

Article

Expressive Attribute-Based Proxy Signature Scheme for UAV Networks

Lei He , Yong Gan * and Songhe Jin

School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450000, China

* Correspondence: ganyong@zzuli.edu.cn

Abstract

Unmanned aerial vehicle (UAV) networks have become an essential component of modern civilian and military infrastructures. However, the communication channels between UAVs and their control entities remain vulnerable to spoofing and message tampering attacks. Although conventional digital signature schemes can ensure message authentication and integrity, they often undermine the real-time responsiveness of UAV operations and fail to protect the privacy of signers. To address these limitations, we propose an expressive attribute-based proxy signature (EABPS) scheme tailored for UAV networks. The scheme enables fine-grained authorization and authentication, ensuring that only entities whose attributes satisfy a specified access structure can generate valid proxy signatures. Furthermore, the scheme preserves signer privacy by decoupling signatures from explicit identities. Comprehensive security analysis and extensive experimental evaluation demonstrate that the proposed EABPS scheme achieves strong security guarantees while offering improved computational efficiency and expressiveness, making it a practical solution for secure communication in UAV networks.

Keywords: expressive; attribute-based signature; proxy signature; unmanned aerial vehicle network

1. Introduction

Unmanned aerial vehicle (UAV) networks are emerging as a critical infrastructure for a wide range of civilian and military applications. A typical UAV network consists of aerial platforms equipped with sensors and communication modules, ground control stations that coordinate missions, and a command center responsible for making strategic decisions, and sometimes satellites or mobile relays that extend coverage. It is illustrated in Figure 1. Unlike conventional wireless networks, UAV networks are characterized by high mobility, rapidly changing topology, and intermittent connectivity, which result from the dynamic flight patterns of UAVs and the variability in wireless links [1]. These features make UAV networks highly flexible and capable of providing on-demand services in scenarios such as disaster response, environmental monitoring, and surveillance. However, the same characteristics also introduce unique challenges for secure and reliable communication. The limited energy, computation, and bandwidth resources of UAVs further complicate the design of security schemes.

Despite these advantages, UAV networks face significant security threats due to their exposure to open and often hostile environments. Communication links between UAVs and control entities are vulnerable to eavesdropping, spoofing, and message tampering, which may lead to mission failure or unauthorized control of UAVs. Ensuring the integrity and



Academic Editor: Raffaele Bruno

Received: 8 November 2025

Revised: 4 December 2025

Accepted: 8 December 2025

Published: 21 December 2025

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

authentication of transmitted commands is therefore a critical requirement, which can be achieved through the implementation of a digital signature scheme. Conventional digital signature schemes provide a foundation for securing these communications; however, they often rely on digital certificates or identity-based signatures, which may not be suitable for dynamic and heterogeneous UAV environments. In practice, it is sometimes necessary to delegate signing authority to intermediate nodes, such as ground control stations, to reduce communication delays and maintain operational continuity. Moreover, in scenarios where privacy preservation is essential, digital certificates and identity-based signatures may inadvertently reveal sensitive information about the command issuer. These limitations motivate the exploration of proxy signature schemes and attribute-based signature (ABS) schemes, which can jointly provide flexible delegation, fine-grained access control, and signer's privacy.

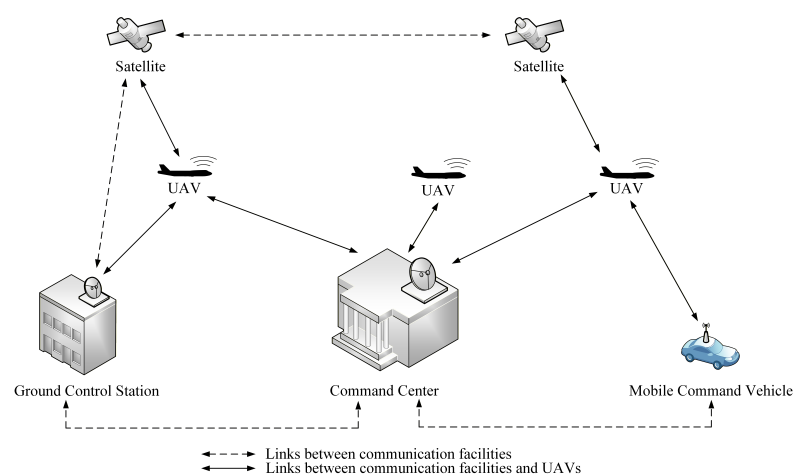


Figure 1. UAV network.

To address these challenges, we propose an expressive attribute-based proxy signature (EABPS) scheme specifically designed for UAV networks. Unlike traditional schemes that either focus solely on delegation or on attribute-based access control, our approach integrates both in a unified framework while enhancing expressiveness. In particular, the proposed scheme supports complex attribute policies that enable fine-grained authorization, ensuring that only entities whose attributes satisfy predefined conditions can act as proxy signers. This capability not only strengthens security but also accommodates the dynamic and heterogeneous nature of UAV networks. Furthermore, the scheme preserves signer anonymity by decoupling signatures from explicit identities, thereby mitigating privacy risks in sensitive applications. By balancing strong security guarantees with computational efficiency suitable for resource-constrained UAVs, the proposed scheme offers a practical solution for secure communications in UAV networks.

1.1. Problem Statement

In a UAV network, the command center usually holds the highest authority and is responsible for issuing instructions that guide UAVs to perform diverse tasks. However, direct communication between the command center and UAVs often faces practical limitations. Due to the wide deployment range of UAVs, communication links may be long-distance, intermittent, or subject to poor channel quality, making them highly vulnerable to interception and modification by adversaries. Although the command center and other infrastructure components may possess sufficient computational power and stable communication resources, UAVs themselves are constrained by limited energy, processing capacity, and memory, which amplifies the impact of insecure or delayed transmissions.

To ensure that UAVs execute only authentic commands, a digital signature scheme is indispensable. However, conventional digital signature schemes introduce two major challenges. First, the inherent delay in transmitting signed commands from a remote command center to UAVs can degrade real-time responsiveness, which is crucial in time-sensitive missions. Second, traditional signature schemes that use digital certificates and identity-based signature schemes couple authentication with user identity, exposing sensitive information about the signer and creating potential privacy risks.

In many deployment scenarios, it is desirable to temporarily delegate signing authority to nearby entities, such as ground control stations or mobile command units, so that UAVs can receive authenticated commands with reduced latency. At the same time, privacy-sensitive missions require that UAVs verify the authenticity of commands without learning the identity of the signer. These requirements highlight the necessity for a signature scheme that simultaneously supports secure delegation and privacy preservation. An attribute-based proxy signature (ABPS) scheme naturally addresses these needs by allowing for flexible delegation of signing rights while decoupling signatures from explicit identities.

1.2. Our Contributions

Motivated by the limitations of existing approaches, this paper introduces an ABPS scheme designed for UAV networks. Unlike prior works that mainly rely on threshold predicates, our scheme adopts a tree-based access structure, thereby providing greater expressiveness and flexibility. The key contributions of this work are summarized as follows:

- **Expressive access control:** In contrast to the previous works, our scheme employs a tree-based access structure that provides higher expressiveness than traditional threshold policies. This enhancement allows the delegation and signing policies to express multi-branch, hierarchical, and fine-grained authorization and authentication conditions that can not be represented by threshold structures. The use of a tree-based access structure not only increases the flexibility of authorization and authentication but also makes the scheme better suited for UAV network environments.
- **Integration of proxy signature schemes and ABS schemes:** Our scheme unifies the advantages of proxy signature schemes and attribute-based signature schemes, allowing a command center to securely delegate signing authority to a proxy entity while ensuring that only proxies with the required attributes can generate valid signatures.
- **Privacy preservation:** By binding signatures to attributes rather than explicit identities, our scheme ensures that the identities of both the original signer and the proxy signer remains hidden. This property is particularly important in sensitive UAV applications where anonymity of command issuers must be preserved.

In summary, the proposed scheme advances the state of the art by enabling expressive, privacy-preserving, and delegation of signing rights in UAV networks.

1.3. Organization of This Paper

The remainder of this paper is organized as follows: Section 2 reviews related work on security solutions for UAV networks and ABS schemes. Section 3 introduces the necessary preliminaries. Section 4 presents the formal model and security model of the proposed EABPS scheme. Section 5 details the design of the EABPS scheme for UAV networks. Section 6 provides a comprehensive analysis of the scheme, covering correctness, security, and efficiency. Finally, Section 7 concludes this paper and discusses possible directions for future research.

2. Related Work

2.1. Security Solutions for UAV Networks

Due to the open and dynamic nature of UAV networks, their communication links are highly vulnerable to a variety of attacks. Zhi et al. analyzed the security and privacy issues of UAVs and emphasized that the communication channels between UAVs and ground control stations are susceptible to attacks [2]. Samanth et al. highlighted the necessity of designing more appropriate digital signature schemes specifically tailored for UAV networks [3].

Khan et al. proposed an identity-based generalized signcryption scheme for flying ad hoc networks [4], but Din et al. demonstrated that this scheme is insecure and presented an improved scheme [5]. Wang et al. proposed an aggregate authentication scheme based on ID-based encryption for UAV cluster networks [6]. Lei et al. proposed an optimized lightweight identity security authentication protocol [7].

Various certificateless signature schemes have been explored in the UAV context. He et al. proposed a certificateless designated verifier proxy signature scheme [8], but Xu et al. later showed its vulnerability to impersonation attacks and suggested an efficient construction [9]. Qu and Zeng presented a certificateless proxy signcryption scheme in the standard model [10]. Sysoyev et al. proposed a certificateless group signature scheme for UAV communications in resource-constrained environments [11].

Li et al. proposed an online/offline revocable identity-based group signature scheme for UAVs [12]. To further enhance lightweight authentication, Liu et al. proposed a trustworthy message exchange scheme combining cryptography with trust management technologies [13]. Aissaoui et al. implemented post-quantum digital signature standards for UAVs and evaluated their performance [14]. Li et al. introduced a secure and reliable UAV network service architecture and a UAV cluster identity management module [15]. Khan et al. proposed a certificate-based proxy signature scheme for UAV communications, which uses hyperelliptic curve cryptography [16].

A comparison of representative security solutions is summarized in Table 1.

Table 1. Comparison of representative security solutions.

Security Solutions	Integrity	Authentication	Protection of Signer's Identity
Din et al. [5]	✓	✓	×
Qu et al. [10]	✓	✓	×
Khan et al. [16]	✓	✓	×

The comparison in Table 1 shows that existing security solutions for UAV networks generally provide integrity and authentication, which are fundamental requirements for secure communication. However, these schemes can not protect the identity of the signer. This means that sensitive information about the signer may be exposed, which is unacceptable in scenarios that require anonymity or privacy protection. Therefore, although these schemes are effective for basic security services, they do not meet the requirement of identity protection in UAV environments.

2.2. Attribute-Based Signature Schemes

ABS has been widely investigated as a promising primitive for providing fine-grained authentication and privacy protection. Maji et al. first formalized ABS and presented a concrete construction in the generic group model [17]. Since then, many extensions have been proposed. Su et al. proposed expressive attribute-based signature scheme [18]. This work introduced the idea of using expressive access structures to control signing

capabilities, and it supports signer privacy and unforgeability. Liu et al. constructed a revocable and comparable lattice-based ABS scheme from lattices [19]. In response to the security risks posed by key leakage, Wu et al. proposed a puncturable ABS scheme [20].

ABS schemes with a single attribute authority suffer from centralization issues. To address this issue, Li et al. proposed a multi-authority ABS scheme [21]. Su et al. proposed a distributed attribute-based signature scheme that contains multiple attribute authorities [22].

Outsourcing has been used to reduce computational costs. Chen et al. presented two outsourced ABS schemes [23]. Tao et al. proposed a lightweight decentralized ABS scheme, which outsources complex computation of the signer to a cloud server [24]. Xiong et al. developed a server-aided ABS scheme with expressive access structure [25]. Huang and Lin proposed a server-aided ABS scheme with perfect anonymity [26]. Li et al. proposed a decentralized server-aided ABS scheme in the Internet of things [27].

Beyond ABS, the ABPS paradigm has been introduced to enable delegation with fine-grained attribute-based control. Liu et al. defined the ABPS model and proposed an initial construction [28]. He et al. subsequently designed ABPS schemes for UAV networks, demonstrating their feasibility and security [29], and further extended them with flexible threshold predicates [30]. He et al. also proposed an efficient threshold ABPS scheme for UAV networks [31]. While these works represent important progress, existing ABPS schemes are still largely constrained to threshold-based access structures, which limits their ability to capture complex and dynamic authorization requirements in the UAV environments. It is important to design an expressive ABPS scheme that leverages tree-based access structures in order to achieve fine-grained, flexible, and privacy-preserving signature suitable for UAV networks.

A comparison of representative ABPS schemes is summarized in Table 2.

Table 2. Comparison of representative ABPS schemes.

Security Solutions	Integrity	Authentication	Protection of Signer's Identity	Access Structure
ABPS-HL1 [29]	✓	✓	✓	Threshold
ABPS-HL2 [30]	✓	✓	✓	Threshold
t-ABPS [31]	✓	✓	✓	Threshold

Table 2 shows that existing ABPS schemes already provide integrity, authentication, and protection of the signer's identity. However, these schemes adopt threshold-based access structures, which limit their expressiveness and prevent support for more fine-grained authorization and authentication policies. Our EABPS adopts a tree-based access structure, which allows for more flexible and expressive delegation and signing policies and is better suited for complex UAV application scenarios.

In summary, the existing security solutions for UAV networks mainly focus on ensuring integrity and authentication but lack protection of signer's identity. Traditional ABPS schemes already provide identity protection, yet they rely on threshold-based access structures that limit their flexibility. These structures cannot represent complex or hierarchical authorization and authentication requirements that often arise in UAV applications. Our EABPS scheme addresses these limitations by adopting a tree-based access structure with higher expressiveness. This enables fine-grained control of signing rights while preserving signer identity privacy, thus providing a more powerful and suitable solution for UAV environments.

3. Preliminaries

3.1. Complexity Assumption

Definition 1 (Computational Diffie-Hellman (CDH) Problem). Let G be a cyclic group of prime order q with generator g . For unknown integers $x, y \in \mathbb{Z}_q^*$, the CDH problem is to compute g^{xy} given the tuple (g, g^x, g^y) .

3.2. Lagrange Interpolation

Let S be a set of d distinct points in the finite field $\mathbb{Z}_q$. Any polynomial $t(x)$ of degree at most $d - 1$ over $\mathbb{Z}_q$ is uniquely determined by its values $\{t(j) \mid j \in S\}$. Specifically, for any point $i \in \mathbb{Z}_q$, the value $t(i)$ can be reconstructed using the Lagrange interpolation formula $t(i) = \sum_{j \in S} t(j) \cdot \Delta_{j,S}(i)$ where Lagrange interpolation coefficient $\Delta_{j,S}(i) = \prod_{\eta \in S, \eta \neq j} \frac{i - \eta}{j - \eta}$.

4. Formal Model and Security Model

4.1. Formal Model

Our EABPS scheme involves four distinct entities within the UAV network architecture: an attribute authority (AA), an original signer (OS), a proxy signer (PS), and a verifier. It is assumed that the OS and PS satisfy the same access structure.

- AA: This fully trusted entity is responsible for initializing the public parameters and generating private keys for OS and PS based on their respective attribute sets.
- OS: This entity, such as a command center, holds the primary authority. The OS can delegate its signing rights to another entity for a specific purpose, which is defined in a warrant. The OS generates a valid delegation only if its attributes satisfy the access structure.
- PS: This entity, for instance, a ground control station, is authorized by the OS to sign messages on its behalf. The PS generates a valid proxy signature only if its attributes satisfy the access structure.
- Verifier: This entity, typically a UAV, validates a received proxy signature to confirm the authentication and integrity of a command.

The formal model of the proposed EABPS scheme consists of seven algorithms: Algorithms 1–7. Each algorithm is responsible for a specific phase in the system. To improve clarity, each algorithm is presented in a graphical step-based format.

Algorithm 1: $Setup(1^\lambda)$

Input: Security parameter 1^λ ;

Generate public parameters $params$ and master secret key MSK ;

Output: $(params, MSK)$;

Algorithm 2: $KGPr(params, MSK, \omega)$

Input: Public parameters, master secret key, attribute set ω ;

Generate private key sk_ω for user with attribute set ω ;

Output: sk_ω ;

Algorithm 3: $KGPub(params, MSK, \Gamma)$

Input: Public parameters, master secret key, attribute tree Γ ;

Derive corresponding public information based on Γ ;

Output: pk_Γ ;

Algorithm 4: $DelGen(params, sk_{\omega_o}, w)$

Input: Public parameters, original signer's private key sk_{ω_o} , warrant w ;
 Generate delegation based on (sk_{ω_o}, w) ;
Output: Delegation σ_o ;

Algorithm 5: $DelVer(params, w, \sigma_o, pk_{\Gamma})$

Input: Public parameters, warrant w , delegation σ_o , public key pk_{Γ} ;
 Verify the correctness and validity of delegation σ_o ;
Output: Accept or Reject;

Algorithm 6: $Sign(params, sk_{\omega_p}, \sigma_o, w, m)$

Input: Public parameters, proxy signer's private key sk_{ω_p} , delegation σ_o , warrant w , message m ;
 Generate proxy signature with the authorized signing capability;
Output: Proxy signature σ_p ;

Algorithm 7: $Verify(params, w, m, \sigma_p, pk_{\Gamma})$

Input: Public parameters, warrant w , message m , proxy signature σ_p , public key pk_{Γ} ;
 Verify correctness of the signature;
Output: Accept or Reject;

4.2. Security Model

The security of our proposed EABPS scheme is analyzed based on the standard notion of existential unforgeability. We formalize this property through a security game played between a challenger $\mathcal{C}$ and a probabilistic polynomial time (PPT) adversary $\mathcal{A}$. Following the security models for proxy signatures, we mainly consider the two types of adversaries, $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$, which reflect realistic threats in a proxy signature scheme [32].

To make the threat model more explicit, we further describe the capabilities of both adversaries. Both $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ are modeled as PPT adversaries and are allowed to access all public parameters of the system. They may intercept, replay, or modify transmitted messages in the communication channel. They are also allowed to query some oracles.

The $\mathcal{A}_{II}$ adversary represents a malicious proxy signer or an external attacker who has compromised the proxy signer's private key (sk_p). However, the original signer's private key (sk_o) remains secure. Therefore, $\mathcal{A}_{II}$ can not generate a valid delegation without access to sk_o . This adversary's goal is to forge a valid proxy signature for which it was not legitimately authorized by the original signer.

The $\mathcal{A}_{III}$ adversary represents a malicious original signer or an external attacker who has compromised the sk_o . In this scenario, the sk_p is considered secure. Therefore, $\mathcal{A}_{III}$ can not generate signatures that require sk_p . This adversary can attempt to misuse its knowledge of sk_o to generate unauthorized delegation, but it remains incapable of producing a valid proxy signature that appears to originate from an honest proxy signer without the corresponding sk_p .

These clarifications ensure that both adversaries are modeled with realistic abilities. Each adversary has partial key exposure, full access to public system components, and full control over the communication channel.

4.2.1. Existential Unforgeability

We define the security notion of existential unforgeability under a selective-policy and chosen-message attack (EUF-SP-CMA). The security is characterized by a game between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ (who can be either $\mathcal{A}_{II}$ or $\mathcal{A}_{III}$).

The game proceeds as follows for a given adversary type:

Initialization: The adversary $\mathcal{A}$ first selects and declares a challenge policy Γ^* that it intends to target for the forgery.

Setup: The challenger $\mathcal{C}$ runs the *Setup* algorithm to generate the public parameters *params* and the master secret key *MSK*. Depending on the adversary type:

- If $\mathcal{A}$ is $\mathcal{A}_{II}$, $\mathcal{C}$ generates the key pairs for the OS and PS. It provides $\mathcal{A}$ with *params* and the proxy signer's private key, sk_p . The OS's private key sk_o is kept secret.
- If $\mathcal{A}$ is $\mathcal{A}_{III}$, $\mathcal{C}$ similarly generates the key pairs. It provides $\mathcal{A}$ with *params* and the original signer's private key, sk_o . The PS's private key sk_p is kept secret.

Queries: The adversary $\mathcal{A}$ is granted access to a set of oracles. It can make a polynomially bounded number of queries to $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$. It can also make a polynomially bounded number of queries:

- Private Key Generation Oracle ($\mathcal{O}_{KGPri}$): $\mathcal{A}$ can request the private key for any attribute set ω .
- Delegation Generation Oracle ($\mathcal{O}_{DG}$): (Only accessible to $\mathcal{A}_{II}$) $\mathcal{A}$ can request a delegation on a warrant w and a policy Γ .
- Signing Oracle ($\mathcal{O}_{Sign}$): $\mathcal{A}$ can request a valid proxy signature on message m with warrant w and policy Γ .

Forgery: Finally, $\mathcal{A}$ outputs a tuple (w^*, m^*, σ^*) corresponding to the challenge policy Γ^* .

The adversary $\mathcal{A}_{II}$ wins the game if the forgery meets the following conditions:

1. $\mathcal{A}_{II}$ did not query any ω^* satisfying $\Gamma^*(\omega^*) = 1$ to $\mathcal{O}_{KGPri}$.
2. $\mathcal{A}_{II}$ did not query the pair (w^*, Γ^*) to $\mathcal{O}_{DG}$.
3. $\mathcal{A}_{II}$ did not query the tuple (m^*, w^*, Γ^*) to $\mathcal{O}_{Sign}$.
4. The signature σ^* is valid.

The adversary $\mathcal{A}_{III}$ wins the game if the forgery meets the following conditions:

1. $\mathcal{A}_{III}$ did not query any ω^* satisfying $\Gamma^*(\omega^*) = 1$ to $\mathcal{O}_{KGPri}$.
2. $\mathcal{A}_{III}$ did not query the tuple (m^*, w^*, Γ^*) to $\mathcal{O}_{Sign}$.
3. The signature σ^* is valid.

Definition 2 (EUF-SP-CMA). *An ABPS scheme provides EUF-SP-CMA if no PPT adversary $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ can win the aforementioned game with a non-negligible probability.*

4.2.2. Attribute Privacy for Signers

The EABPS scheme involves two types of signers: the original signer, who generates a delegation on a warrant w , and the proxy signer, who generates the final proxy signature on a message–warrant pair (m, w) . We define the attribute privacy for an original signer (APOS) by a game between the challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$.

Since a delegation functions as the original signer's signature on w , we should protect the attribute privacy of the original signer. The goal is to guarantee that a delegation reveals no information about the original signer's attribute set beyond the fact that it satisfies the access structure Γ . The game is defined as follows:

$\mathcal{C}$ runs $Setup(1^\lambda)$ and gives $params$ to $\mathcal{A}$. $\mathcal{A}$ may query $\mathcal{O}_{KGPri}$ and $\mathcal{O}_{DG}$.

$\mathcal{A}$ chooses two attribute sets ω_0, ω_1 and a warrant w such that $\Gamma(\omega_0) = \Gamma(\omega_1) = 1$. $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$ and computes σ_o by using $DelGen$ and sends σ_o to $\mathcal{A}$. Finally $\mathcal{A}$ outputs a guess b' and wins if $b' = b$.

The advantage of $\mathcal{A}$ is defined by $\mathbf{Adv}_{\mathcal{A}}^{APOS} = \left| \Pr[b' = b] - \frac{1}{2} \right|$.

Definition 3 (APOS). The EABPS scheme provides attribute privacy for an original signer if $\mathbf{Adv}_{\mathcal{A}}^{APOS} = \left| \Pr[b' = b] - \frac{1}{2} \right|$ is negligible.

Similarly, we define the attribute privacy for a proxy signer (APPS) by a game between the challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$. The goal is to guarantee that a signature reveals no information about the proxy signer's attribute set beyond the fact that it satisfies the access structure Γ . The game is defined as follows:

$\mathcal{C}$ runs $Setup(1^\lambda)$ and gives $params$ to $\mathcal{A}$. $\mathcal{A}$ may query $\mathcal{O}_{KGPri}$, $\mathcal{O}_{DG}$ and $\mathcal{O}_{Sign}$.

$\mathcal{A}$ chooses two attribute sets ω_0, ω_1 and a message m such that $\Gamma(\omega_0) = \Gamma(\omega_1) = 1$. $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$ and computes σ_p by using $Sign$ and sends σ_p to $\mathcal{A}$. Finally $\mathcal{A}$ outputs a guess b' and wins if $b' = b$.

The advantage of $\mathcal{A}$ is defined by $\mathbf{Adv}_{\mathcal{A}}^{APPS} = \left| \Pr[b' = b] - \frac{1}{2} \right|$.

Definition 4 (APPS). The EABPS scheme provides attribute privacy for a proxy signer if $\mathbf{Adv}_{\mathcal{A}}^{APPS} = \left| \Pr[b' = b] - \frac{1}{2} \right|$ is negligible.

The EABPS scheme achieves attribute privacy for signers (APS) if it satisfies both APOS and APPS. This ensures that neither the delegation nor the proxy signature leaks the attribute set used by the signer.

5. EABPS Scheme for UAV Networks

5.1. Overview of the EABPS Scheme

In the proposed EABPS scheme, there are four main entities: an AA, an OS, a PS, and a verifier. The AA is responsible for system initialization and key distribution. The command center acts as the OS and holds an attribute set ω_o together with the corresponding private key sk_o . The ground control station functions as the PS, equipped with its own attribute set ω_p and private key sk_p . Finally, the UAV plays the role of the verifier, receiving commands and verifying their integrity and authentication. Unlike threshold-based constructions, the proposed EABPS scheme employs a tree-based access structure Γ , which provides enhanced expressiveness. Both the OS and PS are associated with attributes that must satisfy Γ in order to generate valid signatures. The overall process of our EABPS scheme is illustrated as Figure 2 and can be summarized in the following phases:

(1) The AA executes the *Setup* algorithm to generate public parameters and the master secret key. It then issues private keys corresponding to the attribute sets of the command center and ground control station.

(2) The command center, acting as the OS, generates a warrant w and uses its private key sk_o to produce a delegation bound to the access structure Γ . The delegation is then securely transmitted to the ground control station.

(3) The ground control station, acting as the PS, verifies the validity of the received delegation. Only if the delegation is correct does the PS proceed.

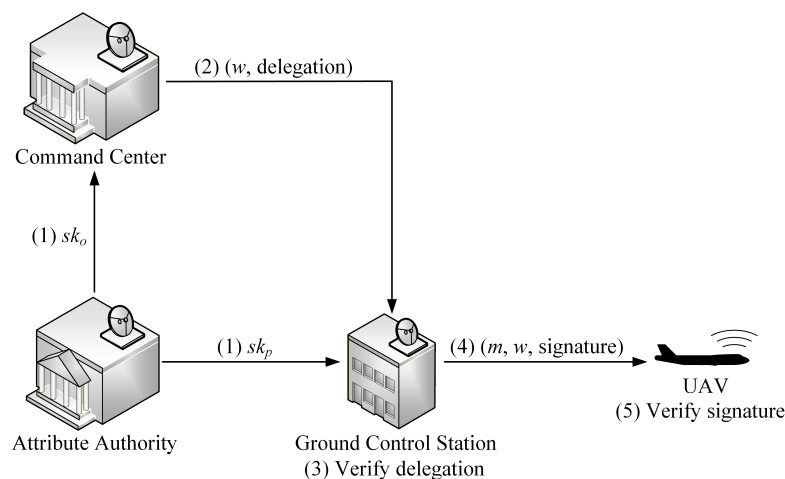


Figure 2. Overall EABPS scheme.

In our EABPS scheme, the delegation of signing rights is achieved through a warrant and a delegation, rather than the transfer of any secret key. This design ensures that the original signer's private key is never exposed during the delegation process.

Specifically, the original signer uses its private key to compute a delegation σ_o that is cryptographically bound to a warrant w , which specifies the delegated permissions in a fine-grained manner. The delegation is generated using the *DelGen* algorithm, and it contains no information that allows the proxy signer to recover the original signer's private key.

The delegation σ_o is transmitted to the proxy signer. Since σ_o cannot be used independently to produce signatures and functions only when combined with the proxy signer's own private key in the *Sign* algorithm, the delegation mechanism prevents misuse and ensures that the signing capability strictly follows the limitations defined in the warrant.

Therefore, the delegation process securely transfers signing rights without transferring any secret signing key.

(4) The PS signs a command message m by combining its private key sk_p with the delegation. The resulting proxy signature is associated with the attributes satisfying Γ , ensuring that only authorized proxies can produce valid signatures.

(5) The UAV, as the verifier, checks the validity of the proxy signature. If the verification succeeds, the UAV accepts the command as authentic.

Through these phases, the EABPS scheme enables secure delegation and proxy signing in UAV networks.

5.2. EABPS Scheme

In this subsection, we present the construction of the EABPS scheme. The proposed EABPS scheme is composed of the following algorithms.

5.2.1. Setup

Given the security parameter λ , the algorithm generates two cyclic groups (G_1, G_2) of prime order q with generator $g \in G_1$. A bilinear map $e : G_1 \times G_1 \rightarrow G_2$ is defined. The authority selects a random $\alpha \in \mathbb{Z}_q$, sets $g_1 = g^\alpha$, and chooses a random $g_2 \in G_1$. It then computes $Z = e(g_1, g_2)$. Three hash functions are defined: $H_1 : \{0, 1\}^* \rightarrow G_1$ for attributes, $H_2 : \{0, 1\}^* \rightarrow G_1$ for warrants, and $H_3 : \{0, 1\}^* \rightarrow G_1$ for messages. The public parameters are $params = (q, G_1, G_2, e, g, g_1, g_2, Z, H_1, H_2, H_3)$, and the master secret key is $MSK = \alpha$.

5.2.2. KGPr

The attribute authority generates private key sk_{ω_o} for the original signer with an attribute set ω_o . Given the master secret key α and an attribute set ω_o , the authority chooses $r_o \in \mathbb{Z}_q$ at random and computes $d_o = g_2^{r_o + \alpha}$. For each attribute $i_o \in \omega_o$, it selects $r_{i_o} \in \mathbb{Z}_q$ and computes

$$d_{i_o0} = g_2^{r_o/\alpha} \cdot H_1(i_o)^{r_{i_o}}, \quad d_{i_o1} = g^{r_{i_o}}.$$

The private key for attribute set ω_o is $sk_{\omega_o} = (d_o, \{(d_{i_o0}, d_{i_o1})\}_{i_o \in \omega_o})$.

In the same way, the attribute authority generates private key sk_{ω_p} for the proxy signer with an attribute set ω_p . The authority chooses $r_p \in \mathbb{Z}_q$ at random and computes $d_p = g_2^{r_p + \alpha}$. For each attribute $i_p \in \omega_p$, it selects $r_{i_p} \in \mathbb{Z}_q$ and computes

$$d_{i_p0} = g_2^{r_p/\alpha} \cdot H_1(i_p)^{r_{i_p}}, \quad d_{i_p1} = g^{r_{i_p}}.$$

The private key for attribute set ω_p is $sk_{\omega_p} = (d_p, \{(d_{i_p0}, d_{i_p1})\}_{i_p \in \omega_p})$.

5.2.3. KGPub

For an access tree Γ , the root node is assigned a polynomial p_{root} with $p_{root}(0) = \alpha$. For each internal node x with threshold k_x , a random polynomial p_x of degree $k_x - 1$ is generated such that $p_x(0) = p_{parent(x)}(\text{index}(x))$. For each leaf node x associated with attribute $i = \text{att}(x)$, the authority computes $D_i = g^{p_x(0)}$, $h_i = H_1(i)^{p_x(0)}$. The public key for access tree Γ is $gpk_{\Gamma} = \{D_i, h_i\}_{i \in \text{leaves}(\Gamma)}$.

5.2.4. DelGen

The original signer holding sk_{ω_o} selects $s_o \in \mathbb{Z}_q$ and computes

$$\sigma_{o,0} = H_2(w)^{s_o} \cdot d_o, \quad \sigma'_{o,0} = g^{s_o}.$$

Let ω^* denote the attribute set associated with leaves in Γ and $|\omega^*| = n$. For each attribute $i_o \in \omega^*$, it chooses $r'_{i_o} \in \mathbb{Z}_q$ and sets

$$(\sigma_{i_o0}, \sigma_{i_o1}) = \begin{cases} (d_{i_o0} \cdot H_1(i_o)^{r'_{i_o}}, d_{i_o1} \cdot g^{r'_{i_o}}), & i_o \in \omega_o \cap \omega^*, \\ (H_1(i_o)^{r'_{i_o}}, g^{r'_{i_o}}), & i_o \in \omega^* / \omega_o \cap \omega^*. \end{cases}$$

The delegation is $\sigma_o = (\sigma_{o,0}, \{\sigma_{i_o0}, \sigma_{i_o1}\}_{i_o \in \Gamma}, \sigma'_{o,0})$.

5.2.5. DelVer

Given a warrant w and a delegation σ_o , the proxy signer recursively runs *VerNode* over the original signer's leaf pairs to obtain $F_{o,root}$. The delegation is accepted if

$$e(g, \sigma_{o,0}) = F_{o,root} \cdot e(H_2(w), \sigma'_{o,0}) \cdot Z,$$

and rejected otherwise.

The function *VerNode*($\sigma_o, gpk_{\Gamma}, x$) is a recursive procedure used to determine whether the original signer's attributes satisfy the access tree Γ . It takes as input the delegation σ_o , the public key $gpk_{\Gamma} = \{D_i, h_i\}_{i \in \omega^*}$, and a node x in Γ , and outputs a group element $F_{o,root} \in G_2$ or $\perp$.

If x is a leaf node, let $i_o = \text{att}(x)$ and compute $F_x = \frac{e(\sigma_{i_o0}, D_{i_o})}{e(\sigma_{i_o1}, h_{i_o})}$. If $F_x = 1$, output $\perp$; otherwise, output F_x .

If x is a non-leaf node, for each child node z of x , recursively compute $F_z = \text{VerNode}(\sigma_o, \text{gpk}_\Gamma, z)$. Let S_x be any subset of child nodes such that $|S_x| = k_x$ and $F_z \neq \perp$ for all $z \in S_x$. If no such subset exists, output $\perp$. Otherwise, let $S'_x = \{\text{index}(z) : z \in S_x\}$ and compute $F_x = \prod_{z \in S_x} F_z^{\Delta_{j, S'_x}(0)}$, where $j = \text{index}(z)$ and $\Delta_{j, S'_x}(0)$ denotes the Lagrange interpolation coefficient at point 0. Finally, output F_x .

When the recursion reaches the root node, the algorithm outputs

$$F_{o, \text{root}} = e(g, g_2)^{r_o / \alpha \cdot p_{\text{root}}(0)} = e(g, g_2)^{r_o}.$$

5.2.6. Sign

The proxy signer selects $s_p \in \mathbb{Z}_q$ and computes

$$\sigma_{p,0} = \sigma_{o,0} \cdot H_3(m)^{s_p} \cdot d_p, \quad \sigma'_{p,0} = g^{s_p}.$$

For each attribute $i_p \in \omega^*$, the signer chooses $r''_{i_p} \in \mathbb{Z}_q$ and sets

$$(\sigma_{i_p,0}, \sigma_{i_p,1}) = \begin{cases} (d_{i_p,0} \cdot H_1(i_p)^{r''_{i_p}}, d_{i_p,1} \cdot g^{r''_{i_p}}), & i_p \in \omega_p \cap \omega^*, \\ (H_1(i_p)^{r''_{i_p}}, g^{r''_{i_p}}), & i_p \in \omega^* / \omega_p \cap \omega^*. \end{cases}$$

The proxy signature σ_p is $\sigma_p = (\sigma_{p,0}, \{\sigma_{i_p,0}, \sigma_{i_p,1}\}_{i_p \in \Gamma}, \sigma'_{p,0}, \{\sigma_{i_o,0}, \sigma_{i_o,1}\}_{i_o \in \Gamma}, \sigma'_{o,0})$. The proxy signer sends (m, w, σ_p) to the verifier.

5.2.7. Verify

The verifier runs *VerNode* on the original signer's and proxy signer's leaf pairs to obtain $F_{o, \text{root}}$ and $F_{p, \text{root}}$. The signature is accepted if

$$e(g, \sigma_{p,0}) = F_{p, \text{root}} \cdot e(H_3(m), \sigma'_{p,0}) \cdot F_{o, \text{root}} \cdot e(H_2(w), \sigma'_{o,0}) \cdot Z^2,$$

and rejected otherwise.

6. EABPS Scheme Analysis

6.1. Correctness

We prove the correctness of our scheme in two steps. First, we verify that a delegation produced by *DelGen* is accepted by *DelVer*. From the function *VerNode*, we can compute $F_{o, \text{root}} = e(g, g_2^{r_o})$. Then, we can proceed as follows:

$$\begin{aligned} e(g, \sigma_{o,0}) &= e(g, H_2(w)^{s_o} g_2^{r_o + \alpha}) \\ &= e(g, H_2(w)^{s_o}) \cdot e(g, g_2^{r_o}) \cdot e(g, g_2^\alpha) \\ &= F_{o, \text{root}} \cdot e(H_2(w), \sigma'_{o,0}) \cdot Z \end{aligned}$$

Therefore a correctly formed delegation always passes *DelVer*, completing the correctness proof for the delegation phase.

Second, we prove that a proxy signature produced by *Sign* is accepted by *Verify*. From the function *VerNode*, we can also compute $F_{p,root} = e(g, g_2^{r_p})$. Then, we can proceed as follows:

$$\begin{aligned} e(g, \sigma_{p,0}) &= e(g, \sigma_{o,0} H_3(m)^{s_p} g_2^{r_p + \alpha}) \\ &= e(g, \sigma_{o,0}) \cdot e(g, H_3(m)^{s_p}) \cdot e(g, g_2^{r_p}) \cdot e(g, g_2^{\alpha}) \\ &= F_{o,root} \cdot e(H_2(w), \sigma'_{o,0}) \cdot Z \cdot F_{p,root} \cdot e(H_3(m), \sigma'_{p,0}) \cdot Z \\ &= F_{p,root} \cdot e(H_3(m), \sigma'_{p,0}) \cdot F_{o,root} \cdot e(H_2(w), \sigma'_{o,0}) \cdot Z^2 \end{aligned}$$

Hence, a proxy signature correctly formed as above will be accepted by *Verify*, completing the correctness proof for the proxy signing phase.

6.2. Security Analysis

6.2.1. Unforgeability

In this subsection we give a formal proof that the proposed EABPS scheme has EUF-SP-CMA against $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ in the random oracle model.

Theorem 1. *In the random oracle model, the proposed EABPS scheme has EUF-sP-CMA against $\mathcal{A}_{II}$, assuming the CDH assumption holds in group G_1 .*

Proof. It is assumed that there exists a PPT adversary $\mathcal{A}_{II}$ that breaks the EABPS scheme with non-negligible advantage ϵ . It is assumed that $\mathcal{A}_{II}$ makes q_{H_1} , q_{H_2} and q_{H_3} queries to the random oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, q_K queries to $\mathcal{O}_{KGPri}$, q_D queries to $\mathcal{O}_{DG}$ and q_S queries to $\mathcal{O}_{Sign}$. We use $\mathcal{A}_{II}$ to construct a PPT algorithm $\mathcal{B}$ that solves the CDH problem with non-negligible probability ϵ' . Given a CDH instance $(g, X = g^x, Y = g^y)$ in G_1 , $\mathcal{B}$ computes g^{xy} without knowing x or y . We denote $g_1 = X = g^x$ and $g_2 = Y = g^y$, and we let $Z = e(g_1, g_2) = e(g, g)^{xy}$.

$\mathcal{A}_{II}$ declares the challenge access policy (attribute tree) Γ^* . Let ω^* denote the set of attributes labeling the leaves of Γ^* . $\mathcal{A}_{II}$ randomly chooses $\nu \in \{1, 2, \dots, q_{H_2}\}$ and $\xi \in \{1, 2, \dots, q_{H_3}\}$.

The adversary $\mathcal{A}_{II}$ queries some oracles. If the query value resides in the relevant list, the oracle returns the corresponding entry; otherwise, it generates the corresponding output and records both the query value and the output in the list. The random numbers $a_i, b_i, a'_{i'}, c'_{i'}, a''_{i''}, c''_{i''}, r_i$ are chosen from $\mathbb{Z}_q$.

$\mathcal{O}_{H_1}$: $\mathcal{A}_{II}$ queries $\mathcal{O}_{H_1}$ with the input i . $\mathcal{B}$ outputs g^{b_i} if $i \in \omega^*$, and $g_1^{-b_i} g^{a_i}$ otherwise.

$\mathcal{O}_{H_2}$: $\mathcal{A}_{II}$ queries $\mathcal{O}_{H_2}$ with the input $w_{i'}$. $\mathcal{B}$ outputs $g^{a'_{i'}}$ if $i' = \nu$, and $g_1^{c'_{i'}} g^{a'_{i'}}$ otherwise.

$\mathcal{O}_{H_3}$: $\mathcal{A}_{II}$ queries $\mathcal{O}_{H_3}$ with the input $m_{i''}$. $\mathcal{B}$ outputs $g^{a''_{i''}}$ if $i'' = \xi$, and $g_1^{c''_{i''}} g^{a''_{i''}}$ otherwise.

$\mathcal{O}_{KGPri}$: $\mathcal{A}_{II}$ queries $\mathcal{O}_{KGPri}$ with the attribute set ω which does not satisfy the challenge access structure, i.e., $\Gamma^*(\omega) \neq 1$.

For any validly queried ω , the set $\omega^* \cap \omega$ also fails to satisfy the access structure, which means $\Gamma^*(\omega^* \cap \omega) \neq 1$. It is assumed that S is the set of attributes corresponding to the leaf nodes in the access tree Γ^* . It satisfies the access structure Γ^* and $\omega^* \cap \omega \subseteq S$. To construct the private key, $\mathcal{B}$ first selects $d = g_2^{r+x}$ uniformly at random.

For $i \in S$, the key components d_{i0} and d_{i1} are then simulated as $d_{i0} = g_2^{\frac{r}{x}} H_1(i)^{r_i}$ and $d_{i1} = g^{r_i}$.

For $i \notin S$, $\mathcal{B}$ sets $r_i = \frac{\Delta_{0,S}(i)}{b_i} y + r'_i$. The key components d_{i0} and d_{i1} are then simulated as $(d_{i0} = g_2^{\frac{\Delta_{0,S}(i)a_i}{b_i} + \frac{r}{x}} g^{-xy\Delta_{0,S}(i)} (H_1(i))^{r'_i}$ and $d_{i1} = g_2^{\frac{\Delta_{0,S}(i)}{b_i}} g^{r'_i}$).

The core idea is that the private key for Γ^* is generated by assigning a random polynomial $p(\cdot)$ to each node from the root downward. The root satisfies $p_{\text{root}}(0) = x$, while for any other node, we set $p_{\text{node}}(0) = p_{\text{parent}(\text{node})}(\text{index}(\text{node}))$ and select the remaining coefficients randomly.

$\mathcal{O}_{DG}$: To simulate $(g_2^x g_2^r H_2(w)^s, \{g_2^{\frac{r}{c_i}} H_1(i)^{r_i}, g_1^{r_i}\}_{i \in \omega^*}, g^s)$, $\mathcal{B}$ sets $s = \frac{-y}{c_i} + s'$ where s' is a random value and $s' \in \mathbb{Z}_q$. It simulates $g_2^x g_2^r H_2(w)^s = g_2^x (g_1^{c_i} g^{a_i})^{s'} g_2^{-\frac{a_i}{c_i}}$ and $g^s = g_2^{\frac{-1}{a_i}} g^{s'}$.

$\mathcal{O}_{\text{Sign}}$: If $H_2(\omega_{i'}) = g^{a_{i'}}$ and $H_3(m_{i''}) = g^{a_{i''}}$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ simulates and outputs the signature.

Finally, $\mathcal{A}_{II}$ outputs a forged signature σ_p^* on m^* and w^* with Γ^* . If $H_2(\omega^*) \neq g^{a_v}$ or $H_3(m^*) \neq g^{a_v'}$, $\mathcal{B}$ aborts. $\mathcal{B}$ constructs the solution to the CDH problem by processing the forged signature σ_p^* . This is achieved via a recursive algorithm that operates bottom-up on the challenge policy tree Γ^* .

Beginning with the leaf nodes, the algorithm leverages the structure of the forged signature components $\{\sigma_{i_0}^*, \sigma_{i_1}^*\}$ and $\{\sigma_{i_p0}^*, \sigma_{i_p1}^*\}$ to compute intermediate values. For each interior node, it then uses Lagrange interpolation to combine the values derived from its child nodes. This process continues until the root node is reached, at which point the term $R_{\text{root}} = g_2^{-r}$ is successfully isolated.

Finally, by combining this isolated term with the signature components $\sigma_{o,0}^*, \sigma_{p,0}^*$ and $\sigma_{p,0}^{\prime*}$, the simulator is able to cancel all remaining unknown exponents. This allows for the direct computation of the CDH solution:

$$g^{xy} = R_{\text{root}} \cdot \left(\frac{\sigma_{p,0}^*}{(\sigma_{p,0}^{\prime*})^{a_v'} (\sigma_{o,0}^*)^{a_v'}} \right)^{\frac{1}{2}} \quad (1)$$

$\mathcal{B}$ solves the CDH problem with probability $\epsilon' \approx \frac{\epsilon}{(q_{H_2})^2 q_{H_3}}$. Therefore, this EABPS scheme has EUF-SP-CMA against $\mathcal{A}_{II}$. $\square$

Theorem 2. *In the random oracle model, the proposed EABPS scheme has EUF-SP-CMA secure against $\mathcal{A}_{III}$, assuming the CDH assumption holds in group G_1 .*

Proof. The proof is similar to the proof Theorem 1. We compute $\epsilon' \approx \frac{\epsilon}{q_{H_2} q_{H_3}}$. Therefore, this scheme has EUF-SP-CMA against $\mathcal{A}_{III}$. $\square$

According to the Theorems 1 and 2, our EABPS scheme has EUF-SP-CMA.

6.2.2. Attribute Privacy for Signer

Theorem 3. *Our EABPS scheme achieves attribute privacy for original signers.*

Proof. Let ω_0 and ω_1 be two distinct attribute sets such that $\Gamma(\omega_0) = \Gamma(\omega_1) = 1$. We show that the delegations generated from these sets are computationally indistinguishable.

In the *DelGen* algorithm, the original signer introduces fresh randomness s_0 for $(\sigma_{o,0}, \sigma_{o,0}')$ and independent random values $\{r_i'\}$ for each leaf component $(\sigma_{i_0}, \sigma_{i_1})$. These randomizers computationally hide all key-related components contained in (d_{i_0}, d_{i_1}) . Consequently, for every $i \in \omega^*$, the exposed leaf components $(\sigma_{i_0}, \sigma_{i_1})$ are distributed independently of whether the signer actually possesses attribute i , and are computationally indistinguishable from a fresh randomization of $(H_1(i)^{r_i'}, g^{r_i'})$.

Moreover, since both ω_0 and ω_1 satisfy the access structure Γ , the reconstruction performed in *VerNode* yields the same root value for both attribute sets. Thus the structure needed for verification leaks no distinguishing information.

Consequently, the complete delegation $\sigma_o = (\sigma_{o,0}, \{\sigma_{i0}, \sigma_{i1}\}_{i \in \omega^*}, \sigma'_{o,0})$ is computationally indistinguishable under ω_0 and ω_1 . The advantage $\text{Adv}_A^{\text{APOS}}$ is negligible. Therefore, the scheme achieves attribute privacy for original signers. $\square$

Using an argument analogous to that of Theorem 3, we obtain the following result.

Theorem 4. *The EABPS scheme achieves attribute privacy for proxy signers.*

In summary, we can conclude that the EABPS scheme protects attribute privacy for signers.

6.2.3. Collusion Resistance

It is important to note that our definition of unforgeability implies collusion resistance. In particular, no group of users or signers can combine their private keys or delegation information to generate a signature that none of them could produce individually. If a coalition of users were able to construct such an unauthorized signature, then one could build a new adversary that simulates the entire colluding group and outputs this signature as a valid forgery in the unforgeability game. This contradicts the security definition and therefore cannot occur.

Hence, any attempt by multiple users to pool their private keys or delegation does not provide them with additional signing capabilities. This establishes that the proposed EABPS scheme is inherently secure against collusion attacks.

6.3. Efficiency Analysis

In this subsection, we analyze feasibility of cryptographic constructions for UAV and assess the efficiency of our proposed EABPS scheme in comparison to related schemes: the ABPS-HL1 [29], ABPS-HL2 [30], and t-ABPS [31] schemes. First, we evaluate their computational costs by analyzing the execution times of the *Sign* and *Verify* algorithms. Second, we examine their communication overheads through a comparison of signature lengths.

6.3.1. Feasibility of Cryptographic Constructions for UAV

We use hash operations, pairing operations, and the Java pairing-based cryptography (jPBC) library [33] to construct and implement our EABPS scheme. They are necessary for achieving the required security properties of ABPS schemes and suitable for the computational capabilities of UAV platforms.

(1) Hash operations

Hash functions are among the most lightweight cryptographic primitives supported by embedded processors. UAV platforms can compute them with low latency and energy consumption. In our scheme, hash functions play a fundamental role. These operations fully comply with the computational constraints of UAV nodes.

(2) Pairing operations

Pairing-based cryptography is necessary to enable expressive attribute-based delegation and proxy signing. In our scheme, UAVs are only required to execute the *Verify* algorithm that involves only a number of pairing operations. The feasibility of the pairing operation for UAV hardware is confirmed by our experimental data.

(3) jPBC

The jPBC library is adopted because it provides efficient and widely used implementations of bilinear pairings. By using jPBC on our platform, we obtain realistic estimates of pairing computational costs. Since UAVs only perform a number of pairing operations in the *Verify* algorithm, these experimental results demonstrate that the computational costs remain well within the capability range of typical UAV processors.

Therefore, hash operations, pairing operations, and the jPBC library employed in our EABPS scheme are not only essential from a security and functional perspective, but also fully compatible with the computational capabilities of UAVs.

6.3.2. Computational Costs

The compared signature schemes employ fundamental cryptographic operations, including exponentiation, hashing, and bilinear pairing. We conduct a theoretical performance analysis to determine the number of these operations required for the *Sign* and *Verify* algorithms. The analytical results are summarized in Tables 3 and 4, where n denotes the number of attributes and k indicates the threshold parameter.

For our EABPS scheme, the expressions in Tables 3 and 4 are obtained directly from the step-based descriptions of the algorithms. Since each step explicitly specifies the cryptographic operations it invokes, we counted the exact occurrences of exponential operations, hash operations, and pairing operations. For example, in the *Sign* algorithm, the signer performs one hash operation and one exponential operation to compute $\sigma_{p,0}$, and one exponential operation to compute $\sigma'_{p,0}$. It also performs $2n$ exponential operations and n hash operations to compute the components $(\sigma_{ip,0}, \sigma_{ip,1})$. Therefore, the signer performs a total of $2n + 2$ exponential operations and $n + 1$ hash operations in the *Sign* algorithm. Using the same method, we derive the numbers of exponential operations, hash operations, and pairing operations in the *Verify* algorithm. For the schemes in [29–31], the expressions are derived in the same manner from the algorithmic descriptions provided in those constructions. These results form the expressions listed in Tables 3 and 4.

Table 3. Amount of some operations in *Sign*.

Signature Schemes	Exponential Operation	Hash Operation	Pairing Operation
EABPS	$2n+2$	$n+1$	0
t-ABPS	$3n$	1	0
ABPS-HL1	$6d$	$d+1$	0
ABPS-HL2	$2n+4d - 2k+2$	$n+d - k+1$	0

Table 4. Amount of some operations in *Verify*.

Signature Schemes	Exponential Operation	Hash Operation	Pairing Operation
EABPS	$2n$	2	$4n+3$
t-ABPS	$2d$	$2d+2$	$5d$
ABPS-HL1	$2d$	$2d+2$	$4d+1$
ABPS-HL2	0	$2(n+d - k+1)$	$2(n+d - k)+3$

The computational time of the fundamental operations was measured using the jPBC library. All experiments were conducted on a personal computer equipped with an Intel i5-13400 CPU and 16 GB of RAM. The implementation adopts a Type A pairing, where the parameter $rBits$ is set to 160, representing the bit length of the group order, and $qBits$ is set to 512, indicating the bit length of the underlying finite field. The experimental results demonstrate that the exponentiation, hash, and pairing operations require approximately 10.3 ms, 23.08 ms, and 10.59 ms, respectively. Based on both analytical evaluation and experimental measurements, we estimate the computational time of the *Sign* and *Verify* algorithms in these ABPS schemes. For ease of analysis and visualization, we assume $n = d = 2k$. The corresponding results are presented in Figures 3 and 4.

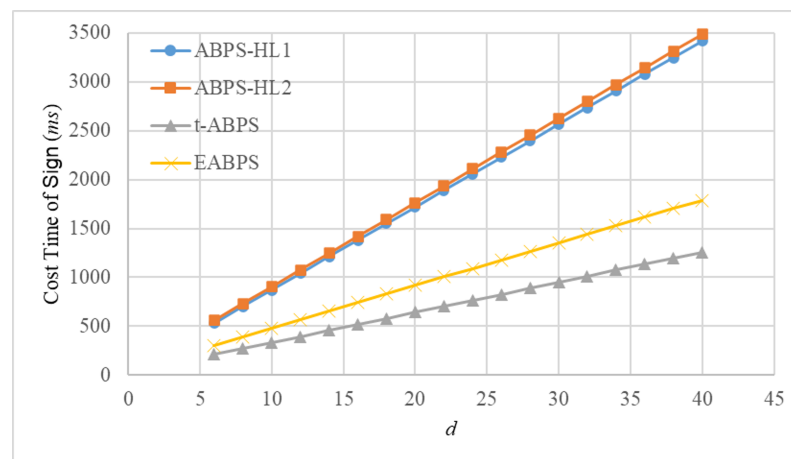


Figure 3. Computational time(ms) of algorithm *Sign* in different ABPS schemes.

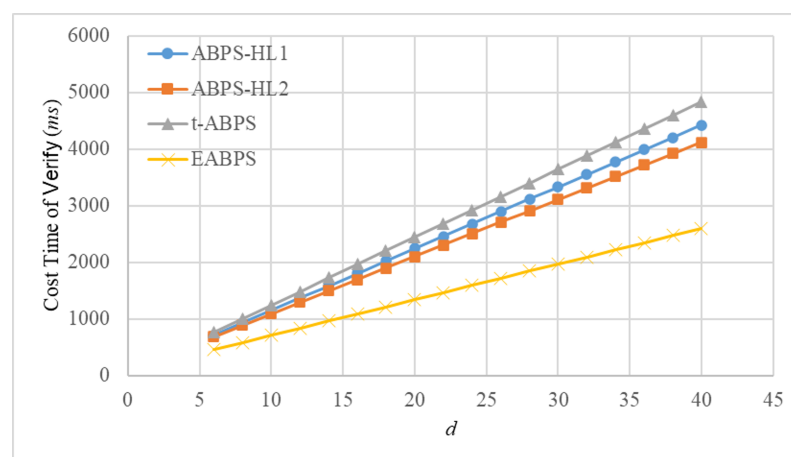


Figure 4. Computational time(ms) of algorithm *Verify* in different ABPS schemes.

We provide the dependency formulas used to compute the execution time shown in Figures 3 and 4. For each selected value of d , based on the assumption $n = d = 2k$, we obtain the corresponding number of exponential operations $N_{exp}(d)$, hash operations $N_{hash}(d)$, and pairing operations $N_{pair}(d)$ required by the *Sign* algorithm. Therefore, the total running time for *Sign* at parameter d is computed as $T_{Sign}(d) = 10.3N_{exp}(d) + 23.08N_{hash}(d) + 10.59N_{pair}(d)$. The same dependency expression is used for *Verify*.

Because each operation count in Table 3 grows linearly with d , the resulting running time $T_{Sign}(d)$ and $T_{Verify}(d)$ also grow linearly. Thus, the linearity observed in Figures 3 and 4 reflects the property of the underlying construction.

Figure 3 illustrates the computational time of the *Sign* algorithm for four ABPS schemes, namely ABPS-HL1, ABPS-HL2, t-ABPS, and the proposed EABPS, as the number of attributes d increases. All schemes exhibit a roughly linear growth in signing cost, which is consistent with the increasing number of operations required for larger attribute sets. However, significant differences appear in their overall efficiency. ABPS-HL2 incurs the highest computational overhead, while ABPS-HL1 performs moderately better but still suffers from substantial computational burden. In comparison, EABPS consistently achieves a lower signing cost than both ABPS-HL1 and ABPS-HL2 for all examined values of d . This performance improvement stems from the optimized signing procedure in the EABPS, which reduces redundant exponential operations. Although t-ABPS demonstrates the lowest cost among all schemes, its performance advantage comes at the expense of limited expressiveness and reduced support for flexible attribute policies. Overall, the results confirm that EABPS provides a favorable balance between expressiveness and effi-

ciency, making it a more practical solution for real-world scenarios that require secure and fine-grained proxy signing capabilities.

Figure 4 shows the verification time of four ABPS schemes as the number of attributes d increases. All schemes exhibit an approximately linear growth pattern, consistent with the increasing number of operations required during verification. However, the efficiency differences among the schemes are substantial. The ABPS-HL1, ABPS-HL2, and t-ABPS schemes incur relatively high verification overhead, with t-ABPS being slightly more expensive.

In contrast, the proposed EABPS scheme consistently achieves the lowest verification cost across all tested values of d . This efficiency gain results from the optimized verification design in EABPS, which reduces redundant hash operations. As a result, the EABPS scheme provides a more practical balance between verification efficiency and policy expressiveness. These results confirm that EABPS significantly enhances verification performance while maintaining flexible and secure attribute-based proxy signing capabilities. Moreover, these results make EABPS particularly suitable for deployment in real-world scenarios where a large number of verifications must be performed efficiently, such as UAV networks and distributed authentication systems.

Figure 5 presents the combined computational time of the *Sign* and *Verify* algorithms for four ABPS schemes as the number of attributes d increases. All schemes exhibit a roughly linear increase in total computation cost. Among the compared schemes, ABPS-HL1 and ABPS-HL2 incur the higher overall cost, with their execution time rising rapidly as d grows. Although t-ABPS performs more efficiently than ABPS-HL1 and ABPS-HL2 schemes, its comparatively lower total cost stems from its restricted expressiveness and inability to support rich attribute policies.

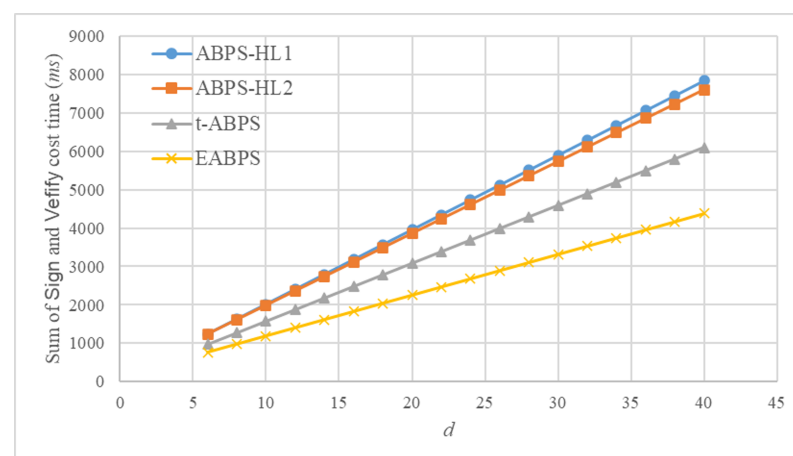


Figure 5. Total computational time(ms) of the algorithms *Sign* and *Verify* in different ABPS schemes.

In contrast, the proposed EABPS scheme consistently achieves the lowest total computational time across all tested values of d . The improvement mainly results from the optimized design of its verification procedure. As a consequence, the EABPS scheme delivers a substantially more efficient end-to-end signing and verification process while preserving full attribute expressiveness and strong security guarantees. These findings confirm that the EABPS scheme offers a superior balance between computational efficiency and functional capability, making it well suited for practical deployment in large-scale, attribute-driven application environments, such as UAV networks and distributed authentication systems.

In addition, it is essential to discuss the applicability of our experimental results to UAV hardware platforms. Although our implementation and measurements are conducted

on a desktop computer, recent studies show that many modern UAVs are equipped with embedded processors. The computational complexity of our scheme depends primarily on exponential operations, hash operations, and pairing operations. The relative efficiency and scalability trends observed in our experiments remain valid for UAV platforms, even though the absolute running time would scale according to the processor performance. Hence, the proposed EABPS scheme is suitable for deployment on typical UAV hardware.

6.3.3. Communication Costs

Since a signer must transmit a digital signature to the verifier, the signature size is regarded as the communication cost. A theoretical analysis of the signature length for these schemes is presented in Table 5.

Table 5. Signature length of different ABPS schemes.

Signature Schemes	Signature Length
EABPS	$(4n + 3) G $
t-ABPS	$5n G $
ABPS-HL1	$(4d + 1) G $
ABPS-HL2	$(2(n + d - k) + 3) G $

We assume $n = d = 2k$ and $|G| = 512$ bits. The results are illustrated in Figure 6.

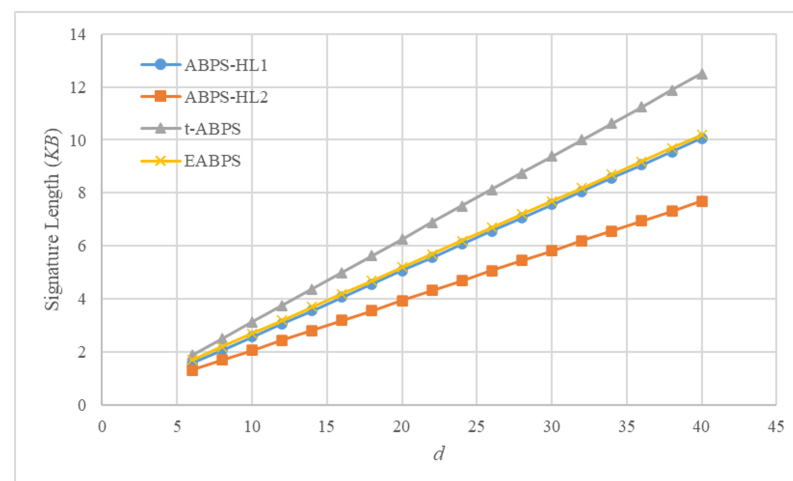


Figure 6. Comparison of signature length of the schemes(KB).

Figure 6 compares the signature lengths of four ABPS schemes as the number of attributes d increases. All schemes exhibit a linear growth pattern, reflecting the fact that signatures contain attribute-related components whose number scales with the size of the underlying attribute set. However, notable differences exist in the communication costs. The t-ABPS scheme yields the largest signature size across all tested values of d . ABPS-HL1 and ABPS-HL2 generate smaller signatures.

In contrast, the proposed EABPS scheme yields signature lengths larger than those of ABPS-HL1 and ABPS-HL2 but smaller than t-ABPS. This intermediate position results from design trade-offs in the EABPS scheme. These trade-offs arise from the structural differences between our expressive tree-based access policies and the simpler threshold-based designs used in related ABPS schemes. Expressive policies inherently require additional structural overhead, which modestly increases the overall signature size. Nevertheless, the proposed EABPS scheme is suitable for UAV environments, where fine-grained authorization and authentication are essential, and where slightly larger signature sizes are acceptable in exchange for enhanced functionality.

7. Conclusions

We presented an EABPS scheme designed to address the security and privacy challenges inherent in UAV networks. Our scheme enables fine-grained authorization and authentication through flexible attribute policies, ensuring that only entities whose attributes satisfy a predefined access structure are permitted to generate valid proxy signatures. The EABPS scheme further preserves the attribute privacy for both original and proxy signers, effectively decoupling signatures from explicit identities and mitigating privacy risks.

We conducted security analysis to demonstrate that the proposed scheme achieves unforgeability and attribute privacy. Experimental evaluation confirmed that the EABPS scheme offers improved efficiency compared with existing ABPS constructions while maintaining strong expressiveness and lower communication overhead. These results indicate that the EABPS scheme provides a practical cryptographic construction for secure delegation, message authentication, and privacy-preserving communication in UAV networks.

Future work will explore optimizing the scheme for large-scale, heterogeneous UAV environments, integrating lightweight cryptographic techniques for resource-constrained UAVs, and extending the construction to support revocation and dynamic attribute updates.

Author Contributions: Conceptualization, L.H., Y.G. and S.J.; software, L.H.; validation, Y.G. and S.J.; formal analysis, L.H.; writing—original draft preparation, L.H.; writing—review and editing, L.H., Y.G. and S.J.; supervision, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Henan grant number 242300420296 and the Doctor Research Fund of Zhengzhou University of Light Industry grant number 2022BSJJZK15.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152.
2. Zhi, Y.; Fu, Z.; Sun, X.; Yu, J. Security and privacy issues of UAV: A survey. *Mobile Netw. Appl.* **2020**, *25*, 95–101.
3. Samanth, S.; K V, P.; Balachandra, M. Security in Internet of drones: a comprehensive review. *Cogent Eng.* **2022**, *9*, 2029080.
4. Khan, M.A.; Ullah, I.; Nisar, S.; Noor, F.; Qureshi, I. M.; Khanzada, F.; Khattak, H.; Aziz, M. A. Multiaccess edge computing empowered flying ad hoc networks with secure deployment using identity-based generalized signcryption. *Mobile Inf. Syst.* **2020**, *2020*, 8861947.
5. Din, N.; Waheed, A.; Zareei, M.; Alanazi, F. An improved identity-based generalized signcryption scheme for secure multi-access edge computing empowered flying ad hoc networks. *IEEE Access* **2021**, *9*, 120704–120714.
6. Wang, H.; Li, J.; Lai, C.; Wang, Z. A provably secure aggregate authentication scheme for unmanned aerial vehicle cluster networks. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 53–63.
7. Lei, Y.; Zeng, L.; Li, Y.; Wang, M.; Qin, H. A lightweight authentication protocol for UAV networks based on security and computational resource optimization. *IEEE Access* **2021**, *9*, 53769–53785.
8. He, L.; Ma, J.; Shen, L.; Wei, D. Certificateless designated verifier proxy signature scheme for unmanned aerial vehicle networks. *Sci. China Inf. Sci.* **2021**, *64*, 112101.
9. Xu, Z.; Luo, M.; Vijayakumar, P.; Peng, C.; Wang, L. Efficient certificateless designated verifier proxy signature scheme using UAV network for sustainable smart city. *Sustain. Cities Soc.* **2022**, *80*, 103771.
10. Qu, Y.; Zeng, J. Certificateless proxy signcryption in the standard model for a UAV network. *IEEE Internet Things J.* **2022**, *9*, 15116–15127.
11. Sysoyev, A.; Nauruzov, K.; Karati, A.; Abramkina, O.; Vitulyova, Y.; Yeskendirova, D.; Popova, Y.; Abdoldina, F. Lightweight group signature scheme based on PUF for UAV communication security. *Drones* **2025**, *9*, 693.
12. Li, T.; Zhang, J.; Obaidat, M. S.; Lin, C.; Lin, Y.; Shen, Y.; Ma, J. Energy-efficient and secure communication toward UAV networks. *IEEE Internet Things J.* **2022**, *9*, 10061–10076.

13. Liu, Z.; Guo, J.; Huang, F.; Cai, D.; Wu, Y.; Chen, X.; Igonovich, K. Lightweight trustworthy message exchange in unmanned aerial vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 2144–2157.
14. Aissaoui, R.; Deneuville, J.C.; Guerber, C.; Pirovano, A. Authenticating civil UAV communications with post-quantum digital signatures. In Proceedings of the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), Barcelona, Spain, 1–5 October 2023.
15. Li, Z.; Chen, Q.; Li, J.; Huang, J.; Mo, W.; Wong, D. S.; Jiang, H. A secure and efficient UAV network defense strategy: Convergence of blockchain and deep learning. *Comput. Stand. Interfaces* **2024**, *90*, 103844.
16. Khan, M. A.; Ullah, I.; Kumar, N.; Akhonzada, A.; Anisi, M. H.; Alqhatani, A.; Afghah, F.; Barb, G.; Waqas, A. An improvised certificate-based proxy signature using hyperelliptic curve cryptography for secure UAV communications. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 5264–5275.
17. Maji, H. K.; Prabhakaran, M.; Rosulek, M. Attribute-based signatures. In Proceedings of the Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, 14–18 February 2011.
18. Su, J.; Cao, D.; Zhao, B.; Wang, X.; You, I. ePASS: an expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things. *Future Gener. Comput. Syst.* **2014**, *33*, 11–18.
19. Liu, L.; Hsu, C.; Au, M. H.; Harn, L.; Cui, J.; Zhao, Z. A revocable and comparable attribute-based signature scheme from lattices for IoMT. *J. Syst. Archit.* **2024**, *154*, 103222.
20. Wu, Y.; Xiong, H.; Khan, F.; Ijaz, S.; Alturki, R.; Aljohani, A. Efficient and provably secured puncturable attribute-based signature for Web 3.0. *Future Gener. Comput. Syst.* **2025**, *164*, 107568.
21. Li, J.; Au, M. H.; Susilo, W.; Xie, D.; Ren, K. Attribute-based signature and its applications. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, 13–16 April 2010.
22. Su, Q.; Zhang, R.; Xue, R.; Sun, Y.; Gao, S. Distributed attribute-based signature with attribute dynamic update for smart grid. *IEEE Trans. Ind. Inform.*, **2023**, *19*, 9424–9435.
23. Chen, X.; Li, J.; Huang, X.; Li, J.; Xiang, Y.; Wong, D. S. Secure outsourced attribute-based signatures. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 3285–3294.
24. Tao, Q.; Cui, X.; Iftekhhar, A. A novel lightweight decentralized attribute-based signature scheme for social co-governance. *Inf. Sci.* **2024**, *654*, 119839.
25. Xiong, H.; Bao, Y.; Nie, X.; Asoor, Y. I. Server-aided attribute-based signature supporting expressive access structures for industrial Internet of things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 1013–1023.
26. Huang, Z.; Lin, Z. Secure server-aided attribute-based signature with perfect anonymity for cloud-assisted systems. *J. Inf. Secur. Appl.* **2022**, *65*, 103066.
27. Li, J.; Chen, Y.; Han, J.; Liu, C.; Zhang, Y.; Wang, H. Decentralized attribute-based server-aid signature in the Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 4573–4583.
28. Liu, X.; Ma, J.; Xiong, J.; Zhang, T.; Li, Q. Personal health records integrity verification using attribute based proxy signature in cloud computing. In Proceedings of the 6th International Conference on Internet and Distributed Computing Systems, Hangzhou, China, 28–30 October 2013.
29. He, L.; Ma, J.; Wei, D. Attribute-based proxy signature scheme for unmanned aerial vehicle networks. *J. Commun.* **2021**, *42*, 87–96.
30. He, L.; Gan, Y.; Zhang, Y. Attribute-based proxy signature scheme supporting flexible threshold predicate for UAV networks. *Electronics* **2023**, *12*, 4854.
31. He, L.; Gan, Y.; Yin, Y. Efficient threshold attribute-based signature scheme for unmanned aerial vehicle (UAV) networks. *Electronics* **2025**, *14*, 339.
32. Huang, X.; Susilo, W.; Mu, Y.; Wu, W. Proxy signature without random oracles. In Proceedings of the Second International Conference on Mobile Ad-hoc and Sensor Networks, Hong Kong, China, 13–15 December 2006.
33. Caro, A. D.; Iovino, V. jPBC: Java pairing based cryptography. In Proceedings of the 2011 IEEE Symposium on Computers and Communications, Kerkyra, Greece, 28 June–1 July 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.