Yang *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2022) 11:10
https://doi.org/10.1186/s13677-022-00283-9

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Multi-Recipient encryption with keyword search without pairing for cloud storage

Ningbin Yang[1], Quan Zhou[1*], Qiong Huang[2] and Chunming Tang[1]

Abstract

With the rapid development of cloud computing technology and communication technology, cloud storage has become a tool used by people in daily life. Cloud storage service enables users to outsource data to cloud servers and retrieve desired document efficiently. Individual privacy in outsource data are very sensitive and should be prevented from any leakage. Public-key encryption with keyword search (PEKS) scheme resolves this tension, while public-key authentication encryption with keyword search (PAEKS) scheme improve its keyword guessing attacks problem potentially. Whereas, the loss of keyword privacy, the limitation of single user interaction and low efficiency make PEKS/PAEKS schemes far from enough in practical applications.

In this paper, we develop a multi-recipient public key encryption scheme with keyword search without pairing (MREKS) for cloud storage under public key infrastructure. The proposed scheme has the merits of supporting multi-recipient keyword search way as well as requiring no expensively bilinear pairing operations under standard model. We present a concrete and efficient construction of MREKS, and prove its security based on discrete logarithm assumptions. Furthermore, we embed the algorithm of data plaintext encryption and decryption into the scheme, which makes the scheme more practical. We show that our scheme enjoys much more efficiency than previous PEKS/PAEKS scheme in the simulation experiment, especially the keyword encryption is optimized by 79.5%.

Keywords: Cloud storage, Multi-Recipient, Public key encryption, Keyword guessing attacks

Introduction

In recent years, the amount of electronic data generated on various platforms such as the internet has seen an explosive growth. From the view of government, enterprises or individual, the increasing amount of data creates data management issues. To store this data, the user needs to maintain the hardware, software and systems for the data storage locally. It caused great overhead on the user's server, which has seriously affected the efficiency and flexibility of the user to utilize the data.

Cloud storage services in cloud computing technology alleviate this tension, which means users can obtain and pay for the server resources provided by cloud server without interaction largely and only need management work slightly. Due to the convenience and flexible of cloud service and varied charging properties, users are willing to store their local data in the cloud server. People can upload their data, such as email address, personal health record and financial data, into the cloud for sharing with other person or using it by themselves in anywhere. Moreover, cloud storage services are widely used in medical institutions, enterprises, schools and other application scenarios.

However, cloud storage has an inevitable drawback: users share or store data in the cloud server, so the ownership of the data is held by the cloud server. As a result, the cloud server can inadvertently obtain the data uploaded by users, leading to the divulge of sensitive privacy data without user's authority. To avoid this case, users can only

*Correspondence: zhouqq@gzhu.edu.cn
[1]School of Mathematics and Information Science, Guangzhou University, Guangzhou, China
Full list of author information is available at the end of the article

encrypt and upload document to the cloud. However, if users want to acquire target document, they download all the ciphertext data and decrypt it locally necessarily. It is unfriendly to users with large data storage capacity, which will result in huge resource waste and computing overhead. Moreover, this approach is hardly applicable to users with low broadband networks.

To address the above issue, the concept of searchable encryption has been proposed. As depicted in Fig. 1, a searchable encryption scheme works.

Therefore, the data security and privacy becomes an important issue. To date, many methods are proposed to protect privacy and security of cloud data [1–8].

Based on previous studies by researchers, searchable encryption divide into symmetric and asymmetric searchable encryption (SE). The work of Song et al. [1] is pioneering in constructing a symmetric SE scheme in 2000. His ideas were groundbreaking, but there were inevitable efficiency problems because the efficiency of finding the target document is linear length. Boneh et al. [2] constructed public-key encryption with keyword search, denoted by BDOP-PEKS. It is a branch of SE that keeps the confidentiality of the encrypted data. The BDOP-PEKS scheme is mainly applied in the mail routing scenario, in which three participants, namely the sender, the recipient and the mail server. The sender encrypts the message and keyword corresponding to the message via recipient's public key, and the recipient generates the search trapdoor via private key by himself. Finally, the mail server performs data retrieval and returns the message ciphertext with corresponding keyword to recipient.

Later, Baek et al. [3] found flaws in PEKS scheme and developed a secure channel free public key encryption with keyword search based on the BDOP-PEKS scheme, denoted by BSS-PKE/PEKS, which solved the issue of supplying secure channel when delivering keywords to the server. BSS-PKE/PEKS scheme performs via public channel, but it's still subject to a connatural security restriction: suffering off-line keyword guessing attacks (KGA). Specifically, given a keyword trapdoor, an adversary encrypts whole keyword candidates by using the recipient's public key and identifies the ciphertext which matches the targeted trapdoor, this enables the adversary to recover the keyword hidden in keyword trapdoor to invade the users' privacy. Public key authentication encryption with keyword search (PAEKS) was first proposed by Huang et al. [9], in which the sender's secret key is presented into the keyword encryption, so as to achieve the keyword trapdoor privacy and resist the keyword guessing attacks. Soon afterwards, Huang's scheme was proposed that it could not ensure the keyword ciphertext indistinguishability.

In such an architecture, previous PEKS and PAEKS schemes have been based on bilinear pairings operation, which can greatly restrict efficiency when running on devices with limited communication and computing capacity.

Traditional PEKS schemes for mail routing take into account single-user interactions, especially PAEKS scheme, where sharing data requires generating search trapdoor for the uniform keywords for each receiver. In fact, it will greatly reduce the desire of enterprise prac-

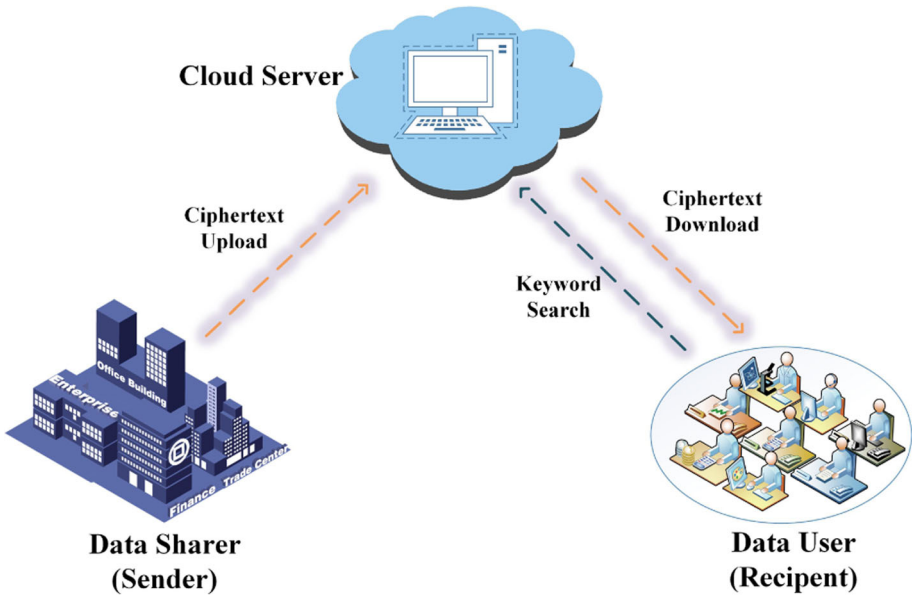

Fig. 1 Model of searchable encryption

tical application, since it still consumes a lot of storage resources to meet this search requirement. At present, the security of most PEKS schemes is not good enough to resist KGA. One reason is that the low-entropy feature of the keywords leads to KGA.

Therefore, we initiated the proposal of MREKS scheme to address the defects mentioned above.

Contribution

In this paper, we put forward a multi-recipient encryption with keyword search scheme without pairing for cloud storage based on public key infrastructure in virtue of the idea of Lu et al. [10](see related work). The proposed scheme not only supports multi-recipient authentication keyword search function, but also does not use the expensively bilinear pairing. We formally define the system model and security for the proposed MREKS and demonstrate the security of its under standard model. More specifically, our contributions are summarized as below:

Functionality: We construct a new multi-recipient PAEKS scheme without pairing for cloud storage under public key infrastructure. Let's consider a scenario where a user (i.e., a data sender) gathers transaction data and shares them with multiple recipients (e.g., a group of colleague in the company). Most PEKS and PAEKS schemes [2, 3, 9, 11, 12] merely support single recipient. The user has to generate a search trapdoor of same keyword for each recipient individually by using the above scheme. In this case, it will be inefficient and inconvenient awfully. To address the above issues, we create a single keyword encryption for a set of authorized recipients with high efficiency communication and computation.

Practicality: We embed message encryption and decryption to make MREKS scheme more practical. Most of PEKS and PAEKS schemes hardly support message encryption and decryption. In this case, the scheme are incompletely. In consequence, we adds this algorithm to keep the transmission of symmetric key confidentiality in the public channel and avoid transmitting the symmetric key via security channel. It is amicable for us to decrypt ciphertext commodiously. Moreover, the message decryption must match the corresponding keyword to decrypt it, which ensures the privacy of message and keyword in the transmission.

Security: The proposed of MREKS scheme provides privacy-preserving keyword search and data encryption. We prove the scheme prevent keyword guessing from attack successfully under standard model and plaintext privacy security. It is worth noting that we embed the recipient's private key in the keyword encryption process to avoid the possibility of outside adversary attack. Without the ability to produce valid ciphertext, the adversary is not able to carry out a successful keyword guessing attack. In this way, our scheme provides to resist attacks from adversary.

Efficiency: Our scheme avoids the expensively bilinear pairing. In various application scenarios, the computations are often performed on smart devices with constrained resources, such as telephone or handheld terminals. Most of the previous PEKS and PAEKS schemes [2, 3, 9, 13] were built with the bilinear pairing. If we use the without pairing scheme, the efficiency will be greatly improved. Also, it has more practical significance in the use of equipment with limited communication and computing capacity. We analyze the running overhead of MREKS theoretically and implement it utilizing C language and PBC library [14]. The analysis and experiment results show that our scheme has more efficiency running overhead with previous PEKS and PAEKS schemes.

Related work

The first asymmetric SE is presented by Boneh et al. [2] in 2004. Baek et al. addresses the Boneh's problem of working via security channel in 2008. Soon afterwards, with kinds of functions of PEKS scheme have been proposed. The working of Byun et al. [15] and Yau et al. [16] clearly that the current PEKS program are suffering from a novel attack, calls off-line keyword guessing attack. In their research, the previous program could not resist off-line keyword guessing attacks from the cloud servers. Based on Baek et al's work, Fang et al. [5] enhance security property and ensure the keyword security of the scheme under standard model. While the work of Fang et al. seems perfect, there are still keyword privacy problems. Therefore, the privacy of keywords in public key encryption with keyword search scheme has become an issue to be addressed by researchers.

The idea of "Trapdoor Indistinguishability" is proposed by Rhee et al. [6]. In their work, trapdoor indistinguishability is a sufficient condition under keyword security. Therefore, KGA under different assumption context is whether the success of determines the security of scheme. Based on various scenarios, we classify attackers as internal attackers or external attackers. In other words, an external adversary's attacks can be considered online KGA, since the adversary can produce the keyword ciphertext to guess in testing process by intercepting the user's search trapdoor. Similarly, an internal adversary's attacks (denotes semi-honest cloud server) can be considered off-line KGA, since the adversary is able to carry out test algorithm. The authority of the semi-honest cloud servers is power than the external attacker due to the cloud servers' testing executive capability.

Later, Huang et al. [9] constructed a new public key authentication encryption with keyword search to against inside adversary's attack. Ma et al. [17] put forward to

certificateless public key encryption with keyword search in the internet of thing (denote IOT) environment. Lu et al. [18] introduced a search trapdoor via key agreement between sender and receiver, which can resist the known KGA. Later, Ma et al. [19] constructed the scheme of SCF-CLSPE to achieve IND-CKA security for smart healthcare. Noroozi et al. [20] put forward to a generic construction secure against online and offline KGA scheme. Qin et al. [13] aimed at the revisited of the scheme proposed by Huang et al. [9], and introduced that the keyword privacy of Huang et al.'s scheme was insufficient, that is, it could not meet the multi-keyword ciphertext guessing attack securely. A verifiable public key SE was proposed after its improvement, which can achieving multi-keyword ciphertext indistinguishability. Pan et al. [11] has improved the work of Qin et al., and proposed to simultaneously ensure the multi-keyword ciphertext indistinguishability and multi-keyword trapdoor security. Whereafter, Cheng et al. [12] point out the work of Pan et al. a serious mistake in the security proof and Qin et al. [21] improved their multi-keyword ciphertext indistinguishability security model[13].

Chen et al. [22] brought up with a new type of public-key SE that can resist inside adversary's off-line keyword guessing attacks, namely server-aid public-key SE. In this scheme, blind keyword signature is provided by the server and returned to the user for keyword encryption. The key of blind signature of the server has the merit of key update for each sub-server, which makes the scheme more flexible. Zhang et al. [23] promoted the public key searchable encryption scheme based on the blockchain-based public chain application and was able to resist keyword guessing attacks. He et al. [24] and Li et al. [25] came up with PAEKS into certificateless keyword search and identity based encryption settings, respectively. Li et al. [26] put forward to a new public key searchable encryption scheme for single-user to multi-user interaction under the hierarchical identity mechanism and attribute encryption mechanism, and this scheme designed a public key searchable encryption scheme that supports transparent user access control. The scheme not only protects the privacy of keyword search, but also supports the users with private key to search ciphertext. Lu et al. [10] presented a new multi-recipient cetificateless public key searchable encryption scheme for IIOT, which supporting muti-user interaction function and no costly computation. Based on this contribution, we introduce this contribution into our scheme to better apply to cloud storage in PKI.

In addition to keyword searching, some schemes of public key cryptosystem in PEKS variants are also studied, including fuzzy keyword search [27], verifiable keyword search [28], lattice-based encryption with keyword search [29] and attribute-based keyword search [30].

Preliminaries

Complexity assumptions

Definition 1.(Discrete Logarithm(DL) assumption [31]) 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 exists an algorithm $A(t, \varepsilon)$ for solving DL problem, if $\Pr[A(g, g^a) = a] < \varepsilon$.

Definition 2. (Hash Diffie-Hellman(HDH) problem [32]) Let G be a cyclic group of prime q and g be a generator of G. $H : \{0,1\}^* \rightarrow \{0,1\}^l$ is a hash function, where l is a binary number. 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 Z is a random element of $\{0,1\}^l$. HDH problem is to judge whether $Z = H(g^{ab})$.

Definition 3. (Computational Diffie-Hellman (CDH) Problem [32]) Let G be a cyclic group of prime q and g be a generator of G. Given a binary tuple $(g^a, g^b) \in G^2$ for unknown integers $a, b \in Z_q^*$, the CDH problem in the group G is to calculate g^{ab}.

System model of mREKS

The proposed of MREKS model display in Fig. 2, including six polynomial time algorithms:

1) *GlobalSetup*(λ): Input a security parameter λ, and output global parameter GP.

2) *KeyGen*(GP): Input global parameter GP, and output a secret/public key pair (sk_u, pk_u) for user.

3) *Encrypt*$(GP, sk_S, (pk_1, pk_2, ..., pk_n)_R, w, M)$: Input GP, sk_S, multi-recipient's public key $(pk_1, pk_2, ..., pk_n)_R$, a keyword w and a message M, where n is number of recipient. Outputs ciphertext $C = (C_w, C_M)$, where C_w is keyword ciphertext and C_M is message ciphertext.

4) *Trapdoor*(GP, sk_R, pk_S, w'): Input GP, sk_R, pk_S, and a search keyword w', and output a keyword trapdoor $T_{w'}$.

5) *Test*$(GP, C_w, T_{w'})$: Input GP, C_w, $T_{w'}$, and output a symbol "1" if $w = w'$ or "0" otherwise.

6) *Decrypt*$(GP, w', C_M, pk_S, sk_R)$: Input GP, C_M, a keyword w', pk_S and sk_R. Output plaintext message M.

Security definition

This section we introduce the security definition of our proposed MREKS scheme. The security definition of ciphertext indistinguishability MREKS under the chosen keyword guessing attacks (denote CMREKS-CKA), trapdoor indistinguishability MREKS under the chosen keyword guessing attacks (denote TMREKS-CKA) and plaintext privacy MREKS against chosen plaintext attacks (denote PP-MREKS-CPA) are as follow:

CMREKS-CKA game

This game is simulated between A and a challenger B, where A is inside or outside adversary.

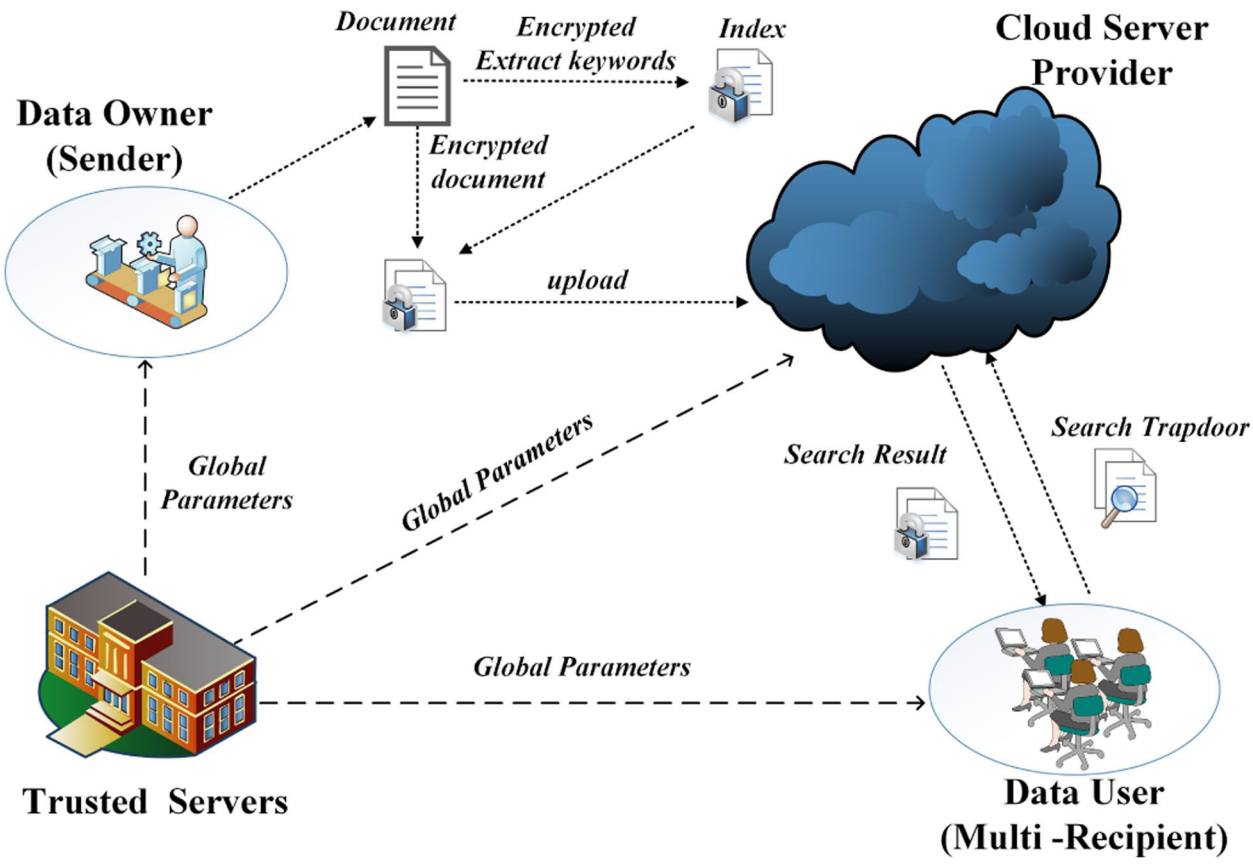

Fig. 2 System model of MREKS

GlobalSetup: Given security parameters λ, B produces global parameters GP, a sender and recipients' secret/public key pair (sk_S, pk_S) and (sk_R, pk_R), and sends pk_S, pk_R and GP to A.

Query Phase 1: A does $O^{Ciphertext}$, $O^{Trapdoor}$ and O^{Test} to B adaptively, then B simulates the corresponding algorithm in MREKS scheme and return the results.

Challenge: A submits two keywords (w_0, w_1) to B, which he/she has not submit to $O^{Ciphertext}$ in above Query phase 1. Finally, B returns a keyword ciphertext C_{w_b} with $b \in_R \{0, 1\}$.

Query Phase 2: A continues to ask for B adaptively, but with the restrictions that A can not queries w_0 or w_1 in ciphertext or trapdoor.

Guess: A returns $b' \in \{0, 1\}$ and A wins in this game, if $b = b'$.

The advantage of A in CMREKS-CKA Game is defined as follows:

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

Definition 3. An MREKS scheme achieve the CMREKS-CKA security if no polynomial time adversary can obtain a non-negligible advantage in CMREKS-CKA game.

TMREKS-CKA game

This game is simulated between A and a challenger B, where A is inside or outside adversary.

GlobalSetup: Same as that in CMREKS-CKA Game.

Query Phase 1: Same as that in CMREKS-CKA Game.

Challenge: A submits two keywords (w_0, w_1) to B, which he/she has not submit to $O^{Ciphertext}$ in above Query phase 1. Finally, B returns a keyword trapdoor T_{w_b} with $b \in_R \{0, 1\}$.

Query Phase 2: A continues to ask for B adaptively, but with the restrictions that A can not queries w_0 or w_1 in ciphertext or trapdoor.

Guess: A returns $b' \in \{0, 1\}$ and A wins in this game, if $b = b'$.

The advantage of A in TMREKS-CKA Game is defined as follows:

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

Definition 4. An MREKS scheme achieve the TMREKS-CKA security if no polynomial time adversary

can obtain a non-negligible advantage in TMREKS-CKA game.

PP-MREKS-CPA game

This game is simulated between A and a challenger B.

Setup: Same as that in CMREKS-CKA Game.

Query Phase 1: A can issue at most q_M queries to the encryption oracle O^M below.

O^M: A submits plaintext M with keyword w to B, and then B returns ciphertext C.

Challenge: A submits a keyword w and two plaintext M_0 and M_1. The constraint is that A cannot be submitted M_0 or M_1 to O^E. B picks a bit $b \in \{0,1\}$ randomly. Next, B generates a ciphertext C. Finally, B returns ciperhetext C to A.

Query Phase 2: A issues queries to the oracle same as in Query Phase 1 with the constraints that A cannot be submitted M_0 or M_1 with w to O^E.

Guess: A returns a bit b' and wins the game if $b' = b$.

The advantage of A in PP-MREKS-CPA Game is defined as follows:

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

Definition 5. An MREKS scheme achieve the PP-MREKS-CKA security if no polynomial time adversary can obtain a non-negligible advantage in PP-MREKS-CKA game.

The proposed mREKS scheme

This section we introduce our MREKS scheme. The scheme is described as follows.

1) $GlobalSetup(\lambda)$: Given the security parameter 1^λ, trusted servers picks a q-order cyclic group G. Let g is the generator of G. Furthermore, it selects four hash functions $H_1 : G \rightarrow \{0,1\}^l$, $H_2 : \{0,1\}^* \times \{0,1\}^l \rightarrow Z_q^*$, $H_3 : G \rightarrow Z_q^*$, $H_4 : \{0,1\}^l \times \{0,1\}^* \times Z_q^* \times G \times \{0,1\}^* \times \{0,1\}^* \times Z_q^* \rightarrow \{0,1\}^l$, where l is denotes the binary length of hash values. Finally, it outputs the global parameters $GP = \{q, g, G, H_1, H_2, H_3, H_4\}$.

2) $KeyGen(GP)$: Takes GP as input. The user (including sender and recipients) generates its secret/public key as follow.

- Selects $sk_{u_1}, sk_{u_2} \in_R Z_q^*$;
- Computes $pk_{u_1} = g^{sk_{u_1}}$ and $pk_{u_2} = g^{sk_{u_2}}$;
- Sets $sk_u = (sk_{u_1}, sk_{u_2})$ and $pk_u = (pk_{u_1}, pk_{u_2})$ as user's secret/public key pair.

3) $Encrypt(GP, pk_S, sk_S, (pk_1, pk_2, ..., pk_n)_R, w, M)$: Takes GP, pk_S, sk_S, a keyword w, multi-recipient's public key $(pk_1, pk_2, ..., pk_n)_R$ and a message M as input, where the subscript S indicates sender, the subscript R indicates recipient and n is the number of recipients. The sender selects $r \in Z_q^*$, $K \in \{0,1\}^l$ randomly and encrypt w and M as below:

- Computes $\mu_i = H_1(pk_{iR_1}^{sk_{S_1}})$ and $\theta_i = H_1(pk_{iR_2}^{sk_{S_2}})$ for each $i = 1, 2, ..., n$ and n is the number of recipients;
- Selects two random integer $\eta, \gamma \in Z_q^*$ and then define two polynomial $f(x)$ and $g(x)$ of degree n as follows:
  $f(x) = \prod_{i=1}^{n} (x - v_i) + \gamma =$
  $x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0$, where $\alpha_i \in Z_q^*$ and $v_i = H_3(g^{rH_2(w||\mu_i)})$ and the operator "||" denotes the concatenation of two strings;
  $g(x) = \prod_{i=1}^{n} (x - s_i) + \eta =$
  $x^n + \beta_{n-1}x^{n-1} + \cdots + \beta_1 x + \beta_0$, where $\beta_i \in Z_q^*$ and $s_i = H_3\left(pk_{S_2}^{(H_2(w||\theta_i)-r)}\right)$;
- Sets
  $C_1 = K \oplus H_1(pk_{S_2}^{\eta})$,
  $C_2 = AESEnc_K(M)$,
  $C_3 = r \cdot sk_{S_1}^{-1}$,
  $C_4 = g^{-sk_{S_2} \cdot r}$,
  $C_5 = (\alpha_0, \alpha_1, ..., \alpha_{n-1})$,
  $C_6 = (\beta_0, \beta_1, ..., \beta_{n-1})$,
  $C_7 = H_4(C_1, C_2, C_3, C_4, C_5, C_6, \gamma)$;

Outputs the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6, C_7)$.

4) $Trapdoor(GP, sk_{iR}, pk_S, w')$: The recipient executes as below:

- Computes $\mu_i' = H_1((pk_{S_1})^{sk_{iR_1}})$;
- Sets $t_1 = pk_{S_1}^{H_2(w'||\mu_i')}$;

Outputs the search trapdoor $T_{w'} = t_1$.

5) $Test(GP, C, T_{w'})$: The cloud sever executes as below:

- Parse C_5 as $(\alpha_0, \alpha_1, ..., \alpha_{n-1})$ and reconstruct the polynomial $f(x) = x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0$;
- Computes $v_i' = H_3({t_1}^{C_3})$ and $\gamma' = f(v_i')$ check whether $C_7 = H_4(C_1, C_2, C_3, C_4, C_5, C_6, \gamma')$ holds. If it does, output "1" or "0" otherwise.

6) $Decrypt(GP, C, w', pk_S, sk_R)$: The recipient executes as below:

- Parse C_5 as $(\alpha_0, \alpha_1, ..., \alpha_{n-1})$ and reconstruct the polynomial $f(x) = x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0$;
- Computes $v_i' = H_3({t_1}^{C_3})$ and $\gamma' = f(v_i')$ check whether $C_7 = H_4(C_1, C_2, C_3, C_4, C_5, C_6, \gamma')$ holds. If it does, turn to next phase or abort otherwise;
- Parse C_6 as $(\beta_0, \beta_1, ..., \beta_{n-1})$ and reconstruct the polynomial $g(x) = x^n + \beta_{n-1}x^{n-1} + \cdots + \beta_1 x + \beta_0$;
- Computes $\theta_i' = H_1((pk_{S_2})^{sk_{iR_2}})$.
- Sets $t_2 = pk_{S_2}^{H_2(w||\theta_i')}$ and $s_i' = H_3(C_4 \cdot t_2)$.
- Computes $\eta' = g(s_i')$ and $K = C_1 \oplus H_1(pk_{S_2}^{\eta'})$, then returns plaintext M, where $M = AESDec_K(C_2)$.

Remark. The decryption algorithm cannot be performed until the cloud server has passed the test algorithm and returned ciphertext C to the recipient. Otherwise, the decryption algorithm is not performed.

Correctness verification.

$$\mu_i = H_1\left(pk_{iR_1}^{sk_{S_1}}\right) = H_1\left(pk_{S_1}^{sk_{iR_1}}\right) = \mu_i', i = 1, 2, ..., n.$$

$$\theta_i = H_1\left(pk_{iR_2}^{sk_{S_2}}\right) = H_1\left(pk_{S_2}^{sk_{iR_2}}\right) = \theta_i', i = 1, 2, ..., n.$$

$$\begin{aligned} H_3\left(t_1{}^{C_3}\right) &= H_3\left(pk_{S_1}^{H_2(w'||\mu_i')\cdot r\cdot sk_{S_1}^{-1}}\right) \\ &= H_3\left(g^{r\cdot H_2(w'||\mu_i')}\right) = v_i \end{aligned}$$

$$\begin{aligned} H_3(t_2\cdot C_4) &= H_3\left(pk_{S_2}^{H_2(w||\theta_i)}\cdot g^{-sk_{S_2}\cdot r}\right) \\ &= H_3\left(g^{sk_{S_2}(H_2(w||\theta_i)-r)}\right) \\ &= H_3\left(pk_{S_2}^{(H_2(w||\theta_i)-r)}\right) = s_i \end{aligned}$$

If the target keyword $w = w'$, then the above equation are equal. Thus, our scheme is correct.

Security proof

This section we analysis the security of MREKS via game hopping [33].

Lemma 1(Difference Lemma [33]) Let E be some "error event" such that $S_1|\neg E$ occurs if and only if $S_2|\neg E$ occurs. Then

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

Theorem 1. The MERKS scheme realizes CMREKS-CKA game security under standard model, if $H_1 \sim H_4$ is the collision resistance hash function and HDH assumption is intractable.

Proof 1: Suppose that A is an internal or external adversary against the security of the proposed CMREKS-CKA game in polynomial time, A_H is the adversary of the hash function and A_{HDH} is the adversary of breaking the HDH assumption.

We prove the theorem 1 via five sub-game programs Game-j($j = 0, 1, 2, 3, 4$), and define Y_j are the events of A guessing correctly, that is $b = b'$. Therefore, the game-hopping proof of CMREKS-CKA is as follow:

Game-0: Game-0 is the original attack CMREKS-CKA game, so A have $Adv(\lambda)_A = |\Pr[Y_0] - 1/2|$.

Game-1: In this sub-game, B picks $sk_{S_2}, sk_{iR_2}, a, c_i \in Z_q^*$ randomly to calculate $pk_S = (g^a, g^{sk_{S_2}})$ and $pk_{iR} = (g^{c_i}, g^{sk_{iR_2}})$ for each the number of recipients $i = 1, 2, ..., n$, where g is the generator of group G. Other parameters is the same as Game-0. Obviously, Game-0 and Game-1 are indistinguishable from A. So, two sub-game is equal with the advantage of $\Pr[Y_0] = \Pr[Y_1]$.

Game-2: Game-2 is similar to Game-1, except that B transforms to the respond queries and challenge pattern. B does the following queries:

-$O^{Ciphertext}$: A submits a keyword w to B, then B picks a random integer $r \in Z_q^*$ and returns $C = (C_1, C_2, ..., C_7)$ to A.

-$O^{Trapdoor}$: A submits a keyword w' to B, and returns $T_{w'} = pk_{S_1}^{H_2(w'||\mu_i')}$, where $\mu_i' = H_1\left(pk_{S_1}^{sk_{iR_1}}\right)$.

-O^{Test}: A submits C and $T_{w'}$ to B, then B returns 1 if $v_i' = H_3(t_1{}^{C_3})$ and $C_7 = H_4(C_1, C_2, C_3, C_4, C_5, C_6, \gamma = f(v_i'))$ or 0 otherwise.

Challenge: A submits two different keywords (w_0, w_1), where w_0 or w_1 are not challenged in previous phase. B chooses $r^* \in_R Z_q^*$ and $b \in_R \{0, 1\}$ and performs as follow:

a) Sets $C_3^* = r^* \cdot (a^{-1})$ and $C_4^* = g^{-sk_{S_2}\cdot r^*}$;

b) Computes $\mu_i^* = H_1((g^{c_i})^a)$.

c) Selects random integers $s_1^*, s_2^*, ..., s_n^*, \eta^*, \gamma^* \in Z_q^*$ and define two polynomial

$$\begin{aligned} f^*(x) &= \prod_{i=1}^{n}\left(x - v_i^*\right) + \gamma^* \\ &= x^n + \alpha_{n-1}^* x^{n-1} + \cdots + \alpha_1^* x + \alpha_0^*, \end{aligned}$$

where $\alpha_i^* \in Z_q^*$ and $v_i^* = H_3\left(g^{r^* H_2(w_b||\mu_i^*)}\right)$;

$$\begin{aligned} g^*(x) &= \prod_{i=1}^{n}(x - s_i^*) + \eta^* \\ &= x^n + \beta_{n-1}^* x^{n-1} + \cdots + \beta_1^* x + \beta_0^*, \end{aligned}$$

where $\beta_i^* \in Z_q^*$;

d) Selects $C_1^* \in \{0, 1\}^l$, $C_2^* \in \{0, 1\}^l$ randomly

e) Sets

$$C_5^* = (\alpha_0^*, \alpha_1^*, ..., \alpha_{n-1}^*),$$

$$C_6^* = \left(\beta_0^*, \beta_1^*, ..., \beta_{n-1}^*\right),$$

$$C_7^* = H_4\left(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, \gamma^*\right);$$

f) Returns $C^* = \left(C_1^*, C_2^*, ..., C_7^*\right)$ to A.

Therefore, the challenge ciphertext $C^* = (C_1^*, ..., C_7^*)$ is the effective ciphertext of the keyword w_b .

Game-1 and Game-2 will be uniform, if B asks for queries and challenge correctly. It means that A guesses correctly in both sub-game with the advantage of $\Pr[Y_2] = \Pr[Y_3]$.

Game-3: Game-3 is the same as Game-2, except that B will abort the sub-game, if the following events occur.

Event E_1: A submits w to B in $O^{Ciphertext}$, including the keyword's input satisfies $w \neq w_b$, but

a. $f(x) = f^*(x) = \prod_{i=1}^{n}\left(x - v_i^*\right) + \gamma^*$ for $C_4 = (\alpha_0^*, \alpha_1^*, ..., \alpha_{n-1}^*)$, where $v_i = v_{i_b}$ and $\gamma^* \in Z_q^*$.

b. $C_7^* = H_4\left(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, \gamma^*\right)$.

Event E_2: A submits w to B in $O^{Trapdoor}$, including the keyword's input satisfies $w \neq w_b$, but $H_2(w||\mu_i) = H_2(w_b||\mu_i^*)$.

Remark. In the Event E_1, we do not consider the computation of polynomial function $g(x)$. Even if A calculates the polynomial function $g(x) = \prod_{i=1}^{n}\left(x - s_i^*\right) + \eta^*$, where $s_1^*, s_2^*, ..., s_n^*, \eta^* \in Z_q^*$, the keyword ciphertext cannot be matched in the cloud server.

Obviously, Game-2 and Game-3 are indistinguishable to A unless the event $E_1 \vee E_2$ occurs. Due to Difference Lemma, we have

$$|\Pr[Y_2] - \Pr[Y_3]| \leq \Pr[E_1 \vee E_2].$$

Furthermore, it will be have A_H, if the event E_1 occurs. Therefore, A_H has the advantage of winning, if

$$(Adv(\lambda)_{A_H})^{n+1} \cdot \frac{1}{q} \geq \Pr[E_1],$$

where n is the number of recipient and q is random number of Z_q^*.

Similarly, it will be have A_H, if the event E_2 occurs. Therefore, A_H has the advantage of winning, if

$$Adv(\lambda)_{A_H} \geq \Pr[E_2].$$

Therefore, we induce the equation

$$|\Pr[Y_2] - \Pr[Y_3]| \leq Adv(\lambda)_{A_H} + (Adv(\lambda)_{A_H})^{n+1} \cdot \frac{1}{q}.$$

Game-4: Game-4 is the same as Game-3, except that B picks a random element $Z \in \{0,1\}^l$ instead of $H_1(g^{ac_i})$ when generating the challenge of ciphertext. Obviously, B responds queries and chanllenge via HDH tuples $(H_1, g, g^a, g^{c_i}, Z)$ without revealing the integer of a and c_i. In consequence, Game-3 is equivalent to Game-4. A_{HDH} distinguish the element of $\mu_i{}' = H_1(g^{ac_i})$ (for $i = 1, 2, ..., n$) and Z with non-negligible advantage, if the HDH problem is addressed. Hence, A_{HDH} has the advantage to win Game-4 with

$$|\Pr[Y_3] - \Pr[Y_4]| \leq Adv(\lambda)_{A_{HDH}}.$$

Z is a random integer of G, so A has the advantage of winning with $\Pr[Y_4] = 1/2$.

Next, A can guess correctly in the above sub-games with the advantage

$$\begin{aligned} Adv(\lambda)_A = |\Pr[Y_0] - 1/2| &\leq |\Pr[Y_0] - \Pr[Y_1]| \\ &+ |\Pr[Y_1] - \Pr[Y_2]| + |\Pr[Y_2] - \Pr[Y_3]| \\ &+ |\Pr[Y_3] - \Pr[Y_4]| + |\Pr[Y_4] - 1/2|. \end{aligned}$$

Based on the triangle inequality, the above sub-games induce as follow:

$$Adv(\lambda)_A = Adv(\lambda)_{A_H} + \frac{1}{q} Adv(\lambda)_{A_H}^{n+1} + Adv(\lambda)_{A_{HDH}}.$$

The collision resistance property of the hash function H and the complication of HDH problem is complicated so that $Adv(\lambda)_A$ is negligible in theorem 1.

Theorem 2. The MERKS scheme realizes TMREKS-CKA game security under standard model, if $H_1 \sim H_4$ is the collision resistance hash function and HDH assumption is intractable.

Proof 2: Suppose that A is an internal or external adversary against the security of the proposed TMREKS-CKA game in polynomial time, A_H is the adversary of the hash function and A_{HDH} is the adversary of breaking the HDH problem.

We prove the theorem 2 via five sub-game programs Game-j($j = 0, 1, 2, 3, 4$), and define Y_j are the events of A guessing correctly, that is $b = b'$. Therefore, the game-hopping proof of TMREKS-CKA is as follow:

Game-0: Game-0 is the original attack TMREKS-CKA game, so A have $Adv(\lambda)_A = |\Pr[Y_0] - 1/2|$.

Game-1: This sub-game is the same as the Game-1 of theorem 1.

Game-2: Game-2 is similar to Game-1, except that B transforms to the respond queries and challenge pattern. B does the following queries:

-$O^{Ciphertext}$: A submits a keyword w to B, then B picks a integer $r \in_R Z_q^*$ and returns $C = (C_1, C_2, ..., C_7)$ to A.

-$O^{Trapdoor}$: A submits a keyword w' to B, and returns $T_{w'} = pk_{S_1}^{H_2(w'||\mu_i')}$, where $\mu_i' = H_1((pk_{S_1})^{sk_{iR_1}})$.

-O^{Test}: A submits C and $T_{w'}$ to B, then B returns 1 if $v_i{}' = H_3\left(t_1{}^{C_3}\right)$ and $C_7 = H_4(C_1, C_2, C_3, C_4, C_5, C_6, \gamma = f(v_i{}'))$ or 0 otherwise.

Challenge: A submits two different keywords (w_0, w_1) to B, where w_0 and w_1 are not challenged in previous phase. B chooses $b \in \{0, 1\}$ randomly for a keyword trapdoor $T_{w_b} = pk_{S_1}^{r \cdot H_2(w_b||\mu_i')}$, where $\mu_i' = H_1(g^{ac_i})$. And then returns them to A.

Therefore, the challenge trapdoor is the effective trapdoor of the keyword w_b.

Game-1 and Game-2 will be uniform, if B asks for queries and challenge correctly. It means that A guesses correctly in both sub-game with the same advantage $\Pr[Y_2] = \Pr[Y_1]$.

Game-3: This sub-game is the same as the Game-3 of theorem 1.

Therefore, we induce the equation

$$|\Pr[Y_2] - \Pr[Y_3]| \leq Adv(\lambda)_{A_H} + (Adv(\lambda)_{A_H})^{n+1} \cdot \frac{1}{q}.$$

Game-4: This sub-game is the same as the Game-4 of theorem 1.

Therefore, A has the advantage of winning with $Pr[Y_4] = 1/2$.

Next, A can guess correctly in the above game with the advantage

$$\begin{aligned} Adv(\lambda)_A &= |\Pr[Y_0] - 1/2| \le |\Pr[Y_0] - \Pr[Y_1]| \\ &+ |\Pr[Y_1] - \Pr[Y_2]| + |\Pr[Y_2] - \Pr[Y_3]| \\ &+ |\Pr[Y_3] - \Pr[Y_4]| + |\Pr[Y_4] - 1/2|. \end{aligned}$$

Based on the triangle inequality, the above sub-games induce as follow:

$$Adv(\lambda)_A = Adv(\lambda)_{A_H} + \frac{1}{q} Adv(\lambda)_{A_H}^{n+1} + Adv(\lambda)_{A_{HDH}}.$$

The collision resistance property of the hash function H and the complication of HDH problem is complicated so that $Adv(\lambda)_A$ is negligible in theorem 2.

Theorem 3: The MREKS scheme realizes PP-MREKS-CPA game secure if AES encryption is IND-CPA secure and the CDH and DL assumptions holds.

Proof 3: The MREKS scheme leverages the AES to encrypt the plaintext M and hides the session key K into C_1 . Hence, if C_1 does not divulge any information about the encryption key K, security of our MREKS will be based on AES. As long as we ensure the security of η is equivalent to ensuring the security of K, that is, we need to keep the keyword's security, if the hash function is collision resistant. The following game is played between a PPT adversary A and the challenger B. Given a DL instances (G, g, g^a) and CDH instances (H_1, g, g^a, g^η), where $a, \eta \in Z_q^*$, B works as follows.

GlobalSetup: B initializes the system to produce $GP = \{q, g, G, H_1, H_2, H_3, H_4\}$. B sends GP, the public key of senders and recipients $pk_S = (g^{sk_{S_1}}, g^{sk_{S_2}}) = (g^{sk_{S_1}}, g^a)$ and $pk_{iR} = (g^{sk_{iR_1}}, g^{sk_{iR_2}})$ to A, where g is the generator of group G and each recipient denotes $i = 1, 2, ..., n$.

Phase 1: A can issue queries to the hash oracle and encryption oracle O^{H_1} , O^{H_3}, O^{H_4} and $O^{Ciphertext}$, respectively.

-O^{H_1}: Given an element $g^* \in G$, it returns l-bit random number h^* as the hash value $H_1(g^*)$.

-O^{H_3}: Given an element $g' \in G$, it returns a random number $h' \in Z_q^*$ as the hash value $H_3(g')$.

-O^{H_4}: Given an arbitrary string length $\{0,1\}^*$, it returns l bit string length $\{0,1\}^l$ as the hash value $H_4(\{0,1\}^*)$.

-$O^{Ciphertext}$: A submits a keyword w and a plaintext M to B, then B picks a random integer $r \in Z_q^*$ and returns $C = (C_1, C_2, ..., C_7)$ to A.

Challenge: A submits to B its keyword w and two plaintexts (M_0, M_1). B generates a ciphertext C_b, where the random bit b decides which plaintext is encrypted in this ciphertext. B chooses $r^* \in Z_q^*$, $b \in \{0,1\}$ randomly and performs as follow:

a) Selects random integers $v_1^*, v_2^*, ..., v_n^*, \gamma^*, \eta^* \in Z_q^*$.

b) Computes $s_i^* = H_3\left(pk_{S_2}^{(H_2(w_b||\theta_i^*)-r^*)}\right)$, where $\theta_i^* = H_1(pk_{iR_2}^{sk_{S_2}})$ and define two polynomial

$$\begin{aligned} f^*(x) &= \prod_{i=1}^{n} (x - v_i^*) + \gamma^* \\ &= x^n + \alpha_{n-1}^* x^{n-1} + \cdots + \alpha_1^* x + \alpha_0^*, \end{aligned}$$

where $\alpha_i^* \in Z_q^*$;

$$\begin{aligned} g^*(x) &= \prod_{i=1}^{n} (x - s_i^*) + \eta^* \\ &= x^n + \beta_{n-1}^* x^{n-1} + \cdots + \beta_1^* x + \beta_0^*, \end{aligned}$$

where $\beta_i^* \in Z_q^*$;

c) Computes

$C_1^* = K \oplus H_1(pk_{S_2}^{\eta^*})$,
$C_2^* = AESEnc_K(M_b)$,
$C_3^* = (sk_{S_1})^{-1} r^*$,
$C_4^* = g^{-ar^*}$;

d) Sets

$C_5^* = (\alpha_0^*, \alpha_1^*, ..., \alpha_{n-1}^*)$,
$C_6^* = (\beta_0^*, \beta_1^*, ..., \beta_{n-1}^*)$,
$C_7^* = H_4(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, \gamma^*)$;

f) Returns the ciphertext $C_b = (C_1^*, ..., C_7^*)$ to A.

Phase 2: A still can issue queries to the oracles same as in phase 1 except that the ciphertext C_b cannot appears in the decrypt oracle O^D.

Guess: A returns a bit b' and wins the game if $b' = b$.

We define event E_1 and E_2.

E_1: A issues h^* to O^{H_1}.

E_2: A issues $g' = g^{a(H_2(w'||\theta_i')-r)}$ to O^{H_3},

In case E_1 happens, the challenger B solves the CDH problem via computing $g^* = g^{a\eta}$.

In case E_2 happens, the challenger B solves the DL problem via computing $g^{(H_2(w'||\theta_i')-r)} = g'^{\frac{1}{a}}$.

If the DL and CDH assumption holds, E_1 and E_2 happens with a negligible probability. That is

$$Adv(\lambda)_A = t \cdot Adv(\lambda)_{A_{DL}} \cdot Adv(\lambda)_{A_{CDH}},$$

where t is a (polynomial) upper bound on the number of queries.

In another case, E_1 and E_2 does not happen, the ciphertext C is random in A's view and the session key K can be revealed with a negligible probability. That is

$$\left[Adv(\lambda)_A = \left|\Pr\left[b = b' \mid \neg E_1 \wedge \neg E_2\right] - 1/2\right|.\right]$$

Therefore, A's wining advantage is equal to or less than a negligible probability if AES encryption is IND-CPA secure and the DL and CDH assumption holds in this game.

Notice. We deduce that the computation of η is approximately the computation g^η, since the computation of s_i in the polynomial $g(x)$ is to solve the DL assumption.

Table 1 Computation cost comparison

Schemes	Encrypt	Trapdoor	Test	MR	PF	IKGA
BDOP-PEKS[2]	$n(2t_e + 2t_h + t_p)$	$t_e + t_h$	$t_p + t_h$	No	No	No
BSS-PKE/PEKS[3]	$n(3t_e + 4t_h + t_p)$	$t_e + t_h$	$t_p + t_e + t_h$	No	No	No
HL-PAEKS[9]	$n(3t_e + t_h)$	$t_p + t_e + t_h$	$2t_p$	No	No	Yes
QY-PAEKSR[13]	$n(3t_e + 2t_h + t_p)$	$2t_e + t_h$	$t_h + t_p$	No	No	Yes
Ours	$(2n+1)t_h + (2n+1)t_e$	$2t_e + 2t_h$	$t_e + 2t_h$	Yes	Yes	Yes

"MR" denotes if the scheme withstand multi-recipient.
"PF" denotes if the scheme withstand pairing free operation.
"IKGA" denotes if the scheme withstand the inside off-line keyword guessing attack

Performance analysis

This section we evaluates the efficiency computation and communication cost of our scheme.

Now we present the following notions for basic operations in Table 1:

t_h: the cost for computing a map-to-point hash.
t_p: the cost for a bilinear pairing.
t_e: the cost for a modular exponentiation.
n: the number of recipients.

To give a more intuitive comparison, we test the time cost of the compared schemes by employing the PBC library on a laptop running Ubuntu 16.04 with Interl Core i5-4210U CPU @1.7-GHz and 11-GB RAM memory. A Type-A pairing was chosen and used to initialize the system, which owns the same security level as a 1024-bit RSA encryption.

The schemes proposed in [2, 3, 9, 13] are the based on bilinear pairing operation. Let $G \times G \rightarrow G_T$, where G_T is the bilinear map group.

The computation cost of the keyword encryption algorithm, trapdoor algorithm and test algorithm in MREKS and schemes [2, 3, 9, 13]. See Fig. 3 and Table 2. We run the keyword encryption algorithm 100 times for one keyword and recipient in our scheme's average is about 1.594 ms. When the number of recipients increases to 10, our scheme costs about 15.42 ms. Compared to scheme [9], the computation cost of MREKS is reduced by 79.5% in keyword encryption phase. If the number of recipients is infinite, the keyword encryption efficiency of MREKS scheme will be more excellent than other PEKS schemes.

In addition, the time cost of trapdoor in our scheme is fast than previous PEKS scheme. We set the recipients is 10 and the time cost of keyword testing in MREKS scheme is about 1.53 ms, while that in [2, 3, 9, 13] is about 2.1 ms, 3.1 ms, 2.9 ms and 2.13 ms, respectively.

Remark. In order to prevent indeterminate and affected by the length of the plaintext as well as making a better comparison with PEKS schemes, the encryption and decryption of AES algorithm are not included in the ciphertext computation and communication.

Communication cost

To visually display the comparison of storage length between different schemes based on PBC library's parameters, we now describle communication costs in Table 3 with the following notations:

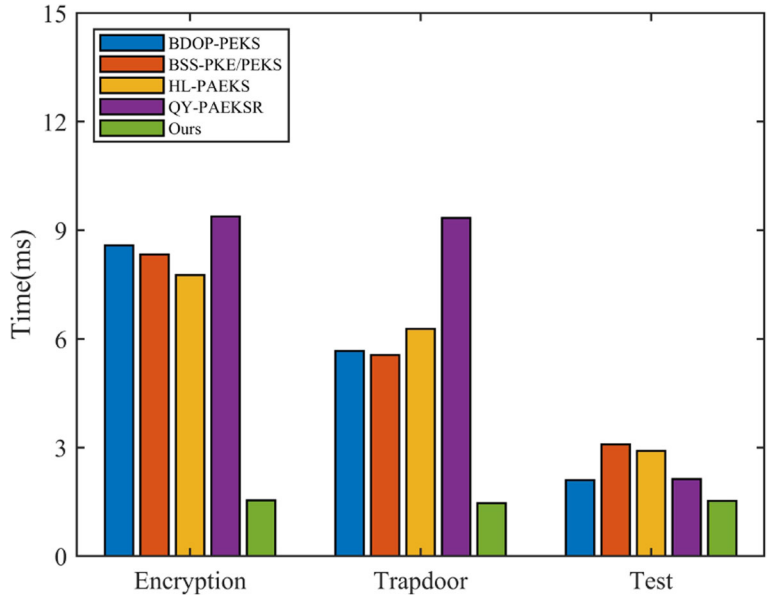

Fig. 3 Time cost of ten keyword operations

Table 2 Computation cost comparison of single recipient

Schemes	Encryption	Trapdoor
[2]	8.583*ms*	5.67*ms*
[3]	8.331*ms*	5.56*ms*
[9]	7.759*ms*	6.28*ms*
[13]	9.378*ms*	9.34*ms*
Ours	1.594*ms*	1.46*ms*

$|G|$: the 512 bit-size of an element in G.
$|G_T|$: the 1024 bit-size of element in G_T.
$|Z_q^*|$: the 128 bit-size of integer in Z_q^*.
h: the 256 bit-size of a hash value.
n: the number of recipients.

We clearly have that MREKS scheme is less than the PEKS and PAEKS schemes [2, 3, 9, 13] in the size of keyword encryption algorithm. Especially as the number of recipients n increases, our scheme is relatively more efficient. Furthermore, the size of trapdoor in MREKS scheme is smaller than schemes [2, 3, 9, 13].

Conclusion

PAEKS scheme is a useful cryptographic paradigm that supplies a feasible solution to the issue of encrypted data retrieval for cloud storage. MREKS techniques are used to simultaneously provide authentication, no costly bilinear pairing operations as well as multi-recipient keyword search function. Furthermore, we embed the encryption of message to our scheme, and the decryption needs to match the corresponding keyword information, which ensures the privacy of message and keywords. We formally prove that it ensures keyword security without random oracles and plaintext security. Moreover, we evaluate the performance of the proposed of our scheme with the previous PEKS and PAEKS scheme. The results demonstrate that our scheme is much more efficient than the previous schemes, especially on the computation efficiency. It is expedite for user to search over encrypted data for cloud storage due to the feature.

Table 3 Communication cost comparison

Schemes	Encryption Size/bit	Trapdoor Size/bit
[2]	$n(\|G\| + h) = 768n$	$\|G\| = 512$
[3]	$n(\|G\| + h) = 768n$	$\|G\| = 512$
[9]	$2n\|G\| = 1024n$	$\|G_T\| = 1024$
[13]	$n(\|G\| + h) = 768n$	$\|G\| = 512$
Ours	$h + (n+1)\|Z_q^*\| = 384 + 128n$	$\|G\| = 512$

Acknowledgements
The authors would like to thank to anonymous reviewers for their valuable comments on the manuscript.

Authors' contributions
Ningbin Yang put forward the main ideas and drafted the manuscript. Quan Zhou guided the research and participated in the discussion of the manuscript. Qiong Huang and Chunming Tang made suggestions for the article. All authors read and approve the final manuscript. All authors approve to be submitted in "Journal of Cloud Computing-Advances Systems and Applications".

Funding
This work of Quan Zhou is supported by the National Key Research and Development Program of China (No. 2021YFA1000600). This work of Qiong Huang is supported by National Natural Science Foundation of China (61872152), Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), and the Science and Technology Program of Guangzhou (201902010081). This work of Chunming Tang is supported by National Natural Science Foundation of China (61772147).

Availability of data and materials
The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Mathematics and Information Science, Guangzhou University, Guangzhou, China. [2]College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China.

Received: 30 September 2021 Accepted: 28 February 2022
Published online: 15 March 2022

References

1. Song DX, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000. pp 44–55. https://doi.org/10.1109/SECPRI.2000.848445
2. Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G (2004) Public key encryption with keyword search. In: Cachin C., Camenisch JL (eds). Advances in Cryptology - EUROCRYPT 2004. Springer, Berlin, Heidelberg. pp 506–522
3. Baek J, Safavi-Naini R, Susilo W (2008) Public key encryption with keyword search revisited. In: Gervasi O, Murgante B, Laganà A, Taniar D, Mun Y, Gavrilova ML (eds). Computational Science and Its Applications – ICCSA 2008. Springer, Berlin, Heidelberg. pp 1249–1259
4. Fang L, Susilo W, Ge C, Wang J (2013) Public key encryption with keyword search secure against keyword guessing attacks without random oracle. Inf Sci 238:221–241. https://doi.org/10.1016/j.ins.2013.03.008
5. Fang L, Susilo W, Ge C, Wang J (2009) A secure channel free public key encryption with keyword search scheme without random oracle. In: Garay JA, Miyaji A, Otsuka A (eds). Cryptology and Network Security. Springer, Berlin, Heidelberg. pp 248–258
6. Rhee HS, Park JH, Susilo W, Lee DH (2010) Trapdoor security in a searchable public-key encryption scheme with a designated tester. J Syst Softw 83(5):763–771. https://doi.org/10.1016/j.jss.2009.11.726
7. Baek J, Safavi-Naini R, Susilo W (2006) On the integration of public key data encryption and public key encryption with keyword search. In: Katsikas SK, López J, Backes M, Gritzalis S, Preneel B (eds). Information Security. Springer, Berlin, Heidelberg. pp 217–232
8. Zhang L, Xiong H, Huang Q, Li J, Choo KR, LI J (2019) Cryptographic solutions for cloud storage: Challenges and research opportunities. IEEE Trans Serv Comput:1–1. https://doi.org/10.1109/TSC.2019.2937764
9. Huang Q, Li H (2017) An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. Inf Sci 403-404:1–14. https://doi.org/10.1016/j.ins.2017.03.038
10. Lu Y, Li J, Zhang Y (2020) Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted iiot. IEEE Internet Things J 7(4):2553–2562. https://doi.org/10.1109/JIOT.2019.2943379

11. Pan X, Li F (2021) Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. J Syst Archit 115:102075. https://doi.org/10.1016/j.sysarc.2021.102075
12. Cheng L, Meng F (2021) Security analysis of pan et al.'s "public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability". J Syst Archit 119:102248. https://doi.org/10.1016/j.sysarc.2021.102248
13. Qin B, Chen Y, Huang Q, Liu X, Zheng D (2020) Public-key authenticated encryption with keyword search revisited: Security model and constructions. Inf Sci 516:515–528. https://doi.org/10.1016/j.ins.2019.12.063
14. Lynn B, et al. (2013) Pairing-based cryptography library. https://crypto.stanford.edu/pbc/
15. Byun JW, Rhee HS, Park H-A, Lee DH (2006) Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In: Jonker W, Petković M (eds). Secure Data Management. Springer, Berlin, Heidelberg. pp 75–83
16. Yau W-C, Heng S-H, Goi B-M (2008) Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In: Rong C, Jaatun MG, Sandnes FE, Yang LT, Ma J (eds). Autonomic and Trusted Computing. Springer, Berlin, Heidelberg. pp 100–105
17. Ma M, He D, Kumar N, Choo KR, Chen J (2018) Certificateless searchable public key encryption scheme for industrial internet of things. IEEE Trans Ind Inform 14(2):759–767. https://doi.org/10.1109/TII.2017.2703922
18. Lu Y, Wang G, Li J (2019) Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement. Inf Sci 479:270–276. https://doi.org/10.1016/j.ins.2018.12.004
19. Ma M, He D, Fan S, Feng D (2020) Certificateless searchable public key encryption scheme secure against keyword guessing attacks for smart healthcare. J Inf Secur Appl 50:102429. https://doi.org/10.1016/j.jisa.2019.102429
20. Noroozi H, Eslami X (2020) Public-key encryption with keyword search: a generic construction secure against online and offline keyword guessing attacks. J Ambient Intell Human Comput 11:879–890. https://doi.org/10.1007/s12652-019-01254-w
21. Qin B, Cui H, Zheng X, Zheng D (2021) Improved security model for public-key authenticated encryption with keyword search. In: Huang Q, Yu Y (eds). Provable and Practical Security. Springer, Cham. pp 19–38
22. Chen R, Mu Y, Yang G, Guo F, Huang X, Wang X, Wang Y (2016) Server-aided public key encryption with keyword search. IEEE Trans Inf Forensic Secur 11(12):2833–2842. https://doi.org/10.1109/TIFS.2016.2599293
23. Zhang Y, Xu C, Ni J, Li H, Shen XS (2019) Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage. IEEE Trans Cloud Comput:1–1. https://doi.org/10.1109/TCC.2019.2923222
24. He D, Ma M, Zeadally S, Kumar N, Liang K (2018) Certificateless public key authenticated encryption with keyword search for industrial internet of things. IEEE Trans Ind Inform 14(8):3618–3627. https://doi.org/10.1109/TII.2017.2771382
25. Li H, Huang Q, Shen J, Yang G, Susilo W (2019) Designated-server identity-based authenticated encryption with keyword search for encrypted emails. Inf Sci 481:330–343. https://doi.org/10.1016/j.ins.2019.01.004
26. Li H, Huang Q, Susilo W (2020) A secure cloud data sharing protocol for enterprise supporting hierarchical keyword search. IEEE Trans Dependable Secure Comput:1–1. https://doi.org/10.1109/TDSC.2020.3027611
27. Xu P, Jin H, Wu Q, Wang W (2013) Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack. IEEE Trans Comput 62(11):2266–2277. https://doi.org/10.1109/TC.2012.215
28. Miao Y, Weng J, Liu X, Choo KKR, Liu Z, Li H (2018) Enabling verifiable multiple keywords search over encrypted cloud data. Inf Sci 465:21–37. https://doi.org/10.1016/j.ins.2018.06.066
29. Zhang X, Xu C, Wang H, Zhang Y, Wang S (2019) Fs-peks: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial internet of things. IEEE Trans Dependable Secure Comput:1–1. https://doi.org/10.1109/TDSC.2019.2914117
30. Li J, Lin X, Zhang Y, Han J (2017) Ksf-oabe: Outsourced attribute-based encryption with keyword search function for cloud storage. IEEE Trans Serv Comput 10(5):715–725. https://doi.org/10.1109/TSC.2016.2542813
31. Sadeghi A-R, Steiner M (2001) Assumptions related to discrete logarithms: Why subtleties make a real difference. In: Pfitzmann B (ed). Advances in Cryptology — EUROCRYPT 2001. Springer, Berlin, Heidelberg. pp 244–261
32. Abdalla M, Bellare M, Rogaway P (2001) The oracle diffie-hellman assumptions and an analysis of dhies. In: Naccache D (ed). Topics in Cryptology — CT-RSA 2001. Springer, Berlin, Heidelberg. pp 143–158
33. Dent AW (2006) A Note On Game-Hopping Proofs. Cryptology ePrint Archive, Report 2006/260. https://eprint.iacr.org/2006/260

Publisher's Note