Distributed Multi-Attribute Anonymous Certificate Management Scheme with Fine-Grained Revocation for Unmanned Aerial Vehicles

Qingyang Zhang, Bin Ren, Hong Zhong, Fengqun Wang, Mingwei Zeng, and Jie Cui

***Abstract*—In unmanned aerial vehicle (UAV) scenarios, the execution of specific flight missions requires anonymous certificates containing multiple attributes as proof of authorization. However, existing certificate- management schemes are ineffective in achieving an optimal trade-off between verification overhead and attribute-level revocation. First, most existing schemes bind multiple attributes to a single certificate but typically lack the capability of fine-grained revocation at the individual attribute level. Second, most existing schemes rely on centralized certificate authorities, necessitating UAVs to apply for certificates from multiple regions separately when performing cross-regional access. This scenario increases the certificate management burden. To address these challenges, this paper proposes a distributed multi-attribute anonymous certificate management scheme for UAVs. First, the proposed scheme integrates redactable signatures and dynamic accumulators, enabling the selective disclosure and fine-grained revocation of attributes within a single certificate. Second, the proposed scheme utilizes a distributed key-generation mechanism, enabling decentralized certificate issuance and secure management.**

***Index Terms*—Anonymous Authentication, UAVs, redactable signature, distributed, traceable, fine-grained revocation.**

I. Introduction

As key components of low-altitude intelligent networks [1], [2], unmanned aerial vehicles (UAVs) are responsible for core functions such as perception, communication, and service coordination. They drive the development of intelligence and networking in low-altitude spaces. Considering their broad application potential across various mission scenarios, the ”Low-altitude Intelligent Networked Technology System White Paper” explicitly states the necessity of accelerating UAV deployment in regions such as emergency response, low-altitude logistics, and ecological monitoring [2], [3]. This enables UAVs to be fully integrated into all aspects of public production and daily life, thereby becoming powerful drivers of low-altitude economic growth [4].

Fig. 1 illustrates the typical architecture for UAV multi-service cross-region access. For this task, UAVs must subscribe to services provided by multiple service providers based on their specific requirements, such as high-precision maps, data analytics, and network communication services [5], [6]. Each service provider manages their region, and during task execution, the UAV must submit valid credentials to the service provider of the corresponding region to gain access [6]. To improve availability and reduce access latency, each service provider deploys the same service across the edge nodes in multiple physical regions. UAVs can traverse different regions during task execution because of their high mobility. Therefore, a UAV can access the edge nodes in any nearby region based on its current location to obtain the required services [5]. To ensure security, service eligibility is defined as a set of attributes signed by service providers to generate anonymous certificates for UAVs. First, UAVs must register with the Air Traffic Control (ATC) center to obtain a legitimate identity. Second, when requesting services, UAVs must provide valid certificates to the roadside base stations for authentication purposes. If the certificates are valid, then the UAV can access the corresponding services. Finally, if a dispute arises between the UAV and the service provider, the ATC center will intervene to arbitrate and reveal the true identity of the malicious UAV.

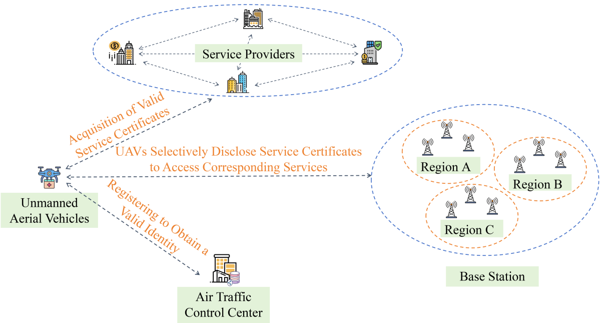

Service Providers
Acquisition of Valid Service Certificates
UAVs Selectively Disclose Service Certificates to Access Corresponding Services
Unmanned Aerial Vehicles
Registering to Obtain a Valid Identity
Air Traffic Control Center
Region A
Region B
Region C
Base Station

Fig. 1: UAV Multi-Service Cross-Region Access Framework.

In the multi-service cross-region access system for UAVs, fine-grained revocation enables issuers to precisely control the UAVs' service permissions, which not only prevents unauthorized access but also ensures stable system operation. However, existing revocation mechanisms are often too coarse-grained, and once specific permission is revoked, the UAV may be unable to access all related services (e.g., navigation), potentially leading to UAV malfunction. Therefore, this area still faces numerous challenges.

(1) **It is difficult to efficiently verify the validity of certificates in distributed multi-service cross-region scenarios.** UAVs can subscribe to a large number of services to perform various tasks. Therefore, reducing the burden on certificate management to ensure a timely service provision

Q. Zhang, B. Ren, H. Zhong, F. Wang, M. Zeng and J. Cui are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China, and the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn). (*Corresponding authors: Hong Zhong; Fengqun Wang.*)

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3668965

for UAVs is crucial. Traditional centralized certificate generation systems typically rely on trusted centers to issue signing keys [7], [8], [9]. Once the trusted center is compromised and keys are leaked to the attacker, the attacker can forge a valid certificate [10], [11]. Furthermore, in the existing schemes, UAVs must register with and maintain the corresponding certificates for each regional service provider to access their services [12], [13]. When UAVs move across different regions, they must frequently update their authentication certificates, thereby increasing the operational complexity. Therefore, utilizing distributed systems to build service provider alliances and enhance the fault tolerance of a system is an important issue that cannot be ignored.

(2) **It is difficult to achieve fine-grained revocation of UAV service permissions while efficiently verifying the validity of multi-attribute certificates.** In existing schemes, a single certificate can only validate the legitimacy of one attribute [7], [14], leading to computational and communication overheads that increase linearly with the number of disclosed attributes, making such schemes unsuitable for multi-attribute authentication scenarios. Multi-attribute anonymous certificate systems have been studied extensively to enhance certificate verification efficiency [15], [16]. However, in these schemes, the revocation authority directly adds user identity information to the revocation list. When a user requests a service, to ensure the unlinkability of the user's certificate, the verifier must scan the entire revocation list through bilinear mappings to verify if the user's identity is in the revocation list [17], [18]. If it does exist, the user is denied access to all services. Therefore, existing schemes revoke all UAV permissions simultaneously during revocation [19], [20], [21], failing to support attribute-based revocation (ABR) while efficiently verifying the validity of multi-attribute certificates.

A. Contribution

This study proactively address the aforementioned challenges by proposing a novel cryptographic system called a distributed fine-grained redactable anonymous certificate. Our contributions are summarized as follows:

- **Efficient distributed multi-service cross-region access:** Based on a distributed key generation (DKG) protocol, we design a distributed anonymous authentication algorithm, which enables the effective authorization of UAV permissions in distributed scenarios. The proposed algorithm innovatively adopts the DKG protocol, in which multiple publishers spontaneously ally to jointly generate public and private keys for a certificate signature, avoiding reliance on a single trusted center. By establishing a unified distributed trust infrastructure, UAVs are required to apply for a certificate only once and can subsequently authenticate and access services across multiple regions managed by different service providers. This design eliminates the need for frequent certificate applications and renewals, reduces the operational complexity of UAVs, and ensures stable and seamless service provisioning.
- **An effective trade-off between fine-grained revocation and verification efficiency:** We developed a fine-grained revocable algorithm that resolves the inherent trade-off in existing schemes between verification efficiency and attribute-level revocation granularity. This algorithm integrates redactable signature schemes (RSS) with dynamic accumulator techniques, enabling efficient verification of multi-attribute certificate validity while supporting the fine-grained revocation of UAV service permissions, and allows non-revoked attributes to remain usable. By employing redactable signatures, multiple attributes are aggregated into a concise, constant-size certificate, and the UAV's identity information is bound to the services to be revoked. When a UAV requests access to a particular service, specific attribute information in the certificate can be selectively disclosed or concealed as required, and a dynamic accumulator nonmembership proof for that service is provided to the roadside base station. A roadside base station can use this to verify that the UAV's service access permission has been revoked.
- **Security Proof and Implementation:** We formally define and analyze the security properties of our scheme, demonstrating the security requirements for UAV multi-service cross-region access. Furthermore, we implemented a scheme based on the BLS12381 curve with a 128-bit security level and compared it with several state-of-the-art schemes. The results show that our scheme offers significant advantages for both computational and communication overheads.

B. Organization

Section II summarizes relevant prior studies. Section III reviews the foundational concepts used in the construction. Section IV introduces the system model and security definitions. Subsequently, Section V details the proposed scheme. Section VI provides security proof of the proposed scheme. The experimental results are presented in Section VII. Finally, Section VIII presents the conclusions drawn.

II. Related work

Recent years have seen notable progress in certificate issuance, presentation, and revocation.

Certificate issuance authorizes UAVs with valid certificates to enable controlled access to required services. Traditional certificate generation systems typically rely on a single issuer holding a signing key. Once the issuer is compromised and the key is leaked to an attacker, the attacker can forge arbitrary valid certificates [8]. Therefore, traditional service certificate generation systems depend on a centralized authorization authority, which introduces security risks such as single points of failure [24]. To address the single-point failure issue associated with relying on a single certificate issuer, research on threshold certificate systems has been widely explored [25]. Schemes [22] and [26] employ Shamir's secret sharing and linear secret sharing, respectively, to distribute the power of certificate issuance across multiple entities. This effectively reduces the threat of single points of failure and significantly

TABLE I: Functionality Comparison

Scheme	Threshold Issuing	Identity Tracing	Re-randomization	Attribute Disclosure Proof	Revocation
BBS [22]	✓	✗	✗	✗	✗
SPS-EQ [8]	✓	✗	✓	TAM-Sign	✗
DTACB [12]	✓	✗	✓	ACC	✗
TCM [15]	✓	✗	✓	ZKP	IBR
TDAC [23]	✓	✗	✗	✗	✗
TABC [17]	✓	✓	✓	URS	IBR
PS [19]	✗	✓	✓	PS	IBR
Our scheme	✓	✓	✓	RSS	ABR

enhances the system's fault tolerance. However, their schemes still rely on a trusted center to issue signing keys, making them difficult to adapt to distributed environments.

Certificate presentation ensures that UAVs perform tasks within the authorized scope, preventing misuse or abuse of system resources. Most existing authentication methods are designed based on cryptographic techniques such as zero-knowledge proofs and bilinear pairings, which provide a solid foundation for certificate display [16], [27]. However, these schemes issue separate certificates for each attribute [28], [29], resulting in computation and communication overhead that grows linearly with the number of disclosed attributes, making them unsuitable for direct application in multi-attribute certificate authentication environments. Jia et al. [15] introduced a threshold certificate management scheme based on non-interactive zero-knowledge proofs, which enables multiple authority organizations to collaborate in managing certificates. However, this scheme incurs high overhead when using zero-knowledge proofs to verify the legitimacy of multi-attribute certificates. Schemes [12] and [13] employ cryptographic accumulators and aggregatable Merkle signatures, respectively, to effectively display multiple certificates from different issuers, achieving efficient batch verification of certificates. However, in their schemes, when displaying certificates, the attributes to be disclosed must be aggregated into a signature and provided to the verifier, resulting in lower efficiency.

Revocation of permissions effectively blocks unauthorized access by revoked UAVs, ensuring the validity of certificates and the security of the system. As the demand for multi-service cross-region access increases, more researchers have begun to introduce revocation mechanisms in anonymous certificates to effectively manage user permissions [10]. Mir et al. [23] presented a delegable anonymous certificate authentication scheme built upon threshold predicate encryption, enabling effective delegation of user permissions. However, this scheme omits embedding user identities or user-related secrets within the certificate, thus preventing effective user revocation. Schemes [17] and [18] use revocation lists to store the identity information of revoked users, achieving identity-based revocation (IBR). However, to ensure unlinkability during certificate access, these schemes require scanning the entire revocation list via bilinear mappings during verification, which incurs significant computational overhead. Sun et al. [19] designed a revocable anonymous certificate authentication scheme based on cryptographic accumulators, enabling effective user certificate revocation. However, this scheme directly incorporates user identifiers into the revocation list, leading to the simultaneous revocation of all user permissions during revocation, thus failing to achieve fine-grained permission revocation. Liu et al. [20] introduced an anonymous certificate authentication scheme built upon threshold signatures that limits the number of times a user can present a certificate within a certain time frame. However, like Sun et al.'s scheme [19], this scheme revokes all of the user's permissions at once, preventing fine-grained permission revocation.

In summary, to address the difficulty of existing schemes in effectively verifying certificate validity in distributed multi-service cross-domain scenarios, the proposed scheme is built upon a DKG protocol, enabling efficient authorization of UAV permissions in distributed settings. To tackle the challenge faced by existing schemes in balancing verification overhead and attribute-level revocation, this paper designs a fine-grained revocable algorithm that achieves efficient verification of multi-attribute certificate validity while supporting fine-grained revocation of UAV service permissions. Table I compares the functional differences between these schemes and the proposed scheme, highlighting the strengths of our design with respect to decentralization, traceability, and attribute-level revocation.

III. Preliminaries

A. Accumulator Based on Bilinear-Maps

Cryptographic accumulators [30], [31] can aggregate all elements in a set and efficiently provide a non-membership proof for any element, i.e., to verify whether an element exists in the set. We use the Bilinear-Map (BM) accumulator proposed by Srinivasan et al. [32] as our revocation blacklist. Let χ represent the region of the accumulator. Typically, a Bilinear-Map accumulator consists of the following algorithms:

- $Acc.Setup(1^{\lambda}) \rightarrow pp$: This algorithm initializes with a security parameter $1^{\lambda}$and generates the public parameters pp. For simplicity, subsequent algorithms assume pp as an implicit input.
- $Acc.Commit(D) \rightarrow A_D$: Given a set $D \subseteq \chi$, this algorithm returns the accumulator value A_D.
- $Acc.Add(A_D, D, I) \rightarrow A_{D \cup I}$: Given an accumulator value acc_D, the set D it represents, and a disjoint subset $I \subseteq \chi$ with $I \cap D = \emptyset$, this algorithm returns the updated accumulator value $A_{D \cup I}$.
- $Acc.NonMemProve(D, I) \rightarrow (\overline{\pi}_I, c_I)$: This non-membership proof algorithm takes two sets D and $I \subseteq \chi$ that satisfy $I \nsubseteq \chi$, it outputs a commitment c_I and the associated proof $\overline{\pi}_I$.

- $Acc.NonMemVerify(A_D, c_I, \overline{\pi}_I) \rightarrow (0/1)$: This verification algorithm checks whether all elements in I are not in X. It takes as input A_D, the commitment c_I, and the proof $\overline{\pi}_I$, and returns 1 if $I \nsubseteq D$, and 0 otherwise.

B. Redactable Signatures

An redactable signatures scheme [14], [33], defined by a set of algorithms $(KeyGen, Sign, Verify, Redact)$, is used to generate certificates for UAV attributes.

- $RSS.KeyGen(1^\lambda) \rightarrow (pk, sk)$: On input the security parameter 1^λ, this key generation algorithm produces a key pair (pk, sk), where pk is employed for verifying signatures and sk is used to generate them.
- $RSS.Sign(sk, M) \rightarrow \sigma$: Taking as input the private key sk and a message M composed of n blocks $\{m_i\}_{i=1}^n$, this algorithm outputs a signature σ over the entire message M.
- $RSS.Verify(pk, M_I, \sigma_I) \rightarrow (0/1)$: This verification algorithm takes the public key pk, a partial message M_I, and its corresponding signature $\sigma_I$as inputs. It outputs a bit 0 or 1, where 1 represents successful verification.
- $RSS.Redact(pk, M, I, \sigma) \rightarrow \sigma_I$: Provided with pk, the full message M, a subset of indices $I \subset [1, n]$, and a signature σ, this algorithm generates a new signature σ_I over the selected message blocks $\{m_i\}_{i \in I}$.

C. Hardness Assumptions

- Definition 1 (**Discrete logarithm (DL) assumption**): Let $\mathbb{G}$ denote a cyclic group with prime order p, and let g be its generator. The DL assumption holds that, for any randomly selected $x \in \mathbb{Z}_p^*$, it is computationally infeasible for any probabilistic polynomial-time (PPT) adversary to recover x from the tuple $(g, g^x) \in G^2$ with more than a negligible probability.
- Definition 2 (**Assumption 1 [34]**): Let $(\mathbb{G}_1, \mathbb{G}_2, e, p, g, \tilde{g})$ be a bilinear group of Type-III, where g and $\tilde{g}$ are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. For an arbitrary length message m, the oracle $\mathcal{O}(m)$ is defined as follows: randomly select an element $\sigma_1 \in \mathbb{G}_1$ and output a pair $(\sigma_1, \sigma_1^{a+H(m)b})$, where a and b are random elements from $\mathbb{Z}_p$. For any PPT adversary, even if it can query $\mathcal{O}(m)$ and $(\tilde{g}, \mathbb{G}_1, \mathbb{G}_2)$ arbitrarily many times, it cannot generate a valid pair $(\sigma_1, \sigma_1^{a+H(m^*)b})$ for a new message m^* not previously queried, where $\sigma_1 \neq 1_{\mathbb{G}_1}$.

IV. System And Security Model

A. System Model

As illustrated in Fig. 2, our system architecture involves five categories of participants: multiple issuers (I), the Air Traffic Control center (ATC), Unmanned Aerial Vehicles (UAVs), and verifiers. Additionally, it incorporates a public ledger that facilitates the publication and retrieval of UAV revocation records. The detailed responsibilities of each participant are outlined below:

1) **ATC**: The ATC, acting as a trusted government authority or court, is responsible for maintaining the system's public keys and global revocation status. It is the only entity capable of tracking the true identity of malicious UAVs.
2) $\boldsymbol{I_i}$: Issuers generate signature keys using a DKG protocol to issue certificates to UAVs and support threshold-based bulk revocation of UAV services.
3) **UAV**: UAVs need to register with the ATC and request certificates from issuers. Additionally, UAVs may anonymously disclose a subset of attributes to verifiers.
4) **Verifier**: The verifier honestly verifies the correctness of the certificate showing token generated by the UAV, but remains curious about the hidden attribute information and the real identity of the UAV.
5) **Ledger**: Ledger can be a blockchain, where the ATC writes the revocation list into the blockchain using smart contracts.

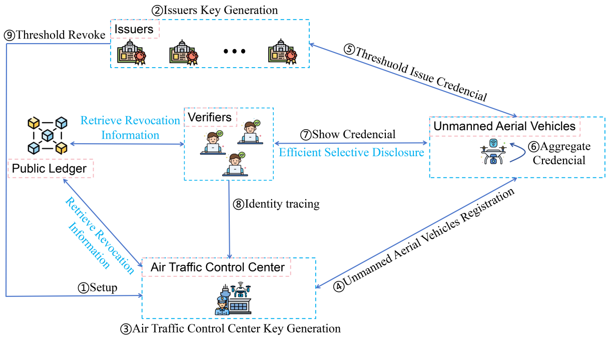

②Issuers Key Generation
⑨Threshold Revoke
Issuers
⑤Threshuold Issue Credencial
Retrieve Revocation Information
Verifiers
⑦Show Credencial
Efficient Selective Disclosure
Unmanned Aerial Vehicles
⑥Aggregate Credencial
Public Ledger
⑧Identity tracing
Retrieve Revocation Information
④Unmanned Aerial Vehicles Registration
Air Traffic Control Center
①Setup
③Air Traffic Control Center Key Generation

Fig. 2: System model.

B. High-Level Workflows

The high-level workflow consists of the following phases:

①**System Initialization:** The issuers and the ATC collaboratively perform the system initialization to generate the system's public parameters.

②**Issuer Key Generation:** Each issuer executes a DKG protocol to generate their public and private key pairs.

③**ATC Key Generation:** The ATC generates its own public and private key pair and initializes the revocation list. Finally, the ATC writes the public parameters required by the system to the blockchain.

④**UAV Registration:** The UAV submits a registration request to the ATC. Upon receiving the request, the ATC issues a pseudonym certificate to the UAV and preloads the UAV's pseudonym list for identity tracking purposes.

⑤**Certificate Issuance:** After receiving the pseudonym certificate, the UAV submits requests to n_I issuers. Each online issuer generates a partial service certificate for the UAV.

⑥**Certificate Aggregation:** Upon receiving partial service certificates from at least t_I issuers, the UAV aggregates them into a complete service certificate.

⑦**Show certificate:** When a certificate needs to be presented, the UAV computes a certificate presentation token and randomly generates a pseudonym in real-time, embedding it into the certificate. The UAV then signs the certificate presentation token using the pseudonym and reveals a subset of its attributes to the verifier. Additionally, the UAV provides non-membership proofs for the revealed attribute subset to demonstrate that its access rights have not been revoked.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3668965

⑧**UAV Identity Tracking:** When the real identity of a malicious UAV needs to be traced, the ATC can reveal the UAV's true identity by consulting the pseudonym list.

⑨**Threshold Revocation:** The ATC updates the global revocation list to include a UAV's revoked permissions only when at least t_T issuers agree to revoke the UAV's partial permissions. Moreover, the scheme supports batch revocation.

C. Formal Definition

Table II summarizes the symbols used in the system and their corresponding descriptions. The formal definitions of the algorithms are provided below. Since Ledger is publicly available to all participants, it is assumed to be an input parameter in the following algorithm definitions.

- **Setup**$(1^\lambda, n_I, t_I, n_A, t_A, q) \rightarrow pp$: This initialization algorithm is jointly executed by the I_i and ATC. By invoking the Acc.Setup algorithm, the public parameters pp are generated. Specifically, given the 1^λ, the issuer's threshold value (n_I, t_I), the revocation threshold value (n_A, t_A), and the maximum number of attributes q for UAVs, The algorithm outputs the system parameters pp.
- **IssuerKeygen**$(pp) \rightarrow \{isk_i, ipk_i\}_{i=1}^{n_I}$: This key generation algorithm is executed by each issuer I_i. By executing the RSS.KeyGen algorithm with the system parameters pp as input, the algorithm generates a private key isk_i and public key ipk_i for each issuer, where $i \in [1, n_I]$.
- **ATCKeyGen**$(pp) \rightarrow (ask, apk)$: This algorithm is executed by the ATC. It takes the system parameters pp as input and outputs the private key ask and the public key apk for the ATC, and initializes the revocation list st and the verification key pk. Finally, the ATC writes the public parameters (pp, pk, apk, st) to the blockchain.
- $<$ **UApply**$(id, upk, pp) \leftrightarrow$ **URegister**$(ask, apk, pp) >\rightarrow pseucred$: This pseudonym registration algorithm is interactively called by the UAV and the ATC. To obtain a legitimate identity, the UAV inputs the identity id, public key upk, and system parameters pp. The ATC inputs the private key ask, the public key apk, and the system parameters pp. If successful, the algorithm outputs the UAV's pseudonym certificate $pseucred$.
- **PseuGen**$(pseucred, k^*, i^*, pp) \rightarrow pseu_{k^*,i^*}$: This pseudon- ym generation algorithm is called by either the UAV or the ATC. It takes the pseudonym certificate $pseucred$, index k^*, time slot i^*, and system parameters pp as input and outputs a pseudonym $pseu_{k^*,i^*}$.
- $<$ **CredObtion**$(id, upk, Sig, \vec{A}, pp) \leftrightarrow$ **CredIssue**$(isk_i, ipk_i, pp) >\rightarrow (cred_i, uk_i)$: This certificate issuance algorithm is an interactive process between UAV and I_i. The I_i generates a service certificate for each UAV by invoking the RSS.Sign algorithm. To obtain a partial service certificates from I_i, the UAV inputs the identity id, public key upk, signature Sig, attribute set $\vec{A}$, and system parameters pp. The I_i the private key isk_i, public key ipk_i, and system parameters pp. If successful, the algorithm outputs the partial service certificate $cred_i$ and the partially updatable key uk_i for the UAV.
- **CredAgg**$(\{cred_i, uk_i\}_{i\in\mathcal{T}}, pp) \rightarrow (cred, uk)$: This certificate aggregation algorithm is called by the UAV. It takes the system parameters pp, t_I partial service certificates $\{cred_i\}_{i\in\mathcal{T}}$, and t_I partial update keys $\{uk_i\}_{i\in\mathcal{T}}$, where $\mathcal{T} \subset [1, n_I]$, as input. The algorithm outputs a complete service certificate $cred$ and a complete update key uk.
- $<$ **CredShow**$(usk, uk, \vec{A}, I, pseu_{k^*,i^*}, cred, \pi_y, pp) \leftrightarrow$ **CredVerify**$(pk, W_{i^*}, pp) >\rightarrow (0/1)$: This certificate presentation algorithm is an interactive process between UAV and verifier. In the certificate presentation phase: (1) the UAV generates a random pseudonym by itself and embeds it into the attribute certificate, while constructing a zero-knowledge proof of the pseudonym's validity; (2) the UAV, according to the actual service requirements, invokes the redactable signature algorithm RSS.Redact to selectively disclose or hide attributes in the certificate; (3) the UAV executes the non-membership proof generation algorithm Acc.NonMemProve to demonstrate that the disclosed attributes have not been revoked. In the certificate verification phase: (1) the verifier checks the validity of the UAV's pseudonym; (2) if the pseudonym is verified, the verifier invokes the RSS.Verify algorithm to further verify the validity of the attribute certificate, ensuring that the UAV possesses the corresponding access rights; (3) the verifier invokes the Acc.NonMemVerify algorithm to verify the non-membership proof, confirming that the relevant attributes have not been revoked.
- **Trace**$(PseuL, pseu, i) \rightarrow (id/\perp)$: This UAV tracing algorithm is executed by the ATC. If the pseudonym $pseu$ appears within a valid signature, the algorithm outputs identity id corresponding to the UAV that holds the pseudonym $pseu$.
- **Revoke**$\big(\{ipk_i, isk_i, id, S, st\}_{I_i, i\in t_A}, pp\big) \rightarrow st'$: This revocation algorithm is jointly executed by T_A issuers and the ATC. The algorithm takes as input each issuer's key pair (ipk_i, isk_i), the UAV identity id, the attribute set S to be revoked, and the system parameters pp, and invokes the Acc.Add algorithm to generate an updated accumulator value. If successful, the algorithm outputs an updated revocation list st' and stores it in Ledger.

TABLE II: Symbols Used and Their Descriptions

Symbol	Description
q	Maximum number of attributes of the UAV
st	Revocation list
k	Number of attributes disclosed by the UAV
i^*	time interval $(i^* = T(time))$
$\mathcal{R}$	the set of identifiers of permissions to be revoked
I	the attribute set disclosed by the UAV
S	the attribute set of revoked UAV permissions
id	UAV identity
n_I/t_I	number/threshold of issuers
$f(x)$	a polynomial of degree $t_I - 1$
$\mathcal{T}$	a subset of $[1, n_I]$ with $\vert\mathcal{T}\vert = t_I$
D	a subset of $[1, n_I]$ with $\vert D\vert = t_A$
$pseucred$	Pseudonymous certificate of the UAV
$pseu_{k^*,i^*}$	the k^*-th pseudonym of the UAV
$\{m_j\}_{j=1}^{q}$	Attribute set of the UAV
usk/upk	UAV private/public key
isk_i/ipk_i	Issuer private/public key
ask/apk	ATC private/public key
$cred_i/cred$	Partial/aggregated certificate
H:$\{0,1\}^* \rightarrow \mathbb{Z}_p$	a collision resistant hash function
H_1:$\{0,1\}^* \rightarrow \mathbb{G}_1$	a collision resistant hash function

D. Security Model

The objective of the proposed scheme is to satisfy the following security requirements:

Definition 1 (Correctness): For any chosen disclosure strategy $\{m_j\}_{j\in\mathcal{D}}$, the display token tok of the service certificate $cred$ is always verifiable, where $cred$ is the aggregated certificate of the attribute set $\{m_j\}_{j=1}^{q}$ generated by t_I honest issuers, i.e.,

$$Pr\left[\begin{array}{c|l} \langle\text{CredShow} & \text{Setup}(1^\lambda, n_I, t_I, n_A, t_A, q) \rightarrow pp; \\ (usk, uk, \vec{A} & \text{IssuerKeyGen}(pp) \rightarrow \\ , I, pseu_{k^*,i^*}, & \{isk_i, ipk_i\}_{i=1}^{n_I}; \\ cred, \pi_y, & \langle\text{CredObtion}(id, upk, Sig, \vec{A}, pp \\ pp) \leftrightarrow &) \leftrightarrow \text{CredIssue}(isk_i, ipk_i, pp)\rangle \\ \text{CredVerify} & \rightarrow (cred_i, uk_i); \\ (pk, W_{i^*}, pp & \text{CredAgg}\left(\{cred_i, uk_i\}_{i\in\mathcal{T}}, pp\right) \\ )\rangle \rightarrow 1 & \rightarrow (cred, uk). \end{array}\right] = 1$$

Unforgeability ensures that no malicious UAV can forge valid certificates and successfully pass verification without access to the signing key isk. This implies that only honest UAVs can generate valid certificate presentation tokens. For completeness, we adopt the security model established by Liu et al [35].

Definition 2 (Unforgeability): The scheme is considered secure against existential forgery under adaptive chosen-message attacks (EUF-CMA) if every probabilistic polynomial-time adversary $\mathcal{A}$ can succeed in the following game with only negligible probability $\epsilon(\lambda)$.

Game 1: $Unforgeability_{\mathcal{A}}(1^\lambda, n)$

- Setup Phase: To generate the signing key pair (isk, ipk), the challenger runs the $IssuerKeyGen$ algorithm, initializing a counter $d \leftarrow 0$ and a query set $Q_1 \leftarrow \emptyset$. The challenger then sends ipk to the adversary $\mathcal{A}$.
- Query Phase: The adversary $\mathcal{A}$ adaptively selects at most $|Q_1|$ attribute sets M_1,$M_2, ...$,$M_{|Q_1|}$ and issues certificate generation queries to the challenger holding ipk. For each query, the challenger runs the certificate generation algorithm $CredIssue(isk, ipk, pp) \rightarrow (M_j, cred_j)$, and forwards the generated $(M_j, cred_j)$ to $\mathcal{A}$. The challenger then logs $Q_1[d] = (M_j, cred_j)$ and modifies $d \leftarrow d + 1$.
- Output Phase: Once the query phase ends, $\mathcal{A}$ provides a tuple $(M^*, cred^*)$. $\mathcal{A}$ is deemed successful in the above game if $M^* \neq \emptyset$ and the following conditions are satisfied:(1)$\forall j < d$,$\exists m_k \in M^* : m_k \notin M_j$. (2)$CredVerify(pk, W_{i^*}, pp) \rightarrow 1$.

If for any PPT adversary $\mathcal{A}$, even with access to arbitrarily chosen service certificates, the winning advantage of $\mathcal{A}$ in Game 1 remains negligible, the scheme satisfies unforgeability.

Unlinkability [36] guarantees that even if malicious issuers and verifiers cooperate, they are unable to link two different certificates originating from the same UAV. When requesting access to a service, the UAV first generates a random pseudonym and embeds it into its attribute certificate. Subsequently, the UAV independently randomizes both the attribute certificate and the accumulator's non-membership proof using large random integers, preventing the verifier from linking the pseudonym, the attribute certificate, and the accumulator non-membership proof to the same UAV. Here, the unlinkability of the accumulator is based on Scheme [32].

Definition 3 (Unlinkability): Unlinkability is achieved by a scheme if, for every probabilistic polynomial-time adversary $\mathcal{A}$, its advantage in succeeding in the following experiment remains negligible with respect to the security parameter λ.

Game 2: $Unlinkability_{\mathcal{A}}(1^\lambda, n)$

- Setup Phase: The challenger executes the $IssuerKeyGen$ procedure to generate the key pair (isk, ipk), and transmits ipk to the adversary $\mathcal{A}$.
- Phase 1: The adversary $\mathcal{A}$ adaptively selects no more than $|Q_1|$ attribute sets $(M_1, M_2, ..., M_{|Q_1|})$ and requests the generation of certificates for these sets. For each attribute set M_j , the challenger runs the certificate issuance algorithm $CredIssue(isk, ipk, pp) \rightarrow (M_j, cred_j)$ and sends $(M_j, cred_j)$ to $\mathcal{A}$ as the response.
- Challenge Phase:
  1) Upon completing Phase 1, the adversary $\mathcal{A}$ submits two attribute sets $M^0 = \{m_j^{(0)}\}_{j=1}^{n}$ and $M^1 = \{m_j^{(1)}\}_{j=1}^{n}$, ensuring that for all $j \in D$, $m_j^{(0)} = m_j^{(1)}$. $\mathcal{A}$ then sends M^0, M^1, and the index set D to the challenger.
  2) The challenger picks a random bit $c \in \{0, 1\}$ and runs the algorithm $CredIssue(isk, ipk, pp) \rightarrow (M^c, cred^c)$ to generate the certificate $cred^c$. Subsequently, for the attribute set M_D^c, the challenger computes an adaptively redacted certificate $cred_D^c$.
- Phase 2: The adversary $\mathcal{A}$ is allowed to issue further certificate queries, similar to those in Phase 1.
- Guess Phase: Eventually, $\mathcal{A}$ provides a prediction c'. If $c' = c$, i.e., $\mathcal{A}$ correctly guesses the bit c chosen by the challenger, $\mathcal{A}$ wins the game.

If, for every PPT adversary $\mathcal{A}$, the advantage in Game 2 is negligible, i.e., $Adv_{\mathcal{A}}^{Unlinkability}(\lambda) = \left|Pr[c' = c] - \frac{1}{2}\right| \leq \epsilon(\lambda)$ then the scheme satisfies unlinkability.

V. Construction

Inspired by previous certificate-based authentication techniques, including redactable signature and dynamic accumulator technology, we design a distributed anonymous authentication protocol with support for fine-grained revocation. The remainder of this section outlines the construction details of our proposed scheme.

Our proposed scheme offers the following advantages: By leveraging dynamic accumulator [32] technology, the scheme achieves fine-grained and threshold-based batch revocation of UAV permissions. Additionally, by integrating DKG [37], [38] and RSS [33] techniques, the scheme enables efficient selective disclosure of UAV authorizations in distributed environments, thereby ensuring that any UAV with appropriate access rights can seamlessly obtain the corresponding services.

A. Concrete Construction

Setup $(1^\lambda, n_I, t_I, n_A, t_A, q) \rightarrow pp$: Given the input security parameter 1^λ, the issuer threshold (n_I, t_I), the revocation threshold (n_A, t_A), and and the maximum number of attributes q for the UAV. The output is the system parameters $pp =$

$(\mathbb{Z}_p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \mathfrak{g}, \tilde{g}, \mathfrak{h}, h_2, \tilde{u}, \tilde{v}, H, H_1, s, T(), k_1, ..., k_n)$, where $T()$ is a function that takes time values from the system clock as input and outputs the corresponding time interval i, s is a random element in $\mathbb{Z}_p$. $g, \mathfrak{g} \in \mathbb{G}_1$, $\tilde{g}, \mathfrak{h}, h_2 \in \mathbb{G}_2$, $\tilde{u}, \tilde{v} \in \mathbb{G}_T$. The index set $\{k_1, ..., k_n\}$ represents the maximum set of pseudonyms that the UAV can generate.

IssuerKeygen$(pp) \rightarrow \{isk_i, ipk_i\}_{i=1}^{n_I}$: Given the system parameters pp, the issuer executes the distributed key generation (DKG) protocol to generate random polynomials of degree $t_I - 1$, $f_x(z) = x + a_1 z + a_2 z^2 + \cdots + a_{t_I - 1} z^{t_I - 1}$ and $f_{y_i}(z) = y_i + b_1 z + b_2 z^2 + \cdots + b_{t_I - 1} z^{t_I - 1}$ and generates key pairs $\{isk_i, ipk_i\}_{i=1}^{n_I}$ for the n_I issuers. Here $f_x(0) = x$, $f_{y_1}(0) = y_1, \ldots, f_{y_{q+3}}(0) = y_{q+3}$, $x_i = f_x(i)$, $\{y_{i,j} = f_{y_j}(i)\}_{j=1}^{q+3}$. The private key $isk_i = (x_i, y_{(i,1)}, y_{(i,2)}, \ldots, y_{(i,q+3)})$ and the public key $ipk_i = (\tilde{X}_i, \tilde{Y}_{i,1}, \tilde{Y}_{i,2}, ..., \tilde{Y}_{i,q+3})$, where $\tilde{X}_i = \tilde{g}^{x_i}$, $\{\tilde{Y}_{i,j} = \tilde{g}^{y_{i,j}}\}_{j=1}^{q+3}$.

ATCKeyGen$(pp) \rightarrow (ask, apk)$: The ATC generates a public-private key pair and initializes the revocation list as follows: (1)Given system parameters pp, the ATC generates a private key ask, a random element in $\mathbb{Z}_p$, and a public key $apk = (apk_1 = g^{ask}, apk_2 = \tilde{g}^{ask})$. (2)Set the verification key $pk = (\tilde{X}, \{\tilde{Y}_j\}_{j=1}^{q+3})$, where $\tilde{X} = \tilde{g}^x$, $\tilde{Y}_1 = \tilde{g}^{y_1}$,..., $\tilde{Y}_{q+3} = \tilde{g}^{y_{q+3}}$. (3)Initialize the revocation list $st = (acc, \mathcal{R})$, where $\mathcal{R} = \{\}$, and $acc = g^{\prod_{y \in \mathcal{R}}(s+y)}$. (4)Write the system public parameters pp, verification key pk, ATC public key apk, and the revocation list st to the blockchain. (5)Computes $Q_{i^*} = H_1(i^*)$, $W_{i^*} = {Q_{i^*}}^{ask}$, and write to the blockchain before each time slot $i^* = T(time)$ activation.

$<$ **UApply**$(id, upk, pp) \leftrightarrow$ **URegister**$(ask, apk, pp) > \rightarrow pseucred$: The UAV and the ATC issue a pseudonym certificate to the UAV through the following steps:

- UAV Computes Registration Information: The UAV sends its public key upk, a zero-knowledge proof of knowledge $\Pi_1 = ZKSok\{usk : upk = H_1(id)^{usk} = h^{usk}\}$ for usk, and the identifier id, where $upk = H_1(id)^{usk}$.
- ATC Issues the Pseudonym Certificate: (1)If π_1 is invalid, the ATC outputs $\perp$. Otherwise, it randomly selects $\mu \in \mathbb{Z}_p$. (2) It computes $Su = \tilde{g}^{\frac{1}{(ask+\mu)}}$, $Sig = H_1(id)^{ask}$. (3)The ATC sends the certificate $pseucred = (\mu, Su, Sig)$ to the UAV via a secure channel. (4)The tuple $(id, cred)$ is stored in the registration list. (5)Before each time slot activation, the ATC center and the UAV use the same pseudonym generation algorithm to generate the corresponding pseudonyms for the index set $\{k_1, ..., k_n\}$. Then, the ATC center creates the pseudonym list $PseuL$, where each entry corresponds to a UAV and takes the form $\{id, pseu_{i^*,k_1}, ..., pseu_{i^*,k_n}\}$.
- UAV verifies the correctness of the pseudonym certificate issued by the ATC center using the following equations: $e(g^{\mu} \cdot apk_1, Su) = e(g, \tilde{g}), e(Sig, \tilde{g}) = e(H_1(id), apk_2)$.

Remark 1: For completeness, the instantiation of Π_1 is as follows: (1)UAV randomly selects $a \in \mathbb{Z}_p$, computes $R = h^a$, and $c = H(upk, h, R)$. (2)UAV then computes $s = a - c \cdot usk$ and sends (c, s). (3)ATC computes $c' = H(upk, h, h^s \cdot upk^c)$ and verifies whether $c' = c$.

PseuGen$(pseucred, k^*, i^*, pp) \rightarrow pseu_{k^*,i^*}$: The UAV and the ATC generate a pseudonym through the following steps:

- Compute the time variant parameter $Q_{i^*} = H_1(i^*)$.
- Compute $d = H^{k^*}(i^*)$, where H^{k^*} represents the recursive application of H for k^*, forming a hash chain.
- Compute $\mu' = (d - \mu)/2(modp)$, $Pu = Q_{i^*}^{(\mu+\mu')}$, $\hat{P}u = Su^{\mu'}$.
- Output the pseudonym $pseu_{k^*,i^*} = (Pu, \hat{P}u)$.

$<$ **CredObtion**$(id, upk, Sig, \vec{A}, pp) \leftrightarrow$ **CredIssue**$(isk_i, ipk_i, pp) > \rightarrow (cred_i, uk_i)$: The UAV and issuer I_i generate a partial service certificate $cred_i$ for the UAV through the following steps:

- UAV Requests a certificate: The UAV sends the attribute set $\vec{A}$, identifier id, signature Sig, public key upk, and a zero-knowledge proof π_1 of usk's knowledge.
- Issuer I_i Issues Partial certificate: (1) Check whether π_1 is correct. If π_1 is incorrect, terminate the process; otherwise, recompute $h = H_1(id)$. (2) Verify the correctness of the UAV's signature Sig using the following equation: $e(Sig, \tilde{g}) = e(H_1(id), apk_2)$. (3)Compute the partial certificate: $cred_i = (h, \sigma_i = h^{x_i + H(Sig) \cdot y_{i,q+1} + \sum_{j=1}^{q} H(m_j) \cdot y_{i,j}} \cdot upk^{y_{q+2}})$. (4)Send the partial certificate $cred_i$ and the auxiliary key $uk_i = h^{y_{i,q+2}}$ to the UAV.

CredAgg$(\{cred_i, uk_i\}_{i \in \mathcal{T}}, pp) \rightarrow (cred, uk)$: The UAV executes this algorithm after receiving t_I partial service certificates.

(1)When UAV receives a partial certificate $cred_i$, it verifies the correctness of $cred_i$ and uk_i using Equations (1) and (2).

$$e(\sigma_i, \tilde{g}) = e\left(h, \tilde{X}_i \cdot \prod^{q} \tilde{Y}_{i,j}^{H(m_j)} \cdot \tilde{Y}_{i,q+1}^{H(Sig)} \cdot \tilde{Y}_{i,q+2}^{usk}\right) \quad (1)$$

$$e(uk_i, \tilde{g}) = e(g, \tilde{Y}_{i,q+2}) \quad (2)$$

(2)UAV then aggregates the complete certificate using Equations (3) and (4):

$$\sigma = \prod_{i \in \mathcal{T}} \sigma_i^{\lambda_i} = h^{x + H(Sig) \cdot y_{q+1} + usk \cdot y_{q+2} + \sum_{j=1}^{q} H(m_j) \cdot y_j} \quad (3)$$

$$uk = \prod_{i \in \mathcal{T}} uk_i^{\lambda_i} = h^{y_{q+2}} \quad (4)$$

where $\lambda_i = [\prod_{j \in \mathcal{T}, j \neq i}(j)][\prod_{j \in \mathcal{T}, j \neq i}(j - i)]^{-1}$.

$<$ **CredShow**$(usk, uk, \vec{A}, I, pseu_{k^*,i^*}, cred, \pi_y, pp) \leftrightarrow$ **CredVerify**$(pk, W_{i^*}, pp) > \rightarrow (0/1)$: As shown in Fig. 3, UAV interacts with verifier to execute the certificate showing protocol. In this protocol, the UAV can anonymously disclose a subset of attributes I to verifier.

- Selective Disclosure of certificate by UAV: (1) Using uk and usk, UAV computes: σ', embedding the pseudonym into the certificate. (2) UAV randomly selects $r, w \in \mathbb{Z}_p$ and computes the redactable signature: $cred' = (\sigma_1, \sigma_2, \tilde{\sigma})$. (3) UAV computes the zero-knowledge proof (C_I, π_y) to prove to verifier that the service permissions corresponding to the attribute subset I have not been revoked. (4) Next, UAV computes C and Π_2 to prove to verifier knowledge of Sig. (5) To ensure the integrity of the showing token tok and the validity of the UAV's

Unmanned Aerial Vehicles UAV

- Denote $\bar{I} \cup I = \vec{A}, X(s) = \prod_{y_i \in \mathcal{R}} (s + y_i)$
- Compute $\sigma' = \sigma \cdot uk^{-usk} \cdot uk^{H(Pu||\widehat{P}u)}$
- Select $(r, w) \overset{R}{\leftarrow} \mathbb{Z}_p$, compute $\sigma_1 = h^r, \sigma_2 = (\sigma')^r \sigma_1^w, \widetilde{\sigma} = \widetilde{g}^w \prod_{j \in \bar{I}} \widetilde{Y}_j^{H(m_j)}$, set $cred' = (\sigma_1, \sigma_2, \widetilde{\sigma})$
- Select $z \overset{R}{\leftarrow} \mathbb{Z}_p$, compute $y = H(Sig) \cdot \prod_{j \in I} H(m_j), \alpha \cdot X(s) + \beta(s)(s + y) = 1$, set $C_I = h_2^z \cdot \widetilde{g}^y, \pi_y = (\widetilde{g}^\alpha, \widetilde{g}^{\beta(s)})$
- Compute $C = (\sigma_1)^{H(Sig)}, \Pi_2 = ZKSok\{H(Sig): C = (\sigma_1)^{H(Sig)}, C_I = h_2^z \cdot \widetilde{g}^y\}$, set $tok = (cred', C, \Pi_2)$
- Compute $Q_{i^*} = H_1(i^*)$
- Pick random $r_1, r_2, r_3, r_4, r_5, \gamma, \delta \overset{R}{\leftarrow} \mathbb{Z}_p$
- Compute $T_{G_1} = Q_{i^*}^{r_1}, t_2 = [e(Q_{i^*}, \widetilde{g} \cdot \widehat{P}u)]^{r_2}, \tilde{y}_1 = \widetilde{u}^\gamma \widetilde{v}^{\mu + \mu'}, \tilde{y}_2 = \widetilde{u}^\delta \widetilde{v}^{\mu'}, t_3 = \widetilde{u}^{r_3} \widetilde{v}^{r_1}, t_4 = \widetilde{u}^{r_4} \widetilde{v}^{r_2}, t_5 = \widetilde{u}^{r_5}$
- Set $c = H(tok||k^*||\tilde{y}_1||\tilde{y}_2||T_{G_1}||t_2||t_3||t_4||t_5||Pu||\widehat{P}u||Q_{i^*})$
- Compute $s_1 =- c(\mu + \mu') + r_1, s_2 =- c\mu' + r_2, s_3 =- c\gamma + r_3, s_4 =- c\delta + r_4, s_5 =- c(\delta + \gamma) + r_5$
- Set $\Pi = (tok, k^*, c, s_1, s_2, s_3, s_4, s_5, \tilde{y}_1, \tilde{y}_2, Pu, \widehat{P}u)$

$\xrightarrow{\Pi}$

Verifiers

- Compute $Q_{i^*} = H_1(i^*), d = H^{k^*}(i^*)$
- Compute $\overline{T}_{G_1} = Q_{i^*}^{s_1} \cdot Pu^c, \bar{t}_2 = e(Q_{i^*}, \widetilde{g} \cdot \widehat{P}u)^{s_2} \cdot e(Pu \cdot W_{i^*}, \widehat{P}u)^c$
- Compute $\bar{t}_3 = \widetilde{u}^{s_3} \widetilde{v}^{s_1} \tilde{y}_1^c, \bar{t}_4 = \widetilde{u}^{s_4} \widetilde{v}^{s_2} \tilde{y}_1^c, \bar{t}_5 = \widetilde{u}^{s_5} (\frac{\tilde{y}_1 \tilde{y}_2}{\widetilde{v}^d})^c$
- Recompute $c' = H(tok||k^*||\tilde{y}_1||\tilde{y}_2||\overline{T}_{G_1}||\bar{t}_2|| \bar{t}_3||\bar{t}_4||\bar{t}_5||Pu||\widehat{P}u||Q_{i^*})$
- Check $c' = c$, verify the proof Π_2
- Check $e(\sigma_2, \widetilde{g}) = e(\sigma_1, \widetilde{X} \cdot \widetilde{\sigma} \cdot \prod_{j \in I} \widetilde{Y}_j^{H(m_j)} \cdot \widetilde{Y}_{q+2}^{H(Pu||\widehat{P}u)}) \cdot e(C, \widetilde{Y}_{q+1})$

Fig. 3: Certificate Showing Algorithm.

pseudonym $(Pu, \hat{P}u)$, UAV computes the pseudonym signature $(c, s_1, s_2, s_3, s_4, s_5, \tilde{y}_1, \tilde{y}_2)$.

- Verifier Validates certificate Validity: (1) Upon receiving Π, verifier computes Q_{i^*} and d, and obtains W_{i^*} from the blockchain before the activation of each time interval. (2) Verifier computes $\bar{T}_{G_1}, \bar{t}_2, \bar{t}_3, \bar{t}_4$ and $\bar{t}_5$, and verifies the validity of the pseudonym signature by checking whether $c' = c$. (3) Next, verifier checks if Π_2 is correct. If Π_2 is valid, then checks the validity of the UAV's certificate $cred'$ using the equation defined in Fig. 3.
- To check if the certificate has been revoked, the following steps are executed: Let $\bar{A} = \tilde{g}^\alpha, \bar{B} = g^{\beta(s)}$.

1) UAV computes: $C'_I = C_I \cdot \tilde{g}^s$.
2) UAV randomly selects $\tau_1, \tau_3, \tau_4 \in \mathbb{Z}_p$ and computes:
   - $\bar{A}_2 = \bar{A}\mathfrak{h}^{\tau_1}$,
   - $\bar{B}_1 = g^{\tau_3}\mathfrak{g}^{\tau_4}$,
   - $\bar{B}_2 = \bar{B}\mathfrak{g}^{\tau_3}$.
3) UAV randomly selects $r_z, r_{\tau_1}, r_{\tau_3}, r_{\tau_4}, r_{\delta_3}, r_{\delta_4} \in \mathbb{Z}_p$, and computes
   - $R_{2,1} = g^{r_{\tau_3}} g^{r_{\tau_4}}$,
   - $R_{2,2} = (\bar{B}_1)^{r_z} g^{-r_{\delta_3}} \mathfrak{g}^{-r_{\delta_4}}$,
   - $R_3 = e(acc, \mathfrak{h})^{r_{\tau_1}} \cdot e(\mathfrak{g}, C'_I)^{r_{\tau_3}} \cdot e(\mathfrak{g}, h_2)^{-r_{\delta_3}} \cdot e(\bar{B}_2, h_2)^{r_z})$,

   and sends: $(\bar{A}_2, \bar{B}_1, \bar{B}_2, R_{2,1}, R_{2,2}, R_3)$.
4) Verifier sends a challenge $c \in \mathbb{Z}_p$.
5) UAV computes the responses:
   - $s_z = r_z + cz, s_{\tau_1} = r_{\tau_1} + c\tau_1$,
   - $s_{\tau_3} = r_{\tau_3} + c\tau_3, s_{\tau_4} = r_{\tau_4} + c\tau_4$,
   - $s_{\delta_3} = r_{\delta_3} + c\delta_3, s_{\delta_4} = r_{\delta_4} + c\delta_4$,

   where $\delta_3 = \tau_3 \cdot z$ and $\delta_4 = \tau_4 \cdot z$.
6) Verifier checks the following equations:
   - $R_{2,1} = (\bar{B}_1)^{-c} g^{s_{\tau_3}} g^{s_{\tau_4}}$,
   - $R_{2,2} = (\bar{B}_1)^{s_z} g^{-s_{\delta_3}} g^{-s_{\delta_4}}$,
   - $R_3 \cdot \left(\frac{e(acc, \bar{A}_2) \cdot e\left(\bar{B}_2, C'_I\right)}{e(g, \tilde{g})}\right)^C = e(acc, \mathfrak{h})^{s_{\tau_1}} \cdot e(\mathfrak{g}, C'_I)^{s_{\tau_3}} \cdot e(\mathfrak{g}, h_2)^{-s_{\delta_3}} \cdot e(\bar{B}_2, h_2)^{s_z}$.

Remark 2: For completeness, the instantiation of Π_2 is as follows: (1)UAV randomly selects $a_1, a_2, a_3 \in \mathbb{Z}_p$ and compute: $R_1 = \sigma_1^{a_1}, R_2 = \sigma_1^{a_2}, R_3 = h_2^{a_3} \cdot \tilde{g}^{a_2}$, $c = H(C, C_I, \sigma_1, h_2, \tilde{g}, R_1, R_2, R_3)$. (2)UAV computes responses: $s_1 = a_1 - c \cdot H(Sig), s_2 = a_2 - c \cdot H(Sig) \cdot \prod_{j \in I} H(m_j), s_3 = a_3 - c \cdot z$, and sends (c, s_1, s_2, s_3). (3)Verifier computes $c' = H(C, C_I, \sigma_1, h_2, \tilde{g}, \sigma_1^{s_1} \cdot C^c, \sigma_1^{s_2} \cdot (C^{\prod_{j \in I} H(m_j)})^c, h_2^{s_3} \cdot \tilde{g}^{s_2} \cdot C_I^c)$ and verifies if $c' = c$.

Trace$(PseuL, pseu, i^*) \to (id/\bot)$: When the real identity of a malicious UAV needs to be traced, the verifier sends the pseudonym to the ATC. If the pseudonym $pseu$ is part of a valid signature during time slot i^*, the ATC performs a binary search in $PseuL$ during time slot i^* to identify the corresponding UAV identity id.

Revoke$(\{ipk_i, isk_i, id, S, st\}_{I_i, i \in t_A}, pp) \to st'$: The revocation of UAVs is collaboratively executed by t_A issuers. When certain attributes in a certificate are revoked, the certificate holder can still generate a valid proof based on the subset of non-revoked attributes; any proof containing revoked attributes will be rejected by the verification algorithm. (1)If an issuer I_i intends to revoke the service privileges $\prod_{j \in S} H(m_j)$of a UAV, the issuer first computes $y = H(Sig) \cdot \prod_{j \in S} H(m_j), acc' = g^{(s+y)}$ and sends (y, id, S) to other issuers. Subsequently, the other issuers sign y, where $sig_i = acc' y_{i,q+3}$. Here, S is the set of attributes corresponding to the UAV services that the issuers wish to revoke. (2)When the ATC receives t_A valid signatures, it checks whether the following equation holds: $e\left(\prod_{i \in D} sig_i^{\lambda_i}, \tilde{g}\right) = e(acc', \tilde{Y}_{q+3})$, where $D \subset [1, n_I]$. (3)If the equation holds, the ATC updates the revocation list st', such that $acc = acc^{(s+y)}, \mathcal{R} = \mathcal{R} \cup \{y\}$, and writes the updated revocation list to the blockchain.

Remark 3: Issuer I_i can revoke the service privileges of multiple UAVs simultaneously $(y_1, y_2, ..., y_n)$ by sending these values to other issuers for signing. When the ATC receives t_A valid signatures, it performs a batch update to the revocation list st', such that $acc = acc^{\prod_{i \in n}(s+y_i)}$ and $\mathcal{R} = \mathcal{R} \cup \{y_1, y_2, ..., y_n\}$. Notably, when multiple issuers submit revocation requests based on the same y_i, the ATC can aggregate updates to eliminate duplicate revocation requests.

VI. Security Proof And Analysis

In our security analysis, H is treated as a collision-resistant function and is modeled as a random oracle, which uniformly maps inputs to $\mathbb{Z}_p$. Therefore, we demonstrate that the scheme satisfies the EUF-CMA security requirements only under the condition that Assumption 1 holds. It is important to emphasize that we consider only forgery attacks on the original certificate, since redacting it does not require any trapdoor information.

A. Correctness

Theorem 1 (Correctness): Our scheme satisfies correctness.

Proof: First, we demonstrate the correctness of the certificate generation algorithm by showing that the UAV can validate the certificate through the following equation.

$$\begin{aligned} e(\sigma_i, \tilde{g}) &= e\left(h^{x_i + H(Sig)\cdot y_{i,q+1} + usk \cdot y_{i,q+2} + \sum_{j=1}^{q} H(m_j)\cdot y_{i,j}}, \tilde{g}\right) \\ &= e\left(h, \tilde{g}^{x_i + H(Sig)\cdot y_{i,q+1} + usk \cdot y_{i,q+2} + \sum_{j=1}^{q} H(m_j)\cdot y_{i,j}}\right) \\ &= e(h, \tilde{X}_i \cdot \prod_{j=1}^{q} \tilde{Y}_{i,j}^{H(m_j)} \cdot \tilde{Y}_{i,q+1}^{H(Sig)} \cdot \tilde{Y}_{i,q+2}^{usk}) \end{aligned}$$

Secondly, we prove that the certificate presentation algorithm is correct. For any valid token $tok = (cred', C, \Pi_2)$

$$\begin{aligned} & e\left(\sigma_1, \tilde{X} \cdot \tilde{\sigma} \cdot \prod_{j \in I} \tilde{Y}_j^{H(m_j)} \cdot \tilde{Y}_{q+2}^{H(Pu||\hat{P}u)}\right) e(C, \tilde{Y}_{q+1}) \\ &= e\left(h^r, \tilde{g}^x \tilde{g}^w \tilde{Y}_{q+2}^{H(Pu||\hat{P}u)} \prod_{j \in \bar{I}} \tilde{Y}_j^{H(m_j)} \prod_{j \in I} \tilde{Y}_j^{H(m_j)}\right) \\ &\quad \cdot e\left((\sigma_1)^{H(Sig)}, \tilde{Y}_{q+1}\right) \\ &= e\left(h^{r(w+x+H(Pu||\hat{P}u)\cdot y_{q+2} + \sum_{j=1}^{q} H(m_j)\cdot y_j)}, \tilde{g}\right) \\ &\quad \cdot e\left(h^{r \cdot H(Sig)\cdot y_{q+1}}, \tilde{Y}_{q+1}\right) \\ &= e\left(h^{r(w+x+H(Sig)\cdot y_{q+1} + H(Pu||\hat{P}u)\cdot y_{q+2} + \sum_{j=1}^{q} H(m_j)\cdot y_j)}, \tilde{g}\right) \\ &= e(\sigma_2, \tilde{g}) \end{aligned}$$

Finally, the disclosed certificate attributes are shown to be non-revoked. For any disclosed attribute set I

$$\begin{aligned} & R_3 \cdot \left(\frac{e(acc, \bar{A}_2) \cdot e(\bar{B}_2, C_I')}{e(g, \tilde{g})}\right)^c \\ &= R_3 \cdot (e\left(g^{\beta(s)} \cdot g^{\tau_3}, h_2^z \cdot \tilde{g}^{\left(H(Sig)\cdot \prod_{j \in I} H(m_j) + s\right)}\right) \\ &\quad \cdot e(g^{\prod_{i \in \mathcal{R}}(s+y_i)}, \tilde{g}^{\alpha} \cdot \mathfrak{h}^{\tau_1}))/e(g, \tilde{g}))^c \\ &= R_3 \cdot (e\left(g^{\alpha \prod_{i \in R}(s+y_i) + \beta(s)\left(H(Sig)\cdot \prod_{j \in I} H(m_j) + s\right)}, \tilde{g}\right) \\ &\quad \cdot e(acc, \mathfrak{h})^{c\tau_1} \cdot e(\mathfrak{g}, h_2)^{c \cdot z\tau_3} \cdot e(\bar{B}_2, h_2)^{cz})/e(g, \tilde{g}) \\ &= e(acc, \mathfrak{h})^{s_{\tau_1}} \cdot e(\mathfrak{g}, C_I')^{s_{\tau_3}} \cdot e(\mathfrak{g}, h_2)^{-s_{\delta_3}} \cdot e(\bar{B}_2, h_2)^{s_z} \end{aligned}$$

B. Unforgeability

Theorem 2 (Unforgeability): Suppose an adversary $\mathcal{A}$ can forge a valid certificate proof $((C_I, \pi_y), cred)$ for a revoked attribute set I with probability p_1, and can forge a valid certificate proof for a non-existent certificate $cred$ with probability p_2. Then A succeeds in forging a valid certificate proof with probability $p = p_1 + p_2$.

For a revoked certificate, (C_I, π_y) can be viewed as a non-revocation proof, which essentially corresponds to a non-membership proof in a dynamic accumulator. Based on the soundness of the accumulator [32], the adversary can only construct a non-membership proof that simultaneously satisfies $\{y\} \nsubseteq R$ and $\{y\} \subseteq R$ with negligible probability ϵ_1, such that it passes verification. For a non-existent certificate, under Assumption 1, our scheme guarantees EUF-CMA security. Specifically, if the adversary is capable of compromising the EUF-CMA security of our scheme with a non-negligible advantage ϵ, then one can construct an adversary that violates Assumption 1 with success probability exceeding $\epsilon - \frac{Q_1}{q}$.

Proof: Let $\mathcal{A}$ be a probabilistic polynomial-time adversary that attacks the EUF-CMA security of the proposed construction, and let $\mathcal{C}$ be the challenger in the security game. We design an efficient algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the hardness of Assumption 1. Without loss of generality, we suppose that whenever $\mathcal{A}$ interacts with the certificate issuance oracle $CredIssue$ with an attribute set M, or returns a forged certificate $(M, cred)$, $\mathcal{A}$ has queried the hash function H on M beforehand.

Setup Phase: The algorithm $\mathcal{B}$ obtains the public key pk^* from the challenger $\mathcal{C}$, which includes public parameters $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ and $(\tilde{g}, \tilde{X}, \tilde{Y})$, configured in accordance with Assumption 1. Next, $\mathcal{B}$ selects $\{\alpha_j, \beta_j\}_{j=1}^{j=n} \overset{\$}{\leftarrow} \mathbb{Z}_p$, and sets $\tilde{Y}_j \leftarrow \tilde{Y}^{\beta_j} \tilde{g}^{\alpha_j}$. Furthermore, $\mathcal{B}$ defines a random hash function $H{:}\{0,1\}^* \rightarrow \mathbb{Z}_p$. Finally, it constructs $pk \leftarrow (\tilde{g}, \tilde{X}, \tilde{Y}_j)$ and sends it to $\mathcal{A}$.

Query Phase: During this stage, the adversary $\mathcal{A}$ is allowed to query the random oracle H and the certificate issuance oracle $CredIssue_{pk}(\cdot)$ in polynomial time. When $\mathcal{B}$ obtains a certificate issuance request from $\mathcal{A}$ for an attribute set $M_i = \{m_{i,1}, ..., m_{i,n}\}$, it responds to $\mathcal{A}$'s random oracle queries with uniformly random elements from $\mathbb{Z}_p$. Then, $\mathcal{B}$ queries the certificate issuance oracle for the certificate corresponding to $m_i = \sum \beta_j H(m_{i,j})$, obtaining $cred = (\sigma_1, \sigma_2)$. This certificate satisfies: $e(\sigma_2, \tilde{g}) = e(\sigma_1, \tilde{X} \cdot \tilde{Y}^{\sum \beta_j H(m_{i,j})})$. Finally, $\mathcal{B}$ computes $\sigma_2' \leftarrow \sigma_1^{\Sigma \alpha_j H(m_{i,j})} \cdot \sigma_2$, and returns $\sigma' = (\sigma_1, \sigma_2')$ to $\mathcal{A}$. From $\mathcal{A}$'s perspective, this is a valid signature for the attribute set M_i, because:

$$\begin{aligned} & e(\sigma_2', \tilde{g}) \\ &= e(\sigma_2 \cdot \sigma_1^{\sum \alpha_j H(m_{i,j})}, \tilde{g}) \\ &= e(\sigma_2, \tilde{g}) \cdot e(\sigma_1^{\sum \alpha_j H(m_{i,j})}, \tilde{g}) \\ &= e(\sigma_1, \tilde{X} \cdot \tilde{Y}^{\Sigma \beta_j H(m_{i,j})}) \cdot e(\sigma_1, \tilde{g}^{\Sigma \alpha_j H(m_{i,j})}) \\ &= e\left(\sigma_1, \tilde{X} \cdot \prod \tilde{Y}^{\beta_j H(m_{i,j})} \tilde{g}^{\alpha_j H(m_{i,j})}\right) \\ &= e\left(\sigma_1, \tilde{X} \cdot \prod (\tilde{Y}^{\beta_j} \tilde{g}^{\alpha_j})^{H(m_{i,j})}\right) \\ &= e(\sigma_1, \tilde{X} \cdot \prod \tilde{Y}_j^{H(m_{i,j})}). \end{aligned}$$

TABLE III: Comparing Communication and Computation Overheads

Scheme	Communication Complexity	Computational Complexity	
	Certificate Showing	Show Certificate	Verify Certificate
DTACB [12]	$(5q+4)\lvert\mathbb{G}_1\rvert+4(q+1)\lvert\mathbb{G}_2\rvert+\lvert\mathbb{G}_T\rvert+(2q+5)\lvert\mathbb{Z}_p\rvert$	$(3q+k+8)t_{e_1}+(5q+6)t_{e_2}+3t_p$	$2(q+1)t_{e_1}+4qt_{e_2}+4(q+2)t_p$
TABC [17]	$5\lvert\mathbb{G}_1\rvert+\lvert\mathbb{G}_2\rvert+(k+1)\lvert\mathbb{Z}_p\rvert$	$[2(q+2)+1]t_{e_1}+(q-k+2)t_{e_2}$	$(k+2)t_{e_1}+kt_{e_2}+(5+2\lvert L\rvert)t_p$
PS [19]	$4\lvert\mathbb{G}_1\rvert+9\lvert\mathbb{G}_2\rvert+2\lvert\mathbb{G}_T\rvert+(q+9)\lvert\mathbb{Z}_p\rvert$	$[3(q-k)+10]t_{e_1}+15t_{e_2}+4t_p$	$(q+2)t_{e_1}+8t_{e_2}+8t_p$
Ours	$9\lvert\mathbb{G}_1\rvert+2\lvert\mathbb{G}_2\rvert+\lvert\mathbb{G}_T\rvert+(k+17)\lvert\mathbb{Z}_p\rvert$	$15t_{e_1}+(q-k+6)t_{e_2}+5t_p$	$(k+9)t_{e_2}+8t_p$

Output Phase: Finally, after Q_1 queries within polynomial time, the adversary $\mathcal{A}$ outputs a certificate $cred = (\sigma_1, \sigma_2)$ for an attribute set M^*. If the attribute set M^* was never submitted to the certificate issuance oracle, and the certificate $cred$ satisfies the following verification equations, then this constitutes a successful forgery: 1)$e(\sigma_2, \tilde{g}) = e(\sigma_1, \tilde{X} \cdot \prod \tilde{Y}_j^{H(m_j^*)})$; 2)$M^* \neq M_i$.

If there exists $i \in \{1, ..., Q_1\}$ such that $\sum \beta_j H(m_{i,j}) = \sum \beta_j H(m_j^*)$, algorithm $\mathcal{B}$ terminates. Otherwise, $\mathcal{B}$ outputs $\sigma^* = (\sigma_1^*, \sigma_2^*)$ and $m^* \leftarrow \Sigma \beta_j H(m_j^*)$, where $\sigma_1^* \leftarrow \sigma_1$, $\sigma_2^* \leftarrow \sigma_2 \cdot \sigma_1^{-\sum \alpha_j \cdot H(m_j^*)}$, and

$$\begin{aligned}
&e(\sigma_2^*, \tilde{g}) \\
&= e\left(\sigma_2 \cdot \sigma_1^{-\Sigma \alpha_j H(m_j^*)}, \tilde{g}\right) \\
&= e(\sigma_2, \tilde{g}) \cdot e\left(\sigma_1^{-\sum \alpha_j H(m_j^*)}, \tilde{g}\right) \\
&= e\left(\sigma_1, \tilde{X} \cdot \prod \tilde{Y}_j^{H(m_j^*)}\right) \cdot e\left(\sigma_1, \tilde{g}^{-\sum \alpha_j H(m_j^*)}\right) \\
&= e\left(\sigma_1, \tilde{X} \cdot \prod (\tilde{g}^{\alpha_j} \tilde{Y}^{\beta_j})^{H(m_j^*)}\right) \cdot e(\sigma_1, \tilde{g}^{-\sum \alpha_j H(m_j^*)}) \\
&= e\left(\sigma_1, \tilde{X} \cdot \prod \tilde{Y}^{\beta_j H(m_j^*)}\right) \\
&= e(\sigma_1, \tilde{X} \cdot \tilde{Y}^{m^*}).
\end{aligned}$$

Under Assumption 1, if the attribute m^* was never submitted to the certificate issuance oracle, then the certificate (σ_1^*, σ_2^*) constitutes a legitimate forgery under the public key pk^*. This indicates that the valid forgery produced by adversary $\mathcal{A}$ allows $\mathcal{B}$ to break Assumption 1. Unless there exists some attribute set M_i previously submitted to the certificate issuance oracle, such that $\sum \beta_j H(m_j^*) = \sum \beta_j H(m_{i,j})$, and M^* has been queried to the random oracle, it must be demonstrated that the likelihood of such a linear relationship occurring is negligible.

Let $\{\zeta_j\}_{j=1}^n \leftarrow \mathbb{Z}_p$, and define $\beta_j' \leftarrow \beta_j - \zeta_j$ and $\alpha_j' \leftarrow \alpha_j + y\zeta_j$, where y is a random element in $\mathbb{Z}_p$, and the public parameter $\tilde{Y} = \tilde{g}^y$. It can be derived that: $\tilde{g}^{\alpha_j'} \tilde{Y}^{\beta_j'} = \tilde{g}^{\alpha_j + y\zeta_j} \tilde{Y}^{\beta_j - \zeta_j} = \tilde{g}^{\alpha_j} \tilde{Y}^{\beta_j} = \tilde{Y}_j$. This demonstrates that the public parameter $\tilde{Y}$ bears no dependency on ζ_j and, thus, does not reveal any information about β_j. Similarly, this property also holds for σ_2, which depends on the public key and the certificate issuance oracle. Thus, from the adversary $\mathcal{A}$'s perspective, all certificate issuance queries are handled by the challenger's certificate issuance oracle, rather than by the simulated version constructed by $\mathcal{B}$. Moreover, in $\mathcal{A}$'s complete view, the values β_j are entirely independent. In conclusion, the probability that algorithm $\mathcal{B}$ aborts is bounded by $\frac{Q_1}{q}$.

C. Unlinkability

Theorem 3 (Unlinkability): Within the framework of cryptographic information theory, our scheme satisfies the unlinkability property defined in Definition 3.

Proof: From an information-theoretic perspective, the certificates generated by our scheme achieve unlinkability, meaning that the secret bit b selected during the unlinkability experiment remains completely hidden. During the challenge phase, the adversary outputs attributes $m_j^{(0)} = m_j^{(1)}$ for all $j \in D$, and the challenger outputs the attributes $m_j^{(b)}$ belonging to D. This indicates that the adversary's ability to distinguish which attribute set the certificate belongs to is negligible. Consequently, for the certificate $cred_D$ of the attribute set $\{m_i\}_{i \in D}$, its distribution remains independent of both the hidden attribute set $\{m_i\}_{i \in \bar{D}}$ and the original certificate $cred$ generated from the attribute set $\{m_i\}_{i=1}^n$.

Setup Phase: The challenger runs the $IssuerKeyGen$ algorithm to generate a signature key pair (isk, ipk) and sends the public key ipk to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ can adaptively select multiple attribute sets $(M_1, M_2, \ldots, M_{|Q_1|})$ and request the corresponding certificates from the challenger. The challenger runs $CredIssue(isk, ipk, pp)$ to generate a certificate $cred = (\sigma_1, \sigma_2, \tilde{\sigma})$ for each attribute set M and returns it to $\mathcal{A}$.

Challenge: The adversary outputs two attribute sets $M^0 = \{m_j^{(0)}\}_{j=1}^n$ and $M^1 = \{m_j^{(1)}\}_{j=1}^n$, satisfying $m_j^{(0)} = m_j^{(1)}$ for all $j \in D$. The adversary then sends M^0, M^1, and the index set D to the challenger.

The challenger runs the $CredShow$ algorithm, and using random elements $\xi \in \mathbb{Z}_p$ and $\eta \in \mathbb{G}_1$. For any subset $D \subset [1, n]$, define $b = \xi - \sum_{i \in \bar{D}} y_i H(m_i)$, $a = \frac{\kappa}{\tau}$, where κ and τ satisfy $\eta = g^\kappa$ and $\sigma_1 = g^\tau$.Since ξ and η are chosen uniformly at random, it ensures that a and b in the certificate showing algorithm are also uniformly random. Using these parameters, the $CredShow$ algorithm outputs a redactable certificate $cred_D = (\sigma_1', \sigma_2', \tilde{\sigma}')$ for $\{m_i\}_{i=1}^n$, where: $\sigma_1' \leftarrow \sigma_1^a = \eta$, $\sigma_2' \leftarrow \sigma_2^a \cdot \sigma_1'^b = \eta^{x + \sum_{i \in D} y_i H(m_i)} \cdot \eta^\xi$, $\tilde{\sigma}' \leftarrow \tilde{g}^b \prod_{i \in \bar{D}} \tilde{Y}_i^{H(m_i)} = \tilde{g}^\xi$. It is evident that this certificate satisfies the verification equation.

Phase 2: Similar to Phase 1, the adversary $\mathcal{A}$ can continue to adaptively request new certificates from the challenger.

Guess: Due to the randomness of ξ and η, the distribution of the redactable certificate $cred_D$ remains entirely unrelated to both the concealed attributes $\{m_i\}_{i \in \bar{D}}$ and the initial certificate. Thus, even against an unbounded adversary, the bit b cannot be guessed with non-negligible advantage.

VII. Implementation

A. Theoretical Analysis and Comparison

Table III computational complexities of our scheme with Scheme [19], Scheme [12], and Scheme [17]. Here, $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, $|\mathbb{G}_T|$ and $|\mathbb{Z}_p|$ denote the sizes of elements in the groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and $\mathbb{Z}_p$, respectively. t_{e_1}, t_{e_2}, and t_p represent the time costs of exponentiation in $\mathbb{G}_1$, $\mathbb{G}_2$, and pairing operations, respectively. $|L|$ denotes the length of the revocation list. In Scheme [12], a single certificate only supports the verification of one service's subscription status. Users need to communicate with the trusted center to convert their certificates into aggregate accumulator membership proofs. During certificate presentation, the user must aggregate the membership proofs for disclosure and provide them to the verifier, resulting in high communication and computational costs. Scheme [17] ensures unlinkability during certificate access by scanning the entire revocation list via bilinear mappings during verification, which significantly increases computational costs. Scheme [19] utilizes zero-knowledge proofs to support selective attribute disclosure, incurring substantial communication and computational overhead. Theoretical analysis indicates that our scheme achieves fine-grained revocation of UAV permissions while significantly reducing communication and computational overhead, thus exhibiting superior performance. Notably, our scheme supports efficient selective disclosure in distributed scenarios, enabling fine-grained service revocation and efficient identity tracing. These improvements are of great significance for UAV cross-region multi-service access scenarios.

B. Experimental Analysis

To assess the real-world efficiency of our system, we implemented the proposed scheme using the cryptographic library miracl core with C++ programming, enabling the use of the BLS12381 curve to achieve a 128-bit security level. We conducted tests separately on a personal laptop, a laboratory server, and a UAV. The laptop is equipped with an AMD Ryzen 5 7600X 6-core processor, with a CPU clock speed of 4.70 GHz, 32 GB of RAM, and running the Ubuntu 20.04 operating system. As shown in Fig. 4, the UAV used is the P600 model from AmovLab [39], equipped with an ARM64 Cortex-A78AE 8-core processor, 8 GB of RAM, and the same Ubuntu operating system. Communication between devices is established via the Homer module. Fig. 5 illustrates the average time required to perform a single cryptographic primitive on both the host and UAV platforms. To evaluate the performance of on-chain operations, we implemented the corresponding read and write operations in Go. The testbed consisted of a ten-node Hyperledger Fabric network deployed on identical PCs. Each node ran Ubuntu 18.04.3 equipped with an Intel Core i7-11700 CPU @2.5GHz and 16GB RAM. The network topology comprised one client node, one orderer node, and eight peer nodes configured as endorsers.

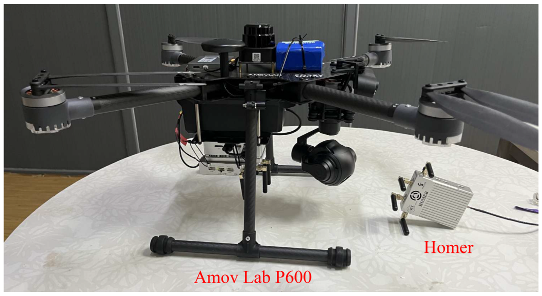


Fig. 4: Unmanned Aerial Vehicle Device.

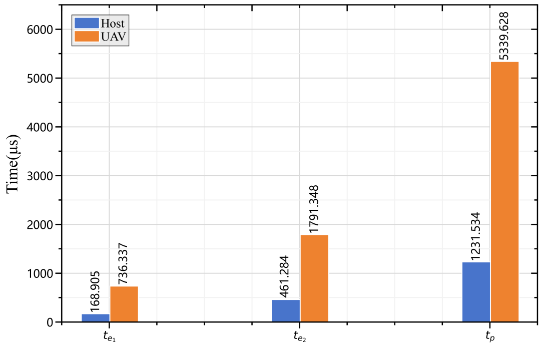
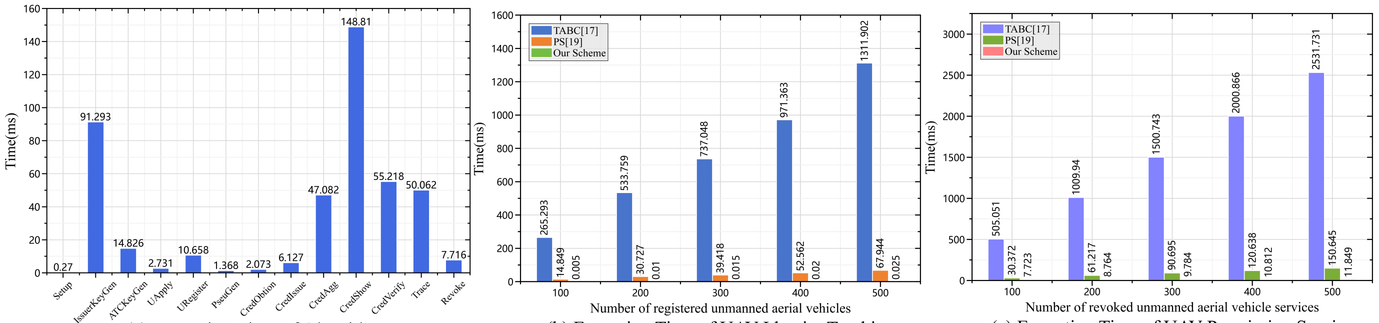


Fig. 5: Execution Time of Basic Cryptographic Operations.

Fig. 6(a) presents the execution time of each algorithm in the system, with the parameters set as $n_I = 5$, $t_I = 3$, $n_A = 5$, $t_A = 3$, $q = 10$, $k = 3$, and an initial revocation list containing 100 revoked UAV permissions. When the registration list length is 10000, the time overhead of the tracing algorithm is 50.062ms. In the threshold revocation algorithm, the time overhead for batch revocation of 100 UAV permissions is 7.716ms. Fig. 6(b) compares the time overhead of the tracing algorithm in our scheme with that of other schemes. Schemes [19] and [17] require scanning the entire registration list using bilinear pairing and g^{id}, respectively, to retrieve the true identity of the user, resulting in high computational costs. In contrast, our scheme employs a self-generated pseudonym algorithm to achieve an efficient and flexible anonymous authentication mechanism. The air traffic control center can directly search for the UAV identity id in $PseuL$ through binary search, incurring minimal computational overhead. Fig. 6(c) also compares the time overhead of the revocation algorithm in our scheme with other schemes. Our scheme supports batch threshold revocation, whereas schemes [19] and [17] do not support batch revocation, and scheme [19] does not support threshold revocation. Therefore, the revocation algorithm in our scheme demonstrates a significant advantage in computational overhead.

In real-world scenarios, the number of attributes q requested by a UAV is typically much larger than the number of attributes k that need to be disclosed. For instance, a UAV may subscribe to services such as high-precision mapping, data analytics, and network communication. However, when performing low-altitude logistics tasks, it only needs to selectively disclose a subset of attributes to access the high-precision map service, while the majority of attributes must remain hidden. Therefore, this section focuses on analyzing the communication and computational overhead of the $CredShow$ and $CredVerify$ algorithms in scenarios where k is relatively low.

Fig. 7 compares the communication and computational overhead of the $CredShow$ and $CredVerify$ algorithms in the certificate presentation process. In the tests, $k = 10$ is fixed while q increases linearly. Fig. 7(a) illustrates the communication overhead of the certificate presentation algorithms for each scheme. The communication overhead of Schemes [19] and [12] increases linearly with the number of UAV attributes q. In contrast, our scheme produces certificates whose size does not depend on q, and when k is fixed, the communication overhead remains constant and is significantly smaller than that of Schemes [19] and [12]. Additionally, although the communication overhead of Scheme [17] is slightly lower than

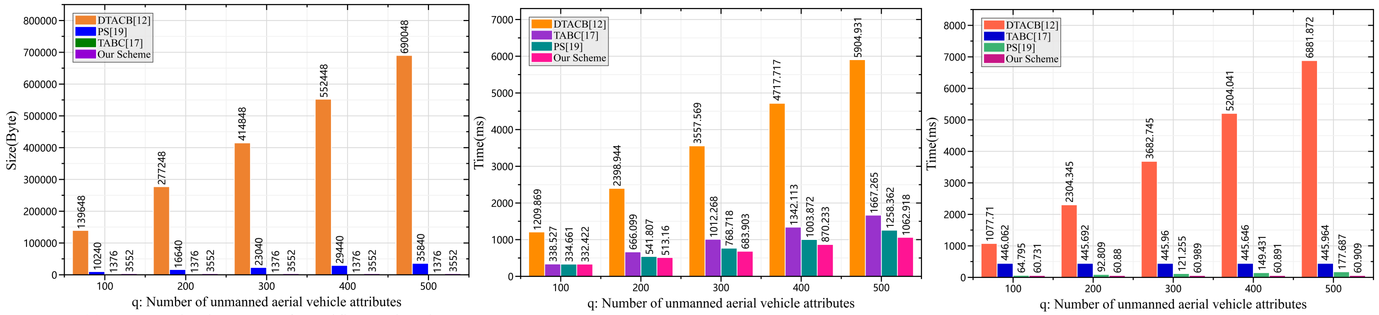


(a) Execution Time of Algorithms

(b) Execution Time of UAV Identity Tracking

(c) Execution Time of UAV Permission Service

Fig. 6: Execution Time of the Algorithms.



(a) Communication Cost of Certificate Showing

(b) Certificate Showing Execution Time

(c) Certificate Verifying Execution Time

Fig. 7: Overhead of Certificate Showing (k=10).

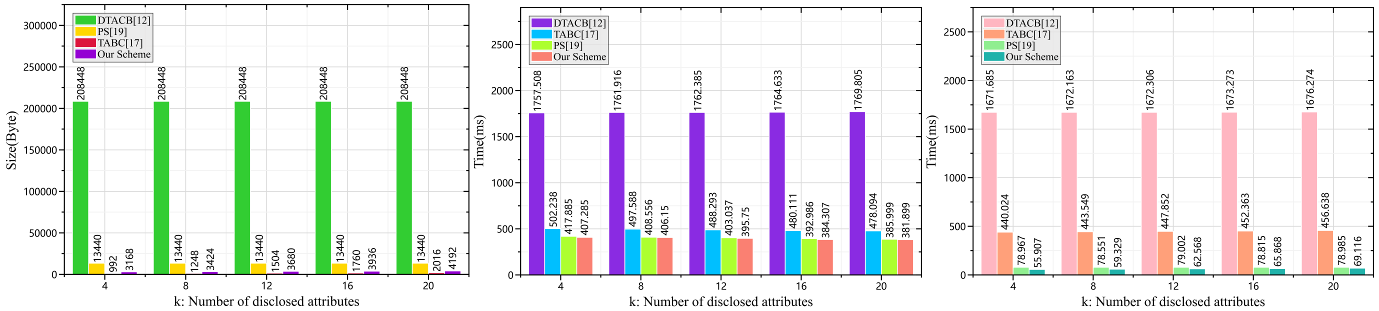
that of our scheme, our scheme supports fine-grained revocation of UAV permissions, offering more comprehensive functionality. Fig. 7(b) and Fig. 7(c) compare the computational overhead of the CredShow and CredVerify algorithms, respectively. As q increases, the computational overhead of both our construction and the comparison schemes grows linearly. However, the computational cost of our construction is considerably less than that of the other schemes, demonstrating higher efficiency.

Fig. 8 compares the communication and computational overhead of the $CredShow$ and $CredVerify$ algorithms in the certificate presentation process. In the tests, $q = 150$ is fixed while k increases linearly. Fig. 8(a) illustrates the communication overhead of the certificate presentation algorithms for each scheme. Although our scheme incurs a communication cost that increases linearly with k, it remains significantly lower than that of Schemes [19] and [12]. Fig. 8(b) shows the computational overhead of the $CredShow$ algorithm. The computational overhead of our construction decreases proportionally as k increases, whereas Scheme [12] exhibits linear growth as k increases. Fig. 8(c) compares the computational overhead of the CredVerify algorithm. Both our scheme and Schemes [12] and [17] experience linear growth in computational overhead as k increases, while Scheme [19] maintains constant computational costs. Nonetheless, the computational cost incurred by our scheme remains lower compared to that of the existing alternatives. In summary, our scheme achieves fine-grained revocation of UAV permissions while significantly reducing communication and computational overhead, offering superior performance.

Following the established experimental configuration, we conducted a systematic evaluation of the blockchain system's read and write performance, with results presented in Fig. 9. Fig. 9(a) demonstrates that with a constant transmission rate of 120 TPS, the system throughput remained stable at 120 TPS across varying batch write sizes. Fig. 9(b) further indicates that under the same transmission rate, increasing the batch write size only resulted in a gradual rise in average latency. These findings confirm that by selecting an appropriate batch write size, our scheme can effectively leverage the blockchain's batch processing capabilities to support large-scale UAV permission revocation scenarios. As shown in Fig. 9(c), with a fixed batch read size of 40, the system throughput increased linearly with the transmission rate. Notably, Fig. 9(c) reveals that the read latency remained consistently stable at 0.02 seconds, unaffected by increases in the transmission rate. Since our certificate verification process relies exclusively on read operations with a fixed data size per verification, the observed latency characteristics demonstrate the scheme's capability to support large-scale UAV verification scenarios.

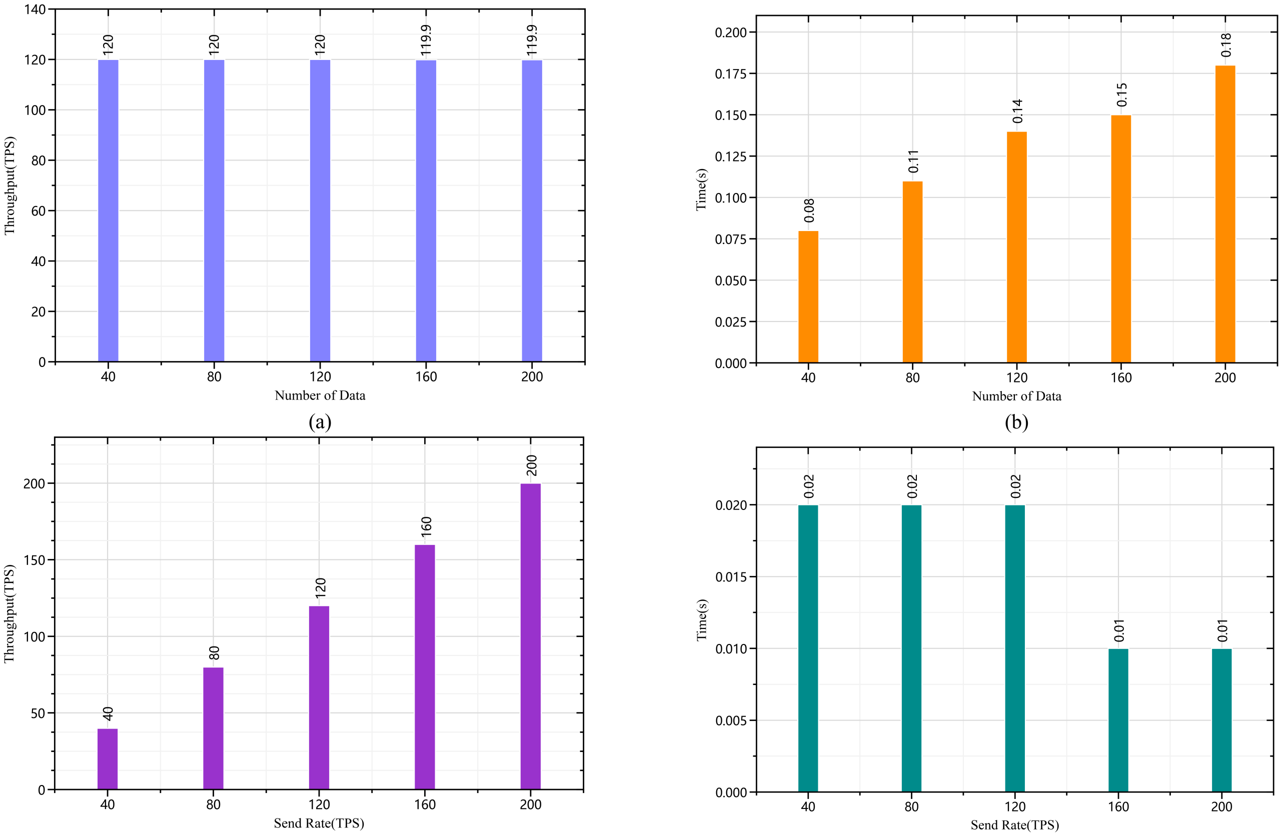
VIII. Conclusion

When executing multiple tasks, UAVs are typically required to present anonymous certificates containing multiple attributes for accessing specific services. However, existing schemes encounter challenges in realizing an effective trade-off between verification efficiency and fine-grained revocation. Specifically, one class of schemes issues separate certificates for each attribute, leading to computational and communication overheads that increase linearly with the number of disclosed attributes during verification. Another class aggregates multiple attributes into a single certificate but cannot revoke individual attributes in a fine-grained manner. Moreover, existing schemes



(a) Communication Cost of Certificate Showing (b) Certificate Showing Execution Time (c) Certificate Verifying Execution Time

Fig. 8: Overhead of Certificate Showing (q=150).



(a) (b)

(c) (d)

Fig. 9: On-chain read and write operations.

typically rely on a trusted authority to issue signing keys; this requires UAVs to maintain certificates for multiple regions, resulting in a heavy burden on certificate management. To address these issues, this study devised a distributed anonymous authentication scheme that supports fine-grained revocation. By integrating the redactable signature and dynamic accumulator techniques, the scheme enables efficient selective disclosure and attribute-level revocation of UAV attributes. Additionally, the DKG protocol provides UAVs with stable and seamless cross-regional services. Security proofs and experimental results demonstrate that the proposed scheme satisfies security requirements while significantly reducing communication and computational overhead.

References

[1] J. Liu, X. Du, J. Cui, M. Pan, and D. Wei, "Task-oriented intelligent networking architecture for the space–air–ground–aqua integrated network," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5345–5358, 2020.

[2] H. Gao, J. Feng, Y. Xiao, B. Zhang, and W. Wang, "A uav-assisted multi-task allocation method for mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3790–3804, 2023.

[3] K. Meng, D. Li, X. He, and M. Liu, "Space pruning based time minimization in delay constrained multi-task uav-based sensing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2836–2849, 2021.

[4] J. Fan, L. Fan, Q. Ni, J. Wang, Y. Liu, R. Li, Y. Wang, and S. Wang, "Perception and planning of intelligent vehicles based on bev in extreme off-road scenarios," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 4, pp. 4568–4572, 2024.

[5] H. Huang, J. Su, and F.-Y. Wang, "The potential of low-altitude airspace: The future of urban air transportation," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 8, pp. 5250–5254, 2024.

[6] H. A. H. Alobaidy, R. Nordin, M. J. Singh, N. F. Abdullah, A. Haniz, K. Ishizu, T. Matsumura, F. Kojima, and N. Ramli, "Low-altitude-

platform-based airborne iot network (lap-ain) for water quality monitoring in harsh tropical environment," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20034–20054, 2022.

[7] O. Sanders, "Efficient redactable signature and application to anonymous credentials," in *Public-Key Cryptography – PKC 2020* (A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, eds.), (Cham), pp. 628–656, Springer International Publishing, 2020.

[8] R. Shi, Y. Yang, Y. Li, H. Feng, G. Shi, H. H. Pang, and R. H. Deng, "Double issuer-hiding attribute-based credentials from tag-based aggregatable mercurial signatures," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 2585–2602, 2024.

[9] Q. Zhang, Y. Fu, J. Cui, D. He, and H. Zhong, "Efficient fine-grained data sharing based on proxy re-encryption in iiot," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5797–5809, 2024.

[10] S. Griffy, A. Lysyanskaya, O. Mir, O. Perez Kempner, and D. Slamanig, "Delegatable anonymous credentials from mercurial signatures with stronger privacy," in *Advances in Cryptology – ASIACRYPT 2024* (K.-M. Chung and Y. Sasaki, eds.), (Singapore), pp. 296–325, Springer Nature Singapore, 2025.

[11] D. Zeng, A. Badshah, S. Tu, M. Waqas, and Z. Han, "A security-enhanced ultra-lightweight and anonymous user authentication protocol for telehealthcare information systems," *IEEE Transactions on Mobile Computing*, pp. 1–13, 2025.

[12] 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*, vol. 19, pp. 7744–7758, 2024.

[13] 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*, CCS '23, (New York, NY, USA), p. 30–44, Association for Computing Machinery, 2023.

[14] O. Sanders, "Improving revocation for group signature with redactable signature," in *Public-Key Cryptography – PKC 2021* (J. A. Garay, ed.), (Cham), pp. 301–330, Springer International Publishing, 2021.

[15] 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*, vol. 19, pp. 2549–2564, 2024.

[16] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers," in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*, The Internet Society, 2019.

[17] R. Shi, H. Feng, Y. Yang, F. Yuan, Y. Li, H. H. Pang, and R. H. Deng, "Threshold attribute-based credentials with redactable signature," *IEEE Transactions on Services Computing*, vol. 16, no. 5, pp. 3751–3765, 2023.

[18] J. Liu, J. Yang, X. Huang, L. Xu, and Y. Xiang, "Privacy enhanced authentication for online learning healthcare systems," *IEEE Transactions on Services Computing*, vol. 17, no. 4, pp. 1670–1681, 2024.

[19] M. Sun, J. Lai, W. Wu, Y. Yang, C.-K. Chu, and R. H. Deng, "How to securely delegate and revoke partial authorization credentials," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–18, 2024.

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

[21] M. Zeng, J. Cui, Q. Zhang, H. Zhong, and D. He, "Efficient revocable cross-domain anonymous authentication scheme for iiot," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 996–1010, 2025.

[22] 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)*, pp. 773–789, 2023.

[23] O. Mir, D. Slamanig, and R. Mayrhofer, "Threshold delegatable anonymous credentials with controlled and fine-grained delegation," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 2312–2326, 2024.

[24] X. Ai, A. Badshah, S. Tu, M. Waqas, and I. Ahmad, "An improved ultra-lightweight anonymous authenticated key agreement protocol for wearable devices," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2025.

[25] Y. Wang, Y. Zhang, A. Ye, J. Shen, D. Wang, and Y. Xiang, "Anonymous and efficient (t, n)-threshold ownership transfer for cloud emrs auditing," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 1710–1723, 2025.

[26] B. Gong, C. Guo, C. Guo, C. Guo, Y. Sun, M. Waqas, and S. Chen, "Slim: A secure and lightweight multi-authority attribute-based signcryption scheme for iot," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1299–1312, 2024.

[27] F. Wang, J. Cui, Q. Zhang, D. He, C. Gu, and H. Zhong, "Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 1587–1604, 2024.

[28] D. Xie, J. Yang, B. Wu, W. Bian, F. Chen, and T. Wang, "An effectively applicable to resource constrained devices and semi-trusted servers authenticated key agreement scheme," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3451–3464, 2024.

[29] F. Wang, J. Cui, Q. Zhang, D. He, and H. Zhong, "Blockchain-based secure cross-domain data sharing for edge-assisted industrial internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3892–3905, 2024.

[30] L. Nguyen, "Accumulators from bilinear pairings and applications," in *Topics in Cryptology – CT-RSA 2005* (A. Menezes, ed.), (Berlin, Heidelberg), pp. 275–292, Springer Berlin Heidelberg, 2005.

[31] D. Derler, C. Hanser, and D. Slamanig, "Revisiting cryptographic accumulators, additional properties and relations to other primitives," in *Topics in Cryptology — CT-RSA 2015* (K. Nyberg, ed.), (Cham), pp. 127–144, Springer International Publishing, 2015.

[32] S. Srinivasan, I. Karantaidou, F. Baldimtsi, and C. Papamanthou, "Batching, aggregation, and zero-knowledge proofs in bilinear accumulators," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, (New York, NY, USA), p. 2719–2733, Association for Computing Machinery, 2022.

[33] J. Liu, J. Yang, W. Wu, X. Huang, and Y. Xiang, "Lightweight authentication scheme for data dissemination in cloud-assisted healthcare iot," *IEEE Transactions on Computers*, vol. 72, no. 5, pp. 1384–1395, 2023.

[34] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Topics in Cryptology - CT-RSA 2016* (K. Sako, ed.), (Cham), pp. 111–126, Springer International Publishing, 2016.

[35] J. Liu, J. Hou, X. Huang, Y. Xiang, and T. Zhu, "Secure and efficient sharing of authenticated energy usage data with privacy preservation," *Computers & Security*, vol. 92, p. 101756, 2020.

[36] D. Derler, H. C. Pöhls, K. Samelin, and D. Slamanig, "A general framework for redactable signatures and new constructions," in *Information Security and Cryptology - ICISC 2015* (S. Kwon and A. Yun, eds.), (Cham), pp. 3–19, Springer International Publishing, 2016.

[37] L. Zhang, F. Qiu, F. Hao, and H. Kan, "1-round distributed key generation with efficient reconstruction using decentralized cp-abe," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 894–907, 2022.

[38] I. Cascudo and B. David, "Publicly verifiable secret sharing over class groups and applications to dkg and yoso," in *Advances in Cryptology – EUROCRYPT 2024* (M. Joye and G. Leander, eds.), (Cham), pp. 216–248, Springer Nature Switzerland, 2024.

[39] Amovlab, "amov-lab/Prometheus," Jan. 2025. Available at https://github.com/amov-lab/Prometheus.