Cross-domain heterogeneous signcryption with keyword search for wireless body area network

Ming Luo (lmhappy21@163.com)
Nanchang University

Dashi Huang
Nanchang University

Minrong Qiu
GongQing Institute of Science and Technology

Research Article

Keywords: Searchable encryption, WBAN, Inside keyword guess attack, Heterogeneous signcryption

Posted Date: June 8th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1647848/v2

Cross-domain heterogeneous signcryption with keyword search for wireless body area network

Ming Luo[1] · Dashi Huang[1] · Minrong Qiu[2]

Abstract
In wireless body area network (WBAN), the WBAN sensors usually transmit encrypted data for the sake of security. But because of this, finding and getting the required data quickly becomes a challenge. Thus, many scholars have proposed searchable encryption (SE) schemes. Regrettably, many proposed SE schemes are not resistant to inside keyword guessing attack (IKGA), and most schemes are not suited for cross-domain communication in WBAN because they use single cryptosystem and the same cryptographic system parameters. To address the aforementioned issues, we put forward a new SE scheme for WBAN based on the public key cryptosystem in this paper. Our scheme allows WBAN sensors in certificateless public key cryptography (CLC) and receivers in public key infrastructure (PKI) environment use different cryptographic system parameters to realize cross-domain heterogeneous communication. At the same time, our scheme can effectively resist IKGA. Compared to the five existing schemes, the total computation cost of our scheme reduced by at least 59.99%.

Keywords Searchable encryption · WBAN · Inside keyword guess attack· Heterogeneous signcryption

1 Introduction

The development of cloud storage facilitates access to data and makes it increasingly important in the application of WBAN [1-2]. For instance, the sensors in WBAN transmit the collected physiological data via the Internet to a third-party cloud server for storage and allow users to quickly find the data they need [3-4]. Despite the convenience of cloud storage, data stored on cloud servers also face additional security challenges [5].

To preserve data confidentiality, the sensors in WBAN usually upload encrypted data to the cloud server, but there is a drawback to it is that the standard search tools become useless since encryption ruins the original structure of the data. Fortunately, SE simplified this problem [6-9]. Currently symmetric and asymmetric searchable encryption are the two major classifications of SE. Song et al. [6] firstly presented searchable symmetric encryption (SSE), after that, many scholars put forward relevant new schemes [10-14]. These schemes have good efficiency due to the characteristics of symmetric cryptosystem, but they also confront the difficulty of how to distribute keys safely. Then, Boneh et al. [15] homeopathically presented the concept of public key encryption with keyword search (PEKS) and gave a PEKS scheme that can resist chosen keyword attack (CKA). After that, some PEKS-based schemes are put forward [16-17]. However, for some PEKS schemes, when a trapdoor is given, the adversary can use exhaustive enumeration to gather keyword information [18], which is known as keyword guessing attack (KGA). IKGA is a more hazardous KGA that is launched by an inside adversary, so a secure PEKS scheme needs to be able to strongly resist CKA and IKGA [18]. In addition, a large number of WBAN sensors and receivers are typically in separate domains and use different cryptographic system parameters, it naturally makes sense to design a secure searchable encryption scheme with different cryptographic system parameters and satisfying heterogeneity.

✉ Ming Luo
E-mail: lmhappy21@163.com, Phn: +86-791-88305116, Fax: +86-791-88337187

Dashi Huang
412849610@qq.com

Minrong Qiu
13576203266@qq.com

[1] School of Software, Nanchang University, No.235 Nanjing East Road, Nanchang, Jiangxi, 30047, China

[2] GongQing Institute of Science and Technology, No.1 Gongqing Avenue, Gongqing, JiangXi, 332020, China

1.1 Related work

For the schemes based on the notion of PEKS, the space of password is usually much bigger than that of keyword, and receivers frequently use certain keywords for data search. Based on this fact, Byun et al. [18] claimed that scheme [15] is insecure under KGA in 2006. To resist KGA, many scholars have made efforts. Secure channel-free public key encryption with keyword search (SCF-PEKS), also known as PEKS with a designated server/tester (dPEKS), was introduced by Baek et al. [19]. Baek et al. add the tester's public key to the keyword ciphertext generation in [19] ensuring that the test operation can only be performed by the server with the associated private key. By limiting the one who can do the test operation, Ma et al [20] proposed a PEKS scheme, which is able to resist KGA. Wang et al. [21] proposed a SE scheme with two servers working together, the servers perform the ciphertext retrieval operation by sharing the secret retrieval trapdoor. Unfortunately, the KGA initiated by the inside adversary, namely IKGA, makes the schemes [18-21] that can only resist outside KGA no longer safe. To further address this problem. In 2017, Huang et al. [22] firstly proposed a concept called "public-key authenticated encryption with keyword search (PAEKS)", which requires the sender to add its own private key when generating keyword ciphertext, so that the attacker cannot generate effective keyword ciphertext at will, so as to resist IKGA. In 2021, Liu et al. [23] proposed a concept called "designated ciphertext searchable encryption (DCSE)". Basically, DCSE means that the receiver needs to rely on the tag related to the keyword ciphertext generated by the sender to generate a corresponding trapdoor, and the information of this tag can only be obtained by the receiver, so the attacker cannot generate the ciphertext matching the trapdoor for IKGA. Compared with PAEKS, although DCSE can also resist IKGA, the addition of tag increases the communication cost of the scheme.

For many years, the major cryptographic systems generally used by scholars were identity-based cryptography (IBC) with key escrow issues and PKI with the concerns of certificate management. Surprisingly, CLC [24] proposed by Al-riyami and Paterson solves these two problems. In recent years, quite a few searchable encryption schemes based on CLC are presented [25-30]. The algorithms adopted by Zhang et al. [25] and Yang et al. [28] are relatively complex, resulting in high computational cost. He et al. [29] pointed out that two proposed schemes [27] is insecure because they are vulnerable to IKGA. And He et al. [29] provided a SE scheme proven to be safe under IKGA. In 2020, Ma et al. [30] presented a new SE scheme based on CLC without pairing, but the scheme fails to resist IKGA initiated by collusive internal attackers. It should be noted that the above schemes have a common regret that they do not satisfy the heterogeneity. To improve it, in 2018, a heterogeneous keyword search scheme (HSC-KW) for WBAN [31] that assures the data being transferred is not only secure but also authenticated was presented by Omala et al. Unfortunately, in [31], the same system parameters are used by senders and receivers in separate network domains.

1.2 Our contributions

Based on the notion of PEKS, we propose a new searchable encryption scheme in this paper named cross-domain heterogeneous signcryption with keyword search (CHSKS), which entitles senders working within the CLC system and receivers in the PKI environment to communicate with each other. Our CLC-PKI CHSKS is symbolized by the symbol "CP-CHSKS", which makes the following innovations:

1) We present a new cross-domain heterogeneous signcryption with keyword search scheme. In our scheme, the sensors in WBAN and receivers are allowed not to be in the same domain, and different cryptographic system parameters are used in each domain.
2) The proposed scheme realizes ciphertext indiscernibility, ciphertext unforgeability and trapdoor indiscernibility in the random oracle model (ROM). In the face of CKA, the CP-CHSKS scheme is secure. In addition, our scheme adds the features of authentication, which makes it hard for adversaries to forge keyword ciphertexts at will, that means our scheme is also resistant to IKGA.
3) Our scheme has outstanding performance. For some previous schemes, they need more computation cost than our proposed scheme. Compared with [25], [28], [29], [30], and [31], the total computation cost of our scheme decreased by about 87.61%, 69.93%, 71.83%, 61.17% and 59.99%, respectively.

1.3 Organization

The following sections make up the remainder of this paper: section 2 contains the descriptions of the system model of our scheme and the mathematical assumptions necessary to prove the security of the CP-CHSKS. Section 3 introduces the generic model of our scheme and its security model. The detailed descriptions of the proposed scheme and its security analysis are included in sections 4 and 5 respectively. The sixth section analyzes the performance of our scheme, while the last section summarizes this study.

2 Preliminaries

2.1 System model

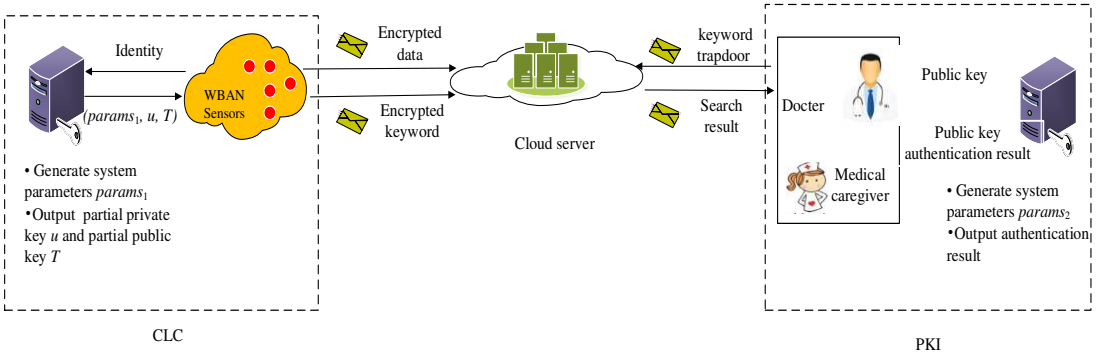

Fig.1 System model

As shown in Fig.1, three entities work in the system model of our scheme, including WBAN sensors under the circumstance of CLC, the medical service provider like doctor or medical caregiver under the circumstance of PKI, and the cloud server. WBAN sensors use the system parameters and partial private key generated by the key generation center (KGC) which in this case is the CLC server. The PKI server is equivalent to a certificate authority (CA), which undertakes the tasks of producing the system parameters of PKI as well as the certified public key of medical service provider. Note that the system parameters of CLC and PKI are not the same.

The main relationship of the three entities is as follows: WBAN sensors collect physiological data and extract a keyword from the data, then encrypt these data and upload the encrypted data to cloud server. For the sake of desired data, the medical service provider generates a keyword trapdoor and sends it to the cloud server, the server checks if the trapdoor supplied by the medical service provider matches the stored encrypted data, if so, returns the matching data.

2.2 Computational assumptions

Definition 1. $\hat{e}: G_1 \times G_1 \to G_2$ is a bilinear pairing, G_1 and G_2 have the same order q, G_1 and G_2 are the cyclic additive group and the cyclic multiplication group respectively. Bilinear pairing has the following properties:

1) Bilinear: For any $P, Q \in G_1$ and $x, y \in Z_q^*$, it must exist that $\hat{e}(xP, yQ) = \hat{e}(P, Q)^{xy}$.

2) Non-degenerate: $\exists P, Q \in G_1$, it makes $\hat{e}(P, Q) \neq 1_{G_2}$.

3) Computable: There is a valid algorithm to calculate $\hat{e}(P, Q)$.

Definition 2. Discrete Logarithm Problem (DLP): there is a tuple (P, aP), where $a \in Z_q^*$ is sealed. The purpose is to figure out a.

Definition 3. Bilinear Diffie-Hellman Inversion Problem (BDHIP): there is a tuple (P, aP), where $a \in Z_q^*$ is sealed. The purpose is to figure out $\hat{e}(P, P)^{1/a}$.

Definition 4. Computational Diffie-Hellman problem (CDHP): There is a tuple (P, aP, bP), where $a, b \in Z_q^*$ is sealed. The purpose is to figure out the value of abP.

3 Generic construction

3.1 Generic model

The following eight algorithms are available in the generic CP-CHSKS:

1) Setup: As long as a security parameter s is provided, KGC utilize it to run this algorithm to get the necessary parameters, which including the master secret key α and public system parameters $PParams_1$. CA can similarly generate PKI system parameters $PParams_2$.
2) CLC-Partial key extraction (CL-PKE): when an identity ID_i and a master secret key α are input, KGC runs this algorithm to produce a partial private key u_i and a partial public key T_i.
3) CLC-Secret value generation (CL-SVG): To get a secret value d_i, when an identity ID_i is input, the data sender in the context of CLC needs to run this algorithm. Note that a secret value d_i and a partial private key u_i can compose a user's full private key $SK_i = (u_i, d_i)$.
4) CLC-Public key generation (CL-PKG): The data sender in the context of CLC computes public key PPK_i after getting a secret key d_i. Then, the whole public key $PK_i = (T_i, PPK_i)$ is output.
5) PKI-Key generation (PKI-KG): Enter a receiver's private key d_j selected randomly by receiver, to get a corresponding public key PK_j, the receiver in PKI environment runs this algorithm.
6) CLC-PKI PEKS(CP-PEKS): A keyword $w \in W$ (all the keywords are in W) extracted from data m, the public key of receiver PK_r and the full sender's private key are the inputs of this algorithm. To generate the keyword ciphertext σ_w, the date sender needs to run this algorithm.
7) PKI-Trapdoor generation(PKI-TG): The receiver in PKI executes this algorithm to generate a keyword trapdoor T_w by taking a keyword w, the system parameters $PParams_1$ of sender's environment and the private key d_r of receiver as inputs.
8) Test: Cloud server takes system parameters, a trapdoor T_w and a keywork ciphertext σ_w as inputs, returns true if the verification is successful. Otherwise, $\perp$ is returned.

3.2 Security model

A CP-CHSKS should not only satisfy ciphertext indistinguishability and trapdoor indistinguishability, but also unforgeability is needed. Two adversaries A_1 and A_2 exist in CLC according to [24]. A_1 is unable to get the master secret key, but A_2 is able to do so. A_2 is unable to replace the sender's public key, whereas A_1 is capable of doing so. To facilitate the distinction, we add an adversary A_3, where A_3 is the adversary who has the same ability as A_1 and tries to break the indistinguishability of trapdoor. The security model of CP-CHSKS is illustrated by the following three games, each of these three games is completed by a challenger C and an adversary A (A could be one of A_1, A_2 and A_3). The oracles listed below may be used:

- *Hash-query*: A executes this query according to the required parameters of hash function $H_{i(i=1,2,3)}$, then C computes and returns the hash value.
- *CL-partial key query*: A executes this query with the purpose of obtaining a user's partial private key u_i. Given ID_i, C calculates and returns u_i.
- *CL-secret value query*: A queries C with an identity ID_i, then C performs CL-SVG algorithm to obtain d_i and returns it to A.

- *CL-public key query*: A provides C with ID_i. To get and return the related public key PK_i, challenger C needs to executes CL-PKG algorithm.
- *CL-replace public key query*: Any sender's public key in CLC environment can be replaced with a valuable value by A (A could not be A_2).
- *CL-PKI-SE query*: A sender's identity ID_S, a receiver's identity ID_r and a keyword w are given to C, then C runs CP-PEKS algorithm to generate ciphertext σ and return it to A.
- *PKI-public key query*: A provides C with ID_j, then C executes PKI-KG algorithm and returns PK_j to A.
- *PKI-trapdoor query*: When C receives a keyword w and a receiver's identity ID_r sent by A, C performs PKI-TG algorithm to generate corresponding trapdoor T_w and return it to A.

Definition 5. If any polynomially bounded adversary $A_{I(I=1,2)}$ is not able to win Game 1 with a non-negligible advantage, then the proposed CP-CHSKS possesses ciphertext indistinguishability and is able to resist A_I's CKA.

Game 1

Initialization. The security parameter s is given, C generates cryptographic system parameters and master secret key α by performing Setup algorithm. C provides A_I with system parameters, then sends A_2 the master secret key α and keeps the value confidential to the adversary A_1.

***Phase* 1** A_I can initiate a series of queries to C during this phase, these queries are consistent with the queries defined in the security model. Additionally, A_2 does not need to perform *CL-partial query* and *CL-replace public key query*.

Challenge A_I provides C with a receiver's identity ID_B, a sender's identity ID_A and a pair of keywords (w_0, w_1), the restriction is that the *PKI-trapdoor query* on keywords (w_0, w_1) has never been asked before. Then C chooses a bit λ from $\{0,1\}$ randomly and computes a keyword ciphertext $\sigma_\lambda^* = \text{CP-PEKS}(w_\lambda, SK_A, PK_A, PK_B)$. Finally, the σ_λ^* is returned.

***Phase* 2** C is queried continuously by adversary A_I, but A_I has no chance to perform a *PKI-trapdoor query* on keyword $w_{\lambda(\lambda=0,1)}$ at this phase.

Guess A_I is the winner of this game only if A_I outputs a bit λ' that is equal to λ.

Definition 6. If any polynomially bounded adversary A_3 is not able to win Game 2 with a non-negligible advantage, then the proposed CP-CHSKS possesses trapdoor indistinguishability and is able to resist CKA initiated by A_3.

Game 2

Initialization This game's initialization needs the same procedures as the initialization of Game 1.

***Phase* 1** Adversary A_3 can query challenger C the queries contained in phase 1 of Game 1.

Challenge A_3 sends C a receiver's identity ID_B and a pair of chosen keywords (w_0, w_1), the restriction is that the *PKI-trapdoor query* and *CL-PKI-SE query* on keywords (w_0, w_1) have never been asked before. Then C determines a random selection λ form $\{0,1\}$ and computes a trapdoor $T_\lambda^* = \text{PKI-TG}(w_\lambda, d_B, PParams_I)$. Finally, the T_λ^* is returned.

***Phase* 2** A_3 is able to perform various queries continuously except for the *CL-PKI-SE query* and *PKI-trapdoor query* on keyword $w_{\lambda(\lambda=0,1)}$.

Guess A_3 is the winner of this game only if A_3 outputs a bit λ' that is equal to λ.

Definition 7. If any polynomially bounded adversary $A_{l(l=1,2)}$ is not able to win Game 3 with a non-negligible advantage, then the proposed CP-CHSKS possesses unforgeability and is able to resist A_l's IKGA.

Game 3

Initialization This game's initialization follows the same procedures as the initialization of Game 1.

***Phase* 1** Adversary A_l is allowed to perform a series of queries contained in phase 1 of Game 1.

Forgery A_l picks a keyword w, a sender's identity ID_A and an identity ID_B of receiver, then outputs σ_w^* as forged keyword ciphertext. What is needed for A_l to win the game is the satisfaction of the following conditions:

1) The match of σ_w^* and T_w is successful when the Test algorithm is executed.

2) A_1 cannot perform *CL-replace public key query* and *CL-partial key query* on ID_A simultaneously.

3) σ_w^* is not be generated by the algorithm CP-PEKS.

4 The proposed scheme

Now, we describe our CP-CHSKS in detail.

Setup: After selecting a security parameter s, KGC chooses a cyclic addition group G_1 and a cyclic multiplication group G_2 with the same order of prime q_1, selects P_1 as G_1's generator and confirms a bilinear pairing $\hat{e}: G_1 \times G_1 \to G_2$. The KGC selects a value $\alpha \in Z_{q_1}^*$ as its master secrete key and uses α to compute $P_{pub} = \alpha P_1$, then KGC needs to confirm three hash functions $H_1: \{0,1\}^* \times G_1 \to Z_{q_1}^*$, $H_2: \{0,1\}^* \to Z_{q_1}^*$, $H_3: \{0,1\}^{2*} \times Z_{q_1}^* \times Z_{q_1}^* \times G_1 \to Z_{q_1}^*$. After theses operations are completed, the system parameters $PParams_1 = \{G_1, P_1, q_1, P_{pub}, H_1, H_2, H_3\}$ of CLC are determined. Similarly, CA generates the system parameters $PParams_2 = \{G_1', P_2, q\}$ of PKI, G_1 is a subgroup of G_1' and the order of G_1' is prime q, P_2 is a generator of group G_1'.

CL-PKE: KGC firstly enters an identity of sender $ID_i \in \{0,1\}^*$ and a random number $r_i \in Z_{q_1}^*$ chosen by itself, then computes $D_i = r_i P_1$, $t_i = H_1(ID_i, D_i)$, and finally outputs the partial private key $u_i = r_i + \alpha(t_i + 1) (\mathrm{mod}\, q_1)$ and the part of public key $T_i = D_i + t_i P_{pub}$.

CL-SVG: The secret value $d_i \in Z_{q_1}^*$ is a random selection of sender ID_i. Note that the user's full private key can be interpreted as $SK_i = (u_i, d_i)$ now.

CL-PKG: Another part of public key $PPK_i = d_i P_1$ of sender ID_i is computed by itself, then $PK_i = (T_i, PPK_i)$ is set as the full public key of sender.

PKI-KG: Private key $d_j \in Z_q^*$ is randomly selected by the receiver in PKI, and $PK_j = d_j P_2$ is set as the receiver's public key.

CP-PEKS: A keyword w, a sender's private SK_S and the public key of receiver PK_r are the most necessary inputs of this algorithm. Sender carries out this algorithm as follows:

1) Chooses a random number $k \in Z_{q_1}^*$.

2) Computes $h_w = H_2(w)$.

3) Computes $R = h_w^2 k d_s PK_r$.

4) Computes $h = H_3(ID_S, PPK_S, T_S, R)$

5) Computes $y = h^{-1} d_s (h_w k + 1) \Big/ u_s \bmod q_1$, then sender outputs the ciphertext $\sigma = (R, y)$.

PKI-TG: Receiver takes a keyword w, the system parameter $PParams_1$ and the private key d_r of receiver as inputs, then performs the following steps to generate a trapdoor:

1) Computes $h_w = H_2(w)$

2) Computes $T_w = (h_w d_r)^{-1} P_1$, then, receiver outputs the trapdoor T_w.

Test: The cloud server that received trapdoors performs this algorithm to detect whether equation $\hat{e}(PPK_S, P_2) = \hat{e}(T_S + P_{pub}, P_2)^{hy} / \hat{e}(R, T_w)$ holds, where $h = H_3(ID_S, PPK_S, T_S, R)$. If the verification is successful, the test server returns the corresponding data, otherwise, $\perp$ is returned.

Now, we verify the correctness of the scheme.

$$\hat{e}(PPK_S, P_2)$$
$$= \hat{e}(d_S P_1, P_2)$$
$$= \hat{e}(P_1, P_2)^{d_S}$$
$$\hat{e}(T_S + P_{pub}, P_2)^{hy} / \hat{e}(R, T_w)$$
$$= \hat{e}(D_S + (t_s + 1)P_{pub}, P_2)^{hy} / \hat{e}(h_w^2 k d_s d_r P_2, (h_w d_r)^{-1} P_1)$$
$$= \hat{e}(P_1(r_s + (t_s + 1)\alpha), P_2)^{hy} / \hat{e}(h_w k d_s P_2, P_1)$$
$$= \hat{e}(P_1, P_2)^{d_s(h_w k + 1)} / \hat{e}(P_2, P_1)^{d_s h_w k}$$
$$= \hat{e}(P_1, P_2)^{d_S}$$

5 Security analysis

Theorem 1. Under the hypothesis of the complexity of CDHP, the proposed CP-CHSKS achieves ciphertext indistinguishability to resist CKA comes from adversary $A_{I(I=1,2)}$ in the ROM.

Proof: Challenger C and adversary A_I play Game 1 together. C knows the tuple (P, aP, bP) of CDHP but does not know the value of a and b. The purpose of C is to compute abP.

Initialization C executes Setup algorithm using the given security parameter s to produce system parameters and master key α, then sends system parameters to A_I. Especially, C sends A_2 the master secret key α and keeps the value confidential to adversary A_1.

***Phase* 1** For the smooth progress of the game, C maintains five lists, $L_{i(i=1,2,3)}$, LK_c and LK_p. The outputs of hash queries are recorded by three lists $L_{i(i=1,2,3)}$, and the results of public key queries in the CLC and PKI environment are recorded by LK_c and LK_p respectively. C sets $P_{pub} = \alpha P$ and chooses two challenged identity $ID_{x(1 \le x \le q_H)}$ and $ID_{y(1 \le y \le q_P)}$ (Suppose that adversary can make q_H times *CL-public key query* and q_P times *PKI-public key query* at most) at random, and then adaptively handles various queries submitted by A_I:

- H_1 *query*: A_I submits this query on ID_i, if tuple (ID_i, D_i, t_i) is existed in L_1, then t_i is returned to A_I by C. Otherwise, C selects $t_i \in Z_{q_1}^*$ randomly as the return and inserts (ID_i, D_i, t_i) into list L_1.
- H_2 *query*: A_I makes this query on a keyword w, C checks whether there is a tuple (w, h_w) in the list L_2, if it exits, returns h_w. Otherwise, C randomly selects h_w from $Z_{q_1}^*$ as the return and inserts (w, h_w) into L_2.
- H_3 *query*: A_I submits H_3 query on tuple (ID_i, PPK_i, T_i, R_w). C inspects if the tuple $(ID_i, PPK_i, T_i, R_w, h_i)$ exists in the list L_3, if there is corresponding tuple, C returns h_i.

Otherwise, C randomly chooses a value h_i from $Z_{q_1}^*$ as the return and inserts $(ID_i, PPK_i, T_i, R_w, h_i)$ into L_3.

- *CL-secret value query*: C needs to determine whether ID_x and ID_i are equal when receives a *CL-secret value query* on ID_i. If $ID_x = ID_i$, C aborts this game, if this is not the case, C checks if the relevant entry $(ID_i, d_i, T_i, u_i, PPK_i, r_i)$ exists in LK_c, if it exists, returns d_i, if it does not exists, runs a *CL-public key query*, then the queried user's secret value d_i is returned.
- *CL-partial key query*: For adversary A_2, it knows the master secret key, so it can compute user's partial private key and is no need to perform this query. When this query on ID_i is submitted by A_1, C checks list LK_c, if the corresponding tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i)$ exists in LK_c and the related value is available, returns u_i to A_1. Otherwise, C performs a *CL-public key query*, then the queried user's partial key u_i and T_i will be returned.
- *CL-public key query*: A_l submits this on ID_i. In the case of $ID_x \neq ID_i$, challenger C checks if the tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i)$ exists in LK_c, if the corresponding tuple exists in LK_c, C provides A_l with $PK_i=(T_i, PPK_i)$, if it does not exists, C selects $d_i, t_i, r_i \in Z_{q_1}$ randomly, then computes $PPK_i=d_iP$, $D_i = r_iP$, $u_i = r_i + \alpha(t_i + 1) (\mathrm{mod}\, q_1)$ and $T_i=D_i+t_iP_{pub}$. Finally, C returns $PK_i=(T_i, PPK_i)$ as the response, inserts $(ID_i, d_i, T_i, u_i, PPK_i, r_i)$ and (ID_i, D_i, t_i) into LK_c and L_1 respectively. If $ID_x = ID_i$, C selects $t_x, r_x \in Z_{q_1}$ randomly, then computes $D_x = r_xP$, $u_x = r_x + \alpha(t_x + 1) (\mathrm{mod}\, q_1)$, $T_x=D_x+t_xP_{pub}$ and sets $PPK_x=aP$, then inserts $(ID_x, \perp, T_x, u_x, aP, r_x)$ and (ID_x, D_x, t_x) into LK_c and L_1 respectively, and returns $PK_x=(T_x, PPK_x)$ to A_l.
- *CL-replace public key query*: In addition to ID_x, any sender's public key is easy to be replaced by A_1. A_2 is not allowed to perform this query.
- *CL-PKI-SE query*: A_l submits this query with a keyword w, a sender's identity ID_i and a receiver's identity ID_j. In the case of $ID_x \neq ID_i$ C generates ciphertext σ by running CP-PEKS algorithm and then sends it to A_l. Otherwise, this game is aborted by C.
- *PKI-public key query*: A_l submits this query on ID_j. In the case of $ID_y \neq ID_j$, challenger C firstly checks the list LK_p, PK_j is returned if the tuple (ID_j, d_j, PK_j) is found in the list LK_p. If the tuple (ID_j, d_j, PK_j) does not exist in PK_j, C picks $d_j \in Z_q^*$ at random and computes $PK_j=d_jP$ as return, then inserts (ID_j, d_j, PK_j) into the list LK_p. If $ID_y = ID_j$, C returns $PK_y=bP$ and inserts $(ID_y, \perp, bP)$ into the list LK_p.
- *PKI-trapdoor query*: When A_l submits this query with an identity ID_j and a keyword w, C aborts this game if $ID_y = ID_j$. Otherwise, C needs to search (w, h_w) from L_2, if the tuple (w, h_w) is found in the list L_2, then runs PKI-TG algorithm to compute T_w and returns it to A_l. Otherwise, C makes a H_2 *query* to obtain h_w, then inserts (w, h_w) into list L_2 and uses h_w to run PKI-TG algorithm to compute T_w. Finally, C returns T_w to A_l.

Challenge A_l sends C a sender's identity ID_A, a receiver's identity ID_B and a chosen pair of keywords (w_0, w_1), the restriction is that the *PKI-trapdoor query* on keywords (w_0, w_1) has never been asked before. If $ID_x \neq ID_A$ and $ID_y \neq ID_B$, C aborts this game. Otherwise, C randomly selects $\lambda \in \{0,1\}$, chooses $k, m \in Z_{q_1}^*$, $CP \in G_1$ and runs a H_2 *query* to acquire h_{w_λ}, then sets $R_{w_\lambda}^* = kh_{w_\lambda}^2 CP$, $y^* = m$ and returns $\sigma^* = (R^*, y^*)$ to A_l.

***Phase* 2** A_I can make more queries except for the *PKI-trapdoor* query on keywords w_0 and w_1.

Guess A_I outputs a bit λ' as its guess. In order to make a correct guess, A_I computed $R_{w_\lambda}=h_{w_\lambda}^2 kd_s PK_B = h_{w_\lambda}^2 kabP$. Hence C can use the value of k which is chosen by itself at the challenge phase and select h_{w_λ} from L_3, then compute $abP = h_{w_\lambda}^{-2} k^{-1} R_{w_\lambda}$ as the answer of the CDHP.

Finally, we can draw a conclusion that as long as A_I wins, C can settle the CDHP. Nevertheless, it is all know that mathematical difficulties such as CDHP cannot be solved effectively at present, which explains our scheme can realize ciphertext indistinguishability.

Theorem 2. Under the hypothesis of the complexity of BDHIP, the proposed CP-CHSKS achieves trapdoor indistinguishability to resist any adversary A_3's CKA in the ROM.

Proof: C chooses an instance of BDHIP (P, aP), where $a \in Z_q^*$ is unknown. The purpose of C is to compute $\hat{e}(P,P)^{1/a}$.

Initialization The same initialization is used in the proof of Theorem 2 as it was in Theorem 1.

***Phase* 1** In the proof of Theorem 2, the operations required for C in phase 1 are similar to the proof of Theorem 1. A_3 can make the queries executed by A_1 during the proof of Theorem 1. H_1 *query*, H_2 *query*, H_3 *query*, *CL-partial key query* and *PKI-trapdoor query* are the same as Theorem 1, other queries requiring different methods to answer in this phase are listed below:

- *CL-public key query*: When receiving a *CL-public key query* on ID_i submitted by A_3, C can normally provide A_3 with $PK_i=(T_i, PPK_i)$ without identity restriction.
- *CL-secret value query*: When A_3 submits this query on ID_i, C can normally returns A_3 user's secret value d_i, there is no identity restriction.
- *CL-replace public key query*: The public key of any sender can be replaced by A_3.
- *PKI-public key query*: If C receives a *PKI-public key query* on ID_j and $ID_y = ID_j$, C sets $PK_y = aP$, then returns PK_y to adversary and inserts $(ID_y, \perp, aP)$ into LK_p. Other operations are comparable to the proof of Theorem 1 in other cases.
- *CL-PKI-SE query*: A_3 submits this query with a keyword w, a sender's identity ID_i and a receiver's identity ID_j, C generates ciphertext σ by running CP-PEKS algorithm and then sends it to A_3.

Challenge Adversary A_3 provides challenger C with a receiver's identity ID_B and a pair of keywords (w_0, w_1), the restriction is that the *CL-PKI-SE query* and *PKI-trapdoor query* on keywords w_0 and w_0 have never been asked before. In the case of $ID_y \neq ID_B$, C aborts this game. Otherwise C selects $f \in Z_q^*$ randomly and $\lambda \in \{0,1\}$, then returns $T_{w_\lambda}^* = fP$ to A_3.

***Phase* 2** C allows A_3 to make more queries except for the *PKI-trapdoor query* and *CL-PKI-SE query* on keywords w_0 and w_1.

Guess A_3 outputs a bit λ' as its guess. We can draw a conclusion that as long as A_3 wins, which means A_3 worked out $T_{w_\lambda} = (h_{w_\lambda} d_y)^{-1} P = (h_{w_\lambda} a)^{-1} P$, then C can settle the BDHIP by computing $\hat{e}(T_{w_\lambda}, P)^{h_{w_\lambda}} = \hat{e}((h_{w_\lambda} d_y)^{-1} P, P)^{h_{w_\lambda}} = \hat{e}(P,P)^{d_y^{-1}} = \hat{e}(P,P)^{1/a}$. Nevertheless, it is all know that mathematical

difficulties such as BDHIP cannot be solved effectively at present, which explains our scheme can realize trapdoor indistinguishability.

Theorem 3. **U**nder the hypothesis of the complexity of DLP, the proposed CP-CHSKS achieves ciphertext unforgeability to resist any adversary A_1's IKGA in the ROM.

Proof: C and A_1 play Game 3 together. Given C a tuple (P, aP) of DLP, and a is unknown to C. The purpose of C is to compute a.

Initialization The same initialization is used in the proof of Theorem 3 as it was in Theorem 1.

***Phase* 1** In the proof of Theorem 3, the operations required for C in phase 1 are similar to the proof of Theorem 1 except that P_{pub} is set as aP. H_1 *query*, H_2 *query* and H_3 *query* are the same as Theorem 1, other queries requiring different methods to answer in this phase are listed below:

- *CL-secret value query*：Now, C can normally provide A_1 with the corresponding user's secret value d_i when receiving a *CL-secret value query* on ID_i.
- *CL-public key query*: Now, C needs to randomly determine a value $c_i \in \{0,1\}$ and use it to decide the progress of this game. When receiving a *CL-public key query* on ID_i submitted by A_1, C checks list LK_c, if the corresponding tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i, c_i)$ is found in LK_c, then $PK_i=(T_i, PPK_i)$ is returned. Otherwise, C selects $c_i \in \{0,1\}$, if $c_i = 1$, C chooses $r_i, d_i, t_i \in Z_{q_1}$ at random, then sets $D_i = r_i P$, $PPK_i = d_i P$ and $T_i = D_i + t_i P_{pub}$, and finally returns $PK_i=(T_i, PPK_i)$ as the answer, inserts tuple $(ID_i, d_i, T_i, \perp, PPK_i, r_i, 1)$ and (ID_i, D_i, t_i) into LK_c and L_1 respectively. If $c_i = 0$, C chooses $u_i, r_i, d_i, t_i \in Z_{q_1}$ randomly, then sets $D_i = r_i P$, $T_i = u_i P - P_{pub}$ and $PPK_i = d_i P$. Finally, C inserts tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i, 0)$ and (ID_i, D_i, t_i) into LK_c and L_1 respectively, $PK_i=(T_i, PPK_i)$ is returned.
- *CL-partial key query*: A_1 submits this query on ID_i, If the ID_i related tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i, c_i)$ exists in LK_c and $c_i = 1$, the challenger stops the simulation. Otherwise, C returns the partial private key u_i to A_1.
- *CL-replace public key query*: As long as the adversary chooses a legitimate value, any sender's public key can be replaced.
- *CL-PKI-SE query*: A_1 submits this query on sender's identity ID_i, If the ID_i related tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i, c_i)$ exists in LK_c and $c_i = 1$, the challenger stops the simulation.
- *PKI-trapdoor query*: C is no need to consider the case of $ID_y = ID_j$, it can answer this query normally.
- *PKI-public key query*: C is no need to consider the case of $ID_y = ID_j$, it can answer this query normally.

Forgery Now, a forged ciphertext $\sigma_w^* = (R^*, y^*)$, a sender's identity ID_A and a receiver's identity ID_B are output by A_1. Through the above process, the conditions defined in the definition of Game 3 should be met. If the ID_A related tuple $(ID_A, d_A, T_A, u_A, PPK_A, r_A, c_A)$ exists in LK_c and $c_i = 0$, the challenger stops the simulation. Otherwise, according to the forking lemma in literature [31], another valid keyword ciphertext $\sigma_w^\Upsilon = (R^\Upsilon, y^\Upsilon)$ can be generated in the same way, so we can get $y^* = h^{-1^*} d_A (h_w^* k + 1) / u_A$, $y^\Upsilon = h^{-1^\Upsilon} d_A (h_w^\Upsilon k + 1) / u_A$, then C obtains $y^* = h^{-1^*} d_A (h_w^* k + 1) / r_A + a(t_A + 1)$,

$y^{\Upsilon} = h^{-1^{\Upsilon}} d_A(h_w^{\Upsilon} k + 1) / r_A + a(t_A + 1)$. Finally, C returns $a = \frac{(h^{-1^*} h_w^* - h^{-1^{\Upsilon}} h_w^{\Upsilon}) d_A k + (h^* - h^{\Upsilon}) d_A - r_A (y^* - y^{\Upsilon})}{(t_A + 1)(y^* - y^{\Upsilon})}$ as the answer of DLP. The proof is as follows:

$$y^* - y^{\Upsilon} = \frac{h^{-1^*} d_A(h_w^* k + 1)}{r_A + a(t_A + 1)} - \frac{h^{-1^{\Upsilon}} d_A(h_w^{\Upsilon} k + 1)}{r_A + a(t_A + 1)}$$

$$= \frac{(h^{-1^*} h_w^* - h^{-1^{\Upsilon}} h_w^{\Upsilon}) d_A k + (h^{-1^*} - h^{-1^{\Upsilon}}) d_A}{r_A + a(t_A + 1)}$$

$$a = \frac{(h^{-1^*} h_w^* - h^{-1^{\Upsilon}} h_w^{\Upsilon}) d_A k + (h^{-1^*} - h^{-1^{\Upsilon}}) d_A - r_A (y^* - y^{\Upsilon})}{(t_A + 1)(y^* - y^{\Upsilon})}$$

From the statements above, we can draw a conclusion that as long as the keyword ciphertext is successfully forged by A_1, C is certain to solve DLP. Nevertheless, it is all know that mathematical difficulties such as DLP cannot be solved effectively at present which explains our scheme is resistant to IKGA.

Theorem 4. Under the hypothesis of the complexity of DLP, the proposed CP-CHSKS achieves ciphertext unforgeability to resist any adversary A_2's IKGA in the ROM.

Proof: C and A_2 play Game 3 together. Given C a tuple (P, aP) of DLP, and a is unknown to C. The purpose of C is to calculate a.

Initialization. The same initialization is used in the proof of Theorem 4 as it was in Theorem 1.

***Phase* 1** In the proof of Theorem 4, the operations required for C in phase 1 are similar to the proof of Theorem 1. H_1 *query*, H_2 *query*, H_3 *query*, *CL-partial key query* and *CL-PKI-SE query* are the same as Theorem 1, other queries requiring different methods to answer in this phase are listed below:

- *CL-secret value query*: Now, C needs to distinguish identity, if $ID_x = ID_i$, C aborts this game. Otherwise, it can normally provide A_2 with the corresponding user's secret value d_i when receiving a *CL-secret value query* on ID_i.
- *CL-public key query*: A_2 submits this on ID_i. In the case of $ID_x \neq ID_i$, challenger C checks if the tuple $(ID_i, d_i, T_i, u_i, PPK_i, r_i)$ exists in LK_c, if the corresponding tuple exists in LK_c, C provides A_2 with $PK_i = (T_i, PPK_i)$, if it does not exists, C selects $d_i, t_i, r_i \in Z_{q_1}$ randomly, then computes $PPK_i = d_i P$, $D_i = r_i P$, $u_i = r_i + \alpha(t_i + 1) (\mathrm{mod}\, q_1)$ and $T_i = D_i + t_i P_{pub}$. Finally, C returns $PK_i = (T_i, PPK_i)$ as the response, inserts $(ID_i, d_i, T_i, u_i, PPK_i, r_i)$ and (ID_i, D_i, t_i) into LK_c and L_1 respectively. If $ID_x = ID_i$, C selects $t_x, r_x \in Z_{q_1}$ randomly, then computes $D_x = r_x P$, $u_x = r_x + \alpha(t_x + 1) (\mathrm{mod}\, q_1)$, $T_x = D_x + t_x P_{pub}$ and sets $PPK_x = aP$, then inserts $(ID_x, \perp, T_x, u_x, aP, r_x)$ and (ID_x, D_x, t_x) into LK_c and L_1 respectively, and returns $PK_x = (T_x, PPK_x)$ to A_2.
- *CL-replace public key query*: C has no chance to perform this query.
- *PKI-trapdoor query*: C is no need to consider the case of $ID_y = ID_j$, it can answer this query normally.
- *PKI-public key query*: C is no need to distinguish identity, it can normally provide A_2 with the corresponding user's public key PK_j when receiving a *PKI-public key query* on ID_j.

Forgery Now, a forged ciphertext $\sigma_w^* = (R^*, y^*)$, a sender's identity ID_A and a receiver's identity ID_B are output by A_2. Through the above process, the conditions defined in the definition of Game 3 should be met. If $ID_A \neq ID_x$, the challenger stops the simulation. Otherwise, another valid keyword

ciphertext $\sigma_w^{\Upsilon} = (R^{\Upsilon}, y^{\Upsilon})$ can be generated in the same way, then C obtains $y^* = h^{-1^*} d_x (h_w^* k + 1) / u_x$, $y^{\Upsilon} = h^{-1^{\Upsilon}} d_x (h_w^{\Upsilon} k + 1) / u_x$. Finally, C returns $a = \frac{(y^* - y^{\Upsilon}) u_x}{(h^* h_w^* - h^{\Upsilon} h_w^{\Upsilon}) k + (h^* - h^{\Upsilon})}$ as the answer of DLP. The proof is as follows:

$$y^* - y^{\Upsilon} = \frac{h^{-1^*} a(h_w^* k + 1)}{u_x} - \frac{h^{-1^{\Upsilon}} a(h_w^{\Upsilon} k + 1)}{u_x}$$

$$= \frac{[(h^{-1^*} h_w^* - h^{-1^{\Upsilon}} h_w^{\Upsilon}) k + (h^{-1^*} - h^{-1^{\Upsilon}})] a}{u_x}$$

$$a = \frac{(y^* - y^{\Upsilon}) u_x}{(h^{-1^*} h_w^* - h^{-1^{\Upsilon}} h_w^{\Upsilon}) k + (h^{-1^*} - h^{-1^{\Upsilon}})}$$

From the statements above, we can draw a conclusion that as long as the keyword ciphertext is successfully forged by A_2, C is certain to solve DLP. Nevertheless, it is all know that mathematical difficulties such as DLP cannot be solved effectively at present. Therefore, our scheme has ideal ciphertext unforgeability and is able to against any adversary A_2's IKGA in the ROM.

6 Performance analysis

In order to enable a reasonable evaluation of our scheme, in this section, we chose five existing schemes ([25], [28], [29], [30], [31]) to compare with ours in the field of computation cost, features, and communication overhead.

6.1 Computation cost and features comparison

In order to make the comparison results more intuitive, we conducted quantitative comparative analysis. The MIRACLE library was run on a personal computer with an Intel 2.90 GHz CPU, 4GB RAM to obtain experimental data, and this experimental environment is similar to that of scheme [27]. Table 1 summarizes the calculation symbols used and the corresponding time required for the calculation represented by these symbols. The computation cost and features comparison results are shown in Table 2. DDDCSP indicates that users in different domains are able to use different cryptographic system parameters, and $(l+)$ $(l \in N)$ denotes l operations that can be calculated offline, the cost of offline computation is not included in our comparison results. Fig. 2 shows the comparative results of the computation cost in the form of a column chart.

From Table 2 and Fig. 2, we can clearly see that our scheme has outstanding performance. Compared with [25], [28], [29], [30] and [31], our scheme has a considerably lower total computation cost than the other five schemes, the total computation cost of our scheme decreased by about 87.61%, 69.93%, 71.83%, 61.17% and 59.99%, respectively. In addition to the excellent computation cost, our scheme can resist IKGA while schemes [25] and [30] cannot. In the test phase of scheme [25], the tester needs to obtain the hash value of the keyword, which means that the tester needs to get the keyword information. As for scheme [30], [30] allows trapdoors to propagate over public channels by specifying two test servers. However, although two test servers are specified in [30], due to the lack of the sender's private key at the phase of keyword ciphertext generation, internal attackers such as two collusive servers can still execute IKGA. Also, our scheme allows communication entities in different domains to use different cryptographic system parameters, but all the schemes added to the comparison do not have this feature. Now, we can get the result that our scheme has better efficiency in the applications of WBAN.

Table 1 Notations

Symbols	Explanation	Time(ms)
T_{sm}	The amount of time it takes to do a scalar multiplication.	2.165
T_p	The amount of time it takes to do a bilinear pairing operation	5.427

T_H	The amount of time it takes to do a hash to point operation	5.493
T_h	The amount of time it takes to do a general hash function operation	0.007
T_a	The amount of time it takes to do a point addition operation	0.013
T_e	The amount of time it takes to do an exponentiation operation in G_2	0.339

Table 2 Computation cost and features comparison

Schemes	Computation cost				Features	
	Ciphertext	Trapdoor	Test	Total	IKGA	DDDCSP
Zhang [25]	$(1+)2T_H+2T_h+5T_{sm}+2T_p+(2+)1T_a=32.692$	$(1+)1T_H+T_h+3T_{sm}+2T_a=12.021$	$2T_H+2T_h+2T_{sm}+3T_a+(1+)4T_p=37.007$	81.72	No	No
Yang [28]	$(1+)2T_p+(1+)2T_{sm}+(2+)0T_H+(1+)0T_a+2T_e+(2+)1T_h=15.869$	$(1+)0T_p+(1+)3T_{sm}+2T_a+(2+)1T_h=6.528$	$2T_p+T_a+T_e=11.206$	33.603	Yes	No
He [29]	$T_H+(2+)3T_{sm}+(2+)1T_a+(3+)0T_h=12.001$	$T_H+(2+)1T_{sm}+T_p+(3+)0T_h+(2+)0T_a=13.085$	$2T_p+(2+)0T_h+(2+)0T_{sm}+(2+)0T_a=10.854$	35.94	Yes	No
Ma [30]	$(6+)4T_{sm}+(6+)2T_a+(6+)1T_h=8.693$	$(4+)3T_{sm}+(4+)1T_a+(5+)1T_h=6.515$	$5T_{sm}+(3+)3T_a+(2+)0T_h=10.864$	26.072	No	No
Omala [31]	$3T_{sm}+2T_h+T_e=6.848$	$T_h+T_{sm}=2.172$	$3T_p+(2+)0T_a+(2+)0T_h+(1+)0T_{sm}=16.281$	25.301	Yes	No
Ours	$2T_h+T_{sm}=2.179$	$T_h+T_{sm}=2.172$	$(2+)1T_p+T_e+T_h=5.773$	10.124	Yes	Yes

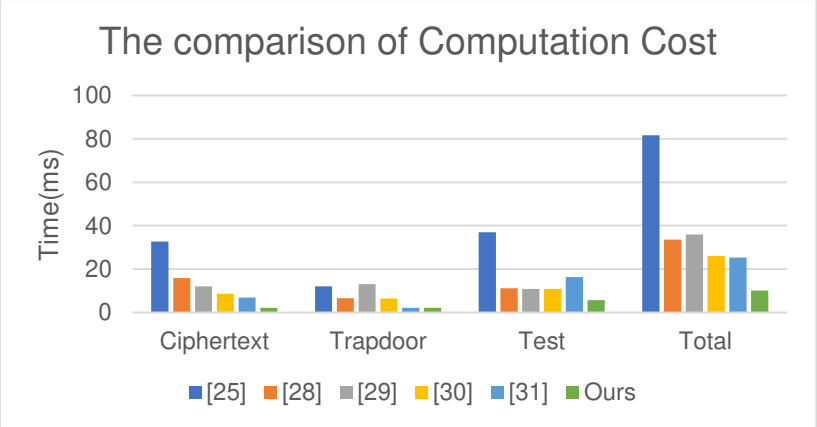

Fig. 2 Comparison of computation cost

6.2 Communication overhead comparison

In this part, we compared the communication overhead of our scheme with that of the other five schemes [25, 28-31] based on the respective sizes of the public key, ciphertext and trapdoor. To obtain a more concise description, we make $|CT|$ and $|TD|$ respectively denote the size of ciphertext and trapdoor. At the same time, the size of an element in G_1, G_2 and Z_q^* are denoted as $|G_1|$, $|G_2|$ and $|Z_q^*|$

respectively. According to reference [32], let $|G_1| = 512$ bits, $|G_2| = 1024$ bits, $|Z_q^*| = 160$ bits, and Table 3 shows the comparative results of communication overhead.

Table 3 Communication Overhead Comparison

Schemes	*CT*	*TD*	Total
Zhang [25]	$3|G_1| + |Z_q^*|$	$2|G_1|$	2720 bits
Yang [28]	$2|G_1| + |G_2|$	$2|G_1|$	3072 bits
He [29]	$|G_1| + |Z_q^*|$	$|G_1|$	1184 bits
Ma [30]	$2|G_1|$	$2|G_1|$	2048 bits
Omala [31]	$3|G_1|$	$|G_1|$	2048 bits
Ours	$|G_1| + |Z_q^*|$	$|G_1|$	1184 bits

According to Table 3, our scheme requires only 1184 bits for total communication, which is the same as scheme [29] while the other four schemes ([25], [28], [30] and [31]) need more. Therefore, it is clear that our scheme is superior to schemes [25], [28], [30], and [31] in the field of communication overhead. Now, it is possible to conclude that our scheme is feasible in WBAN since it has low communication burden.

7 Conclusions

On the basis of our research, we presented a new PEKS scheme for WBAN in this paper that combines the features of PEKS and authentication, called cross-domain heterogeneous signcryption with keyword search scheme. The proposed scheme is shown to fulfill ciphertext indistinguishability, trapdoor indistinguishability, and ciphertext unforgeability in the ROM, indicating that it is immune to both CKA and IKGA. In addition, our scheme allows senders and receivers to communicate in different domains and use different cryptographic system parameters, making it more suitable for actual WBAN applications. Compared with [25], [28], [29], [30] and [31], the total computation cost of our scheme decreased about 87.61%, 69.93%, 71.83%, 61.17% and 59.99%, respectively. In addition, our scheme requires a low communication overhead of only 1184 bits in total. To sum up, the scheme we proposed has excellent performance and is practical for the application of WBAN.

Declarations

Ethical approval and consent to participate

Not applicable.

Human and animal ethics

This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication

Not applicable.

Availability of supporting data

No dataset was generated or analyzed during this study.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Funding

This study was funded by the National Natural Science Foundation of China (grant number 61862042).

Authors' contributions

Ming Luo and Dashi Huang wrote the main manuscript text, Minrong Qiu prepared tables 1-3 and figures 1-2. All authors reviewed and approved the final version of the manuscript.

Acknowledgements

The authors thank the anonymous reviewers for their valuable suggestions and comments.

Authors' information

Ming Luo Ming Luo received the Ph.D degree in computer application technology from Northeastern University, Shenyang, China in 2010. He is currently a professor and the deputy dean of the School of Software, Nanchang University, Nanchang, China. He worked as a Visiting scholar with the Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia from August 2018 to August 2019. His research interests cover Internet of Things, Cyberspace Security and cryptography.

Dashi Huang Dashi Huang received his B.E. degree from the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China in June 2019. He is currently pursuing his M.E degree from the School of Software, Nanchang University. His current research interests include information security and cryptography.

Minrong Qiu Minrong Qiu received Ph.D degree in industrial economics from Wuhan University of Technology, Wuhan, China in 2018. She is currently a associate professor of the GongQing Institute of Science and Technology, Nanchang, China. Her research interests cover Internet of Things, Cyberspace Security and Information System Management.

Reference

[1] Arfat, Y., Usman, S., Mehmood, R., Katib, I.: Big data tools, technologies, and applications: A survey. In: Smart Infrastructure and Applications. pp. 453–490. Springer, Switzerland (2020). https://doi.org/10. 1007/978-3-030-13705-2_19.

[2] Ashabi, A., Sahibuddin, S. Bin, Haghighi, M.S.: Big Data: Current Challenges and Future Scope. In: IEEE 10th Symposium on Computer Applications and Industrial Electronics (ISCAIE 2020). pp. 131–134. IEEE (2020). https://doi.org/10.1109/ISCAIE47305.2020. 9108826.

[3] Ullah, I., Amin, N.U., Khan, M.A., Khattak, H., Kumari, S.: An Efficient and Provable Secure Certificate-Based Combined Signature, Encryption and Signcryption Scheme for Internet of Things

(IoT) in Mobile Health (M-Health) System. J. Med. Syst. 45, 4 (2021). https://doi.org/10.1007/s10916-020-01658-8.
[4] Karthiga, I., Sankar, S.: An IoT-based secure data transmission in WBSN. Int. J. Cloud Comput. 9, 311–329 (2020). https://doi.org/10.1504/IJCC. 2020.109383.
[5] Mohammed Sadeeq, M., Abdulkareem, N.M., Zeebaree, S.R.M., Mikaeel Ahmed, D., Saifullah Sami, A., Zebari, R.R.: IoT and Cloud Computing Issues, Challenges and Opportunities: A Review. Qubahan Acad. J. 1, 1–7 (2021). https://doi.org/10. 48161/qaj.v1n2a36.
[6] Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE symposium on security and privacy. pp. 44–55. IEEE (2000). https://doi.org/10.1109/secpri.2000.848445.
[7] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. J. Cryptol. 21, 350–391 (2008). https://doi.org/10.1007/s00145-007-9006-6.
[8] Priya, N., Ponnavaikko, M.: Keyword search with two-side verification in encrypted data using blockchain. In: 2020 International Conference on Computer Communication and Informatics (ICCCI 2020). pp. 1–5. IEEE (2020). https://doi.org/10. 1109/ICCCI48352.2020.9104169.
[9] Das, D., Amin, R., Kalra, S.: Algorithm for Multi Keyword Search over Encrypted Data in Cloud Environment. In: 2020 International Wireless Communications and Mobile Computing (IWCMC). pp. 733–739. IEEE (2020). https://doi.org/10. 1109/IWCMC48107.2020.9148472.
[10] Chai, Q., Gong, G.: Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In: 2012 IEEE International Conference on Communications (ICC). pp. 917–922. IEEE (2012). https://doi.org/10.1109/ICC.2012.6364125.
[11] Shi, Z., Fu, X., Li, X., Zhu, K.: ESVSSE: Enabling Efficient, Secure, Verifiable Searchable Symmetric Encryption. IEEE Trans. Knowl. Data Eng. (2020). https://doi.org/10.1109/tkde.2020.3025348.
[12] Najafi, A., Javadi, H.H.S., Bayat, M.: Efficient and dynamic verifiable multi-keyword searchable symmetric encryption with full security. Multimed. Tools Appl. 80, 26049–26068 (2021). https://doi.org/10.1007/s11042-021-10844-w.
[13] Zhang, Y., Li, Y., Wang, Y.: Efficient Searchable Symmetric Encryption Supporting Dynamic Multikeyword Ranked Search. Secur. Commun. Networks (2020). https://doi.org/10.1155/2020/7298518.
[14] Li, L., Xu, C., Liu, Z., Mei, L.: Forward Secure Conjunctive-Keyword Searchable Symmetric Encryption Using Shamir Threshold Secret Sharing Scheme. In: Communications in Computer and Information Science. pp. 14–28. Springer, Singapore (2020). https://doi.org/10. 1007/978-981-33-4706-9_2.
[15] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: International conference on the theory and applications of cryptographic techniques. pp. 506-522. Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30.
[16] Park, D.J., Kim, K., Lee, P.J.: Public key encryption with conjunctive field keyword search. In: International Workshop on Information Security Applications. pp. 73–86. Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-31815-6_7.
[17] Zhang, B., Zhang, F.: An efficient public key encryption with conjunctive-subset keywords search. J. Netw. Comput. Appl. 34, 262–267 (2011). https://doi.org/10.1016/j.jnca.2010.07.007.
[18] Byun, J.W., Rhee, H.S., Park, H.A., Lee, D.H.: Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In: Workshop on secure data management. pp. 75–83. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/11844662_6.
[19] Baek, J., Safavi-Naini, R., Susilo, W.: Public key encryption with keyword search revisited. In: International conference on Computational Science and Its Applications. pp. 1249–1259. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69839-5_96.
[20] Ma, S., Huang, Q., Zhang, M., Yang, B.: Efficient public key encryption with equality test supporting flexible authorization. IEEE Trans. Inf. Forensics Secur. 10, 458–470 (2015). https://doi.org/10.1109/TIFS.2014. 2378592.
[21] Wang, C.H., Tu, T.Y.: Keyword search encryption scheme resistant against keyword-guessing attack by the untrusted server. J. Shanghai Jiaotong Univ. 19, 440–442 (2014). https://doi.org/10.1007/s12204-014-1522-6.
[22] Huang, Q., Li, H.: An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. Inf. Sci. (Ny). 403–404, (2017). https://doi.org/10.1016/j.ins.2017.03.038.
[23] Liu, Z.Y., Tseng, Y.F., Tso, R., Mambo, M.: Designated-ciphertext searchable encryption. J. Inf. Secur. Appl. 58, 102709 (2021). https://doi.org/10.1016/j.jisa. 2020.102709.

[24] Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: International conference on the theory and application of cryptology and information security. pp. 452–473. Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_29.
[25] Zhang, Y., Liu, X., Lang, X., Zhang, Y., Wang, C.: VCLPKES: Verifiable Certificateless Public Key Searchable Encryption Scheme for Industrial Internet of Things. IEEE Access. 8, 20849–20861 (2020). https://doi.org/10.1109/ACCESS.2020.2968501.
[26] Pakniat, N.: Designated tester certificateless encryption with keyword search. J. Inf. Secur. Appl. 49, 102394 (2019). https://doi.org/10.1016/j.jisa. 2019.102394.
[27] Ma, M., He, D., Khan, M.K., Chen, J.: Certificateless searchable public key encryption scheme for mobile healthcare system. Comput. Electr. Eng. 65, 413–424 (2018). https://doi.org/10.1016/j.compeleceng.2017.05. 014.
[28] Yang, G., Guo, J., Han, L., Liu, X., Tian, C.: An improved secure certificateless public-key searchable encryption scheme with multi-trapdoor privacy. Peer-to-Peer Netw. Appl. 15, 503–515 (2022). https://doi.org/10.1007/s12083-021-01253-9.
[29] He, D., Ma, M., Zeadally, S., Kumar, N., Liang, K.: Certificateless public key authenticated encryption with keyword search for industrial internet of things. IEEE Trans. Ind. Informatics. 14, 3618–3627 (2018). https://doi.org/10.1109/TII.2017.2771382.
[30] Ma, M., Luo, M., Fan, S., Feng, D.: An Efficient Pairing-Free Certificateless Searchable Public Key Encryption for Cloud-Based IIoT. Wirel. Commun. Mob. Comput. 2020, 8850520 (2020). https://doi.org/10.1155/2020/8850520.
[31] Omala, A.A., Ali, I., Li, F.: Heterogeneous signcryption with keyword search for wireless body area network. Secur. Priv. (2018). https://doi.org/10. 1002/spy2.25.
[32] Lu, Y., Li, J., Wang, F.: Pairing-Free Certificate-Based Searchable Encryption Supporting Privacy-Preserving Keyword Search Function for IIoTs. IEEE Trans. Ind. Informatics. 17, 2696–2706 (2021). https://doi.org/10.1109/TII.2020.3006474.