ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 721-729
doi: 10.18178/wcse.2017.06.126

Efficient and Secure Authentication Scheme Based on Certificateless Aggregate Signature in Space Networks

Hu Zhiyan [1 +], Du Xuehui [1] and Cao Lifeng [1]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Abstract. According to the requirements of the space networks for computing cost, communication cost and security, a new efficient certificateless aggregate signature algorithm is proposed. The signature length is only 2 group elements, the verification stage requires only 3 pairing operations, and the algorithm is proved secure based on the computational Diffie-Hellman problem under the random oracle model. Compared with the other similar algorithm, this algorithm achieves higher security with higher efficiency and lower cost of communication. Then according to the signature algorithm proposed in this paper, an authentication scheme with the function of mutual authentication, aggregate authentication and group key agreement is presented. The analysis results show that the scheme is secure and efficient, especially suitable for the large-scale space networks.

Keywords: space networks, certificateless cryptography, aggregate signature, authentication

1. Introduction

The space network structures with double plane of heaven and earth, which is based on the ground network and expanding with space-based network. The network consists of space-based backbone network, space-based access network and ground-based node network. As a result of its complex structure, it has the characteristics of heterogeneous, intermittent connectivity, large scale and high exposure, which brings great challenges to access authentication for space networks. It is required that not only to achieve rapid and safety verification, but also to ensure low communication cost to suit the unique characteristic of space networks.

The use of traditional public key cryptosystems (PKC) [1-6] to realize authentication for the space network, which brings the problem of complex certificate management and produces a large amount of computational and storage overhead. As a result of it uses the public key infrastructure (PKI) as a trusted third party to distribute digital certificates, it is not suitable for deployment in resource constrained space networks. Since Shamir proposed the identity-based public key cryptography (ID-PKC) [7], a large number of identity-based authentication scheme [8-11] has been proposed to simplify the certificate management pressure for PKI which brings the key escrow problem, it cannot satisfy the demand of high security in space networks.

In order to overcome the shortcoming of key escrow problem in the ID-PKC and complex certificate management in the PKC, Al-Riyami and Paterson[12] proposed certificateless public key cryptosystem (CL-PKC). The same year, Boneh et al. [13] first proposed the concept of aggregate signature in EUROCRYPT, it can aggregate multiple signatures to form a signature, so that only need to verify the aggregate signature to achieve verification. Gong et al. [14] first proposed two certificateless aggregate signature (CLAS) scheme combining aggregate signature algorithm with CL-PKC to achieve the purpose of verification, but bilinear pairing is proportional to the number of signers, there exists a problem of big calculation. Zhang et al. [15] proposed the scheme which pairings operation is reduced, but the length of the signature and the number of

+ Corresponding author. Tel.: +8615617764010; fax: +0371-81630094.
E-mail address: huzhiyan1234@126.com

signature is linear correlated, the communication cost is high. The scheme of Zhang et al. [16] used 5 pairing operations to realize aggregate authentication, scheme of Liu [17] is more efficient which only needs 3 pairing operations but introduces state information. Although the scheme of Xiong et al. [18] requires only 3 pairing operation, the signature length is long and the security is poor. Although the scheme of Chen et al. [19] achieves high security, but there exists a shortcoming of introduction of state information. The scheme proposed by He et al. [20] overcomes the defects of state information, but the computational overhead is relatively large. And the length of the [18-20] aggregate signature is n+1 group elements, the communication cost is larger. In summary, the existing authentication scheme based on certificateless aggregation signature scheme in the communication overhead, computation cost and security cannot adapt to the characteristics of the space network that needs to ensure the safety at the same time to achieve fast and efficient authentication.

We first introduce the background knowledge of CLAS scheme, and then propose a new and efficient CLAS algorithm, which proves to be safe based on the CDH problem in the random oracle model. Compared with the existing schemes, this scheme reduces the computational cost of verification and improves the communication efficiency while ensures the security. Finally, based on the CLAS scheme proposed in this paper, we propose an efficient and secure authentication scheme which is suitable for concurrent access authentication for a number of users in the space network.

2. Preliminaries

2.1. Bilinear Maps

Let l be a security parameter, q is a prime number of $l-bit$, G_1 represents a cyclic additive group of order q, G_2 represents a cyclic additive group of order q, G_T represents a cyclic multiplicative group of the same order, P is a generator of G_1, Q is a generator of G_2, we call map $e: G_1 \times G_2 \to G_T$ a bilinear map if the following properties are satisfied:

（1） Bilinearity：for all $a,b \in Z_q^*$， such that $e(aP,bQ) = e(P,Q)^{ab}$；

（2） Non-degeneracy：exist $P \in G_1, Q \in G_2$,such that $e(P,Q) \neq 1$；

（3） Computability： $\forall P,Q \in G_1$， it's efficient to compute $e(P,Q)$.

2.2. Mathematical difficulties

Computational Diffie–Hellman problem(CDH): G_1 is a cyclic additive group of order q with the generator P, the CDH problem is to compute g^{ab} when（g^a, g^b）is given, $a,b \in Z_q^*$.

3. System model

In space networks, the system model is composed of space-based access network, space-based backbone networks, ground-based node network and base station (BS). As shown in Figure 1. The base station completes the work of system initialization. Ground nodes makes a request to the space-based network nodes when it's in the space-based network node's coverage range. Then ground nodes and space-based network nodes can communicate safely with each other after mutual authentication and key establishment.

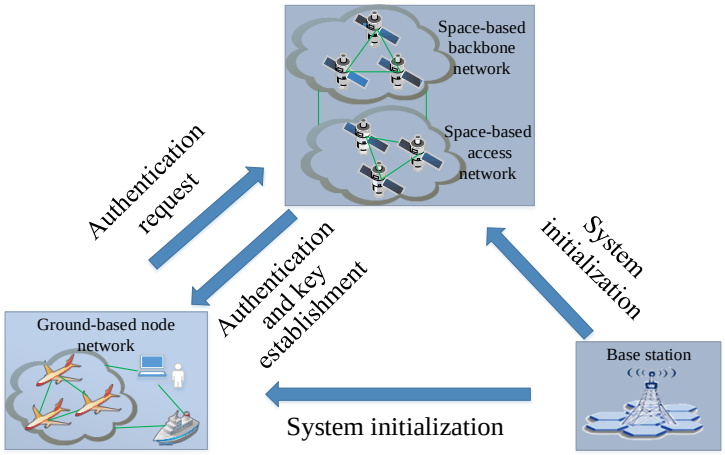

Figure. 1: Simplified model of space network authentication system

4. An efficient CLAS algorithm

We propose an efficient CLAS algorithm which consists of 7 polynomial time algorithms:

Setup: KGC uses the security parameters k to generate system parameter sets and the system master key as follows:

(1) Generate a cyclic additive group G_1 of order q, a cyclic multiplicative group G_2 of the same order and a bilinear map $e: G_1 \times G_1 \to G_2$.

(2) KGC selects $\lambda \in Z_q^*$ as the system master key and selects two random generator $P, Q \in G_1$. Then computes $P_{pub} = \lambda P$.

(3) KGC sets three secure hash functions $H_0: \{0,1\}^* \to G_1, H_1, H_2: \{0,1\}^* \to Z_q^*$.

(4) KGC publishes $paras = (G_1, G_2, e, P, Q, P_{pub}, H_0, H_1)$ and keeps master key λ secret.

Partial-private-key generation Gen_{PPK}: It is operated by KGC with a user's identity ID_i, system parameters *paras* and master key λ to generate user's partial private key D_i. Computes $Q_i = H_0(ID_i)$ and outputs $D_i = \lambda Q_i$.

Key generation Gen_K: It is operated by a user with the user's identity ID_i and a random number $x_i \in Z_q^*$ to generate the whole private key (x_i, D_i) and public key $P_i = x_i P$.

Sign: It is operated by each user with their own identity ID_i, message $M_i \in \mu$, plaintext space $\mu = \{0,1\}^*$, public key P_i and private key (x_i, D_i) to generate a valid signature σ_i by following steps:

(1) Choose $r_i \in Z_q^*$ and compute $R_i = r_i P$.

(2) Compute $hi = H_2(M_i \| ID_i \| P_i \| R_i)$, $t_i = H_1(M_i \| ID_i \| P_i \| R_i)$.

(3) Compute $S_i = D_i + x_i t_i Q + (h_i x_i + r_i) P_{pub}$.

(4) Output signature $\sigma_i = (R_i, S_i)$.

Single verification *SingleVerify*: Given the user's identity ID_i, the corresponding public key P_i and the aggregate signature σ_i on message M_i. The verifier computes $hi = H_2(M_i \| ID_i \| P_i \| R_i)$, $t_i = H_1(M_i \| ID_i \| P_i \| R_i)$, $Q_i = H_0(ID_i)$. Check equation $e(P, S_i) = e(P_{pub}, Q_i + h_i P_i + R_i) e(Q, P_i t_i)$, if it holds, output true, otherwise false.

Aggregate: Aggregate signature generator (any user can serve as an aggregate signature generator) runs the algorithm. On input a set of n users with identity set $\{ID_1, ID_2, ..., ID_n\}$, the corresponding public key set $\{P_1, P_2, ..., P_n\}$ and n distinct message-signature pairs set $\{(M_1, \sigma_1), (M_2, \sigma_2), ..., (M_n, \sigma_n)\}$, the generator computes $R = \sum_{i=1}^{n} R_i, S = \sum_{i=1}^{n} S_i$ and outputs the aggregate signature $\sigma = (R, S)$ on message set $\{M_1, M_2, ..., M_n\}$.

AggregateVerify: Given n users with identity set $\{ID_1, ID_2, ..., ID_n\}$, the corresponding public key set $\{P_1, P_2, ..., P_n\}$ and the aggregate signature $\sigma = (R, S)$, the verifier performs the following steps:

(1) For all $i (1 \le i \le n)$, computes $h_i = H_2(M_i \| ID_i \| P_i \| R_i)$, $t_i = H_1(M_i \| ID_i \| P_i \| R_i)$, $Q_i = H_0(ID_i)$.

(2) Check $e(P, S) = e(P_{pub}, \sum_{i=1}^{n} (Q_i + h_i P_i) + R) e(Q, \sum_{i=1}^{n} P_i \cdot t_i)$. If holds, output true, otherwise false.

5. Security analysis

5.1. Correctness analysis

$$
\begin{aligned}
& e(P, S) = e(P, \textstyle\sum_{i=1}^{n} (D_i + x_i t_i Q + (h_i x_i + r_i) P_{pub})) \\
& = e(P, \textstyle\sum_{i=1}^{n} (D_i + (h_i x_i + r_i) P_{pub})) e(P, \sum_{i=1}^{n} x_i t_i Q) \\
& = e(P_{pub}, \textstyle\sum_{i=1}^{n} Q_i + (h_i x_i + r_i) P) e(Q, \sum_{i=1}^{n} P_i \cdot t_i) \\
& = e(P_{pub}, \textstyle\sum_{i=1}^{n} (Q_i + h_i P_i) + R) e(Q, \sum_{i=1}^{n} P_i \cdot t_i)
\end{aligned}
$$

5.2. Security proof

Theorem 1. In the random oracle model, if a polynomial time type I adversary A_1 who has the non-negligible advantage ε forging the proposed CLAS scheme with at most q_{H_i} times $H_i (H_i = H_0, H_1, H_2)$ queries, q_k times $PPK(ID_i)$ queries, q_p times $PK(ID_i)$ queries, q_s times $S(M_i, ID_i, P_i)$ queries, there is an algorithm can solve CDH problem in time $t' = t + (q_{H_0} + q_k + q_p + 4q_s) t_{sm}$ and probability $\varepsilon' = \frac{1}{q_k + n} (1 - \frac{1}{q_k + n})^{q_k + n - 1}$, where

n is the number of signer, t_{sm} is the time spent on scalar multiplication in group G_1.

Proof: It is assumed that a type Ⅰ adversary can break the proposed CLAS with non-negligible advantage. Let C be an algorithm that can solve CDH problem. Given a CDH problem instance (P, aP, bP), C can solve CDH problem (compute abP) by interacting with A_1

Setup :The algorithm C sets $P_{pub}=aP$, chooses $w \in Z_q^*$ and sets $Q = wP$, returns set $paras = (G_1, G_2, e, P, Q, P_{pub}, H_0, H_1, H_2)$ to A_1.

Query： Hash functions H_0, H_1, H_2 are random oracles. It is assumed that A_1 can make at most q_{Hi} times $H_i(H_i = H_0, H_1, H_2)$ queries, q_k times $PPK(ID_i)$ queries, q_p times $PK(ID_i)$ queries and q_s times $S(M_i, ID_i, P_i)$ queries. A_1 can make following queries in an adaptive way.

H_0 queries: C manages an initially empty list L_0. The format of L_0 is $(ID_i, \alpha_i, Q_i, c_i)$.When C receiving the H_0 query on ID_i, C returns the corresponding value to A_1 if the corresponding entry already exists in the list L_0 .Otherwise, C tosses a coin $c_i \in \{0,1\}$ ($\Pr[c_i = 0] = \delta, \Pr[c_i = 1] = 1 - \delta$). When $c_i = 0$, C randomly picks $\alpha_i \in Z_q^*$, sets $Q_i = \alpha_i bP$ and returns Q_i to adversary A_1, then adds $(ID_i, \alpha_i, Q_i, c_i)$ to L_0. When $c_i = 1$, C randomly picks $\alpha_i \in Z_q^*$, sets $Q_i = \alpha_i P$ and returns Q_i to adversary A_1, then adds $(ID_i, \alpha_i, Q_i, c_i)$ to L_0.

H_1 queries: C manages an initially empty list L_1. The format of L_1 is $(M_i, ID_i, P_i, R_i, t_i)$.When C receiving the H_1 query on t_i, C returns the corresponding value to A_1 if the corresponding entry already exists in the list L_1. Otherwise, C randomly picks $t_i \in Z_q^*$, returns t_i to adversary A_1 and adds $(M_i, ID_i, P_i, R_i, t_i)$ to L_1.

H_2 queries: C manages an initially empty list L_2. The format of L_2 is $(M_i, ID_i, P_i, R_i, h_i)$.When C receiving the H_2 query on h_i, C returns the corresponding value to A_1 if the corresponding entry already exists in the list L_2. Otherwise, C randomly picks $h_i \in Z_q^*$, returns h_i to adversary A_1 and adds $(M_i, ID_i, P_i, R_i, h_i)$ to L_2.

Partial private key $PPK(ID_i)$ queries: C manages an initially empty list L_k. The format of L_k is (ID_i, x_i, P_i, D_i).When C receiving the $PPK(ID_i)$ query, C returns the corresponding value to A_1 if the corresponding entry already exists in the list L_k. Otherwise, C makes H_0 queries on ID_i to obtain the corresponding $(ID_i, \alpha_i, Q_i, c_i)$ at first, then performs the following steps:

(1) If $c_i = 0$, C aborts.

(2) If there is a four tuples (ID_i, x_i, P_i, D_i) in the list L_k, C sets $D_i = \alpha_i P_{pub}$ and returns D_i.

(3) Otherwise, C computes $D_i = \alpha_i P_{pub}$, sets $x_i = P_i = \perp$, returns the answer D_i and adds (ID_i, x_i, P_i, D_i) to the list L_k.

Public key $PK(ID_i)$ queries: When C receiving the $PK(ID_i)$ query, C returns the corresponding value to A_1 if the query has been made before, otherwise, C performs the following steps:

(1) If there is a four tuples (ID_i, x_i, P_i, D_i) in the list L_k, the public key is $\perp$, C randomly picks $x_i' \in Z_q^*$, sets $P_i' = x_i' P$ and updates tuples (ID_i, x_i, P_i, D_i) to (ID_i, x_i', P_i', D_i).

(2) Otherwise, C randomly chooses $x_i \in Z_q^*$, sets $P_i = x_i P$, returns P_i as the answer, sets $D_i = \perp$ and adds the tuples (ID_i, x_i, P_i, D_i) to the list L_k.

Secret value $SV(ID_i)$ queries: When C receiving the $SV(ID_i)$ query, C first checks whether there is a corresponding x_i in the list L_k, if exists, C returns x_i to A_1. Otherwise, C makes $PK(ID_i)$ queries to find the corresponding x_i and returns it to A_1.

Public key substitution $PKR(ID_i, P_i')$ queries: When C receiving the $PKR(ID_i, P_i')$ query, C first checks whether there is a corresponding tuple (ID_i, x_i, P_i, D_i) in the list L_k, if exists and $P_i \neq \perp$, C updates P_i in the tuple to P_i'. Otherwise, C makes $PK(ID_i)$ queries to find P_i and updates P_i to P_i'.

Sign $S(M_i, ID_i, P_i)$ queries: When C receiving the $S(M_i, ID_i, P_i)$ query, C makes H_0, H_1 and H_2 queries to obtain $(ID_i, \alpha_i, Q_i, c_i)$ from the list L_0, $(M_i, ID_i, P_i, R_i, t_i)$ from the list L_1 and $(M_i, ID_i, P_i, R_i, h_i)$ from the list L_2. Then forms the signature.

(1) If $c_i = 0$, C randomly chooses $r_i \in Z_q^*$, sets $R_i = r_i P - \alpha_i bP$, computes $S_i = x_i t_i Q + (h_i x_i + r_i) P_{pub}$ and outputs the signature $\sigma_i = (R_i, S_i)$.

(2) If $c_i = 1$, C randomly chooses $r_i \in Z_q^*$, sets $R_i = r_i P$, computes $S_i = \alpha_i P_{pub} + x_i t_i Q + (h_i x_i + r_i) P_{pub}$ and outputs signature $\sigma_i = (R_i, S_i)$.

Forgery: A_1 outputs the forged signature $\sigma^* = (R^*, S^*)$ on the users set $U^* = \{U_1^*, U_2^*, ...U_n^*\}$, the identity set $L_{ID}^* = \{ID_1^*, ID_2^*, ..., ID_n^*\}$, the corresponding public key set $L_{PK}^* = \{P_1^*, P_2^*, ..., P_n^*\}$ and the message set $L_M^* = \{M_1^*, M_2^*, ..., M_n^*\}$. And C obtains $(ID_i, \alpha_i, Q_i, c_i)$ from the list L_0, $(M_i, ID_i, P_i, R_i, t_i)$ from the list L_1, $(M_i, ID_i, P_i, R_i, h_i)$ from the list L_2. It is required that there is at least one $i \in [1, n]$ that A_1 has not made $PPK(ID_i)$ queries $(c_i^* = 0)$ and $S(M_i, ID_i, P_i)$ queries. Without losing generality, we assume that $i = 1$. The forged signature $\sigma^* = (R^*, S^*)$ satisfies $e(P, S) = e(P_{pub}, \sum_{i=1}^{n}(Q_i + h_i P_i) + R)e(Q, \sum_{i=1}^{n} P_i \cdot t_i)$. So we can get $e(P, S^*) = e(P_{pub}, \sum_{i=1}^{n}(Q_i^* + h_i^* P_i^*) + R^*)e(Q, \sum_{i=1}^{n} P_i^* \cdot t_i^*)$ and then $= e(P, \alpha_1^* abP + h_1^* x_1^* P_{pub} + r_1^* P_{pub})e(P, \sum_{i=2}^{n}(\alpha_i^* P_{pub} + h_i^* x_i^* P_{pub} + r_i^* P_{pub}))e(P, Q\sum_{i=1}^{n} x_i^* \cdot t_i^*)$ compute $abP = (S^* - h_1^* x_1^* P_{pub} - r_1^* P_{pub} - \sum_{i=2}^{n}(\alpha_i^* P_{pub} + h_i^* x_i^* P_{pub} + r_i^* P_{pub}) - Q\sum_{i=1}^{n} x_i^* \cdot t_i^*)\alpha_1^{*-1}$. Otherwise, C aborts.

C can compute abP through the following events:

(1) $E1$: C does not abort when A_1 makes $PPK(ID_i)$ queries.

(2) $E2$: A_1 produces a valid aggregate signature.

(3) $E3$: When $E2$ happens, $c_1^* = 0$ and $c_i^* = 1$ for $i \in [2, n]$ are required.

The success probability of C is $\Pr[E1 \wedge E2 \wedge E3] = \Pr[E1]\Pr[E2 \mid E1]\Pr[E3 \mid E1 \wedge E2]$.

C does not abort with a probability of $1-\delta$ during partial private key queries. $\Pr[E1] \geq (1-\delta)^{q_k}$ as a result of A_1 makes q_k times partial private key queries. C does not abort when making $PPK(ID_i)$ queries , it is difficult for adversary A_1 to distinguish between simulated and real environments, so $\Pr[E2 \mid E1] = \varepsilon$. $\Pr[E3 \mid E1 \wedge E2] = \delta(1-\delta)^{n-1}$ when $c_1^* = 0$ and $c_i^* = 1$ for all $i \in [2, n]$. In the end, we can get $\varepsilon' = \Pr[E1 \wedge E2 \wedge E3] \geq \delta(1-\delta)^{q_k+n-1}\varepsilon$, when $\varepsilon = \frac{1}{q_k + n}$, the maximum value of ε' is $\varepsilon'_{\max} = \frac{1}{q_k + n}(1 - \frac{1}{q_k + n})^{q_k+n-1}$.

The operation time of algorithm C is the sum of forging time of adversary A_1, query time of q_{H_i} times $H_i(H_i = H_0, H_1, H_2)$ queries, the time of q_k times $PPK(ID_i)$ queries, the time of q_p times $PK(ID_i)$ queries, the time of q_s times $S(M_i, ID_i, P_i)$ queries. Every q_{H_0} queries, q_k queries or q_p queries requires one time scalar multiplication, every q_s queries requires 4 times scalar multiplication. So $t' = t + (q_{H_0} + q_k + q_p + 4q_s)t_{sm}$.

Therefore if there is a type Ⅰ adversary who can break the proposed CLAS with non-negligible advantage, C can solve CDH problem.

Theorem 2. In the random oracle model, if a polynomial time type Ⅱ adversary A_2 who has the non-negligible advantage ε forging the proposed CLAS scheme with at most q_{H_i} times $H_i(H_i = H_0, H_1, H_2)$ queries, q_v times $SV(ID_i)$ queries, q_p times $PK(ID_i)$ queries, q_s times $S(M_i, ID_i, P_i)$ queries, there is an algorithm can solve CDH problem in time $t' = t + (q_v + q_p + 4q_s)t_{sm}$ and probability $\varepsilon' = \frac{1}{q_k + n}(1 - \frac{1}{q_k + n})^{q_k+n-1}$.

Proof: Be similar to the theorem 1.

6. The authentication scheme based on CLAS

In this paper, we design an authentication scheme based on CLAS scheme, which includes the system initialization phase, access authentication in the space network and the aggregate authentication phase.

6.1. System initialization

The base station complete the system initialization phase, which consists of setup, partial-private-key generation, key generation in section 4. So I won't go into much detail here. All network nodes (including network access node, etc.) complete the system initialization before deployment.

6.2. Access authentication in the space network

6.2.1 Authentication process

Access authentication will happen between ground nodes and space-based access nodes, space-based access nodes and space-based backbone nodes, ground nodes and space-based backbone nodes. The

authentication process is similar. It is assumed that the access authentication and key agreement process is completed by the user terminal UN (include ground nodes) and the network access node AN (include space-based access nodes and space-based backbone nodes) after the system initialization phase is finished. Before the formal process of the access authentication, it is necessary to select the access node. When the node UN is within the coverage of AN, it can receive the periodic broadcast message of AN, the format of Broadcast-message is $AN \to *: ID_{AN}, paras_{AN}, \sigma_{AN}(ID_{AN}, paras_{AN})$. The UN can check whether they are in the coverage range of AN When the UN receives this message. The UN can get system parameters $paras_{AN}$ and the identity of AN ID_{AN}. Compared with the parameters owned by AN, the UN can check whether the parameters are the same to confirm the identity of AN. The UN may be within the coverage of multiple ANs. When a UN receives a broadcast message transmitted by a plurality of ANs, the AN selects the highest signal strength to issue an access request. Access authentication process requires 2 interactions, as shown in Figure 2.

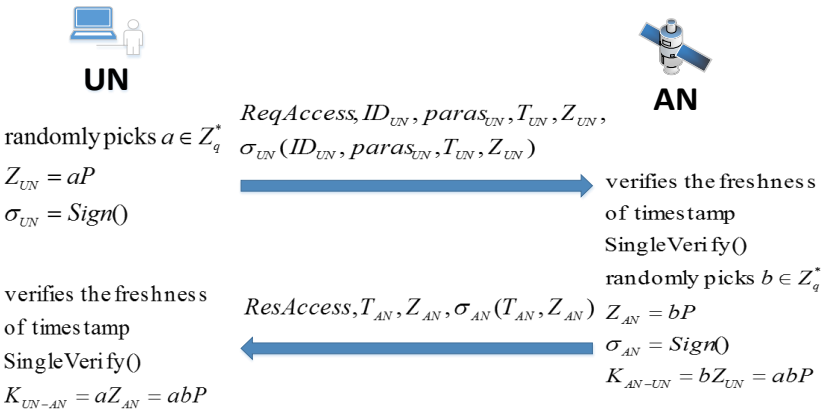

Figure. 2: Access authentication process in space network

(1) The UN randomly chooses $a \in Z_q^*$, computes $Z_{UN} = aP$, generates the aggregate signature σ_{UN} with the proposed CLAS algorithm and send the message $UN \to AN: ReqAccess, ID_{UN}, paras_{UN}, T_{UN}, Z_{UN}, \sigma_{UN}(ID_{UN}, paras_{UN}, T_{UN}, Z_{UN})$ to the AN.

(2) The AN verifies the freshness of timestamp T_{UN} at first when it receives the message. And then it verifies the legality of UN and the message through single verification algorithm *SingleVerify*(). If it does not pass, the process is abandoned, otherwise, the AN randomly picks $b \in Z_q^*$, computes $Z_{AN} = bP$ and the signature σ_{AN}, computes session key $K_{AN-UN} = bZ_{UN} = abP$ based on the received message. The transmitted message *ResAccess* is as follows: $AN \to UN: ResAccess, T_{AN}, Z_{AN}, \sigma_{AN}(T_{AN}, Z_{AN})$

(3) The UN verifies the freshness of timestamp at first when it receives the message. And then it verifies the legality of AN through single verification algorithm *SingleVerify*() to achieve mutual authentication. Computes session key $K_{UN-AN} = aZ_{AN} = abP$.

6.2.2 Key update

In order to enhance the security of the communication between AN and UN, the session key needs to be updated periodically.

(1) The UN issues key update request *ReqRefresh* to AN, randomly chooses $a' \in Z_q^*$, computes $Z_{UN}' = a'P$, signs the message using *Sign*(), then encrypts the messages with K_{UN-AN} and transmits it to AN. The format of the message is as follows: $UN \to AN: K_{UN-AN}(ReqRefresh, ID_{UN}, T_{UN}, Z_{UN}', \sigma_{UN}(ReqRefresh, ID_{UN}, T_{UN}, Z_{UN}'))$.

(2) The AN decrypts the message with the session key K_{AN-UN}, verifies the freshness of timestamp T_{UN} and the validity of the signature. If it passes, the AN randomly picks $b' \in Z_q^*$, computes $Z_{AN}' = b'P$, signs the message using *Sign*() with key refresh response *ResRefresh*, encrypts the messages with K_{AN-UN} and transmits it to UN. The format of the message is as follows: $AN \to UN: K_{AN-UN}(ResRefresh, T_{AN}, Z_{AN}', \sigma(ResRefresh, T_{AN}, Z_{AN}'))$.

The AN and UN can update the session key to $K_{AN-UN} = K_{UN-AN} = a'b'P$ through one round interaction after mutual authentication.

6.3. Aggregate authentication in the space network

In space networks, there is a situation that the cluster nodes are connected in parallel, such as the battleplane group or the ground cluster user accessing to the satellite. Because the resource of space networks is limited, the overall cost of signature verification affects the overall performance of the authentication service node directly. According to the group key agreement protocol proposed by Sun et al. [21], we design the aggregate authentication scheme to achieve batch verification for cluster nodes and negotiate a group key to realize security communication in group members.

This scheme needs three steps and two rounds to complete authentication and group key agreement:

(1) Every user $UN_i(1 \le i \le n)$ randomly picks $a_i \in Z_q^*$, computes L_i, Z_i using other users' public key as follows: $Li = a_i x_o P, Zi = x_i(x_{i+1}P - x_{i-1}P)$, computes $\sigma_i = (R_i, S_i)$ based on the proposed CLAS scheme $Sign()$. U_i sends (σ_i, L_i, Z_i) to AN.

(2) When the AN receives all parameters transmitted from all users $UN_i(1 \le i \le n)$, it computes parameters that are needed based on $Aggregate()$ and $AggragateVerify()$. The AN aggregate verifies all users' signatures through $e(P,S) = e(P_{pub}, \sum_{i=1}^{n}(Q_i + h_i P_i) + R)e(Q, \sum_{i=1}^{n} P_i \cdot t_i)$. If it passes, the AN randomly picks $a_0 \in Z_q^*$, computes $A_i = L_i x_0^{-1} = a_i P, Y = a_0 P + A_1 + A_2 + \ldots + A_n, O_i = a_0 A_i (1 \le i \le n)$ and $Z_i^{'} = (n-1)Z_i + (n-2)Z_{i+1} + \ldots + Z_{i-2}(1 \le i \le n)$, computes $\sigma_0 = (R_0, S_0)$ with the algorithm $Sign()$ and broadcasts $(\sigma_0, O_1, O_2, \ldots On, Y, Z_1^{'}, Z_2^{'}, \ldots Z_n^{'})$ to all users $UN_i(1 \le i \le n)$。

(3) Every user $UN_i(1 \le i \le n)$ verifies the legality of the AN through the proposed algorithm $SingleVerify()$. If it passes, each UN computes $Ka_i = Y - O_i a_i^{-1}, Kb_i = nx_i x_{i-1}P + Z_i^{'}$ .The group session key is $GK = Ka_i + Kb_i = A_1 + A_2 + \ldots + A_n + x_{i-1}x_i P + x_i x_{i+1} P + x_{i+1}x_{i+2}P + \ldots + x_{i-2}x_{i-1}P$.

7. Performance comparison

At first, we compare the proposed CLAS algorithm in efficiency, communication cost and security with the schemes in literatures [14-20], as shown in Table 1, the efficiency consists of single signature algorithm, aggregate verification. Communication cost includes aggregate signature length, the length of the public key and private key length. Security includes resistance to the first class of adversary and resistance to the second type of adversary. Symbol definitions are shown in table 2. At the same point, all these algorithms used the characteristic of bilinear pairings to realize aggregation verification, and they are based on the certificateless public key cryptosystem to overcome the key escrow problem. At different points, the proposed CLAS scheme in this paper is relatively efficient in the single signature algorithm. The aggregate verification only needs 3 pairing operations which is independent of the number of signers, the efficiency is high. The private key length is one group element length, aggregate signature length is two group element length, and the communication cost is low. What's more, our scheme can resist the first and second types of adversaries in safety. Compared with other schemes, it can achieve low communication cost and high efficiency while ensuring security.

Table 1: Comparison of CLAS schemes

Scheme	Efficiency		Communication cost			Security	
	Single signature algorithm	Aggregate verification	aggregate signature length	public key length	private key length	First type	Second type
[14-1]	2SM+1H+1Add	nH+(2n+1)P	(n+1)L	2L	L	√	-
[14-1]	3SM+2H+2Add	2nH+(n+2)P	2L	2L	2L	-	-
[15]	3SM+2H+2Add	(n+3)P+(2n+1)H+(2n-2)Add	(n+1)L	L	L	√	√
[16]	5SM+4H+4Add	2nSM+5P+(3n+3)H+(4n-4)Add	2L	L	2L	√	√
[17]	4SM+2H+2Add	2nSM+3P+(2n+1)H+(2n-2)Add	2L	L	L	√	√
[18]	3SM+1H+2Add	2nSM+3P+2nH +(3n-2)Add	(n+1)L	L	L	√	×
[19]	4SM+4H+2Add	2nSM+4P+(3n+2)H+(3n-3)Add	(n+1)L	L	L	√	√
[20]	3SM+2H+2Add	2nSM+3P+3nH +(3n-2)Add	(n+1)L	L	L	√	√
Our scheme	3SM+2H+2Add	2nSM+3P+3nH+(2n-1)Add	2L	L	L	√	√

Table 2: Symbol definition

Symbol	Definition
SM	Scalar multiplication in G_1
H	Hash function operation
Add	Add in G_1
P	Bilinear pairing operation

e	exponentiation in G_2
MAC	Message authentication code
L	Element length in G_1
√	Be able to resist the attack
×	Unable to resist the attack
-	Non formal proof

The authentication scheme in this paper is compared with the existing authentication schemes (Our scheme achieves aggregate verification and group key agreement, this part only considers the cost on aggregate verification in comparison with other schemes). As shown in Table 3. Compared with [17], although two schemes used 3 pairing to achieve authentication, the interaction times and computational cost is lower in the independent verification. Compared with the literature [20][22], although our scheme computation cost is high in independent verification, the bilinear pairing operation is reduced to constant magnitude in the aggregate authentication, the computation is significantly lower than before. In the aspect of security, the security of this scheme is based on the proposed CLAS algorithm. Therefore, the scheme is more efficient and more suitable for large scale space networks.

Table 3: The comparison of authentication scheme

Scheme	Independent authentication		Aggregate authentication	
	Computation cost	Communication cost	Computation cost	Communication cost
[17]	3P+8SM+5H+3Add+1MAC	3	3P+(2n+4)SM+(2n+3)H+(4n-2)Add	n+1
[20]	2P+1SM+2H+1Add	2	2nP+1SM +nAdd+(n+1)H	n+1
[22]	2P+8SM	2	(n+1)P+(5n+1)SM+e	n+1
Our scheme	3P+7SM+5H+4Add	2	3P+(2n+3)SM+(3n+2)H+(4n-1)Add	n+1

8. Conclusions and future work

According to the characteristics of the space network, we propose an efficient CLAS scheme based on bilinear map, and puts forward to an authentication scheme based on it. Compared with the existing CLAS algorithm, our algorithm realizes higher security with the lower communication cost and the higher efficiency. And the authentication scheme can realize fast mutual authentication, aggregate authentication and session key agreement while achieve high efficiency with high safety. It is more suitable for authentication in large-scale space networks with limited resource.

In future work, how to realize privacy protection of the node will be the key issue in the next step. The space network is a high exposure network, it is easy to be attacked, so we should take steps to achieve anonymous aggregate authentication in the space network.

9. Acknowledgements

This paper is supported by the National Key Research and Development Program of China (No.2016YFB0501900), the National Programs for High Technology Research and Development of China (No. 2015AA016006) and the National Natural Science Foundation of China (No. 61502531).

10.References

[1] Kambourakis G, Rouskas A, Kormentzas G, et al. Advanced SSL/TLS-based authentication for secure WLAN-3G interworking. *IEE Proceedings – Communications.* 2004, 151 (5): 501-506.

[2] Yang G, Wong D S, Deng X. Anonymous and Authenticated Key Exchange for Roaming Networks. *IEEE Transactions on Wireless Communications*. 2007, 6 (9): 3461-3472.

[3] Hou H F, Liu G Q, Ji X S, et al. Provable security authentication scheme based on public key for heterogeneous wireless network. *Journal of Electronics & Information Technology*. 2009, 31 (10): 2385-2391(in Chinese).

[4] He D, Ma M, Zhang Y, et al. A strong user authentication scheme with smart cards for wireless communications. *Computer Communications*. 2011, 34 (3): 367-374.

[5] Mahshid M K, Eslamipoor R. An optimized authentication protocol for mobile networks. *Neural Computing and Applications*. 2014, 25 (2): 1-7.

[6] Mallissery S, Pai M M M, Smitha A, et al. Improvising the Public Key Infrastructure to Build Trust Architecture

for VANET by using Short - Time Certificate Management and Merkle Signature Scheme. *Ultrasound in Obstetrics & Gynecology*. 2014, 44 (S1): 40-40.

[7] A. Shamir, Identity based cryptosystems and signature schemes. In: G.R.Blakley, D. Chaum (Eds.). *Crypto'84, LNCS 196*, Springer-Verlag, Santa Barbara,California, USA, 1984, pp. 47–53.

[8] Kate A, Zaverucha G M, Hengartner U. Anonymity and security in delay tolerant networks. *International Conference on Security and Privacy in Communications Networks and the Workshops*. *2007. SECURECOMM*. IEEE Xplore, 2007, pp. 504-513.

[9] Yang G, Huang Q, Wong D S, et al. Universal authentication protocols for anonymous wireless communications. *IEEE Transactions on Wireless Communications*. 2010, 9 (1): 168-174.

[10] Jiang Q, Ma J F, Li G S, et al. Identity-based roaming protocol with anonymity for heterogeneous wireless networks. *Journal on Communications*. 2010, 31 (10): 138-145(in Chinese).

[11] Lo N W, Tsai J L. An Efficient Conditional Privacy-Preserving Authentication Scheme for Vehicular Sensor Networks Without Pairings. *IEEE Transactions on Intelligent Transportation Systems*. 2016, 17 (5): 1319-1328.

[12] S. Al-Riyami, K. Paterson, Certificateless public key cryptography. In: C.S. Laih(Ed.). *ASIACRYPT 2003, LNCS 2894*, Springer-Verlag, Taipei, 2003, pp. 452–473.

[13] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps. In: E. Biham (Ed.). *EUROCRYPT 2003, LNCS2656*, Springer-Verlag, Warsaw, Poland, 2003, pp. 416–432.

[14] Gong Z, Long Y, Hong X, et al. Two Certificateless Aggregate Signatures From Bilinear Maps. *Eighth Acis International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/distributed Computing*. IEEE Computer Society, 2007, pp. 188-193.

[15] Zhang L, Zhang F. A new certificateless aggregate signature scheme. *Computer Communications*. 2009, 32 (6): 1079-1085.

[16] Zhang L, Qin B, Wu Q, et al. Efficient many-to-one authentication with certificateless aggregate signatures. *Computer Networks*. 2010, 54 (14): 2482-2491.

[17] Liu H, Shi R H, Zhang S, et al. Efficient anonymous roaming authentication scheme using certificateless aggregate signature in wireless network. *Journal on Communications*. 2016, 37 (7): 182-192 (in Chinese).

[18] Xiong H, Guan Z, Chen Z, et al. An efficient certificateless aggregate signature with constant pairing computations. *Information Sciences*. 2013, 219 (10): 225-235.

[19] Chen H, Wei SM, Zhu CJ, Yang Y. Secure certificateless aggregate signature scheme. *Journal of Software*. 2015, 26 (5): 1173-1180 (in Chinese).

[20] Liu H, Wang S, Liang M, et al. New Construction of Efficient Certificateless Aggregate Signatures. *International Journal of Security & Its Applications*. 2014, 8 (1): 411-422.

[21] Sun H M, He B Z, Chen C M, et al. A provable authenticated group key agreement protocol for mobile environment. *Information Sciences*. 2015, 321 (10): 224-237.

[22] Xu G Y, Chen X Y, Du X H, An authentication scheme using hierarchical identity based signature in large-scale delay tolerant networks. *Journal of Electronics & Information Technology*. 2013, 35 (11): 2615-2622(in Chinese).