Comprehensive Study on Public-Key Searchable Encryption Scheme against Inside Keyword Guessing Attacks

Mrs. Nithya Ramakrishna #1, Mr. S.V. Sandeep #2, Mr. T. Akhil krishna #3, Mr. B. Koteswara Rao #4, Mr. N. Raja Pavan Kumar Reddy #5, Mr. M. Kiran Kumar #6

#1 Assistant Professor, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#2 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#3 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#4 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#5 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#6 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)

***Abstract*:** For a smart healthcare system, a cloud based paradigm with numerous user terminals is to support and improve more reliable, convenient, and intelligent services. Considering the resource limitation of terminals and communication overhead in cloud paradigm, we propose a hybrid IoT-Fog-Cloud framework. In this framework, we deploy a geo-distributed fog layer at the edge of networks. The fogs can provide the local storage, sufficient processing power, and appropriate network functions. For the fog-based healthcare system, data confidentiality, access control, and secure searching over ciphertext are the key issues in sensitive data. Furthermore, how to adjust the storage and computing requirements to meet the limited resource is also a great challenge for data management. To address these, we design a lightweight keyword searchable encryption scheme with fine-grained access control for our proposed healthcare related IoT-Fog-Cloud framework. Through our design, the users can achieve a fast and efficient service by delegating a majority part of the workloads and storage requirements to fogs and the cloud without extra privacy leakage. We prove our scheme satisfies the security requirements and demonstrate the excellent efficiency through experimental evaluation.

Key Words: Searchable public key encryption, certificate-based cryptography, implicit authentication

I. INTRODUCTION

In conventional public key cryptography (PKC), anyone possesses a pair of public and private keys. Because the keys have no connection to the user's identity, a trusted public key infrastructure (PKI) has to be employed for vouching the relation between a public key and an identity by a digital certificate. But, the requirement of PKI certificates is regarded as the major impediment in the deployment of conventional public key cryptosystems.

To remove the heavy burden caused by cumbersome certificate management, Shamir [1] presented identity based cryptography (IBC) in Crypto'84. The merit of IBC is that it eliminates the requirement for PKI certificates, because anyone can use his/her personal identity as his/her public key. However, IBC inherently suffers from the key escrow problem, due to the fact that a completely-trusted private key generator is employed for issuing a private key for each user in the system. Moreover, the private keys should be conveyed to users via secure channel, which leads to the private key distribution problem.

To address the key escrow problem, Al- Riyami and Paterson [2] put forward the notion of certificate less public key cryptography (CLPKC) in Asia crypt' 03. In a CLPKC system, each user should combine a partial private key issued by a key generation center with a secret value of his/her choice to produce his/her private key. In this way, the key generation center does not know the user's private key, and thus CLPKC avoids the key escrow problem.

However, the key generation center should distribute the partial private keys to the users secretly. Therefore, CLPKC has the key distribution problem, which also leads to the requirement of secure channel. In Eurocrypt'03, Gentry [3] presented a practical public key cryptographic primitive named certificate-based cryptography (CBC). This primitive lies between IBC and conventional PKC, but offers an interesting and useful balance. In a CBC system, a user should first produce a pair of public and private keys independently. Then, the user submits his/her identity information and public key to a trusted certificate authority (CA) to apply for a certificate. Unlike the PKI certificates in conventional PKC, each certificate in CBC is merely pushed to its holder

and acts as a partial decryption key or a partial signing key.

As introduced in [3], this interesting property offers an implicit authentication function so that a user requires both his/her private key and certificate to execute the decryption/signing tasks, while the others need not be concerned about this user's certificate status. Therefore, the implicit authentication mechanism enables CBC.

TABLE 1
PROPERTIES OF FOUR DIFFERENT PUBLIC KEY CRYPTOGRAPHIC PRIMITIVES

Primitives	Certificateless	Implicit authentication	No key escrow	No key distribution	No secure channel
Traditional PKC	no	no	yes	yes	yes
IBC	yes	yes	no	no	no
CL-PKC	yes	yes	yes	no	no
CBC	no	yes	yes	yes	yes

to avoid the problem of third-party queries for the certificate status and predigest the complicated certificate management in conventional PKI-assisted PKC systems. In addition, CBC addresses the key escrow problem (because all users' private keys are unknown to CA) and the key distribution problem (because the certificates are pushed to their holders publicly).

In recent years, CBC has attracted much attention in academia and a lot of cryptographic schemes in the setting of CBChave been proposed [4-13]. TABLE 1 summarizes the properties of the abovementioned public key cryptographic primitives. As an extension of standard public key encryption (PKE), searchable PKE (SPKE) [14] offers a promising cryptographic solution to the cipher text retrieval issue in PKE systems. With a SPKE system, a user can authorize an un trusted third-party storage server to test whether the message cipher texts sent to him/her contain some specified keywords without divulging either the message contents or the search keywords. More specifically, a SPKE system works as follows: When producing the cipher text *Cm* of a message *m* by using a standard PKE scheme, the sender selects a keyword *w* which is related to the message *m* and executes the keyword encryption algorithm in SPKE to produce a keyword cipher text *Cw* for the keyword *w* by using the receiver's public key.

The message cipher text *Cm* appended with the keyword cipher text *Cw* is then sent to the storage server who stores the cipher texts for the receiver. When the receiver wants to download the message cipher texts related to a keyword *w* from the storage server, he/she runs a trapdoor generation algorithm to produce a keyword trapdoor *Tw* for *w* by using his/her private key. Where after, *Tw* is sent to the storage server secretly. Once receiving *Tw*, the storage server executes a testing algorithm to locate all keyword cipher texts that match *Tw* (namely that the keyword cipher texts contain the same keyword *w*).

Finally, the storage server returns all matching message cipher texts to the receiver. Since its invention, SPKE has been found to be useful in many practical applications, such as encrypted email routing [14], encrypted audit logs [15], cryptographic cloud storage [16], electronic medical/healthcare system [17, 18] and internet of things [19], etc.

II. RELATED WORK

In Eurocrypt'04, Boneh et al. [14] recommend the primary SPKE framework - public key encryptions with key-word seek (PEKS), along with a green PEKS scheme from the 9aaf3f374c58e8c9dcdd1ebf10256fa5 Boneh-Franklin identity-based encryption (IBE) scheme [20]. Following Boneh et al.'s pioneering paintings [14], some of PEKS schemes [21-23] and variants [22-24] had been offered inside the literature. However, the main shortcoming of PEKS is that it calls for conveying the keyword trapdoors through relaxed channel. If the keyword trapdoors are dispatched to the storage server publicly, any PEKS scheme is insecure underneath the adaptive selected-key-word attacks with the aid of out of doors attacker [23]. So one can restoration this trouble, Baek et al. [23] brought added the framework of comfy channel unfastened PEKS (SCFPEKS). In a SCF-PEKS machine, a garage server ought to be targeted as the tester and handiest the precise server has the capability to execute the checking out set of rules to check whether a keyword ciphertext and a keyword trapdoor correspond to the equal keyword by using the use of its personal key. In this manner, the requirement for conveying the keyword trapdoors via secure channel is eliminated. The SCF-PEKS framework is also referred to as certain server PEKS (dPEKS) [24]. Inspired by means of Baek et al.'s work [23], masses of SCF-PEKS/dPEKS schemes were provided, e.g. [16-20].

Maximum of the formerly proposed SPKE schemes were built over conventional p.c. however, it

is generally identified that conventional percent has the heavy certificate management trouble. In [21], Abdalla et al. introduced SPKE into the setting of IBC and proposed the framework of IBE with keyword seek (IBEKS). They additionally provided a customary IBEKS creation from two level hierarchical IBE. Later, numerous IBEKS and exact server IBEKS (dIBEKS) schemes [21-24] have been proposed. The IBEKS/dIBEKS schemes have the advantage of no certificate. However, they are inherently stricken with the important thing escrow and key distribution troubles. To remove the important thing escrow hassle, the concept of SPKE became similarly extended into the setting of CL-%. In [49], Peng et al. first offered a certificate less encryption with key-word seek (CLEKS) scheme. Quickly afterwards, some CLEKS and specific server CLEKS (dCLEKS) schemes have been provided [20-22]. The advantages of the CLEKS/dCLEKS schemes are certificate less and key escrow loose. However, they have the important thing distribution hassle which additionally leads to the secure channel requirement. This shortcoming limits the application of CLEKS/dCLEKS in public networks. Moreover, Li et al. [23, 24] provided new kind key-word seek schemes primarily based on attribute-based totally encryption.

In practice, users usually select key phrases from a small key-word space (which includes an English dictionary) to produce the keyword ciphertext/trapdoors. This is due to the fact that customers typically search their encrypted information by using a few keywords as a depend of comfort. But, the low-entropy feature of keywords consequences inside the key-word guessing (KG) assault on the SPKE schemes. By means of appearing the assault, an outside attacker or a malicious garage server is in a position to reveal the keyword in given keyword trapdoor/ciphertext in an acceptably quick time. Till now, three sorts of KG attacks have been presented. They're the out of doors offline KG attack, the outside online KG attack and the interior offline KG assault. In [15], Byun et al. first found the out of doors/interior offline KG attack and confirmed such attack on the PEKS schemes proposed with the aid of Boneh et al. [14] and Park et al. [21]. Later, Jeong et al. [16] indicated that it's far in reality impossible to build a PEKS scheme withstanding the interior offline KG assault, which means that the PEKS framework inherently suffers from such attack. In [17], Yau et al. confirmed that the dPEKS scheme proposed by way of Baek et al. [23] can't face up to the out of doors/inside offline KG assault.

In [23], Rhee et al. described the perception of keyword trapdoor indistinguishability and verified that a dPEKS scheme is comfy in opposition to outside offline KG attacks only whilst it achieves the trapdoor indistinguishability. But, Yau et al. [18] confirmed that the dPEKS framework suffers from the outdoor on line KG assault if the key-word trapdoors are sent through public channel. unlike the outside offline KG attack, the out of doors on-line KG attack depends at the detailed garage server's paintings, namely that the out of doors attacker takes the designated storage server as an oracle to get the trying out results in a web way. Yau et al.'s work [18] additionally means that the dPEKS framework fails in addressing the secure channel trouble in PEKS. Furthermore, Shao and Yang [19] indicated that the dPEKS framework is inherently vulnerable to the internal offline KG attack, because the precise garage server has both abilities to execute the key-word encryption set of rules and the testing set of rules. The KG attacks presented in [16, 18, 19] imply that different above-referred to SPKE frameworks (such as IBEKS, dIBEKS, CLEKS and dCLEKS) additionally cannot protect in opposition to KG assaults, considering that they are the herbal extensions of PEKS or dPEKS.

PRELIMINARIES

Bilinear Pairing

At some point of the paper, λ denotes a security parameter, q denotes a large top variety of λ bits and (G_1, G_2) respectively denote multiplicative cyclic businesses of same order q. permit g be the generator of the institution G1. A bilinear pairing is a map e: $G_1 \times G_1 \rightarrow G_2$ that has the subsequent 3 attributes:

1) Bilinearity: $\forall x, y \in Z^*_q$, $e(g^x, g^y) = e(g, g)^{xy}$.
2) Non-degeneracy: $e(g, g) \neq 1$.
3) Computability: $\forall x, y \in Z^*_q$, $e(g^x, g^y)$ can be calculated successfully.

Complexity Assumptions

The safety of the CBSE scheme proposed in this paper is verified underneath the complexity assumptions of the hash Diffie-Hellman (HDH) hassle [15] and the bilinear Diffie-Hellman (BDH) hassle [20].

Definition 1 (HDH problem) allow f: $G_1 \rightarrow \{0, 1\}^l$ be a cryptographic hash feature, in which $l \in Z^+$ denotes the bit-period of the hash value. Given the hash characteristic f and a tuple $(g, g^a, g^b, x) \in (G_l)^3 \times \{0, 1\}^l$ for unknown numbers a, b $\times Z^*_q$, the HDH

problem within the institution G_1 is to decide whether $x = f(g^{ab})$.

The HDH assumption states that the advantage $Adv^{HDH}(\lambda)$ in fixing the HDH hassle is negligible for any polynomial-time set of rules A, in which $Adv^{HADH}(\lambda) = |Pr[A(\lambda, q, G_1, f, g, g^a, g^b, f(g^{ab})) = 1] - Pr[A(\lambda, q, G_1, f, g, g^a, g^b, x) = 1]|$.

As mentioned by means of Abdalla et al. in [15], the HDH assumption is more potent than the computational Diffie-Hellman assumption however is weaker than the decisional Diffie-Hellman assumption.

Definition 2 (BDH hassle) given a tuple $(g, g^a, g^b, g^c) \in (G_1)$ four for unknown numbers a, b, c $\in Z^*_q$, the BDH trouble in (G_1, G_2) is to calculate $e(g, g)^{abc} \in G_2$.

The BDH assumption states that the benefit A $Adv^{BDH}(\lambda)$ in fixing the BDH hassle is negligible for any polynomial-time algorithm A, where $A\ Adv^{BDH}(\lambda) = Pr[A(\lambda, q, G_1, G_2, g, g^a, g^b, g^c) = e(g, g)^{abc}]$.

III. CBSE FRAMEWORK AND PROTECTION DEFINITIONS

Framework Definition

As illustrated in Fig. 1, our CBSE framework consists of four unique entities, particularly: a CA, a sender, a receiver

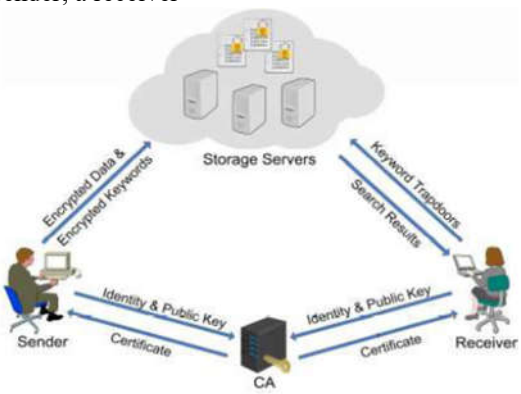

Fig. 1. The proposed CBSE framework

and a storage server. The CA controls the generation of the grasp secret key and the machine global parameters. It's also liable for issuing a certificate for every consumer in the gadget. The sender produces the statistics ciphertext and the corresponding keyword ciphertext, and outsources them to the storage server. The receiver can retrieve the statistics ciphertext dispatched to him/her at the garage server. To do so, the receiver must provide the storage server with a key-word trapdoor. Upon receiving a trapdoor from a receiver, the storage server searches the receiver's encrypted records through the usage of the key-word trapdoor and then returns all matching encrypted information.

Officially, our CBSE framework is defined as follows.

Definition 3 (CBSE) A CBSE scheme is certain by a tuple of four algorithms (GlobalSetup, KeyGen, CertGen, Encrypt) such that:

GlobalSetup(λ): Given a safety parameter λ, a CA runs this set of rules to create a list of world parameters gp and a grasp mystery key mk.

KeyGen(gp): Given gp, a consumer U runs this set of rules to create a couple of public and mystery keys (pk_U, sk_U). Specially, the sender S's and the receiver R's key-pairs are denoted by using (pk_S, sk_S) and (pk_R, sk_R) respectively.

CertGen(gp, mk, id_U, pk_U): Given gp, mk and a user U's identity id_U and public key pk_U, a CA runs this set of rules to trouble a certificate $cert_U$ for the user U. mainly, the sender S's and the receiver R's certificates are denoted by using $cert_S$ and $cert_R$ respectively.

Encrypt(gp, w, idS, skS, certS, idR, pkR): Given gp, a key-word w, the sender S's identity, mystery key and certificates (idS, skS, certS) and the receiver R's identification and public key (id_R, pk_R), the sender S runs this algorithm to produce a key-word ciphertext Cw.

SECURITY DEFINITIONS

as a way to withstand the KG attacks, a CBSE scheme have to simultaneously fulfill the keyword ciphertext indistinguishability underneath the adaptive selected-keyword assault (KC-IND-CKA), the key-word ciphertext unforgeability under the adaptive chosen-keyword assault (KCUNF-CKA) and the key-word trapdoor indistinguishability below the adaptive selected-key-word assault (KT-INDCKA). Introduce the following six oracles. those oracles are managed by means of a challenger and answered as follows:

$\Theta^{CreateUser}$**:** On input an identity id_U, a public key pk_U is spoke back if the identification id_U has already been created. Else, the identity id_U is created with the aid of generating a public/mystery key pair (pk_U, sk_U) and the general public key pk_U is again. We think that

the opposite 5 oracles simply respond to an identity id_U that has been created.

$\mathcal{O}^{SecretKey}$: On input an identity id_U, a secret key sk_U is answered.

$\mathcal{O}^{certificates}$: On input an identification id_U, a certificate $cert_U$ is answered.

$\mathcal{O}^{Encrypt}$: On input a keyword *w*, an identification id_S and an identification id_R, a keyword ciphertext C_w is answered.

$\mathcal{O}^{Trapdoor}$: On input a keyword *w*, an identity id_S and an identification id_R, a keyword trapdoor Tw is replied.

$\mathcal{O}^{take\ a\ look\ at}$: On input a keyword ciphertext C_w and a key-word trapdoorT_w', this oracle responds 1 meaning that C_w and T_w' correspond to the same keyword or zero else. This oracle models the adversary's potential to run the trying out set of rules or employ the storage server as a testing oracle.

IV. THE PROPOSED CBSE SCHEME

On this phase, we describe the proposed CBSE scheme and strictly prove its correctness and safety.

Description of the Scheme

Our CBSE scheme is composed of the following algorithms:

GlobalSetup(λ): This set of rules first produces multiplicative organizations (G_1, G_2) of λ-bit top order q and a bilinear pairing e: $G_1 \times G_1 \rightarrow G_2$. Assume that g is the generator of G_1. It then selects a random number $\alpha \in Z^*_q$ and calculates $g_1 = g^{\alpha}$. Let H_1: $\{0, 1\}^* \times G_1 \rightarrow G_1$, H_2: $G_2 \rightarrow \{0, 1\}^l$, H_3: $\{0, 1\}^* \times \{0, 1\}^l \times \{0, 1\}^l \times Z^*_q$, H_4:$G_1 \rightarrow \{0, 1\}^l$ and H_5: $G_2 \rightarrow \{0, 1\}^l$ be five cryptographic hash capabilities, where $l \in Z^+$. The algorithm ultimately units the CA's master mystery key mk = α and the worldwide parameters gp = {q, G_1, G_2, e, g, g_1, H_1, H_2, H_3, H_4, H_5}.

KeyGen(gp): This set of rules first selects a random wide variety $x_U \in Z^*_q$, then units x_U as a mystery key sk_U for the person U and computes the corresponding public key $pk_U = gx_U$.

CertGen(gp, mk, id_U, pk_U): This set of rules first computes $h_U = H_1(id_U, pk_U)$. It then computes $cert_U = (h_U)^{mk}$ as a certificates for the user U.

Encrypt*(gp, w, id_S, sk_S, $cert_S$, id_R, pk_R)*: Given gp, a key-word w, the sender S's identification, mystery key and certificate (id_S, sk_S, $cert_S$) and the receiver R's identity and public key (id_R, pk_R), this set of rules first computes $k_1 = H_4((pk_R)^{skS})$ and $k_2 = H_5(e(cert_S , h_R))$, in which $h_R = H_1(id_R, pk_R)$. It then selects a random range $r \in Z^*_q$, computes $c_1 = g^r$ and $c_2 = H_2(e(pk_R\ g_1\ , h_R)^{rw})$, wherein $w = H_3(w, K_1, K_2)$. Ultimately, it units the keyword ciphertext $C_w = (c_1, c_2)$.

Correctness of the Scheme

Theorem 1. Our CBSE scheme is accurate.

Evidence. In line with the above description of the scheme, we've

$K_1 = H_4((pk_R)^{skS}) = H_4((gsk_R)^{skS}) = H_4((pk_S)^{skR})$,

$K_2 = H_5(e(cert_S , h_R)) = H_5(e(h_S , h_R)) = H_5(e(h_S , cert_R))$,

$c'_2 = H_2(e(c_1, T_{w'}))$

$= H_2(e(g^r, ((h_R)^{skR} \bullet cert_R)^{w'}))$

$= H_2(e(g^r, ((h_R)^{skR})^{w'}) \bullet e(g^r, (cert_R)^{w'}))$

$= H_2(e(gsk_R, h_R)^{r \bullet w'} \bullet e(g^{\alpha} , h_R)^{r \bullet w'})$

$= H_2(e(pk_R\ g_1 , h_R)^{r \bullet H3\ (w', \kappa1, \kappa2)})$.

Glaringly, if w = w′, then $H_3(w, K_1, K_2) = H_3(w', K_1, K_2)$. Consequently, $c'_2 = c_2$. Consequently, our CBSE scheme is correct.

Safety Proofs

Next, we prove our CBSE scheme to meet the KC-INDCKA safety, the KC-UNF-KGA protection and the KTIND-CKA protection.

Theorem 2. Our CBSE scheme achieves the KC-IND-CKA safety beneath the BDH assumption inside the random oracle model.

This theorem may be validated by the following Lemma 1 and Lemma 2.

Lemma 1. If H1 ~ H5 are five random oracles and there is an adversary A_1 towards our CBSE scheme who can win the adversarial game KC-IND-CKA-1 with gain ξ, then there is always a polynomial-time set of rules A_{BDH} to clear up the BDH trouble inside the businesses (G_1, G_2) with gain $\xi' \geq \xi\ (1/q_2q_x) \bullet (1 - 1/q_x)^{qt}$, in which q_2, q_u and q_t respectively denote the maximal number of the adversary A_1's queries to H_2, $\mathcal{O}^{CreateUser}$ and $\mathcal{O}^{Trapdoor}$.

V. COMPARISONS

We first evaluate our CBSE framework with the previous SPKE frameworks (together with PEKS [14], dPEKS [23], IBEKS [21], dIBEKS [24], CLEKS [10], dCLEKS, au-PEKS [19], im-dPEKS [19], SEK-IA [20] and PEKS+SSE [21]). The properties of all as compared frameworks are indexed in desk 3, wherein KGA1, KGA2 and KGA3 respectively denote whether the framework is cozy beneath the out of doors offline KG assault, the outside online KG assault and the internal offline KG attack. The evaluation suggests that our CBSE framework enjoys higher residences at the same time as imparting resistance to the existing recognized three varieties of KG assaults.

Next, we check the efficiency of our CBSE scheme. Thinking about that CBC and CL-p.c percentage numerous similar capabilities [4], we make a evaluation of our scheme and a few recently proposed CLEKS/dCLEKS schemes [20-22] in phrases of computation and conversation charges. desk four shows the overall performance of the as compared schemes, in which t_p, te_1, te_2, t_{mh} are the time fees for computing a bilinear pairing, an exponentiation in G_1, an exponentiation in G_2 and a map-to-point hash, respectively, $|G_1|$ and l are the bit duration of an element in G_1 and a hash fee, respectively. The time cost of a set of rules is evaluated with the aid of the sum of the time expenses of all involved time-consuming operations. As ordinary, the fees of the general cryptographic hash operations are neglected.

TABLE 2
COMPARISON OF THE CBSE FRAMEWORK AND THE PREVIOUS SPKE FRAMEWORKS

Frameworks	Implicit authentication	No key escrow	No key distribution	No secure channel	Non interactive	KGA1	KGA2	KGA3
PEKS [14]	no	yes	yes	no	yes	secure	secure	insecure
dPEKS [33]	no	yes	yes	yes	yes	secure'	insecure	insecure
IBEKS [41]	yes	no	no	no	yes	secure	secure	insecure
dIBEKS [44]	yes	no	no	no	yes	secure'	insecure	insecure
CLEKS [50]	yes	yes	no	no	yes	secure	secure	insecure
dCLEKS [52]	yes	yes	no	no	yes	secure'	insecure	insecure
im-dPEKS [59]	no	yes	yes	yes	yes	secure'	secure	insecure
SEK-IA [60]	no	no	no	no	yes	secure	secure	secure
PEKS+SSE [61]	no	yes	no	no	no	secure	secure	secure
au-PEKS [19]	no	yes	yes	yes	no	secure	secure	secure
CBSE	yes	yes	yes	yes	yes	secure	secure	secure

TABLE 3
EFFICIENCY OF THE COMPARED SCHEMES

Schemes	Computation cost			Communication cost	
	Encrypt	Trapdoor	Test	Ciphertext size	Trapdoor size
CLEKS [50]	$5t_{e1}+t_{mh}$	$8t_{e1}+t_{mh}$	$4t_p$	$4\|G_1\|$	$4\|G_1\|$
CLEKS [51]	$t_p+5t_{e1}+t_{mh}$	$t_{e1}+t_{mh}$	$2t_p+3t_{e1}$	$3\|G_1\|$	$\|G_1\|$
dCLEKS [52]	$3t_p+4t_{e1}+3t_{mh}$	$t_{e1}+t_{mh}$	$t_p+t_{e1}+2t_{mh}$	$\|G_1\|+l$	$\|G_1\|$
Our CBSE	$2t_p+2t_{e1}+t_{e2}+t_{mh}$	$t_p+3t_{e1}+2t_{mh}$	t_p	$\|G_1\|+l$	$\|G_1\|$

as an example, to encrypt a key-word w, the keyword encryption algorithm Encrypt in our CBSE scheme want to calculate bilinear pairings, two exponentiation in G1, one exponentiation in G2 and one map-to-factor hash. Hence, the time price of the key-word encryption set of rules is $2t_p + 2te_1 + te_2 + t_h$. Further, the communication cost of a key-word ciphertext/trapdoor is measured with the aid of the total variety of the worried organization factors and hash values. As an instance, a keyword ciphertext in our CBSE scheme carries one group detail in G_1 and one hash cost. Accordingly, the bit-period of a key-word ciphertext is $(|G1| + l)$ bits.

As shown via table three, our scheme is less efficient inside the key-word trapdoor generation set of rules in comparison with the schemes. But, the experimental outcomes (see Appendix A) display that it outperforms the schemes in both the keyword encryption algorithm and the testing set of rules. Honestly, to fight against KG attacks, our scheme has to embed two shared secrets in each the keyword ciphertext and the trapdoor, which introduces one pairing and one exponentiation in G1 in the calculation of key-word ciphertext/trapdoor. This greater computation price is profitable since it makes our scheme withstand both out of doors and internal KG attacks. For the verbal exchange fee, our scheme is as green because the dCLEKS scheme and is higher than the CLEKS schemes.

VI. CONCLUSION

In this paper, we propose the CBSE framework to resolve the security problems in the previous SPKE frameworks. The presented framework provides resistance against both the outside and inside KG attacks and has the merits of implicit authentication, no key escrow, no key distribution and no secure channel. Under this framework, we construct a concrete CBSE scheme and prove it to satisfy the keyword cipher text indistinguishability, the keyword cipher text unforgeability and the keyword trapdoor indistinguishability against KG attacks under the

BDH and HDH assumptions in the random oracle model. Comparisons indicate that our CBSE scheme is secure and practicable. The limitation of the CBSE framework is that the receiver should involve the sender's public key in the generation of keyword trapdoor. This implies that the receiver must designate the sender when he/she makes search queries on his/her cipher texts. It may be less efficient when the receiver wants to search the cipher texts from many different senders. So, it would be more interesting to devise a CBSE framework that is secure against the existing known types of KG attacks while providing full search function (namely that a user is able to search all his/her cipher texts with a single keyword trapdoor, regardless of who sends him/her the cipher texts). This seems to be a more challenging work. So, we leave it as our future work and also pose it as an open problem.

VII. REFERENCE

[1] Abdalla, M. et al.: 'Searchable encryption revisited: consistency properties, relation to anonymous ibe, and extensions'. J. Cryptol., 21, 2008. pp. 350-391.

[2] Baek, J., Safavi-Naini, R., Susilo, W.: 'Public Key Encryption with Keyword Search Revisited'. ICCSA'08. 5072(2008). pp.1249-1259.

[3] K. Emura, A. Miyaji, M. S. Rahman, and K.Omote, "Generic constructions of securechannel free searchable encryption with adaptive security," Secur. Commun. Netw., vol. 8, no. 8, pp. 1547–1560, 2015.

[4] C. Cocks, "An identity based encryption scheme based on quadratic residues," in Cryptography and Coding. Cirencester, U.K.: Springer, 2001, pp. 360–363.

[5] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in EUROCRYPT, 2004, pp. 506–522.

[6] S Bhagyashri, YB Gurave, A survey on privacy preserving techniques for secure cloud storage. International Journal of Computer Science and Mobile Computing (IJCSMC) 3(2), 675–680 (2014).

[7] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, "Building an encrypted and searchable audit log," in NDSS, 2004.

[8] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. MaloneLee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions," in CRYPTO, 2005, pp. 205–222.

[9] D. Khader, "Public key encryption with keyword search based on k-resilient IBE," in Computational Science and Its Applications -ICCSA, 2006, pp. 298–308.

[10] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," IEEE Trans. Computers, vol. 62, no. 11, pp. 2266– 2277, 2013.

[11] K. Ashton, "That 'internet of things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[12] D. L. Brock, "The electronic product code (epc)," *Auto-ID Center White Paper MIT-AUTOID-WH-002*, 2001.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, ACM, pp. 13–16, Finland, August 2012.

[14] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[15] Y. Ding, Y. Jin, L. Ren, and K. Hao, "An intelligent self-organization scheme for the internet of things," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 41–53, 2013.

[16] Y. Yuehong, Y. Zeng, X. Chen, and Y. Fan, "The internet of things in healthcare: an overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3–13, 2016.

[17] C. Koop, R. Mosher, L. Kun et al., "Future delivery of health care: cybercare," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 6, pp. 29–38, 2008.

[18] C. Koop, R. Mosher, L. Kun et al., "Future delivery of health care: cybercare," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 6, pp. 29–38, 2008.

[19] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: a location difference-based proximity detection protocol for fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1117–1124, 2017.

[20] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14)*, pp. 1–8, IEEE, Warsaw, Poland, September 2014.

[21] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: real-time data aggregation with adaptive ω-event differential privacy for fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6285719, 13 pages, 2018.

[22] J. Mao, W. Tian, J. Jiang, Z. He, Z. Zhou, and J. Liu, "Understanding structure-based social network de-anonymization techniques via empirical analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, p. 279, December 2018.

[23] Y. Huo, Y. Tian, L. Ma, X. Cheng, and T. Jing, "Jamming strategies for physical layer security," *IEEE Wireless Communications Magazine*, vol. 25, no. 1, pp. 148–153, 2018.

[24] M. Sookhak, F. R. Yu, M. K. Khan, Y. Xiang, and R. Buyya, "Attribute-based data access control in mobile cloud computing: taxonomy and open issues," *Future Generation Computer Systems*, vol. 72, pp. 273–287, 2017.

Authors Profile

Mrs. Nithya Ramakrishna working as Assistant Professor of CSE Department in QIS College of Engineering and Technology (Autonomous), Ongole, Andhra Pradesh, India.

Mr. **S. V. Sandeep** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **T. Akhil Krishna** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **B. Koteswara Rao** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **N. Raja Pavan Kumar Reddy** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **M. Kiran Kumar** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.