Journal of Cyber Security
DOI:10.32604/jcs.2021.017217
Article

IoT Services: Realizing Private Real-Time Detection via Authenticated Conjunctive Searchable Encryption

Chungen Xu[1,*], Lin Mei[1], Jinxue Cheng[2], Yu Zhao[1] and Cong Zuo[3]

[1]School of Science, Nanjing University of Science and Technology, Nanjing, 210094, China
[2]China Mobile (Hangzhou) Information Technology Co., Ltd., Hangzhou, 310011, China
[3]Faculty of Information Technology, Monash University, Clayton, 3168, Australia
[*]Corresponding Author: Chungen Xu. Email: xuchung@njust.edu.cn

Received: 24 January 2021; Accepted: 06 April 2021

Abstract: With the rapid development of wireless communication technology, the Internet of Things is playing an increasingly important role in our everyday. The amount of data generated by sensor devices is increasing as a large number of connectable devices are deployed in many fields, including the medical, agricultural, and industrial areas. Uploading data to the cloud solves the problem of data overhead but results in privacy issues. Therefore, the question of how to manage the privacy of uploading data and make it available to be interconnected between devices is a crucial issue. In this paper, we propose a scheme that supports real-time authentication with conjunctive keyword detection (RA-CKD), this scheme can realize the interconnection of encrypted data between devices while ensuring some measure of privacy for both encrypted data and detection tokens. Through authentication technology, connected devices can both authenticate each other's identity and prevent malicious adversaries from interfering with device interconnection. Finally, we prove that our scheme can resist inside keyword guessing attack through rigorous security reduction. The experiment shows that the efficiency of RA-CKD is good enough to be practical.

Keywords: Searchable encryption; conjunctive keyword search; Internet of Things; authentication

1 Introduction

The Internet of Things (IoT), as its name implies, is a kind of network that enables interactive devices to be interconnected and to become an integral part of the Internet [1]. Through the IoT, the physical world is more directly integrated into computer-based systems, resulting in efficiency improvement, economic benefits, and reduced human exertion. Due to the good performance and unlimited storage of cloud computing, uploading IoT data to the cloud has become a solution to the problem of an excessive load of IoT data. However, since most cloud servers in the real world are semi-honest (i.e., they will honestly perform all queries but be curious about the data and the query), uploading data in plaintext to the cloud will pose data privacy issues. One possible approach to this issue is to encrypt the data before outsourcing. Encrypting data does protect the privacy of data, but the usability of data will lose. A feasible solution is to exploit searchable symmetric encryption (SSE) [1], a cryptography primitive that enables users to perform searches over encrypted data. Many researchers have constructed several searchable encryption (SE) schemes to support more functionalities with enhanced efficiency.

Sometimes users want to query multiple keywords within one search operation, e.g., in the medical IoT, doctor query the heartbeat and blood pressure of the patients. To accomplish this goal, Golle et al. [3] were the first to apply conjunctive keyword search technology to an SE scheme and proposed two secret key encryption schemes with conjunctive keyword search. However, their proposal cannot meet the need for public key models. Thus, Park et al. [4] proposed two public key encryption schemes with conjunctive keyword search. Since the space of the keywords used in practice is limited, it is important to consider the keyword guessing attack (KGA) [5], which can be divided into inside KGA (IKGA) and outside KGA based on the identity of the adversary [6].

Existing public key encryption with conjunctive keyword search (PECKS) schemes have paid less attention to resisting IKGA, raising security problems. Furthermore, ensuring the reliability of encrypted data or the identity of the data generator, which is of great significance in the IoT, is a particular cause of concern. For example, a server uses the public key of the data owner (who uploads its encrypted data to the cloud server) to generate a valid but illegal ciphertext and access the data transmission channel to replace the legal ciphertext. When another device submits a detection token to the server, the server performs the detect operation in the illegal ciphertext and the detect token. The wrong result is then fed back to the device. This security risk cannot be ignored, especially when the device plays an important role in areas related to personal safety, such as alarm devices or emergency brake devices. The reason is that once the feedback is wrong, it is likely to cause irreparable consequences. In short, in the IoT field, constructing an IKGA secure PECKS is an urgent issue to be solved.

In this paper, to solve the challenges presented above, we propose a scheme that supports real-time authentication with conjunctive keyword detection (RA-CKD) for the IoT. To realize the real-time detection of multiple keywords, we exploit the conjunctive keyword search technology. Considering the existence of malicious internal adversaries, unlike the general public key encryption scheme, we combine the dual public key and private key pair technique proposed in [7]. We then demonstrate that the proposed RA-CKD scheme is IKGA secure by strict security proof and evaluate the performance of our scheme through experiments. The results show that our scheme is efficient.

Contributions. With the multi-keyword setting, our scheme enables the DM to simultaneously test whether the encrypted data contain multiple keywords while protecting the privacy of original data. Compared with previous works, our scheme not only enables the data manager (DM) to detect the encrypted data but also realizes the identity authentication of both the data owners and the DM. This property makes sure that only legal DM can perform test operation and hence enhance the security of the scheme. We also give a formal proof to show that our scheme can obtain ciphertext and token indistinguishability against IKGA. Since the encryption operation is a one-time job and is completed before detection, in terms of efficiency, our scheme takes only the test process into account. The system performs a total of two bilinear pairing operations and one exponential operation during the test operation, gaining a better balance between storage and efficiency compared with other schemes.

1.1 Related Work

Following seminal work on SSE [2], since the search time is linear with the size of the database, more attention has been paid to improving the efficiency of SSE schemes [8,9]. In addition to the efficiency, the security of SSE schemes has been studied by many scholars. Curtmola et al. presented an enhanced security model and proposed the first semantic secure scheme that could against adaptive adversaries [10]. In recent years, to address security concerns caused by various kinds of attacks, such as file-injection attack, Bost et al. proposed a dynamic SSE scheme that ensures forward security for the leakage caused by the access pattern [11] and later gave several schemes achieving both forward and backward privacy [12]. In addition to the improvements in efficiency and security mentioned above, much work has been done to extend the functionality of the SSE schemes, including dynamic SSE that supports file additions and deletions [13,14], conjunctive SSE that enables users to search multiple keywords [15,16] and fuzzy SSE [17,18].

The PEKS scheme was first proposed by Boneh et al. [19]. Their scheme relies on the need for mail routing, which consists of three parties: the sender, receiver, and mail server. Independent of Boneh's work, in the same year, Waters et al. [20] presented a new approach to constructing searchable encrypted audit logs that can be combined with any of the existing audit schemes to achieve tamper resistance. Then, since no previous work had considered the capability of a stronger adversary who can adaptively query the oracles, Curtmola et al. [10] described an adaptive adversary security model. To solve the security problem of search tokens, Baek et al. [21] creatively proposed the idea of registered keyword search. Based on the work in [21], Rhee et al. [22] constructed a PEKS scheme with a designated tester and proved the security of the search token. In order to achieve IKGA security, Chen et al. [23] proposed a server-aided public key encryption with keyword search scheme. However, it introduced high communication overhead as it requires an extra keyword server to run authentication protocol for keyword ciphertext/trapdoor generation. Then, Huang et al. [7] introduced the notion of public-key authenticated encryption with keyword search to solve the above issues. Except for ensuring the security of PEKS schemes, the development of a more efficient and versatile PEKS is a trend due to the real-life needs. For example, in addition to IoT data privacy, the real requirement is that the scheme must not only satisfy the PEKS (because different IoT devices want to share data between each other) but also support some other functionalities, e.g., multi-keyword, multi-user, etc.

To capture the properties needed in a multi-keyword condition, the concept of conjunctive keyword search (CKS) was proposed [3]. Then work on PECKS started based on the two schemes proposed in [4], subsequently focusing on efficiency improvement [24], with most work adopting a bilinear map or Shamir's secret sharing. Since security is one of the most indispensable elements of a CKS scheme, research on improving the security of the CKS has been ongoing [25,26]. However, some CKS schemes have been proposed to meet IKGA security at the expense of efficiency [27]. Thus, challenges remain with respect to constructing an efficient CKS scheme with IKGA security and enabling verification between the user and sender, which is more suitable for application in an IoT system. In general, SE with different functionalities and improved security and efficiency is always a hot topic in research. To smoothly integrate CKS technology into the IoT, the remaining problems in CKS schemes are still waiting to be solved.

Organizations. The rest of this paper is organized as follows: In Section 2, we give the overview and threat model of our scheme. The scheme and correctness definition are both presented in Section 3; we introduce bilinear pairing, some hardness assumptions and the security models in Section 3 as well. In Sections 4 and 5, we describe our RA-CKD scheme in detail and give the strict security proof. Finally, we present a brief conclusion in Section 7.

2 Problem Statement

2.1 System Overview

The overview in Fig. 1 illustrates how our scheme works during one detection operation. First, the central authority (CA) generates the secret key and the public key pair for data owners and the data manager (DM). Then the data owner outsources its data in encrypted form and generates the index with its secret key and the public key of the DM. Then, the encrypted data and the index are outsourced to the cloud server. When a user wants to detect some conditions of a specific device, it will send the query keywords set, the public key of the device and the positions of the query keywords in the keyword fields to the DM. Once the DM receives the detection request sent by the user, it will generate the detection token with its secret key and the public key of the query device and send the token to the cloud server. The cloud server will detect whether the ciphertext meets the conditions based on the detection token and return the corresponding result to the DM. Finally, the DM transmits the result to the user, who promptly responds based on the result.

- CA: The CA is a fully trusted entity in our scheme, who is responsible for distributing public key and secret key pairs for data owners and the DM.
- Data owners: The data owners are devices that collect data from sensors and encrypt the data with their secret key and the public key of the DM, ensuring that only the DM can generate a valid token. The encrypted data and index are then sent to the cloud.
- Users: In our scheme, users can be devices that want to detect some special conditions of another device's data. They submit the query keywords, the public key of the query device and

the positions of the query keywords in keyword fields to the DM. Then, based on the results feedback from the DM, they promptly take measures.

- Data manager: After receiving the keywords, the public key of the query device and the positions of the query keywords in the keyword fields from users, the DM takes its secret key and the public key of the queried device as input, then generates a detection token and sends it to the cloud server. As the DM receives the detection result from the cloud server, it promptly feedbacks to the users.Cloud server: The cloud server is responsible for detecting over the encrypted database and returning the corresponding ciphertext to the DM based on the token submitted by the DM.

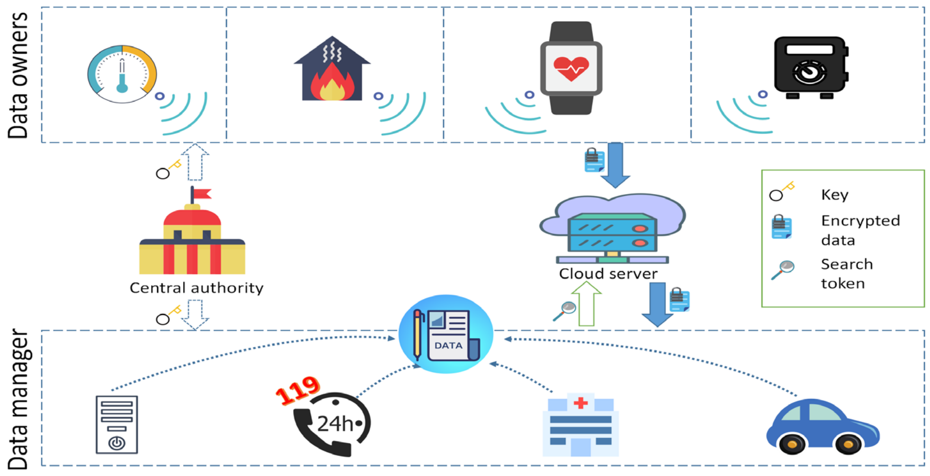

Figure 1: System overview

2.2 Threat Model

In our scheme, the CA is reliable and independent, we believe that it will not collude with other entities during the key derivation process. The cloud server is assumed to be honest-but-curious, that is, the cloud server will honestly perform the search operation and return the results, but it is curious to obtain hidden information from the information it owns. When a device acts as a data owner and uploads encrypted data to the cloud, we consider it to be trusted. When we consider the reliability of the DM and the devices act as user, to simplify the threat model, we combine them into an object. It is not trusted because it will try to obtain unauthorized information. Furthermore, in our scheme, users can collude with the honest-but-curious cloud server to break the privacy of data that is not authorized to them.

3 Preliminaries

In this section, we define the components of our RA-CKD scheme. Then we revisit bilinear pairing and the hardness assumption related to the security proof of our scheme. The definition of the security model is presented at the end of this section.

3.1 Scheme Definition

Definition 3.1: An RA-CKD scheme consists of the following four algorithms:

1. **KeyGen** $(\lambda) \rightarrow (SK_{di}, PK_{di}), (SK_p, PK_p)$: It takes as input the security parameter λ and then outputs public/secret key pairs (SK_{di}, PK_{di}) and (SK_p, PK_p) for the *ith* data owner and DM. The algorithm is run by the CA.

2. **Encrypt** $(W, SK_{di}, PK_p) \rightarrow C$: It takes as input the keyword set W, the ith data owner's secret key SK_{di} and the public key PK_p of the DM and then outputs a conjunctive ciphertext C. The algorithm is run by the ith data owner.

3. **TokGen** $(Q, I, SK_p, PK_{di}) \rightarrow T_Q$: It takes as input the query set Q, the positions of the query keywords in keyword fields I, the DM's secret key SK_p and the public key of the ith data owner and then outputs a detection token T_Q. The algorithm is run by the DM.

4. **Test** $(T_Q, C, PK_p) \rightarrow 0$ or 1: It takes as input the detection token T_Q, the ciphertext C and the public key of the DM and then outputs 1 if the ciphertext C contains the same keyword as the detection token T_Q; otherwise, it outputs 0. The algorithm is run by the cloud server.

An RA-CKD scheme is correct if the **Test** algorithm always returns the correct response on an encrypted database by a legal input that has been generated by **TokGen** algorithm.

3.2 Bilinear Pairing

A mathematical tool, bilinear pairing, is used in our scheme. Due to its non-generality, we introduce its definition in this section.

Let G_1 and G_2 be two cyclic groups with the same prime order p. A bilinear pairing is a map: $e: \mathsf{G}_1 \times \mathsf{G}_1 \rightarrow \mathsf{G}_2$, which satisfies the following three properties:

1. Bilinearity: $\forall a, b \in Z_p$ and $\forall u, v \in \mathsf{G}_1, e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: If g is a generator of G_1, then $e(g, g)$ is a generator of G_2.
3. Computability: Randomly choose two elements from G_1 : g, h , there's a polynomial time algorithm to compute $e(g, h)$.

3.3 Hardness Assumption

In this paper, we consider the IND-IKGA security of our RA-CKD scheme. We introduce the hardness assumption on which the proof of security relies.

First, the Decisional Bilinear Diffie-Hellman (DBDH) problem is stated as follows:

Given a tuple $(g,\ g^a, g^b, g^c, R)$ as inputs, where $g, g^a, g^b, g^c \in \mathsf{G}_1$ and R is randomly chosen from G_2. The DBDH problem in $(\mathsf{G}_1, \mathsf{G}_2)$ is to distinguish R from $e(g, g)^{abc}$.

The advantage for a Probabilistic Polynomial-Time (PPT) adversary A to solve the above problem is $Adv_{DBDH}(\mathsf{A}) =| \Pr[\mathsf{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1] - \Pr[\mathsf{A}(g, g^a, g^b, g^c, R) = 1] |$.

Definition 3.2 (**DBDH Assumption**)**:** We say that the DBDH assumption holds if for any PPT adversary A , $Adv_{DBDH}(\mathsf{A})$ is negligible.

Then, the modified Decisional Linear (mDLIN) problem is stated as follows:

Given a tuple $(g, g^a, g^b, g^{ac}, g^{s/b}, r)$, where $g, g^a, g^b, g^{ac}, g^{s/b} \in \mathsf{G}_1$ and r is randomly chosen from Z_p. The mDLIN problem is to distinguish g^{c+s} from g^r.

The advantage for a PPT adversary A to solve the above problem is as follows:

$$Adv_{mDLIN}(\mathsf{A}) = |\Pr[\mathsf{A}(g, g^a, g^b, g^{ac}, g^{s/b}, g^{c+s}) = 1] - \Pr[\mathsf{A}(g, g^a, g^b, g^{ac}, g^{s/b}, g^{c+s}) = 1]|$$

Definition 3.3 (**mDLIN Assumption**)**:** We say that the mDLIN assumption holds if for any PPT adversary A , $Adv_{mDLIN}(\mathsf{A})$ is negligible.

3.4 Security Model

The security of the RA-CKD scheme is a natural extension of the security of the PAEKS [7]. The difference is the chosen of keyword in [7] is a single keyword, which is a keyword set in our scheme. Intuitively, we require, an RA-CKD scheme to guarantee that any two different encrypted keyword sets (and detection tokens) cannot be distinguished by an internal adversary, even if the adversary performs KGA. For the sake of explanation, here, we refer to the data owner and DM as the sender and receiver, respectively. The security of the keyword ciphertext and token privacy in our scheme is defined as follows:

Definition 3.4 (**CT-IND-CKA Game**)**:** Let Π = (**KeyGen**, **Encrypt**, **TokGen**, **Test**) be a probabilistic secure detection scheme over security parameter λ , A be the adversary, B be the challenger, and W be the keyword space.

- **Setup:** The **KeyGen** algorithm is executed by B to generate the sender's public key PK_s and the receiver's public key PK_r. Then B sends the public key pair (PK_s, PK_r) to A .
- **Query phase 1:** In this phase, adversary A adaptively makes queries to ciphertext oracle O_C and token oracle O_T.
- **Challenge:** Once A determines to end query phase 1, she will choose two different keyword sets W_0 and W_1, and submit them to B as her challenge. The only restriction is that the ciphertexts of these two keyword sets cannot be queried in phase 1. Then, B will randomly select a bit $b \in \{0,1\}$ and produce a challenge ciphertext CT_{W_b} that is returned to A .
- **Query phase 2:** In this phase, adversary A adaptively makes queries to ciphertext oracle O_C and token oracle O_T. Notice that any subset of W_0 and W_1 cannot be submitted to the oracles.
- **Guess:** A outputs her guess $b' \in \{0,1\}$. We say A wins the game if and only if $b' = b$.

We define the advantage of A in the CT-IND-CKA game as $Adv^{CT}(\mathsf{A}) = |\Pr[b' = b] - 1/2|$.

Definition 3.5 An RA-CKD scheme satisfies the CT-IND-CKA security if and only if the $Adv^{CT}(\mathsf{A})$ defined above is negligible for any polynomial-time adversary A .

Definition 3.6 (TO-IND-CKA Game): Due to space limitation, we only show the parts that are different from Definition 3.4, which does not affect the understanding of our security game. In **TO-IND-CKA game**, all are the same as **CT-IND-CKA game**, except for replacing the ciphertext challenge to token challenge in Challenge phase.

We define the advantage of A in the TO-IND-CKA game as $Adv^{DT}(\mathsf{A}) = |\Pr[b' = b] - 1/2|$.

Definition 3.7 An RA-CKD scheme satisfies the TO-IND-CKA security if and only if the $Adv^{TO}(\mathsf{A})$ defined above is negligible for any polynomial-time adversary A .

4 Our Construction

In this section, we present the details of our RA-CKD scheme. We start from the description of the four algorithms that compose our scheme.

KeyGen: Taking security parameter λ as input, the CA performs the following steps. 1). Pick two cyclic groups of prime order p : G_1 and G_2 , choose a bilinear map $e: \mathsf{G}_1 \times \mathsf{G}_1 \to \mathsf{G}_2$, select a generator $g \in \mathbb{G}_1$ and a hash function $H:\{0,1\}^* \to \mathbb{G}_1$. 2). Randomly choose two numbers x, y from Z_p and let $SK_s = x$ and $SK_r = y$ be the secret key of the data owner and DM, respectively. 3). Compute $PK_s = g^x$ and $PK_r = g^y$, then publish both. 4). Send the secret key and public key pair (SK_s, PK_s) and (SK_r, PK_r) to the data owner and DM through a secure channel.**Encrypt**: In this phase, given the keywords set W, the secret key SK_s of the data owner and the public key PK_r of the DM, the data owner executes the following steps. 1). Select a random number k from Z_p. 2). Compute $C_1 = H(w_1)^x \cdot g^k, C_2 = H(w_2)^x \cdot g^k, \ldots, C_m = H(w_m)^x \cdot g^k$ and $V = g^{yk}$ in turn. 3). Let $C = (C_1, C_2, \ldots, C_m, V)$ and send it to the cloud server.

TokGen: To better clarify the algorithm, here, we unify the user-initiated query step into the token generation step performed by the DM. This operation is rational as the user sends the query to the MD in plaintext; thus, the query can be directly used as part of the input for the DM to generate the detection token. The inputs of this phase are the query set $Q = (q_1, q_2, \ldots, q_t)$, the positions of the keywords in keyword fields I, the secret key SK_r of the DM and the public key PK_s of the data owner. Then the DM computes $T_W = e(\prod_{i=1}^{t} H(q_i)^y, g^x)$, $T_Q = (T_W, I)$ and sends T_Q to the cloud server.

Test: Once the cloud server receives the detection token T_Q from the DM, it will take ciphertext C and public key PK_r of the DM as inputs and then executes the steps as follows: 1). Compute $C' \leftarrow C_{I_1} \cdot C_{I_2} \cdot \ldots \cdot C_{I_t}$, where $I_i \in I (i = 1, \ldots, t)$. 2). Compute $U = g^t$, where t is the number of elements in I. 3). Compute $V_L = T_W \cdot e(U, V)$ and $V_R = e(C', g^y)$, where V is the last element of C. 4). Verify $V_L = V_R$, if it holds, then outputs "1"; otherwise, outputs "0".

Correctness: Let $(SK_r, PK_r) = (x, g^x)$ and $(SK_r, PK_r) = (y, g^y)$ be the secret key and public key pairs of the receiver and sender, respectively. Let W be the keywords set contained in ciphertext C and Q be the query keywords in token T_Q. Then, we can verify the correctness of our scheme as follows:

$$V_L = T_W \cdot e(g^t, V) = e(\prod_{i=1}^{t} H(q_i)^x \cdot g^k, g^y), \quad V_R = e(\prod_{i=1}^{t} C_{I_i}, g^y) = e(\prod_{i=1}^{t} H(w_{I_i})^x \cdot g^k, g^y)$$

Then, if the query keywords set $(q_1, q_2, \ldots, q_t)$ is the same as the keywords set $(w_{I_1}, w_{I_2}, \ldots, w_{I_t})$, i.e., $q_i = w_{I_i}$, then $H(q_i) = H(w_{I_i})$; thus, $V_L = V_R$ holds. In summary, our scheme is correct.

5 Security Analysis

In this section, we show that our proposed scheme is IKGA secure. First, we demonstrate that our scheme satisfies the indistinguishability of ciphertext under the mDLIN assumption. Then, the indistinguishability of the token is proved under the DBDH assumption. Based on the above two steps, the IKGA security of our scheme is proved. We introduce the proof in detail as follows:

Theorem 5.1 Our RA-CKD scheme is semantically secure against IKGA in the random oracle model, which is based on the DBDH assumption and mDLIN assumption.

The proof of Theorem 5.1 relies on Lemmas 5.1 and Lemmas 5.2, which are defined below.

Lemma 5.1 If there exists an adversary $\mathcal{A}_{CT}$ that can break the CT-IND-CKA security of our scheme with a non-negligible advantage ε, then we can solve the mDLIN problem with a PPT algorithm A_{mDLIN}.

Proof. Given an instance of the mDLIN problem: $(g, g^a, g^b, g^{ac}, g^{s/b}, Z)$, algorithm A_{mDLIN} tries to distinguish g^{c+s} from Z. Let d be a bit such that $d=0$ if $Z=g^{c+s}$, and $d=1$ if Z is random. Before stating the interaction between the adversary and the algorithm, we need to give three assumptions to ensure the simplicity of our proof. 1). The adversary never issues a repeat query to the oracles. 2). The adversary issues a query of keywords set W to O_C and O_T after it has issued W to O_H. 3). The queries that the adversary issues to the hash oracle O_H, ciphertext oracle O_C and token oracle O_T can at most be q_H, q_C and q_T, respectively.

Then, A_{mDLIN} interacts with A_{CT} as follows:

Setup. Algorithm A_{mDLIN} first sets the public parameters: $PP=(\mathsf{G}_1, \mathsf{G}_2, e, p, g)$; then, it randomly chooses two numbers: $a, b \in Z_p$ and sets $PK_s = g^a$, $PK_r = g^b$. Finally, it sends the PP and (PK_s, PK_r) to A_{CT}.

Query phase 1. A_{CT} adaptively queries the oracles. Algorithm A_{mDLIN} responds as follows:

- Hash Oracle O_H: Given a keyword set $W_i=(w_{i1}, w_{i2}, \ldots, w_{it})$, A_{mDLIN} first maintains an empty list: $L_H=\langle w_{ij}, h_{ij}, x_{ij}, y_{ij}\rangle$ at first. Then, for each $w_{ij} \in W(j=1,\ldots,t)$, it randomly selects $x_{ij} \in Z_p$, and tosses a biased coin y_{ij}, which satisfies $Pr[y_{ij}=0]=\delta$. The value of δ will be determined in the Guess step. If $y_{ij}=0$, then A_{mDLIN} sets $h_{ij}=g^{s/b}\cdot g^{x_{ij}}$ and sets $h_{ij}=g^{x_{ij}}$ otherwise. It then adds the tuple $\langle w_{ij}, h_{ij}, x_{ij}, y_{ij}\rangle$ to list L_H and returns $\{H(w_{ij})=h_{ij} | j=1,\ldots,t\}$ as the hash value of W_i to A_{CT}.
- Ciphertext Oracle O_C: Given the keyword set $W_i=(w_{i1}, w_{i2}, \ldots, w_{it})$, A_{mDLIN} retrieves the tuple $\langle q_{ij}, h_{ij}, x_{ij}, y_{ij}\rangle$ from list L_H. If $\forall j \in \{1,\ldots,t\}, y_{ij}=0$, then A_{mDLIN} aborts and outputs a random d'. Otherwise, it randomly chooses $r_i \in Z_p$ and outputs the ciphertext as follows:

$$C_{W_i}=(C_{i1}, C_{i2}, \ldots, C_{it}, V_i)=(H(w_1)^a\cdot g^{r_i}, H(w_2)^a\cdot g^{r_i}, H(w_t)^a\cdot g^{r_i}, g^{br_i})$$

The ciphertexts C_{W_i} is then sent to A_{CT}.

- Token Oracle O_T : Given the query set $\{Q_i, I_i\}$, where $Q_i=(q_{i1}, q_{i2}, \ldots, q_{it})$ and $I_i=(I_{i1}, I_{i2}, \ldots, I_{it})$, A_{mDLIN} retrieves tuple $\langle q_{ij}, h_{ij}, x_{ij}, y_{ij}\rangle$ from list L_H. If $\forall j \in \{1,\ldots,t\}, y_{ij}=0$, then A_{mDLIN} aborts and outputs a random d'. If $\forall j \in \{1,\ldots,t\}, y_{ij}=1$, then A_{mDLIN} computes the token:

$$T_{Q_i}=\prod_{j=1}^{t} e(g^a, g^b)^{x_{ij}}=\prod_{j=1}^{t} e(g^{bx_{ij}}, g^a)=e(\prod_{j=1}^{t} H(q_{ij})^b, g^a).$$

It is clear that T_{Q_i} is a correct token for query Q_i. A_{mDLIN} then returns T_{Q_i} to A_{CT}.

Challenge. At some point, A_{CT} chooses two different query sets Q_0 and Q_1 that have never been queried in query phase 1. Here, Q_0 and Q_1 are two sets of $\{q_{01}, q_{02}, \ldots, q_{0t}, I_{01}, I_{02}, \ldots, I_{0t}\}$ and

$\{q_{11}, q_{12}, \ldots, q_{1t}, I_{11}, I_{12}, \ldots, I_{1t}\}$ respectively. Then, it submits these two query sets to adversary A_{mDLIN} The A_{mDLIN} first retrieves tuple $\langle q_{0j}, h_{0j}, x_{0j}, y_{0j} \rangle$ from list L_H .

- If $\exists j,k \in \{1,\ldots,t\}, s.t. y_{0j} = y_{1k} = 1$, then A_{mDLIN} aborts and outputs a random d'.
- If $\forall j \in \{1,\ldots,t\}, y_{0j} = 0$, or $\forall j \in \{1,\ldots,t\}, y_{1j} = 0$, let $\hat{d}$ be a bit such that $y_{\hat{d}j} = 0$; then, we obtain $h_{\hat{d}j} = g^{s/b} \cdot g^{x_{\hat{d}j}} = g^{(s+bx_{\hat{d}j})/b}$. Randomly choose a number from the set $\{1, 2, \ldots, t\}$ and write it as $\hat{t}$. Next, A_{mDLIN} computes the ciphertext as follows:

  $$C^* = (C_1^*, \ldots, C_t^*, V^*) = (Z \cdot g^{x_{\hat{d}1}} \cdot g^{bx_{\hat{d}1}}, \ldots, Z \cdot g^{x_{\hat{d}t}} \cdot g^{bx_{\hat{d}t}}, g^{ac} \cdot (g^a)^{x_{\hat{d}\hat{t}}})$$

  If $Z = g^{c+s}$, then for each $j \in \{1,\ldots,t\}, C_j^* = g^{c+s} \cdot g^{x_{\hat{d}j}} \cdot g^{bx_{\hat{d}j}} = g^{(c+x_{\hat{d}1})+(s+bx_{\hat{d}j})}$ and $V^* = g^{a(c+x_{\hat{d}\hat{t}})}$, the value of $c + x_{\hat{d}\hat{t}}$ is random from the perspective of A_{CT}. If Z is randomly selected from G_1, then so is C_i^*. V^* is random from the perspective of A_{CT} because of the randomness of $x_{\hat{d}\hat{t}}$.

A_{mDLIN} returns C^* to the adversary A_{CT}.

Query phase 2. A_{CT} adaptively makes queries to the oracles as in phase 1, with the restriction that it cannot issue Q_0 and Q_1 to O_C or O_T.

Guess. A_{CT} outputs its guess $\hat{d}' \in \{0,1\}$. If $\hat{d}' = \hat{d}$, then A_{mDLIN} outputs $d' = 0$ and outputs $d' = 1$ otherwise.

Now, we first discuss the probability that A_{mDLIN} aborts during the whole process, which we denote by Pr_{abt}. The computation approach is given as follows:

1) $y_{ij} = 0$, in the simulation of O_C and O_T. Denote this event by abt_1. The probability that A_{mDLIN} continue in this case is $Pr[\overline{abt_1}] = (1-\delta)^{t(q_T+q_C)}$.
2) $\exists j,k \in \{1,\ldots,t\}, s.t. y_{0j} = y_{1k} = 1$ in the challenge phase. Denote this event by abt_2. The probability that A_{mDLIN} continue in this case is $Pr[\overline{abt_2}] = 1-(1-\delta^t)^2$.

Thus, the probability that A_{mDLIN} continues in the whole process is bounded by the following:

$$\Pr[a\bar{b}t] = \Pr[a\bar{b}t_1] \cdot \Pr[abt_2] = 1-(1-\delta^t)^2 = (1-\delta)^{t(qT+qC)} \cdot \delta^t \cdot (2-\delta^t)$$

It is easy to obtain

$$Pr[\overline{abt}] = (1-\delta)^{t(q_T+q_C)} \cdot \delta^t \cdot (2-\delta^t) \geq (1-\delta)^{t(q_T+q_C)} \cdot \delta^t$$

Additionally, when $\delta = 1/(q_T + q_C + 1)$, the value of $Pr[\overline{abt}]$ is maximum:

$$Pr[\overline{abt}] \geq ((q_T + q_C)/q_T + q_C + 1)^{t(q_T+q_C)} \cdot (1/q_T + q_C + 1)^t$$

which is almost equal to $1/e^t \cdot (q_T + q_C + 1)^t$, and thus non-negligible.

It is easily found that if A_{mDLIN} continues in the whole process, then the perspective of A_{CT} is the same as it is in real attack. Therefore, under this circumstance, if A_{CT} successfully breaks the ciphertext privacy of our scheme, A_{mDLIN} also succeeds in distinguishing g^{c+s} from a random number of G_1. Then, we get the probability that $\mathcal{A}_{mDLIN}$ succeeds in guessing the bit d :

$\Pr[d' = d] = \Pr[d' = d \wedge abt] + \Pr[d' = d \wedge a\bar{b}t] = 1/2 + \varepsilon \cdot \Pr[a\bar{b}t]$

If ε is non-negligible, then so is $|\ Pr[d' = d] - 1/2|$. The proof is completed.

Lemma 5.2. If there exists an adversary $\mathcal{A}_{TO}$ that can break the TO-IND-CKA security of our scheme with a non-negligible advantage ε, then we can solve the DBDH problem with a PPT algorithm $\mathcal{A}_{DBDH}$.

Proof. DBDH problem: (g, g^a, g^b, g^c, Z), the algorithm $\mathcal{A}_{DBDH}$ tries to distinguish g^{abc} from Z. Let d be a bit such that $d = 0$ if $Z = g^{abc}$ and $d = 1$ if Z is random. The assumptions used in the proof are the same as those in Lemma 5.1. Then, $\mathcal{A}_{DBDH}$ interacts with $\mathcal{A}_{TO}$ as follows:

Setup. Algorithm $\mathcal{A}_{DBDH}$ first sets the public parameters $PP = (\mathbb{G}_1, \mathbb{G}_2, e, p, g)$, and then randomly chooses two numbers $a, b \in Z_p$ and sets $PK_s = g^a$, $PK_r = g^b$. Finally, it sends ***PP*** and $*(K_s, PK_r)$ to adversary $\mathcal{A}_{TO}$.

Query phase 1. $\mathcal{A}_{TO}$ adaptively queries the oracles. Algorithm $\mathcal{A}_{DBDH}$ responds as follows:

- Hash Oracle $\mathcal{O}_H$: Given keyword set $W_i = (w_{i1}, w_{i2}, \ldots, w_{it})$, $\mathcal{A}_{DBDH}$ first maintains an empty list: $L_H = \langle w_{ij}, h_{ij}, x_{ij}, y_{ij} \rangle$ at first. Then, for each $w_{ij} \in W(j = 1, \ldots, t)$, it randomly selects $x_{ij} \in Z_p$, and tosses a biased coin y_{ij}, which satisfies $Pr[y_{ij} = 0] = \delta$. The value of δ is determined in the **Guess** step. If $y_{ij} = 0$, $\mathcal{A}_{DBDH}$ sets $h_{ij} = g^c \cdot g^{x_{ij}}$, otherwise sets $h_{ij} = g^{x_{ij}}$. It then adds tuple $\langle w_{ij}, h_{ij}, x_{ij}, y_{ij} \rangle$ to list L_H and returns $\{H(w_{ij}) = h_{ij} |\ j = 1, \ldots, t\}$ as the hash value of W_i to $\mathcal{A}_{TO}$.
- Ciphertext Oracle $\mathcal{O}_C$: $\mathcal{O}_C$ responds to the adversary's queries as in the proof of Lemma 5.1.
- Token Oracle $\mathcal{O}_T$: $\mathcal{O}_T$ responds to the adversary's queries as in the proof of Lemma 5.1.

Challenge. At some point, $\mathcal{A}_{TO}$ chooses two different query sets Q_0 and Q_1 that have never been queried in the query phase 1. Here, Q_0 and Q_1 are two sets of $\{q_{01}, q_{02}, \ldots, q_{0t}, I_{01}, I_{02}, \ldots, I_{0t}\}$ and $\{q_{11}, q_{12}, \ldots, q_{1t}, I_{11}, I_{12}, \ldots, I_{1t}\}$ respectively. Then, it submits these two query sets to adversary $\mathcal{A}_{DBDH}$.

The $\mathcal{A}_{DBDH}$ first retrieves tuple $\langle q_{0j}, h_{0j}, x_{0j}, y_{0j} \rangle$ from list L_H.

- If $\exists j, k \in \{1, \ldots, t\}, s.t. y_{0j} = y_{1k} = 1$, then $\mathcal{A}_{DBDH}$ aborts and outputs a random d'.
- If $\forall j \in \{1, \ldots, t\}, y_{0j} = 0$, or $\forall j \in \{1, \ldots, t\}, y_{1j} = 0$, let $\hat{d}$ be the bit such that $y_{\hat{d}j} = 0$, then we get $h_{\hat{d}j} = g^c \cdot g^{x_{\hat{d}j}} = g^{c + x_{\hat{d}j}}$. Next, $\mathcal{A}_{DBDH}$ computes the trapdoor $T^* = \prod_{j=1}^{t} Z \cdot e(g^a, g^b)^{x_{\hat{d}j}}$. If $Z = e(g, g)^{abc}$, then $T^* = \prod_{j=1}^{t} e(g^{c + x_{\hat{d}j}}, g^{ab}) = \prod_{j=1}^{t} e(h_{\hat{d}j}, g^{ab})$. If Z is randomly selected from $\mathbb{G}_1$, then so is T^*. $\mathcal{A}_{DBDH}$ returns C^* to the adversary $\mathcal{A}_{TO}$.

Query phase 2. Adversary $\mathcal{A}_{TO}$ adaptively makes the same queries to the oracles as in phase 1, with the restriction that it cannot issue Q_0 and Q_1 to $\mathcal{O}_C$ or $\mathcal{O}_T$.

Guess. $\mathcal{A}_{TO}$ outputs its guess $\hat{d}' \in \{0,1\}$. If $\hat{d}' = \hat{d}$, then $\mathcal{A}_{DBDH}$ outputs $d' = 0$ and outputs $d' = 1$ otherwise.

Now we first discuss the probability that A_{DBDH} aborts during the whole process, which we denote by Pr_{abt}. The value of Pr_{abt} is the same as that in the proof of Lemma 5.1. Therefore, when $\delta = 1/(q_T + q_C + 1)$, the probability $Pr[\overline{abt}]$ takes the maximum value:

$$Pr[\overline{abt}] \geq ((q_T + q_C)/q_T + q_C + 1)^{t(q_T+q_C)} \cdot (1/q_T + q_C + 1)^t$$

which is almost equal to $1/e^t \cdot (q_T + q_C + 1)^t$ and thus non-negligible.

It is easily found that if A_{DBDH} continues in the whole process, then the perspective of A_{TO} is the same as that in real attack. Therefore, under this circumstance, if A_{TO} successfully breaks the ciphertext privacy of our scheme, then A_{DBDH} also succeeds in distinguishing g^{abc} from a random number of G_1. Then, we obtain the probability that A_{DBDH} succeeds in guessing bit d:

$$\Pr[d' = d] = \Pr[d' = d \wedge abt] + \Pr[d' = d \wedge a\overline{b}t] = 1/2 + \varepsilon \cdot \Pr[a\overline{b}t]$$

If ε is non-negligible, then so is $|\ Pr[d^{'} = d] - 1/2|$. The proof is completed.

6 Performance Analysis

In this section, to demonstrate the efficiency and effectiveness of our scheme, we first compare the RA-CKD scheme with previous works related to CKS [4,26,27] and the PAEKS [7] scheme that motivates us. In Tab. 1, we list the theoretical computation overhead of the Encypt, TokGen and Test protocols. Note that m denotes the number of keywords during encryption, t denotes the number of keywords during token generation, E denotes a modular exponentiation, H denotes a collision-resistant hash function and P denotes a bilinear pairing.

Table 1: Performance Comparison

Schemes	Ciphertext	Token	Test
[4]	(m + 2)E + m(H + P)	t(E + H)	E + P
[7]	m(3E+H)	t(E + H + P)	2P
[26]	(m + 4)E + (m + 3)H + P	3E + (t + 2)H + P	3(P + E)
[27]	(m2 + 2m + 2)E	2(t + 5)E	(mt + t + 1)E + 3P
Ours	(m + 2)E + mH	tE + H + P	E + 2P

Now, we discuss the performance of our proposed scheme through a program written in Java. We conduct all the experiments on a PC with an Intel Core i7-8700 3.20 GHz CPU with 16G RAM. In our experiments, the keywords are extracted from 1000 e-mail data files. The data table is constructed by the order index. First, for a given keyword field with fixed keywords, we study the effect of the number of files “nf” on computation efficiency. As shown in the left figure of Fig. 2, we find that when the size of the keyword fields (i.e., “kf”) is fixed, the time cost of the Encrypt protocol will increase with the size of “nf”. Because KeyGen is a one-time job, we omit its cost in our experiment.

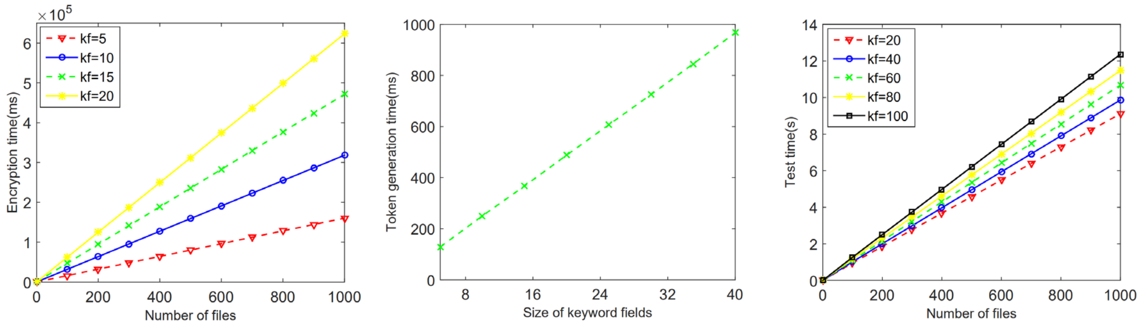

Figure 2: Experiment performance

To further explore the relationship between the size of the keyword fields and computation efficiency, we change the value of "kf" and conduct several experiments on the **Encrypt** and **TokGen** protocols. The left figure of Fig. 2 indicates that if "kf" is larger, the time cost of encryption will increase when "nf" is fixed. When kf = 5, it takes approximately 32 s to generate 200 encrypted files. From the middle figure of Fig. 2, it is clear that the time cost of token generation is linear with the value of "kf". When kf = 10, it takes approximately 250 ms to generate tokens for any number of files.

Finally, we focus on the time cost of the **Test** protocol, which is the most important issue in a real-time system. As shown in the right figure of Fig. 2, the test time is dominated by the values of "nf" and "kf", with the value of "kf" having a lower impact on it than the value of "nf". We find that the test time becomes larger with the increase in "kf". It costs only 12 s to test in one thousand files when the value of "kf" is 100. With respect to the effect of the value of "nf", the greater the number of files involved in a **Test** protocol is, the greater the cost time. When kf = 20, it takes an average of 0.9 s on to fulfil one test over 100 files, which is sufficient for application in an IoT system.

7 Conclusion

In this paper, we construct a real-time system to protect the data privacy of the outsourced IoT data because the amount of data has been rapidly increasing for a broader application range of IoT devices. To ensure the usability of the encrypted data, we developed an efficient detection technique to support the detection of multiple conditions at one time. Specifically, in the proposed scheme, we employ CKS to realize efficient detection. Then, a special approach (i.e., in which the secret key is used in both the **Enc** and **TokGen** protocols) is applied to obtain authentication between the data owner and user. It is proved that if the mDLIN and DBDH problems hold, then our scheme is IKGA secure. Finally, experiments on an encrypted database show that our scheme is practical and feasible for the IoT.

Funding Statement: This work is supported by the National Natural Science Foundation of China (No. 62072240) and the National Key Research and Development Program of China (No. 2020YFB1804604).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IOT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] D. X. Song, D. A. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. S&P*, Berkeley, California, USA, pp. 44–55, 2000.

[3] P. Golle, J. Staddon and B. R. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. ACNS*, Yellow Mountain, China, pp. 31–45, 2004.

[4] D. J. Park, K. Kim and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *Proc. WISA*, Jeju Island, Korea, pp. 73–86, 2004.

[5] J. W. Byun, H. S. Rhee, H. Park and D. H. Lee, “Off-line keyword guessing attacks on recent keyword search schemes over encrypted data,” in *Proc. SDM*, Seoul, Korea, pp. 75–83, 2006.

[6] 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.

[7] Q. Huang and H. Li, “An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks,” *Information Sciences*, vol. 403, pp. 1–14, 2017.

[8] Y. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” *in Proc. ACNS*, New York, NY, USA, pp. 442–455, 2005.

[9] E. Goh, “Secure indexes,” [Online]. Available: IACR Cryptology ePrint Archive, 2003.

[10] R. Curtmola, J. A. Garay, S. Kamara and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and effficient constructions,” in *Proc. CCS*, Alexandria, VA, USA, pp. 79–88, 2006.

[11] R. Bost, “Σ o φ oς: Forward secure searchable encryption,” in *Proc. SIGSAC*, Vienna, Austria, pp. 1143–1154, 2016.

[12] R. Bost, B. Minaud and O. Ohrimenko, “Forward and backward private searchable encryption from constrained cryptographic primitives,” in *Proc. CCS*, Dallas, TX, USA, pp. 1465–1482, 2017.

[13] S. Kamara, C. Papamanthou and T. Roeder, “Dynamic searchable symmetric encryption,” in *Proc. CCS*, Raleigh, NC, USA, pp. 965–976, 2012.

[14] L. Xu, C. Xu, J. K. Liu, C. Zuo and P. Zhang, “A multi-client dynamic searchable symmetric encryption system with physical deletion,” in *Proc. ICICS*, Beijing, China, pp. 516–528, 2017.

[15] L. Ballard, S. Kamara and F. Monrose, “Achieving efficient conjunctive keyword searches over encrypted data,” in *Proc. ICICS*, Beijing, China, pp. 414–426, 2005.

[16] H. He, J. Zhang, P. Li, Y. Jin and T. Zhang, “A lightweight secure conjunctive keyword search scheme in hybrid cloud,” *Future Generation Computer Systems*, 2018.

[17] Z. Fang, J. Wang, B. Wang, J. Zhang and Y. Shi, “Fuzzy search for multiple Chinese keywords in cloud environment,” *Computers, Materials & Continua*, vol. 60, no. 1, pp. 351–363, 2019.

[18] X. Li, F. Li, J. Jiang and X. Mei, “Paillier-based fuzzy multi-keyword searchable encryption scheme with order-preserving,” *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1707–1721, 2020.

[19] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano, “Public key encryption with keyword search,” in *Proc. EUROCRYPT*, Interlaken, Switzerland, pp. 506–522, 2004.

[20] B. R. Waters, D. Balfanz, G. Durfee and D. K. Smetters, “Building an encrypted and searchable audit log,” in *Proc. NDSS*, San Diego, California, USA, 2004.

[21] J. Baek, R. Safavi-Naini and W. Susilo, “Public key encryption with keyword search revisited,” in *Proc. ICCSA*, Perugia, Italy, pp. 1249–1259, 2008.

[22] 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,” *Journal of Systems and Software*, vol. 83, no. 5, pp. 763–771, 2010.

[23] R. Chen, Y. Mu, G. Yang, F. Guo, X. Huang, X. Wang and Y. Wang, “Server-aided public key encryption with keyword search,” *IEEE Trans. Information Forensics and Security*, vol. 11, no. 12, pp. 2833–2842, 2016.

[24] J. W. Byun, D. H. Lee and J. Lim, “Efficient conjunctive keyword search on encrypted data storage system,” in *Proc. EuroPKI*, Turin, Italy, pp. 184–196, 2006.

[25] W. Sun, X. Liu, W. Lou, Y. T. Hou and H. Li, “Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data,” in *Proc. of INFOCOM*, Kowloon, Hong Kong, pp. 2110–2118, 2015.

[26] Y. Lu, G. Wang, J. Li and J. Shen, “Efficient designated server identity-based encryption with conjunctive keyword search,” Annales des Telecommunications, vol. 72, no. 5-6, pp. 359–370, 2017.

[27] L. Xu, J. Li, X. Chen, W. Li, S. Tang and H. Wu, “Tc-pedcks: Towards time controlled public key encryption with delegatable conjunctive keyword search for internet of things,” *Journal of Network and Computer Applications*, vol. 128, pp. 11–20, 2019.