A Novel Certificateless Aggregation Signcryption Scheme Under Cloud Computing

Mingju Zhao and Yuping Peng
(Corresponding author: Yuping Peng)

School of Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450000, China
(Email:352720214@qq.com)

(Received Mar. 2, 2020; Revised and Accepted Nov. 10, 2020; First Online Feb. 15, 2021)

Abstract

The traditional certificateless aggregation signcryption scheme (CLASC) scheme has low computational efficiency and time-consuming. Therefore, this paper proposes a novel CLASC with non-bilinear pairings under the cloud computing environment. Based on the discrete logarithm problem, it is proved that the new scheme satisfies the confidentiality and unforgerability under the random oracle model. In the verification phase of the aggregation signature, the third party's secret information is not needed, so the new scheme meets the public verifiability. Compared with the state-of-the-art signcryption schemes, it reveals that the new scheme can achieve higher security at a lower computing rate under cloud computing.

Keywords: Certificateless Aggregation Signcryption; Cloud Computing; Non-bilinear Pairings; Random Oracle Model

1 Introduction

At present, more and more countries have invested in the research on the cloud computing and achieved fruitful results. The cloud computing has been widely used in food safety, public safety, health monitoring, intelligent transportation, security, environmental protection and many other industries [6, 8, 9, 15]. The network scale has also been expanding from a laboratory to a building to a community, and different systems have been integrated. With the expansion of network scale, the problems of cloud computing system are exposed. The application industries of the cloud computing, such as food safety and intrusion detection, require the cloud computing to be able to provide fast and accurate response to emergencies, users and managers, so as to achieve accurate communication between people and Things [7, 12, 18]. Meanwhile, it also needs to ensure that the network infrastructure has an economic deployment. This requires the system to operate in an efficient, reliable and secure mode, so cryptography is used to design secure and efficient algorithms and protocols, which is the focus of cloud computing research. The core technology to ensure information security is modern cryptography, which can ensure the confidentiality, integrity, availability and non-repudiation of information in the network environment. Where, confidentiality can be achieved by encryption, and authentication can be achieved by digital signature [21]. If you need to achieve confidentiality and authentication, the traditional public key cryptography is to use "sign first and then encrypt", but this method is inefficient. In 1997, Zheng *et al.* proposed the concept of signcryption and gave a specific scheme [20]. In 2002, Baek *et al.* defined the security model of signcryption scheme for the first time [2]. In practical application, when there are a large number of signers, recipients need to verify multiple ciphertexts at the same time. In order to enhance the efficiency of ciphertext validation, Selvi *et al.* [13] proposed the concept of aggregation signature making full use of the advantages of aggregation signature. In 2003, AlRiyami *et al.* [4] first proposed the certificateless aggregation signcryption (CLASC) system, which not only avoided the problem of public key certificate management and verification, but also solved the key escrow problem. In 2008, Barbosa *et al.* [1] proposed a certificate-free sign-off scheme for the first time and presented its corresponding security model. Subsequently, references [16, 17, 19] proposed the certificateless aggregation signcryption scheme.

2 Preliminaries

The equation of the elliptic curve is defined as $y^2 = x^3 + ax + b$ $(a, b \in F_p)$ on F_p(F_p represents a finite field with p elements, $p > 3$ is prime). The discriminant is $4a^3 + 27b^2 \neq 0 \mod p$. A set of all solutions on the elliptic curve and an infinite point O is represented by $E(F_p)$, that is, $E(F_p) = \{(x, y)|x, y \in F_p\}$, and satisfies the equation $y^2 = x^3 + ax + b \cup O$. The number of points on $E(F_p)$ is represented by q, which becomes the order of the elliptic curve.

- Discrete logarithm problem (DLP). Let G be an additive cyclic group with order q, and P is the generator of G. For $b \in Z_q^*$, finding an integer makes $b = aP$ be difficult.
- Computational Diffie-Hellmanproblem (CDHP). For unknown $a, b \in Z_q^*$, computing abP is difficult.

3 The Proposed Security Model

The security model for certificateless signcryption schemes is introduced by Barbosa and Farshim (2008). In this section, we propose a security model for certificateless aggregate signcryption schemes. The ciphertext indistinguishability and the existential unforgeability security models are used to capture the confidentiality and authenticity requirements, respectively. As for the adversarial model, we follow the common approach in the certificateless setting, which considers two types of adversaries. A Type I adversary A_I who does not have access to the master secret key but can replace the public key of any entity with another value and a Type II Adversary A_{II} who has access to the master secret key but is unable to perform public key replacement. We now define the required security games to capture.

The confidentiality property is defined based on the concept of indistinguishability of encryptions under adaptively chosen ciphertext attacks (IND-CCA2). We define the following two games against Type I and Type II adversaries.

Game I. The game is performed by a challenger C and a Type I adversary A_I.

1) Initialization. C runs the Setup algorithm to generate a master secret key msk and the public system parameters $params$. C keeps msk secret and gives $params$ to A_I. Note that A_I does not know msk.
2) Phase 1. A polynomially bounded number of the following queries is performed by A_I. The queries can be made adaptively so that answers to the previous queries might affect subsequent ones.
   a. $RequestPublicKey$. When A_I supplies an identity ID_u and requests u's public key, C responds with the public key P_u for the identity.
   b. $ExtractPartialPrivateKey$. When A_I supplies an identity ID_u and requests u's partial private key, C responds with the partial private key D_u for the identity.
   c. ReplacePublicKey. When A_I supplies an identity ID_u and a new valid public key value P'_u; C replaces the current public key value with the value P'_u.
   d. $ExtractSecretValue$. When A_I requests the secret value of an identity ID_u, the challenger returns the secret value x_u of u. The public key of u should not have been replaced by A_I.
   e. $Signcrypt$. When A_I submits a sender with an identity ID_S, a receiver with an identity ID_R, a message M and some state information Δ to the challenger, C responds by running the Signcrypt algorithm on the message M, the state information Δ, the sender's private key (D_S, x_S) and the receiver's public key P_R.
   f. $AggregateUnsigncrypt$. When A_I submits an aggregate ciphertext c, some state information Δ, senders with identities $ID_{i\,i=1}^{n}$ and a receiver with the identity ID_R, C checks the validity of c and if it is a valid ciphertext, then C returns the result of running the AggregateUnsigncrypt algorithm on the ciphertext c, the state information Δ, the receiver's private key (D_R, x_R) and the senders' public keys $P_{i\,i=1}^{n}$.
3) Challenge. When Phase 1 ends, the adversary outputs $n+1$ distinct identities $ID_{i\,i=1}^{*n}$, ID_R^*, some state information Δ^* and two sets of n messages $M_0^* = m_{0i\,i=1}^{*\,n}$, $M_1^* = m_{1i\,i=1}^{*\,n}$. Now, a bit μ is randomly chosen by C who then produces c^* as the aggregate signcryption of messages M_μ^* using the state information Δ^*, the private keys corresponding to $ID_{i\,i=1}^{*n}$ and the public key and the identity of u_R^*. The challenger returns c^* to the adversary.
4) Phase 2. The adversary can continue to probe the challenger as in Phase 1.
5) Response. The adversary returns a bit μ'.

We say that the adversary wins the game if $\mu' = \mu$, subject to the following conditions:

1) A_I never queries the partial private key for ID_R^*.
2) A_I cannot make an AggregateUnsigncrypt query on c^* under ID_R^* and $ID'^{n}_{i\,i=1}$ where at least for one i, $ID_i^* = ID'_i$. The only exception is when the public key P_i^* of all of the senders ID_j^* with $ID_j^* = ID'_j$ or that of the receiver P_R^* used to signcrypt M_μ^* have been replaced after the challenge was issued.

The advantage of A_I is defined as follows:

$$Adv_{A_I}^{IND-CLASC-CCA2} = |2Pr[\mu' - \mu] - 1|.$$

where $Pr[\mu' - \mu]$ denotes the probability that $\mu' = \mu$.

Game II. The game is performed by a challenger C and a Type II adversary A_{II}.

1) Initialization. C first generates $(params, msk)$ and outputs them to A_{II}.
2) Phase 1. A_{II} may adaptively make a polynomially bounded number of queries as in Game I. The only constraint is that A_{II} cannot replace any public keys. Note that since A_{II} knows the master secret key, it can compute the partial private key of any identity.

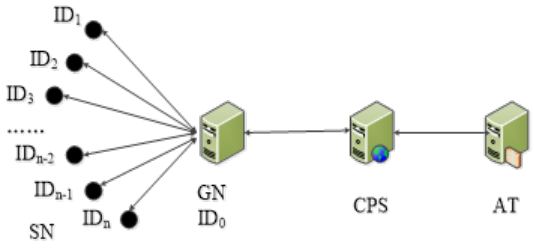

Figure 1: WSN architecture

3) Challenge. When Phase 1 ends, the adversary outputs $n+1$ distinct identities $ID^{*n}_{i\,i=1}$, ID^*_R, some state information Δ^* and two sets of n messages $M^*_0 = m^{*\,n}_{0ii=1}$, $M^*_1 = m^{*\,n}_{1ii=1}$. Now, a bit μ is randomly chosen by C who then produces c^* as the aggregate signcryption of messages M^*_μ using the state information Δ^*, the private keys corresponding to $ID^{*n}_{i\,i=1}$ and the public key and the identity of u^*_R. The challenger returns c^* to the adversary.

4) Phase 2. The adversary can continue to probe the challenger as in Phase 1.

5) Response. The adversary returns a bit μ'.

We say that the adversary wins the game if $\mu = \mu'$,and the following constraints are fulfilled:

1) A_{II} never queries the secret value for the challenge identity ID^*_R.

2) In Phase 2, A_{II} cannot make an AggregateUnsigncrypt query for the challenge ciphertext c^* under ID^*_R, where at least for one i, $ID^*_i = ID'_i$.

As in Game I, the advantage of A_{II} is defined as follows:

$$Adv^{IND-CLASC-CCA2}_{A_{II}} = |2Pr[\mu = \mu'] - 1|.$$

4 Proposed Certificateless Aggregation Signcryption Scheme

This paper proposes a novel Certificateless aggregation signcryption Scheme (CLASC) under cloud computing. A complete cloud computing system is composed of sensory node ($SN_i, 1 \le i \le n$), gateway node (GN), cloud platform server (CPS) and application terminal (AT), as shown in Figure 1.

The function of the SN is to transmit the collected data hop by hop along other sensing nodes and send it to the gateway node. The gateway node automatically saves the data and periodically transfers the automatically collected data to the Internet cloud platform server within a certain time interval. The cloud platform server sends the data to the application terminal for decryption and analysis. The cloud platform server is honest and reliable, responsible for system management and maintenance, including SN, GN and AT registration, private key distribution, etc. The cloud platform server communicates wirelessly with GN, GN and SN, and GN and GN. The implementation process is as follows:

1) System initialization. The algorithm is executed by GN. Enter the security parameter k and select a large prime number $q > 2^k$. Let G be a cyclic group of elliptic curves. And P is the generator of G. GN selects four collash-resistant hash functions: $H_1 : 0,1^* \times G \times G \to Z^*_q$, $H_2 : G \times G \to Z^*_q$, $H_3, H_4 : G \times G \times G \times G \to Z^*_q$. GN randomly selects the master key $s \in Z^*_q$ and preserves it secretly. GN computes the master key $P_{pub} = sP$. CPS publishes system public parameter $params = \{q, P, G, P_{pub}, H_1, H_2, H_3\}$.

2) Key generation. This step is performed by SN_i. SN_i randomly selects secret value $x_i \in Z^*_q$ and saves it, then computes $X_i = x_iP$. The (ID_i, X_i) is sent to CPS. Where x_i is the private key and X_i is the public key.

3) Part private key generation. This step is performed by the CPS. CPS randomly selects the secret value $r_i \in Z^*_q$ and calculates $R_i = r_iP$, $h_{i1} = H_1(ID_i, R_i, X_i)$, $D_i = r_i + sh_{i1}$. And it sends (R_i, D_i) to each sensing node SN_i through the security channel. Where R_i is the user's partial public key and D_i is the user's partial private key. So, the private key of SN_i is $SK_i = (D_i, x_i)$, and the public key is $PK_i = (R_i, X_i)$. Similarly, the private key of the application terminal AT is $SK_B = (D_B, x_B)$, and the public key is $PK_B = (R_B, X_B)$.

4) Individual signcryption. The algorithm is implemented by SN_i. The steps for encrypting the message m_i sent by SN_i to AT are as follows.

   a. SN_i randomly selects $k_i, t_i \in Z^*_q$.
   b. Computing $K_i = k_iP$, $T_i = t_iP$.
   c. Computing $Q_{i1} = k_iX_B$, $Q_{i2} = t_i(R_B + P_{pub}H_1(ID_B, R_B, X_B))$.
   d. Computing $h_{i2} = H_2(Q_{i1}, Q_{i2})$.
   e. Encrypting $C_i = h_{i2} \oplus (m_i||ID_i)$.
   f. Computing $h_{i3} = H_3(C_i, Q_{i1}, Q_{i2}, K_i)$, $h_{i4} = H_4(C_i, Q_{i1}, Q_{i2}, T_i)$.
   g. Signature. $S_i = k_i + t_i + h_{i3}D_i + h_{i4}x_i$.

   The signcryption of the key m_i sent by SN_i to AT is $\sigma_i = (C_i, K_i, T_i, S_i)$.

5) Aggregation signcryption. The algorithm is executed by the gateway node CN. It receives signcryptioner' information $\sigma_i = (C_i, K_i, T_i, S_i)$, the aggregator CN calculates $S = \sum_{i=1}^n S_i$. Then the aggregation signcryption is $\sigma = (\{K_i, T_i, C_i\}^n_{i=1}, S')$, and it is sent to AT.

6) De-signcrypt. This step is performed by the application terminal AT. The steps to decrypt the signcryption $\sigma_i = (C_i, K_i, T_i, S_i)$ sent by AT to SN_i are as follows:

   a. Computing $Q_{i1} = k_i x_B$, $Q_{i2} = T_i(r_B + sH_1(ID_B, R_B, X_B)) = T_i D_B$.

   b. Computing $h_{i2} = H_2(Q_{i1}, Q_{i2})$.

   c. Decrypting $(m_i||ID_i) = h_{i2} \oplus C_i$.

   d. Computing $h_{i3} = H_3(C_i, Q_{i1}, Q_{i2}, K_i)$, $h_{i4} = H_4(C_i, Q_{i1}, Q_{i2}, T_i)$.

   It verifies that whether the signature is correct $S_i P = K_i + T_i + h_{i3}(R_i + P_{pub}H_1(D_i, R_i, X_i)) + h_{i4}X_i$. If it is correct, it proves that the aggregation signcryption is valid, and outputs $(m_i||ID_i)$. Otherwise, output false.

7) Aggregation de-signcrypt. This step is performed by the application side AT. The decryption steps of signcryption $\sigma = (\{K_i, T_i, C_i\}_{i=1}^n, S')$ sent by AT to SN_i are as follows:

   a. Computing $Q_{i1} = k_i x_B$, $Q_{i2} = T_i(r_B + sH_1(ID_B, R_B, X_B)) = T_i D_B$.

   b. Computing $h_{i2} = H_2(Q_{i1}, Q_{i2})$.

   c. Decrypting $(m_i||ID_i) = h_{i2} \oplus C_i$.

   d. Computing $h_{i3} = H_3(C_i, Q_{i1}, Q_{i2}, K_i)$, $h_{i4} = H_4(C_i, Q_{i1}, Q_{i2}, T_i)$.

   It verifies that whether the signature is correct $SP = \sum_{i=1}^n K_i + \sum_{i=1}^n T_i + \sum_{i=1}^n h_{i3}(R_i + P_{pub}H_1(D_i, R_i, X_i)) + \sum_{i=1}^n h_{i4}X_i$. If it is correct, it proves that the aggregation signcryption is valid, and outputs $(m_i||ID_i)$. Otherwise, output false.

5 Analysis of Proposed Scheme

5.1 Correctness of Proposed Scheme

Theorem 1. *The receiver can verify the correctness of the signcryption and aggregation signature, and can get the correct decrypted plaintext m_1.*

Proof.

1) AT can verify the correctness of signcryption $\sigma_i = (C_i, K_i, T_i, S_i)$.

$$\begin{aligned} S_i P &= [k_i + t_i + h_{i3}D_i + h_{i4}x_i]P \\ &= [k_i + t_i + h_{i3}D_i + h_{i4}x_i]P \\ &= K_i + T_i + h_{i3}(R_i + P_{pub}H_1(ID_i, R_i, X_i)) \\ &\quad + h_{i4}X_i \end{aligned}$$

2) AT can verify the correctness of aggregation signcryption $\sigma = (\{K_i, T_i, C_i\}_{i=1}^n, S)$.

$$\begin{aligned} SP &= \sum_{i=1}^n [k_i + t_i + h_{i3}D_i + h_{i4}x_i]P \\ &= \sum_{i=1}^n [K_i + T_i + h_{i3}D_i + h_{i4}(R_i + P_{pub}H_1(ID_i, R_i, X_i))] \\ &= \sum_{i=1}^n K_i + \sum_{i=1}^n T_i + \sum_{i=1}^n [h_{i3}(R_i + P_{pub}H_1(ID_i, R_i, X_i))] \\ &\quad + \sum_{i=1}^n h_{i4}D_i \end{aligned}$$

3) AT can obtain the correct decrypted plaintext m_i.

$$\begin{aligned} h_{i2} &= H_2(Q_{i1}, Q_{i2}) \\ &= H_2(k_i X_B, t_i(X_B + R_B + P_{pub}H_1(ID_B, R_B, X_B))) \\ &= H_2(k_i x_B P, t_i P(x_B + r_B + sH_1(ID_B, R_B, X_B))) \\ &= H_2(k_i X_B, T_i(x_B + r_B + sH_1(ID_B, R_B, X_B))) \\ &= H_2(K_i x_B, T_i D_B) \\ &= h'_{i2} \end{aligned}$$

□

Since SN_i encrypts the plaintext by calculating $C_i = h_{i2} \oplus (m_i||ID_i)$. AT decrypts ciphertext by calculating $m_i||ID_i = h'_{i2} \oplus C_i$, and $h_{i2} = h'_{i2}$, CPS can finally get the correct plaintext.

5.2 Unforgeability of Proposed Scheme

Theorem 2. *In the case of random prediction model and DLP situation, the proposed CLASC scheme in this paper is unforgeability under adaptive selective message attack.*

Lemma 1. *Under the random prediction model, if there is a probability polynomial time attacker A_I wins the game with a non-negligible probability, then there is algorithm C_1 that can solve the DLP (where A_I can execute at most q_{H_i} $(i = 1, 2, 3, 4)$ times of H_i query, q_{SK} times of private key query, q_{PSK} times of partial private key query, q_{PK} times of public key query and q_{SC} times of signcryption query. The user number of aggregation signcryption is n).*

Proof. Supposing algorithm C_1 is a DLP solver with input tuple (P, bP), where $b \in Z_q^*$ is unknown. The goal is to compute b with A_I as the challenger of the subroutine. C_1 maintains the following six lists L_1, L_2, L_3, L_4, L_{ID} and L_{SC} to record query data for predictor H_1, H_2, H_3, H_4, user creation and signcryption, respectively. The list is initialized with empty.

- System initialization stage. C_1 sets $P_{pub} = bP$ (here b is the default system key and secret to A_I, selects and sends the system parameter $params = \{q, P, G, P_{pub}, H_1, H_2, H_3\}$ to the adversary A_I.
- Query phase.

1) H_1 query. C_1 maintains list $L_1 = \{ID_i, R_i, X_i, h_{i1}\}$. When A_I inputs (ID_i, R_i, X_i), C_1 responds to this challenge in the following ways. If the query for this (ID_i, R_i, X_i) already exists in the list L_1, then it returns the corresponding h_{i1} to A_I. Otherwise, C_1 randomly selects $h_{i1} \in Z_q^*$, adds $\{ID_i, R_i, X_i, h_{i1}\}$ to listing L_1 and returns to A_I.
2) H_2 query. C_1 maintains list $L_2 = \{Q_{i1}, Q_{i2}, h_{i2}\}$. When A_I inputs (Q_{i1}, Q_{i2}), C_1 responds to this challenge in the following ways. If the query for (Q_{i1}, Q_{i2}) already exists in the list L_2, then it returns the corresponding h_{i2} to A_I. Otherwise, C_1 randomly selects $h_{i2} \in Z_q^*$, adds $\{Q_{i1}, Q_{i2}, h_{i2}\}$ to listing L_2 and returns h_{i2} to A_I.
3) H_3 query. C_1 maintains list $L_3 = \{C_i, Q_{i1}, Q_{i2}, K_i, h_{i3}\}$. When A_I inputs $\{C_i, Q_{i1}, Q_{i2}, K_i\}$, C_1 responds to this challenge in the following ways. If the query for $\{C_i, Q_{i1}, Q_{i2}, K_i\}$ already exists in the list L_3, then it returns the corresponding h_{i3} to A_I. Otherwise, C_1 randomly selects $h_{i3} \in Z_q^*$, adds $\{C_i, Q_{i1}, Q_{i2}, K_i, h_{i3}\}$ to listing L_3 and returns h_{i3} to A_I.
4) H_4 query. C_1 maintains list $L_4 = \{C_i, Q_{i1}, Q_{i2}, T_i, h_{i4}\}$. When A_I inputs $\{C_i, Q_{i1}, Q_{i2}, T_i\}$, C_1 responds to this challenge in the following ways. If the query for $\{C_i, Q_{i1}, Q_{i2}, T_i\}$ already exists in the list L_4, then it returns the corresponding h_{i4} to A_I. Otherwise, C_1 randomly selects $h_{i4} \in Z_q^*$, adds $\{C_i, Q_{i1}, Q_{i2}, T_i, h_{i4}\}$ to listing L_4 and returns h_{i4} to A_I.
5) User creation query. C_1 maintains initialization list $L_{ID_i} = \{ID_i, h_{i1}, D_i, r_i, R_i, x_i, X_i\}$. It submits user ID_i, if $\{ID_i, h_{i1}, D_i, r_i, R_i, x_i, X_i\}$ already exists in L_{ID_i}, then it will be ignored. Otherwise, C_1 executes the H_1 query and obtains the h_{i1}. If $ID_i = ID_j$, C_1 randomly selects $r_j, x_j \in Z_q^*$, calculates $R_j = r_j P$ and $X_j = x_j P$, inserts $\{ID_j, h_{j1}, \perp, r_j, R_j, x_j, X_j\}$ into L_{ID}. Otherwise, C_1 randomly selects $D_i, x_i \in Z_q^*$, computes $R_i = D_i P - h_{i1} P_{pub}$ and $X_i = x_i P$, inserts $\{ID_j, h_{j1}, \perp, r_j, R_j, x_j, X_j\}$ into L_{ID}.
6) Partial private key query. A_I submits the user ID_i. C_1 makes the following response: if $ID_i = ID_j$, C terminates the game; Otherwise, C_1 returns D_i to A_I.
7) Private key query. A_I submits user identity ID_i. C_1 returns the corresponding x_i to A_I.
8) Public key query. A_I submits ID_i, C_1 returns public key (R_i, X_i) corresponding to ID_i as response.
9) Public key substitution query. A_I adopts a new public key (X_i', R_i') to replace the original public key (X_i, R_i) of the signcryption ID_i.
10) Signcryption query. C_1 maintains initialization list $L_{SC} = \{m_i, ID_i, ID_B, K_i, T_i, h_{i2}, h_{i3}, h_{i4}, S_i, c_i\}$. A_I submits un-signcryption information m_i, sender identity ID_i and receiver identity ID_B. If $ID_i = ID_j$, C_1 randomly selects $S_i, h_{i3}, h_{i4}, k_i \in Z_q^*$, calculates $K_i = k_i P$, $T_i = S_i P - h_{i3} x_j - h_{i4} D_j - K_i$ and $h_{i2} = H_2(K_i x_B, T_i(x_B + r_B + sH_1(ID_B, R_B, X_B)))$. It queries list L_3 and L_4, if L_3 exists in $(C_i, Q_{i1}, Q_{i2}, K_i, h_{i3}')$ or L_4 exists in $(C_i, Q_{i1}, Q_{i2}, K_i, h_{i4}')$, and $h_{i3} \neq h_{i3}' \vee h_{i4} \neq h_{i4}'$, C_1 re-selects $(S_i, h_{i3}, h_{i4}, k_i)$. Otherwise, C_1 computes $C_i = h_{i2} \oplus (m_i || ID_i)$ and returns ciphertext $\sigma_i = (C_i, K_i, T_i, S_i)$. If $ID_i \neq ID_j$, C_1 is calculated according to the signcryption algorithm. H_i query and key query are performed as required, and then the signcryption message $\sigma_i = (C_i, K_i, T_i, S_i)$ is returned. Finally, C_1 inserts $\{m_i, ID_i, ID_B, K_i, T_i, h_{i2}, h_{i3}, h_{i4}, S_i, c_i\}$, $(C_i, Q_{i1}, Q_{i2}, K_i, h_{i3})$ and $(C_i, Q_{i1}, Q_{i2}, T_i, h_{i4})$ into the L_{SC}, L_3 and L_4, respectively.

- Forgery phase. After the query phase, A_I submits the challenge user identity (ID_j, ID_B), the challenge message m_j and its signcryption ciphertext (C_j, K_j, T_j, S_j). C_1 calculates the $h_{i2} = H_2(K_i x_B, T_i(x_B + r_B + sH_1(ID_B, R_B, X_B)))$ to decrypt the message $m_j = h_{i2} \oplus C_j$. According to the forking lemma [5], C_1 uses predictor replay attack technique that can obtain two legal signatures $(m_j, ID_j, ID_B, K_j, T_j, h_{j3}, h_{j4}, S_j)$ and $(m_j, ID_j, ID_B, K_j, T_j, h_{j3}', h_{j4}, S_j')$, where $S_i \neq S_j'$, $h_{j3} \neq h_{j3}'$ and it satisfies:

$$\begin{aligned} S_j &= k_j + t_j + h_{j3} D_j + h_{j4} x_i. \\ S_j' &= k_j + t_j + h_{j3}' D_j + h_{j4} x_i. \end{aligned}$$

Therefore, C_1 calculates:

$$\begin{aligned} S_j' - S_j &= (h_{j3}' - h_{j3}) D_j. \\ b &= \frac{S_j' - S_j - (h_{j3}' - h_{j3}) r_j}{(h_{j3}' - h_{j3}) H_1(ID_j, R_j, X_j)}. \end{aligned}$$

The results are as the response to DLP. Therefore, C_1 successfully obtains an example of DLP problem. The advantage of successfully solving DLP problems is:

$$\varepsilon' = \varepsilon \frac{1}{q_{PSK} + n} (1 - \frac{1}{q_{PSK} + n})^{q_{PSK} + n - 1}.$$

So Theorem 2 and Lemma 1 are correct. □

Lemma 2. *Under the random prediction model, if there is a probability polynomial time A_{II} attacker wins the game with a non-negligible probability, then there is algorithm C_2 that can solve the DLP (where A_{II} can execute at most q_{H_i} $(i = 1, 2, 3, 4)$ times of H_i query, q_{SK} times of private key query, q_{PSK} times of partial private*

key query, q_{PK} times of public key query and q_{SC} times of signcryption query. The user number of aggregation signcryption is n).

Proof. Supposing algorithm C_2 is a DLP solver with input tuple (P, bP), where $b \in Z_q^*$ is unknown. The goal is to compute b with A_I as the challenger of the subroutine. C_1 maintains the following six lists L_1, L_2, L_3, L_4, L_{ID} and L_{SC} to record query data for predictor H_1, H_2, H_3, H_4, user creation and signcryption, respectively. The list is initialized with empty.

- System initialization stage. Supposing $P_{pub} = sP$, $s \in Z_q^*$. The system parameter $params = \{q, P, G, P_{pub}, H_1, H_2, H_3\}$, C_2 sends (q, P, G, P_{pub}, s) to A_{II}.
- Query phase. A_{II} performs the following polynomial bounded query.
  1) H_1, H_2, H_3, H_4 queries are same as Theorem 1.
  2) User creation query. C_2 maintains initialization list $L_{ID_i} = \{ID_i, h_{i1}, D_i, r_i, R_i, x_i, X_i\}$. It submits user ID_i, if $\{ID_i, h_{i1}, D_i, r_i, R_i, x_i, X_i\}$ already exists in L_{ID_i}, then it will be ignored. Otherwise, C_2 executes the H_1 query and obtains the h_{i1}. If $ID_i = ID_j$, let $X_j = bP$, it calculates $R_j = r_jP$ and $D_j = r_j + sH_1(ID_j, R_j, X_j)$, inserts $\{ID_j, h_{j1}, \perp, r_j, R_j, x_j, X_j\}$ into L_{ID}. Otherwise, C_2 randomly selects $D_i, x_i \in Z_q^*$, computes $R_i = r_iP$ and $X_i = x_iP$, inserts $\{ID_j, h_{j1}, \perp, r_j, R_j, x_j, X_j\}$ into L_{ID_i}.
  3) Partial private key query. A_{II} submits the user ID_i. C_2 makes the following response: if $ID_i = ID_j$, C_2 terminates the game; Otherwise, C_2 returns corresponding D_i to A_{II}.
  4) Private key query. A_{II} submits user identity ID_i. C_2 makes the following response: if $ID_i = ID_j$, C_2 terminates the game; Otherwise, C_1 returns x_i to A_I.
  5) Public key query. A_{II} submits ID_i, C_1 returns public key (R_i, X_i) corresponding to ID_i as response.
  6) Public key substitution query. A_{II} submits ID_i and X_i', if $ID_i = ID_j$, C_2 terminates the game; Otherwise, A_{II} adopts X_i' to replace the original public key X_i of the signcryption ID_i.
  7) Signcryption query. C_2 maintains initialization list $L_{SC} = \{m_i, ID_i, ID_B, K_i, T_i, h_{i2}, h_{i3}, h_{i4}, S_i, c_i\}$. A_{II} submits unsigncryption information m_i, sender identity ID_i and receiver identity ID_B. If $ID_i = ID_j$, C_1 randomly selects $S_i, h_{i3}, t_i \in Z_q^*$, calculates $T_i = t_iP$, $K_i = S_iP - h_{i3}(x_i + D_j) - T_i$ and $h_{i2} = H_2(K_ix_B, T_i(x_B + r_B + sH_1(ID_B, R_B, X_B)))$. It queries list L_3 and L_4, if L_3 exists in $(C_i, Q_{i1}, Q_{i2}, K_i, h_{i3}')$ or L_4 exists in $(C_i, Q_{i1}, Q_{i2}, K_i, h_{i4}')$, and $h_{i3} \neq h_{i3}' \vee h_{i4} \neq h_{i4}'$, C_2 re-selects $(S_i, h_{i3}, h_{i4}, t_i)$. Otherwise, C_2 computes $C_i = h_{i2} \oplus (m_i || ID_i)$ and returns ciphertext $\sigma_i = (C_i, K_i, T_i, S_i)$. If $ID_i \neq ID_j$, C_1 is calculated according to the signcryption algorithm. H_i query and key query are performed as required, and then the signcryption message $\sigma_i = (C_i, K_i, T_i, S_i)$ is returned. Finally, C_2 inserts $\{m_i, ID_i, ID_B, K_i, T_i, h_{i2}, h_{i3}, h_{i4}, S_i, c_i\}$, $(C_i, Q_{i1}, Q_{i2}, K_i, h_{i3})$ and $(C_i, Q_{i1}, Q_{i2}, T_i, h_{i4})$ into the L_{SC}, L_3 and L_4, respectively.
- Forgery phase. After the query phase, A_{II} submits the challenge user identity (ID_j, ID_B), the challenge message m_j and its signcryption ciphertext (C_j, K_j, T_j, S_j). C_2 calculates the $h_{i2} = H_2(K_ix_B, T_i(x_B + r_B + sH_1(ID_B, R_B, X_B)))$ to decrypt the message $m_j = h_{i2} \oplus C_j$. According to the forking lemma [5], C_2 uses predictor replay attack technique that can obtain two legal signatures $(m_j, ID_j, ID_B, K_j, T_j, h_{j3}, h_{j4}, S_j)$ and $(m_j, ID_j, ID_B, K_j, T_j, h_{j3}', h_{j4}, S_j')$, where $S_i \neq S_j'$, $h_{j3} \neq h_{j3}'$ and it satisfies:

$$\begin{aligned} S_j &= k_j + t_j + h_{j3}D_j + h_{j4}x_i \\ S_j' &= k_j + t_j + h_{j3}D_j + h_{j4}'x_i \end{aligned}$$

Therefore, C_2 calculates:

$$\begin{aligned} S_j' - S_j &= (h_{j4}' - h_{j4})x_j \\ b &= \frac{S_j' - S_j}{(h_{j4}' - h_{j4})} \end{aligned}$$

The results are as the response to DLP. Therefore, C_2 successfully obtains an example of DLP problem. The advantage of successfully solving DLP problems is:

$$\varepsilon' = \varepsilon \frac{1}{q_{PSK} + n}(1 - \frac{1}{q_{PSK} + n})^{q_{PSK}+n-1}$$

So Lemma 2 is correct. □

5.3 Confidentiality of Proposed Scheme

Theorem 3. *Under the random prediction model, based on CDHP, the proposed CLASC scheme in this paper is indistinct under the adaptive selective ciphertext attack, that is, IND-CLASC-CCA2 is security.*

Lemma 3. *Under the random prediction model, if there is a probability polynomial time adversary A_I (A_{II}) wins the game with non-negligible probability, then there is an instance of CDPH where the challenger can solve with non-negligible probability.*

The proof method of Lemma 3 is similar to the confidentiality proof in document [5]. Due to the limited space, we will not give the process.

Table 1: Comparison of computation and security performance of aggregation signcryption

Scheme	PF-CLRSC	PAS	ESAS	ASS	Proposed
signcryption	np+ne	np+2ns	ne+4ns	3ne+np+ns	(2n+1)s
De-signcrypt	(2n+3)p+(n+1)s	3p+np	(2+n)p+ns	np+ns	(5n+1)s
Total operation	ne+(3n+3)p+(n+1)s	(2n+3)p+2ns	ne+5ns+(n+2)p	2np+3ne+2ns	(7n+2)s
Cost consumption	72.06n+60.85	41.68n+60.03	35.36n+40.02	75.28n	5.81n+1.66
Security	Provable Security	Provable Security	Provable Security	Provable Security	Provable Security
Public verifiability	YES	NO	NO	NO	YES

5.4 Public Verifiability of Proposed Scheme

In this scheme, any third party can verify the following equation when there is a dispute between the signcryption sender and the signcryption receiver about the authenticity of the aggregation signcryption text.

$$\begin{aligned} SP &= \sum_{i=1}^{n} K_i + \sum_{i=1}^{n} T_i + \sum_{i=1}^{n} h_{i3} W + \sum_{i=1}^{n} h_{i4} X_i \\ W &= R_i + P_{pub} H_1(ID_i, R_i, X_i). \end{aligned}$$

Because the verification of this equation does not require the participation of the receiver, and does not require any secret information of the signcryptioner, so the scheme is publicly verifiable.

5.5 Performance Analysis and Discussion

In order to compare the computational efficiency of the proposed scheme, it is assumed that there are n users participating in the scheme. In here, three operations are considered: the exponential operation (e), the multiplication operation on group G(s), and the bilinear pair operation (p). Compared with the three operations, the effect of hashing and XOR operation on the overall efficiency is negligible.

In the proposed scheme, in the signcryption phase, n signcryptioners calculate $Q_{i1} = k_i X_B$, $Q_{i2} = t_i(R_B + P_{pub} H_1(ID_B, R_B, X_B))$ that requires $2n+1$ point multiplication operations. The value of $P_{pub} H_1(ID_B, R_B, X_B)$ is fixed, it only needs to be calculated once. In the de-signcrypt phase, computing SP, $Q_{i1} = K_i x_B$, $Q_{i2} = T_i D_B$ needs $5n+1$ point multiplication operations.

As can be seen from Table 1, when the same number of messages are executed with aggregation signcryption, the operation efficiency of this scheme is greatly improved compared with the schemes in references [3, 10, 11, 14]. Compared with the scheme with relatively high operation efficiency, the operation efficiency is improved by nearly 6 times. From the perspective of the security performance of the scheme, only reference [14] and proposed scheme satisfy the public verifiability. Considering the operation efficiency and security of the scheme, this scheme is better than the above four schemes.

The following is an example of two-column Table 1.

6 Conclusion

Aggregation signcryption has many features such as encryption, signature and batch processing, it is of great application value in the cloud computing environment. In order to improve the computational efficiency of certificateless aggregation signcryption, an non-bilinear pairless aggregation signcryption scheme is proposed based on the random prediction model. Compared with the existing schemes, this scheme has a faster computing speed and is more suitable for application in the Internet of Things. In the future, we will research deep learning methods to improve the certificateless aggregation signcryption.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. S. Al-Riyami, K. G. Paterson, "Certificateless public key cryptography," *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 452-473, 2003.

[2] J. Baek, R. Steinfeld, Y. Zheng, "Formal proofs for the security of signcryption," *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems*, vol. 20, pp. 203-235, 2007.

[3] S. Chandrasekhar, M. Singhal, "Efficient and scalable aggregate signcryption scheme based on multitrapdoor hash functions," in *IEEE Conference on Communications and Network Security (CNS'15)*, 2015. DOI: 10.1109/CNS.2015.7346875.

[4] L. Cheng, Q. Wen, Z. Jin, *et al.*, "Cryptanalysis and improvement of a certificateless aggregate signature scheme," *Information Sciences*, vol. 295, pp. 337-346, 2015.

[5] Z. Eslami, N. Pakniat, "Certificateless aggregate signcryption: Security model and a concrete construction secure in the random oracle model," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 3, pp. 276-286, 2014.

[6] C. Lan, H. Li, S. Yin, *et al.*, "A new security cloud storage data encryption scheme based on identity proxy re-encryption," *International Journal of Network Security*, vol. 19, no. 5, pp. 804-810, 2017.

[7] P. Li, Z. Chen, L. T. Yang, *et al.*, "An incremental deep convolutional computation model for feature learning on industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1341-1349, 2019.

[8] H. Li, S. L. Yin, C. Zhao and L. Teng, "A proxy re-encryption scheme based on elliptic curve group," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 218-227, Jan. 2017.

[9] J. Liu, S. L. Yin, H. Li and L. Teng, "A density-based clustering method for k-anonymity privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 12-18, Jan. 2017.

[10] J. Liu, C. Zhao, K. Mao, "Efficient certificateless aggregate signcryption scheme based on XOR," *Computer Engineering and Applications*, vol. 52, no. 12, pp. 131-135, 2016.

[11] S. Niu, L. Niu, C. Wang, *et al.*, "A provable aggregate signcryption for heterogeneous systems," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 39, no. 5, pp. 1213-1218, 2017.

[12] L. Peng, Z. Chen, L. T. Yang, *et al.*, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790-798, 2018.

[13] S. S. D. Selvi, S. S. Vivek, J. Shriram, *et al.*, "Identity based aggregate signcryption schemes," in *Progress in Cryptology*, pp. 378-397, 2009.

[14] G. Sharma, S. Bala, A. K. Verma, "Pairing-free certificateless ring signcryption (PF-CLRSC) scheme for wireless sensor networks," *Wireless Personal Communications*, vol. 84, no. 2, pp. 1469-1485, 2015.

[15] L. Teng, H. Li, "A high-efficiency discrete logarithm-based multi-proxy blind signature scheme," *International Journal of Network Security*, vol. 20, no. 6, pp. 1200-1205, Nov. 1, 2018.

[16] L. Teng, H. Li and S. Yin, "IM-Mobishare: An improved privacy preserving scheme based on asymmetric encryption and bloom filter for users location sharing in social network," *Journal of Computers (Taiwan)*, vol. 30, no. 3, pp. 59-71, 2019.

[17] S. Yin, J. Liu and L. Teng, "Improved elliptic curve cryptography with homomorphic encryption for medical image encryption," *International Journal of Network Security*, vol. 22, no. 3, pp. 421-426, 2020.

[18] Q. Zhang, C. Bai, L. T. Yang, Z. Chen, P. Li, and H. Yu, "A unified smart Chinese medicine framework for healthcare and medical services," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019. DOI: 10.1109/TCBB.2019.2914447.

[19] J. Zhang, J. Mao, "On the security of a pairing-free certificateless signcryption scheme," *The Computer Journal*, vol. 61, no. 4, pp. 469-471, 2018.

[20] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption)," *Lecture Notes in Computer Science*, vol. 1294, pp. 165-179, 1997.

[21] L. Zou, X. Wang, S. Yin, "A data sorting and searching scheme based on distributed asymmetric searchable encryption," *International Journal of Network Security*, vol. 20, no. 3, pp. 502-508, 2018.

Biography

Mingju Zhao is a lecturer in the School of Electrical Engineering & Zhengzhou University of Science and Technology. His research interests focus on computer and network security.

Yuping Peng is a lecturer in the School of Electrical Engineering & Zhengzhou University of Science and Technology. His research interests focus on computer and network security.