Computer Networks 185 (2021) 107599

Contents lists available at ScienceDirect

Computer Networks

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

A new provable hierarchical anonymous certificateless authentication protocol with aggregate verification in ADS-B systems

Amirhossein Asari [a], Mahdi R. Alagheband [b,*], Majid Bayat [c], Maryam Rajabzadeh Asaar [a]

[a] Department of Electrical and Computer Engineering, Science and Research Branch, IAU, Tehran, Iran
[b] Cybersecurity Research Lab (CRL), Ryerson University, Toronto, Canada
[c] Department of Electrical and Computer Engineering, Shahed university, Tehran, Iran

ARTICLE INFO

Keywords:
Digital signature
Certificateless aggregate signature
Random oracle model
Privacy-preserving
Surveillance-broadcast systems

ABSTRACT

Automatic Dependent Surveillance-Broadcast (ADS-B) technology is a new solution for communication among aircraft and ground controller stations. In this new surveillance technology, critical messages (e.g., the location of aircraft) are achieved by navigation satellites, and then an on-board equipment multicasts the unencrypted messages twice per second to the others. The former ADS-B protocols suffer from weak authentication protocols with a few security vulnerabilities and privacy issues including key-escrow problem, user profiling, time consuming verification processes, and difficulties with certificate management. In this paper, we propose a new hierarchical authentication protocol used Certificateless Public Key Cryptography (CL-PKC) technique to avoid using Public Key Infrastructure (PKI) certificate management and solve the key-escrow concern in Identity-based Public Key Cryptography (ID-PKC). Further, unlike many of certificateless schemes, our proposed scheme is secure against malicious-but-passive Key Generation Center (KGC). We prove that our scheme preserves conditional privacy, which means that distinct identities map to varied pseudonyms. The security and privacy features of our scheme are provably modeled under the widely-accepted random oracle model by computational Diffie-Hellman (CDH) assumption against adaptive chosen-message attack. Finally, we show that the time needed to aggregate verification of 50 messages reduced by 84% and 48% compared to those of Yang et al.'s and He et al.'s schemes respectively.

1. Introduction

According to the technological advancement of air traveling and the high level of safety, air traveling has become more popular around the world. On the other hand, non-traveling applications of aircraft like using Unmanned Aerial System (UAS) for different purposes are expanding rapidly. Thus, the safety of air traffic becomes more momentous.

In the first generation of Air Traffic Control (ATC), the ground controllers are equipped with Primary Surveillance Radar (PSR) based on bouncing electromagnetic waves. The fixed ground antennas send the waves to air targets and then interprets the reflected signals and estimates information like location and velocity of the targets. The major weakness of this system is poor accuracy especially in vertical angles [1]. By developing the military Identification Friend or Foe (IFF) technology, Secondary Surveillance Radar (SSR) was born. Unlike the PSR, SSR relied on targets equipped by a radar transponder, that replied to each interrogation signal by transmitting a response containing encoded data.

The Automatic Dependent Surveillance-Broadcast (ADS-B) is a new SSR-based surveillance technology that provides air to air and air to ground communication links for aircraft. The word "automatic" emphasizes that it does not need a pilot or an operator to exchange messages. The word "dependent" refers to dependency of ADS-B to navigation satellites. In this new technology, aircrafts equipped by ADS-B use an ordinary Global Navigation Satellite System (GNSS) to achieve precise information like position, velocity, and flight direction. Furthermore all ADS-B aircrafts periodically send their information to each other and ground stations every 0.5 s. The architecture of ADS-B has shown in Fig. 1. The ADS-B is supported by two physical layer protocols [2]:

- UAT. Universal Access Transceiver (UAT) standard applies a communication channel on 978 MHZ frequency with a bandwidth of 1Mbps. It has a packet size of 144 or 272-bit. Special onboard devices needed to provide the above channel. The UAT is not support in class A airspace (above 18000 ft mean sea level).

* Corresponding author.
E-mail address: m.alaghband@ryerson.ca (M.R. Alagheband).

https://doi.org/10.1016/j.comnet.2020.107599
Received 16 June 2020; Received in revised form 5 September 2020; Accepted 9 October 2020
Available online 16 October 2020

- 1090ES. The 1090 MHZ Extended Squitter mode S (1090ES) is a standard based on the old 1090 MHZ aviation communication channel. 1090ES supports packets size of 56 or 112-bit.

The Security and privacy issues of ADS-B are of utmost importance. To achieve security in different aspects of the ADS-B, developers should apply both non-cryptographic and cryptographic solutions. For instance, multilateration methods [3,4] and distance-bounding mechanism [5] as non-cryptographic approaches cover location verification that validate the claimed location of ADS-B nodes.

In this paper, we focus on cryptographic tools to reach a secure and fast authentication protocol for ADS-B communications. Furthermore, we put forward methods that preserve privacy.

There are many vulnerabilities in different ADS-B protocols on physical, data link, and network layers. These vulnerabilities can be categorized as follow [6–8]:

- **Eavesdropping** or passive attack is the act of stealthily listening to a communication link. This attack easily happens when ADS-B protocols have no mechanism for confidentiality. It is almost impossible to prevent eavesdropping without encryption and it is difficult to detect.
- **Jamming**. Easily by sending a powerful electromagnetic wave on 1090 MHz of mode S, nodes are disabled to send or receive ADS-B messages [9].
- **Data injection**. Without authentication measures, everyone can send non-legitimate data on the ADS-B communication link. Besides attackers can repudiate this malign activity.
- **Data modification**. With the lack of message integrity, an attacker can modify the bits of data. It is more critical than vulnerability against message injection because of the retouched message was a valid message.
- **Data deletion**. Due to the superposition principle, if an attacker produces the invert of the targeted signal, she can highly attenuated it, leads to destructive interference. Attacks like this are more dangerous than jamming because the fingerprint of the attacker is so pale.

1.1. Our contribution

In this paper, we introduce a novel authentication protocol for the below scenario.

First, users should not be the victim of user profiling attack which causes privacy leakage. Also, when nodes crash or misbehave, a legal authority should link between pseudonyms and their real identities, therefore the scheme must preserve privacy in a conditional way. We use short-live pseudonyms to achieve conditional privacy-preserving and anonymity.

Due to the content of exchanged messages that are public information, like height and velocity, the topic of confidentiality does not matter anymore. In addition, we cover message integrity and origin authentication by using public key cryptography solutions in the authentication mechanism for mobile ad-hoc network(MANET) structure. Our scheme that called HACA uses certificateless-public key cryptography (CL-PKC) authentication mechanism to solve the weaknesses of the aforementioned approaches. The scheme also obtains the advantage of aggregate signature. By using the aggregate signature, the computational cost of time-consuming verification process impressively reduced. At last, we use the random oracle model to prove the security of the proposed scheme against adaptive chosen message attack and malicious-but-passive Key Generation Center (KGC).

All in all, the specifications of our scheme are as follow:

- The scheme uses short-lived pseudonyms to achieve anonymity. None of the nodes, airlines or KGC can reveal the real identity of the anonymous node and also they cannot link two pseudonyms that related to a real identity. If there are a criminal case or a user credit finishes then, the Trace Authority (TRA), which is liable for the production of the pseudonyms can track the pseudonyms and reveal the real identity of it.
- By using CL-PKC principles in the designing phase, firstly, we eliminate traditional certificate management overhead in PKI cryptosystems. Secondly, the key escrow problem that is a structural privacy problem of ID-PKC has been resolved. Therefore we decline the level of 'trust' needed in the network.
- Our Scheme is secure against the advance malicious-but-passive KGC attack.
- Due to the use of aggregate verification, the scheme is faster than traditional verification methods. We show that the time needed to aggregate verification of 50 messages reduced by 84% and 48% compared to those of Yang et al. and He et al. schemes respectively.
- The hierarchical topology of our scheme better matches with network realities and causes better load balance in the network.

1.2. Related work

In this section, we look through the applicable proposed solutions to achieve the security and privacy goals of ADS-B systems.

On the one hand, in traditional PKI-based cryptography, each user has at least two kinds of keys. The first one is a private key that is remained secret by the user and the second one is a public key, usually derived from private key in a simple way and distributed through the network. For concreting public keys and users identities a certificate authority (CA) issues digital certificates. All users need to confirm the digital certificate before using of related public key. As a result, the network needs a huge amount of storage and computation power to handle certificate management problem in PKI.

For the first time in 1984, Shamir introduced the concept of IDentity based-Public Key Cryptography (ID-PKC) to solve digital certificate management [10], and after 16 years Boneh suggested the first practical ID-PKC based on the pairing transform [11]. In ID-PKC, the user's public key is their identity information such as email addresses or phone numbers. Private Key Generator (PKG) is the responsible authority for generation of private keys for users. The PKG obtains private keys from users' public keys. Since ID-PKC simplifies PKI certificate management but it brings a new major problem called key-escrow. The PKG owns all user's secrets so they must fully trust the PKG. Key-escrow means that the PKG is able to decrypt all encrypted messages and also forges every desired signature in the network. This thorough trustworthiness is not a real assumption in many scenarios including ADS-B.

On the other hand, digital signatures provide three main services in the network: authentication, integrity and, non-repudiations [12]. Digital signatures are part of X.509 standard [13] which is a well-known and international standard that prevents modification and forgery of signed messages. We require a specific kind of digital signatures. In 2003, Boneh presented the notion of aggregate signature [14]. In this way, the n number of distinct signature-message pairwise signed by n different identities is converted to one signature. The length of signatures and the computational overhead are reduced.

In 2003, Alriami and Paterson [15] introduced the concept of certificateless signature (CLS). A third party KGC is responsible for generation of partial-private key instead of full-private key. The partial-private keys are distributed among users via a secure channel. Then, all users can derive their full-private key from their secrets and their partial private key. Thus, CLS eliminates key escrow problem in ID-PKC via decreasing the level of *trust* needed to third party. Yum et al. [16] proposed a CLS scheme and claimed it is secure against adaptive chosen message attack. Hu et al. [17] pointed out that the Yum scheme is vulnerable against public key replacement attack. Yeh et al. [18] proposed a CLS scheme without bilinear pairing for IoT-based smart objects. Yeh claimed that his proposed scheme is secure against two adversary models defined in [15]. Jia et al. [19] explained two vulnerabilities

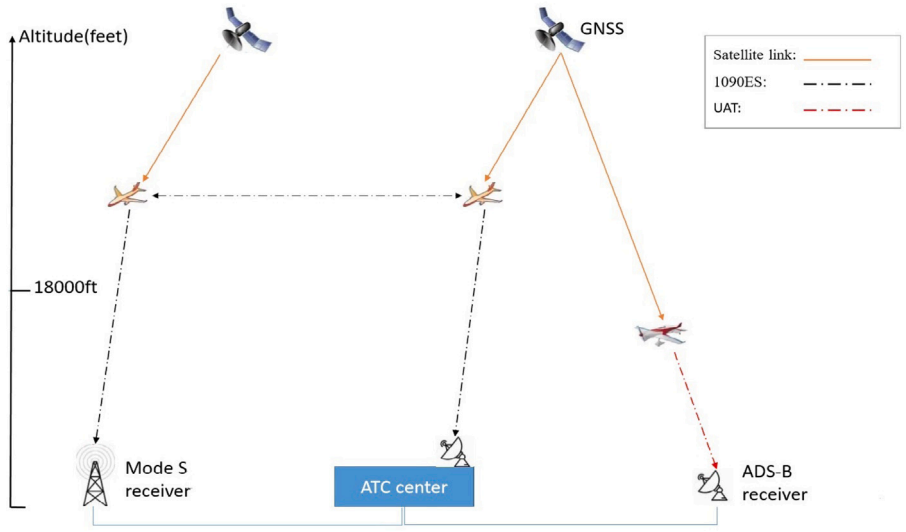

Fig. 1. The ADS-B communication links.

in [18]. The first one is about simulating the KGC to forge partial-private keys of users and the second one is the vulnerability against key replacement attack. Jia et al. also proposed a CLS scheme in [19] and asserted their scheme is unforgeable versus two super adversary models. Du et al. [20] show that scheme in [19] could forge by normal adversary of first type and propose a CLS scheme without pairing that resists against two super adversary models.

Zhang combined the CLS and aggregated signatures, designed certificateless aggregate signature (CLAS) scheme, and proved its security under the random oracle model [21]. Shim showed that the Zhang scheme could not resist an insider attacker [22]. Moreover, many of the proposed CLAS schemes like [21] have a non-fixed number of the pairing operation. The computation overhead dramatically grows by the moderately increasing number of signers. In some other CLAS schemes like [23] all involved users share the same synchronized clock to generate aggregate signature. Xiong et al. [24] proposed a new CLAS scheme that does not require clock synchronization and benefits from constant number of pairings. The authors of [25] and [26] proved that the Xiong's scheme is forgeable. Then, Horng et al. proposed a CLAS scheme for Vehicular Ad-hoc Network (VANET) that does not require time synchronization [27]. Their scheme has a constant number of pairing transform which is independent of the number of generated signatures. In addition [27] Supports privacy-preserving due to the use of pseudonyms. Li et al. showed that the Horng scheme is insecure against malicious-but-passive KGC attack [28]. In 2018, Kumar et al. introduced another CLAS authentication mechanism for health care networks that is secure against malicious-but-passive KGC attack [29]. However, the scheme has no mechanism for privacy-preserving.

Privacy-preserving is the other significant issue. Lin et al. [30] proposed group signature to achieve privacy-preserving. In this way, each member of group uses own private key to generate a signature on a message, but on the receiver side, all receiving messages are validated with group public key. As a consequence, the receiver only knows the user membership status in the specific group and the identity of each user are hidden from receiver. An inefficient revocation mechanism is the main drawback of [30]. Another way to achieve privacy-preserving has been proposed by Maxim et al. [31]. A CA is responsible for producing anonymous certificates that each of them is related to one pseudonym. A sender picks up a certificate from the pool of certificates, then signs the data with related private key and sends the data and chosen certificate to the receiver. The receiver can validate the signature with an attached anonymous certificate. Raya et al. [32] improved the idea of [31] by using hardware secure module (HSM). Lu et al. [33] used short lived pseudonyms to achieve anonymity. The main disadvantage of [33] is that using an updating Certificate Revocation List (CRL) is expensive. Ying et al. [34] proposed an authentication protocol that used pseudonyms to conceal the real identity of the vehicle owner in VANET. Du to use of symmetric verification process in [34], the scheme is efficient in term of computational overhead, but there are vulnerabilities against Sybil and Replay attacks [35]. Pan et al. [36] used frequently changing pseudonyms based on the number of neighbors to preserve privacy in the VANET. In [37] each user has a joint pseudonym list with the HA server in order to achieve anonymity in global mobility networks (GLOMONET). Wu et al. [38] shows that a successful de-synchronization attack carried out against the [37]. Li et al. [39] proposed another pseudonym based privacy-preserving authentication protocol for GLOMONET. The structure of the [39] does not need time synchronization, therefore the scheme's complexity and computational costs have reduced.

For a more specialized review on the subject, we focus on papers that try to propose a secure authentication protocol for ADS-B. Yang et al. [40] proposed an ID-based authentication framework for ADS-B systems. His hierarchical scheme supports partial and full batch verification of messages. Then, He et al. [41] designed another ID-based authentication protocol. They did not use map-to-point hash functions to reduce computational overhead. The scheme supports (full) batch verification.

In 2019, Gowri et al. proposed another ID-based authentication protocol for ADS-B. [42] that slightly decreased the computational overhead compared with other discussed schemes due to not using the time-consuming bilinear pairing transform function. However, the Gowri's scheme [42] similar to [40,41] has a major drawback and suffers from the key-escrow vulnerability as an inherent privacy issue of ID-PKC architecture.

The rest of this paper is organized as follows. In Section 2 some preliminaries including bilinear pairing transform and some cryptographic hard problems have been explained. In Section 3 we clarify our system model and in Section 4 we prove its security and privacy features based on formal model. We compare our proposed scheme with some related schemes in terms of computational overhead and security aspects in Section 5. The conclusion of this paper placed in Section 6.

2. Preliminaries

In this section, we review some prerequisite basics of cryptography used in the following sections. The CL-PKC cryptosystems apply the bilinear transform.

Definition 1. Let G_1 is an additive group with order q and generator P and G_2 is a multiplicative group with same order $e : G_1 \times G_1 \rightarrow G_2$ is bilinear map and satisfies this properties [43]:

- *Bilinearity*: for points like $P, Q, R \in G_1$, $e(P, Q + R) = e(P, Q)\, e(P, R)$ and for any $a, b \in Z_q^*$ we have $e(aP, bP) = e(abP, P) = e(P, abP) = e(a, b)^{ab}$.
- *Non-degenerate*: $e(P, P) \neq 1$ where 1 is identity of G_2.
- *Computability*: it is efficient to calculate $e(P, Q)$ for any $P, Q \in G_1$.

Definition 2. A function $f(x)$ is negligible function if for a given $\epsilon \geq 0$ there exists an ϵ_0 such that: $f(x) \leq \frac{1}{x^{\epsilon}}$ for every $x \geq \epsilon_0$ [44].

Definition 3. CDH problem: for given ternary $(P, aP, bP) \in G_1$ and admissible pairing $e : G_1 \times G_1 \rightarrow G_2$ calculate $a.b.P$, for unknown $a, b \in Z_q^*$ and P is generator of G_1. The success probability of any probabilistic polynomial time (PPT) algorithm like $\mathcal{A}$ in solving the CDH problem in G_1 is defined to be [44]:

$$Adv_{\mathcal{A},G_1}^{CDH} = Pr[(P, aP, bP) = abP : a, b \in Z_q^*] \tag{1}$$

3. Proposed authentication protocol

In this section, we propose a new provable Hierarchical Anonymous Certificateless Authentication (HACA) protocol with aggregate verification as well as key escrow resistance.

3.1. Network model

There are two authorities in the top level of our topology:

- The KGC that is responsible for production partial private keys.
- The Trace Authority (TRA) that is liable for outputting and tracing pseudonyms.

The second level of the scheme consist of:

- The Airlines that Produces partial-private keys for its own aircrafts.

and the third level includes:

- The aircrafts that equipped with ADS-B transceiver and the ground stations are authorities which supposed to send and receive data under the proposed protocol.

3.2. System model

The HACA protocol is divided into CLS and CLAS structures. The CLS has seven algorithms including: *Setup, Pseudo Identity Generation, Partial Private Key Generation for airlines, Partial Private Key Generation for aircraft, Private Key Generation, signing, and Verification.* The CLAS structure has two extra algorithms*Aggregator* and *Aggregate Verification*. The CLS structure is not practical due to the lack of support for aggregate verification. We define it to facilitate the procedure of formal proof. We prove the security features for the CLS structure and then we relate the CLS security to the CLAS security.

Now, we express all nine algorithms of HACA protocol. A summarize of useful notation exists in Table 1.

1. *setup*
   - Having received security parameter $1^{\prime}$, KGC creates additive group G_1 with generator P with order q and multiplicative group G_2 with the same order.
   - KGC randomly chooses $msk \in Z_q^*$ as his master key and calculates public key as $mpk = msk.P$. Then, KGC selects two cryptographic hash function $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \rightarrow Z_q^*$.
   - KGC spreads public parameters $param = \{G_1, G_2, q, P, e, H_1, H_2, mpk\}$ and keeps msk secret.
   - TRA randomly chooses $\beta \in Z_q^*$ as his secret key to generate and track pseudonyms.
2. *Pseudo − Identity Generation*:
   - Every aircraft with real identity RID_{ij} randomly picks $k_i \in Z_q^*$, calculates $ID_{ij,1} = K_i.P$, and then sends $(RID_{ij}, ID_{ij,1})$ to TRA through a secure channel.
   - TRA calculates:

     $$ID_{ij,2} = RID_{ij} \oplus H_2(\beta.ID_{ij,1}, T_{ij}) \tag{2}$$

     The T_{ij} in Eq. (2) denotes the valid period of pseudonym. Then, $ID_{ij} = (ID_{ij,1}, ID_{ij,2}, T_{ij})$ and the pseudonym value $ID_{ij}^* = (ID_{ij}, AN_j)$ are calculated. AN_j represents the name of respective airline. TRA signs ID_{ij}^* and sends it to the aircraft.
   - In this way no one can track the pseudonym to find real identity. If there are a criminal case or a user credit finishes, TRA can easily find the real identity as it seen Eq. (3):

     $$RID_{ij} = ID_{ij,2} \oplus H_2(\beta.ID_{ij,1}, T_{ij}) \tag{3}$$
3. *Partial Private Key Generation for airlines* :
   - In this step KGC calculates $Q_{AN_j} = H_1(AN_j)$ and partial private key $Apsk_j = msk.Q_{AN_j}$. The partial private key will send to the airline. Every airline can check the validity of $Apsk_j$ by Eq. (4):

     $$e(Apsk_j, P) = e(Q_{AN_j}, mpk) \tag{4}$$

     If Eq. (4) holds, the received $Apsk_j$ is valid and receiver parses it as $Apsk_j = ((Apsk_j)_x, (Apsk_j)_y)$, then each airline randomly chooses $ran_j \in Z_q^*$ and calculates

     $$Ask_j = H_2((Apsk_j)_x, ran_j) \tag{5}$$

     and

     $$Apk_j = Ask_j.P \tag{6}$$

Table 1
The list of used parameters.

Notation	Description
i	Counter of aircraft
j	Counter of Airlines
N_1	Number of aircrafts
N_2	Number of airlines
msk	KGC master secret key
mpk	KGC public key
β	TRA secret key
AN_j	Name of jth airline
Ask_j	Secret key of jth airline
Apk_j	Public key of jth airline
$Apsk_j$	Partial private key of jth airline
RID_{ij}	Real identity of ith aircraft belong to jth airline
ID_{ij}^*	Pseudonym of ith aircraft belong to jth airline
T_{ij}	Validation period of ID_{ij}^*
x_{ij}	Private key of ith aircraft belong to jth airline
y_{ij}	Public key of ith aircraft belong to jth airline
psk_{ij}	Partial private key of ith aircraft belong to jth airline
SI	State information
σ_{ij}	Signature of ith aircraft belong to jth airline
Ω	Aggregate signature
S	Simulator
$\mathcal{A}_1$	Type1 adversary
$\mathcal{A}_2$	Type2 adversary
G_1	Additive group
G_2	Multiplicative group
P	Generator of G_1
q	Order of G_1

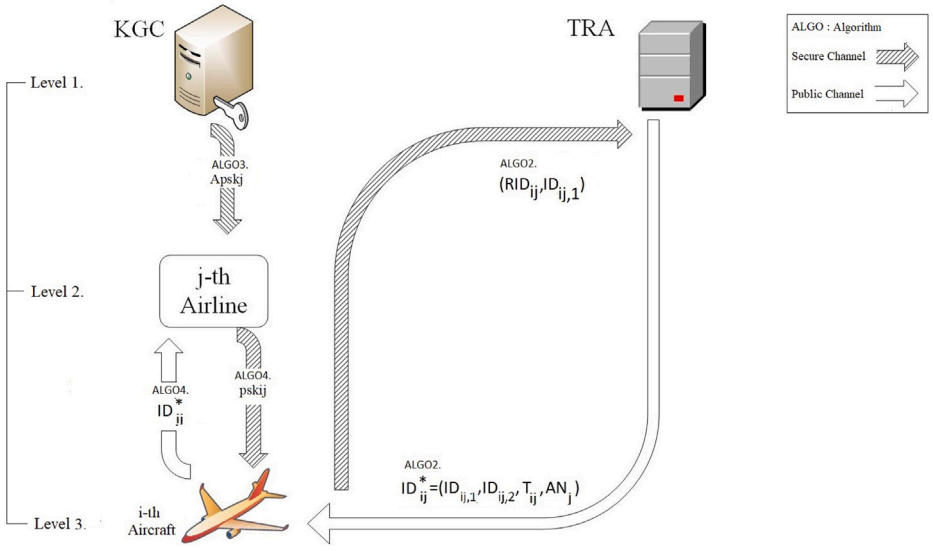

Fig. 2. The hierarchical structure of the HACA.

as his private–public key pair.

4. *Partial Private Key Generation for aircraft*:

   - In this step every aircraft who is the applicant for partial private key sends his signed pseudonym ID^*_{ij} to the respective airline. After validating pseudonym, the airline computes $Q_{ID^*_{ij}} = H_1(ID^*_{ij})$ and the partial private key $psk_{ij} = Ask_j.Q_{ID^*_{ij}}$. The partial private key will send to the aircraft with a public or secure channel. Every aircraft can check the validity of psk_{ij} as follow:

$$e(psk_{ij}, P) = e(Q_{ID^*_{ij}}, Apk_j) \tag{7}$$

   Fig. 2. illustrates an overview of the above algorithms. In algorithm 2 ($Pseudo - Identity\,Generation$) the real identity of the users must send via a secure way due to guarantee the user anonymity. According on the security model used in the following, it is necessary that both partial private keys are sent through the secure channel.

5. *Private Key Generation*
   According to the process of generation of the private key in CL-PKC, we derived the private key (x_{ij}) from two parameters, a secret value that randomly selected by the aircraft ($secret_{ij}$) and the partial private key of the aircraft (psk_{ij}) as follow: every aircraft parses received partial private key as $psk_{ij} = ((psk_{ij})_x, (Apsk_{ij})_y)$, then the aircraft randomly chooses $secret_{ij} \in Z^*_q$ and calculates $x_{ij} = H_2((psk_{ij})_x, secret_{ij})$ as the aircraft's private key. The public key simply derived as $y_{ij} = x_{ij}.P$.

6. *Signing*
   Every aircraft with pseudonym ID^*_{ij}, partial private key psk_{ij}, key pair (x_{ij}, y_{ij}) and state information SI can produce signature σ_{ij} on message m_{ij} after following steps:

   - Signer picks random $r_{ij} \in Z^*_q$ and computes:

$$R_{ij} = r_{ij}.P \tag{8}$$

   - At this step signer aircraft computes

$$h_{ij} = H_2(m_{ij}, ID_{ij}, y_{ij}, R_{ij}) \tag{9}$$

   and

$$W = H_1(SI) \tag{10}$$

   Parameters h_{ij} and W will use to compute V_{ij} at next step.

   - Signer computes V_{ij} as follow:

$$V_{ij} = psk_{ij} + r_{ij}.W + h_{ij}.x_{ij}.Apk_j. \tag{11}$$

   The digital signature of the aircraft on message m_{ij} will be:

$$\sigma_{ij} = (R_{ij}, V_{ij}) \tag{12}$$

7. *Verification*
   The verifier whether the aircraft or the airport who receives (σ_{ij}, m_{ij}) can verify the signature as follow:

   - Verifier computes:
     $W = H_1(SI)$,
     $h_{ij} = H_2(m_{ij}, ID_{ij}, y_{ij}, R_{ij})$,
     $Q_{ID^*_{ij}} = H_1(ID^*_{ij})$.
   - Verifier checks below equation:

$$e(V_{ij}, P) = e(Q_{ID^*_{ij}} + h_{ij}.y_{ij}, Apk_j)e(R_{ij}, W) \tag{13}$$

   If Eq. (13) holds, he accepts signature otherwise rejects. The correctness of Eq. (13) is shown by using Eq. (11):

$$e(V_{ij}, P) = e(psk_{ij} + r_{ij}.W + h_{ij}.x_{ij}.Apk_j, P) =$$
$$e(psk_{ij}, P)e(h_{ij}.x_{ij}.Apk_j, P)e(r_{ij}.W, P) =$$
$$e(Q_{ID^*_{ij}}, Apk_j)e(h_{ij}.y_{ij}, Apk_j)e(r_{ij}.P, W) =$$
$$e(Q_{ID^*_{ij}} + h_{ij}.y_{ij}, Apk_j)e(R_{ij}, W).$$

8. *Aggregator*: One of the nodes who has role of aggregator, aggregates all $\{\sigma_{ij}\}|^{i=N_1,j=N_2}_{i=1,j=1}$ on message set $\{m_{ij}\}|^{i=N_1,j=N_2}_{i=1,j=1}$ related to pseudonym-public key pair $\{ID_{ij}, y_{ij}\}|^{i=N_1,j=N_2}_{i=1,j=1}$. The aggregate signature is:

$$\Omega = (\{R_{ij}|^{i=N_1,j=N_2}_{i=1,j=1}\}, V)$$

   The parameter V in above equation is calculated as follow:

$$V = \Sigma_{j=1}\Sigma_{i=1}V_{ij}$$

   The aggregator also calculate set
   $HY = \{hy_{ij} = h_{ij}.y_{ij}\}|^{i=N_1,j=N_2}_{i=1,j=1}$. After that he sends Ω, HY, and related $\{ID_{ij}, m_{ij}\}|^{i=N_1,j=N_2}_{i=1,j=1}$ to Aggregate verifier.

9. *Aggregate Verification*:
   The aggregate verifier does the following steps:

- The aggregate verifier computes:
  $W = H_1(SI)$ for all users,
  $Q_{ID^*_{ij}} = H_1(ID^*_{ij})$.
- He checks Eq. (14). if it holds, the receiver accepts Ω otherwise he rejects.

$$e(V,P) = \Pi_{j=1} e((\Sigma_{i=1}(Q_{ID^*_{ij}} + hy_{ij})), Apk_j) e(\Sigma_{i=1} R_{ij}, W) \tag{14}$$

The correctness of Eq. (14) is shown below: $e(V,P) = e(\Sigma_{j=1}\Sigma_{i=1}V_{ij}, P) =$
$e(V_{11} + V_{12} + \cdots + V_{N_1N_2}, P) =$
$e(V_{11}, P)e(V_{12}, P)...e(V_{N_1N_2}, P) =$
$\Pi_{j=1}\Pi_{i=1}e(V_{ij}, P) =$
$\Pi_{j=1}\Pi_{i=1}(e(Q_{ID^*_{ij}} + h_{ij}.y_{ij}, Apk_j)e(R_{ij}, W)) =$
$\Pi_{j=1}e((\Sigma_{i=1}(Q_{ID^*_{ij}} + hy_{ij})), Apk_j)e(\Sigma_{i=1}R_{ij}, W)$.

4. Security formal proof

In this section, we define our necessary security queries based on random oracle model and then prove that our proposed scheme has unique characteristics through three theorems.

4.1. Formal model definition

We should point out that the formal model proposed in [15] is considered as a major model for formal security analysis of CL-PKC schemes. Security models in [27,29,45–47] are almost similar and are based on [15]. We almost follow the security model proposed in [29] and merely add an new oracle (*Revealpseudonym*). By using [15] there are two adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ in our security model. These adversaries have the following properties:

- $\mathcal{A}_1$ is in charge of an outsider attacker who can replace user's public keys but cannot access to KGC's master secret.
- $\mathcal{A}_2$ has the role of malicious KGC. He has the KGC's master secret but cannot change the user's public keys.

There are two different cryptographic hash functions in our proposed scheme (H_1 and H_2). We modeled H_1 and H_2 by random oracles. When the adversary sends different queries on these oracles, the answer of hash functions has been returned to the adversary. In addition $\mathcal{A}_1$ and $\mathcal{A}_2$ have access to the following seven oracles through the simulator S:

1. *CreateUser*: When the adversary sends his query on $RID_{ij} \in \{0,1\}^*$, S checks whether this query submitted before or not. If the answer is yes, S has nothing to do otherwise S runs *revealsecretkey* and *revealpartialkey* queries and make list $L = \{RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij}\}$. In both cases S returns y_{ij} to adversary.
2. *Revealpartialkey*: When adversary sends his query on $RID_{ij} \in \{0,1\}^*$, S looks up into the list L. If corresponding line is found, S answers the query with psk_{ij} otherwise he returns $\perp$.
3. *Revealpseudonym*: When adversary sends his query on $RID_{ij} \in \{0,1\}^*$, S looks up into the list L. If corresponding line is found, S answers the query with ID_{ij} otherwise he returns $\perp$.
4. *Revealsecretkey*: When adversary sends his query on $RID_{ij} \in \{0,1\}^*$, S looks up into the list L. If corresponding line is found, S answers the query with x_{ij} otherwise he returns $\perp$.
5. *Revealpublikkey*: When adversary sends his query on $RID_{ij} \in \{0,1\}^*$, S looks up into the list L. If corresponding line is found, S answers the query with y_{ij} otherwise he returns $\perp$.
6. *Replacepublickey*: When adversary sends his query on $RID_{ij} \in \{0,1\}^*$ and the key pair (x^*_{ij}, y^*_{ij}), S looks up into the list L. If corresponding line is found, S updates list $L = \{RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij}\}$ to $L = \{RID_{ij}, x^*_{ij}, y^*_{ij}, psk_{ij}, ID^*_{ij}\}$ otherwise he returns $\perp$.
7. *Sign*: when adversary sends his intended $RID_{ij}, m_{ij} \in \{0,1\}^*$ to the oracle, S performs one of next three options:
   - If *Replacepublickey* was not done on RID_{ij}, S returns valid signature σ_{ij}.
   - If *Replacepublickey* was done, S returns the result of $sign(x^*_{ij}, psk_{ij}, m_{ij})$.
   - If RID_{ij} was not found, he returns $\perp$.

The following games are designed for $\mathcal{A}_1$ and $\mathcal{A}_2$.

Game 1: This game occurs between $\mathcal{A}_1$ and simulator S. It has three following phases:

1. *Setupphase*: S gets the security parameter 1^t and start *setup* algorithm. The system's public parameters will generate and send to $\mathcal{A}_1$. The S keep master key secret.
2. *Queryphase*: during the simulation $\mathcal{A}_1$ has access to all oracles.
3. *Guessphase*: $\mathcal{A}_1$ outputs σ_{ij} on the user with identities RID_{ij}, related public keys y_{ij} and corresponding pseudonym ID_{ij} on the message m_{ij}.

$\mathcal{A}_1$ wins the *Game*1 if all following conditions satisfy:

- σ_{ij} be a valid signature.
- The RID_{ij} is not submitted during *Revealpartialkey* query.
- *Sign* oracle never been performed on m_{ij} by RID_{ij}.

Game 2: This game occurs between $\mathcal{A}_2$ and simulator S. It has three following phases:

1. *Setupphase*: S gets the security parameter 1^t and start *setup* algorithm. System's public parameters will generate and send to $\mathcal{A}_2$.
2. *Queryphase*: During the simulation, $\mathcal{A}_2$ has access to all oracles.
3. *Guessphase*: $\mathcal{A}_2$ outputs σ_{ij} on the user with identities RID_{ij}, related public keys y_{ij} and corresponding pseudonym ID_{ij} on the message m_{ij}.

$\mathcal{A}_2$ wins the *Game*2 if all following conditions satisfy:

- σ_{ij} be a valid signature.
- The RID_{ij} is not submitted during *Revealsecretkey* query.
- *Sign* oracle never been performed on m_{ij} by RID_{ij}.

4.2. Formal proofs

In this section, we prove three theorems to represent the robustness of HACA protocol against the two different attackers $\mathcal{A}_1$ and $\mathcal{A}_2$. Also, we describe the security relation between the CLS and the CLAS structures.

Theorem 1 scrutinizes the CLS structure security against an outsider forger, who can change the public key of users but cannot access to the KGC secret.

Theorem 1. *If $\mathcal{A}_1$ wins the Game*1 *and forge the HACA CLS scheme with non-negligible advantage ϵ in random oracle model then he can solve the CDH problem with non-negligible probability in G_1.*

Proof. Let $\mathcal{A}_1$ can solve the CDH problem with non-negligible probability and $(P, X = aP, Y = bP)$ is a random ternary. $\mathcal{A}_1$ should execute the *Game*1 in three phases.

Setupphase: In this phase of *Game*1, S chooses some random RID_t, generates system parameters and sends *params* to $\mathcal{A}_1$. The S also puts $Apk_j = X = aP$. At last S generates list L and will answer to the queries.

Queryphase: $\mathcal{A}_1$ can send the following queries:

1. *HashAnswer*1 query: After sending some ID^*_{ij} to the oracle H_1, S maintains a list L_{H_1} in form of $(ID^*_{ij}, Q_{ID^*_{ij}}, \alpha_i)$. If L_{H_1} contains the line $(ID^*_{ij}, Q_{ID^*_{ij}}, \alpha_i)$, there is nothing to do and S returns

$Q_{ID^*_{ij}}$ to $\mathcal{A}_1$. Otherwise if $ij = t$ then S randomly picks up $\alpha_i \in Z^*_q$ and computes $Q_{ID^*_{ij}} = \alpha_i.Y \in G_1$ and inserts in L_{H_1} and returns to $\mathcal{A}_1$. If $ij \neq t$, S randomly picks up $\alpha_i \in Z^*_q$ and computes $Q_{ID^*_{ij}} = \alpha_i.P \in G_1$ and inserts in L_{H_1} and returns α_i to $\mathcal{A}_1$.

2. $HashAnswer2$ query: When $\mathcal{A}_1$ sends this query to H_2 oracle, S maintains a list L_{H_2} in form of $(m_{ij}, ID^*_{ij}, y_{ij}, R_{ij}, h_{ij})$. If L_{H_2} contains the line $(m_{ij}, ID^*_{ij}, y_{ij}, R_{ij}, h_{ij})$, there is nothing to do and S returns h_{ij} to $\mathcal{A}_1$. Otherwise S puts some random as h_{ij}, then inserts it in L_{H_2}, and sends to the $\mathcal{A}_1$.
3. $HashAnswer3$ query: When $\mathcal{A}_1$ sends this query to H_2 oracle, S maintains a list L_{H_3} in form of $(\beta.ID_{ij,1}, T_{ij}, ID_{ij,2} \oplus RID_{ij})$. If L_{H_3} contains the line $(\beta.ID_{ij,1}, T_{ij}, ID_{ij,2} \oplus RID_{ij})$, there is nothing to do and S returns $ID_{ij,2} \oplus RID_{ij}$ to $\mathcal{A}_1$. Otherwise S puts some random as $(ID_{ij,2} \oplus RID_{ij})$ then inserts it in L_{H_3} and sends $(ID_{ij,2} \oplus RID_{ij})$ to the $\mathcal{A}_1$.
4. $HashAnswer4$ query: When $\mathcal{A}_1$ sends this query to H_1 oracle, S maintains a list L_{H_4} in form of (SI, c_{ij}, W). If L_{H_4} contains the line (SI, c_{ij}, W), there is nothing to do and S returns W to $\mathcal{A}_1$. Otherwise S puts some random as c_{ij}, and computes $W = c_{ij}.X$ and update L_{H_4} and returns W to $\mathcal{A}_1$.
5. $Revealpseudonym$ query: When $\mathcal{A}_1$ sends $RID_{ij} \neq RID_t$ and $ID_{ij,1}$ to the related oracle, S searches whether list L contains $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$ and checks ID^*_{ij}. If $ID^*_{ij} \neq \perp$, S returns ID^*_{ij}. If $ID^*_{ij} = \perp$, S chooses two randoms as $(ID_{ij,2} \oplus RID_{ij})$ and T_{ij}. The S updates L_{H_3} and sends $ID^*_{ij} = (ID_{ij,1}, ID_{ij,2}, T_{ij})$ to $\mathcal{A}_1$.
6. $Revealpartialkey$ query: When $\mathcal{A}_1$ sends $RID_{ij} \neq RID_t$ to the related oracle, S searches whether list L contains $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$ and checks psk_{ij}. If $psk_{ij} \neq \perp$, S returns psk_{ij} to $\mathcal{A}_1$. If $psk_{ij} = \perp$, S searches L_{H_1} for $Q_{ID^*_{ij}}$. Then he computes psk_{ij} and returns it to $\mathcal{A}_1$. If L does not contain such a line, $psk_{ij} = \perp$ and S puts $psk_{ij} = \alpha_i.Apk_j = \alpha_i.X$ and sends it to $\mathcal{A}_1$.
7. $Revealpublickey$ query: When $\mathcal{A}_1$ sends $RID_{ij} \neq RID_t$ to the related oracle, S searches whether list L contains tuple $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$ and checks y_{ij}. If $y_{ij} \neq \perp$, S returns y_{ij} to $\mathcal{A}_1$. If $y_{ij} = \perp$, S chooses some random $ran_{ij} \in Z^*_q$ and computes $y_{ij} = ran_{ij}.P$. The S updates list L and returns y_{ij} to $\mathcal{A}_1$.
8. $Revealsecretkey$ query: When $\mathcal{A}_1$ sends $RID_{ij} \neq RID_t$ to related the oracle, S searches whether list L contains tuple $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$ and checks x_{ij}. If $x_{ij} \neq \perp$, S returns x_{ij} to $\mathcal{A}_1$. If $x_{ij} = \perp$, S chooses some random $ran_{ij} \in Z^*_q$ and computes $y_{ij} = ran_{ij}.P$. The S updates list L and returns x_{ij} to $\mathcal{A}_1$.
   If list L does not contain corresponding line S sets $x_{ij} = \perp$, and if $y_{ij} \neq \perp$, S chooses some random $ran_{ij} \in Z^*_q$ and computes $y_{ij} = ran_{ij}.P$. Now S updates y_{ij} in L and return ran_{ij} to $\mathcal{A}_1$.
9. $Replacepublickey$ query: When $\mathcal{A}_1$ sends pair (RID_{ij}, y^*_{ij}) to the related oracle, if L includes tuple $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$, S sets $y_{ij} = y^*_{ij}$ and $x_{ij} = \perp$, and updates L. Otherwise S sets $y_{ij} = y^*_{ij}$ and puts $x_{ij} = psk_{ij} = \perp$.
10. $Sign$ query: When $\mathcal{A}_1$ sends pair (RID_{ij}, m^*_{ij}) to $Sign$ oracle, S searches lists $L_{H_1}, L_{H_2}, L_{H_3}, L_{H_4}$ and L.

    (a) If $RID_{ij} = RID_t$ then S sets $Q_{ID^*_{ij}} = \alpha_i.Y$ from L and L_{H_1}:
    If L includes tuple $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$, S checks status of x_{ij}, if $x_{ij} = \perp$, S chooses some random $ran_{ij} \in Z^*_q$ and computes $y_{ij} = ran_{ij}.P$.
    If L does not contain tuple $(RID_{ij}, x_{ij}, y_{ij}, psk_{ij}, ID^*_{ij})$ S submits $revealpublickey$ and $revealsecretkey$ to achieve (x_{ij}, y_{ij}), then S updates L.
    S computes $W = c_{ij}.X$ from list L_{H_4}. Now he chooses random $r_{ij} \in Z^*_q$, computes

$$R_{ij} = r_{ij}.P - c^{-1}_{ij}.Q_{ID^*_{ij}} \tag{15}$$

    and

$$V_{ij} = r_{ij}.W + h_{ij}.x_{ij}.Apk_j \tag{16}$$

    S sends $\sigma_{ij} = (R_{ij}, V_{ij})$ to $\mathcal{A}_1$. The $\mathcal{A}_1$ can checks the validity of receiving signature as follow:

$$e(V_{ij}, P) = e(r_{ij}.W + h_{ij}.x_{ij}.Apk_j, P) =$$
$$e(h_{ij}.x_{ij}.Apk_j, P)e(r_{ij}.W, P) =$$
$$e(h_{ij}.x_{ij}.Ask_j.P, P)e(r_{ij}.W, P) =$$
$$e(h_{ij}.x_{ij}.P, Apk_j)e(r_{ij}.P, W) =$$
$$e(h_{ij}.x_{ij}.P, Apk_j)e(R_{ij} + c^{-1}_{ij}.Q_{ID^*_{ij}}, W) =$$
$$e(h_{ij}.x_{ij}.P, Apk_j)e(R_{ij}, W)e(c^{-1}_{ij}.Q_{ID^*_{ij}}, W) =$$
$$e(Q_{ID^*_{ij}} + h_{ij}.y_{ij}, Apk_j)e(R_{ij}, W)$$

    (b) If $RID_{ij} \neq RID_t$, S sets $Q_{ID^*_{ij}} = \alpha_i.P$ from L and L_{H_1}. Now he chooses random $r_{ij} \in Z^*_q$, computes $R_{ij} = r_{ij}.P$ and $V_{ij} = psk_{ij} + r_{ij}.W + h_{ij}.x_{ij}.Apk_j$. The S sends $\sigma_{ij} = (R_{ij}, V_{ij})$ to $\mathcal{A}_1$.

Guessphase: In both aforementioned cases, σ_{ij} is valid. By forking lemma [10] if S has an algorithm that can generate two valid signatures $\sigma^*_{ij} = (R^*_{ij}, V^*_{ij})$ and $\sigma^{**}_{ij} = (R^{**}_{ij}, V^{**}_{ij})$ by same randomness (different random oracles), he can solve the CDH problem. We can write Eq. (11) as:

$$V^*_{ij} = psk^*_{ij} + r^*_{ij}.W^* + h^*_{ij}.x^*_{ij}.Apk^*_j$$
$$V^{**}_{ij} = psk^*_{ij} + r^*_{ij}.W^* + h^{**}_{ij}.x^*_{ij}.Apk^*_j$$

For simulating above equations to generic CDH problem we can act as follow:

$$h^{*-1}_{ij}(V^*_{ij} - r^*_{ij}.W^*) = h^{*-1}_{ij}.psk^*_{ij} + x^*_{ij}.Apk^*_j$$
$$h^{**-1}_{ij}(V^{**}_{ij} - r^*_{ij}.W^*) = h^{**-1}_{ij}.psk^*_{ij} + x^*_{ij}.Apk^*_j$$
$$\rightarrow h^{*-1}_{ij}(V^*_{ij} - r^*_{ij}.W^*) - h^{**-1}_{ij}(V^{**}_{ij} - r^*_{ij}.W^*) =$$
$$psk^*_{ij}(h^{*-1}_{ij} - h^{**-1}_{ij}) =$$
$$a.Q_{ID^*_{ij}}(h^{*-1}_{ij} - h^{**-1}_{ij}) = \alpha_i.a.b.P.(h^{*-1}_{ij} - h^{**-1}_{ij}) \rightarrow$$
$$a.b.P = (\alpha_i(h^{*-1}_{ij} - h^{**-1}_{ij}))^{-1}.$$
$$(h^{*-1}_{ij}(V^*_{ij} - r^*_{ij}.W^*) - h^{**-1}_{ij}(V^{**}_{ij} - r^*_{ij}.W^*)) \tag{17}$$

Therefore, the answer of CDH problem founded. Now we will calculate the probability that $\mathcal{A}_1$ finds the CDH answer in polynomial time. We define the three following events:
$Event1$: S does not abort all $Revealpartialkey$ queries.
$Evnt2$: $\mathcal{A}_1$ can forge a valid signature.
$Event3$: The output of $\mathcal{A}_1$ is valid even if S does not abort all queries submitted by $\mathcal{A}_1$.

The probability of A1 be a winner after all above events have happened is (by the rule of product):

$$Pro[Event1 \cap Event2 \cap Event3] =$$
$$Pro[Event1].Pro[Event2 \mid Event1]. \tag{18}$$
$$Pro[Event3 \mid Event2 \cap Event1]$$

By assuming that the numbers of $Revealpartialkey$ queries and $HashAnswer1$ queries are q_k and q_{H_1},we have:

$$Pro[Event1] \geq (\frac{q_{H_1} - 1}{q_{H_1}})^{q_k} \tag{19}$$

$$Pro[Event2 \mid Event1] \geq \epsilon \tag{20}$$

$$Pro[Event3 \mid Event2 \cap Event1] \geq \frac{1}{q_{H_1}} \tag{21}$$

$$\rightarrow Pro[Event1 \cap Event2 \cap Event3] \geq (\frac{q_{H_1} - 1}{q_{H_1}})^{q_k}.\frac{\epsilon}{q_{H_1}} \tag{22}$$

Because of the value of ϵ is non-negligible, thus the value of $(\frac{q_{H_1}-1}{q_{H_1}})^{q_k}.\frac{\epsilon}{q_{H_1}}$ will be non-negligible, then the probability of finding the answer of CDH problem will be non-negligible too. Therefore we have a contradiction against the hardness of CDH. Theorem 1 was proved. □

We have shown that the CLS scheme is unforgeable against an outsider under random oracle model assumption. Now, we are describing the unforgeability status when a malicious KGC wants to forge.

Theorem 2. *If $\mathcal{A}_2$ wins the Game2 and forge the HACA CLS scheme with non-negligible advantage ϵ in the random oracle model, he can solve the CDH problem with non-negligible probability in G_1.*

Proof. Let us assume that $\mathcal{A}_2$ can solve the CDH problem with non-negligible probability and $(P, X = aP, Y = bP)$ be a random ternary in G_1.
Setupphase: In this phase of *Game2*, S chooses some random RID_t, generates system parameters and sends *params* to $\mathcal{A}_2$. The S also puts $Apk_j = X = aP$ and sends *params* to $\mathcal{A}_2$. Because $\mathcal{A}_2$ presenting a malicious KGC he has the secret key of airlines and therefore he has access to partial private keys, so $\mathcal{A}_2$ does not submit $HashAnswer1$ queries. The S holds list L as $\{RID_{ij}, x_{ij}, y_{ij}, ID^*_{ij}\}$.
Queryphase: $\mathcal{A}_2$ can send the following queries:

1. *CreateUser*: After sending some ID^*_{ij} to the oracle, S checks list L for y_{ij}. If L contains the line $\{RID_{ij}, x_{ij}, y_{ij}, ID^*_{ij}\}$ then there is nothing to do and S returns y_{ij} to $\mathcal{A}_2$. Otherwise if $ij = t$ then S randomly picks up $\alpha_{ij} \in Z^*_q$ and computes $y_{ij} = \alpha_{ij}.Y \in G_1$ and inserts in L and returns to $\mathcal{A}_2$. If $ij \neq t$, S randomly picks up $\alpha_{ij} \in Z^*_q$ and computes $y_{ij} = \alpha_{ij}.P \in G_1$ and inserts in L and returns y_{ij} to $\mathcal{A}_2$. In both cases $x_{ij} = \alpha_{ij}$.
2. $HashAnswer2$ query: When $\mathcal{A}_2$ sends query to H_2 oracle, S maintains a list L_{H_2} in form of $(m_{ij}, ID^*_{ij}, y_{ij}, R_{ij}, h_{ij})$. If L_{H_2} contains the line $(m_{ij}, ID^*_{ij}, y_{ij}, R_{ij}, h_{ij})$, there is nothing to do and S returns h_{ij} to $\mathcal{A}_2$. Otherwise S puts some random as h_{ij} then inserts it in L_{H_2}, and sends to the $\mathcal{A}_2$.
3. $HashAnswer3$ query: When $\mathcal{A}_2$ sends query to H_2 oracle, S maintains a list L_{H_3} in form of $(\beta.ID_{ij,1}, T_{ij}, ID_{ij,2} \oplus RID_{ij})$. If L_{H_3} contains the line $(\beta.ID_{ij,1}, T_{ij}, ID_{ij,2} \oplus RID_{ij})$, there is nothing to do and S returns $ID_{ij,2} \oplus RID_{ij}$ to $\mathcal{A}_2$. Otherwise S puts some random as $(ID_{ij,2} \oplus RID_{ij})$ then inserts it in L_{H_3} and sends $(ID_{ij,2} \oplus RID_{ij})$ to the $\mathcal{A}_2$.
4. $HashAnswer4$ query: When $\mathcal{A}_2$ sends query to H_1 oracle S maintains a list L_{H_4} in form of (SI, c_{ij}, W). If L_{H_4} contains the line (SI, c_{ij}, W), there is nothing to do and S returns W to $\mathcal{A}_2$. Otherwise S puts some random as c_{ij}, and computes $W = c_{ij}.P$ and update L_{H_4} and returns W to $\mathcal{A}_2$.
5. *Revealpseudonym* query: When $\mathcal{A}_2$ sends $RID_{ij} \neq RID_t$ and $ID_{ij,1}$ to the related oracle, S searches whether list L contains tuple $(RID_{ij}, x_{ij}, y_{ij}, ID^*_{ij})$ and checks ID^*_{ij}. If $ID^*_{ij} \neq \perp$, S returns ID^*_{ij}. If $ID^*_{ij} = \perp$, S chooses two randoms as $(ID_{ij,2} \oplus RID_{ij})$ and T_{ij}. The S updates L_{H_3} and sends $ID^*_{ij} = (ID_{ij,1}, ID_{ij,2}, T_{ij})$ to $\mathcal{A}_2$.
6. *Revealsecretkey* query: When $\mathcal{A}_2$ sends $RID_{ij} \neq RID_t$ to *Revealsecretkey* oracle, S searches whether list L contains tuple $(RID_{ij}, x_{ij}, y_{ij}, ID^*_{ij})$ and checks x_{ij}. If $x_{ij} \neq \perp$, S returns x_{ij} to $\mathcal{A}_2$. If $x_{ij} = \perp$, S chooses some random $ran_{ij} \in Z^*_q$ and computes $y_{ij} = ran_{ij}.P$. The S updates list L and returns x_{ij} to $\mathcal{A}_2$.
   If list L does not contain corresponding line S sets $x_{ij} = \perp$ and if $y_{ij} \neq \perp$ S chooses some random $ran_{ij} \in Z^*_q$ and computes $y_{ij} = ran_{ij}.P$. Now S updates y_{ij} in L and return ran_{ij} to $\mathcal{A}_2$.
7. *Sign* query: When $\mathcal{A}_2$ sends his request on (RID_{ij}, m^*_{ij}), S searches lists $L_{H_2}, L_{H_3}, L_{H_4}$ and L. If $ij = t$ then: $Apk_j = X, W = c_{ij}.P, y_{ij} = \alpha_{ij}.Y$ and $Q_{ID^*_{ij}} = H_1(ID^*_{ij})$.
   Guessphase: We have:

$$e(V_{ij}, P) = e(Q_{ID^*_{ij}} + h_{ij}.x_{ij}.Apk_j, P)e(R_{ij}, W) \rightarrow$$
$$e(h_{ij}.y_{ij}, Apk_j) =$$
$$e(V_{ij}, P)(e(Q_{ID^*_{ij}}, Apk_j)e(R_{ij}, W))^{-1} \rightarrow$$
$$e(V_{ij}, P) = e(V_{ij} - Ask_j.Q_{ID^*_{ij}} - c^{-1}_{ij}.R_{ij}, P) \quad (23)$$

as aforementioned above we know $Ask_j = a$, $y_{ij} = \alpha_{ij}.Y = \alpha_{ij}.b.P$, so above equation gets more simpler as follow:

$$\rightarrow e(h_{ij}.\alpha_{ij}.a.b.P, P) = e(V_{ij} - a.Q_{ID^*_{ij}} - c^{-1}_{ij}.R_{ij}, P)$$
$$\rightarrow a.b.P = (V_{ij} - psk_{ij} - c^{-1}_{ij}.R_{ij})(h_{ij}.\alpha_{ij})^{-1} \quad (24)$$

As a result, the answer of CDH problem founded in Eq. (24). Now we will calculate the probability that $\mathcal{A}_2$ finds the CDH answer in polynomial time. We define the three following events:
$Event1$: S does not abort all $Revealsecretkey$ queries.
$Evnt2$: $\mathcal{A}_2$ can forge a valid signature.
$Event3$: The output of $\mathcal{A}_2$ is valid even if S does not abort all queries submitted by $\mathcal{A}_2$.
The probability of $\mathcal{A}_2$ be a winner after all the above events achieved from Eq. (18). By assuming that the numbers of $Revealpsecretkey$ queries and $HashAnswer1$ queries are q_k and q_{H_1}, we have:

$$Pro[Event1] \geq (\frac{q_{H_1}-1}{q_{H_1}})^{q_k} \quad (25)$$
$$Pro[Event2 \mid Event1] \geq \epsilon \quad (26)$$
$$Pro[Event3 \mid Event2 \cap Event1] \geq \frac{1}{q_{H_1}} \quad (27)$$
$$\rightarrow Pro[Event1 \cap Event2 \cap Event3] \geq (\frac{q_{H_1}-1}{q_{H_1}})^{q_k}.\frac{\epsilon}{q_{H_1}} \quad (28)$$

Because of the value of ϵ is non-negligible, the value of $(\frac{q_{H_1}-1}{q_{H_1}})^{q_k}.\frac{\epsilon}{q_{H_1}}$ will be non-negligible again, then the probability of finding the answer of CDH problem will be non-negligible too. Therefore we have a contradiction against the hardness of CDH. Theorem 2 was proved. □

Here, we will illustrate the relation between the unforgeability of CLS scheme and the $CLAS$ one.

Theorem 3. *If the HACA CLS scheme is secure against adaptive chosen message attack, HACA CLAS scheme is secure against existential forgery in chosen aggregate model.*

Proof. Let us assume the adversary has an algorithm to solve the CDH problem with non-negligible probability. The algorithm chose random ternary $(P, X = aP, Y = bP)$ in the additive group G_1 of prime order q, selects challenge identity RID_i and then sends *param* to adversary. The algorithm puts $X = Apk_j$. Due to simplifying the relations we bring the proofs for a specific airline and respective aircraft ($j = constant$). Adversary submits the following queries:

- $HashAnswer1$ query: After sending some identity to the oracle H_1, S maintains a list L_{H_1} in form of $(ID^*_i, Q_{ID^*_i}, \alpha_i)$. If L_{H_1} contains the line $(ID^*_i, Q_{ID^*_i}, \alpha_i)$, there is nothing to do and S returns $Q_{ID^*_i}$ to $\mathcal{A}_1$. Otherwise if $ID^*_i = ID^*_t$ then S randomly picks up $\alpha_i \in Z^*_q$ and computes $Q_{ID^*_i} = \alpha_i.Y \in G_1$ and inserts in L_{H_1} and returns to $\mathcal{A}_1$. If $ID^*_i \neq ID^*_t$ then S randomly picks up $\alpha_i \in Z^*_q$ and computes $Q_{ID^*_i} = \alpha_i.P \in G_1$ and inserts in L_{H_1} and returns α_i to $\mathcal{A}_1$. In both cases algorithm maintains L_{H_1} and returns $Q_{ID^*_i}$ to the adversary.
  Adversary has output $\Omega^* = \{R^*_1, R^*_2, \ldots, R^*_{N_1}, V^*\}$ on set of n users with identities $\{RID^*_i\}|^{i=N_1}_{i=1}$ with public keys $\{y^*_i\}|^{i=N_1}_{i=1}$ and corresponding pseudonyms $\{ID^*_i\}|^{i=N_1}_{i=1}$ on message set $\{m^*_i\}|^{i=N_1}_{i=1}$. Algorithm achieves $(ID^*_i, Q^*_{ID_i}, \alpha_i)$ from list L_{H_1} only when $RID^*_k =$

Table 2
The comparison of computational overhead.

Scheme	Sign	Individual verification	Aggregate verification
Yang [40]	$2T_s = 0.78$ ms	$3T_p + T_s = 10.02$ ms	$(2N_1N_2 + 3)T_p + N_1T_s + 2N_1N_2T_h$
He [41]	$2T_s = 0.78$ ms	$2T_p + 3T_s = 7.59$ ms	$2T_p + (4N_1N_2 + 1)T_s$
Gowri [42]	$T_s = 0.39$ ms	$3T_s = 1.17$ ms	$4N_1N_2T_s$
HACA (our scheme)	$3T_s + T_h = 1.26$ ms	$3T_p + T_s + 2T_h = 10.2$ ms	$(2N_2 + 1)T_p + 2N_1N_2T_h$

RID_t^*, $RID_t^* \neq RID_i^*$ and $k \neq t$. validity of the aggregate signature can check by following equation:

$$e(V^*, P) = e(\sum_{i=1}(Q_{ID_i^*} + h_i^*.y_i^*), Apk)e(\sum_{i=1} R_i^*, W) \quad (29)$$

Algorithm can find h_i from list L_{H_2} and recover list L. At next step the algorithm puts $V_i^* = \alpha_i^*.Apk$, $R_i = r_i.P$ and can verify by $e(V_i^*, P) = e(Q_{ID_i^*}^*, Apk)$. Now algorithm presents two new parameters V^{**} and R^{**} as Equation:

$$V^{**} = V^* - \sum_{i=1,i\neq k} V_i^* = V^* - \sum_{i=1,i\neq k} \alpha_i^*.Apk =$$
$$\sum_{i=1}(psk_i^*.r_i^*.W + h_i^*.x_i^*.Apk) - \sum_{i=1,i\neq k} \alpha_i^*.Apk =$$
$$\sum_{i=1}(psk_i^*.r_i^*.W + h_i^*.x_i^*.Apk) - \sum_{i=1,i\neq k} \alpha_i^*.Ask.P =$$
$$\sum_{i=1}(psk_i^*.r_i^*.W + h_i^*.x_i^*.Apk) - \sum_{i=1,i\neq k} Ask.Q_{ID_i^*}^* =$$
$$\sum_{i=1}(psk_i^*.r_i^*.W + h_i^*.x_i^*.Apk) - \sum_{i=1,i\neq k} psk_i^* =$$
$$psk_k^* + \sum_{i=1} r_i^*.W + h_i^*.x_i^*.Apk \quad (30)$$

and

$$R^{**} = \sum_{i=1} R_i^* \quad (31)$$

At next step algorithm submits *Replacepublickey* query to replace y_i^* by $y_k^{**} = h_k^{*-1}.\sum_{i=1} h_i^*.y_i^*$. value of h_k^* has been defined as $h_k^* = H_2(m_k^*, ID_k^*, y_k^*, R_k^*)$. Now we can claim that (V^{**}, R^{**}) is a valid signature on message m_k^* that done by ID_k^*:

$$e(V^{**}, P) = e(psk_k^* + \sum_{i=1} r_i^*.W + h_i^*.x_i^*.Apk, P) =$$
$$e(psk_k^* + \sum_{i=1} h_i^*.x_i^*.Apk, P)e(\sum_{i=1} r_i^*.W, P) =$$
$$e(Q_{ID_k^*}^* + \sum_{i=1} h_i^*.x_i^*.P, Apk)e(\sum_{i=1} R_i^*, W) =$$
$$e(Q_{ID_k^*}^* + h_k^*.h_k^{*-1} \sum_{i=1} h_i^*.y_i^*, Apk)e(\sum_{i=1} R_i^*, W) =$$
$$e(Q_{ID_k^*}^* + h_k^*.y_k^{**}, Apk)e(R^{**}, W) \quad \square \quad (32)$$

Eq. (32) is the same as 13, therefore the attacker can forge signature of CLAS scheme with non-negligible probability. This gives contradiction against hardness of CDH problem.

5. Performance analysis and comparison

The main time-consuming operations in descending order are the pairing transform, elliptic curve multiplication, and map-to-point hash function respectively denoted by T_p, T_s and T_h. We waive time needed for point addition, multiplication and exponentiation operations.

According to the result of [48], if we apply Tate-pairing of 159-bit subgroup of a MNT curve and 80-bit security level on Intel Core i7 3.07 GHz CPU, then $T_p = 3.21$ ms, $T_s = 0.39$ ms and $T_h = 0.09$ ms. Table 2 represents the comparison between some ADS-B schemes with aggregate verification capability in terms of required time for signing, individual, and aggregate verifying. We assume that there are N_2 airlines that each of them has N_1 aircraft that sending messages to our ADS-B receiver.

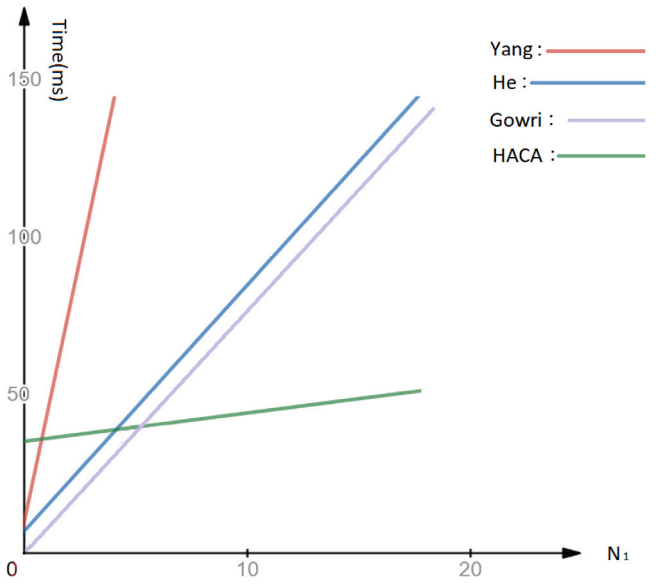

Fig. 3. The time needed to aggregate verification, a comparison between [40–42] and HACA ($N_2 = 5$).

As mentioned above, most time consuming operations in aggregate verification are bilinear pairings. As can be seen in Table 2, in our proposed scheme, N_2 is a "constant" and T_p is independent of N_1. The number of pairing operations required in the aggregate verification algorithm is a constant. In practice, because the number of airlines (N_2) is much lesser than aircraft (N_1), this independency better matches with the purpose of using aggregate signatures in our scheme. To clarify the effect of this independency, Fig. 3. shows a comparison between the HACA, the Yang et al. scheme [40], He et al. scheme [41], Gowri et al. scheme [42] in term of time needed for aggregate verification (assuming that $N_2 = 5$).

In Table 3 we bring a comparison between some wireless sensor network authentication protocols that support aggregate verification in terms of the security properties.

Horng's scheme [27], uses short-lived pseudonyms due to provide anonymity and privacy-preserving, but the others have no mechanism for anonymity.

ID-based schemes in [40,41], suffer from key escrow problem, because the PKG authority in these schemes is in charge of creating private keys, therefore existence of any malicious PKG leads to privacy disclosure. Other mentioned schemes [27,29,49] make a profit from KGC authority, that does not access to whole private key, accordingly they solve the key-escrow problem.

As mentioned in [28], a malicious-but-passive KGC can forge CLAS scheme in [27,49] by implanting a backdoor in public system parameters.

6. Conclusion

In this paper, we proposed a new Hierarchical Anonymous Certificateless Authentication (HACA) protocol for the ADS-B environment. The hierarchical topology of our scheme better matches with network realities. The scheme profits from using aggregate signatures

Table 3
The comparison of security properties.

Scheme	Application	Anonymity	Key-escrow resistant	Malicious KGC resistant
Yang [40]	ADS-B	×	×	×
He [41]	ADS-B	×	×	×
Gowri [42]	ADS-B	×	×	×
Horng [27]	VANET	✓	✓	×
Nong [49]	Robotic network	×	✓	×
Kumar [29]	Healthcare	×	✓	✓
Du [20]	IoT	×	✓	✓
HACA (our scheme)	ADS-B/UAS	✓	✓	✓

leads to acceleration of verification process in receiver side. By using certificateless public key cryptography we solve both key-escrow problem in identity-based authentication protocols and certificate management duty in public key infrastructure. The scheme provides conditional privacy-preserving and anonymity due to the use of unlinkable pseudonyms. This feature can be widely used in military to make links between the anonymous UAVs. Security of the construction is proved against adaptive chosen message attack under the widely accepted random oracle model.

CRediT authorship contribution statement

Amirhossein Asari: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing. **Mahdi R. Alagheband:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing. **Majid Bayat:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing. **Maryam Rajabzadeh Asaar:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing.

Acknowledgments

Approval of the version of the manuscript to be published (the names of all authors must be listed): Amirhossein Asari, Mahdi R. Alagheband, Majid Bayat, Maryam Rajabzadeh Asaar.

Declaration of competing interest

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

References

[1] M. Angelilli, L. Infante, P. Pacifici, A family of secondary surveillance radars based on conformal antenna array geometries, in: 2017 IEEE Radar Conference, RadarConf, IEEE, 2017, pp. 1681–1684.
[2] E. Cook, ADS-B, friend or foe: ADS-B message authentication for NextGen Aircraft, in: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, IEEE, 2015, pp. 1256–1261.
[3] A. Smith, R. Cassell, T. Breen, R. Hulstrom, C. Evers, Methods to provide system-wide ADS-B back-up, validation and security, in: 2006 IEEE/AIAA 25th Digital Avionics Systems Conference, IEEE, 2006, pp. 1–7.
[4] J. Johnson, H. Neufeldt, J. Beyer, Wide area multilateration and ADS-B proves resilient in Afghanistan, in: 2012 Integrated Communications, Navigation and Surveillance Conference, IEEE, 2012, pp. A6–1.
[5] A. Ranganathan, N.O. Tippenhauer, B. Škorić, D. Singelée, S. Čapkun, Design and implementation of a terrorist fraud resilient distance bounding system, in: European Symposium on Research in Computer Security, Springer, 2012, pp. 415–432.
[6] M. Strohmeier, V. Lenders, I. Martinovic, Security of $ADS-B$: State of the Art and Beyond, DCS, 2013.
[7] M.R. Manesh, N. Kaabouch, Analysis of vulnerabilities, attacks, countermeasures and overall risk of the Automatic Dependent Surveillance-Broadcast (ADS-B) system, Int. J. Crit. Infrastruct. Prot. 19 (2017) 16–31.
[8] M.R. Manesh, M. Mullins, K. Foerster, N. Kaabouch, A preliminary effort toward investigating the impacts of ADS-B message injection attack, in: 2018 IEEE Aerospace Conference, IEEE, 2018, pp. 1–6.
[9] M. Leonardi, E. Piracci, G. Galati, ADS-B jamming mitigation: a solution based on a multichannel receiver, IEEE Aerosp. Electron. Syst. Mag. 32 (11) (2017) 44–51.
[10] A. Shamir, Identity-based cryptosystems and signature schemes, in: Workshop on the Theory and Application of Cryptographic Techniques, Springer, 1984, pp. 47–53.
[11] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Annual International Cryptology Conference, Springer, 2001, pp. 213–229.
[12] E.S. Ismail, N. Tahat, R. Ahmad, A new digital signature scheme based on factoring and discrete logarithms, J. Math. Stat. 4 (4) (2008) 222.
[13] R. Housley, W. Polk, W. Ford, D. Solo, Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Tech. rep., 2002.
[14] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2003, pp. 416–432.
[15] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2003, pp. 452–473.
[16] D.H. Yum, P.J. Lee, Generic construction of certificateless signature, in: Australasian Conference on Information Security and Privacy, Springer, 2004, pp. 200–211.
[17] B.C. Hu, D.S. Wong, Z. Zhang, X. Deng, Key replacement attack against a generic construction of certificateless signature, in: Australasian Conference on Information Security and Privacy, Springer, 2006, pp. 235–246.
[18] K.-H. Yeh, C. Su, K.-K.R. Choo, W. Chiu, A novel certificateless signature scheme for smart objects in the Internet-of-Things, Sensors 17 (5) (2017) 1001.
[19] X. Jia, D. He, Q. Liu, K.-K.R. Choo, An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment, Ad Hoc Netw. 71 (2018) 78–87.
[20] H. Du, Q. Wen, S. Zhang, M. Gao, A new provably secure certificateless signature scheme for Internet of Things, Ad Hoc Netw. 100 (2020) 102074.
[21] L. Zhang, F. Zhang, A new certificateless aggregate signature scheme, Comput. Commun. 32 (6) (2009) 1079–1085.
[22] K.-A. Shim, Security models for certificateless signature schemes revisited, Inform. Sci. 296 (2015) 315–321.
[23] L. Zhang, B. Qin, Q. Wu, F. Zhang, Efficient many-to-one authentication with certificateless aggregate signatures, Comput. Netw. 54 (14) (2010) 2482–2491.
[24] H. Xiong, Z. Guan, Z. Chen, F. Li, An efficient certificateless aggregate signature with constant pairing computations, Inform. Sci. 219 (2013) 225–235.
[25] L. Cheng, Q. Wen, Z. Jin, H. Zhang, L. Zhou, Cryptanalysis and improvement of a certificateless aggregate signature scheme, Inform. Sci. 295 (2015) 337–346.
[26] D. He, M. Tian, J. Chen, Insecurity of an efficient certificateless aggregate signature with constant pairing computations, Inform. Sci. 268 (2014) 458–462.
[27] S.-J. Horng, S.-F. Tzeng, P.-H. Huang, X. Wang, T. Li, M.K. Khan, An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks, Inform. Sci. 317 (2015) 48–66.
[28] J. Li, H. Yuan, Y. Zhang, Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks, Networks 317 (2015) 48–66.
[29] P. Kumar, S. Kumari, V. Sharma, A.K. Sangaiah, J. Wei, X. Li, A certificateless aggregate signature scheme for healthcare wireless sensor network, Sustain. Comput. Inf. Syst. 18 (2018) 80–89.
[30] X. Lin, X. Sun, P.-H. Ho, X. Shen, GSIS: A secure and privacy-preserving protocol for vehicular communications, IEEE Trans. Veh. Technol. 56 (6) (2007) 3442–3456.
[31] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, 2005, pp. 11–21.
[32] M. Raya, P. Papadimitratos, J.-P. Hubaux, Securing vehicular communications, IEEE Wirel. Commun. 13 (5) (2006) 8–15.

[33] R. Lu, X. Lin, H. Zhu, P.-H. Ho, X. Shen, ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications, in: IEEE INFOCOM 2008-The 27th Conference on Computer Communications, IEEE, 2008, pp. 1229–1237.
[34] B. Ying, D. Makrakis, H.T. Mouftah, Privacy preserving broadcast message authentication protocol for VANETs, J. Netw. Comput. Appl. 36 (5) (2013) 1352–1364.
[35] M.A. Ferrag, L. Maglaras, A. Ahmim, Privacy-preserving schemes for ad hoc social networks: A survey, IEEE commun. Surv. Tutor. 19 (4) (2017) 3015–3045.
[36] Y. Pan, J. Li, Cooperative pseudonym change scheme based on the number of neighbors in VANETs, J. Netw. Comput. Appl. 36 (6) (2013) 1599–1609.
[37] P. Gope, T. Hwang, An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks, J. Netw. Comput. Appl. 62 (2016) 1–8.
[38] F. Wu, L. Xu, S. Kumari, X. Li, A.K. Das, M.K. Khan, M. Karuppiah, R. Baliyan, A novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks, Secur. Commun. Netw. 9 (16) (2016) 3527–3542.
[39] J. Li, W. Zhang, V. Dabra, K.-K.R. Choo, S. Kumari, D. Hogrefe, AEP-PPA: An anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities, J. Netw. Comput. Appl. 134 (2019) 52–61.
[40] A. Yang, X. Tan, J. Baek, D.S. Wong, A new ADS-B authentication framework based on efficient hierarchical identity-based signature with batch verification, IEEE Trans. Serv. Comput. 10 (2) (2015) 165–175.
[41] D. He, N. Kumar, K.-K.R. Choo, W. Wu, Efficient hierarchical identity-based signature with batch verification for automatic dependent surveillance-broadcast system, IEEE Trans. Inf. Forensics Secur. 12 (2) (2016) 454–464.
[42] G. Thumbur, N. Gayathri, P.V. Reddy, M.Z.U. Rahman, et al., Efficient pairing-free identity-based ADS-B authentication scheme with batch verification, IEEE Trans. Aerosp. Electron. Syst. 55 (5) (2019) 2473–2486.
[43] L. Martin, Introduction to Identity-Based Encryption, Artech House, 2008.
[44] C. Paar, J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer Science & Business Media, 2009.
[45] P. Kumar, S. Kumari, V. Sharma, X. Li, A.K. Sangaiah, S.H. Islam, Secure CLS and CL-AS schemes designed for VANETs, J. Supercomput. 75 (6) (2019) 3076–3098.
[46] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, Y. Tang, Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks, J. Netw. Comput. Appl. 106 (2018) 117–123.
[47] Y. Zhang, R.H. Deng, G. Han, D. Zheng, Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things, J. Netw. Comput. Appl. 123 (2018) 89–100.
[48] K.-A. Shim, CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks, IEEE Trans. Veh. Technol. 61 (4) (2012) 1874–1883.
[49] Q. Nong, Practical secure certificateless cryptographic protocol with batch verification for intelligent robot authentication, in: International Conference on Mechatronics and Intelligent Robotics, Springer, 2017, pp. 483–488.

Amirhossein Asari received his B.Sc degree in communication engineering from Semnan University in 2016 and M.Sc. degree from Azad University, science and research branch in secure communication and cryptography major. He won an award for top student and in the master program in 20162019. His research interests are applied cryptography, MANET authentication protocols, and user anonymity.

Mahdi R. Alagheband received his Ph.D. in 2013 from Azad University, Research and Science branch in Iran with his thesis "Lightweight cryptography in wireless sensor networks and RFID systems". Also, he was a research assistant at Information Systems and Security (ISSL) Laboratory, Sharif University of Technology and Assistant Professor at Electrical Engineering of Azad University for 6 years. Recently, he has published noticeable papers on applied cryptography in different networks. He has been a senior research fellow at Cybersecurity Research Lab (CRL) since early 2019. His research interests are lightweight cryptography, IoT security, WSN security, authentication protocols, and privacy-preserving solutions.

Majid Bayat received his Ph.D. from the Department of Mathematics and Computer Sciences at Kharazmi University in Tehran, Iran. He is presently an assistant professor of computer engineering of Shahed University, Tehran, Iran. His research interests include cryptographic protocols, smart grid and IoT security.

Maryam Rajabzadeh Asaar received her B.S. degree in Electrical Engineering from Shahid Bahonar University of Kerman, Kerman, Iran, in 2004, and received her M.S. and Ph.D. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 2008 and 2014, respectively. She is currently an assistant professor at Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. Her research interests include provable security, digital signatures, design and analysis of cryptographic protocols and network security and security in industrial control systems.