Hindawi
Security and Communication Networks
Volume 2018, Article ID 7254305, 7 pages
https://doi.org/10.1155/2018/7254305

Research Article

A Secure Ciphertext Retrieval Scheme against Insider KGAs for Mobile Devices in Cloud Storage

Run Xie,[1] Chanlian He,[2] Dongqing Xie,[3] Chongzhi Gao ,[3] and Xiaojun Zhang[4]

[1] *School of Mathematics, Yibin University, Yibin, China*
[2] *School of Computer and Information Engineering, Yibin University, Yibin, China*
[3] *School of Computer Science, Guangzhou University, Guangzhou, China*
[4] *School of Computer Science, Southwest Petroleum University, Chengdu, China*

Correspondence should be addressed to Chongzhi Gao; czgao@gzhu.edu.cn

Received 13 February 2018; Revised 24 April 2018; Accepted 10 May 2018; Published 6 June 2018

Academic Editor: Ilsun You

With the advent of cloud computing, data privacy has become one of critical security issues and attracted much attention as more and more mobile devices are relying on the services in cloud. To protect data privacy, users usually encrypt their sensitive data before uploading to cloud servers, which renders the data utilization to be difficult. The ciphertext retrieval is able to realize utilization over encrypted data and searchable public key encryption is an effective way in the construction of encrypted data retrieval. However, the previous related works have not paid much attention to the design of ciphertext retrieval schemes that are secure against inside keyword-guessing attacks (KGAs). In this paper, we first construct a new architecture to resist inside KGAs. Moreover we present an efficient ciphertext retrieval instance with a designated tester (dCRKS) based on the architecture. This instance is secure under the inside KGAs. Finally, security analysis and efficiency comparison show that the proposal is effective for the retrieval of encrypted data in cloud computing.

1. Introduction

With the development and deployment of cloud computing, more and more mobile devices are connected to the cloud and receiving services provided by the cloud servers. Cloud storage service is one of the most critical applications of cloud computing, which offers great convenience to users, including the storage of data and sharing of data [1]. However, the cloud service providers are usually cannot be completely trusted because they are managed and controlled by a third party such as Google or Amazon. Thus, the users have to take the risk that the cloud servers may find and leak their sensitive information.

To tackle the challenge of security in cloud storage, a lot of research has been performed to address the security and privacy issues in cloud computing, such as the cloud data access control [2–4], cloud data outsourcing computation [5–8], and privacy in data processing [9–13].

To protect data privacy, data are usually encrypted before they are uploaded to the cloud server. Nevertheless, this way generates the new obstacles that the cloud server is not able to carry out data retrieval over ciphertext data [14]. When users would like to access the part of encrypted data, they have to get entire data back or share the keys with the cloud server. As a result, users have to pay more for the bandwidth or give up their data privacy.

To overcome this obstacle, the concept of searchable encryption [15] was first proposed by Song et al. at 2000. Meanwhile, based on symmetric encryption, they proposed the keyword search scheme based on symmetric encryption, namely, Searchable Symmetric Encryption (abbreviation SSE). The searchable encryption permits users to retrieve a particular keyword over encrypted data by sending a trapdoor to the cloud server. However, the SSE involves detailed secret key management.

To overcome weakness and improve security, a new searchable encryption primitive was presented by Boneh and Boyen, which is called Public key Encryption with Keyword Search (PEKS in short) [16]. In their solution, with public information, the sender can encrypt the keyword associated with encrypted data and store its ciphertext to cloud server. To achieve keyword retrieval over the encrypted data, the receiver creates a trapdoor corresponding to the keyword, then he delivers the trapdoor to server. By the testing procedure, the server can find the ciphertext of keyword associated with the trapdoor. Then it sends corresponding data to receiver. Yet, there is the requirement of secure channel in their PEKS [16]. Usually, it is difficult to fulfill this requirement. This weakness limits the applications of their scheme. Following Boneh's PEKS, Baek et al. presented a new PEKS solution, which is called Secure Channel Free Public Key Encryption with Keyword Search [17] (denoted as SCF-PEKS). However, Baek's scheme was proved to be insecure by Yau et al. [18] with the following reasons. When the outside adversary acquires a trapdoor in channel, he can launch keyword-guessing attacks. This attack is called outside KGAs. So far, many of existing solutions concentrate on building up the security to resist outside KGAs [19–23]. Only a few schemes [24–28] are secure against this attacks.

Additionally, the most difficult issue is to resist keyword-guessing attacks launched by the cloud server, namely, inside KGAs. The inside KGAs are that the test server launched keyword-guessing attacks. Actually, such kind of attack also has been considered in deduplication system [30] and other security protocols [3, 31–33]. Specifically, the malicious server can create their ciphertext since the production of a PEKS ciphertext of keyword involves only public parameters. Given the trapdoor, the malicious cloud server can perform the test procedure with the guessing keyword. Therefore the server is able to know if the guessing keyword matches given trapdoor. Repeating guessing-then-testing process, the server can find the correct keyword. Because of the weakness of the small size of keyword space, this attack is available. Based on dual-server, Chen et al. [34] presented a new PEKS scheme which is considered to be secure against insider KGAs. However, in their solution, the front server can test whether the ciphertext of keyword relates to given trapdoor. Hence, under the original framework of [16], designing a secure scheme against inside KGAs is out of the question.

Recently, a new scheme [29] has been proposed by Jiang et al. based on slightly different architecture. With the aid of TTP (trusted third party), their solution can resist inside KGAs. In [29], TTP delivers its secret key to the sender from a safe channel. With his own secret key, the sender produces the legal ciphertext of keywords. Without the sender's secret key, the server is not able to generate a correct ciphertext of keyword as inputs of the test procedure. Hence, the server is not able to launch inside KGAs.

1.1. Our Contributions. In this paper, we present a new ciphertext retrieval system with a designated tester (dCRKS) based on the new security model. Compared with some analogous works, such as the cloud data retrieval schemes [35–37], the advantages of this system can be summarized as follows.

Firstly, we build security model of ciphertext retrieval system. Here, the server will not be considered as special attacker. So this model is more simple.

Second, we design an instance of dCRKS. This dCRKS instance can resist inside KGAs. In the instance, the server can not produce a correct ciphertext of keywords without the secret key of sender. Meanwhile the server can not generate a valid trapdoor without the secret key of receiver. Therefore, the malicious server is not able to launch inside KGAs. Most of the existing literatures (as [25, 28]) can not resist inside KGAs. Although the [29] is secure against inside KGAs, the TTP (trusted third party) is required in their scheme.

Thirdly, in this dCRKS instance, only a specified server is able to test whether given trapdoor relates to a dCRKS ciphertext. So, the proposal is stronger than [29].

Last, the analysis proves that the generation method of trapdoor and the testing algorithm are more effective than those of [29].

2. Preliminaries

Here, we will build the framework of dCRKS and its security model. Next, we introduce the hard assumptions which are used to prove the security of the instance of dCRKS system. In security model, let $\mathcal{F}$ be an adversary. The challenger denoted by $\mathcal{G}$. The dCRKS ciphertexts refer to the list of encrypted keywords.

2.1. Framework of dCRKS and Security Model

2.1.1. Framework of dCRKS. The dCRKS system is a ciphertext retrieval approach. In this system, only the specified server can carry out the testing procedure with the correct dCRKS ciphertext. The framework of dCRKS consists of the four algorithms. They are defined as follows.

Setup. Here n is the essential parameter. Let $\mathcal{PP}$ be the set public parameter. Inputting n, this algorithm outputs $\mathcal{PP}$.

KeyGen ($\mathcal{PP}$). When the $\mathcal{PP}$ are imported, this algorithm outputs (P_d, K_d), (P_r, K_r), and (P_s, K_s). The P_d (or K_d) is the sender's public (or private) key. Similarly, the receiver's public (private) is the P_r (K_r). The server's public (private) is the P_s (K_s).

EndCRKS ($\mathcal{PP}, K_d, P_d, P_r$). Let w be a keyword. Taking P_d, K_d, P_r, w, and the public parameter $\mathcal{PP}$ as input, this EndCRKS produces C_w corresponding to w, where C_w is dCRKS ciphertext.

dTrapdoor ($P_d, K_r, P_s, \mathcal{PP}$). When the keywords w', K_r, P_s, and $\mathcal{PP}$ are imported, this algorithm produces a trapdoor $T_{w'}$ corresponding to w'.

dTest ($K_s, C_w, T_{w'}, \mathcal{PP}$). In this algorithm, the server takes a trapdoor $T_{w'}$, $\mathcal{PP}$, a dCRKS ciphertext C_w, and its private key K_s as input. If $w' = w$, it replies "yes"; otherwise it replies "no".

2.1.2. Security Model. Here, we construct the security architecture of the dCRKS. The security of dCRKS ciphertext and trapdoor are defined by this architecture. The security of dCRKS is based on the two games.

In game 1, the adversary can be a malicious server or a malicious receiver or other attackers. So, the adversary can know the server's secret key or the receiver's secret key. The only limitation is that the adversary can not query the trapdoors corresponding to the challenge keywords w_0, w_1. The dCRKS ciphertext is secure. That means the adversary $\mathcal{F}$ is unable to differentiate between the dCRKS ciphertext of w_1 and the dCRKS ciphertext of keyword w_0 when the coupling trapdoor has not be obtained.

In game 2, the adversary can be a malicious server or a malicious sender or other attackers. Clearly, he can obtain the server's secret key or the receiver's secret key. The security of trapdoor requires that the $\mathcal{F}$ is unable to differentiate between a trapdoor of w_1 and a trapdoor of w_0 when the coupling trapdoor has not be obtained, where w_0 and w_1 are the challenge keywords.

The game 1 is described as follows.

Game 1. In this game, the $\mathcal{F}$ can enquire for private key and trapdoor. Yet $\mathcal{F}$ is not permitted to enquire the coupling trapdoors of the challenge keywords w_0, w_1, where both w_0 and w_1 are his choice. It requires that $\mathcal{F}$ differentiates between the dCRKS ciphertext of keyword w_1 and the dCRKS ciphertext of w_0. If the $\mathcal{F}$ can not win this game with nonnegligible probability, the dCRKS scheme is secure to resist the chosen keyword attacks.

Init. In this phase, $\mathcal{F}$ issues P_d as the challenge public key P_d.

Setup. Running the setup procedure, $\mathcal{G}$ generates the public parameters $\mathcal{PP}$ and gives the public parameters $\mathcal{PP}$ to adversary $\mathcal{F}$.

Phase 1. $\mathcal{F}$ performs repeatedly inquiries. The restriction is that the number of enquiries is no more than polynomially bounded.

Pk-Query (Private Key Query). For $d_i \neq d^*$, $\mathcal{F}$ sends P_{d_i} to $\mathcal{G}$, then $\mathcal{G}$ replies $\mathcal{F}$ with K_{d_i}.

T-Query (Trapdoor Query). $\mathcal{F}$ issues P_{r_i} and w_i to $\mathcal{G}$. $\mathcal{G}$ runs the trapdoor procedure and replies the trapdoor T_{w_i} for P_{r_i}, w_i to $\mathcal{F}$.

Challenge. $\mathcal{F}$ chooses the pair keywords (w_0, w_1) and P_{d^*} as the challenge keywords for P_{d^*}. The restriction is that the private key corresponding to P_{d^*} or the trapdoors corresponding to w_0 and w_1 have not been enquired by $\mathcal{F}$. $\mathcal{G}$ generates the challenge ciphertext $C^*_{w_b}$ and replies $C^*_{w_b}$ to $\mathcal{F}$, where $b \in \{0, 1\}$ is a random bit.

Phase 2. In this phase, $\mathcal{F}$ can still enquire the secret keys ($d \neq d^*$) and the trapdoors ($d \neq d^*$) or the trapdoor for $d = d^*$ with $w_i \neq w_0, w_1$. $\mathcal{G}$ replies as Phase 1.

Outputs. In the end, $\mathcal{F}$ guesses $b \in \{0, 1\}$. When $b' = b$, it means that $\mathcal{F}$ wins this game.

Game 2. Here, $\mathcal{F}$ can enquire for dCRKS ciphertext and secret key. Yet, $\mathcal{F}$ is not allowed to enquire the dCRKS ciphertexts corresponding to the challenge keywords w_0 and w_1, where both w_0 and w_1 are his choice. It requires that $\mathcal{F}$ differentiates between the trapdoor of keyword w_1 and the trapdoor of w_0. If the $\mathcal{F}$ can not win this game with nonnegligible advantage probability, the dCRKS system can resist the chosen keyword attacks.

Init. $\mathcal{F}$ issues the challenge public key P_r.

Setup. Running setup procedure, the $\mathcal{G}$ produces public parameters $\mathcal{PP}$ and delivers the parameters $\mathcal{PP}$ to $\mathcal{F}$.

Phase 1. $\mathcal{F}$ performs repeatedly inquiries. The restriction is that the number of inquiries is no more than polynomially bounded.

Pk-Query. $\mathcal{F}$ sends P_{r_i} to $\mathcal{G}$, $r_i \neq r^*$, and $\mathcal{G}$ responds K_{r_i} to $\mathcal{F}$ with r_i.

dc-Query (dCRKS Ciphertext Query). $\mathcal{F}$ sends P_{d_i} and w_i to $\mathcal{G}$. Running the EndCRKS procedure, $\mathcal{G}$ replies the ciphertext C_{w_i} for P_{d_i}, w_i to $\mathcal{F}$.

Challenge. $\mathcal{F}$ chooses a pair keywords w_0, w_1, and P_{r^*} as the challenge keywords for P_{r^*}. The restriction is that the secret key for P_{r^*} or the dCRKS ciphertext of w_0, w_1 for P_{r^*} has not been inquired by $\mathcal{F}$. $\mathcal{G}$ generates $T^*_{w_b}$ as challenge trapdoor and replies $T^*_{w_b}$ to $\mathcal{F}$, where $b \in \{0, 1\}$ is a random bit.

Phase 2. In this phase, $\mathcal{F}$ can still enquire the dCRKS ciphertexts and the secret key for $r \neq r^*$ or the dCRKS ciphertexts for $r = r^*$ and $w_i \neq w_0, w_1$. $\mathcal{G}$ replies as the first phase.

Outputs. In the end, $\mathcal{F}$ guesses $b' \in \{0, 1\}$. When $b' = b$, it means that $\mathcal{F}$ wins this game.

2.2. Complexity Assumptions

2.2.1. Bilinear Map. Let G and G_T be multiplicative cyclic groups with the order p (prime). Let $e : G \times G \rightarrow G_T$ be a bilinear map. e has the following properties:

(1) $g_1, g_2 \in G$ exist, such that $e(g_1, g_2)$ is not equal to 1_{G_T}.

(2) For all $\eta, \pi \in G$ and $a, t \in Z$, the $e(\eta^a, \pi^t) = e(\eta, \pi)^{at}$ is true.

(3) For all $\eta, \pi \in G$, the $e(\eta, \pi)$ can be calculated in polynomial time.

2.2.2. Complexity Assumptions. According to [38], the Computational Diffie-Hellman (CDH) problem is considered to be hard on G and G_T. Meanwhile, we know that the Decision Diffie-Hellman (DDH) problem [39] is hard on G_T.

In additional, to prove the security of dCRKS system, we need to introduce a new hard problem on G and G_T, namely, the Strong Decisional Diffie-Hellman assumption (in short SDDH). The SDDH problem is defined as follows.

The SDDH problem in (G, G_T) is as follows.

Given $\mathcal{L}' = (g, h, g^a, h^c, e(g, h)^{1/a}, e(g, g)^t, e(g, h)^q)$ as input, output "yes" if $q = ct/a$ and "no" otherwise, where a, $c, t, q \in Z^*_p$.

As is known, g^t is not able to infer from $e(g,g)^t$ because the e is considered to be one-way functions. Moreover, $e(g,h)^{ct/a}$ can not be calculated from g^a, h, and h^c. In fact, even the DDH problem is easy, the SDDH problem is seemingly still intractable.

Additionally, the DLP (discrete logarithm problem) is assumed to hold over G and G_T.

3. dCRKS against Insider Attacks

3.1. The Instance of dCRKS. Now, we will describe the instance of dCRKS. Here, the G and G_T are given groups as the previous definition. Let $H : \{0,1\}^* \rightarrow Z_p^*$ be the hash function, which is considered as random oracle in security model.

Setup. Let e be a bilinear map on G. Let p be the order of G and G_T, where both G and G_T are multiplicative cyclic group. This procedure produces $\mathscr{PP} = (g, G_T, G, p, H, e, h)$, where $\mathscr{PP}$ is the set of public parameters. The generator of G is g.

KeyGen ($\mathscr{PP}$). Takes as input x, y, z, g, and h, where x, y, $z \in Z_p^*$. This procedure generates key as the following way.

$K_d = z$ and $P_d = h^z$, where K_d and P_d are the sender's private key and public key, respectively.

$K_r = x$ and $P_r = (P_{r1}, P_{r2}) = (e(g,h)^{1/x}, g^x)$, where $K_r = x$ and P_r are the receiver's private key and public key.

$K_s = y$ and $P_s = g^y$, where $K_s = y$ and $P_s = g^y$ are the server's private key and public key.

EndCRKS (P_d, K_d, P_r, w). The sender takes a keyword w, P_r, P_d, K_d, $\mathscr{PP}$, and a random $u \in Z_p^*$ as input. This procedure produces $C = (C_1, C_2, C_3)$ as the dCRKS ciphertext of w output. C_1, C_2, and C_3 are calculated as follows:

$$C_3 = e(g,g)^u$$
$$C_2 = (P_{r2})^u \cdot g^{-uH(w)} = g^{(x-H(w))u}$$
$$C_1 = (P_{r1})^{zu} = e(g,h)^{zu/x}$$

dTrapdoor (P_d, K_r, P_s, w'). The receiver takes a keyword w', $r \in Z_p^*$, P_d, K_r, and P_s as inputs. This procedure produces $T = (T_1, T_2)$ as the trapdoor of w' output. T_1 and T_2 are calculated as follows:

$$T_1 = r;$$
$$T_2 = (P_d^{1/x} \cdot P_s^{-r})^{1/(x-H(w'))}.$$

dTest. Receiving a trapdoor T, the server runs dTest algorithm over the dCRKS ciphertexts with his private key y. Let C be a dCRKS ciphertext. The retrieving operation is executed by checking

$$e(C_2, T_2) C_3^{yT_1} = C_1 \qquad (1)$$

If the above equation is true, the algorithm returns 1; otherwise it returns 0.

3.2. Correctness of dCRKS. Now, we show that the above instance is correct. Let C be a dCRKS ciphertext which matches the trapdoor T. By the following equations, we can verify the correctness of dTest.

$$T_1 = r;$$
$$T_2 = (P_d^{1/x} \cdot P_s^{-r})^{1/(x-H(w'))};$$
$$T = (T_1, T_2);$$
$$C_2 = P_{r2}^u \cdot g^{-uH(w)};$$
$$e(C_2, T_2) = e(g^{(x-H(w))u}, (h^{z/x} g^{-yr})^{1/(x-H(w'))}).$$

When $w' = w$, we can obtain the equations

$$e(C_2, T_2) = e(g^u, h^{z/x} g^{-yr}) = e(g^u, h^{z/x}) e(g^u, g^{-yr})$$

As a result, the correctness of dTest is verified as follows:

$$C_1 = e(g,h)^{zu/x} = e(C_2, T_2) C_3^{yT_1}.$$

3.3. Security of dCRKS

3.3.1. Security of dCRKS Ciphertext. In this section, we demonstrate that the ciphertexts of keywords are secure under the chosen keyword attack in the instance.

Theorem 1. *Suppose the SDDH problem is hard; the dCRKS instance can achieve dCRKS ciphertext indistinguishability.*

Proof. Let $\mathscr{F}$ be polynomial-time adversary. If $\mathscr{F}$ can break the dCRKS instance with nonnegligible advantage probability, we construct an algorithm $\mathscr{G}$ as the challenger, who can solve the SDDH problem with nonnegligible advantage probability.

Init. The $\mathscr{F}$ issues the challenge public key $P_{d^*} = h^{z^*}$ and a keyword set w_i.

Setup. Let $\mathscr{L} = (g, h, g^a, h^c, e(g,h)^{1/a}, e(g,g)^t, e(g,h)^q)$ be a SDDH instance. $\mathscr{G}$ is given $\mathscr{L}$ and (G, G_T, $e(\cdot)$). The setup procedure produces parameters $\mathscr{PP}$, then $\mathscr{G}$ sends $\mathscr{PP}$ to $\mathscr{F}$.

Phase 1. $\mathscr{F}$ can carry out multiple queries. The restriction is that the number of enquiries is no more than polynomially bounded.

Pk-Query. To inquire d_i's private key, $\mathscr{F}$ transmits P_{d_i} to $\mathscr{G}$. If $d_i \neq d^*$, $\mathscr{G}$ returns $z_i = K_{d_i}$ to $\mathscr{F}$.

T-Query. To inquire the trapdoor of w_i, $\mathscr{F}$ issues w_i and $P_{r_i} = (e(g,h)^{1/x}, g^x)$ to $\mathscr{G}$. $\mathscr{G}$ responds the trapdoor T_{w_i} for P_{r_i}, w_i by calling the trapdoor oracle.

Challenge. Let d^* be the sender's identity. $\mathscr{F}$ selects w_0, w_1, and P_{d^*} as challenge. The restriction is that the secret key for P_{d^*} or the trapdoor for w_0, w_1 for P_{d^*} have not been inquired by $\mathscr{F}$. $\mathscr{G}$ replies the challenge ciphertext $C_{w_b}^*$ to $\mathscr{F}$, where $C_{w_b}^*$ is a tuple (C_1, C_2, C_3) and $C_3 = e(g,g)^t$, $C_2 = g^{at'} g^{-t' H(w_b)}$, and $C_1 = e(g,h)^q$.

Phase 2. $\mathscr{F}$ can still enquire the trapdoor and the secret key for $d \neq d^*$ or the trapdoor for $d = d^*$ and $w_i \neq w_0, w_1$. $\mathscr{G}$ replies as the front phase.

Outputs. In the end, $\mathscr{F}$ outputs $b' \in \{0, 1\}$.

Analysis. Because the actual dCRKS ciphertext C_{w_b} is $C_1 = e(g,h)^{zu/x}$, $C_2 = g^{x-H(w_b)u}$, and $C_3 = e(g,g)^u$, the distribution of challenge ciphertext $C^*_{w_b}$ is identical to that in the actual system. In fact, for the uniformly random t', $t \in Z^*_p$, $\mathscr{F}$ needs to differentiate between the tuple $(g^{t'(-H(w_b)+a)}$, $e(g,g)^t)$ and the tuple $(g^{t(-H(w_b)+a)}, e(g,g)^t)$. If $q = ct/a$, the simulation is perfect. ε denotes nonnegligible probability. As a result, if $\mathscr{F}$ has advantage probability $1/2 + \varepsilon$ to determine the bits b correctly, then the $\mathscr{G}$ can solve the SDDH problem with identical advantage probability ε.

This completes the proof of dCRKS ciphertexts indistinguishability. □

3.3.2. Security of Trapdoor. Now, we show that the trapdoor is secure under the chosen keyword attack.

Theorem 2. *The dCRKS instance can achieve the trapdoor indistinguishability to resist the chosen keyword attack under random oracle model in game 2.*

Proof. In this section, we show that the polynomial-time algorithm $\mathscr{F}$ is able to differentiate between the ciphertext of keyword w_0 and the ciphertext of keyword w_1 if and only if he can distinguish two uniform distributions on G.

Init. $\mathscr{F}$ issues $P_{r^*} = (g^{x^*}, e(g,h)^{1/x^*})$ as the challenge public key.

Setup. Running the setup procedure, $\mathscr{G}$ gives the public parameters to $\mathscr{F}$.

Phase 1. $\mathscr{F}$ implements multiple queries without exceeding polynomial bounded.

Pk-Query. To inquire r_i's private key, $\mathscr{F}$ sends $P_{r_i} = (e(g,h)^{1/x_i})$ to $\mathscr{G}$, $r_i \neq r^*$. Then $\mathscr{G}$ returns $K_{r_i} = x_i$.

dc-Query. $\mathscr{F}$ issues P_{d_i} and w_i to $\mathscr{G}$. Running EndCRKS oracle, $\mathscr{G}$ returns C_{w_i} for P_{d_i}, w_i to $\mathscr{F}$.

Challenge. $\mathscr{F}$ chooses keywords w_0, w_1, and P_{r^*} as his challenge. The restriction is that the ciphertext for w_0, w_1 for P_{r^*} or the private key for P_{r^*} has not be enquired by $\mathscr{F}$. $\mathscr{G}$ picks two random r', z' and computes the challenge trapdoor $T^*_{w_b}$, where $T^*_{w'_b} = (r', h^{z'} g^{-yr'})$. Then $\mathscr{G}$ replies the challenge trapdoor $T^*_{w'_b}$ to $\mathscr{F}$.

Phase 2. $\mathscr{F}$ can still enquire the ciphertext and the secret key for $r \neq r^*$ or the ciphertext for $r = r^*$ and $w_i \neq w_0, w_1$. $\mathscr{G}$ replies as first phase.

Outputs. $\mathscr{F}$ outputs $b' \in \{0, 1\}$.

Analysis. By enquiring, $\mathscr{F}$ can obtain g^{x^*}, $e(g,h)^{1/x^*}$, and $T^*_{w'_b} = (r', h^{z'} g^{-yr'})$. In scheme, $T^*_{w_b} = (r,$ $(h^{z/x^*} g^{-yr})^{1/(x^*-H(w_b))})$, where r is uniform random value. Thus the distribution of $(h^{z/x^*} g^{-yr})^{1/(x^*-H(w_b))}$ is a uniform distribution on G. Meanwhile, the $T^*_{w'_b} = (r', h^{z'} g^{-yr'})$ is uniform distribution on G with taking uniform random r', z'. As a result, the simulation is perfect, namely, the distribution of $T^*_{w'_b}$ is identical to that in the actual system.

Moreover, as game 2, $\mathscr{F}$ can not know z and x^*. Even if $\mathscr{F}$ can calculate the following value:

$$e(g^{(x^*-H(w_b))}, T^*_{w_b}) = e(g,h)^{(z/x^*-yr)((x^*-H(w_{b'}))/(x^*-H(w_b)))}$$

$\mathscr{F}$ cannot distinguish

$$e(g,h)^{(z/x^*-yr)((x^*-H(w_{b'}))/(x^*-H(w_b)))} \text{ (where } b' \neq b)$$

from $e(g,h)^{z/x^*} \cdot e(g,h)^{-yr}$ (where $b' = b$).

Therefore, $\mathscr{F}$ can guess $b' = b$ with nonnegligible advantage probability, then $\mathscr{G}$ can distinguish between $(h^{z/x^*} g^{-yr})^{1/(x^*-H(w_b))}$ and uniformly distribution on G with identical advantage probability. □

3.3.3. Analysis of against Inside KGAs. In this section, we show that the dCRKS instance is secure against inside KGAs as follows.

First, given the trapdoor T^d_w, the server can generate the legal ciphertext C corresponding to T^d_w if and only if it can obtain the specified senders private key. Maybe the server can select z' and calculate $h^{z'}$ to produce ciphertext $C = (C'_1, C'_2, C'_3)$ corresponding to w', where

$$e(g,h^{z'})^{u/x} = (P_{r1})^{z'u} = C'_1$$
$$g^{(x-H(w'))u} = (P_{r2})^u \cdot g^{-uH(w')} = C'_2$$
$$e(g,g)^u = C'_3$$

Let $T^d_w = (T_1, T_2)$, then

$$e(C_2, T_2) = e(g^{(x-H(w))u}, (h^{z/x} g^{-yr})^{1/(x-H(w'))})$$

Based on the dTest, the $e(C_2,T_2)C_3^{yT_1} = C_1$ is true if and only if $w = w'$ and $z' = z$, where z and w correspond to T^d_w. However, given the trapdoor T^d_w, the probability of selecting $z' \in Z^*_p$ such that $z' = z$ is negligible, even $w' = w$. Therefore, the malicious cloud server is not able to launch keyword-guessing attacks by computing the dCRKS ciphertext of all possible keywords.

Second, given the ciphertext $C_w = (C_1, C_2, C_3)$, the cloud server can not produce a legal trapdoor T_w corresponding to C_w. Although the server may select a x', P_d, and w' to generate the trapdoor, the probability of selecting $x' \in Z^*_p$ such that $x' = x$ is negligible, where x is the receiver's private key associated with the C_w. Based on the same analysis, we know that the malicious cloud server is not able to launch

Table 1: A comparison of various schemes.

Schemes	Inside KGAs	Outside KGAs	ZC	ZT	TrC	TeC	CiC
[28]	NO	YES	$l_G + l_H$	$2l_G$	$2E_t$	$P_t + 2E_t$	$P_t + 2E_t$
[17]	NO	NO	$l_G + l_H$	l_G	E_t	$P_t + E_t$	$P_t + E_t$
[29]	YES	YES	$2l_G$	$2l_G + 2l_p$	$2E_t$	$P_t + 2E_t$	$2E_t$
[25]	NO	YES	$3l_G + 2l_{GT}$	l_s	$l_G + l_p$	E_t	$4P_t + 3E_t + tv$
[27]	NO	YES	$2l_G + l_{GT}$	$2l_{GT}$	E_t	$3P_t + E_t$	$P_t + 4E_t$
Ours	YES	YES	$l_G + 2l_{GT}$	$l_G + l_p$	E_t	$P_t + E_t$	$3E_t$

keyword-guessing attacks by creating the trapdoor of all possible keywords.

Lastly, taking g^x, $H(w)$, and $e(g,h)^{1/x}$, the server may build the following equation: $e(g^{(x-H(w_b))}, T_{w_b}) = e(g,h)^{(z/x)((x-H(w_{b'}))/(x-H(w_b)))} \cdot e(g,h)^{-yr((x-H(w_{b'}))/(x-H(w_b)))}$

However, this equation can not help to find correct trapdoor or launch KGAs since T_{w_b} contains a random number. Summarize these reasons; the proposal is secure under the inside KGAs.

4. Performance Analysis

Now, we demonstrate efficiency of the proposal by analyzing its security and calculation cost. With the analysis in Table 1, it shows that only [29] and our scheme are secure to resist inside KGAs. Furthermore, the TTP is removed in our scheme.

To compare performance, let E_t and P_t be the exponential operation and the pairing operation over a bilinear group, respectively. The size of Z_p's element is denoted by l_p. Similarly, the character l_G and l_{G_T} denote the size of G's element and the size of G_T's element, respectively. The size of hash value denotes l_H. For brevity, the calculation cost of creating trapdoor and keyword ciphertext denote TrC and CiC, respectively. The character ZC denotes the size of keyword ciphertext. The ZT denotes the size of trapdoor.

From Table 1, it shows that one E_t is required to create trapdoor in our scheme. Compared with [29], our solution is more efficient to create trapdoor. Meanwhile, the performance overhead of testing is one E_t and one P_t. In [29], it requires two E_t and one P_t.

Lastly, the test procedure can only be run by a specified server. This improves the security of the system.

5. Conclusion

With the wide application of cloud computing, data privacy has become one of critical security issues for mobile users. The ciphertext retrieval is one of the most useful approaches to achieve data privacy for cloud storage. In this paper, we first proposed a new architecture and security model to resist inside KGAs, which is a strong attack on the keyword search scheme. We also proposed an instance of the dCRKS system. Under the security model proposed in our paper, this new scheme has been proven secure to resist inside KGAs. Security analysis and efficiency comparison show that our scheme is effective for the retrieval of encrypted data in cloud computing.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by Scientific Research Fund of Sichuan Provincial Education Department (no. 18ZA0546) and Guangzhou Scholars Project for Universities of Guangzhou.

References

[1] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[2] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Gao, "Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures," *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[3] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[4] J. Cai, Y. Wang, Y. Liu, J.-Z. Luo, W. Wei, and X. Xu, "Enhancing network capacity by weakening community structure in scale-free network," *Future Generation Computer Systems*, 2017.

[5] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 10, pp. 3184–3195, 2016.

[6] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New Publicly Verifiable Databases with Efficient Updates," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 546–556, 2015.

[7] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li, "Insight of the protection for data security under selective opening attacks," *Information Sciences*, vol. 412-413, pp. 223–241, 2017.

[8] B. Li, Y. Huang, Z. Liu, J. Li, Z. Tian, and S. Yiu, "HybridORAM: Practical oblivious cloud storage with constant bandwidth," *Information Sciences*, 2018.

[9] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: Towards a practical oblivious RAM with variable block size," *Information Sciences*, vol. 447, pp. 1–11, 2018.

[10] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When Intrusion Detection Meets Blockchain Technology: A Review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018.

[11] C. Yuan, X. Li, Q. Wu M J, J. Li, and M. Sun X, "Fingerprint Liveness Detection from Different Fingerprint Materials Using Convolutional Neural Network and Principal Component Analysis," *CMC: Computers, Materials and Continua*, vol. 53, no. 3, pp. 357–371, 2015.

[12] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51–62, 2018.

[13] W. Chen, H. Lei, and K. Qi, "Lattice-based linearly homomorphic signatures in the standard model," *Theoretical Computer Science*, vol. 634, pp. 47–54, 2016.

[14] C. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack," *Information Sciences*, vol. 444, pp. 72–88, 2018.

[15] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '00)*, pp. 44–55, IEEE, Berkeley, Calif, USA, May 2000.

[16] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," in *Advances in cryptology–EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Comput. Sci.*, pp. 223–238, Springer, Berlin, Germany, 2004.

[17] J. Baek, R. Safavinaini, and W. Susilo, "Public key encryption with keyword search revisited," in *Computational Science and Its Applications-ICCSA*, pp. 1249–1259, Springer, Berlin, Germany, 2008.

[18] W. C. Yau, S. H. Heng, and M. B. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *International Conference on Autonomic and Trusted Computing*, pp. 100–105, Springer, 2008.

[19] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: a provably secure scheme under keyword guessing attack," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

[20] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2014.

[21] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 127–139, 2015.

[22] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, 2016.

[23] L. Fan, X. Lei, N. Yang, T. Q. Duong, and G. K. Karagiannidis, "Secrecy Cooperative Networks with Outdated Relay Selection over Correlated Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7599–7603, 2017.

[24] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[25] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *Information Sciences*, vol. 238, pp. 221–241, 2013.

[26] R. Xie, C. Xu, C. He, and X. Zhang, "Lattice-based searchable public-key encryption scheme for secure cloud storage," *International Journal of Web and Grid Services*, vol. 14, no. 1, pp. 3–20, 2018.

[27] Y. Zhao, H. Ma, X. Chen, Q. Tang, and H. Zhu, "A new trapdoor-indistinguishable public key encryption with keyword search," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 3, no. 1-2, pp. 72–81, 2012.

[28] 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," *The Journal of Systems and Software*, vol. 83, no. 5, pp. 763–771, 2010.

[29] P. Jiang, Y. Mu, F. Guo, X. Wang, and Q. Wen, "Online/offline ciphertext retrieval on resource constrained devices," *The Computer Journal*, vol. 59, no. 7, pp. 955–969, 2016.

[30] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1206–1216, 2015.

[31] Q. Liu, Y. Guo, J. Wu, and G. Wang, "Effective query grouping strategy in clouds," *Journal of Computer Science and Technology*, vol. 32, no. 6, pp. 1231–1249, 2017.

[32] Y. Liu, J. Ling, Z. Liu, J. Shen, and C. Gao, "Finger vein secure biometric template generation based on deep learning," *Soft Computing*, vol. 21, no. 1, pp. 1–9, 2017.

[33] H. Wang, W. Wang, Z. Cui, X. Zhou, J. Zhao, and Y. Li, "A new dynamic firefly algorithm for demand estimation of water resources," *Information Sciences*, vol. 438, pp. 95–106, 2018.

[34] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword Search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, 2016.

[35] Q.-A. Wang, C. Wang, K. Ren, W.-J. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

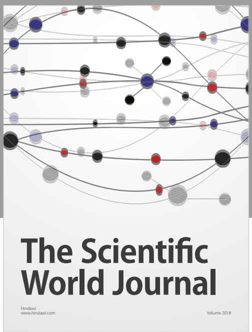
[36] J. Li, X. Tan, X. Chen, D. S. Wong, and F. Xhafa, "OPoR: Enabling proof of retrievability in cloud computing with resource-constrained devices," *IEEE Transactions on Cloud Computing*, vol. 3, no. 2, pp. 195–205, 2015.

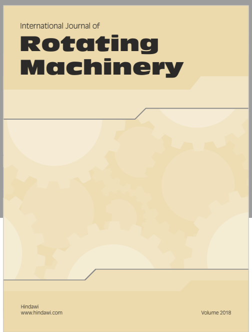
[37] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, 2018.

[38] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

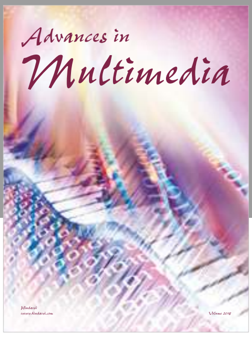
[39] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 17, no. 4, pp. 297–319, 2004.

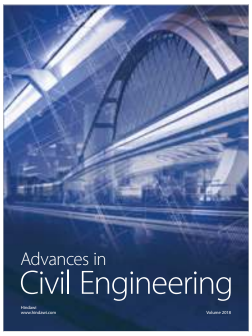
Journal of
Engineering

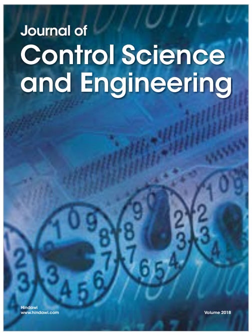

The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

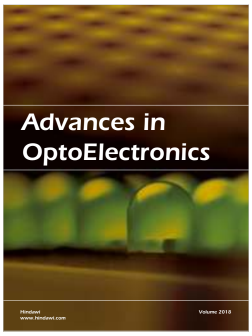

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

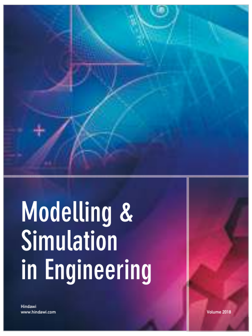
Journal of
Robotics

Journal of
Electrical and Computer Engineering


Advances in
OptoElectronics

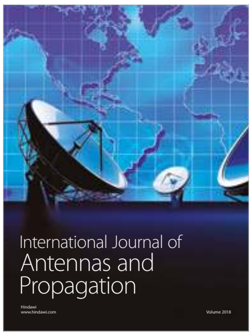
VLSI Design

International Journal of
Navigation and Observation

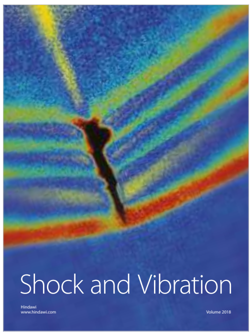

Modelling & Simulation in Engineering

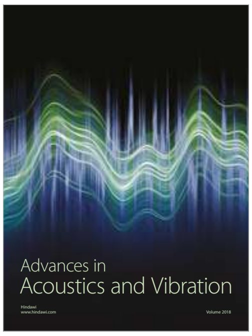
International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration