IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 2, 2019, accepted September 26, 2019, date of publication October 7, 2019, date of current version October 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945813

Designated Server Certificateless Deniably Authenticated Encryption With Keyword Search

YULEI ZHANG[1], LONG WEN[1], YONGJIE ZHANG[2], AND CAIFEN WANG[3]
[1]College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
[2]Gansu Health Vocational College, Lanzhou 730000, China
[3]College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518000, China

Corresponding author: Yulei Zhang (zhangyl@nwnu.eud.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662069, and in part by the Higher Educational Scientific Research Foundation of Gansu Province under Grant 2017A-003 and Grant 2018A-207.

ABSTRACT In email system, the cryptography technology has been used to defend email secrets, so it is important to search specific encrypted emails on cloud sever without local decryption. The Public key encryption with keyword search (PEKS) might is a suitable way to perform the email ciphertext search. However, most existing PEKS schemes cannot protect the identity privacy of data sender. Deniably authenticated encryption (DAE) technique allows the data sender to deny his/her involvement after the communication. Moreover, the receiver can verify the authenticity of ciphertext in DAE, which assures the identity privacy of data sender. In this paper, so as to solve the above shortages in existing PEKS schemes, we introduce an original scheme called designated server certificateless deniably authenticated encryption with keyword search (dCLDAEKS), where leverages the techniques of DAE and designated server. In dCLDAEKS, data sender authenticates the messages and simultaneously encrypt them. Meanwhile, only designated server has ability to execute search ciphertext operation for receivers. So there is no adversary including the server can launch inside or outside offline KGA. Therefore dCLDAEKS scheme can better protect the identity privacy of data sender. In addition, compared the related schemes in the literature, dCLDAEKS scheme perform less efficient in some procedure, but it can against inside KGA and better protect the sender's identity privacy.

INDEX TERMS Certificateless, designated tester, deniably authenticated, identity privacy, searchable encryption.

I. INTRODUCTION

With the prosperity development of cloud server, users can store their data in the cloud server and take advantage of its powerful computing ability to execute complicate computation [1]. However, cloud service provider (CSP) might try to read user's emails to discover some privacy information. Therefore, these sensitivity emails should be encrypted before sending to cloud sever to ensure user's privacy. Generally, user may have lots of emails. When user wants to obtain specific email, they should download all of encrypted emails and decrypt all of them. Obviously, this operation is too inefficient.

To solve this problem, Song *et al.* [2] introduced the symmetric searchable encryption (SSE). In SSE scheme, receiver must negotiate with sender, which cause it unsuitable for some specific scenarios (e g. data-sharing). Then Public key encryption with keyword search (PEKS) was proposed [3], which different from SSE schemes [4], [5] is that data sender can share ciphertext with receiver. Moreover, receiver can give others a trapdoor which contain keyword to authorize them execute search ciphertext operation. However, PEKS exists the risk of exposing the search pattern which means adversary might learn some sensitivity emails from the searching frequency in trapdoor searching history. To solve this issue, Baek *et al.* [6] introduced designated tester to search encrypted data, in which only the designated tester can execute search ciphertexts operation. Later, many researchers proposed different schemes [7], [8] with designated tester to hide search pattern.

However, Byun *et al.* [9] pointed that almost PEKS schemes are vulnerable suffer from the offline keyword guessing attacks (KGA), Which means that adversary can try each possible keyword and encrypt it, then adversary can

The associate editor coordinating the review of this manuscript and approving it for publication was Hongwei Du.

discover specific keyword contained in the trapdoor by testing encrypted keyword with trapdoor. Due to the chosen space of keyword is small in real world, such attack is feasible. Hence, cloud sever might turn into the inside KGA adversary who can recover sensitive message from some emails.

Lately, Huang and Li [10] proposed public key authenticated encryption with keyword search (PAEKS), where sender authenticates keyword while encrypting it to counter inside KGA attacks.

However, there are three drawbacks in scheme [10]. Firstly, if the outside adversary breaks into cloud server, she/he might acquire the search pattern of user. Secondly, scheme [10] is public key infrastructure (PKI) based, which exists complicated certificate management problem. Some improved identity based solutions [11] also suffer from key escrow problem. Thirdly, almost PAEKS [10], [11] and its derivative scheme [12] cannot guarantee the identity privacy of the sender.

A. RELATED WORK

In IOT medical, doctors often need some specific data of patients to support their research. They usually need to collect the change of patient's health condition. Hence, the researchers need patient to report their health information truthfully by sending emails. These emails will be stored in the cloud sever. When researchers want to find out some specific symptom, they can search over the related emails and downloads them. Generally, researchers are required to determine the source of the email and verify its validity. However, patients want to protect their privacy, they expect that the research institution can explain to others the authenticity and reliability of the data, but researchers cannot let the third party knows the source of these health information.

To settle these issues, Song *et al.* [2] first introduced a SSE scheme. But the model of SSE scheme cannot satisfy the need of flexible data sharing. Then Boneh *et al.* [3] introduced PEKS scheme for the setting of multi user. However, the trapdoor in PEKS must be transmitted via security channel. Then Baek *et al.* [6] constructed designated tester PEKS scheme (dPEKS), in which only the designated tester has ability to execute the search operation. And Byun *et al.* [9] found that PEKS schemes exist risk of offline KGA. Since then, researcher [7], [8] enhanced the ability of resisting offline KGA in scheme [6]. But in schemes [7], [8], inside adversary still can launch offline KGA and it also suffer from complicated key escrow problem. To address this problem, Peng *et al.* [13] first proposed certificateless public key encryption with keyword search (CLPEKS), but it was found existed the risk of offline KGA attacks [14]. Ma *et al.* [15], [16] introduced two different CLPEKS tried to counter KGA, but these schemes still existed the risk of inside offline KGA. Lately, Huang and Li [10] pointed that sender authenticated the keyword while encrypting email can resist the inside offline KGA. Then Li *et al.* [11] proposed designated sever identity based authenticates encryption with keyword search, it reduce the cost of public key certification in his prior work. However, all of the above schemes cannot protect the identity privacy of data sender.

To achieve the goal of protecting sender's identity privacy, researcher found that authentication encryption (AE) combine deniability might is suitable way. Authenticated encryption (AE) can be divided: symmetric authenticated encryption [18] and public key authenticated encryption [17]. In symmetric AE scheme, it's easy to make receiver parties produce the same probability distribution ciphertexts to achieve deniability. However, how to achieve deniability in public key AE scheme is a problem.

To solve this issue, researchers proposed some public key AE scheme [19], [20]. Moreover, Li *et al.* [21] first proposed deniably authenticated encryption scheme (DAE) for email system. Later, to reduce the heavy cost of certificate management, Wu and Li [22] proposed identity based deniability authenticated encryption scheme. In 2018, Emmanuel *et al.* [23] improved the scheme in literature [22] and proposed certificateless deniably authenticated encryption (CLDAE) to avoid the key escrow problem. However, it cannot execute the keyword search operation nor resist offline KGA form inside adversaries.

B. MOTIVATION AND CONTRIBUTIONS

We can get from the above, there is no such certificateless PEKS can hide search pattern from outside adversaries and counter inside offline KGA, meanwhile, protecting the identity privacy of data sender. In this paper, we construct a designated server certificateless deniably authenticated encryption with keyword search (dCLDAEKS) scheme. Our contribution is mainly in the following four points:

1) We present the security models for dCLDAEKS, and further we prove that dCLDAEKS not only against inside offline KGA but also assure the indistinguishability of ciphertext and indistinguishability of trapdoor.
2) In dCLDAEKS, only the designated server can execute the search operation which means no adversary can lunch KGA even if it acquires the user's trapdoor. Unless adversary gets the secret key of server.
3) We combine deniably authenticated encryption with PEKS, which make the scheme realize deniability to protect the identity privacy of sender, and it also achieve keyword search function.
4) We compare our scheme with some related schemes in security and computational complexity. Then we demonstrate our scheme‘s computation efficiency by simulating in JPBC. Although dCLDAEKS is slightly less efficient in some sections performance but it can against inside KGA and better protect the sender's identity privacy.

II. DESIGNATED SERVER CERTIFICATELESS DENIABLY AUTHENTICATED ENCRYPTION WITH KEYWORD SEARCH

In this section, we show the system model, definition and security model of dCLDAEKS scheme.

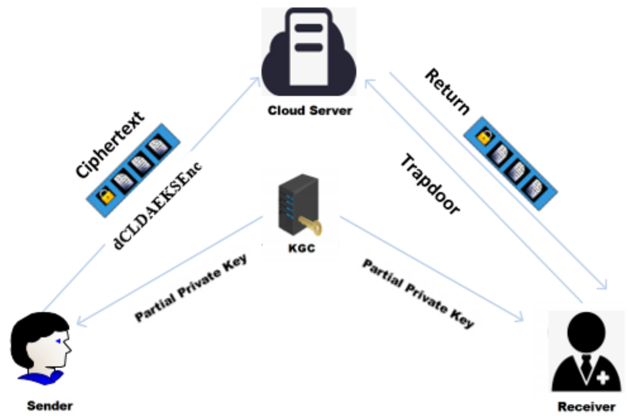

FIGURE 1. System model.

A. PRELIMINARY KNOWLEDGE

1) BILINEAR PAIRING

A map $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear if it satisfies the following factors:

(1) Bilinear: $e(kP, zP) = e(P, P)^{kz}$ where $k, z \in Z_q^*$.
(2) Nondegenerate: If $P \in G_1$, $e(P, P) \neq 1$, where 1 is identity element of G_2.
(3) Computable: $e(P, P)$ is efficiently computable for any $P \in G_1$.

2) DECISIONAL BILINEAR DIFFIE–HELLMAN PROBLEM (DBDH)

Given G_1 and G_2 as groups with order q, P is a generator of G_1 and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, the DBDH problem is to decide whether $Y = e(P, P)^{kzc}$ or not with (P, kP, zP, cP) and $Y \in G_2$ where $k, z, c \in Z_q^*$. Let β be a bit such that $\beta = 0$ if $Y = e(P, P)^{kzc}$, and $\beta = 1$ if Y is randomly selected from G_2.

3) COMPUTATIONAL DIFFIE–HELLMAN PROBLEM (CDH)

Given G_1 and G_2 as groups with order q, P is a generator of G_1, and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, the CDH problem is to compute kzP with the tuple (P, kP, zP) where $k, z \in Z_q^*$.

B. SYSTEM MODEL OF dCLDAEKS SCHEME

As show in FIGURE 1, there are four entities in our dCLDAEKS scheme

(i) KGC: Key generation center (KGC) can generate the system parameters and sender/receiver's partial private keys.
(ii) Data Sender: She/he encrypts his/her email by using traditional encryption (i.e. Enc (M)), moreover she/he extracts keywords from each email and encrypted it. Finally sender uploads the encrypted data to cloud server.
(iii) Receiver: She/he can search for specific email interested by sending a trapdoor which contains corresponding keyword to cloud.
(iv) Cloud server: Cloud server is a semi-trusted entity which can perform intricate data analysis and computation. After received a trapdoor from the receiver, cloud server searches over the stored ciphertexts by using its own secret key. If it exists, cloud server returns the corresponding encrypted emails to user.

C. DEFINITION OF dCLDAEKS SCHEME

The specific algorithms of dCLDAEKS are as follow:

- $(pp, s) \leftarrow Setup\ (k)$: Given a security parameter$k$, KGC returns master key s and the public parameter pp.
- $D_u \leftarrow Extract\ partial\ private\ key\ (pp, s, ID_u)$: Given public parameter pp, master key s, user's identity ID_u, it outputs the partial private key of user D_u.
- $(PK_u, K_u) \leftarrow Set\ public\ key\ (pp, ID_u)$: Given user's identity ID_u, user selects secret value x_u. Then it outputs user's public key PK_u, K_u.
- $SK_u \leftarrow Set\ private\ key\ (x_u, D_u)$: Given D_u obtained from the KGC and secret value x_u, it returns user's private key SK_u.
- $(SK_{svr}, PK_{svr}) \leftarrow Set\ server\ key\ (pp)$: Given public parameter pp, it outputs the public key and secret key of server (SK_{svr}, PK_{svr}).
- $CT \leftarrow$ d$CLDAEKSEnc\ (pp, w, PK_{svr}, SK_s, PK_r)$: Given public parameter$pp$, keyword w, sender's private key SK_s, server's public key PK_{svr} and receiver's public key PK_r, it outputs the ciphertext CT.
- $T_w \leftarrow Trapdoor\ (pp, w, PK_{svr}, SK_r, PK_s)$: Given public parameter pp, keyword w, receiver's private key SK_r, sender's public key PK_s and server's public key PK_{svr}, it outputs a trapdoor T_w.
- $\beta \leftarrow Test\ (pp, SK_{svr}, CT, T_w)$: Given public parameter pp, server's private key SK_{svr}, ciphertext CT and trapdoor T_w, if $CT$and T_w contain the same keyword it outputs a bit $\beta = 1$ and returns corresponding ciphertext, and 0 otherwise.
- $\beta' \leftarrow Verify\ (SK_r, CT)$: Given public parameter$pp$, ciphertext CT and receiver's private key SK_r, it verify whether the email is sent by data sender. Then it outputs a bit β', if it is, receiver accepts the ciphertext and return $\beta' = 1$, otherwise it outputs 0 and discard CT.

D. SECURITY MODEL OF dCLDAEKS

We prove that the semantic security of dCLDAEKS against inside offline KGA via the following games between a challenger C and two type adversary A_I and A_{II}, where A_I can replace the user's public key but cannot access the master key andA_{II} can access the master key but cannot replace any public key.

1) CIPHERTEXT INDISTINGUISHABILITY

We assure our scheme satisfies ciphertext indistinguishability via **Game 1** and **Game 2**.

Game 1: This game is interactive between adversary A_I and challenger C.

– Setup: C generates the system parameter pp and the master secret key s by running Setup algorithm. Then C keeps s and sends pp to A_I.
– Phase 1: A_I runs the *private key queries, partial key queries, public key queries and public key replacement queries* adaptively as follows:
  – *Public key queries*: A_I queries on identity ID_u. C returns PK_u to A_I as the public key.
  – *Partial private key queries*: A_I queries on identity ID_u, C replies D_u to A_I as the answer of partial private key.
  – *Private key queries*: A_I queries on identity ID_u. C replies SK_u to A_I as the corresponding private key.
  – *Public key replacement queries*: A_I can replace the public key PK_u with PK'_u.
  – *Ciphertext queries*: A_I queries on (ID_s, ID_r, w). C first obtains the private key SK_s by consulting private key queries on identity ID_s, and then it runs **dCLDAEKSEnc**to generate ciphertext CT.
– Challenge: A_I submits ID_s^* of sender, ID_r^* of receiver and two challenge keyword (w_0, w_1) to C. C randomly chooses a bit $\beta \in \{0, 1\}$, computes corresponding CT_β^* and returns it to A_I.
– Phase 2: A_I continues to query similar to Phase 1.
– Guess: A_I outputs a bit $\beta' \in \{0, 1\}$. If $\beta = \beta'$ and ID_s^*, ID_r^* has not been queried for private key, A_I wins the game. The advantage of A_I wins the game is defined as

$$Adv_C^{DBDH}(\lambda) = \left|\Pr\left[\beta = \beta'\right] - 1/2\right|$$

Game 2: This game is interactive between adversary A_{II} and challenger C.

– Setup: C executes the Setup algorithm to generate the system parameter pp and the master secret key s. Then C sends(s, pp) to adversary A_{II}.
– Phase 1: A_{II} runs the *public key queries*, *partial private key queries*, *private key queries* and*public key* adaptively same as **Game 1**.
– Phase 2: A_{II} continues to issue queries similar to Phase 1.
– Challenge: A_{II} submits ID_s^* of sender, ID_r^* of receiver and two challenge keyword (w_0, w_1) to C. C randomly chooses a bit $\beta \in \{0, 1\}$, computes corresponding CT_β^* and returns it to A_{II}.
– Guess: A_{II} outputs a bit $\beta' \in \{0, 1\}$. If $\beta = \beta'$ and ID_r^* has not been queried for private key, A_{II} wins the game. The advantage of A_{II} wins the game is defined as $Adv_C^{CDH}(\lambda) = \left|\Pr\left[\beta = \beta'\right] - 1/2\right|$.

2) TRAPDOOR INDISTINGUISHABILITY

Game 3: This game is interactive between adversary A_I and challenger C.
The procedures of Setup, Phase 1 and Phase 2 are same as **Game 1**, except:

– Challenge: A_I submits ID_s^* of sender, ID_r^* of receiver and two challenge keyword (w_0, w_1) to C. C randomly chooses a bit $\beta \in \{0, 1\}$, computes corresponding trapdoor T_β^* and returns it to A_I.
– Guess: A_I outputs a bit $\beta' \in \{0, 1\}$, if $\beta = \beta'$ and ID_s^*, ID_r^* has not been queried for private key and tuple $\langle w_0^*, ID_s^*, ID_r^*\rangle$, $\langle w_1^*, ID_s^*, ID_r^*\rangle$ has not been queried for Trapdoor nor ciphertext, A_I wins the game. The advantage of A_I wins the game is defined as

$$Adv_C^{DBDH}(\lambda) = \left|\Pr\left[\beta = \beta'\right] - 1/2\right|.$$

Game 4: This game is interactive between adversary A_{II} and challenger C.
The procedures of Setup, Phase 1 and Phase 2 are identical to **Game 2**, except:

– Challenge: A_{II} submits ID_s^*, ID_r^* and two challenge keyword (w_0, w_1) to C. C randomly chooses $\beta \in \{0, 1\}$, computes corresponding T_β^* and returns it to A_{II}.
– Guess: A_{II} outputs a bit $\beta' \in \{0, 1\}$, A_{II} wins the game if $\beta = \beta'$, ID_r^* has not been queried for private key and $\langle w_0^*, ID_s^*, ID_r^*\rangle$, $\langle w_1^*, ID_s^*, ID_r^*\rangle$ have not been queried for Trapdoor nor ciphertext, A_{II} wins the game. The advantage of A_{II} wins the game is defined as

$$Adv_C^{CDH}(\lambda) = \left|\Pr\left[\beta = \beta'\right] - 1/2\right|.$$

3) DENIABLE AUTHENTICATION

We assure our scheme satisfies DA-CMA via **Game 5** and **Game 6**, which include the interactions between two type adversaries F_I, F_{II} and the challenger C.

Game 5: This game is interactive between adversary F_I and challenger C.

– Setup: C generates the system parameters pp. Then C returns pp to F_I.
– Attack: F_I issues queries which are same with **Game 1**.
– Forgery: F_I outputs a tuple $(\delta^*, ID_s^*, ID_r^*, PK_s^*, PK_r^*)$, if δ^*i s valid with ID_s^*, ID_r^* and the public keys PK_s^*, PK_r^* respectively, F_I wins the game. At same time,F_I cannot make *private key queries*, *public key replacement* and *partial private key queries*on ID_s^*, ID_r^*.

Game 6: This game is interactive between adversary F_{II} and challenger C.

– Setup: C generates the system parameters pp and master key s. Then C returns them to F_{II}.
– Attack: F_{II} issues queries which are same with **Game 2**.
– Forgery: F_{II} outputs a tuple $(\delta^*, ID_s^*, ID_r^*, PK_s^*, PK_r^*)$ and F_{II} wins the game under the condition which is similar with **Game 5**, except that F_{II} is only disallowed to make *private key queries*.

III. THE PROPOSED SCHEME

In this section, a complete dCLDAEKS will be shown.

A. dCLDAEKS SCHEME

As follow, we propose our scheme with following algorithms:

- **Setup**: Given a security parameter k, KGC randomly selects number $s \in Z_q^*$ as the master key and computes

$P_{pub} = sP$. Finally KGC generates the public parameters $pp = (G_1, G_2, e, q, P, P_{pub}, H, h, H_1, H_2, H_3)$, where G_1 is a cyclic addition group, G_2 is a cyclic multiplicative group, $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing, P is the generator of G_1 and the hash function $H : G_2 \times G_1 \times \{0,1\}^* \rightarrow G_1, h : \{0,1\}^* \rightarrow Z_q^*, H_1 : \{0,1\}^* \rightarrow G_1, H_2 : G_2 \times G_1 \rightarrow G_1, H_3 : G_1 \rightarrow Z_q^*$.

- **Extract partial private key**: KGC takes the identity of user as input to generate user's partial private key. Then KGC executes the following steps:
  1) Takes the identity ID_u of user, KGC computes $Q_u = H_1(ID_u)$.
  2) Computes the partial private key $D_u = sQ_u$ where s is the master key. Finally KGC returns D_u to user.
- **Set public key:** User selects a random number $x_u \in Z_q^*$ as his secret value and computes public keys $PK_u = x_u Q_u$ and $K_u = x_u P$.
- **Set private key:** User generates his own private key $SK_u = (x_u, D_u)SK_u = (x_u, D_u)$ by using the partial private key D_u obtained from KGC and the secret value x_u selected by himself.
- **Set server key:** KGC randomly selects $t \in Z_q^*$, then returns the server's private/public key pair $PK_{svr} = tP, SK_{svr} = t$.
- **dCLDAEKSEnc:** Given a keyword $w \in \{0,1\}^*$, the private key $SK_s$of data sender, receiver's public key PK_r, K_r and the server's public key PK_{svr} as input.
  (1) Sets $U = x_s K_r$and computes $k = e(D_s, PK_r)$.
  (2) Randomly selects $r \in Z_q^*$ and computes $X = rQ_s$, $C_1 = e(H(k, U, w), PK_{svr})^r, C_2 = rK_r$, $C_3 = rH_2(C_1, C_2)$.
  (3) Computes $z = h(w, PK_s, PK_r, C_1, U)$, $V = e((r+z) \cdot D_s, Q_r)$.
  (4) Returns $CT = (C_1, C_2, C_3, X, V)$ as the ciphertext.
- **Trapdoor:** Takes the receiver's private key SK_r and the public key PK_s of the sender as input.
  (1) Computes $k = e(Q_s, x_r D_r)$, $U = x_r K_s$.
  (2) Randomly chooses $r' \in Z_q^*$, computes $T_1 = r'P$ and $T_2 = \frac{1}{x_r} H_3(r' PK_{svr}) H(k, U, w)$.
  (3) Returns $T_w = (T_1, T_2)$ as the Trapdoor.
- **Test:** Takes the server's private key, ciphertext CT and trapdoor T_w as input. Sever parses CT as (C_1, C_2, C_3) and T_w as (T_1, T_2), then it computes

$$T = \frac{T_2}{H_3(SK_{svr} T_1)} = \frac{1}{x_r} H(k, U, w).$$

Finally, cloud server returns corresponding ciphertext if equation (1) and (2) holds, otherwise 0.

$$e(C_2, H_2(C_1, C_2)) = e(K_r, C_3) \quad (1)$$
$$C_1 = e(SK_{svr} T, C_2) \quad (2)$$

- **Verify:** Takes the private key of receiver SK_r and ciphertext CT.

Data receiver verifies whether $V = e(X + zQ_s, D_r)$ holds. If it holds, receiver accepts the ciphertext, otherwise outputs 0.

B. CORRECTNESS ANALYSIS

We set the trapdoor of keyword w' as $T_{w'}$, w is the keyword included in ciphertext CT. Only when $w' = w$, can we compute the following equation holds.

$$C_1 = e(H(k, U, w), PK_{svr})^r \quad (3)$$

$$\begin{aligned} e(SK_{svr} T, C_2) &= e(t \cdot \frac{1}{x_r} \cdot H(k, U, w), rK_r) \\ &= e(t \cdot H(k, U, w), \frac{1}{x_r} \cdot r \cdot x_r P) \\ &= e(H(k, U, w), PK_{svr})^r \quad (4) \end{aligned}$$

$$\begin{aligned} e(C_2, H_2(C_1, C_2)) &= e(rK_r, H_2(C_1, C_2)) \\ &= e(K_r, rH_2(C_1, C_2)) = e(K_r, C_3) \quad (5) \end{aligned}$$

$$\begin{aligned} V &= e((r+z)D_s, Q_r) = e((r+z)Q_s, D_r) \\ &= e(X + zQ_s, D_r) \end{aligned}$$

C. DENIABILITY

The legal data receiver can use his own private key to generate a ciphertext which is indistinguishable from the ciphertext generated by data sender. The simulation include the following steps:

(1) Computes $U' = x_r K_s$ and $k' = e(X_s, x_r D_r)$.
(2) Given keyword w, it randomly selects $r' \in Z_q^*$ and computes $X = r'Q_s$, $C_1' = e(H(k', U', w), PK_{svr})^{r'}$, $C_2' = r'K_r$, $C_3' = rH_2(C_1', C_2')$.
(3) Computes $z' = h(w, PK_s, PK_r, C_1', U')$ and $V' = e(X + z'Q_s, D_r)$.
(4) Computes $CT' = (C_1', C_2', C_3', X', V')$.

The ciphertext CT' generated by receiver is indistinguishable from CT of the sender produces in **dCLDAEKSEnc** algorithm. Due to the ciphertext CT' generated by the random value $r' \in Z_q^*$, the probability of ciphertext $CT' = CT$ is

$$\Pr\ [(C_1, C_2, C_3, X, V) = (C_1', C_2', C_3', X', V')] = \frac{1}{q-1}.$$

Therefore, we can say that the ciphertext CT' has the same distributions of probability with CT.

IV. PROVABLE SECURITY

In this section, we prove our**dCLDAEKS** scheme can against inside KGA of Type I adversary A_I and Type II adversary A_{II}. Furthermore we also prove that our scheme satisfies ciphertext indistinguishability and trapdoor indistinguishability and deniable, if DBDH assumption and CDH assumption holds.

A. CIPHERTEXT INDISTINGGUISHABILITY

Lemma 1: We suppose that there exists a PPT adversary A_I who can win **Game 1** with advantage ε under the condition that A_I queries the random oracles h, H_i (where $i = 1, 2$) for most q_h, q_H, q_{H_i}. Then we can construct an algorithm C to solve the DBDH problem with an advantage:

$$Adv_C^{DBDH}(\lambda) \geq \frac{1}{2q_{H_1}(q_{H_1}-1)} \cdot Adv_A^C(\lambda)$$

Proof: Suppose A_I has ability to break the security of dCLDAEKS, then C can use A_I to determine the answer of the DBDH problem by following steps:

Setup: C randomly selects $t \in Z_q^*$, and sets $pp = (G_1, G_2, e, q, P, P_{pub} = cP)$ and the server's private/public key pair (t, tP). Then, C keeps record lists L_h, L_i (where $i = 1, 2$). Moreover, C maintains list L_3 which contains the output of private keys and public keys.

Phase 1: A_I adaptively issues queries to C as follows:

- *H queries*: Given $k \in G_2$, $U \in G_1$ and a keyword w, it randomly selects an element from G_1, then returns it as the reply of $H(k, U, w)$.
- *H_1 queries*: A_I can queries H_1 with different identities. Upon receiving A_I's query on ID_i, if it has been queried, outputs the record in L_1. C randomly chooses $y, \eta \in \{1, \ldots, q\}$ and sets ID_y and ID_η as challenge identities. C replies A_I as follows:
  1) At the y th query, C responds with $H(ID_y) = aP$ and adds $(ID_y, aP, \perp)$ to the record L_1.
  2) At the η th query, C responds with $H(ID_\eta) = bP$ and adds $(ID_\eta, bP, \perp)$ to the record L_1.
  3) Otherwise, C randomly picks $b_i \in Z_q^*$, computes $H(ID_i) = b_iP$ and returns it to A_I. Then C updates L_1 with (ID_i, b_iP, b_i).
- *h queries*: A_I queries on $h(w, PK_s, PK_r, X, U)$, C first checks up the list L_h, if there exists corresponding value in L_h, then C returns it to A_I, if not C randomly picks $z \in Z_q^*$, updates L_h with $h(w, PK_s, PK_r, X, U, z)$.
- *H_2 queries*: Given (C_1, C_2), C randomly picks $\delta \in Z_q^*$, returns $H_2(C_1, C_2, C_3) = \delta aP$ to A_I, and adds the tuple $\langle (C_1, C_2), H_2(C_1, C_2), \delta \rangle$ into L_2.
- *Partial private queries*: Upon receiving A_I's query on ID_i. If $ID_i = ID_y$, C aborts simulation. Otherwise C checks up the record L_1 and returns partial private key $D_i = b_iaP$.
- *Public key queries*: Upon receiving A_I's query on ID_i. C first checks whether L_3 contains $(ID_i, D_i, PK_i, K_i, x_i)$. If it exists, C responses A_I with (PK_i, K_i), else C randomly picks $x_i \in Z_q^*$, returns $PK_i = x_ib_iP$ and $K_i = x_iP$ then updates L_3 with $(ID_i, \perp, PK_i, K_i, x_i)$.
- *Private key queries*: A_I queries identity ID_i. If the public key about ID_i has not been replaced and $ID_i \neq ID_y$, then C checks the list L_3 and replies A_I with $SK_i = (x_i, D_i)$, If id-entity $ID_i \neq ID_y$ or the public key about ID_i has been replaced, then C terminates.
- *Replace public key queries*: A_I can replaces ID_i's public keys (PK_i, K_i) with valid values PK_i' and K_i'. Then C updates L_3 with $(ID_i, D_i, PK_i', K_i', \perp)$.
- *Ciphertext queries*: A_I queries on tuple (ID_s, ID_r, w), C first consults the public key queries to obtain public keys of ID_s and ID_r. Then it retrieves sender's private key from L_3. C randomly selects $r \in Z_q^*$, and returns the ciphertext as follow:

If $(ID_s, ID_r) = (ID_\eta, ID_\gamma)$ or $(ID_r, ID_s) = (ID_\eta, ID_\gamma)$, C computes $X = rP, C_1 = e(H(Z, U, w), PK_{svr})^r$, $C_2 = rK_r, C_3 = rH_2(C_1, C_2), V = rP_{pub}$.

Otherwise, it retrieves D_s from L_3, then computes $k = e(cP, aK_r)^{b_i}$ and returns $X = rQ_s, C_1 = e(H(k, U, w), PK_{svr})^r, C_2 = rK_r, C_3 = rH_2(C_1, C_2), V = e((r+z)D_s, X_r)$

- *Trapdoor queries*: A_I queries on tuple (ID_s, ID_r, w), C first consults the public key queries to obtain public keys of ID_s and ID_r. Then it retrieves receiver's private key from L_3. C randomly selects $r' \in Z_q^*$ and returns the trapdoor as follow:

If $ID_s = ID_y$, it computes $T_1 = r'P$, $T_2 = 1/x_r \cdot H_3(r'PK_{svr}) \cdot H(Z, U, w)$. Otherwise it retrieves D_r, x_r from L_3, then computes $k' = e(H(ID_s), x_raP)^{b_i}$ and returns $T_1 = r'P$, $T_2 = 1/x_r \cdot H_3(r'PK_{svr})H(k', U, w)$.

Challenge: A_I queries on two keywords w_0^*, w_1^* with two identities ID_s^*, ID_r^* which it expects to be challenged. C randomly chooses a bit$\beta \in \{0, 1\}$, an number $r^* \in Z_q^*$ and returns the ciphertext

$$\begin{aligned} X^* &= r^*P, \quad V^* = r^*P_{pub}, \\ C_{1,\beta} &= e(H(Z, U, w_\beta), PK_{svr})^r, \\ C_{2,\beta} &= rK_r, \quad C_{3,\beta} = rH_2(C_1, C_2). \end{aligned}$$

Phase 2: A_I can adaptively issue queries similar to **Phase 1**.

Guess: A_I outputs a bit $\beta' \in \{0, 1\}$. C outputs $\beta' = 0$, if $\beta' = \beta$. Otherwise it outputs 1.

Analysis: C would abort **Game 1** if challenge identity is ID_η, ID_y (E_1 denotes this event). The probability that E_1 does not happen is $\frac{1}{q_{H_1}(q_{H_1}-1)}$. Suppose that C does not abort, if $Z = e(P, P)^{abc}$. Then the probability that A_I would win the game is $Adv_{A_1}^C(\lambda) + 1/2$. If Z is randomly selected from G_2, then $k = H(Z, U, w_\beta)$ is also random element of G_2. Hence, C can solves the DBDH problem with an advantage:

$$\begin{aligned} &Adv_C^{DBDH}(\lambda) \\ &= \left| \Pr[\beta = \beta' | E_1] \cdot \Pr\ [E_1] + \Pr[\beta = \beta' | \bar{E}_1] \cdot \Pr[\bar{E}_1] - \frac{1}{2} \right| \\ &\geq \left| \frac{1}{2}(1 - \Pr\ [\bar{E}_1) + \Pr\ [\bar{E}_1] \cdot ((Adv_{A_1}^C(\lambda) + \frac{1}{2}) \right. \\ &\quad \left. \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2}) - \frac{1}{2} \right| \\ &= \frac{1}{2} \Pr\ [\bar{E}_1] \cdot Adv_{A_I}^C(\lambda) = \frac{1}{2q_{H_1}(q_{H_1}-1)} \cdot Adv_{A_I}^C(\lambda). \end{aligned}$$

However, $Adv_{A_I}^C(\lambda)$ is non-negligible. So $Adv_C^{DBDH}(\lambda)$ is non-negligible.

Lemma 2: We suppose that there exists PPT adversary A_{II} who can wins **Game 2** with advantage ε under the condition that A_{II} queries the random oracles *h, H, H_i* (where $i = 1, 2$) for most q_h, q_H, q_{H_i}.

Then C can solve the CDH problem with an advantage:

$$Adv_C^{CDH}(\lambda) = \Big|\Pr[\beta = \beta'|E_2] \cdot \Pr[E_2] + \Pr[\beta = \beta'|\bar{E}_2] \cdot \Pr[\bar{E}_2] - \frac{1}{2}\Big| \geq \frac{1}{q_{H_1}} Adv_{A_{II}}^{C}(\lambda).$$

Proof: Suppose A_{II} has ability to break the security of proposed scheme, then C can use A_{II} to solve the CDH problem by interacting with A_{II} as follows:

Setup: C randomly selects $t \in Z_q^*$, and sets $pp = (G_1, G_2G_2, e, q, P, P_{pub} = sP)$ and the server's private/public key pair (t, tP). Then, C keeps records L_h and L_i (where $i = 1, 2$). Moreover, C maintains list L_3 which contains the output of private keys and public keys.

Phase 1: A_{II} adaptively issues queries to$C$ as follows:

- *h queries*: A_{II} queries on $h(w, PK_s, PK_r, X, U)$, C first checks up the list L_h. If there exists corresponding value in L_h, then C returns it to A_{II}, if not C randomly picks $z \in Z_q^*$, updates L_h with $h(w, PK_s, PK_r, X, U, z)$.
- *H_1 queries*: A_{II} can queries H_1 oracle with different identities. Upon received A_{II}'s query on ID_i, if it has been queried, outputs the answer recorded in L_1. Then C randomly picks $b_i \in Z_q^*$, computes $H(ID_i) = b_iP$ and returns it to A_{II}. Then C updates L_1 with (ID_i, b_iP, b_i).
- *H_2 queries*: Given (C_1, C_2), C randomly picks $\delta \in Z_q^*$, returns $H_2(C_1, C_2, C_3) = \delta aP$ to A_{II}, and adds the tuple $\langle (C_1, C_2), H_2(C_1, C_2), \delta \rangle$ into L_2.
- *Public key queries*: Upon received A_{II}'s query on ID_i, C first checks whether L_3contains $(ID_i, D_i, PK_i, K_i, x_i)$. If it exists, C responses A_{II} with (PK_i, K_i). Else C randomly picks $x_i \in Z_q^*$. If $ID_i = ID_\gamma$, C replies with $K_\gamma = x_iaP$, $PK_\gamma = x_iaQ_\gamma$ and if $ID_i = ID_\eta$, C replies with $K_\eta = x_ibP$, $PK_\eta = x_ibQ_\eta$. If $ID_i \neq ID_\gamma, ID_\eta$, then C returns $K_i = x_iP$ and $PK_i = x_ib_iP$ then updates L_3 with (ID_i, PK_i, K_i, x_i).
- *Private key queries*: A_{II} queries C with identity ID_i. C checks the list L_3 and replies A_{II} with $SK_i = (x_i, D_i)$. If $ID_i \neq ID_y$ and $ID_i \neq ID_\eta$, C terminates.
- *H queries*: A_{II} queries C with tuple $H(k, U, w)$. C searches the list L_H with $(k, *, w)$, C returns $H(k, U, w)$ where $e(x_i^2P, U) = e(v_iaP, v_ibP)$, if there exists such a tuple, C updates symbol $*$ with U.
- *Ciphertext queries*: A_{II} queries on tuple (ID_s, ID_r, w), C first queries on ID_s and ID_r in the public key queries to obtain their public keys. Then it retrieves sender's private key $SK_s = (x_s, D_s)$ from L_3. C randomly selects $r \in Z_q^*$, and returns the ciphertext as follow:
- If $ID_s \neq ID_y, ID_\eta$ then C runs the **dCLDAEKSEnc** algorithm to reply A_{II}.
- If $ID_s = ID_y$ or $ID_s = ID_\eta$, it computes $X = rP$, $V = rP_{pub}, C_1 = e(H(k, x_i^2abP, w), PK_{svr})^r$, $C_2 = rK_r, C_3 = rH_2(C_1, C_2)$.
- *Trapdoor queries*: A_{II} queries on tuple (ID_s, ID_r, w), C first execute the public key queries to obtain their public keys. Then it retrieves receiver's private key from L_3. C randomly selects $r' \in Z_q^*$, and returns the trapdoor as follow:
- If $ID_r = ID_y, ID_\eta$, it computes $T_1 = r'P$, $T_2 = 1/x_r \cdot H_3(r'PK_{svr})H(k, x_i^2abP, w)$. Otherwise C retrieves D_r, x_r from L_3, then C runs the **Trapdoor** algorithm to reply A_{II}.
- *Challenge*: A_{II} queries on two keywords w_0^* and w_1^* with two challenged identities ID_s^* and ID_r^*. C randomly chooses a bit $\beta \in \{0, 1\}$, $r \in Z_q^*$, sets $K_r^* = x_ibP$, then C randomly chooses $k \in G_2$ and an random element from G_2 as the value of $H(k, U, w_\beta)$, finally returns the ciphertext, where

$$X^* = rQ_s^*, \quad V^* = e((r+z)D_s^*, X_r^*),$$
$$C_{1,\beta} = e(H(k, U, w_\beta),$$
$$PK_{svr})^r, C_{2,\beta} = rK_r^* C_{3,\beta} = rH_2(C_{1,\beta}, C_{2,\beta}).$$

Phase 2: A_{II} can continue to query identical with **Phase 1**.

Guess: A_{II} outputs a bit $\beta' \in \{0, 1\}$.

Analysis: C would aborts **Game 2** if adversary A_{II} queries private key with identity ID_y and ID_η (E_2 denotes this event). The probability that E_2 does not happen is $\frac{1}{q_{H_1}}$.Suppose that C does not abort. Only if $U^* = x_i^2abP$, would A_{II} can produces the correct guess. Hence, C verifies whether $e(x_i^2P, U^*) = e(x_iaP, x_ibP)$. If true, it returns $U^* = abP$, and A_{II} would win the game with probability $Adv_{A_{II}}^{C}(\lambda) + 1/2$. Therefore, the advantage of C solving the CDH problem is

$$Adv_C^{CDH}(\lambda) = \Big|\Pr[\beta = \beta'|E_2] \cdot \Pr[E_2] + \Pr[\beta = \beta'|\bar{E}_2] \cdot \Pr[\bar{E}_2] - \frac{1}{2}\Big| \geq \frac{1}{q_{H_1}} \cdot Adv_{A_{II}}^{C}(\lambda)$$

However, $Adv_{A_{II}}^{C}(\lambda)$ is non-negligible. So $Adv_C^{CDH}(\lambda)$ is non-negligible.

B. TRAPDOOR INDISTINGGUISHABILITY

Lemma 3: For type I adversary A_I, we can prove that **dCLDAEKS** satisfies trapdoor indistinguishability via DBDH assumption. The proof is identical with **Lemma 1**, except that C generates challenge trapdoor as $T_\beta = (T_1, T_2)$ where $T_1 = r^*P$, $T_2 = 1/x_r \cdot H_3(r^*PK_{svr}) \cdot H(Z, U, w_\beta)$, $r \in Z_q^*$ is randomly selected by C. Here we omit the detail of proof process.

Lemma 4: For type II adversary A_{II}, we can also prove that dCLDAEKS satisfies trapdoor indistinguishability via CDH assumption. The proof is identical with **Lemma 2**, expect that C generates the challenge trapdoor as $T_\beta = (T_1, T_2)$ where $T_1 = r^*P$, T_2=$1/x_r \cdot H_3(r^*PK_{svr}) \cdot H(k, U, w_\beta)$, $r^* \in Z_q^*$ is randomly selected by C. Here we omit the detail of process.

C. DENIABLE AUTHENTICATION

In the random oracle model, we can prove that our **dCLDAEKS** scheme is DA-CMA secure. The scheme can

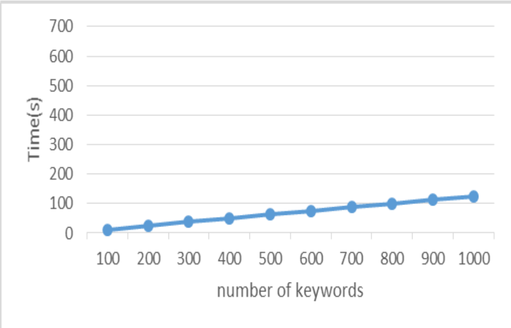

FIGURE 2. The section of encryption.

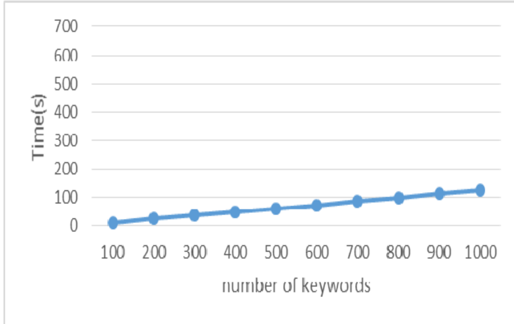

FIGURE 3. The section of trapdoor.

against two type adversary F_I and adversaryF_{II}, if DBDH and CDH problem holds.

Lemma 5: The proof is similar with the section of deniable authentication in literature [17], except that $\delta_2 = V$ in our scheme.

Lemma 6: The proof is similar with section of the deniable authentication in literature [17], except that in **Forgery** phase of our scheme, F_{II} can produces a ciphertext $\delta^* = (X^*, C_1^*, C_2^*, C_3^*, V^*)$. Only when F_{II} queries for hash value $H(k^*, U^* = x_i^2 abP, w^*)$, can$F_{II}$ discern that δ^* is an invalid ciphertext. So we have $e(P, x_i^2 U^*) = e(K_s, K_r) = e(x_i aP, x_i bP) = e(P, x_i^2 abP)$. Hence, C can successfully compute $U^* = abP$.

Analysis process is identical to the description in **Lemma 2**.

V. PERFORMANCE ANALYSIS

We carry out our scheme using JPBC in a personal computer with i5-2400 3.10GHZ processor, 4GB memory, and Windows 10 operating system. FIGURE. 2 and FIGURE. 3 respectively demonstrate the running time of keyword encryption and Trapdoor algorithms of **dCLDAEKS** as the number of keywords increases. FIGURE. 4 demonstrates the running time of Test algorithm of **dCLDAEKS** as the number of ciphertexts increases. Moreover we compare our scheme with related schemes [10], [11], [24] in terms of security

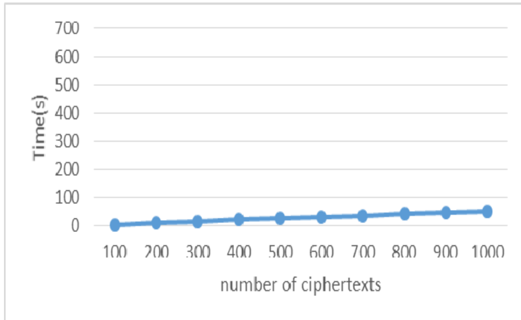

FIGURE 4. The section of test.

and computation efficient. From TABLE 2, we can get the information that scheme [10] can only resist outside and inside offline KGA, but it does not satisfy the deniability and it without designated tester. Scheme [24] needs transmitted via secure channel and it only resist outside KGA. Although scheme [11] avoids some above problems, but it does not satisfy deniability and suffer key escrow issue. In computation performance, **dCLDAEKS** is slightly less computationally efficient with the related schemes, but it is still competitive. Because **dCLDAEKS** scheme provides better sender's privacy protection and stronger security.

A. PERFORMANCE COMPARISON

TABLE 1. Computational comparison.

	Encrypt	Trapdoor	Test
Scheme [10]	3e + H + m	e + H + Pa	2Pa + m
Scheme [24]	3e + H + Pa + m	e + H	H + Pa
Scheme [11]	3e + H + 2Pa	2e + H + 2Pa	2e + 3Pa + m
dCLDAEKS	4m + 3Pa + 3H	5m + 2H + sm	2m + 2Pa + sm + H

e denotes the modular exponentiation of G1 element, H denotes the computation of a hash function, Pa denotes computation of a bilinear pairing, m denotes a multiplication , sm denotes a scalar multiplication.

TABLE 2. Security comparison.

	KGA/SC/O	KGA/WSC/O	IKGA	designated tester	Deniability
Scheme [10]	✓	✓	✓	×	×
Scheme [24]	✓	×	×	×	×
Scheme [11]	✓	✓	✓	✓	×
dCLDAEKS	✓	✓	✓	✓	✓

KGA/SC/O: resist outside keyword guessing attacks (KGA) with secure channel. KGA/WSC/O: resist outside KGA with free channel. IKGA: resist inside KGA.

B. SIMULATION EXPERIMENT

See Figs. 2–4.

VI. CONCLUSION

In this paper, we construct designated server certificateless deniably authenticated encryption with keyword search (dCLDAEKS). Moreover, we prove that our scheme is secure against inside offline KGA of two type adversary. Meanwhile, **dCLDAEKS** scheme can satisfies ciphertext indistinguishability and trapdoor indistinguishability. In protecting user identity privacy, we combine denial authentication technology to achieve the goal of protecting data sender identity privacy. In addition, we adopt the method of designated tester to execute ciphertext searching operation, which further ensures that no adversary can launch offline KGA in our scheme. All in all, though **dCLDAEKS** scheme perform less efficient in some procedure, but it can provide better sender's privacy protection and stronger security.

REFERENCES

[1] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 912–925, Apr. 2018. doi: 10.1109/TIFS.2017.2774439.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, May 2000, pp. 44–55.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Interlaken, Switzerland, 2004, pp. 506–522.

[4] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2706–2716, Dec. 2016. doi: 10.1109/TIFS.2016.2596138.

[5] Z. Fu, X. Wu, Q. Wang, and K. Ren, "Enabling central keyword-based semantic extension search over encrypted outsourced data," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 2986–2997, Dec. 2017. doi: 10.1109/TIFS.2017.2730365.

[6] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. Int. Conf. Comput. Sci. Appl.*, Perugia, Italy, 2008, pp. 1249–1259.

[7] H. S. Rhee, J. H. Park, and D. H. Lee, "Generic construction of designated tester public-key encryption with keyword search," *Inf. Sci.*, vol. 205, no. 1, pp. 93–109, Nov. 2012. doi: 10.1016/j.ins.2012.03.020.

[8] K. Emura, A. Miyaji, M. S. Rahman, and K. Omote, "Generic constructions of secure-channel free searchable encryption with adaptive security," *Secur. Commun. Netw.*, vol. 8, no. 8, pp. 1547–1560, Sep. 2015. doi: 10.1002/sec.1103.

[9] J. W. Byun, H. S. Rhee, D. H. Lee, and H.-A. Park, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Proc. Workshop Secure Data Manage.*, Seoul, South Korea, 2006, pp. 75–83.

[10] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Inf. Sci.*, vols. 403–404, pp. 1–14, Sep. 2017. doi: 10.1016/j.ins.2017.03.038.

[11] H. Li, Q. Huang, J. Shen, G. Yang, and W. Susilo, "Designated-server identity-based authenticated encryption with keyword search for encrypted emails," *Inf. Sci.*, vol. 481, pp. 330–343, May 2019. doi: 10.1016/j.ins.2019.01.004.

[12] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3618–3627, Aug. 2017. doi: 10.1109/TII.2017.2771382.

[13] Y. Peng, J. Cui, C. Peng, and Z. Ying, "Certificateless public key encryption with keyword search," *China Commun.*, vol. 11, no. 11, pp. 100–113, Nov. 2014. doi: 10.1109/CC.2014.7004528.

[14] T.-Y. Wu, F. Meng, C.-M. Chen, J.-S. Pan, and S. Liu, "On the security of a certificateless searchable public key encryption scheme," in *Proc. Int. Conf. Genetic Evol. Comput.*, Fuzhou, China, 2016, pp. 113–119.

[15] M. Ma, D, He, D. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, May 2017. doi: 10.1109/TII.2017.2703922.

[16] M. Ma, D. He, M. K. Khan, and J. Chen, "Certificateless searchable public key encryption scheme for mobile healthcare system," *Comput. Electr. Eng.*, vol. 65, pp. 413–424, Jan. 2017. doi: 10.1016/j.compeleceng.2017.05.014

[17] T. S. Wu and H. Y. Lin, "Provably secure proxy convertible authenticated encryption scheme based on RSA," *Inf. Sci.*, vol. 278, no. 10, pp. 577–587, Sep. 2014. doi: 10.1016/j.ins.2014.03.075.

[18] D. Maimut and R. Reyhanitabar, "Authenticated encryption: Toward next-generation algorithms," *IEEE Security Privacy*, vol. 12, no. 2, pp. 70–72, Mar./Apr. 2014. doi: 10.1109/MSP.2014.19.

[19] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 4, pp. 33-1–33-16, Dec. 2010. doi: 10.1145/1880022.1880027.

[20] F. Li, J. Deng, and T. Takagi, "An improved authenticated encryption scheme," *IEICE Trans.*, vol. 94, no. 11, pp. 2171–2172, Feb. 2011. doi: 10.1587/transinf.E94.D.2171.

[21] F. Li, D. Zhong, and T. Takagi, "Efficient deniably authenticated encryption and its application to E-mail," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2477–2486, Nov. 2016. doi: 10.1109/TIFS.2016.2585086.

[22] W. Wu and F. Li, "An efficient identity-based deniable authenticated encryption scheme," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 5, pp. 1904–1919, May 2016. doi: 10.3837/tiis.2015.05.020.

[23] E. Ahene, C. H. Jin, and F. Li, "Certificateless deniably authenticated encryption and its application to e-voting system," *Telecommun. Syst.*, vol. 70, no. 3, pp. 417–434, Mar. 2018. doi: 10.1007/s11235-018-0496-3.

[24] K. Tomida, Y. Shiraishi, and M. Mohri, "Keyword searchable encryption with access control from a certain identity-based encryption," in *Future Information Technology* (Lecture Notes in Electrical Engineering) vol. 276. Berlin, Germany: Springer, 2014, pp. 113–118.

YULEI ZHANG was born in Gansu, China, in 1979. He received the Ph.D. degree in cryptography from Northwest Normal University, in 2015, where he is currently a Professor. His research interests include cryptology and information security.

LONG WEN was born in Wuhan, Hubei, China, in 1996. He is currently pursuing the master's degree with the College of Computer Science and Engineering, Northwest Normal University, Lanzhou, China. His research interests include cryptology and information security.

YONGJIE ZHANG was born in Gansu, China, in 1978. She is currently a Professor with the Gansu Health Vocational College. Her research interests include cryptology and information security.

CAIFEN WANG was born in Anguo, Hebei, China, in 1963. She received the Ph.D. degree in cryptography from the School of Telecommunications Engineering, Xidian University, in 2003. She is currently a Professor with Shenzhen Technology University. Her research interests include cryptology and information security.