

Security is analyzed using a formal game-based model [41] under the random oracle assumption [42]. The game is defined as follows:

***Game 1:* Unforgeability Game**

Setup. The challenger-$\mathcal{C}_1$ run system initialization algorithm $Setup(1^\lambda, 1^h, 1^m)$ generate pp, send pp to adversary $\mathcal{A}_1$. $\mathcal{C}_1$ save issuer private key isk.

Query. In this phase, the adversary $\mathcal{A}_1$ can querie three random oracles, as follows:

1. $\mathcal{H}\_Query$: $\mathcal{A}_1$ query random oracle $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$, $\mathcal{C}_1$ random response and recording.
2. $Query_2$: $\mathcal{A}_1$ query the issuer to registered certificate, $\mathcal{C}_1$ use the simulator $\mathcal{S}$ Simulate the interaction between $IssueReq$ and $IssueGrant$, using the programmability of random oracle to generate effective SPK_2.
3. $Query_3$: $\mathcal{A}_1$ query certificate display, simulate the interaction between $ShowCred$ and $VerifyShow$, and simulate SPK_3 using a zero-knowledge simulator.

Forgery. $\mathcal{A}_1$ output a forged certificate $cred^*$, correspond Merkle tree path θ^*, satisfy that $cred^*$ is not on the list of previously issued credentials. $VerifyShow$ accept $cred^*$ and θ^*. $\mathcal{A}_1$ wins conditional on the output of valid forged credentials.

***Game 2:* Anonymity and Unlinkability Game**

Setup. The challenger-$\mathcal{C}_2$ run system initialization algorithm $Setup(1^\lambda, 1^h, 1^m)$ generate pp, send pp to adversary $\mathcal{A}_2$. $\mathcal{C}_2$ save issuer private key isk.

Query. Adversary $\mathcal{A}_2$ can continue to query issuance and presentation, but cannot query revocation or presentation of challenge credentials.

challenge. The adversary $\mathcal{A}_2$ selects the identity and attribute sets of two users, $\left(I_0, {Attrs_0}^*\right), \left(I_1, {Attrs_1}^*\right)$, which satisfy the same access policy. Send it to the challenger $\mathcal{C}_2$. $\mathcal{C}_2$ randomly selects $b \leftarrow \{0,1\}$ to generate a credential for I_b and display it (i.e., run $ShowCred$ to generate Π_b), and then gives Π_b to $\mathcal{A}_2$.

Guess. $\mathcal{A}_2$ outputs b' and wins if $b' = b$.

4.2. Scheme construction

In this scheme, the user is untrusted, the issuer is semi-trusted, the channel between the verifier and the issuer is trusted, and the rest of the channels are untrusted channels. Attackers can steal information from untrusted channels, forge information and impersonate users. Therefore, this paper adopts the method of zero-knowledge proof to realize the user's verification of the certificate sent by the issuer, and prove to the verifier that the certificate is the user's own, and at the same time, it can reduce the risk of privacy leakage. As shown in Fig. 1.

- **Issuer:** The issuer is the issuer of the certificate, usually an authority or trusted entity (such as government, enterprise, decentralized organization, etc.), which is responsible for verifying the identity or attribute of the user and generating the encrypted credential. Before sending the certificate, the issuing criteria will be verified.
- **User:** The user is the holder of the credential, requests the credential from the issuer, upon receipt, verifies the credential.
- **Verifier:** The verifier is the receiver of credentials, who receives the user's credentials, goes through a secure channel, downloads the criteria and auxiliary verification data, verifies the access criteria, and then verifies the user's identity.

4.2.1. System initialization

$Setup\left(1^\lambda, 1^h, 1^m\right) \rightarrow pp$:

– $\mathcal{I}$ select a cyclic group $\mathbb{G}$ of order q, and generate generators $(g_0, g_1, g_2, \gamma, h_0, h_1, h_2, \widetilde{u}, \{u_i\}_{i\in[0,n]}) \in \mathbb{G}$, along with hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \times \{0,1\}^* \rightarrow \mathbb{Z}_q^*$;

– Define a Merkle tree of height h, where for public input $(T_{root}, cred)$, it can prove $cred \in T_\kappa$ through an authentication path θ;

– Define the global period $epoch$ and pseudorandom function $PRF_{g,s}(k) : k \mapsto \frac{1}{g^{s+k+1}}$;

– $\mathcal{I}$ selects random number $y_1, y_2 \leftarrow \mathbb{Z}_q^*$, computes $Y_1 = h_1^{y_1}$, $Y_2 = h_2^{y_2}$, and sets the issuer secret key $isk = (y_1, y_2)$ and issuer public key $ipk = (Y_1, Y_2)$;

– Set the public parameters $pp := \left(q, \mathbb{G}, g_0, g_1, g_2, \gamma, h_0, h_1, h_2, \widetilde{u}, \{u_i\}_{i\in[0,n]}, H_1, H_2, T_\kappa, T_{root}, epoch, ipk\right)$.

$IssueSetup_I(pp) \rightarrow \left(I, \iota_{pub}\right)$:

– Define the relevant issuance criteria $\iota = (\iota_{zk}, \iota_{pub})$, set $IssueCriteria[I] := IssueCriteria[I] \cup \iota$;

– For the public input auxiliary information $iaux_{zk}$, prove: $\iota_{zk}(Attrs, iaux_{zk}) = 1$;

– Publish (I, ι_{pub}).

$ShowSetup_V(pp) \rightarrow V$:

– $\mathcal{V}$ define access criteria ϕ for user attributes $Attrs$ (Multiple access criteria ϕ_i can be defined), and set $AccessCriteria[V] := AccessCriteria[V] \cup \{\phi_i\}$;

– For public input $(T_{\text{root}}, cred, aux)$, prove: $\phi(Attrs, aux) = 1 \Lambda cred$;

– Publish the access criteria set V.

4.2.2. Credential registration

$IssueReq_U\left(pp, I, Attrs, w, ctx, \left(iaux_{zk}, iaux_{pub}\right)\right) \rightarrow \left(Cm, \left(\Pi_U^1, iaux_{zk}\right), iaux_{pub}\right)$:

– $\mathcal{U}$ generate anonymous key nk and rate-limiting key rk using pseudorandom function PRF and context ctx, calculate $nk := PRF(ctx), rk := PRF(epoch \parallel ctx)$, define m attributes $Attrs = \{attr_1, attr_2, \dots, attr_m\}$;

– Select a random blind factor $r \leftarrow \mathbb{Z}_q^*$ and compute pedersen commitment Cm, where $Cm \in \mathbb{G}$:

$$Cm = Commit(nk, rk, Attrs; r) = g_1^{nk} g_2^{rk} \left(\prod_{i=1}^{m} u_i^{H_1(attr_i)}\right) \cdot h_0^r;$$

– Set $w := (r, nk, rk, Attrs)$ (collect private witness w), select $x_u, s', t \leftarrow \mathbb{Z}_q^*$ and generate Π_U^1:

$$\Pi_U^1 = SPK_1 \left\{ \begin{array}{ll} & \left(x_u, s', t, r, nk, rk, Attrs\right) : \\ & X_u = g_1^{x_u} g_2^{s'} \\ \wedge & \zeta = Y_1^{x_u} Y_2^{s'} \cdot Cm^t \\ \wedge & \iota_{zk}\left(Attrs, iaux_{zk}\right) = 1 \end{array} \right\} \left(X_u, \zeta, iaux_{zk}, iaux_{pub}\right);$$

– $\mathcal{U}$ send $(\Pi_U^1, X_u, \zeta, iaux_{zk}, iaux_{pub})$ to Issuer $\mathcal{I}$;

– $\mathcal{U}$ received Π_V^1. If verification passes, receive the returned authentication path θ, s'' and k;

– Locally store $(nk, rk, r, Attrs, \theta, s, t, epoch, k)$, where $s = s' + s''$ and k is the maximum allowed accesses within epoch $epoch$.

$IssueGrant_I(pp, (I, \iota_{pub}), (\Pi_U^1, iaux_{zk}), iaux_{pub}) \rightarrow \left(cred, s'', \left(\theta, T_{root}\right), k, T_\kappa\right)$:

–$\mathcal{I}$ verify $\iota_{pub}(iaux_{pub})$, ι_{pub} checks for publicly auxiliary information $iaux_{pub}$;

– Verify $\Pi_U^1 := SPK_1$, where Π_U^1 proves the correctness of $(\zeta, X_u, iaux_{zk}, iaux_{pub})$ and that the hidden attributes satisfy the issuance criteria ι_{zk}. If verification fails, reject issuance and abort $\perp$;

– Else verification passes, $\mathcal{I}$ randomly selects $s'' \leftarrow \mathbb{Z}_q^*$, and define the maximum times of accesses k allowed by users within $epoch$, calculate $cred := (\zeta \cdot Y_2^{s''}) \cdot u_0^{H_1(epoch\|k)}$, run $T_\kappa = T$.Insert($cred$) registers the anonymous credential. Where the registered $cred$ is only known privately by the issuer. Then, run $\theta = T_\kappa$.AuthPath($cred$) generate authentication path. Updated Merkle tree root T_{root}, and upload to a public panel such as blockchain;

– Next, select $z_0, z_1 \leftarrow \mathbb{Z}_q^*$ and generate Π_V^1:

$$\Pi_V^1 = SPK_2 \left\{ \begin{array}{ll} & \left(z_0, z_1, y_1, y_2\right) : \\ & Y_u = h_1^{y_1} h_2^{y_2} \\ \wedge & \mathcal{Z} = \left(\zeta \cdot Y_2^{s''}\right)^{z_1} \cdot u_0^{H_2(epoch\|k) \cdot z_0} \end{array} \right\} \left(Y_u, s'', k, \mathcal{Z}\right);$$

– $\mathcal{I}$ store the Merkle tree T_κ and send $(\Pi_V^1, s'', k, \theta)$ to user $\mathcal{U}$.

4.2.3. Show and verification certificate

$ShowCred_U\left(pp, V, T_{root}, \text{cred}, \theta, \left\{w_i, aux_i\right\}_{i=1}^n\right) \rightarrow \left(\widetilde{\Pi}, \left\{aux_i\right\}_{i=1}^n\right)$:

– User $\mathcal{U}$ sends an access request message msg, and the verifier returns a random number $R = H_2(nonce \parallel msg)$;

– $\mathcal{U}$ locally retrieves the verifier's access criteria V and the root node T_{root} of the tree containing $cred$;

– Upon receiving $(nonce, R)$, verify $R \overset{?}{=} H_2(nonce \parallel msg)$, then randomly select $\alpha_0 \leftarrow \mathbb{Z}_q^*$. For n access criteria $\Phi' = \{\phi_1, \phi_2, \dots, \phi_n\}$, partition the attribute set into public attributes $ATTR_D$ and secret attributes $\{attr_j \notin ATTR_D\}$. Compute the commitment using blind factor r:

$$\begin{aligned} Cm &= Commit(nk, rk, \{attr_j \notin ATTR_D\}; r) \\ &= \left(g_1^{nk} g_2^{rk} \cdot \prod_{attr_j \notin ATTR_D} u_i^{H_1(attr_j)} \cdot h_0^r\right) \cdot \prod_{attr_i \notin ATTR_D} u_i^{H_1(attr_i)}; \end{aligned}$$

– Next, the times of certificate displays is initialized to $n_j = 1$, and $n_j = n_j + 1$ $(0 \le n_j < k)$ is set for each generation of zero-knowledge Proof $\tilde{\Pi} = SPK_3$. The generation of $\tilde{\Pi} = SPK_3$ is as follows:

$$\tilde{\Pi} = SPK_3 \left\{ \begin{array}{ll} & \left(nk, rk, Attrs, \alpha_0, x_u, s, t, n_j, attr_j \notin ATTR_D\right) : \\ & X_0 = g_0^{\alpha_0} \gamma^{H_1(\theta)} \\ \wedge & \zeta' = Y_1^{x_u} Y_2^{s} \cdot Cm^t \\ \wedge & \eta = PRF_{rk,\tilde{u}}(n_j) = \dfrac{1}{\tilde{u}^{rk+n_j+1}} \\ \wedge & \Gamma = u_0^{x_u} PRF_{nk,\tilde{u}}(n_j)^R = u_0^{x_u} \cdot \tilde{u}^{\frac{R}{nk+n_j+1}} \\ \wedge & 0 \le n_j < k \\ \wedge & \phi_1(Attrs, aux_1) = 1 \\ \wedge & \vdots \\ \wedge & \phi_i(Attrs, aux_i) = 1 \end{array} \right\}$$

$$\times \left(aux_i, X_0, \zeta', \eta, \Gamma, T_{root}\right);$$

– Send $(\tilde{\Pi}, \{aux_i\}_{i=1}^n, X_0, \zeta', \eta, \Gamma, (\theta, T_{root}), \Phi', attr_i \in ATTR_D)$ to the verifier $\mathcal{V}$.

$VerifyShow_V\left(pp, V, \left(cred, T_{root}\right), \left(\widetilde{\Pi}, \left\{aux_i\right\}_{i=1}^n\right)\right) \rightarrow 0/1$:

– $\mathcal{V}$ checks whether the user's submitted Φ' matches its defined access criteria set Φ. Using θ, verify and calculate $cred \overset{?}{=} \zeta' \cdot u_0^{H_2(epoch\|k)}$. If (η, Γ) is valid, it proves that n_j is within the range allowed to be displayed within $epoch$;

– If verification succeeds, accept the request, otherwise reject it and invoke the $RevokeCred$ function to revoke $cred$. For the specific process, please refer to Fig. 2.

4.2.4. Credential revocation

$RevokeCred\left(pp, T_\kappa, cred\right) \rightarrow T'_\kappa$:

– Search for $cred \in T_\kappa$, if $cred$ is not found, terminate the process;

– Else run $T'_\kappa := T_\kappa$. Remove($cred$), store and update the Merkle tree T'_κ;

– Return T'_k and publicly notify that $cred$ has been revoked.

5. Analysis of correctness and security

5.1. Correctness analysis

5.1.1. Details of SPK_1

SPK_1 can be implemented using standard discrete logarithm proof techniques.

1. **(Commitment.)** User $\mathcal{U}$ randomly selects $s_1, s_2, s_3 \in_R \mathbb{Z}_q^*$ and computes:
   $T_1 = g_1^{s_1} g_2^{s_2}, T_2 = Y_1^{s_1} Y_2^{s_2} \cdot Cm^{s_3} = (h_1^{y_1})^{s_1} (h_2^{y_2})^{s_2} \cdot Cm^{s_3}$.
2. **(Challenge.)** The scheme uses non-interactive zero-knowledge proof, where the user $\mathcal{U}$ generates challenge c:

   $$c = H(T_1 \parallel T_2 \parallel X_u \parallel \zeta \parallel iaux_{zk} \parallel iaux_{pub}).$$

3. **(Proof.)** $\mathcal{U}$ generates proof Π_U^1 that satisfies issuer policy $\iota_{zk}, \iota_{zk}(Attrs, iaux_{zk}) = 1$, and computes $S_1 = s_1 - c \cdot x_u, S_2 = s_2 - c \cdot s', S_3 = s_3 - c \cdot t$. The proof $\Pi_U^1 = (c, S_1, S_2, S_3)$, and sends $((\Pi_U^1, iaux_{zk}), iaux_{pub})$ to the issuer $\mathcal{I}$.
4. **(Verify.)** $\mathcal{I}$ computes $T_1' = X_u^c g_1^{S_1} g_2^{S_2}, T_2' = \zeta^c Y_1^{S_1} Y_2^{S_2} \cdot Cm^{S_3}$, and verify: $c \overset{?}{=} H(T_1' \parallel T_2' \parallel X_u \parallel \zeta \parallel iaux_{zk} \parallel iaux_{pub})$. If verification passes, then Π_U^1 is correct, otherwise abort.

5.1.2. Details of SPK_2

SPK$_2$ can also be implemented using standard discrete logarithm proof techniques.

1. **(Commitment.)** The issuer/trust authority randomly selects $t_1, t_2, t_3, t_4 \in_R \mathbb{Z}_q^*$ and computes:

   $$C_1 = h_1^{t_1} h_2^{t_2}, C_2 = (\zeta \cdot Y_2^{s''})^{t_3} \cdot u_0^{H_2(epoch\|k) \cdot t_4}.$$

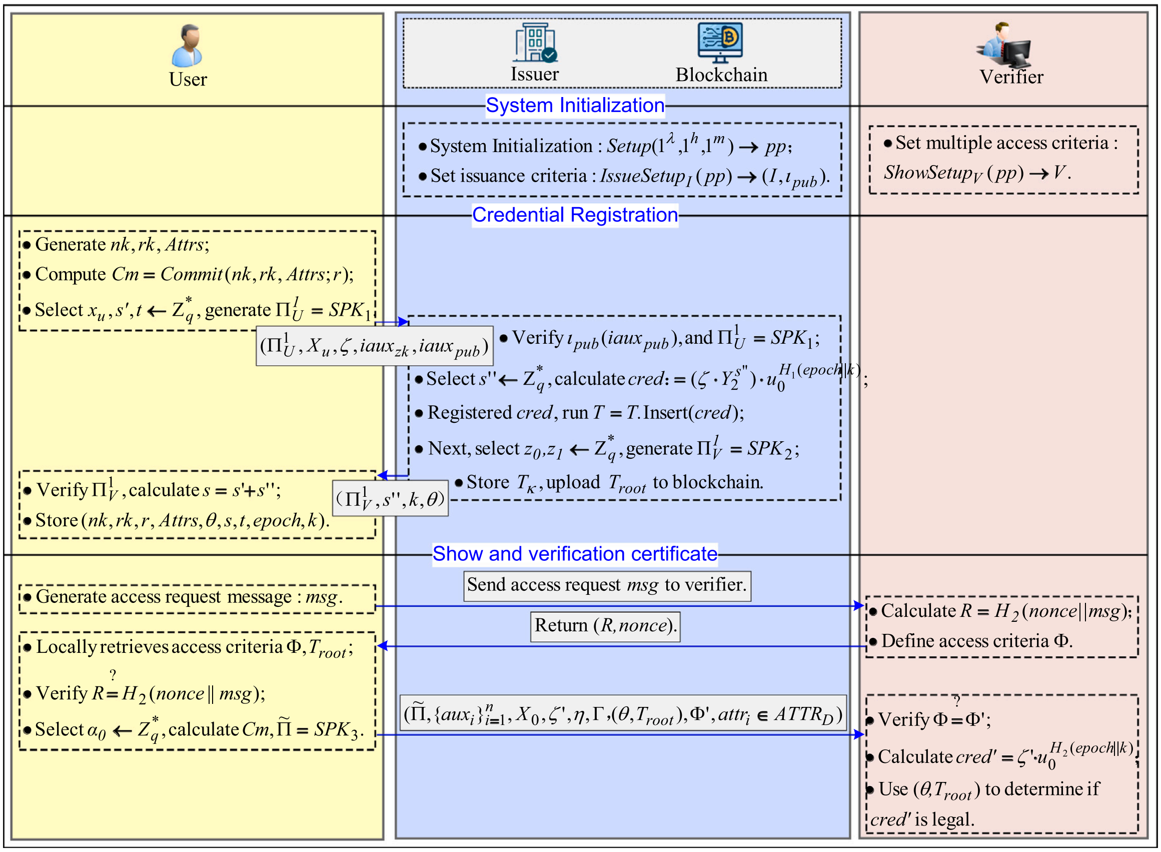

User
Issuer
Blockchain
Verifier
System Initialization
• System Initialization : $Setup(1^\lambda,1^h,1^m) \to pp$;
• Set issuance criteria : $IssueSetup_I(pp) \to (I, t_{pub})$.
• Set multiple access criteria : $ShowSetup_V(pp) \to V$.
Credential Registration
• Generate $nk, rk, Attrs$;
• Compute $Cm = Commit(nk, rk, Attrs; r)$;
• Select $x_u, s', t \leftarrow Z_q^*$, generate $\Pi_U^I = SPK_1$
$(\Pi_U^1, X_u, \zeta, iaux_{zk}, iaux_{pub})$
• Verify $t_{pub}(iaux_{pub})$, and $\Pi_U^1 = SPK_1$;
• Select $s'' \leftarrow Z_q^*$, calculate $cred := (\zeta \cdot Y_2^{s''}) \cdot u_0^{H_1(epoch\|k)}$;
• Registered $cred$, run $T = T.\mathrm{Insert}(cred)$;
• Next, select $z_0, z_1 \leftarrow Z_q^*$, generate $\Pi_V^I = SPK_2$;
• Store T_κ, upload T_{root} to blockchain.
• Verify Π_V^1, calculate $s = s' + s''$;
$(\Pi_V^1, s'', k, \theta)$
• Store $(nk, rk, r, Attrs, \theta, s, t, epoch, k)$.
Show and verification certificate
• Generate access request message : msg.
Send access request msg to verifier.
• Calculate $R = H_2(nonce\|msg)$;
• Define access criteria Φ.
Return $(R, nonce)$.
• Locally retrieves access criteria Φ, T_{root};
• Verify $R \overset{?}{=} H_2(nonce\|msg)$;
• Select $\alpha_0 \leftarrow Z_q^*$, calculate $Cm, \widetilde{\Pi} = SPK_3$.
$(\widetilde{\Pi}, \{aux_i\}_{i=1}^n, X_0, \zeta', \eta, \Gamma, (\theta, T_{root}), \Phi', attr_i \in ATTR_D)$
• Verify $\Phi \overset{?}{=} \Phi'$;
• Calculate $cred' = \zeta' \cdot u_0^{H_2(epoch\|k)}$;
• Use (θ, T_{root}) to determine if $cred'$ is legal.

Fig. 2. System Flowchart.

2. **(Challenge.)** The scheme uses non-interactive zero-knowledge proof, where $\mathcal{U}$ generates challenge c:

$$c = H(C_1 \parallel C_2 \parallel Y_u \parallel \mathcal{Z} \parallel s'' \parallel k).$$

3. **(Proof.)** The issuer generates proof Π_V^1 by computing $C_1' = t_1 - c \cdot y_1, C_2' = t_2 - c \cdot y_2, C_3' = t_3 - c \cdot z_1, C_4' = t_4 - c \cdot z_0$. The proof $\Pi_V^1 = (c, C_1', C_2', C_3', C_4')$, $\mathcal{I}$ sends (Π_V^1, s'', k) to user.
4. **(Verify.)**$\mathcal{U}$ computes, $\mathfrak{C}_1 = Y_u^c h_1^{C_1'} h_2^{C_2'}, \mathfrak{C}_2 = \mathcal{Z}^c(\zeta \cdot Y_2^{s''})^{C_3'} \cdot u_0^{H_2(epoch\|k) \cdot C_4'}$, and verify: $c \overset{?}{=} H(\mathfrak{C}_1 \parallel \mathfrak{C}_2 \parallel Y_u \parallel Z \parallel s'' \parallel k)$. If verification passes, then Π_V^1 is correct, otherwise abort.

5.1.3. Details of SPK_3

The construction of SPK_3 includes zero-knowledge proof and range proof. We divide SPK_3 into two parts SPK_{3A} and SPK_{3B}. The specific details are as follows:

$$SPK_{3A}\left\{\begin{array}{cl} & \left(nk, rk, \alpha_0, x_u, s, t, n_j, \rho_1\right): \\ & X_0 = g_0^{\alpha_0}\gamma^{H_1(\theta)} \\ \wedge & \zeta' = Y_1^{x_u} Y_2^{s} \cdot Cm^t \\ \wedge & \mathcal{N} = g_1^{n_j} g_2^{\rho_1} \\ \wedge & \frac{\tilde{u}}{\eta} = \eta^{rk}\eta^{n_j} \\ \wedge & \frac{\tilde{u}^R \cdot u_0}{\Gamma} = u_0^{-nk} u_0^{-n_j} u_0^{-x_u} \Gamma^{nk}\Gamma^{n_j} \end{array}\right\}\left(aux_i, X_0, \zeta', \eta, \Gamma, T_{root}\right),$$

$$SPK_{3B}\{(n_j, \rho_1) : \mathcal{N} = g_1^{n_j} g_2^{\rho_1} \wedge 0 \le n_j < k\}(m).$$

SPK$_{3B}$ is instantiated as a simple range proof, which will be discussed later. Next, we demonstrate how to implement SPK$_{3A}$.

1. **(Commitment.)** $\mathcal{U}$ randomly selects $\varrho_1, \varrho_2, \mathfrak{t}_3, \mathfrak{t}_4, \mathfrak{t}_5, \mathfrak{t}_6, \mathfrak{n}_7, \mathfrak{n}_8 \in_R \mathbb{Z}_q^n$ and computes:

$$A_1 = g_0^{\mathfrak{t}_3} y^{H_1(\theta)}, A_2 = Y_1^{\mathfrak{t}_4} Y_2^{\mathfrak{t}_5} Cm^{\mathfrak{t}_6}, A_3 = g_1^{\mathfrak{n}_7} g_2^{\mathfrak{n}_8},$$
$$A_4 = \eta^{\varrho_2}\eta^{\mathfrak{n}_7}, A_5 = u_0^{-\varrho_1} u_0^{-\mathfrak{n}_7} u_0^{-t_4} \Gamma^{\varrho_1}\Gamma^{\mathfrak{n}_7}.$$

2. **(Challenge.)** Using non-interactive zero-knowledge proof, the user generates challenge c:

$$c = H(A_1 \parallel A_2 \parallel A_3 \parallel A_4 \parallel A_5 \parallel X_0 \parallel \zeta' \parallel \eta \parallel \Gamma \parallel T_{root} \parallel aux_i).$$

3. **(Proof.)** $\mathcal{U}$ generates proof $\tilde{\Pi}$ by computing:

$$A_1' = \mathfrak{t}_3 - c \cdot \alpha_0, A_2' = \mathfrak{t}_4 - c \cdot x_w, A_3' = \mathfrak{t}_5 - c \cdot s,$$
$$A_4' = \mathfrak{t}_6 - c \cdot t, A_5' = \mathfrak{n}_7 - c \cdot n_j, A_6' = \mathfrak{n}_8 - c \cdot \rho_1,$$

$$A_7' = \varrho_2 - c \cdot rk, A_8' = \varrho_1 - c \cdot nk.$$

The proof $\tilde{\Pi} = (c, A_1', A_2', A_3', A_4', A_5', A_6', A_7', A_8')$, and sends $(\tilde{\Pi}, aux_i, X_0, \zeta', \eta, \Gamma, T_{root})$ to verifier $\mathcal{V}$.
4. **(Verify.)** $\mathcal{V}$ computes:

$$\mathfrak{A}_1 = X_0^c g_0^{A_1'}\gamma^{H_1(\theta)}, \mathfrak{A}_2 = \zeta'^c Y_1^{A_2'} Y_2^{A_3'} Cm^{A_4'},$$
$$\mathfrak{A}_3 = \mathcal{N}^c g_1^{A_5'} g_2^{A_6'}, \mathfrak{A}_4 = \left(\frac{\widetilde{u}}{\eta}\right)^c \eta^{A_7'}\eta^{A_5'},$$

$$\mathfrak{A}_5 = \left[\frac{\widetilde{u}^R \cdot u_0}{\Gamma}\right]^c u_0^{-A_8{'}} u_0^{-A_5{'}} u_0^{-A_2{'}} \Gamma^{A_8{'}} \Gamma^{A_5{'}},$$

and verify: $c \overset{?}{=} H(\mathfrak{A}_1 \parallel \mathfrak{A}_2 \parallel \mathfrak{A}_3 \parallel \mathfrak{A}_4 \parallel \mathfrak{A}_5 \parallel X_0 \parallel \zeta' \parallel \eta \parallel \Gamma \parallel T_{root} \parallel aux_i)$.

In groups of unknown order, range proofs currently widely recognized by academia and industry are based on the square decomposition assumption [43] and n-ary decomposition [40], which can achieve secure and efficient range proofs. However, we note that the range proofs required in authentication protocols always take the form $0 \le n < k$. If we set $k = 2^\kappa$, we can easily construct a simple range proof with complexity $\mathcal{O}(\kappa)$, as shown in Eq. (1):

$$POK_{RANGE}\{(n, r) : C_n = g_0^n g_1^r \wedge 0 \le n < 2^\kappa\}. \qquad (1)$$

In this scheme, we use a Bulletproofs-based instantiation of SPK_{3B}. Here we will briefly describe and provide a detailed proof process. Please refer to the Ref. [29,43].

1. **(Prove.)** First, perform binary decomposition on n, $n = \sum_{i=0}^{k-1} b_i 2^i$, where $b \in \{0,1\}$. Construct vector $\mathbf{a}_L = (b_0, b_1, \dots, b_{k-1}), \mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^k (a_{R,i} = b_i - 1)$. Next, choose blind factor $\alpha, \rho \leftarrow \mathbb{Z}_q, \mathbf{s}_L, \mathbf{s}_R \leftarrow \mathbb{Z}_q^k$, compute the initialization commitment $A = h^{\alpha} \mathbf{g}^{\boldsymbol{a}_L} \mathbf{h}^{\boldsymbol{a}_R}, S = h^{\rho} \mathbf{g}^{\mathbf{s}_L} \mathbf{h}^{\mathbf{s}_R}$. Then, construct a non-interactive proof challenge $y = H\left(A, S, C_n\right), z = H(y, A, S)$ based on Fiat–Shamir and polynomials $\boldsymbol{l}(x) = \boldsymbol{a}_L - z\mathbf{1}^k + \mathbf{s}_L x, \boldsymbol{r}(x) = y^k \circ \left(\boldsymbol{a}_R + z\mathbf{1}^k\right) + \mathbf{s}_R x$, calculate the inner product $t = \langle \boldsymbol{l}(x), \boldsymbol{r}(x) \rangle, \tau_x \leftarrow \mathbb{Z}_p, T = g^t h^{\tau_x}$. The final challenge is $x = H(z, y, T)$, generate response $\boldsymbol{l} = \boldsymbol{l}(x), \boldsymbol{r} = \boldsymbol{r}(x), \hat{t} = \langle \boldsymbol{l}, \boldsymbol{r} \rangle, \tau = \tau_x + x^2 \rho, \mu = \alpha + x\rho$. Finally output the proof $\pi = (A, S, T, \hat{t}, \tau, \mu, \boldsymbol{l}, \boldsymbol{r})$.
2. **(Verify.)** Upon receiving the commitment C_n, proof π, recalculate the challenge $y = H\left(A, S, C_n\right), z = H(y, A, S), x = H(z, y, T)$. Next, compute offset value $\delta_y = \left\langle y^k, z\mathbf{1}^k + z^2 2^k \right\rangle$, and reconstruct the commitment $P = A \cdot S^x \cdot h^{-\mu} \cdot \mathbf{g}^{z\mathbf{1}^k} \cdot \left(\boldsymbol{h}'\right)^{z\mathbf{1}^k + z^2 2^k}$, where $\boldsymbol{h}' = \boldsymbol{h} \circ y^k$. Then, verify inner product $g^{\hat{t}} h^{\tau} \overset{?}{=} T \cdot C_n^{Z^2} \cdot g^{\delta_y}$. If passed, accept, otherwise, reject.

5.2. Theoretical security analysis

5.2.1. Proof of Game1

Theorem 1. *The scheme is unforgeable if the DLP and DDH assumptions hold.*

Proof. Suppose that the adversary $\mathcal{A}_1$ forges the credential with the non-negligible probability ϵ, we construct reduction algorithm $\mathcal{B}$ to solve the DLP or CDH problem with the non-negligible advantage $\epsilon - negl$. The reduction algorithm $\mathcal{B}$ embeds the group parameter tuple $\mathcal{I} = (\mathcal{G}, \mathcal{G}^a, \mathcal{G}^b)$ into the problem instance, $\mathcal{B}$ can control and program the random oracle, and simulates the whole system:

Setup. Challenger C_1 run system initialization algorithm $Setup(1^{\lambda}, 1^h, 1^m)$ generate pp, send pp to simulator $\mathcal{S}$. C_1 save issuer private key $isk = (y_1, y_2)$.

Query. In this phase, $\mathcal{A}_1$ query random Oracle $\mathcal{H}\_Query$, $Query_2$, and $Query_3$, C_1 random response and recording.

$\mathcal{H}\_Query$: The adversary $\mathcal{A}_1$ can query the random oracle $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$. Before any hash query, $\mathcal{S}$ will prepare three empty hash lists $\mathcal{L}_{1,2,3}$, and define the query number size as $q_{H_1}, q_{H_2}, q_{H_3}$ to record the query response.

$\mathcal{H}_1 - Query$: Before $\mathcal{H}_1$ query, $\mathcal{S}$ randomly selected $i_1^* \in \left[1, q_{H_1}\right]$, the input attribute $attr_i$, $\mathcal{S}$ record of all the queries in the list $\mathcal{L}_1$, and make a response. If $i = i_1^*$, $\mathcal{S}$ return values in the list, otherwise $\mathcal{S}$ generated $\mathcal{H}_1(attr_i)$, records $(i, attr_i, \mathcal{H}_1(attr_i))$ in $\mathcal{L}_1$.

$\mathcal{H}_2 - Query$: Before the $\mathcal{H}_2$ query, $\mathcal{S}$ randomly selects $i_2^* \in \left[1, q_{H_2}\right]$, after entering each user time period $epoch_i$, and the maximum number of credentials to be initialized k_i, $\mathcal{S}$ records all queries in the list $\mathcal{L}_2$, and responds. If $i = i_2^*$, $\mathcal{S}$ returns the value in the list, otherwise $\mathcal{S}$ generates $\mathcal{H}_2(epoch \parallel k)$ with the following Eq. (2):

$$\mathcal{H}_2\left(epoch_i \parallel k_i\right) = \begin{cases} w^*, i = i_2^* \\ w_i, \text{ otherwise} \end{cases} . \tag{2}$$

Then, $\mathcal{S}$ record $\left(i, \left(\text{ epoch }_i \parallel k_i\right), \mathcal{H}_2\left(epoch_i \parallel k_i\right)\right)$ in the list $\mathcal{L}_2$.

$\mathcal{H}_3 - Query$: Before $\mathcal{H}_3$ queries, $\mathcal{S}$ randomly selected $i_3^* \in \left[1, q_{H_3}\right]$, the input random $nonce_i$ and message msg_i, $\mathcal{S}$ record of all the queries in the list $\mathcal{L}_3$, and respond. If $i = i_3^*$, $\mathcal{S}$ return values in the list, otherwise $\mathcal{S}$ generated $\mathcal{H}_2(nonce \parallel msg)$ in the following Eq. (3):

$$\mathcal{H}_2\left(nonce_i \parallel msg_i\right) = \begin{cases} r^*, i = i_3^* \\ r_i, \text{ otherwise} \end{cases} . \tag{3}$$

Then, $\mathcal{S}$ record $\left(i, \left(nonce_i \parallel msg_i\right), \mathcal{H}_2\left(nonce_i \parallel msg_i\right)\right)$ in the list $\mathcal{L}_3$, where oracle $\mathcal{H}_2$ and $\mathcal{H}_3$ share a hash function. $Query_2$: In this phase, the adversary $\mathcal{A}_1$ forges parameters $(ctx^*, nk^*, rk^*, Attrs^*)$, selects the random blind factor $r^* \in \mathbb{Z}_q^*$, query $\mathcal{H}_1 - Query$, and generates $Cm^* = Commit\,(nk^*, rk^*, Attrs^*; r^*)$. Next, choose $x_u^*, s'^*, t^* \leftarrow \mathbb{Z}_q^*$, calculate $\Pi_U^{1^*}$:

$$\Pi_U^{1^*} = SPK_1^* \left\{ \begin{array}{l} \left(x_u^*, s'^*, t^*, r^*, nk^*, rk^*, Attrs^*\right) : \\ X_u^* = g_1^{x_u^*} g_2^{s'^*} \\ \Lambda \;\; \zeta^* = (\mathcal{G}^a)^{x_u^*} \left(\mathcal{G}^b\right)^{s'^*} \cdot Cm^{*t^*} \\ \Lambda \;\; \iota_{zk}\left(Attrs^*, iaux_{zk}\right) = 1 \end{array} \right\} \left(X_u^*, \zeta^*, iaux_{zk}, iaux_{pub}\right).$$

Sending $\left(\Pi_U^{1^*}, iaux_{zk}, iaux_{pub}\right)$ to the issuer, $\mathcal{S}$ checks $\iota_{pub}\left(iaux_{pub}\right)$ and validates $\Pi_U^{1^*}$, aborts if it fails, otherwise it selects a random number $s''^* \in \mathbb{Z}_q^*$ and performs $\mathcal{H}_2 - Query$. Embed tuple $\mathcal{I} = (\mathcal{G}, \mathcal{G}^a, \mathcal{G}^b)$, register $cred^* := (\zeta^* \cdot (\mathcal{G}^b)^{s''^*}) \cdot u_0^{w^*}$, generate the forged Merkle tree T^*, update the root node to T_{root}^*, select $z_0^*, z_1^* \leftarrow \mathbb{Z}_q^*$, Calculate $\Pi_V^{1^*} = SPK_2^* \left\{ \left(z_0^*, z_1^*, a, b\right) : Y_u^* = \mathcal{G}^a \mathcal{G}^b \wedge \mathcal{Z}^* = (\zeta^* \cdot (\mathcal{G}^b)^{s''^*})^{z_1^*} \cdot u_0^{w^* \cdot z_0^*} \right\}$ $(Y_u^*, s''^*, k^*, \mathcal{Z}^*)$, send $(\Pi_V^{1^*}, s''^*, k^*, \theta^*)$ to adversary $\mathcal{A}_1$, $\mathcal{A}_1$ calculate $s^* = s'^* + s''^*$ and save to local.

$Query_3$: In this phase $\mathcal{A}_1$ to show the proof, using zero knowledge simulator $\mathcal{S}$, run algorithm $ShowCred$ forged $token^*$ and $VerifyShow$ interact. Adversary $\mathcal{A}_1$ forges the message msg^* requesting access to $\mathcal{S}$. $\mathcal{S}$ selects $nonce^*$, conducts $\mathcal{H}_3 - Query$ query, calculates r^*, and returns it to adversary $\mathcal{A}_1$. Adversary $\mathcal{H}_3 - Query$ hash verification, if by selecting public attribute $attr_i^* \in ATTR_D^*$, the secret attribute is $attr_j^* \notin ATTR_D^*$, calculate $Cm^* = \text{Commit}\left(nk^*, rk^*, attr_j^* \notin ATTR_D^*; r^*\right)$, select $n_j^* \left(0 \le n_j^* < k^*\right)$, $\alpha_0^* \leftarrow \mathbb{Z}_q^*$, generate $\widetilde{\Pi}^*$, send $(\widetilde{\Pi}^*, \left\{aux_i\right\}_{i=1}^{i=n}, \left(\theta^*, T_{root}^*\right), \Phi', attr_i^* \in ATTR_D^*)$ to $\mathcal{S}$.

Forgery. Adversary $\mathcal{A}_1$ outputs the forged certificate $cred^*$ and the corresponding authentication path θ^*, which meets the condition that $cred^*$ was not generated through legal issuance. $\mathcal{S}$ running algorithm VerifyShow, $VerifyShow\left(pp, V, \left(cred^*, T_{root}^*\right), \widetilde{\Pi}^*, \left\{aux_i\right\}_{i=1}^{i=i}\right) = 1$.

Then, requery $\mathcal{H}_3$ by rewinding technique to obtain r^*, modify the new challenge to $c \neq c'$, compute the response and output $\widetilde{\Pi}'^*$ to extract witness $w^* = \left(x_u^*, s^*, t^*, r^*, nk^*, rk^*, attr_j^* \notin ATTR_D^*\right)$, separate from the witness $\zeta'^* = (\mathcal{G}^a)^{x_u^*} (\mathcal{G}^b)^{s^*} \cdot Cm^{*t^*} = (\mathcal{G}^{ab})^{x_u^* \cdot s^*} \cdot Cm^{*t^*}$. According to the above proof, if the forgery credential $cred^*$ and the corresponding authentication path θ^* make it difficult to compute $\mathcal{G}^{ab}$ on $\mathbb{G}$, the probability that adversary $\mathcal{A}_1$ will successfully forge a credential for the first time is ϵ, and the probability of a single retry is about ϵ^2. By the universal bifurcation Lemma, since adversary $\mathcal{A}_1$ performs q_{H_3} queries. The probability of success is ϵ^2 / q_{H_3}, then the advantage of simulator to break CDH hard problem successfully is $\epsilon^2 / q_{H_3} - negl$.

5.2.2. Proof of Game2

Theorem 2. *The Scheme is anonymity and unlinkability if the CDH assumption hold.*

Proof. Suppose that the adversary $\mathcal{A}_2$ distinguishes credentials with a non-negligible advantage ϵ, and construct a reduction algorithm $\mathcal{B}$ to solve the DDH problem with a non-negligible advantage $\epsilon - negl$. The reduction algorithm $\mathcal{B}$ embedded the group parameter tuple $\mathcal{Q} = (\mathcal{G}, \mathcal{G}^a, \mathcal{G}^b, \mathcal{G}^c)$ into the DDH problem instance, and the adversary $\mathcal{A}_2$ determined whether $c = ab$ or random, and simulated the whole process:

Setup. Same with the initialization of *Game 1*.

Query. Adversary $\mathcal{A}_2$ can continue to query issuance and show, but cannot query revocation or presentation of challenge credentials. At the same time also can query $\mathcal{H}_1 - Query$.

Challenge. Adversary $\mathcal{A}_2$ submits two attribute sets $Attrs_0^*$ and $Attrs_1^*$, that satisfy the same access policy to challenger C_2. Since the parameter related to the attribute set in zero-knowledge is ζ'. The challenger C_2 calls the simulator $\mathcal{S}$ to simulate the SPK and prove the embedding group parameter tuple $\mathcal{Q} = (\mathcal{G}, \mathcal{G}^a, \mathcal{G}^b, \mathcal{G}^c)$, randomly select $a, b \leftarrow \mathbb{Z}_q^*$, and calculate $\zeta_1'^*$. Select $c \leftarrow \mathbb{Z}_q^*$ calculate $\zeta_2'^*$. Next,

Table 3
Average times of cryptographic and Merkle tree operations.

Symbol	Definition	secp256k1 (128-bit security)		BLS12-381 (128-bit security)	
		100 s/Leaves	1000 s/Leaves	100 s/Leaves	1000 s/Leaves
T_{bp}	Bilinear pairing operation time	–	–	0.9162 ms	0.9466 ms
T_h	Hash computation time	0.0003 ms	0.0000 ms	0.0001 ms	0.0000 ms
T_{ep}	Exponentiation time in group G	0.0211 ms	0.0314 ms	0.2606 ms	0.2677 ms
T_{mp-ec}	Elliptic curve point multiplication time	0.0254 ms	0.0234 ms	$\mathbb{G}_1$:0.3958 ms $\mathbb{G}_2$:0.8140 ms	$\mathbb{G}_1$:0.2686 ms $\mathbb{G}_2$:0.8009 ms
T_{add-ec}	Elliptic curve point addition time	0.0462 ms	0.0530 ms	$\mathbb{G}_1$:0.0007 ms $\mathbb{G}_2$:0.0018 ms	$\mathbb{G}_1$:0.0006 ms $\mathbb{G}_2$:0.0018 ms
T_κ^G	Generation algorithm of tree T_κ	0.0025 ms	0.0024 ms	0.0029 ms	0.0023 ms
T_κ^V	Verification algorithm of tree T_κ	0.0004 ms	0.0002 ms	0.0020 ms	0.0002 ms
T_κ^U	Update algorithm of tree T_κ	0.0002 ms	0.0002 ms	0.0003 ms	0.0003 ms

Table 4
Computation and communication cost analysis.

Algorithms	Parameter	Phase	Computation cost	Communication cost
$Setup$	pp	–	$2T_{ep}$	$(13+m)\|\mathbb{G}\|$
$IssueSetup_I$	(I, ι_{pub})	–	–	–
$ShowSetup_V$	V	–	–	–
$IssueReq_U$	Cm	–	$(3+m)T_{ep}+mT_h+3T_{mp-ec}$	$\|\mathbb{G}\|$
	Π_U^1	Proof	$(16+m)T_{ep}+3T_{mp-ec}$	$2\|\mathbb{G}\|+5\|\mathbb{Z}_q\|$
		Verify	$7T_{ep}$	–
$IssueGrant_I$	$cred$	–	$1T_{ep}+2T_{mp-ec}+1T_h$	–
	T_κ	–	T_κ^G	–
	Π_V^1	Proof	$8T_{ep}+1T_h+3T_{mp-ec}$	$2\|\mathbb{G}\|+6\|\mathbb{Z}_q\|$
		Verify	$6T_{ep}$	–
$ShowCred_U$	$\tilde{\Pi}$	Proof	$25T_{ep}$	$5\|\mathbb{G}\|+7\|\mathbb{Z}_q\|$
	$\{aux_i\}_{i=1}^n$	–	–	$\mathbb{i}\|\mathbb{Z}_q\|$
$VerifyShow_V$	–	Verify	$26T_{ep}+T_\kappa^V$	–
$RevokeCred$	T_κ'	–	T_κ^U	–

Note*: $\mathbb{i}$ is the number of access criteria defined per verifier.

simulator $\mathcal{S}$ selects $b \leftarrow \{0,1\}$, and uses $Attrs_b{}^*$ to generate the credential display $\widetilde{\Pi}_b$. Send $\left(\widetilde{\Pi}_b, \left\{aux_i\right\}_{i=1}^{i=i}, \left(\theta, T_{root}\right), \Phi', attr_i \in ATTR_{\mathcal{D}}\right)$ to adversary $\mathcal{A}_2$.

Guess. $\mathcal{A}_2$ guesses b' from the output $\widetilde{\Pi}_b$, and the advantage is defined as: $\left|\Pr\left[b' = b\right] - \frac{1}{2}\right|$.

According to the above proof, if two attribute sets satisfying the same access policy are submitted $Attrs_0^*$, $Attrs_1^*$. It is difficult for $\widetilde{\Pi}_b$ to distinguish between $\left(\mathcal{G}^a, \mathcal{G}^b, \mathcal{G}^{a\cdot nk+b\cdot rk+ab\cdot r}\right)$ and $\left(\mathcal{G}^a, \mathcal{G}^b, \mathcal{G}^{a\cdot nk+b\cdot rk+c\cdot r}\right)$ on $\mathbb{G}$, then adversary $\mathcal{A}_2$ succeeds in distinguishing credentials with non-negligible probability ϵ/q_{H_1}. Then the advantage of the simulator $\mathcal{S}$ to break the DDH hard problem successfully is $\epsilon/q_{H_1} - negl$.

Note that even if the underlying Merkle path remains the same for repeated authentications, the simulator ensures that each credential presentation is randomized. Therefore, the adversary's advantage does not increase by observing identical path values, which remain computationally indistinguishable across sessions.

Theorem 3. *The Scheme is attribute Privacy if the CDH assumption hold. Similar anonymity, but in view of the properties rather than identity.*

6. Performance analysis

6.1. Experimental setup

The scheme is based on *AMD Ryzen9 7945HX processor*, *Rust 1.75* and *Ubuntu 22.04 LTS* environment, and the error is controlled within 5%. The test program is written in *Rust* and performs benchmark evaluations on SHA-256 hacks, elliptic curve operations, and Merkle tree operations with the 128-bit security secp256k1, BLS12-381, and sha2 libraries. The experiment measured the average time of 100 and 1000 operations (as shown in Table 3). All tests were compiled based on *–release* optimization to ensure accurate and reliable performance results.

6.2. Algorithm computation and communication cost analysis

Table 4 shows the computational cost and communication cost of the proposed algorithm in the scheme. The algorithm includes 8 algorithms as follows. $Setup, IssueSetup_I, ShowSetup_V, IssueReq_U,$ $IssueGrant_I, ShowCred_U,$

$VerifyShow_V$ and $RevokeCred$. The computational cost increases linearly with the number of attributes m. We compared the single user in Table 4 cases for each verifier $\beth$ access criteria general computation and communication costs. Respectively, $(94 + 2\ m)T_{ep} + (m + 2)T_h + 11T_{mp-ec} + T_\kappa^G + T_\kappa^V$ and $(22+m)|\mathbb{G}| + (18+\beth)|\mathbb{Z}_q|$. The cost of a single algorithm is shown in Table 4 below:

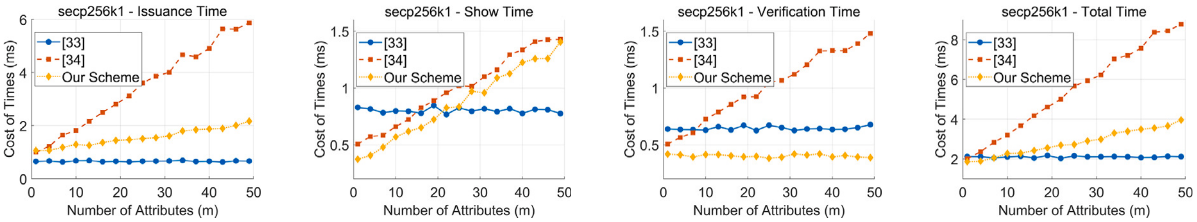
6.3. Computation and communication cost comparison

In Table 1 of Section 2, we have compared the functions of the existing schemes [19,29–31,33–35]. The scheme [32–34] satisfies the k-times period anonymous authentication function. Since the scheme [32] is constructed based on bilinear pairing. Here, we compare the scheme [33,34] with the proposed scheme in the computation cost processes of issuance, show and verification. Using the lightweight curve secp256k1 environment, as shown in Table 5 and Fig. 3. In Table 1, the scheme [33] does not support the attribute selection disclosure function and does not increase with the increase of the number of attributes m. Therefore, the data results in Fig. 3 show that our scheme is better than the scheme [33] when the number of attributes m is small. Throughout the entire process, the overall performance was superior to the scheme [34]. Finally, the data results show that our scheme is superior to the existing schemes under the condition of similar functions.

In addition to the above experimental comparison, we also added the proposed scheme to test the computational overhead under two different curve environments, BLS12-381 supporting bilinear pairing

Table 5
Computation cost comparison.

Scheme	Computation cost (ms)		
	Credential issuance	Certificate showing	Authentication credentials
[33]	$15T_{ep} + 10T_{mp-ec} + 2T_{add-ec}$	$31T_{ep} + 6T_{mp-ec} + T_h$	$20T_{ep} + 9T_{mp-ec} + T_h$
[34]	$(5\ m + 40)T_{ep} + (3\ m + 4)T_h$	$(m + 22)T_{ep} + T_h$	$(m + 23)T_{ep}$
Our Scheme	$(m + 35)T_{ep} + (m + 2)T_h + 11T_{mp-ec} + T_\kappa^G$	$(16 + m)T_{ep} + mT_h$	$19T_{ep} + T_h + T_\kappa^V$



(a) (b) (c) (d)

Fig. 3. Computation cost comparison.

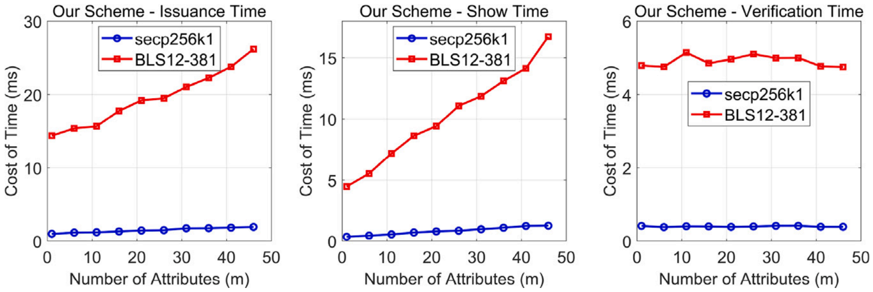
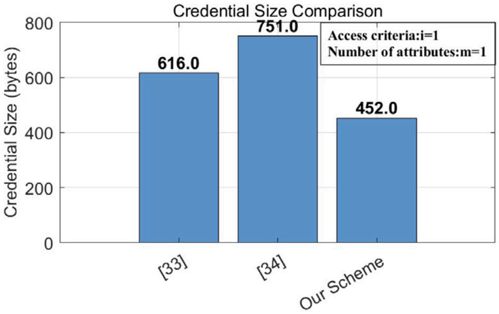


Fig. 4. Computation cost comparison of different curves.



Scheme	Communication Cost
[33]	$8\lvert\mathbb{G}\rvert + 11\lvert\mathbb{Z}_q\rvert$
[34]	$(m + 14)\lvert\mathbb{G}\rvert + 8\lvert\mathbb{Z}_q\rvert$
Our Scheme	$4\lvert\mathbb{G}\rvert + (9 + \mathbb{i})\lvert\mathbb{Z}_q\rvert$

Fig. 5. Communication cost comparison.

and lightweight curve secp256k1, as shown in Fig. 4. The experimental results show that the scheme has more advantages under lightweight curve. It is suggested to apply the proposed scheme under curve secp256k1.

Finally, the communication cost of the existing scheme [33,34] is compared and calculated based on the size of the data to be transmitted during the anonymous certificate display process. We test the communication efficiency on curve secp256k1, where the group element and integer size of curve secp256k1 are $|\mathbb{G}| = 264bits = 33bytes, |\mathbb{Z}_q| = 256bits = 32bytes$, respectively. In the test, it is assumed that the access criterion $\beth$ is 1, and the number of user attributes is 1. The communication costs of the schemes [33,34] are respectively $8|\mathbb{G}| + 11|\mathbb{Z}_q|$, and $(m + 14)|\mathbb{G}| + 8|\mathbb{Z}_q|$. The parameters that our scheme needs to transmit for presentation are $(\tilde{\Pi}, \{aux_i\}_{i=1}^n, X_0, \zeta', \eta, \Gamma, \theta)$, where $\tilde{\Pi} = (c, A_1', A_2', A_3', A_4', A_5', A_6', A_7', A_8')$. Therefore, the total communication cost during the transmission process is $4|\mathbb{G}| + (9 + \beth)|\mathbb{Z}_q|$. As shown in Fig. 5.

7. Conclusion

In this paper, we propose a k-times periodic anonymous authentication that does not require the issuer to hold a key and supports the access criteria. Compared with other existing k-Times periodic anonymous authentication schemes, the proposed scheme not only has lower computational cost, but also eliminates the need for the issuer to hold the issuing information or the user key, and only needs to upload the root path of the Merkle tree to the blockchain or public panel, which ensures that the subsequent authentication can still be carried out even in the case of the failure of the issuing center. In terms of security, it satisfies a series of DAC security properties, including anonymity, unlinkability, unforgeability and attribute privacy. The limitation of current schemes is that they rely on classical cryptography, which cannot resist quantum computing attacks. To address this challenge, we plan to integrate quantum-resistant cryptographic frameworks, such as lattice-based signature, coding cryptography, or multivariate polynomial encryption in future research to construct periodic k-times authentication schemes with post-quantum security.

CRediT authorship contribution statement

Hongyan Di: Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Yinghui Zhang:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition. **Ziqi Zhang:** Writing – original draft, Formal analysis, Data curation. **Yibo Pang:** Project administration, Formal analysis, Data curation. **Rui Guo:** Writing – original draft, Methodology, Formal analysis. **Yangguang Tian:** Writing – original draft, Project administration, Methodology, Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] K.Y. Lam, C.H. Chi, Identity in the internet-of-things (IoT): New challenges and opportunities, in: Information and Communications Security, 2016, pp. 18–26.

[2] K. Shafique, B.A. Khawaja, F. Sabir, S. Qazi, M. Mustaqim, Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios, IEEE Access 8 (2020) 23022–23040, http://dx.doi.org/10.1109/ACCESS.2020.2970118.

[3] L. Ante, C. Fischer, E. Strehle, A bibliometric review of research on digital identity: Research streams, influential works and future research paths, J. Manuf. Syst. 62 (2022) 523–538, http://dx.doi.org/10.1016/j.jmsy.2022.01.005.

[4] M.A. Olivero, A. Bertolino, F.J.D. Mayo, M.J.E. Cuaresma, I. Matteucci, Digital persona portrayal: Identifying pluridentity vulnerabilities in digital life, J. Inf. Secur. Appl. 52 (2020) 102492, URL: https://api.semanticscholar.org/CorpusID:215881538.

[5] M.S. Ferdous, F. Chowdhury, M.O. Alassafi, In search of self-sovereign identity leveraging blockchain technology, IEEE Access 7 (2019) 103059–103079, http://dx.doi.org/10.1109/ACCESS.2019.2931173.

[6] A. Shabtai, Y. Elovici, L. Rokach, List of data breaches and cyber attacks in 2023. Media report. IT governance, 2023, URL: https://www.itgovernance.co.uk/blog/list-of-data-breaches-andcyber-attacks-in-2023.

[7] P.C. Bartolomeu, E. Vieira, S.M. Hosseini, J. Ferreira, Self-sovereign identity: Use-cases, technologies, and challenges for industrial IoT, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2019, pp. 1173–1180, http://dx.doi.org/10.1109/ETFA.2019.8869262.

[8] European Union, Regulation (EU) 2016/679 of the European parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (general data protection regulation), 2016, [Online] Available: URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

[9] A. Mühle, A. Grüner, T. Gayvoronskaya, C. Meinel, A survey on essential components of a self-sovereign identity, Comput. Sci. Rev. 30 (2018) 80–86, http://dx.doi.org/10.1016/j.cosrev.2018.10.002.

[10] European Union, Regulation (EU) 2024/1183 of the European parliament and of the council of 5 June 2024 on European digital identity wallets, 2024, URL: https://eur-lex.europa.eu/eli/reg/2024/1183/oj. (Accessed 13 October 2024).

[11] D. Chaum, Security without identification: transaction systems to make big brother obsolete, Commun. ACM 28 (1985) 1030–1044, http://dx.doi.org/10.1145/4372.4373.

[12] D. Chaum, Showing credentials without identification. Signatures transferred between unconditionally unlinkable pseudonyms, in: Proc. of a Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology—EUROCRYPT '85, 1986, pp. 241–244.

[13] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in: Advances in Cryptology — EUROCRYPT 2001, 2001, pp. 93–118.

[14] C. Garman, M. Green, I. Miers, Decentralized anonymous credentials, in: Proceedings of the 21st NDSS, 2014, URL: https://www.ndss-symposium.org/ndss2014/decentralized-anonymous-credentials.

[15] D. Derler, C. Hanser, D. Slamanig, A new approach to efficient revocable attribute-based anonymous credentials, in: Cryptography and Coding, 2015, pp. 57–74.

[16] T. Bui, T. Aura, Application of public ledgers to revocation in distributed access control, in: Information and Communications Security, 2018, pp. 781–792.

[17] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, G. Danezis, Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers, in: 26th Annual Network and Distributed System Security Symposium, NDSS, 2019, URL: https://arxiv.org/pdf/1802.07344.

[18] H. Halpin, Nym credentials: Privacy-preserving decentralized identity with blockchains, in: 2020 Crypto Valley Conference on Blockchain Technology, CVCBT, 2020, pp. 56–67, http://dx.doi.org/10.1109/CVCBT50464.2020.00010.

[19] H. Cui, M. Whitty, A. Miyaji, Z. Li, A blockchain-based digital identity management system via decentralized anonymous credentials, in: Proceedings of the 6th ACM International Symposium on Blockchain and Secure Critical Infrastructure, 2025, pp. 1–11, http://dx.doi.org/10.1145/3659463.3660027.

[20] C. Lin, D. He, H. Zhang, L. Shao, X. Huang, Privacy-enhancing decentralized anonymous credential in smart grids, Comput. Stand. Interfaces 75 (2021) 103505, http://dx.doi.org/10.1016/j.csi.2020.103505.

[21] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, W. He, An efficient decentralized key management mechanism for VANET with blockchain, IEEE Trans. Veh. Technol. 69 (2020) 5836–5849, http://dx.doi.org/10.1109/TVT.2020.2972923.

[22] J. Zhang, J. Cui, H. Zhong, I. Bolodurina, L. Liu, Intelligent drone-assisted anonymous authentication and key agreement for 5G/B5G vehicular ad-hoc networks, IEEE Trans. Netw. Sci. Eng. 8 (2021) 2982–2994, http://dx.doi.org/10.1109/TNSE.2020.3029784.

[23] D. Liu, H. Wu, C. Huang, J. Ni, X. Shen, Blockchain-based credential management for anonymous authentication in SAGVN, IEEE J. Sel. Areas Commun. 40 (2022) 3104–3116, http://dx.doi.org/10.1109/JSAC.2022.3196091.

[24] D. Liu, H. Wu, J. Ni, X. Shen, Efficient and anonymous authentication with succinct multi-subscription credential in SAGVN, IEEE Trans. Intell. Transp. Syst. 23 (2022) 2863–2873, http://dx.doi.org/10.1109/TITS.2022.3147354.

[25] L. Wei, Y. Zhang, J. Cui, H. Zhong, I. Bolodurina, D. He, A threshold-based full-decentralized authentication and key agreement scheme for VANETs powered by consortium blockchain, IEEE Trans. Mob. Comput. 23 (2024) 12505–12521, http://dx.doi.org/10.1109/TMC.2024.3412106.

[26] M. Zeng, J. Cui, Q. Zhang, H. Zhong, D. He, Efficient revocable cross-domain anonymous authentication scheme for IIoT, IEEE Trans. Inf. Forensics Secur. 20 (2025) 996–1010, http://dx.doi.org/10.1109/TIFS.2024.3523198.

[27] I. Teranishi, J. Furukawa, K. Sako, K-times anonymous authentication (extended abstract), in: Advances in Cryptology - ASIACRYPT 2004, 2004, pp. 308–322.

[28] L. Nguyen, R. Safavi-Naini, Dynamic k-times anonymous authentication, in: Applied Cryptography and Network Security, 2005, pp. 318–333.

[29] M.H. Au, W. Susilo, Y. Mu, Constant-size dynamic k-TAA, in: Security and Cryptography for Networks, 2006, pp. 111–125.

[30] U. Chaterjee, D. Mukhopadhyay, R.S. Chakraborty, 3PAA: A private PUF protocol for anonymous authentication, IEEE Trans. Inf. Forensics Secur. 16 (2021) 756–769, http://dx.doi.org/10.1109/TIFS.2020.3021917.

[31] J. Huang, W. Susilo, F. Guo, G. Wu, Z. Zhao, Q. Huang, An anonymous authentication system for pay-as-you-go cloud computing**, IEEE Trans. Dependable Secur. Comput. 19 (2) (2022) 1280–1291, http://dx.doi.org/10.1109/TDSC.2020.3007633.

[32] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, M. Meyerovich, How to win the clonewars: efficient periodic n-times anonymous authentication, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, 2006, pp. 201–210, http://dx.doi.org/10.1145/1180405.1180431.

[33] B. Lian, G. Chen, M. Ma, J. Li, Periodic K -times anonymous authentication with efficient revocation of violator's credential, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 543–557, http://dx.doi.org/10.1109/TIFS.2014.2386658.

[34] Y. Yang, W. Xue, J. Sun, G. Yang, Y. Li, H. Hwa Pang, R.H. Deng, PkT-SIN: A secure communication protocol for space information networks with periodic k-time anonymous authentication, IEEE Trans. Inf. Forensics Secur. (2024) 6097–6112, http://dx.doi.org/10.1109/TIFS.2024.3409070.

[35] C. Wiraatmaja, S. Kasahara, Scalable anonymous authentication scheme based on zero-knowledge set-membership proof, Distrib. Ledger Technol. 4 (2025) http://dx.doi.org/10.1145/3676285.

[36] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G.N. Rothblum, R.D. Rothblum, D. Wichs, Fiat-Shamir: from practice to theory, 2019, http://dx.doi.org/10.1145/3313276.3316380.

[37] J. Camenisch, M. Stadler, Efficient group signature schemes for large groups, in: Advances in Cryptology — CRYPTO '97, 1997, pp. 410–424.

[38] M. Rosenberg, J. White, C. Garman, I. Miers, zk-creds: Flexible anonymous credentials from zkSNARKs and existing identity infrastructure, in: 2023 IEEE Symposium on Security and Privacy, SP, 2023, pp. 790–808, http://dx.doi.org/10.1109/SP46215.2023.10179430.

[39] Y. Dodis, A. Yampolskiy, A verifiable random function with short proofs and keys, 2004, URL: https://eprint.iacr.org/2004/310. Cryptology ePrint Archive, Paper 2004/310.

[40] J. Groth, On the size of pairing-based non-interactive arguments, in: Advances in Cryptology – EUROCRYPT 2016, 2016, pp. 305–326.

[41] V. Shoup, Sequences of games: a tool for taming complexity in security proofs, IACR Cryptol. EPrint Arch. (2004) 332, URL: http://eprint.iacr.org/2004/332.

[42] M. Bellare, P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, in: Proceedings of the 1st ACM Conference on Computer and Communications Security, 1993, pp. 62–73, http://dx.doi.org/10.1145/168588.168596.

[43] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell, Bulletproofs: Short proofs for confidential transactions and more, in: 2018 IEEE Symposium on Security and Privacy, SP, 2018, pp. 315–334, http://dx.doi.org/10.1109/SP.2018.00020.

Hongyan Di is currently studying for a master’s degree in Cyberspace and Information Security from Xi’an University of Posts and Telecommunications. Her research interests include cross-domain authentication and digital signature security.

Yinghui Zhang received his Ph.D. degree in Cryptography from Xidian University, China, in 2013. He is a professor at School of Cyberspace Security, National Engineering Research Center for Secured Wireless (NERCSW), Xi’an University of Posts & Telecommunications. He was a research fellow at School of Information System, Singapore Management University. He has published over 100 research articles in ACM CSUR, IEEE TDSC, IEEE TCC, Computer Networks, etc. He served on the program committee of several conferences and the editorial member of several international journals in information security. His research interests include public key cryptography, cloud security, and wireless network security.

Ziqi Zhang is currently studying for a master’s degree in Cyberspace and Information Security from Xi’an University of Posts and Telecommunications. Her research interests include digital signature security and its applications.

Yibo Pang received the B.S. degree in Information Security from the School of Cyberspace Security, Xi’an University of Posts and Telecommunications, Xi’an, China, in 2020, and the M.S. degree in Cyberspace Security from the School of Cyberspace Security, Xi’an University of Posts and Telecommunications, Xi’an, China, in 2023. He is currently pursuing a PhD at Xi’an University of Posts and Telecommunications. His research interests include multimedia security and privacy.

Rui Guo is an associate professor and master’s supervisor at Xi’an ’an University of Posts and Telecommunications. He has presided over a total of 9 scientific research projects, including those funded by the National Natural Science Foundation of China, the Key Research and Development Program of Shaanxi Province, and the Basic Research Program of Shaanxi Province. As a major participant, he has participated in and completed more than 10 projects, such as the National Key Research and Development Plan and the National Natural Science Foundation of China. As the first author, I have published over 20 academic papers, among which 12 are indexed by SCI (including 1 TOP 1% ESI highly cited paper).

Dr. Yangguang Tian received his Ph.D. degree in applied cryptography from the University of Wollongong, Australia. After Ph.D., he did post-docs at School of Information System, Singapore Management University, and iTrust, Singapore University of Technology and Design. Before Surrey, he was a research-based assistant professor at Osaka University, Japan. He is currently a lecturer at the University of Surrey, UK. His research interests include applied cryptography, network security, blockchain technologies, and privacy-preserving technologies. Dr. Tian’s recent research works have been published in the cybersecurity-related international conferences and journals, such as USENIX’24, AsiaCCS’24, IEEE TIFS’23, IEEE TDSC’24, etc.