IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 5, 2020, accepted April 20, 2020, date of publication April 22, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989628

An Efficient Public Key Searchable Encryption Scheme for Mobile Smart Terminal

NINGBIN YANG, SHUMEI XU, AND ZHOU QUAN
School of Mathematics and Information Sciences, Guangzhou University, Guangzhou 510006, China

Corresponding author: Zhou Quan (zhouqq@gzhu.edu.cn)

This work was supported in part by the Key-Area Research and Development Plan of Guangdong province under Grant 2019B020215004, in part by the National Key Research and Development Plan under Grant 2018YFB0803600, in part by the Foundation of National Natural Science of China under Grant 61772147, and in part by the National Cryptography Development Fund under Grant MMJJ20170117.

ABSTRACT With the wide application of the mobile smart terminals, the data privacy protection of the mobile smart terminals stored in the cloud is more and more important. Public key encryption with keyword search (PEKS) and secure channel free PEKS (SCF-PEKS) have been proposed for public key searchable encryption previously. However, the security of keyword search is far from enough. In addition, these schemes are mostly based on bilinear pairing and the computational efficiency is relatively low. In this paper, we propose a novel non-bilinear pairs SCF-PEKS schemes for mobile smart terminal that offer a high computational efficiency along with better security assurances than that of the existing alternatives. Without random oracle model, we prove the security and privacy of the scheme's keyword ciphertext and keyword trapdoor through the game hopping method. Therefore, the scheme is capable of resisting outside online keyword guessing attack and inside offline keyword guessing attack. Based on the comparison and experimental results, the scheme turns out to be secure and practicable.

INDEX TERMS Searchable encryption, public key encryption, keyword guessing attacks, without bilinear pair operation.

I. INTRODUCTION

In recent years, with the rapid development and extensive application of cloud computing technology and 5G communication, the number of cloud users has been increasing rapidly. As a result, cloud storage and data analytics services are increasingly available to the public, such as amazon's AWS and Google's Drive. These cloud service platforms all have cloud computing technology capability. Cloud computing has the advantages of unlimited storage space, fast computing, high service availability, and low cost. It allows users to outsource data hosting and program execution to the third party with much greater storage, computational, and network capacities, which known as Cloud Service Provider. The cloud sever provider can avoid tedious data management and storage on battery-limited devices. It provides convenient services while reducing the need of terminal equipment. As a consequence, this provides great convenience for cloud mobile users with limited devices.

In addition, with the upgrading of mobile smart terminals, the application capacity of mobile smart terminals is becoming more and more powerful. A growing number of people upload personal data privacy to cloud service providers for storage with the help of mobile smart terminals. However, there are various security threats to cloud storage. It is easy for suspected personnel to stole the personal privacy data and use illegally. Therefore, the privacy protection of cloud data attracts more and more public attention.

Traditional encryption methods can protect the privacy of data from malicious the cloud sever provider, but also prevent the cloud sever provider from searching the data on behalf of users. Searchable encryption [1], [2] is an effective method to solve the privacy problem of cloud storage, and public key searchable encryption is one of the methods of searchable encryption. For the public key searchable encryption scheme, there are three parties involved, a data owner called Alice, a data user called Bob and a cloud sever provider. First, Alice prepares a file to share with her friend Bob, and sets a keyword "*encryption*" for the file. Then, Alice upload the encrypted file with keyword ciphertext to the cloud sever provider. To search over the encrypted file, Bob can then use his secret key to generate the trapdoor corresponding to keyword "$w = encryption$", and enable the cloud sever provider to retrieve all files that are associated with the keyword w.

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

After the search finishes, the cloud sever provider returns the search results to Bob. Therefore, Bob can determine whether files with the desired keywords are included in the cloud sever provider. If it does, Bob can decrypt the encrypted file. During the search of Bob, the cloud sever provider does not know the detail of file, nor the keyword.

At present, searchable encryption technology is widely used, such as cloud encryption storage [2], intelligent mail routing [2]–[5], mobile electronic medical information system [6], [7], smart grid system [8] and internet of things [9]–[12].

A. RELATED WORK

The proposal of searchable encryption scheme has attracted the attention and research of many scholars. In 2000, Song *et al.* [2] proposed a searchable encryption scheme for the first time, but the scheme required traversing all the files to return the results, which required a large computational cost. In 2004, Boneh *et al.* [3] came up with a public key encryption with keyword search (PEKS) scheme. Soon afterwards, many PEKS schemes and variants [13]–[15] were presented. Nevertheless, there is an obvious weakness in the PEKS scheme that keywords trapdoor need to be secretly transmitted to the cloud server. In 2008, Baek *et al.* [16] proposed a secure channel free public key encryption with keyword search (SCF-PEKS) scheme to address PEKS problem. The proofs of Boneh *et al.* and Baek *et al.* were carried out under the random oracle. The random oracle is a theoretical black box that returns a true uniform random output to any input. In other words, the oracle will output in the same way every time if it takes the same words as input. Fang *et al.* [17] proposed to prove the keyword security of SCF-PEKS without random oracle model. However, both PEKS and SCF-PEKS schemes have the privacy problem of keywords. Rhee *et al.* [29]. introduced the ''trapdoor indistinguishability'', and showed that this is a sufficient condition against keyword guessing attacks. Yau *et al.* [4] described the possible attack scenarios in the current nature of the internet and public key encryption with keyword search applications, e.g. email routing. It claims to be secure against the keyword guessing attacks by the outsider attacker.

Whereas, Shao and Yang [15] argued that the existing PEKS scheme and SCF-PEKS scheme are not safe in the attack of malicious server's keyword guessing. Therefore, the privacy of keywords in public key searchable encryption scheme becomes a problem that researchers need to address. According to the attackers' attack pattern, they can be divided into online keyword guessing attack and offline keyword guessing attack. Noroozi and Eslami [18] brought up a new PEKS scheme to resist offline and online keyword attacks by outside attackers. According to the type of attacker, it can be divided into inside attacker and outside attacker. We often refer to malicious cloud servers as inside attackers and attackers other than the cloud servers as outside attackers. Only by ensuring the keywords ciphertext indistinguishability and keywords trapdoor indistinguishability can the privacy security of keywords be realized. Huang and Li [19] pointed out that almost all existing PEKS schemes are vulnerable to inside keyword guessing attack, an inside adversary can determine the keyword information by exhausting the keyword space offline to test the matching of the search keyword with the trapdoor. Therefore, they introduced a scheme to resist keyword guessing attacks by inside attackers, but there is still possibility of keyword statistical information leakage. Wang *et al.* [20] improved the scheme proposed by Huang and Li [19] and proposed a scheme of trapdoor uncertainty to prevent keyword statistical information from leak out and to resist keyword guessing attack by inside attackers. Hwang *et al.* [21] proposed that the public key improved by ElGamal could be searched and encrypted, which could resist the keyword guessing attack of outside attackers. Xu and Lu [22], Lu *et al.* [23], [24] proposed a keyword trapdoor with access control function, and proved that it could resist the known keyword guessing attack without random oracle model. The game-hopping method proposed by Alexander in literature [25] is a method to verify the security of cryptographic scheme, and the attacker runs an unknown probability of success in a specific attack environment. It bound the increase in the attacker's success probability caused by the changes to the attack environment. Thus, it can deduce a bound for the attacker's success probability in the original environment. Therefore, we can judge the security of the cryptographic scheme.

B. OUR CONTRIBUTION

How to improve the privacy security of the keywords in the public key searchable encryption to resist the online and offline keyword guessing attack and achieve higher computational efficiency in the use of mobile smart terminals is our intensive research.

In this work, we come up with a new efficient secure channel free public key searchable encryption without using bilinear pair operation scheme, which is able to against the existing outside and inside keyword guessing attacks in SCF-PEKS scheme. Roughly, we make the contributions as follows:

1) We propose a public key searchable encryption based on non-bilinear pairings, which is consistent with the standard models of SCF-PEKS [16], SPEKS [13], SCF-PEPCKS [24] and Hwang *et.al.* [21]. We embed keyword ciphertext with random number to ensure the uncertainty of keyword ciphertext. Compared with [3], [13], [16], [21] and [24], our proposed scheme has some good properties, such as no secure channel, no key escrow and no designated server. In addition, compared with the experimental simulation of [16], [21] and [24], our proposed scheme is more efficient and has shorter communication size. The comparison results demonstrate that our scheme is suitable for the deployment of practical applications.

2) Without random oracle model, we construct an efficient secure channel free public key encryption with keyword search scheme. We prove our scheme is able to resist outside

online keyword guessing attacks and inside offline keyword guessing without random oracle by ensuring keyword ciphertext indistinguishability security under adaptive chosen keyword attacks and keyword trapdoor indistinguishability security under adaptive chosen keyword attack.

The proof of our scheme meets the following requirements: i) the complex assumption of the Discrete Logarithm (DL) [26] is required to achieve keyword ciphertext indistinguishability security under adaptive chosen keyword attacks. ii) the complex assumption of the hash Diffie-Hellman (hDH) [27] is required to achieve keyword trapdoor indistinguishability security under adaptive chosen keyword attacks.

C. ORGANIZATION

In the second section, we provide the notation, problem assumption and definition of SCF-PEKS briefly. In the third section, we describe the existing keyword security problems in SCF-PEKS scheme. In the fourth section, we describe the definition and security model of our proposed scheme. In the fifth section, we show the description of the scheme. In the sixth section, we prove the scheme secure by game hopping method. Then, we show the scheme security properties comparison with other previous schemes. In the seventh section, we show our proposed scheme the comparison of computation efficiency and communication efficiency with other previous schemes.

II. PRELIMINARIES

In this section, we show notation and problem assumption, and then we describe definition of SCF-PEKS scheme polynomial time algorithm and definition of against keyword guessing attacks.

TABLE 1. Notations.

Symbol	Description
DO	Data Owner
DU	Data User
CSP	Cloud Service Provider
G, G_T	A cyclic group
g, P	A generator of G
Z_q^*	A prime order q
H, H_1	A secure one-way hash function
λ	A security parameter
KS_w	The keyword space
w	The keyword that the data owner sets from keyword space
w'	The keyword that the data user wants to search
pk_S, sk_S	The data owner's public/secret key pair
pk_R, sk_R	The data user's public/secret key pair
C_w	The ciphertext of keyword that the data owner produces
$T_{w'}$	The trapdoor which contains w'

A. NOTATIONS

Table 1 describes the symbols and description used in our paper.

B. PROBLEM ASSUMPTION

Definition 1: Let G be a cyclic group of prime order q with a generator g. Select $a \in Z_q^*$, for every arbitrary probability ε with a polynomial time t, there is an adversary A in solving DL [26] problem, if $\Pr[A(g, g^a) = a] < \varepsilon$.

Definition 2: Let G be a cyclic group of prime q and g be a generator of G. $H : G \to \{0, 1\}^l$ is a hash function mapping. Given hash function H and tetrad $(g, g^a, g^b, Z) \in G_3 \times \{0, 1\}^l$ where $a, b \in Z_q^*$ and l denotes the binary length of hash values. hDH [27] problem is to judge whether Z and $H(g^{ab})$ are equal.

Assuming that the hDH problem in cyclic group G is difficult, for every arbitrary probability ε with a polynomial time t, there is an adversary A in solving hDH problem, if $|\Pr[A(G, q, g, g^a, g^b, Z) = 1] - Pr[A(G, q, g, H, g^a, g^b, H(g^{ab})) = 1]| < \varepsilon$

C. DEFINITION OF SCF-PEKS SCHEME

Baek *et al.* [16] proposed a secure channel free public key encryption scheme with keyword search, which includes six polynomial time algorithms:

1) $GlobalSetup(\lambda)$: The global parameter generation algorithm takes a security parameter λ as input and outputs global parameter GP.

2) $KeyGen_{server}(GP)$: The server's key pair generation algorithm takes global parameter GP as input, and outputs a secret/public key pair (sk_S, pk_S) for the server.

3) $KeyGen_{receiver}(GP)$: The receiver's key pair generation algorithm takes global parameter GP as input, and outputs a secret/public key pair (sk_R, pk_R) for the receiver.

4) $Encrypt(GP, pk_S, pk_R, w) \to C_w$: The keyword encryption algorithm takes global parameter GP, a server's public key pk_S, a receiver's public key pk_R, and a keyword $w \in KS_w$ as input, and outputs a keyword ciphertext C_w.

5) $Trapdoor(GP, sk_R, pk_S, w') \to T_{w'}$: The keyword trapdoor generation algorithm takes global parameter GP, a server's public key pk_S, a receiver's secret key sk_R, and a search keyword w' as input, and outputs a keyword trapdoor $T_{w'}$.

6) $Test(GP, C_w, sk_S, T_{w'}) \to 0/1$: The test algorithm takes global parameter GP, a keyword ciphertext C_w, a server's secret key sk_S, a keyword trapdoor $T_{w'}$ as input, and outputs a symbol "1" if $w = w'$ or "0" otherwise.

Most of the existing PEKS and SCF-PEKS schemes are constructed based on bilinear pairings. Note that bilinear pairings use two cyclic groups G and G_T with prime order q and g is taken as a generator of G. We say that e is a map $G \times G$ to G_T, and the map e is a bilinear map if the following hold [3], [16].

(1) Bilinearity: $e(P^a, P^b) = e(P, P)^{ab}$, where $a, b \in Z_q^*$ and $P \in G$.

(2) Non-degeneracy: if g is a generator of G, then $e(g, g)$ is a generator of G_T,such that $e(g, g) \neq 1$.

(3) Computation: There is a polynomial time algorithm to compute $e(P, P)$,where $P \in G$.

D. ANALYSIS OF AGAINST KEYWORD GUESSING ATTACKS

There exist two types of attacker against keyword attack in SCF-PEKS scheme. One is the inside attacker, namely that is

the malicious server, and the other is the outside attacker. The attackers can intercept the keyword ciphertext and keyword trapdoor information when the user communicates with the server. However, the attacker cannot make a keyword guess because he doesn't have the secret key of both sides. The malicious server can calculate whether there is a keyword match between the keyword trapdoor and the keyword ciphertext by testing algorithm. Therefore, a malicious server has more authority than an outside attacker.

In this subsection, we will introduce three different keyword guessing attacks.

1) OUTSIDE OFFLINE KEYWORD GUESSING ATTACK [28]

This is performed by an outside attacker in the offline model. The vulnerabilities of keyword guessing attack come from the trapdoors which are simply generated by just combining keywords and secret key. In the outside offline keyword guess attack, the outside attacker can intercept keyword trapdoor information to guess. Secure channel free public key encryption with keyword search designed by Baek *et al.* [16] solved the security of trapdoor. Baek *et al.* [16] improved the scheme proposed by Boneh *et al.* [3] and solved the problem of keyword trapdoor. Rhee *et al.* [29] proposed that it can guarantee the privacy of keyword trapdoor, if the trapdoor is indistinguishable for the outside attack.

2) OUTSIDE ONLINE KEYWORD GUESSING ATTACK [4]

This is performed by an outside attacker in the online model. In the outside online keyword guessing attack, the outside attacker creates a collection of all possible keyword ciphertext, and then the attacker transfers the data ciphertext to the cloud server. The attacker then monitors the communication between the cloud server and the target receiver. Once it observes that the returned search results are related to the previously injected ciphertext, it knows the keyword information being searched by the target receiver.

3) INSIDE OFFLINE KEYWORD GUESSING ATTACK [28]

This is performed by an inside attacker in the offline model. The malicious cloud service provider traverses the keyword that set in offline mode and tries to find the keyword information in the keyword trapdoor. This attack is similar to an outside offline attack. Since the malicious cloud service provider stored a large number of keyword ciphertext, the malicious cloud service provider can do the test algorithm. It can also further discover which data ciphertext contains the same keyword. Hence, Huang and Li [19]. proposed a public key searchable encryption scheme to against inside keyword guessing attacks.

III. EXISTING KEYWORD SECURITY PROBLEMS IN SCF-PEKS SCHEME

In this section, we describe the existing keyword security problems in SCF-PEKS scheme both outside online keyword guessing attack and inside offline keyword guessing attack.

A. OUTSIDE ONLINE KEYWORD GUESSING ATTACK ON SCF-PEKS SCHEME

Step 1: The outside attacker identifies the specified receiver.

Step 2: The outside attacker prepare to upload the plaintext file $f_1, f_2, \cdots, f_n$ and the corresponding keywords $w_1, w_2, \cdots, w_n$. The outsider attacker uses the public key of the specified receiver and the server to generate keyword ciphertext matching the file ciphertext $< C_{f_1}, C_{w_1} >, < C_{f_2}, C_{w_2} >, \cdots, < C_{f_n}, C_{w_n} >$ through $Encrypt(GP, pk_S, pk_R, w)$ algorithm of SCF-PEKS. Finally, the outside attacker injects searchable file ciphertext to the cloud service provider.

Step 3: The outside attacker can inject keyword trapdoors because there is secure channel free property of SCF-PEKS scheme. Therefore, after the cloud service provider receives the search query from the specified receiver, it finds all the matching file ciphertext and returns the search results.

Step 4: The outside attackers monitor the communication channel between the specified receiver and the cloud service provider. If the returned result is observed to contain injected file ciphertext C_{f_i}, the outside attacker will confirm that the keyword trapdoor from the specified receiver involves attackers' keyword w_i. Therefore, the outside attacker guess correctly.

B. INSIDE OFFLINE KEYWORD GUESSING ATTACK ON SCF-PEKS SCHEME

In the above online keyword guessing attack, since the outside attacker does not know the key of the cloud service provider, the test algorithm cannot be used to verify its guess directly. However, by monitoring the communication between the cloud service provider and the specified receiver, it is easy to inject the keyword ciphertext into the cloud service provider to obtain the test results. Therefore, a data sender should not be able to distinguish between an uploaded ciphertext corresponding to a document encrypted by him and other uploaded encrypted documents. This property provides security against online keyword guessing attacks. Nevertheless, the improved scheme still has the problem of inside offline keyword guessing attacks. The malicious cloud service provider performs an inside offline keyword attack on the SCF-PEKS scheme, as follows:

Step 1: The malicious cloud service provider identifies the specified receiver. It can receive the keyword trapdoor T_w from the specified receiver.

Step 2: The malicious cloud service provider picks a keyword w'. By using the specified receiver's public key pk_R and its own public key pk_S, it perform the keyword encryption algorithm to compute the keyword ciphertext $C_{w'}$.

Step 3: The malicious cloud service provider runs the test algorithm SCF-PEKS by using its own key sk_S. Then it check whether the keyword ciphertext $C_{w'}$ and the trapdoor contain the same keyword T_w. If it does, then the malicious cloud service provider can guess correctly. Otherwise, the malicious cloud service provider returns Step 2 and continues to guess.

IV. OUR PROPOSED SCHEME

In this section, we describe our proposed scheme's definition and the security model.

A. DEFINITION OF OUR PROPOSED SCHEME

As shown by the against keyword guessing attacks in the previous section, most of the current SCF-PEKS scheme cannot against outside online keyword guessing attacks and inside offline keyword guessing attacks. Since the inside attacker can use the public key of the server and the data user to generate the keyword trapdoor, the inside attacker can do the test the algorithm to guess the keyword attack. In order to achieve the privacy of keyword search, we enhanced the security of SCF-PEKS scheme to prevent attackers without secret keys from generating keyword trapdoors. Therefore, the scheme has the ability of access control. It means the keyword ciphertext and keyword trapdoor has the property of unforgeability. We use a non-designated server for storage, making the scheme more flexible. We do not use bilinear pair operation, so the scheme is more efficient and more suitable for mobile terminals with limited communication capacity.

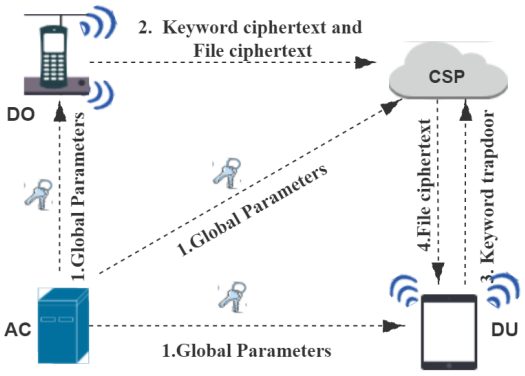


FIGURE 1. The schematic diagram of our proposed scheme.

The schematic diagram of the algorithm flow of our scheme is shown in Figure 1. In our scheme, we have four parties: Authentication Center (AC), Data Owner (DO), Data User (DU) and Cloud Service Provider (CSP). AC take charge running the global setup algorithm to distribute global parameters. DO generate encrypted file with keyword ciphertext and upload them to CSP. Meanwhile, CSP mainly has the characteristics of storing and retrieving data. DU can then use his/her secret key to generate the trapdoor corresponding to keyword w', and enable CSP to retrieve all files that are associated with the keyword w'. After the search finishes, CSP returns the search results to DU. If it does, it will return the file ciphertext to DU, else return 0.

Our proposed scheme includes five polynomial time algorithms:

1) *GlobalSetup*(λ): The global parameter generation algorithm takes a security parameter λ as input and outputs global parameter GP.

2) *KeyGen*(GP): The data owner's key pair generation algorithm takes global parameter GP as input, and outputs a secret/public key pair (sk_S, pk_S) for DO. The data user's key pair generation algorithm takes global parameter GP as input, and outputs a secret/public key pair (sk_R, pk_R) for DU.

3) $Encrypt(GP, sk_S, pk_R, w) \rightarrow C_w$: The keyword encryption algorithm takes global parameter GP, a data owner's secret key sk_S, a data user's public key pk_R, and a keyword $w \in KS_w$ as input, and outputs a keyword ciphertext C_w.

4) $Trapdoor(GP, sk_R, pk_S, w') \rightarrow T_{w'}$: The keyword trapdoor generation algorithm takes global parameter GP, a data user's secret key sk_R, a data owner's public key pk_S, and a search keyword w' as input, and outputs a keyword trapdoor $T_{w'}$.

5) $Test(GP, C_w, T_{w'}) \rightarrow 0/1$: The test algorithm takes global parameter GP, a keyword ciphertext C_w, a keyword trapdoor $T_{w'}$ as input, and outputs a symbol "1" if $w = w'$ or "0" otherwise.

B. SECURITY MODEL

The proposed scheme ought to ensure both keyword ciphertext indistinguishable security under adaptive chosen keyword attacks (CIND-CKA) and keyword trapdoor indistinguishable security under adaptive chosen keyword attacks (TIND-CKA) if the scheme is able to against outside online keyword guessing attacks and inside offline keyword guessing attacks. Generally, we divide our adversaries into inside and outside attackers. The inside attackers usually refer to semi trusted CSP, while the outside attackers usually refer to attackers other than DO, DU and CSP. Combining the two types of adversaries and security, we present two games definitions. The security CIND-CKA game and TIND-CKA game between the attacker and challenger are as follow:

1) CIND-CKA GAME

Assume A_1 is malicious server or outside attacker, and B is a challenger.

Setup: The challenger B takes security parameters λ as input, and outputs global parameters $GP = (G, q, g, H, H_1, KS_w)$, a DO's secret/public key pair (sk_S, pk_S) and DU's secret/public key pair (sk_R, pk_R) through the *GlobalSetup*(λ) and *KeyGen*(GP) algorithm. The challenger B sends the public key pk_S, pk_R and global parameters GP to the attacker A_1.

Phase 1: The attacker A_1 makes a series of query to the challenger B adaptively. The oracles are simulated by challenger B as follow:

a. Ciphertext Query: The challenger B responds to the ciphertext query C_w for the attacker A_1 for a keyword w through $Encrypt(GP, pk_R, sk_S, w)$ algorithm.

b. Trapdoor Query: The challenger B responds to the trapdoor query $T_{w'}$ for the attacker A_1 for a keyword w' through $Trapdoor(GP, pk_R, sk_S, w')$ algorithm.

c. Test Query: The challenger B responds to the test query for the attacker A_1 for keywords ciphertext C_w and trapdoor $T_{w'}$. The test query simulate an attack in which the

attacker A_1 verifies that a keyword ciphertext matches a keyword trapdoor by executing $Test(GP, C_w, T_{w'})$ algorithm or using CSP.

Challenge: The attacker A_1 chooses two keywords w_0 and w_1, which he/she has not asked for the cipertext before, and sends w_0 and w_1 to the challenger B. The challenger B picks $b \in \{0, 1\}$ to compute a keyword ciphertext C_{w_b} randomly, and then returns C_{w_b} to attacker A_1.

Phase 2: The attacker A_1 continues to make a series of queries to the challenger B adaptively, but with the restrictions that A_1 is disallowed to query the keyword ciphertext or trapdoor of either w_0 or w_1.

Guess: The attacker A_1 outputs $b' \in \{0, 1\}$. If $b = b'$, the attacker A_1 wins CIND-CKA game.

We define that the attacker A_1 has the advantage of winning CIND-CKA game, if

$$Adv(\lambda)_{CIND-CKA} = |\Pr[b = b'] - 1/2|.$$

Definition 3 (Security of CIND-CKA): We say that a CIND-CKA scheme satisfies the requirement of security if the advantage is negligible for an attacker A_1 to win CIND-CKA game in polynomial time.

2) TIND-CKA GAME

Assume A_2 is malicious server or outside attacker, and B is a challenger.

Setup: The challenger B takes security parameters λ as input, and outputs global parameters $GP = (G, q, g, H, H_1, KS_w)$, a DO's secret/public key pair (sk_S, pk_S) and DU's secret/public key pair (sk_R, pk_R) through the $GlobalSetup(\lambda)$ and $KeyGen(GP)$ algorithm. The challenger B sends the public key pk_S, pk_R and global parameters GP to the attacker A_2.

Phase 1: The attacker A_2 makes a series of query to the challenger B adaptively. The oracles are simulated by challenger B as follow:

a. Ciphertext Query: The challenger B responds to the ciphertext query C_w for the attacker A_2 for a keyword w through $Encrypt(GP, pk_R, sk_S, w)$ algorithm.

b. Trapdoor Query: The challenger B responds to the trapdoor query $T_{w'}$ for the attacker A_2 for a keyword w' through $Trapdoor(GP, pk_R, sk_S, w')$ algorithm.

c. Test Query: The challenger B responds to the test query for the attacker A_2 for keywords ciphertext C_w and trapdoor $T_{w'}$. The test query simulate an attack in which the attacker A_2 verifies that a keyword ciphertext matches a keyword trapdoor by executing $Test(GP, C_w, T_{w'})$ algorithm or using CSP.

Challenge: The attacker A_2 chooses two keywords w_0 and w_1, which he/she has not asked for the trapdoor before, and sends w_0 and w_1 to the challenger B. The challenger B picks $b \in \{0, 1\}$ to compute a keyword trapdoor T_{w_b} randomly, and then returns T_{w_b} to attacker A_2.

Phase 2: The attacker A_2 continues to make a series of queries to the challenger B adaptively, but with the restrictions that the attacker A_2 is disallowed to query the keyword ciphertext or trapdoor of either w_0 or w_1.

Guess: The attacker A_2 outputs $b' \in \{0, 1\}$. If $b = b'$, the attacker A_2 wins TIND-CKA game.

We define that the attacker A_2 has the advantage of winning the TIND-CKA game, if

$$Adv(\lambda)_{TIND-CKA} = |\Pr[b = b'] - 1/2|.$$

Definition 4 (Security of TIND-CKA): We say that a TIND-CKA scheme satisfies the requirement of security if the advantage is negligible for an attacker A_2 to win TIND-CKA game in polynomial time.

V. DESCRIPTION OF OUR PROPOSED SCHEME

Our proposed scheme consists of the following algorithm:

1) $GlobalSetup(\lambda)$: The globalsetup algorithm is run by AC, and takes a security parameter λ as input. The algorithm generates G to be a cyclic group of prime order q with a generator g. It select two secure hash function $H : \{0, 1\}^* \to {Z_q}^*$ and $H_1 : G \to \{0, 1\}^l$, where l denotes the binary length of hash values. Finally, it outputs the global parameters $GP = (G, q, g, H, H_1, KS_w)$, where KS_w denotes a keyword space.

2) $KeyGen(GP)$: DO randomly selects $sk_{S_1}, sk_{S_2} \in {Z_q}^*$ as the secret key $sk_S = (sk_{S_1}, sk_{S_2})$. The keygen algorithm takes the global parameters GP and DO's secret key $sk_S = (sk_{S_1}, sk_{S_2})$ as input, and then computes the public key $pk_S = (pk_{S_1}, pk_{S_2}) = (g^{sk_{S_1}}, g^{sk_{S_2}})$. Returns a DO's secret/public key pair (sk_S, pk_S).

DU randomly selects $sk_{R_1}, sk_{R_2} \in {Z_q}^*$ as the secret key $sk_R = (sk_{R_1}, sk_{R_2})$, where $\gcd(sk_{R_2}, q - 1) = 1$. The keygen algorithm takes the global parameters GP and DU's secret key $sk_R = (sk_{R_1}, sk_{R_2})$ as input, and then computes the public key $pk_R = (pk_{R_1}, pk_{R_2}) = (g^{sk_{R_1}}, g^{sk_{R_2}})$. Returns a DU's secret/public key pair (sk_R, pk_R).

3) $Encrypt(GP, w, sk_{S_1}, pk_{R_1}, pk_{R_2}, r) \to C_w$: DO takes the global parameters GP, a DO's secret key sk_{S_1}, a DU's public key pair $pk_R = (pk_{R_1}, pk_{R_2})$ and a keyword $w \in KS_w$ as input, and then computes $C_w = (U, V) = (pk_{R_2}^r, H_1(g^{r \cdot H(w||ss)}))$, where $ss = H_1((pk_{R_1})^{sk_{S_1}})$. Returns a keyword ciphertext C_w.

4) $Trapdoor(GP, w', sk_{R_1}, sk_{R_2}, pk_{S_1}) \to T_{w'}$: DU takes the global parameters GP, a DO's public key pk_{S_1}, a DU's secret key pair $sk_R = (sk_{R_1}, sk_{R_2})$ and a keyword $w' \in KS_w$ as input, and then computes $T_{w'} = (sk_{R_2})^{-1} \cdot H(w'||ss^*)$ where $ss^* = H_1((pk_{S_1})^{sk_{R_1}})$. Returns a keyword trapdoor $T_{w'}$.

5) $Test(GP, C_w, T_{w'}) \to 0/1$: CSP takes the global parameters GP, a keyword ciphertext C_w, a keyword trapdoor $T_{w'}$ as input. The test algorithm checks whether the equation $H_1(U^{T_{w'}}) = V$ holds. If it does, output 1; else, output 0.

According to the specifications of the above algorithms, we have

$$\begin{aligned} ss &= H_1((pk_{R_1})^{sk_{S_1}}) = H_1(g^{sk_{R_1} \cdot sk_{S_1}}) \\ &= H_1((pk_{S_1})^{sk_{R_1}}) = ss^* \\ H_1(U^{T_{w'}}) &= H_1((pk_{R_2}^r)^{(sk_{R_2})^{-1} \cdot H(w'||ss^*)}) \end{aligned}$$

$$= H_1((g^{sk_{R_2}\cdot r})^{(sk_{R_2})^{-1}\cdot H(w'||ss^*)})$$
$$= H_1((g^{r\cdot H(w'||ss^*)})$$

Because $ss = ss^*$,we have that the equation $H_1(U^{T_{w'}}) = V$ holds, if $w = w'$. Therefore, our proposed scheme is correct.

VI. SECURE ANALYSIS

In this section, we provide the security proof of our proposed scheme.

A. SECURITY PROOF

We analysis the security of our proposed scheme by using game hopping [25] proof method. Formally, we have the following lemma.

Lemma 1 (Difference Lemma) [25]: Let E be some ''error event'' such that $S_1|\neg E$ occurs if and only if $S_2|\neg E$ occurs. Then

$$|\Pr[S_1] - \Pr[S_2]| \leq \Pr[E].$$

The following are our security statements and the proofs.

Theorem 1: The above CIND-CKA game is secure without random oracle model assuming that H is a collision resistance hash function, and that the DL problem is intractable.

Proof: Assume that A_1 is a polynomial-time attacker against the security of the proposed TIND-CKA game, A_H is a collision resistance hash function attacker and that A_{hDH} is to break the DL problem attacker.

The theorem can be proven by consisting of five games as sub-game programs $Game_i (i = 1, 2, 3, 4, 5)$ with the attacker A_1. We define the attacker A_1 to guess the correct event in the $Game_i$ as X_i, that is $b = b_i$. The attacker will terminate with some final output, which will then be assessed to see if the attacker ''won''. Game-hopping is as follows:

$Game_1$: This game is the original attack CIND-CKA game, so the probability of A_1 guessing correctly is $Adv(\lambda)_{A_1} = |\Pr[X_1] - 1/2|$.

$Game_2$: B randomly picks $a, sk_{R_1}, sk_{R_2} \in Z_q{}^*$ to compute $g' = g^a$ and $pk_R = (g^{sk_{R_1}}, (g^a)^{sk_{R_2}})$, where g is the generator of group G. Other parameters is the same as $Game_1$. Obviously, $Game_2$ and $Game_1$ are indistinguishable from A_1. So, the probability of A_1 guessing correctly is equal, if $Pr[X_1] = \Pr[X_2]$.

$Game_3$: The game is the same as $Game_2$, except that B changes the way he/she to responds to A_1 for the ciphertext query, trapdoor query, test query and challenge. And B as oracle to responds the ciphertext query, trapdoor query and test query as follow:

Ciphertext Query: A_1 makes a ciphertext query with keyword $w \in KS_w$. B picks a random number $r \in Z_q{}^*$ and returns keyword ciphertext $C_w = (U, V)$ to A_1, where $U = (pk_{R_2})^r$, $V = H_1(g^{rH(w||ss)})$ and $ss = H_1((pk_{R_1})^{sk_{S_1}})$.

Trapdoor Query: A_1 makes a trapdoor query with keyword $w' \in KS_w$. B returns trapdoor $T_{w'} = (sk_{R_2})^{-1} \cdot (H(w'||ss^*))$, where $ss^* = H_1((pk_{S_1})^{sk_{R_1}})$.

Test Query: A_1 makes a test query with keyword ciphertext C_w and a keyword trapdoor $T_{w'}$. B returns 1 if $H_1(U^{T_{w'}}) = V$ or 0 otherwise.

Challenge: A_1 sends two keywords w_0 and w_1 to B, where $w_0 \neq w_1$ that he/she has not challenged before. B chooses $r^* \in Z_q{}^*$ and $b \in \{0, 1\}$ randomly for a keyword ciphertext $C_{w_b} = (U^*, V^*)$ where $U^* = (pk_{R_2})^{r^*}$, $V^* = H_1((g')^{H(w_b||ss)\cdot r^*})$, $ss = H_1((pk_{R_1})^{sk_{S_1}})$. And then returns them to A_1.

If we make $r' = r^*/a$, then

$$U^* = ((g^a)^{sk_{R2}})^{r^*/a} = (pk_{R2})^{r'}$$
$$V^* = H_1((g^a)^{H(w_b||ss)\cdot(r^*/a)}) = H_1(g^{H(w_b||ss)\cdot r'}).$$

Therefore, the challenge ciphertext $C_{w_b} = (U^*, V^*)$ is the effective ciphertext of the keyword w_b.

In the above game, if B is able to respond kinds of queries and challenge correctly, $Game_2$ and $Game_3$ will be indistinguishable to A_1. So, the attacker A_1 has the same probability of guessing correctly in both $Game_2$ and $Game_3$, if $Pr[X_2] = \Pr[X_3]$.

$Game_4$: The game is the same as $Game_3$, except that B will terminate the game, if it have any of the following events occur.

Event E_1*:* A_1 makes a ciphertext query to B, including the keyword's input satisfies $w \neq w_b$, but $V = V^*$.

Event E_2*:* A_1 makes a trapdoor query to B, including the keyword's input satisfies $w \neq w_b$, but $H(w||ss^*) = H(w_b||ss^*)$.

Obviously, $Game_3$ and $Game_4$ are indistinguishable to the attacker unless the event $E_1 \vee E_2$ occurs. According to Difference Lemma, we have

$$|\Pr[X_3] - \Pr[X_4]| \leq \Pr[E_1 \vee E_2].$$

In addition, if E_1 occurs, there must be an attacker A_H against the collision-resistant of hash function H. Hence, A_H has the advantage of winning, if $Adv(\lambda)_{A_H} \geq \Pr[E_1]$.

Similarly, if E_2 occurs, there must be an attacker A_H against the collision-resistant of hash function H. Hence, A_H has the advantage of winning, if $Adv(\lambda)_{A_H} \geq \Pr[E_2]$.

Therefore, we have

$$|\Pr[X_3] - \Pr[X_4]| \leq \Pr[E_1 \vee E_2] \leq 2Adv(\lambda)_{A_H}.$$

$Game_5$: The game is the same as $Game_4$, except that B picks a random number $Z \in G$ to compute $V^* = H_1((Z)^{H(w_b||ss)\cdot r^*})$ instead of $V^* = H_1((g^a)^{H(w_b||ss)\cdot r^*})$ when computing the challenge ciphertext $C_{w_b} = (U^*, V^*)$. Obviously, B does not need to know the value of a to respond all the attacker's queries by using only Discrete Logarithm tuples (g, g^a, Z) in $Game_5$. Obviously, $Game_4$ and $Game_5$ are uniform, there is an attacker A_{DL} that can distinguish the values of Z and g^a by a non-negligible advantage, if the DL problem is addressed. Suppose the attacker A_{DL} has the advantage of winning $Game_5$, if

$$|\Pr[X_4] - \Pr[X_5]| \leq Adv(\lambda)_{A_{DL}}.$$

Since Z is a random value of group G, the probability of A_1 guessing correctly is $Pr[X_5] = 1/2$.

End the game-hopping and analyze A_1's advantage. We have

$$\begin{aligned} Adv(\lambda)_{A_1} &= |\Pr[X_1] - 1/2| \\ &\le |\Pr[X_1] - \Pr[X_2]| + |\Pr[X_2] - \Pr[X_3]| \\ &+ |\Pr[X_3] - \Pr[X_4]| + |\Pr[X_4] - \Pr[X_5]| \\ &+ |\Pr[X_5] - 1/2|. \end{aligned}$$

On the basis of the above games, we can conclude as follow:

$$Adv(\lambda)_{A_1} = 2Adv(\lambda)_{A_H} + Adv(\lambda)_{A_{DL}}.$$

$Adv(\lambda)_{A_H}$ and $Adv(\lambda)_{A_{DL}}$ are negligible, because the security proof achieves the collision resistance property of the hash function H and the DL problem is intractable.

Therefore, we can conclude that CIND-CKA game is secure.

Theorem 2: The above TIND-CKA game is secure without random oracle model assuming that H is a collision resistance hash function, and that the hDH problem is intractable.

Proof: Assume that A_2 is a polynomial-time attacker against the security of the proposed TIND-CKA game, A_H is a collision resistance hash function attacker and that A_{hDH} is to break the hDH problem attacker.

The theorem can be proven by consisting of five games as sub-game programs $Game_i (i = 1, 2, 3, 4, 5)$ with the attacker A_2. We define the attacker A_2 to guess the correct event in the $Game_i$ as X_i, that is $b = b_i$. The attacker will terminate with some final output, which will then be assessed to see if the attacker ''won''. Game-hopping is as follows:

$Game_1$: This game is the original attack TIND-CKA game, so the probability of the attacker A_2 guesses correctly is $Adv(\lambda)_{A_2} = |\Pr[X_1] - 1/2|$.

$Game_2$: B randomly picks $a, b, sk_{R_1}, sk_{R_2} \in {Z_q}^*$ to compute $pk_{S_2} = g^a$ and $pk_R = (g^b, g^{sk_{R_2}})$, where g is the generator of group G. Other parameters is the same as $Game_1$. Obviously, $Game_1$ and $Game_2$ are indistinguishable from A_2. So, the probability of A_2 guessing correctly is equal, if $\Pr[X_1] = \Pr[X_2]$.

$Game_3$: This game is the same as $Game_2$, except that B changes the way he/she to responds to A_2 for ciphertext query, trapdoor query, test query and challenge. And B as oracle to responds the ciphertext query, trapdoor query and test query as follow:

Ciphertext Query: A_2 makes a ciphertext query with keyword $w \in KS_w$. B picks a random number $r \in {Z_q}^*$ and returns keyword ciphertext $C_w = (U, V)$ to A_2, where $U = (pk_{R2})^r$, $V = H_1(g^{rH(w||ss_1)})$ and $ss = H_1(g^{ab})$.

Trapdoor Query: A_2 makes a trapdoor query with keyword $w' \in KS_w$. B returns trapdoor $T_{w'} = (sk_{R_2})^{-1} \cdot (H(w'||ss^*))$, where $ss^* = H_1(g^{ab})$.

Test Query: A_2 makes a test query with keyword ciphertext C_w and a keyword trapdoor $T_{w'}$. B returns 1 if $H_1(U^{T_{w'}}) = V$ or 0 otherwise.

Challenge: A_2 sends two keywords w_0 and w_1 to B, where $w_0 \neq w_1$ that he/she has not challenged before. B chooses $b \in \{0, 1\}$ randomly for a keyword trapdoor $T_{w_b'} = (sk_{R_2})^{-1} \cdot (H(w'||ss^*))$, where $ss^* = H_1(g^{ab})$. And then returns them to A_2.

Obviously, the challenge trapdoor is the effective trapdoor of the keyword w_b'.

In the above game, if B is able to respond kinds of queries and challenge correctly, $Game_2$ and $Game_3$ will be indistinguishable to A_2. So, A_2 has the same probability of guessing correctly in both $Game_2$ and $Game_3$, if $\Pr[X_2] = \Pr[X_3]$.

$Game_4$: This game is the same as $Game_3$, except that B will terminates the game, if it have any of the following events occur.

Event E_1*:* A_2 makes a ciphertext query to B, including the keyword's input satisfies $w \neq w_b$, but $V = V^*$.

Event E_2*:* A_2 makes a trapdoor query to B, including the keyword's input satisfies $w \neq w_b$, but $H(w||ss^*) = H(w_b||ss^*)$.

Obviously, $Game_3$ and $Game_4$ are indistinguishable to the attacker unless the event $E_1 \vee E_2$ occurs. According to Difference Lemma, we have

$$|\Pr[X_3] - \Pr[X_4]| \le \Pr[E_1 \vee E_2].$$

In addition, if E_1 occurs, there must be an attacker A_H against the collision-resistant of hash function H. Hence, A_H has the advantage of winning, if $Adv(\lambda)_{A_H} \ge \Pr[E_1]$.

Similarly, if E_2 occurs, there must be an attacker A_H against the collision-resistant of hash function H. Hence, A_H has the advantage of winning, if $Adv(\lambda)_{A_H} \ge \Pr[E_2]$.

Therefore, we have

$$|\Pr[X_3] - \Pr[X_4]| \le \Pr[E_1 \vee E_2] \le 2Adv(\lambda)_{A_H}.$$

$Game_5$: The game is the same as $Game_4$, except that B picks a random number $Z \in \{0, 1\}^l$ instead of $H_1(g^{ab})$ when responding the challenge of trapdoor, ciphertext query and trapdoor query. Obviously, B does not need to know the value of a and b to respond all the attacker's queries and trapdoor challenge by using only hDH tuples (H_1, g, g^a, g^b, Z) in $Game_5$. Obviously, $Game_4$ and $Game_5$ are identical, there is an attacker A_{hDH} that can distinguish the values of $ss^* = H_1(g^{ab})$ and Z by a non-negligible advantage, if the hDH problem is addressed. Suppose the attacker A_{hDH} has the advantage of winning $Game_5$, if

$$|\Pr[X_4] - \Pr[X_5]| \le Adv(\lambda)_{A_{hDH}}.$$

Since Z is a random value of group G, the probability of A_2 guessing correctly is $Pr[X_5] = 1/2$.

End the game-hopping and analyze A_2's advantage. We have

$$\begin{aligned} Adv(\lambda)_{A_2} &= |\Pr[X_1] - 1/2| \\ &\le |\Pr[X_1] - \Pr[X_2]| + |\Pr[X_2] - \Pr[X_3]| \\ &+ |\Pr[X_3] - \Pr[X_4]| + |\Pr[X_4] - \Pr[X_5]| \\ &+ |\Pr[X_5] - 1/2|. \end{aligned}$$

TABLE 2. Security properties comparison.

Compared scheme	Secure channel	No key escrow	Designated server	Bilinear pair operation	OUT-ON-KGA[a]	IN-OFF-KGA[b]
PEKS [3]	yes	yes	no	yes	insecure	insecure
SCF-PEKS [16]	no	yes	yes	yes	insecure	insecure
SPEKS [13]	no	yes	no	yes	secure	insecure
SCF-PEPCKS [24]	no	yes	no	yes	secure	secure
Hwang et al. [21]	no	yes	no	no	secure	insecure
Ours	no	yes	no	no	secure	secure

[a] "OUT-ON-KGA" denotes if the scheme withstand the outside online keyword guessing attack.
[b] "IN-OFF-KGA" denotes if the scheme withstand the inside offline keyword guessing attack.

TABLE 3. Efficiency comparison.

Compared scheme	WRO model[a]	Computational Efficiency		Communication Efficiency	
		Encrypt	Test	Ciphertext size	Trapdoor size
SCF-PEKS [16]	no	$\tau_b + 2\tau_e + \tau_h$	$\tau_b + 2\tau_e$	$\|G\| + \lambda$	$\|G\|$
SCF-PEPCKS [24]	yes	$4\tau_e + 2\tau_h$	$\tau_b + \tau_e$	$\|G\| + \lambda$	$\|G\|$
Hwang et al. [21]	yes	$2\tau_e + \tau_h$	$5\tau_e + \tau_h$	$3\|G\|$	$4\|G\|$
Ours	yes	$3\tau_e + 2\tau_h$	τ_e	$2\|G\|$	$\|Z_q{}^*\|$

[a] "WRO model" denotes Without Random Oracle model.

On the basis of the above games, we can conclude as follow:

$$Adv(\lambda)_{A_2} = 2Adv(\lambda)_{A_H} + Adv(\lambda)_{A_{hDH}}.$$

$Adv(\lambda)_{A_H}$ and $Adv(\lambda)_{A_{hDH}}$ are negligible, because the security proof achieves the collision resistance property of the hash function H and the hDH problem is intractable.

Therefore, we can conclude that TIND-CKA game is secure.

B. SECURITY PROPERTIES ANALYSIS

We compare the security properties of our scheme with other previous schemes, as shown in Table 2. The security properties comparison includes secure channel, no key escrow, designated server and bilinear pair operation, while provide against outside online keyword guessing attacks, against inside offline keyword guessing attacks.

VII. EFFICIENCY ANALYSIS

In this section, we present a comparison of the efficiency and communication of our scheme with other schemes, including SCF-PEKS [16], SCF-PEPCKS [24], Hwang *et al.* [21]. The details are shown in Table 3. The symbols τ_b, τ_e and τ_h stand for the running time for a bilinear pair operation, an exponential operation in the group and one time hash operation in group G or G_T. Respectively, and their coefficients represent the times of such operation. The symbols $|G|$, $|Z_q{}^*|$ and λ represent the size of the elements in group G, the size of the element $Z_q{}^*$, and the size of the hash. We use the time operation to reflect the computational efficiency of the algorithm. For example, our scheme need to calculate three exponentiations in G and two hash functions operation to encrypts a keyword. So the time cost of our scheme is $3\tau_e + 2\tau_h$. In the communicational efficiency comparison, the keyword size length is a measure of the length of the keyword variable output after encryption. For instance, our scheme contains a group element in G. So, a keyword ciphertext size is $2|G|$.

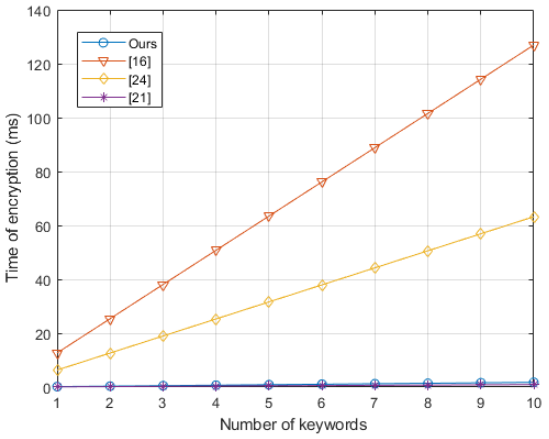


FIGURE 2. Computation cost of keyword encryption.

We implemented our proposed scheme on a Lenovo PRODUCT that runs windows 10 (64bit) with Inter(R) CoreTM i5-3470s CPU @2.9GHz and 8GB RAM memory by employing the gmpy2 (Encapsulation of The GNU Multiple Precision Library [30]) module. In order to achieve the security attributes almost consistent with the [24] schemes, we instantiate 512-bit group size and the general cryptographic hash function is respectively instantiated by SHA-256. The experimental results are shown by Figure 2 to Figure 5. Besides, the time consumption and communication

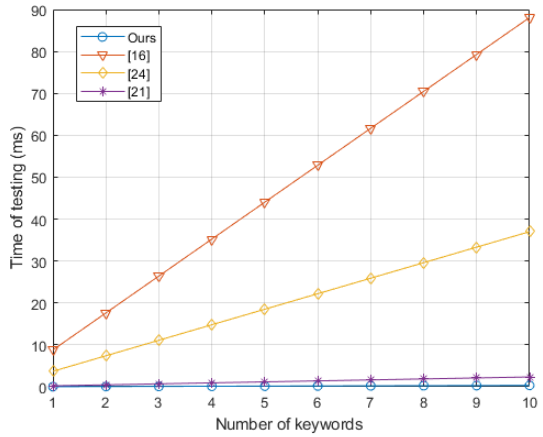


FIGURE 3. Computation cost of testing.

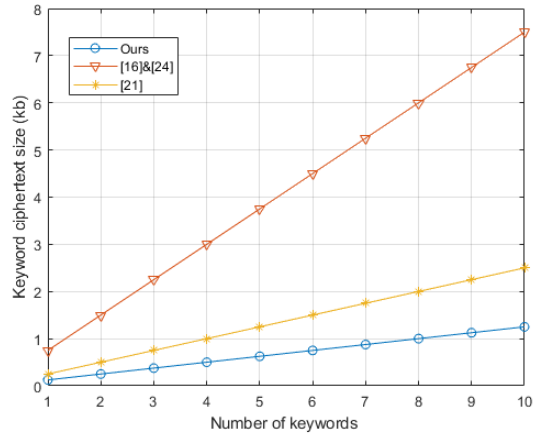


FIGURE 4. Communication cost of keyword ciphertext.

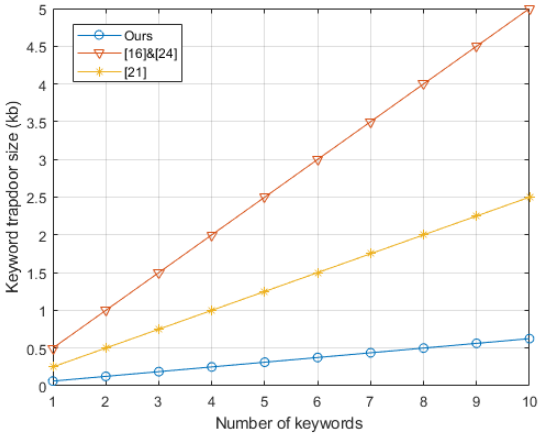


FIGURE 5. Communication cost of keyword trapdoor.

size data of SCF-PEKS and SCF-PEKSCKS schemes are provided by [24].

We come up with our scheme outperforms the SCF-PEKS schemes [16], SCF-PEPCKS [24] and Hwang *et.al.* [21] in both keyword encryption and testing. As illustrated in Figure 2, the time of a single keyword ciphertext in our scheme is about 0.177ms, while that in the schemes [16], [24], and [21] is about 12.693ms, 6.322ms, 0.094ms. In addition, the time consumption of testing in our scheme is about 0.033ms, while that in the schemes [16], [24] and [21] is about 8.809ms, 3.701ms, 0.235ms.

For the communication cost, a keyword ciphertext in our scheme has 0.125kb, while a keyword trapdoor has 0.0625kb. Therefore, the scheme does not need a large storage space to store keyword ciphertext and keyword trapdoor. As shown by Figure 4 and Figure 5, the communication consumption of our scheme is better than that of schemes in [16], [24] and [21]. According to the experimental results, we concluded that our scheme has more practical application significance than [16], [24] and [21] schemes in mobile smart terminals with limited communication and computing power.

VIII. CONCLUSION

In this paper, we present an efficient public key searchable encryption without bilinear pair operation for mobile smart terminal. Our proposed scheme has good security properties, high computational efficiency and low storage space. We prove our scheme is capable of resisting both inside offline keyword guessing attacks and outside online keyword guessing without random oracle model by satisfying keyword ciphertext indistinguishability security under adaptive chosen keyword attacks and keyword trapdoor indistinguishability security under adaptive chosen keyword attacks. The experimental results and comparisons show that it is feasible. These have practical significance for the application of mobile smart terminal.

REFERENCES

[1] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 1–51, Jan. 2015, doi: 10.1145/2636328.

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

[3] D. Boneh, G. D. 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] W.-C. Yau, R. C.-W. Phan, S.-H. Heng, and B.-M. Goi, "Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester," *Int. J. Comput. Math.*, vol. 90, no. 12, pp. 2581–2587, Dec. 2013, doi: 10.1080/00207160.2013.778985.

[5] L. Ibraimi, S. Nikova, P. Hartel, and W. Jonker, "Public-key encryption with delegated search," in *Proc. Appl. Cryptogr. Netw. Secur.*, Nerja, Spain, Jun. 2011, pp. 532–549.

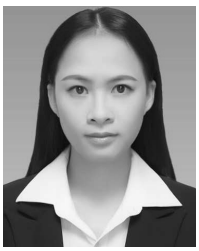
[6] M. Ma, D. He, M. K. Khan, and J. Chen, "Certificateless searchable public key encryption scheme for mobile healthcare system," *Comput. Electr. Eng.*, vol. 65, pp. 413–424, Jan. 2018, doi: 10.1016/j.compeleceng.2017.05.014.

[7] J.-H. Park, J.-A. Seol, and Y.-H. Oh, "Design and implementation of an effective mobile healthcare system using mobile and RFID technology," in *Proc. 7th Int. Workshop Enterprise Netw. Comput. Healthcare Ind. (HEALTHCOM)*, Busan, South Korea, 2005, pp. 263–266.

[8] E. Uwizeye, J. Wang, Z. Cheng, and F. Li, "Certificateless public key encryption with conjunctive keyword search and its application to cloud-based reliable smart grid system," *Ann. Telecommun.*, vol. 74, nos. 7–8, pp. 435–449, Aug. 2019, doi: 10.1007/s12243-019-00716-8.

[9] L. Wu, B. Chen, K.-K.-R. Choo, and D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things," *J. Parallel Distrib. Comput.*, vol. 111, pp. 152–161, Jan. 2018, doi: 10.1016/j.jpdc.2017.08.007.
[10] M. Ma, D. He, N. 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, Feb. 2018, doi: 10.1109/TII.2017.2703922.
[11] 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. 2018, doi: 10.1109/TII.2017.2771382.
[12] X. Zhang, C. Xu, H. Wang, Y. Zhang, and S. Wang, "FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2019.2914117.
[13] Y.-C. Chen, "SPEKS: Secure server-designation public key encryption with keyword search against keyword guessing attacks," *Comput. J.*, vol. 58, no. 4, pp. 922–933, Apr. 2015, doi: 10.1093/comjnl/bxu013.
[14] T. Suzuki, K. Emura, and T. Ohigashi, "A generic construction of integrated secure-channel free PEKS and PKE and its application to EMRs in cloud storage," *J. Med. Syst.*, vol. 43, no. 5, pp. 1–15, May 2019, doi: 10.1007/s10916-019-1244-2.
[15] Z.-Y. Shao and B. Yang, "On security against the server in designated tester public key encryption with keyword search," *Inf. Process. Lett.*, vol. 115, no. 12, pp. 957–961, Dec. 2015, doi: 10.1016/j.ipl.2015.07.006.
[16] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, Perugia, Italy, 2008, pp. 1249–1259.
[17] L. Fang, W. Susilo, C. Ge, and J. Wang, "A secure channel free public key encryption with keyword search scheme without random oracle," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Kanazawa, Japan, 2009, pp. 248–258.
[18] M. Noroozi and Z. Eslami, "Public-key encryption with keyword search: A generic construction secure against online and offline keyword guessing attacks," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 2, pp. 879–890, Feb. 2020, doi: 10.1007/s12652-019-01254-w.
[19] 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.
[20] S. H. Wang, Y. X. Zhang, H. Q. Wang, F. Xiao, and R. C. Wang, "Efficient public-key searchable encryption scheme against inside keyword guessing attack," *Comput. Sci.*, vol. 46, no. 7, pp. 91–95, Jul. 2019, doi: 10.11896/j.issn.1002-137X.2019.07.014.
[21] M.-S. Hwang, C.-C. Lee, and S.-T. Hsu, "An ElGamal-like secure channel free public key encryption with keyword search scheme," *Int. J. Found. Comput. Sci.*, vol. 30, no. 2, pp. 255–273, Feb. 2019, doi: 10.1142/S0129054119500047.
[22] H. L. Xu and Y. Lu, "Searchable public key encryption secure against keyword guessing attacks," *Comput. Eng. Appl.*, vol. 54, no. 24, pp. 108–115, 2018.
[23] Y. Lu, G. Wang, and J. Li, "Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement," *Inf. Sci.*, vol. 479, pp. 270–276, Apr. 2019, doi: 10.1016/j.ins.2018.12.004.
[24] Y. Lu, J. Li, and Y. Zhang, "SCF-PEPCKS: Secure channel free public key encryption with privacy-conserving keyword search," *IEEE Access*, vol. 7, pp. 40878–40892, Mar. 2019, doi: 10.1109/ACCESS.2019.2905554.
[25] A. W. Dent, "A note on game hopping proofs," Cryptol. ePrint Arch., Int. Assoc. Cryptologic Res., NV, USA, Tech. Rep. (2006/260), 2006. [Online]. Available: https://eprint.iacr.org/2006/260.pdf
[26] A. R. Sadeghi and M. Steiner, "Assumptions related to discrete logarithms: Why subtleties make a real difference," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Innsbruck, Austria, May 2001, pp. 244–261.
[27] M. Abdalla, M. Bellare, and P. Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES," in *Proc. Cryptographers Track at RSA Conf.*, San Francisco, CA, USA, Apr. 2001, pp. 143–158.
[28] J. W. Byun, H. S. Rhee, H. A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Proc. Workshop Secure Data Manage. (SDM)*, Seoul, South Korea, Sep. 2006, pp. 75–83.
[29] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *J. Syst. Softw.*, vol. 83, no. 5, pp. 763–771, May 2010, doi: 10.1016/j.jss.2009.11.726.
[30] *The GNU Multiple Precision Arithmetic LibraryInnsbruck*. Accessed: Dec. 1, 2019. [Online]. Available: https://gmplib.org/

NINGBIN YANG is currently pursuing the master's degree with the School of Mathematics and Information Sciences, Guangzhou University, China. His research interests include cryptography and cloud security.



SHUMEI XU is currently pursuing the master's degree with the School of Mathematics and Information Sciences, Guangzhou University, China. Her research interests mainly include authentication agreement, privacy protection, and smart grid.

ZHOU QUAN received the B.S. degree in computer science from South West Normal University, Chongqing, China, in 1996, the M.S. degree in applied mathematics from Guangzhou University, Guangzhou, China, in 2003, and the Ph.D. degree in agricultural water-soil engineering from South China Agricultural University, Guangzhou, in 2017.

Since 2009, he has been an Associate Professor of computer science and technology with Guangzhou University, where he is currently an Associate Professor with the School of Mathematics and Information Sciences. His research interests include cloud computing security, big data security, and wireless sensor networks security.