IBEET-RSA: Identity-Based Encryption with Equality Test over RSA for Wireless Body Area Networks

Mohammed Ramadan[1] · Yongjian Liao[1] · Fagen Li[2] · Shijie Zhou[1] · Hisham Abdalla

Abstract

Wireless body area network (WBAN) constitutes a widely implemented technique for remote acquisition and monitoring of patient health-related information via the use of embodied sensors. Given that security and privacy protection, including, but not limited to user authentication, integrity, and confidentiality, are both key challenges and a matter of deep concern when it comes to the deployment of emerging technologies in healthcare applications, state-of-the-art measures and solutions are needed to fully address security and privacy concerns in an effective and sensible manner by considering all the benefits and limitation of remote healthcare systems. In this paper, we proposed an efficient and secure identity-based encryption scheme under the RSA assumption providing equality test. We then proved the security of our scheme for one-way secure against chosen-identity and chosen-ciphertext attacks (OW-ID-CCA) by means of the random oracle model. The performance evaluation results indicated that our scheme outperforms other security schemes in terms of providing relatively low computational cost and stable compatibility with WBAN applications.

Keywords Public key encryption · Wireless body area network · Identity-based encryption · Equality test

1 Introduction

Present-day wireless communication technologies have been crucial in the development of medical and healthcare systems through a wide range of medical sensors, located inside and outside of the human body, and other electronic devices aimed at offering strong monitoring capacity and other applications [1]. A key element and one of the most innovative and advanced forms of the wearable monitoring technologies are the so-called wireless body area networks or WBANs. WBANs consist of several heterogeneous biological sensors placed in and around different parts of the body that can be wearable or implantable beneath the user's skin. Each of these sensors meets specific requirements and is used for different purposes. For instance, some devices can be used for assessing and monitoring changes in a patient's vital signs, whereas other devices can be used for predicting and/or detecting behavioral and physiological responses, most typically categorized as either fear, anxiety, stress, etcetera. In the case of the second application, behavioral and physiological states are communicated using special coordinator nodes. These are responsible for sending patient's biological signals to the physician, so that the physician can provide real-time medical diagnoses and allow patients to make informed decisions about their healthcare [2].

WBAN applications cover a wide range of fields aimed at improving the quality of life of the users [3]. These applications can be categorized depending on their association with the medical field versus a non-medical field. However, here we are mainly concerned with the applications of WBAN technologies in medical and healthcare fields. These applications focus primarily on providing health care solutions for aging populations, involving, for instance, early

✉ Yongjian Liao
liaoyj@uestc.edu.cn

Mohammed Ramadan
nopatia@gmail.com

Fagen Li
fagenli@uestc.edu.cn

Shijie Zhou
sjzhou@uestc.edu.cn

Hisham Abdalla
hisham_awaw@hotmail.com

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, People's Republic of China

[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, People's Republic of China

detection, prevention, and monitoring of diseases, as well as elderly assistance at home, rehabilitation after surgeries, and biofeedback and assisted living [4]. In a general sense, health monitoring systems based on wearable sensors can be classified either as physiological or biokinetic. As for physiological sensors, these are used to measure human body vital signals, apart from being in charge of gathering and measuring human body parameters internally or externally, including, but not limited to, body temperature, blood pressure, heart rate, respiratory rate, electrocardiogram (ECG), electroencephalogram (EEG), electromyogram (EMG), blood glucose pollution level, and so on. Furthermore, biokinetic sensors are largely used to collect human body movement based on signals as acceleration or angular rate of rotation. Finally, ambient sensors are responsible for offering additional information about surrounding environment temperature such as pressure, light, humidity, among other functions [5].

WBAN sensors in the form of portable devices, especially smartphones, can remotely access the information detected by the sensor and bridge the gap between the patient's medical data and the physician or health provider. The decisions or actions taken by the physician or health provider after monitoring and collection of patient's health record data using wearable sensors are key to medical treatment, and thus medical data must be accurate and protected against unsanctioned access that could be hazardous to the patient's life. This is the reason why strict security mechanisms to protect this data must not only be mandatory, but must also include a high level of system security and privacy [6–8].

Related work Only three schemes aimed at providing equality test over identity-based cryptosystems have been proposed in the past. These schemes can be explained as follows (Fig. 1).

In 2016, Ma [9] proposed an identity-based encryption with outsourced equality test scheme in cloud computing. This scheme was a combination of public key encryption with equality test (PKEET) and identity-based encryption (IBE) to provide identity-based encryption with equality test (IBEET). In this scheme, the receiver computes a trapdoor using the secret value for the identity, then sends it to a cloud server for equality test on its ciphertexts with others' ciphertexts. Using this primitive someone with the trapdoor for its identity can delegate out the capability of equality test on its ciphertexts, without requiring a central authority to act as a delegator.

In 2016, Hyung et al. [10] suggested a semi-generic construction of public key encryption and identity-based encryption scheme with equality test. This scheme only considered the authorization for equality test on all receiver's ciphertexts, and assumes only the existence of IND-CCA2 secure traditional public key encryption (PKE) schemes, the hardness of Computational Diffie-Hellman (CDH) problems, and random oracles.

Finally, in 2017 Wu et al. [11] surveyed and initiated an efficient and secure identity-based encryption scheme with equality test in cloud computing. The scheme had relatively low time-consuming HashToPoint function as compared to Ma's [9]. Besides, the security of this scheme has been proven for one-way secure against chosen-identity and chosen-ciphertext attacks (OW-ID-CCA).

We only covered the research works that related to identity-based with equality test approaches. All of this while making reference to the works of many other researchers [12–18] whose research works focused on public key encryption with equality test (PKEET).

Our contributions The contributions of this paper can be summarized as follows:

1. We propose a concrete identity-based encryption scheme from RSA, which can provide a secure and flexible security model.
2. Our proposed scheme provides identity-based cryptosystem with equality test. Thus, it supports authorization of the medical server to achieve an equality test on ciphertexts through a trapdoor, which is useful for WBAN applications.
3. We prove the security of the new IBC-RSA scheme. The security of our construction is based on the RSA assumption and one-way secure against chosen-identity and chosen-ciphertext attacks (OW-ID-CCA).

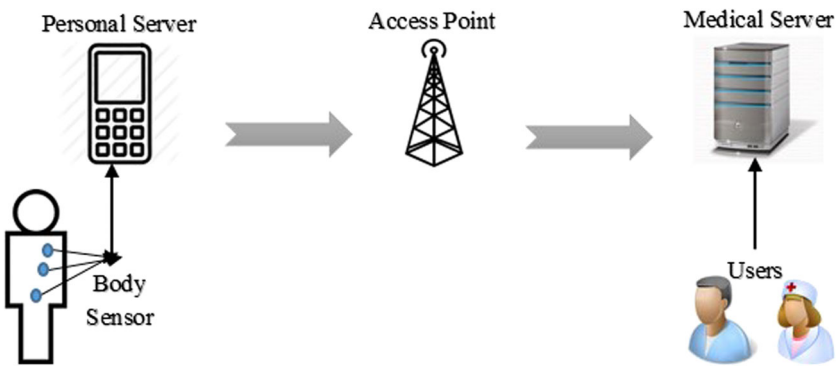

Fig. 1 WBAN architecture

4. We show that the computational cost of our proposed scheme is more efficient as compared to other proposed approaches.

Paper organization The rest of the paper is organized in this fashion.

In Section 2, we review some preliminaries of our proposed scheme. In Section 3, we provide a definition for our proposed IBEET-RSA scheme and its security notion. In Section 4, we explain our IBEET-RSA scheme describe the proposed model. In Section 5, we evaluate the proposed scheme and include security proofs and computational cost aspects. Finally, we conclude the paper in Section 6.

2 Preliminaries

2.1 RSA assumption

In 1978, RSA established a public key cryptosystem that is based on the difficulty of integer factoring. The RSA public key encryption scheme is the first example of a verifiable secure public key encryption scheme against chosen message chosen attacks [19]. In the RSA encryption scheme, the public key consists of $n = pq$, where p and q are primes, and an exponent e, where e is relatively prime to $\Phi(n) = (p-1)(q-1)$ and e should be chosen randomly, kept secret, and satisfy that $ed \equiv 1(mod\Phi(n))$. To encrypt a message $M : C = M^e(modn)$. To decrypt a ciphertext $C : M = (C)^d(modn) = (M^e)^d(modn)$.

Factorization attack This is big issue for security of RSA algorithm because the security of RSA is based on the idea that the modulus is so large that is infeasible to factor it in reasonable time. If the Adversary can factor n and obtain p and q, then can calculate $d = e^{-1}mod\Phi(n)$ and e is public parameter; and the private parameter d is the trapdoor that the Adversary uses to decrypt any encrypted message. Some existing factorization techniques [20–26] can be generating public and private key of RSA algorithm, by factorization of N. However, they are taking a considerable length of time to do so, in case of p and q are very large prime numbers. These methods will work efficiently if the key d is small. However, in our scheme n denotes the product of four big prime numbers p_1, p_2, p_3, p_4 with length at least 256 bit; the numbers $(p_i - 1)/2$ are odd and pairwise relatively prime. Till present there is no technique can achieve factorization of n-1024 length. Therefore, our construction is secure against factorization attack.

2.2 Discrete logarithm problem (DLP)

The DLP problem is as follows:

Given: $x, y \in Z_p^*$, DLP problem is to compute $a \in (Z_p - 1)$ such that $x^a = ymodp$. DLP assumption is hard problem for every probabilistic polynomial-time algorithm, the probability of solving DLP problem is negligible.

2.3 Decision Diffie-Hellman problem (DDH)

DDH Problem is as follows:

Let G be a multiplicative finite cyclic group and g be a generator of group G, and let P be a prime order of group G. Let g^x, g^y and $Z \in G$ be given where x, y are chosen randomly from Z_p^*. DDH problem is to decide whether $Z = g^{xy} \in G$. DDH assumption is hard problem for every probabilistic polynomial-time algorithm, the probability of solving DDH problem is negligible.

3 IBEET: model description

3.1 IBEET: system model

In this section, we present the proposed model description of our proposed scheme IBEET-RSA for WBAN, which will provide a secure and lightweight cryptosystem with equality test feature. The proposed scheme involves a multistep systematic approach. First, the medical sensors sense the patient's body signals that are crucial for medical check. Then, the information that is gathered by the sensors after encryption has taken place is transferred to the hospital network manager, or the individual responsible for processing the message. After the new data has been received, the medical sensors compare the new data with the initial form of data and, if no changes are detected in the patient's health, the system responds by rejecting the message and also by failing the equality test. The operation concludes with another message sent on to the hospital network manager, this time with the encrypted information, which is later transferred to the corresponding physician for further processing. However, it should be mentioned that only the patient and the physician are the ones able to either encrypt or decrypt the aforesaid information. Figure 2 illustrates the proposed IBEET-RSA security model and the handshaking process.

A: User A (Patient), B: Hospital (Physician), ET: Equality Tester (Hospital Network Manager), PKG: Public-Key Generator.

The proposed IBEET-RSA scheme consists of six algorithms: $Setup$, $KeyGen$, $Encrypt$, $Decrypt$, $Trapdoor$ and $Test$. These algorithms provide an efficient and secure cryptosystem and a flexible and reliable equality test technique to make the scheme more compatible with WBAN applications.

3.2 IBEET: security model

In this section, we define the security model of our IBEET-RSA scheme regarding to One-way chosen-ciphertext

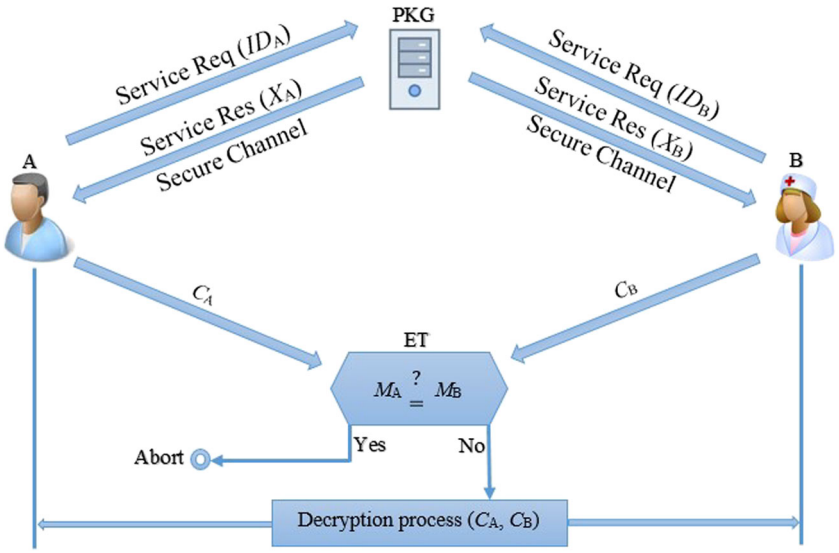

Fig. 2 Proposed IBEET-RSA security model

security against a chosen identity attack security model OW-ID-CCA. If the adversary can get the trapdoor still the adversary cannot be capable to get the plaintext from the corresponding challenge ciphertexts. The OW-ID-CCA security can be presented by the following games.

Assume that $\mathbb{Ch}$ denotes a challenger and $\mathcal{A}$ denotes an adversary.

- *Setup*: $\mathbb{Ch}$ takes a security parameter k as input and runs the *Setup* algorithm. Then $\mathbb{Ch}$ gives the system parameters *params* to the adversary $\mathcal{A}$ and keeps the master key d secure.
- Phase 1. Private decryption key queries (ID_i): $\mathbb{Ch}$ runs the $KeyGen$ algorithm to generate the private decryption key X_i corresponding to the public key (ID_i). Then $\mathbb{Ch}$ sends X_i it to $\mathcal{A}$.
- Phase 1. Trapdoor queries (ID_i): From the private decryption, key queries $\mathbb{Ch}$ can get X_i and generates the trapdoor T_i via the Trapdoor algorithm. Then $\mathbb{Ch}$ sends T_i to $\mathcal{A}$.
- Phase 1. Decryption queries (ID_i, C_i): $\mathbb{Ch}$ runs the Decrypt algorithm to decrypt the ciphertext C_i by running $KeyGen$ algorithm to get the private decryption key X_i corresponding to the public key (ID_i). Then $\mathbb{Ch}$ sends the plaintext M_i to $\mathcal{A}$.
- Challenge: $\mathbb{Ch}$ randomly chooses a plaintext $M \in M$ and runs the $Encryption$ algorithm to get the corresponding ciphertext C. Then $\mathbb{Ch}$ sends C to $\mathcal{A}$.
- Phase 2. Private decryption key queries: In case of $ID_i \neq ID_c$. $\mathbb{Ch}$ responds similar as in Phase 1.
- Phase 2. Trapdoor queries: $\mathbb{Ch}$ responds similar as in Phase 1.
- Phase 2. Decryption queries: In case of $(ID_i, C_i) \neq (ID_c, C)$. $\mathbb{Ch}$ responds similar as in Phase 1.
- Guess: $\mathcal{A}$ outputs M' as a guess of M.

Definition 1 The IBEET-RSA scheme is secure against OW-ID-CCA adversaries if the advantage of A to win the above game $Adv^{OW-ID-CCA(k)} = Pr[M = M']$ is negligible.

4 IBEET-RSA: scheme constructions

The proposed scheme involves a combination of identity-based cryptosystem with RSA algorithm for secure functioning of medical and healthcare monitoring systems [27–32]. First, the private keys are calculated by a trusted third party (PKG) using the master key. Second, the public key is the hash value of the user's identities and it can be computed by all users. Third, the private keys are computed and sent by the PKG to the corresponding users via a secure channel. This way the algorithms $SetPrivateKey$ and $SetPublicKey$ are embedded into one algorithm. However, the security of our proposed scheme rely on the hardness of n factorization rather than just solve the discrete logarithm problem.

4.1 IBEET-RSA: scheme description

The IBEET-RSA scheme consists of six different algorithms: $Setup$, $KeyGen$, $Encrypt$, $Decrypt$, $Trapdoor$, and $Test$. A detailed description of our proposed scheme is shown below:

- *Setup*: This algorithm is run by PKG as follows:
  1. The PKG generates RSA parameters (n, e, d), let $n = p_1p_2p_3p_4$; where $\mid p_1 \mid \approx \mid p_2 \mid \approx \mid p_3 \mid \approx \mid p_4 \mid \approx k/4$ in which p_i are primes, and $ed \equiv 1(mod\ \Phi(n))$; and d is kept secret as master key.
  2. Choose an element $g \in Z_n^*$.

3. Choose cryptographic hash functions, and then the message space: $M \in \{0,1\}^k$, $H_1 : \{0,1\}^k \longrightarrow Z_n^*$, $H_2 : \{0,1\}^k \times \{0,1\}^k \longrightarrow Z_n^*$, $H_3 : Z_n^* \longrightarrow \{0,1\}^k$, $H_4 : \{0,1\}^k \longrightarrow \{0,1\}^k$. The system parameters: $params = (g, e, n, H_1, H_2, H_3, H_4)$.

– *KeyGen*: This algorithm is run by PKG as follows:

1. Set the Public key: $Q_{ID} = H_1(ID)$.
2. Set the private key: $XID = log_g Q_{ID}^d (mod\ \Phi(n))$.

Note: The private key can be computed by using the Chinese Reminder Theory (CRT).

– *Encrypt*: To encrypt a message $M \in \{0,1\}^k$ between user A (patient) and user B (physician), through the hospital network manager ET, and by using their identities $ID_{A/B} \in \{0,1\}^*$ and their public/private keys, user A will start to encrypt M to user B using params and user's B public key Q_B and perform the following steps:

1. Choose a random: $r_1, r_2 \in \{0,1\}^k$.
2. Set: $h_1 = H_2(r_1, M)$.
3. Set: $h_2 = H_2(r_2, M)$.
4. Compute: $C_1 = g^{h_1} (mod\ n)$.
5. Compute: $C_2 = r_1 \oplus H_3((Q_{ID})^{h_1} (mod\ n))$.
6. Compute: $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1} (mod\ n))$.
7. Compute: $C_4 = M \oplus H_4(r_1)$.
8. Compute: $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2} (mod\ n))$.

User A outputs: $C_A = (C_{1A}, C_{2A}, C_{3A}, C_{4A}, C_{5A})$ and send it to user B through the hospital network manager ET. On the other side, user B (Physician) does the same previous steps as user A using params and user A public key Q_A to compute C_B and send it to user A.

– *Decrypt*: After user B received (C_A), then starts to decrypt the ciphertext using $params$ and his private key X_B, just as follows:

1. Compute: $r_1' = C_2 \oplus H_3(C_1^{XID})^e (mod\ n))$.
2. Compute: $M' = C_4 \oplus H_4(r_1')$.
3. Compute: $h_1' = H_2(r_1', M')$.
4. Compute: $h_1' r_2' = C_3 \oplus H_3(C_1^{XID})^e (mod\ n))$.
5. Compute: $h_2' = H_2(r_2', M')$.
6. Check: $C_1 = g^{h_1'} (mod\ n)$ holds.
7. Check: $C_5 = h_1' M' \oplus H_3((Q_{ID})^{h_2'} (mod\ n))$ holds. If yes, output M as the plaintext. Otherwise, abort C.

– *Trapdoor*: Given (C_A, ID_A) and (C_B, ID_B) compute the trapdoors for users A and B respectively as follows: $T_{ID} = g^{h_2 XID}$.

– *Test*: This algorithm is run by the entity ET (Equality Tester) and by given (C_4, T_{ID}) as input as follows:

Compute $E = C_5 \oplus H_3((T_{ID})^e (mod\ n))$. Then ET checks if:

$(C_{1A})^{EB} = (C_{1B})^{EA} (mod\ n)$ holds.

Correctness:

– *Encrypt/Decrypt*:
$r' = C_2 \oplus H_3(C_1{}^{XID})^e (mod\ n))$
$= r \oplus H_3((Q_{ID})^{h_1} (mod\ n)) \oplus H_3(g^{h_1 e d log_g Q_{ID}} (mod\ n))$
$= r \oplus H_3((Q_{ID})^{h_1} (mod\ n)) \oplus H_3(g^{h_1 log_g Q_{ID}} (mod\ n))$
$= r \oplus H_3((Q_{ID})^{h_1} (mod\ n)) \oplus H_3((Q_{ID})^{h_1} (mod\ n))$
$r' = r$.

– *Test*:
$C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2} (mod\ n))$
$E = C_4 \oplus H_3((T_{ID})^e (mod\ n))$
$= h_1 M \oplus H_3((Q_{ID})^{h_2} (mod\ n)) \oplus H_3(g^{h_2 e d log_g Q_{ID}} (mod\ n))$
$= h_1 M \oplus H_3((Q_{ID})^{h_2} (mod\ n)) \oplus H_3(g^{h_2 log_g Q_{ID}} (mod\ n))$
$= h_1 M \oplus H_3((Q_{ID})^{h_2} (mod\ n)) \oplus H_3((Q_{ID})^{h_2} (mod\ n))$
$E = h_1 M$.

4.2 PubKey: scheme description

The public key encryption scheme (Pubkey) consists of the following four algorithms:

– *Setup*: This algorithm is run by PKG as follows:

1. The PKG generates RSA parameters (n, e, d), let $n = p_1 p_2 p_3 p_4$; where $| p_1 | \approx | p_2 | \approx | p_3 | \approx | p_4 | \approx k/4$, and $ed \equiv 1 (mod\ \Phi(n))$; and d is kept secret as master key.
2. Choose an element $g \in Z_n^*$.
3. Choose cryptographic hash functions, and then the message space: $M \in \{0,1\}^k$, $H_1 : \{0,1\}^k \longrightarrow Z_n^*$, $H_2 : \{0,1\}^k \times \{0,1\}^k \longrightarrow Z_n^*$, $H_3 : Z_n^* \longrightarrow \{0,1\}^k$, $H_4 : \{0,1\}^k \longrightarrow \{0,1\}^k$. The system parameters: $params = (g, e, n, H_1, H_2, H_3, H_4)$.

– *KeyGen*: This algorithm is run by PKG as follows:

1. Set the Public key: $Q_{ID} = H_1(ID)$.
2. Set the private key: $XID = log_g Q_{ID}^d (mod\ \Phi(n))$.

Note: The private key can be computed by using the Chinese Reminder Theory (CRT).

– *Encrypt*: To encrypt a message $M \in \{0,1\}^k$ between user A (patient) and user B (physician), through the hospital network manager ET, and by using their identities $ID_{A/B} \in \{0,1\}^*$ and their public/private keys, user A will start to encrypt M to user B using params and user's B public key Q_B and perform the following steps:

1. Choose a random: $r_1, r_2 \in \{0,1\}^k$.
2. Set: $h_1 = H_2(r_1, M)$.
3. Set: $h_2 = H_2(r_2, M)$.

4. Compute: $C_1 = g^{h_1}(mod\ n)$.
5. Compute: $C_2 = r_1 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$.
6. Compute: $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$.
7. Compute: $C_4 = M \oplus H_4(r_1)$.
8. Compute: $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$.

User A outputs: $C_A = (C_{1A}, C_{2A}, C_{3A}, C_{4A}, C_{5A})$ and send it to user B through the hospital network manager ET. On the other side, user B (Physician) does the same previous steps as user A using params and user A public key Q_A to compute C_B and send it to user A.

– *Decrypt*: After user B received (C_A), then starts to decrypt the ciphertext using $params$ and his private key X_B, just as follows:

1. Compute: $r_1' = C_2 \oplus H_3(C_1^{XID})^e(mod\ n))$.
2. Compute: $M' = C_4 \oplus H_4(r_1')$.
3. Compute: $h_1' = H_2(r_1', M')$.
4. Compute: $h_1' r_2' = C_3 \oplus H_3(C_1^{XID})^e(mod\ n))$.
5. Compute: $h_2' = H_2(r_2', M')$.
6. Check: $C_1 = g^{h_1'}(mod\ n)$ holds.
7. Check: $C_5 = h_1' M' \oplus H_3((Q_{ID})^{h_2'}(mod\ n))$ holds.

5 Performance evaluation of our proposed IBEET-RSA

5.1 IBEET-RSA: security proof

To determine the security proof of our proposed IBEET-RSA scheme under OW-ID-CCA security model in the random oracle model, we relied on the use of methods similar to the ones defined in [9–11]. This was based on the IBEET-RSA security models explained in Section 3.2 against OW-ID-CCA and the Public Key Encryption scheme against OW-CCA. We then proved that the Public Key Encryption scheme is secure against OW-CCA. Nevertheless, we found that the OW-ID-CCA security can still be adapted to the OW-CCA security for Public Key Encryption scheme [11].

Theorem 1 If an OW-CCA adversary A could break the Pubic Key Encryption scheme with a non-negligible advantage ϵ in polynomial time t, then, after making hash queries at most $q_{H1} > 0$, $q_{H2} > 0$, $q_{H3} > 0$, $q_{H4} > 0$, decryption queries $q_d > 0$, private decryption key queries $q_X > 0$, and trapdoor queries $q_T > 0$, there would be algorithm able to solve the DLP problem with a non-negligible advantage of at least.

$$[\epsilon(k)/(q_{H3} + q_{H4} + q_d)] - [(q_d \cdot q_{H3})/(2^k \cdot (q_{H3} + q_{H4} + q_d))] + [(1/2^{2k} + 1/2^k + 1/2^k) \cdot ((q_{H3} + q_{H4}) \cdot q_d)/(q_{H3} + q_{H4} + q_d)] \quad (1)$$

Proof Suppose that A is an OW-CCA adversary. The advantage of $\mathcal{A}$ in the OW-CCA security model (Adv^{OW-CCA}) for Public Key Encryption scheme is $\mathcal{A}$ $(q_{H3} + q_{H4} + q_d)$. We present proof for this using games against $\mathcal{A}$. S_i denotes the event that $M' = M$ in Game $i\,(i = 0 \sim 4)$. These games are structured in the following manner: □

Game 0

1. Initial phase: The challenger $\mathbb{Ch}$ runs the *Setup* algorithm to generate RSA parameters (n, e, d) and to generate one random element $g \in Z_n{}^*$, then publishes $params : g, e, n, H_1, H_2, H_3, H_4$, while d is kept secret as master key.
2. Query phase: state $\longleftarrow$ $A^{H_3, H_4, Decryption}(params, Q_i)$. For $\forall(M\{0,1\}^k, r_1, r_2 \in \{0,1\}^k)$, the hash functions $H_3(\cdot)$, $H_4(\cdot)$ and the Decryption algorithm are simulated by the following oracles.
   – $H_3(T_3)$-Query: Given $T_3 \in \{0,1\}^k$, this oracle picks a random point $H_3 \in \{0,1\}^k$ and returns H_3.
   – $H_4(T_4)$-Query: Given $T_4 \in \{0,1\}^k$, this oracle picks a random number $H_4 \in \{0,1\}^k$ as $H_4(T_4)$ and returns it to $\mathcal{A}$.
   – Decryption (C)-Query: Given a ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$, this oracle computes $r_1{}' = C_2 \oplus H_3(C_1^{XID})^e(mod\ n))$, $h_1{}' r_2{}' = C_3 \oplus H_3(C_1^{XID})^e(mod\ n))$ and $M' = C_3 \oplus H_4(r_1{}')$, then compute $h_1{}' = H_2(r_1{}', M')$, $h_2{}' = H_2(r_2{}', M')$ holds, and verifies $C_1 = g^{h_1{}'}(mod\ n)$, and $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$ holds. If the equations hold, then the oracle returns $M' = M$. Otherwise, it returns $\perp$.
3. Computation phase: $C = (C_1, C_2, C_3, C_4, C_5)$ where $C_1 = g^{h_1}(mod\ n)$, $C_2 = r \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_4 = M \oplus H_4(r_1)$, $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$.
4. Output phase: Output $M' \longleftarrow A^{H_3, H_4, Decryption}(C,$ state).

The advantage of $\mathcal{A}$ winning the above game is as follows:

$$Adv(A)^{OW-CCA}(q_{H_3}, q_{H_4}, q_d) = Pr[\mathbf{S0}] \quad (2)$$

Game 1

1. Initial phase: The challenger $\mathbb{Ch}$ runs the *Setup* algorithm to generate RSA parameters (n, e, d) and to generate one random element $g \in Z_n{}^*$, then publishes

$params : g, e, n, H_1, H_2, H_3, H_4$, while d is kept secret as master key.

2. Query phase: state $\longleftarrow A^{H_3,H_4,Decryption}(params, Q_i)$. For $\forall(M\{0,1\}^k, r_1, r_2 \in \{0,1\}^k)$, the hash functions $H_3(\cdot)$, $H_4(\cdot)$ and the Decryption algorithm are simulated by the following oracles.
   - $H_3(T_3)$-Query: Given $T_3 \in \{0,1\}^k$, if $T_3 = {Q_{ID}}^{h_1}$, then $W_1 = H_3({Q_{ID}}^{h_1})$. Otherwise, this oracle picks a random point $H_3 \in \{0,1\}^k$ as $H_3(T_3)$. Then the oracle returns H_3 to $\mathcal{A}$.
   - $H_4(T_4)$-Query: Given $T_4 \in \{0,1\}^k$, this oracle picks a random number $H_4 \in \{0,1\}^k$ as $H_4(T_4)$ and returns it to $\mathcal{A}$.
   - Decryption (C)-Query: Given a ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$, this oracle computes ${r_1}' = C_2 \oplus H_3(C_1^{XID})^e (mod\ n))$, ${h_1}'{r_2}' = C_3 \oplus H_3(C_1^{XID})^e (mod\ n))$ and $M' = C_3 \oplus H_4({r_1}')$, then compute ${h_1}' = H_2({r_1}', M'), {h_2}' = H_2({r_2}', M')$ holds, and verifies $C_1 = g^{{h_1}'}(mod\ n)$, and $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$ holds. If the equations hold, then the oracle returns $M' = M$. Otherwise, it returns $\perp$.
3. Computation phase: randomly choose $W_1 \in \{0,1\}^K$ and compute $C = (C_1, C_2, C_3, C_4, C_5)$ where $C_1 = g^{h_1}(mod\ n)$, $C_2 = r \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_4 = M \oplus H_4(r_1)$, $C_5 = h_1 M \oplus W_1$.
4. Update phase: Add the tuple $({Q_{ID}}^{h_1}, W_1)$ to a list maintained by $\mathcal{GK}$.
5. Output phase: Output $M' \longleftarrow A^{H_3,H_4,Decryption}$ (C, state).

Compared to **Game 0**, we replace the value of H_4 function with W_1 in **Game 1**. The advantage of $\mathcal{A}$ winning **Game 1** is identical to that of **Game 0** as follows:

$$Adv(A)^{OW-CCA}(q_{H_3}, q_{H_4}, q_d) = Pr[\mathbf{S1}] = Pr[\mathbf{S0}] \quad (3)$$

Game 2

1. Initial phase: The challenger $\mathcal{GK}$ runs the *Setup* algorithm to generate RSA parameters (n, e, d) and to generate one random element $g \in {Z_n}^*$, then publishes $params : g, e, n, H_1, H_2, H_3, H_4$, while d is kept secret as master key.
2. Query phase: state $\longleftarrow A^{H_3,H_4,Decryption}(params, Q_i)$. For $\forall(M\{0,1\}^k, r_1, r_2 \in \{0,1\}^k)$, the hash functions $H_3(\cdot)$, $H_4(\cdot)$ and the Decryption algorithm are simulated by the following oracles.
   - $H_3(T_3)$-Query: Given $T_3 \in \{0,1\}^k$, if $T_3 = QID^{h_1}$, then the oracle returns $\perp$ and abort the game. This event denoted by **E1**. Otherwise, this oracle picks a random point $H_3 \in \{0,1\}^k$ as $H_3(T_3)$. Then the oracle returns H_3 to $\mathcal{A}$.
   - $H_4(T_4)$-Query: Given $T_4 \in \{0,1\}^k$, this oracle picks a random number $H_4 \in \{0,1\}^k$ as $H_4(T_4)$ and returns it to $\mathcal{A}$.
   - Decryption(C)-Query: Given a ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$, if C is identical to the challenge ciphertext except C_4, the oracle returns $\perp$. Otherwise, it computes $M = C_4 \oplus H_4(r_1)$, $h_1 = H_2(r_1, M)$, ${h_2}' = H_2({r_2}', M')$ then check if $C_1 = g^{h_1}(mod\ n)$ holds, and $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$ holds. If yes, then the oracle returns M; otherwise, it returns $\perp$.
3. Computation phase: Randomly choose $W_2 \in \{0,1\}^K$ and compute $C = (C_1, C_2, C_3, C_4, C_5)$ where $C_1 = g^{h_1}(mod\ n)$, $C_2 = r \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_4 = M \oplus H_4(r_1)$, $C_5 = W_2$.
4. Update phase: Add the tuple $({Q_{ID}}^{h_2}, h_1 M \oplus W_2)$ to a list maintained by $\mathcal{GK}$.
5. Output phase: Output $M' \longleftarrow A^{H_3,H_4,Decryption}$ (C,state).

Compared to **Game 1**, we replace the value of C_5 with a random W_2 in **Game 2**. Clearly, C_5 in both **Game 1** and **Game 2** are randomly chosen. Therefore, **Game 2** is identical to **Game 1** in the random oracle if the event **E1** does not happen. Thus, we obtain

$$\mid Pr[\mathbf{S2}] - Pr[\mathbf{S1}] \mid \leq Pr[\mathbf{E1}] \quad (4)$$

$$Pr[\mathbf{E1}] \leq q_d \cdot Adv^{DLP} + (q_d + q_{H_3})/2^K \quad (5)$$

Lemma 1 If the event **E1** in **Game 2** occurs with a non-negligible probability ϵ_1, then there is a probabilistic polynomial-time algorithm which can solve the DLP problem with a non-negligible probability.

Proof We adopted the same idea in [11] by constructing a simulator S_1 to solve the DLP problem. However, S_1 is able to solve the DLP problem if the following conditions hold.

1. A does not make $H_3(T_3)$ query on input $({Q_{ID}}^{h_2}(mod\ n))$ before a decryption query on input $C = (C_1, C_2, C_3, C_4, C_5)$. In this case, S_1 returns $\perp$. If the ciphertext C is valid, and that means the adversary $\mathcal{A}$ guess the value of H_3 correctly. This event **E2** happens with the probability $(1/2^K)$.
2. The event **E1** happens during the q_{H_3} times $H_3(T_3)$ queries. It means that the $\mathcal{GK}$ list contains the tuple $({Q_{ID}}^{h_2}(mod\ n), *)$. In this case S_1 retrieve and output h from $({Q_{ID}}^{h_1}(mod\ n))$ as a solution of the DLP problem. This event **E3** happens with the probability

$(Pr[\mathbf{E1}]/q_{H_3})$. For both cases, the advantage of $\mathcal{A}$ is as described below: □

$$Pr[\mathbf{E2}] \leq q_d/2^k \tag{6}$$

$$Pr[\mathbf{E3} \mid -\mathbf{E2}] = Pr[\mathbf{E1}]/q_{H_3} \tag{7}$$

$$\begin{aligned} Pr[\mathbf{E3}] &= Pr[\mathbf{E3} \mid \mathbf{E2}]Pr[\mathbf{E2}] + Pr[\mathbf{E3} \mid -\mathbf{E2}]Pr[-\mathbf{E2}] \\ &\geq Pr[\mathbf{E3} \mid -\mathbf{E2}]Pr[-\mathbf{E2}] \\ &= Pr[\mathbf{E3} \mid -\mathbf{E2}](1 - Pr[\mathbf{E2}]) \\ &= Pr[\mathbf{E3} \mid -\mathbf{E2}] - Pr[\mathbf{E3} \mid -\mathbf{E2}]Pr[\mathbf{E2}] \\ &\geq Pr[\mathbf{E3} \mid -\mathbf{E2}] - Pr[\mathbf{E2}] \\ &= (Pr[\mathbf{E1}]/q_{H_3}) - (q_d/2^k) \end{aligned} \tag{8}$$

Then we can get:

$$Adv_{(S1)}{}^{DLP} \geq (\epsilon_1/q_{H_4}) - (q_d/2^k) \tag{9}$$

From the above equations, ϵ_1 is non-negligible. Thus, $Adv_{(S1)}{}^{DLP}$ is non-negligible. In this case, we can sustain that the simulator S_1 can solve the DLP problem with a non-negligible probability. This contradicts the assumption that the DLP problem is hard.

Game 3

1. Initial phase: The challenger $\mathcal{CH}$ runs the *Setup* algorithm to generate RSA parameters (n, e, d) and to generate one random element $g \in Z_n{}^*$, then publishes $params : g, e, n, H_1, H_2, H_3, H_4$, while d is kept secret as master key.
2. Query phase: state $\longleftarrow$ $A^{H_3,H_4,Decryption}$ $(params, Q_i)$. For $\forall(M\{0,1\}^k, r_1, r_2 \in \{0,1\}^k)$, the hash functions $H_3(\cdot)$, $H_4(\cdot)$ and the Decryption algorithm are simulated by the following oracles.
   - $H_3(T_3)$-Query: Given $T_3 \in \{0,1\}^k$, if $T_3 = {Q_{ID}}^{h_1}$, then the oracle returns $\perp$ and abort the game. This event denoted by **E1**. Otherwise, this oracle picks a random point $H_3 \in \{0,1\}^k$ as $H_3(T_3)$. Then the oracle returns H_3 to $\mathcal{A}$.
   - $H_4(T_4)$-Query: Given $T_4 \in \{0,1\}^k$, this oracle picks a random number $H_4 \in \{0,1\}^k$ as $H_4(T_4)$ and returns it to $\mathcal{A}$.
   - Decryption(C)-Query: Given a ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$, if C is identical to the challenge ciphertext except C_4, the oracle returns $\perp$. Otherwise, it computes $M = C_3 \oplus H_4(r_1)$, $h_1 = H_2(r_1, M)$, ${h_2}' = H_2({r_2}', M')$ then check if $C_1 = g^{h_1}(mod\ n)$ holds, and $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$ holds. If yes, then the oracle returns M; otherwise, it returns $\perp$.
3. Computation phase: Randomly choose $W_3, W_4 \in \{0,1\}^K$ and $C = (C_1, C_2, C_3, C_4, C_5)$ where $C_1 = g^{h_1}(mod\ n)$, $C_2 = r \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_4 = M \oplus W_3$, $C_5 = W_4$.
4. Update phase: Add the tuple $({Q_{ID}}^{h_2}, h_1 M \oplus W_4)$ to a list of H_3 maintained by $\mathcal{CH}$, and the tuple (r_1, W_3) to a list of H_4 maintained by $\mathcal{CH}$.
5. Output phase: Output $M' \longleftarrow A^{H_3,H_4,Decryption}$ (C,state).

Compared to **Game 2**, we replace the hash function $H_4(r)$ with a random W_3 in **Game 3**. Clearly, C_5 in both **Game 2** and **Game 3** are randomly chosen. Therefore, **Game 3** is identical to **Game 2** in the random oracle.

$$Pr[\mathbf{S3}] = Pr[\mathbf{S2}] \tag{10}$$

Game 4

1. Initial phase: The challenger $\mathcal{A}$ runs the *Setup* algorithm to generate RSA parameters (n, e, d) and to generate one random element $g \in Z_n{}^*$, then publishes $params : g, e, n, H_1, H_2, H_3, H_4$, while d is kept secret as master key.
2. Query phase: state $\longleftarrow$ $A^{H_3,H_4,Decryption}$ $(params, Q_i)$. For $\forall(M\{0,1\}^k, r_1, r_2 \in \{0,1\}^k)$, the hash functions $H_3(\cdot)$, $H_4(\cdot)$ and the Decryption algorithm are simulated by the following oracles.
   - $H_3(T_3)$-Query: Given $T_3 \in \{0,1\}^k$, if $T_3 = {Q_{ID}}^{h_1}$, then the oracle returns $\perp$ and abort the game. This event denoted by **E1**. Otherwise, this oracle picks a random point $H_3 \in \{0,1\}^k$ as $H_3(T_3)$. Then the oracle returns H_3 to $\mathcal{CH}$.
   - $H_4(T_4)$-Query: Given $T_4 \in \{0,1\}^k$, $T_4 = r_1$, then the oracle returns $\perp$ and abort the game. This event denoted by **E4**. Otherwise, this oracle picks a random point $H_4 \in \{0,1\}^k$ as $H_4(T_4)$. Then the oracle returns H_4 to $\mathcal{CH}$.
   - Decryption(C)-Query: Given a ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$, if C is identical to the challenge ciphertext except C_4, the oracle returns $\perp$. Otherwise, it computes $M = C_3 \oplus H_4(r_1)$, $h_1 = H_2(r_1, M)$, ${h_2}' = H_2({r_2}', M')$ then check if $C_1 = g^{h_1}(mod\ n)$ holds, and $C_5 = h_1 M \oplus H_3((Q_{ID})^{h_2}(mod\ n))$ holds. If yes, then the oracle returns M; otherwise, it returns $\perp$.
3. Computation phase: Randomly choose $W_5, W_6 \in \{0,1\}^K$ and $C = (C_1, C_2, C_3, C_4, C_5)$ where $C_1 = g^{h_1}(mod\ n)$, $C_2 = r \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_3 = h_1 r_2 \oplus H_3((Q_{ID})^{h_1}(mod\ n))$, $C_4 = W_5$, $C_5 = W_6$.
4. Update phase: Add the tuple $({Q_{ID}}^{h_2}, h_1 M \oplus W_6)$ to a list of H_3 maintained by $\mathcal{CH}$, and the tuple $(r_1, W_5 \oplus M)$ to a list of H_4 maintained by $\mathcal{CH}$.
5. Output phase: Output $M' \longleftarrow A^{H_3,H_4,Decryption}$ (C,state).

Compared to **Game 3**, we replace the hash function C_5 with a random W_6 in **Game 4**. Clearly, C_4 in both **Game 3** and **Game 4** are randomly chosen. Therefore, **Game 4** is identical to **Game 3** in the random oracle if the event **E2** does not happen:

$$| Pr[S_4] - Pr[S_3] | \leq Pr[\mathbf{E4}] \leq Pr[\mathbf{E1} \vee \mathbf{E4}] \tag{11}$$

$$\begin{aligned} Pr[\mathbf{E1} \vee \mathbf{E4}] \leq\ & [(q_{H_3} + q_{H_4})] \cdot Adv^{DLP} + \\ & [(1/2^{2k} + 1/2^k + 1/2^k) \cdot (q_{H_3} + q_{H_4}) q_d] \end{aligned} \tag{12}$$

Lemma 2 If the event **E1** $\vee$ **E4** in **Game 4** occurs with a non-negligible probability ϵ_2, then there is a probabilistic polynomial-time algorithm which can solve the DLP problem with a non-negligible probability.

Proof We adopted the same idea in [11] by constructing a simulator S_1 to solve the DLP problem. However, S_2 is able to solve the DLP problem if the following conditions hold.

1. $\mathcal{A}$ does not make $H_3(T_3)$ query on input $(C_1^{XB}(mod\ n))$ or a $H_4(T_4)$ query on input (r) before a decryption query on input $C = (C_1, C_2, C_3, C_4, C_5)$. In this case, S_2 returns $\perp$. If the ciphertext C is valid, and that means the adversary A guess the value of H_3 or H_4 correctly. This event **E5** happens with the following three different possibilities.
   - $\mathcal{A}$ has never made a $H_3(T_3)$ query on input $((C_1^{XB})^e\ (mod\ n))$ and has made a $H_4(T_4)$ query on input (r). This happens with probability $(1/2^k)$.
   - $\mathcal{A}$ has made a $H_3(T_3)$ query on input $((C_1^{XB})^e\ (mod\ n))$ and has never made a $H_4(T_4)$ query on input (r). This happens with probability $(1/2^k)$.
   - $\mathcal{A}$ has never made a $H_3(T_3)$ query on input $((C_1^{XID})^e\ (mod\ n))$ and has never made a $H_4(T_4)$ query on input (r). This happens with probability $(1/2^{2k})$.
2. The event **E1** $\vee$ **E4** happens during the q_{H_3} times $H_3(T_3)$ queries and q_{H_4} times $H_4(T_4)$. It means that the $\mathbb{Gh}$ list contains the tuple $((C_1{}^{XID})^e\ (mod\ n), *)$ or the $\mathbb{Gh}$ list contains the tuple $(r_1, *)$. In this case S_2 retrieve and output h from $(Q_{ID}{}^{h_1}\ (mod\ n))$ as a solution of the DLP problem. This event **E6** happens with the probability (Pr [**E1** $\vee$ **E4**] / $(q_{H_3} + q_{H_4})$). For both cases the advantage of $\mathcal{A}$ as follows: □

$$Pr[\mathbf{E5}] \leq [(1/2^k) + (1/2^k) + (1/2^{2k})] q_d \tag{13}$$

$$Pr[\mathbf{E6} \mid -\mathbf{E5}] = Pr[\mathbf{E1} \vee \mathbf{E4}]/(q_{H_3} + q_{H_4}) \tag{14}$$

$$\begin{aligned} Pr[\mathbf{E6}] &= Pr[\mathbf{E6} \mid \mathbf{E5}] Pr[\mathbf{E5}] + Pr[\mathbf{E6} \mid -\mathbf{E5}] Pr[-\mathbf{E5}] \\ &\geq Pr[\mathbf{E6} \mid \mathbf{E5}] Pr[-\mathbf{E5}] \\ &= Pr[\mathbf{E6} \mid -\mathbf{E5}](1 - Pr[\mathbf{E5}]) \\ &= Pr[\mathbf{E6} \mid -\mathbf{E5}] - Pr[\mathbf{E6} \mid -\mathbf{E5}] Pr[\mathbf{E5}] \\ &\geq Pr[\mathbf{E6} \mid -\mathbf{E5}] - Pr[\mathbf{E5}] \\ &= Pr[\mathbf{E1} \vee \mathbf{E4}]/(q_{H_3} + q_{H_4}) \\ &\quad - [(1/2^k) + (1/2^k) + (1/2^{2k})] q_d \end{aligned} \tag{15}$$

Then we can get:

$$\begin{aligned} Adv_{(S_2)}{}^{DLP} \geq\ & \epsilon_2/(q_{H_3} + q_{H_4}) - [(1/2^k) + (1/2^k) \\ & + (1/2^{2k})] q_d \end{aligned} \tag{16}$$

According to the assumption, ϵ_2 is non-negligible. Thus, $Adv_{(S_2)}{}^{DLP}$ is non-negligible. Therefore, the simulator S_2 can solve the DLP problem with a non-negligible probability, which contradicts with the hardness of DLP problem.

Theorem 2 For an OW-ID-CCA adversary with advantage ϵ_3 against IBEET-RSA. Let q_X, q_T, q_d denote the times of private decryption key queries, trapdoor queries, and decryption queries, respectively. There is an OW-CCA adversary with advantage of at least: $\epsilon_3/e(q_X + q_T + q_d + 1)$ against Public Key Encryption scheme.

Proof We can convert an OW-ID-CCA attack against IBEET-RSA into an OW-CCA attack against Public Key Encryption scheme by adhering the same notion in [11]. □

Theorem 3 As for the Theorems mentioned above, we concluded that the proposed IBEET-RSA scheme is OW-ID-CCA secure, taking into consideration the difficulty of DLP problem in the random oracle.

5.2 IBEET-RSA: computational efficiency

In this section, we evaluate the performance of our proposed IBEET-RSA scheme as well as the performance of the schemes mentioned in [9–11], mainly in terms of computational costs. To get the implementation time of the basic operations through this comparison, we adopted MIRACL library in [33], and the existing experimental in [34–36], adhering to the following protocol: Ubuntu 14.04 (64 bit); CPU: 2.30 GHz; and RAM: 1 GB. The running time for each operation is defined below:

T_1: Point multiplication operation = 1.97 ms)
T_2: Exponentiation operation = 0.331 ms
T_3: Bilinear pairing operation = 5.275 ms

Table 1 Comparison of computational cost efficiency (ms)

Scheme	[9]	[10]	[11]	Our
Encryption	$6T_2+2T_3+T_1+2T_4+T_5 = 24.717$	$6T_2+2T_5+2T_7 = 2.015$	$2T_2+5T_1+2T_6+T_4+T_5 = 15.646$	$4T_2+6T_5 = 1.378$
Decryption	$2T_2+2T_3+T_4+T_5 = 16.332$	$3T_2+2T_5+2T_7 = 1.022$	$2T_3+2T_1+T_6+T_4+T_5 = 19.612$	$4T_2+6T_5 = 1.378$
Test	$4T_3+2T_1+2T_4 = 35.242$	$2T_2+2T_7 = 0.673$	$2T_3+2T_2+2T - 4 = 21.414$	$2T_2+T_5 = 0.671$
Total	76.291	3.710	56.490	3.427

T_4: HashToPoint function = 5.101 ms
T_5: General hash function = 0.009 ms
T_6: Point addition operation = 0.012 ms
T_7: Symmetric cryptography = 0.00541 ms
Other lightweight operations (e.g. XOR) = 0.008s (Omitted).

In the field of WBAN, including mobile communications, Wi-Fi, Sensors or any other communication methods. The power consumptions, computational cost, and communications overhead cause serious problems due to the restrictions of lightweight communication systems performance, especially with respect to PKC cryptosystem [36]. Many other approaches [37–40] have proposed techniques that require the use of PKC to improve the security level in such lightweight communication systems. The proposed schemes in [9–11] are identity-based cryptosystems used 160-bit length for elliptic curve, while our proposed scheme adopted a new approach by using identity-based over RSA and provides 1024-bit length to guarantee same level of security. As we can clearly observe in both Table 1 and Fig. 3, the computational cost of our scheme decreases compared to the computational costs in schemes [9–11]. As a result, our scheme can achieve a better computational performance than the previously mentioned proposed schemes and, more importantly, our scheme can be compatible with WBAN applications.

6 Conclusion

WBAN applications provide many benefits and challenges to the medical and the healthcare sectors. Among the most important benefits, we can mention a suitable environment that can monitor the daily lives and medical conditions of patients at anytime, anywhere, and without limitations. However, when it comes to challenges, one of the most significant ones are precisely in connection with poor levels of security in all WBAN applications. This not only makes the patient's privacy weak and vulnerable, but also threatens the patient's life by exposing his or her data to potential adversaries. To recap, in this paper, a new IBEET-RSA scheme without pairing was presented. The performance evaluation was given to prove that the proposed scheme is efficient and secure against one-way secure and against chosen-identity and chosen-ciphertext attacks (OW-ID-CCA) under the DDH assumption.

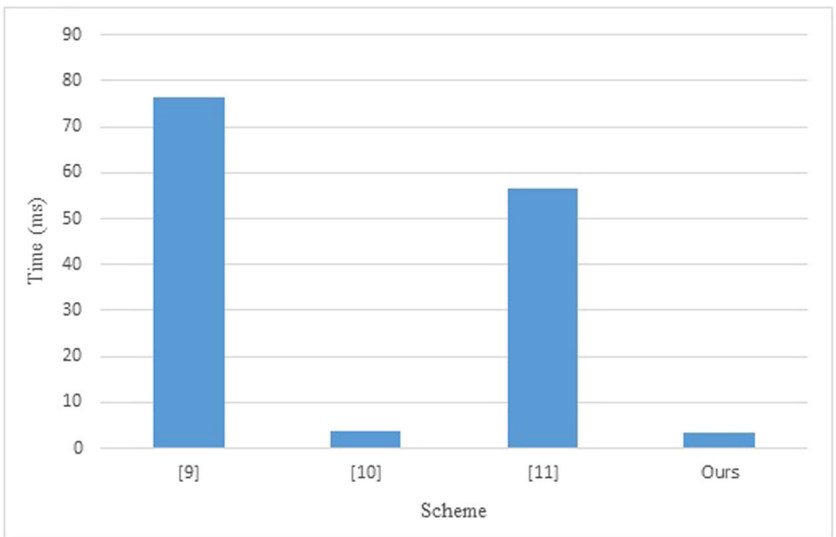

Fig. 3 Comparison of computational cost efficiency (ms)

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant 6147206 6, Sichuan Science and Technology Program (No. 2018GZ0180, 2018GZ0085, 2017GZDZX0001, 2017GZDZX0002).

References

1. Chen M, Gonzalez S, Vasilakos A, Cao H, Leung VC (2011) Body area networks: a survey. Mobile Netw Appl 16:171–93
2. Latr B, Braem B, Moerman I, Blondia C, Demeester P (2011) A survey on wireless body area networks. Wireless Netw 17(1):1–8
3. Alemdar H, Ersoy C (2010) Wireless sensor networks for healthcare: a survey. Comput Netw 54(15):2688–2710
4. Latr B, Braem B, Moerman I, Blondia C, Demeester P (2011) A survey on wireless body area networks. Wireless Netw 17(1):1–8
5. Al Ameen M, Liu J, Kwak K (2012) Security and privacy issues in wireless sensor networks for healthcare applications. J Med Syst 36(1):93–101
6. Javadi SS, Razzaque MA (2013) Security and privacy in wireless body area networks for health care applications. In: Wireless networks and security. Springer, Berlin, pp 165–87
7. Zhao Z (2014) An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem. J Med Syst 38(2):1–7
8. Xiong H, Qin Z (2015) Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks. IEEE Trans Inf Forensics Secur 10:1442–55
9. Ma S (2016) Identity-based encryption with outsourced equality test in cloud computing. Inform Sci 328:389–402
10. Lee HT, Ling S, Seo JH, Wang H (2016) Semi-generic construction of public key encryption and identity-based encryption with equality test. Inform Sci 373:419–440
11. Wu L, Zhang Y, Choo K-KR, He D (2017) Efficient and secure identity-based encryption scheme with equality test in cloud computing. Futur Gener Comput Syst 73:22–31
12. Tang Q (2012) Public key encryption scheme supporting equality test with authorization of different granularity. Int J Appl Cryptogr 2(4):304–321
13. Yang G, Tan C, Huang Q, Wong DS (2010) Probabilistic public key encryption with equality test, The Cryptographers? Track at the RSA conference, (CT-RSA 2010) San Francisco, CA, USA, LNCS, 5985. Springer, Berlin
14. Tang Q (2011) Towards public key encryption scheme supporting equality test with finegrained authorization, 16th Australasian Conference on Information Security and Privacy (ACISP2011), Melbourne, Australia, LNCS, 6812. Springer, Berlin
15. Tang Q (2012) Public key encryption supporting plaintext equality test and user-specified authorization. Sec Commun Netw 5(12):1351–1362
16. Ma S, Huang Q, Zhang M, Yang B (2015) Efficient public key encryption with equality test supporting flexible authorization. IEEE Trans Inf Foren Sec 10(3):458–470
17. Huang K, Tso R, Chen Y-C, Rahman SMM, Almogren A, Alamri A (2015) PKE-AET: public key encryption with authorized equality test. Comput J. https://doi.org/10.1093/comjnl/bxv025
18. Ma S, Zhang M, Huang Q, Yang B (2015) Public key encryption with delegated equality test in a multi-user setting. Comput J 58(4):986–1002
19. Si H et al (2010) An improved RSA signature algorithm based on complex numeric operation function. International Conference on Challenges in Environmental Science and Computer Engineering 2:397–400
20. da Silva JCL (2010) Factoring semi primes and possible implications. In: IEEE in Israel, 26th Convention, pp 182–183
21. Aboud SJ (2009) An efficient method for attack RSA scheme. ICADIWT Second International Conference 4-6:587–591
22. Scripcariu L, Frunza MD (2005) A new character encryption algorithm. ICMCS 2005, pp 83–86
23. Pollard J (1978) Monte Carlo methods for index computation (mod p). Math Comp 32:918–924
24. Pollard J (1974) Theorems on factorization and primality testing. Proc Cambridge Philos Soc 76:521–528
25. Brent RP (1980) MR 82a:10007, Zbl 439.65001. rpb051, BIT, 20, 176–184
26. Bell E (1986) The prince of amateurs: fermat. New York: Simon and Schuster, pp 56–72
27. Tan CC, Wang H, Zhong S, Li Q (2009) IBE-lite: a lightweight identity-based cryptography for body sensor networks. IEEE Trans Inf Technol Biomed 13(6), art. No. 5272415: 926–932
28. Lin X-J et al (2017) An efficient RSA-based certificateless public key encryption scheme. Discret Appl Math. https://doi.org/10.1016/j.dam.2017.02.019
29. Boneh D, Franklin M (2003) Identity-based encryption from the weil pairing. SIAM J Comput 3(32):586–615
30. Yu Y, Xue L, Au MH, Susilo W, Ni J, Zhang Y, Vasilakos AV, Shen J (2016) Cloud data integrity checking with an identity-based auditing mechanism from RSA. Futur Gener Comput Syst 62:85–91
31. Wu L, Zhang Y, Choo K-K, He D (2017) Efficient and secure identity-based encryption scheme with equality test in cloud computing. Futur Gener Comput Syst 73:22–31
32. Boneh D, Goh E, Nissim K (2005) Evaluating 2-DNF formulas on ciphertexts. In: Proceedings of theory of cryptography (TCC)'05, LNCS 3378, pp 325–341
33. Scott M (2003) MIRACLE-Multiprecision integer and rational arithmetic C/C++ Library, Shamus Software Ltd, Dublin, Ireland, (http://www.shamus.ie)
34. He D, Chen J (2013) An efficient certificate-less designated verifier signature scheme. The International Arab Journal of Information Technology 10(4):389–396
35. Hea D, Chen J, Zhang R (2011) An efficient identity-based blind signature scheme without bilinear pairings. Comput Electr Eng 37(4):444–450
36. Ramadan M, Du G, Li F, Xu CX (2016) A survey of public key infrastructure-based security for mobile communication systems. Symmetry 8(9):85
37. Ramadan M, Du G, Li F, Xu CX (2016) EEE-GSM: end-to-end encryption scheme over GSM system. International Journal of Security and Its Applications 10(6):229–240. ISSN: 1738-9976 IJSIA, (https://doi.org/10.14257/ijsia.2016.10.6.22)
38. Ramadan M, Li F, Xu CX, Oteng K, Ibrahim H (2015) Authentication and key agreement scheme for CDMA cellular system. In: Proceedings of the 2015 IEEE international conference on communication software and networks (ICCSN), China, 6C7, p 118C124
39. Ramadan M, Li F, Xu CX, Abdalla A, Abdalla H (2016) An efficient end-to-end mutual authentication scheme for 2G-GSM system. In: 2016 IEEE international conference on big data analysis (ICBDA 2016), Hangzhou, P.R.China, IEEE Xplore Digital Library, https://doi.org/10.1109/ICBDA.2016.7509848, pp 1–6
40. Ramadan M, Li F, Xu CX, Mohamed A, Abdalla H, Abdalla A (2016) User-to-user mutual authentication and key agreement scheme for LTE cellular system. Int J Net Secur 18(4):769–781