Equality test with an anonymous authorization in cloud computing

Hisham Abdalla[1,3] · Hu Xiong[1] · Abubaker Wahaballa[1,2] · Mohammed Ramadan[1] · Zhiguang Qin[1]

Abstract
With the rapid popularity and wide adoption of cloud storage, providing privacy-preserving by protecting sensitive information becomes a matter of grave concern. The most effective and sensible way to address this issue is to encrypt the data before uploading it to the cloud. However, to search over encrypted data with different keys is still an open problem when it comes to the deployment of emerging technologies such as healthcare applications and e-marketplace systems. To address these issues, in this paper, we proposed a secure and efficient public-key encryption with an equality test technique that supports anonymous authorization, abbreviated as (PKEET-AA). Our proposed scheme allows a specific user to identify who can perform the equality test process among various cloud servers without compromising sensitive information. It also provides an anonymous approach to search for some statistical information about specific identical encrypted records in several databases. Moreover, we prove that our proposed PKEET-AA scheme is one-way secure against chosen-ciphertext attack (OW-CCA) and undistinguishable against adaptive chosen ciphertext attack (IND-CCA) in the random oracle model. Thus, to provide authorization/multi-authorization anonymity under the Decisional Diffie Hellman assumption.

Keywords Public key encryption · Equality test · Anonymous authorization · Multi-authorization · Cloud security ·

1 Introduction

The availability of high-speed Internet and cloud services has fueled a trend towards outsourcing data and its management to the cloud service providers (CSP). It brings a flexible, cost-effective, and reliable way for data owners to deal with their data storage. Users can concentrate on their core operations by storing data to the cloud, such as searching the keyword, pattern matching, and other extended computations rather than incurring substantial hardware and personnel costs. However, cloud service providers are semi-trusted. In this sense, they cannot be fully trusted to ensure the availability, confidentiality, or integrity of user data outsourced to the cloud (e.g., a CSP may be corrupted and cloud employees may be malicious or "curious") [1]. Therefore, for the sake of confidential data, a common practice is to encrypt outsourced data in advance.

To perform a searching operation on ciphertext, public-key encryption with keyword search (PKEKS) is one popular research focus [2], in which an encrypted keyword can be searched by an untrusted cloud server but cannot be unknown and decrypted. Later, the concept of public-key encryption scheme with the equality test (PKEET) was first proposed by Yang et al. [3]. This notion enables two ciphertexts under different public keys, as well as the same public key, to check whether they contain the same data. The PKEET immediately drawn the attention of cloud service creators and was adopted to develop several cloud computing amenities, including, but not limited to, facilitating keyword searches on encrypted data, partitioning

✉ Mohammed Ramadan
mramadan8@hotmail.com

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] Arab East Colleges for Graduate Studies, Riyadh, Saudi Arabia

[3] Karary university, Omdurman, Sudan

encrypted emails and managing personal health records, among others.

To tackle the privacy concerns, we expand on the PKEET scheme by introducing an authorization mechanism that enables users to specify who can perform the test process from their ciphertexts in a multi-server scenario. The new primitive denoted as PKEET *with anonymous authorization* (PKEET-AA). The PKEET-AA is different from the notion of anonymous trapdoor defined by Ma et al. [4], since the anonymous trapdoor achieves only the anonymity of the data owner.

Let us consider the following situation to elucidate the meaning and significance of our new approach to PKEET in a multi-server scenario. As to the meaning of PKEET-AA, we must clarify that the term "anonymous authorization" refers to the notion that a given server can verify whether or not it is an assigned server tester. All of this without necessarily being aware of the other assigned server testers. As to the significance, we consider the interactions in our Personal Health Records (PHRs) system, as shown in Fig. 1. Suppose that the health administration (HA) in any given country has (z) the number of hospitals distributed throughout different regions. The local database for each hospital maintains the confidentiality of the patients' records in their local storage, also called "server testers". The table for each hospital (e.g. $P\text{-}Hos_1$ and $P\text{-}Hos_z$ tables for (z) hospitals) contains the patient's number and encrypted disease, denoted by $P\text{-}N$ and $PKEE\text{-}AA$ (pk_{sender}, $pk_{reciver}$, *disease, area)*, respectively. For instance, given two patients, A and B with an identical disease belong to different areas 1 and area 2, respectively. A encrypts his privacy information (disease and area) under the public key pk_A and the doctor's public key pk_D, and transmits his encrypted patient's record PKEE-AA (pk_D , pk_A , disease, area 1) to HA. Then, A generates a warrant w_A and transmits to HA. Similarly, B transmits both the PKEE-AA (pk_D , pk_B , disease, area 2) and w_B to HA. After receiving these data, the HA could determine any server tester to search whether A and B have the identical diseases. However, there is no knowledge of what the real areas and disease. Then, the HA sends the search result to the patient A and B, respectively, which allows them to share their medical records. While for a multi-server scenario, using the PKEE-AA scheme for the department of internal medicine to officially declare a disease either as an outbreak or an epidemic. The health administration, which is the head of all hospital units, has to certify the incidence of more disease cases than expected in a given region or among a specific group of individuals over a particular period. This information usually extracted from a hospital's local storage. The HA first will take one encrypted patient's record from the department of internal medicine

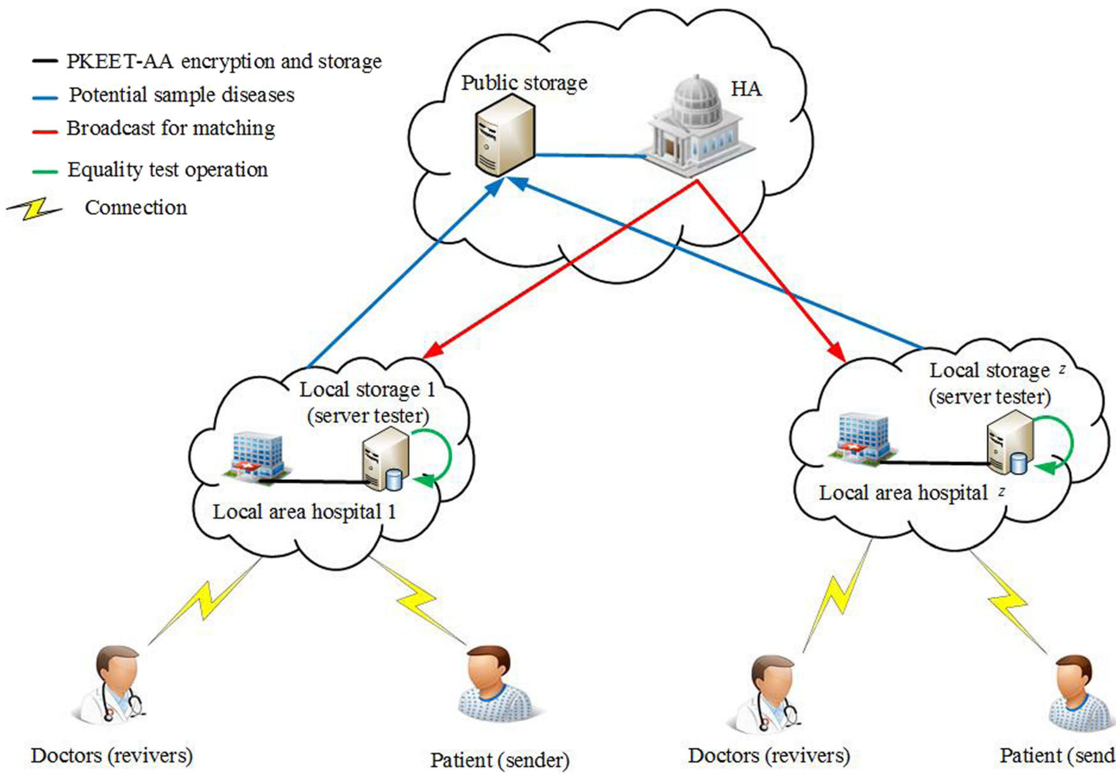

Fig. 1 System architecture

in the *P-Hos*$_1$ table as a sample for matching, represented by *PKEE-AA* (pk_{sender}, $pk_{reciver}$, *disease*, area)*. After that, broadcast this sample for all hospitals to find out all patients with identical diseases in the internal medicine department in (z) hospitals.

We remark that any database server will execute the operation after receiving the sample matching, along with its corresponding disease-warrant w_d. Given that the patients are only identified by their public keys pk_i and the database servers are identified by (pk_{si}), in a nutshell, a PKEE-AA scheme for such scenario will lead to some of the following features:

- *Anon-Auth feature*: It maintains patients' warrant privacy and implies that even the assigned server testers can not recognize the owner's sample matching public key from the warrant value.
- *Anon-MAuth feature*: It protects the receivers an assigned server testers privacy and implies that any database server can verify whether or not he/she is an assigned server tester, without knowing the other database server's public keys. This feature allows the health administration to hide her searching areas regarding the country policy, and also, simultaneously, ensures the protection of all the statistical information (the number of total patients) about the disease, until issuing the official declaration.
- *Verification feature*: it verifies warrants after receiving along with the owner of the warrant privacy-preserving.

Designing secure and anonymous authorization over cross-domain PHRs system that offers the features as mentioned earlier, is by no means an easy task. Therefore, our PKEET-AA provides a feasible solution for such a scenario, which will focus on this paper.

1.1 Our contributions

In order to strengthen the privacy-preserving in the scenario, as mentioned above, we introduce a new approach to the PKEET scheme in a multi-server scenario, referred to as PKEET with anonymous authorization (PKEET-AA). Our scheme allows finding out all the patients having a specific disease in a multi-server scenario. To confirm the new notion of PKEET-AA, we propose an efficient construction based on the construction outlined in [5].

Our main contributions can be summarized as follows.

- We introduce a new primitive of *anonymous authorization*, which allows to find out the statistical information about the specific identical disease among multiple assigned databases in an anonymous way.
- Since our goal is to inventory all patients having the specific identical disease in a multi-server database, a specific encrypted disease record C_i becomes equality testable for the assigned server testers.
- Finally, to protect against leakage and to tamper risks on transportation, the warrant can be verified after receiving along with the owner of the warrant privacy-preserving, which has not seen in previous works.

1.2 Organization

The rest of this paper is organized as follows: Related work and some preliminaries are presented in Sects. 2 and 3, respectively. In Sect. 4, we provide a framework and the definition of PKEET-AA scheme. We also give the security model for PKEET-AA in this section. In Sect. 5, we propose an PKEET-AA scheme. In Sect. 6, we give the security analysis. Finally, we conclude the paper in Sect. 7.

2 Related works

Public key encryption with an equality test (PKEET) was first introduced by Yang et al. in (CT-RSA 2010) [3]. Many improvements to PKEET schemes have carried out ever since. These improvements have dedicated to achieve better performance and to supplement additional functionalities. To illustrate, to achieve a fine-grained authorization mechanism, Tang [6] proposed an enhanced PKEET(FG-PKEET)where only the authorized two users could perform the test processes with the help of a trusted party. Tang [7] later expanded the notion of FG-PKEET to a two-proxy setting, where two proxies collaborated to perform equality test processes. To achieve a coarse-grained authorization, Tang [8] presented an all-or-nothing PKEET (AoN-PKEET), which specified who could perform an equality test process on ciphertexts. Furthermore, to allow only a delegated party to perform the equality test process, Ma et al. [4] proposed a notion of public-key encryption with delegated equality test (PKE-DET). Huang et al. [9] proposes public-key encryption with authorized equality test (PKE-AET), to only allow a tester to perform the equality test process on all of the ciphertexts or specific ciphertexts. Besides, to enhance the privacy of users' data, Ma et al. [10] as well proposes a notion of PKEET with flexible authorization (PKEET-FA), which supports four types of authorization policies at the same time. To verify the equality test results to check whether the cloud performed honestly for the PKEET-FA scheme, Xu et al. [5] proposed a verifiable PKEET scheme, called V-PKEET. Also, to avoid the complex key management related to PKEET [3], the identity-based encryption [11] with the equality test (IBE-ET) [12 16] was presented.

Moreover, a privacy-preserving homomorphism is also a practical and promising solution to do some essential operations on encrypted data without decryption [17]. Adopting somewhat homomorphic encryption secured in the semi-honest model, Tushar [18] proposed the private equality test (PET). It allows two users who want to compare their sensitive information for checking equality without revealing any information to each other if they do not equal.

As a summary of this section, all of the mentioned public-key encryption schemes with (authorized) equality test carried out mainly relying on the traditional single-server scenario, which are different from our problem scenario. As distributed networks are heavily used in modern applications [19, 20], the typical multi-server scenario for the equality test system shown in Fig. 1.

3 Preliminaries

Here, we briefly review the related security assumptions on which our scheme has relied.

A. ***COMPUTATIONAL DIFFIE–HELLMAN (CDH) ASSUMPTION***

Given (g, g^a, g^b) for some $a, b \in \mathbb{Z}_P$, where g is the generator of $\mathbb{G}$ with prime order p. It is intractable to calculate g^{ab}.

B. ***DECISIONAL DIFFIE–HELLMAN (DDH) ASSUMPTION*** Given (g^a, g^b, g^c) for some $a, b, c \in \mathbb{Z}_P$, where g is the generator of $\mathbb{G}$ with prime order p. It is intractable to distinguish (g^a, g^b, g^{ab}) from (g^a, g^b, g^c).

We say that the CDH and DDH assumptions are hard problems for every probabilistic polynomial-time algorithm, the probability of solving CDH and DDH problems is negligible.

4 PKEET *with anonymous authorization* (PKEET-AA)

In this section, we first define the framework and a formal definition for PKEET-AA scheme. Then provide its security model. Finally, we prove the formal security of PKEET-AA scheme.

4.1 Framework of PKEET-AA scheme

Generally speaking, up to now in both PKEET and PKE-AET, there is only one new entity, called tester, who is in charge of performing equality test operations on other receivers' ciphertexts. Instead of that, we present a new PKE-AET framework, based on multiple server instances with equality test, which allows a sender integrated with a third party (health administration (HA) in our scenario) to authorize a multi-warrant mw_i for z assigned server testers, as shown in Fig. 1. A PKEET-AA scheme provides identical warrant to multiple assigned server testers in an anonymous way, making it possible to determine which of the patients have certain diseases among multiple assigned server testers (multi storages). The PKEET-AA scheme consists of two entities. The first one is a cloud, and the second one is a set S of n users (sender (patient) or receiver (doctor) or assigned server testers). Without loss of generality, we assume that a sender will be more prone to authorizing a multi-warrant mw_i to z (with $z \leq n$) assigned server testers chosen from S, denoted by $(pk_{s1}, \cdots, pk_{sz})$.

The PKEET-AA entities can be described as:

1. **Cloud:** It is an entity that holds the third party called Health administration (HA) and the public storage server.
2. **Health administration (HA):** It is the head of all hospitals in the system hierarchy, and it is responsible for authorizing multiple assigned server testers.
3. **Users:** It is a general description of:
   (a) **Sender (patient):** He/she is a user of the system who can generate a ciphertext to the receiver and also issue a disease-warrant.
   (b) **Receiver (doctor):** He/she is a user of the system who can decrypt the received ciphertext using his/her private key. Note that only the receiver could decrypt the ciphertext encrypted by the sender.
   (c) **Server testers:** It is an additional entities in the system that selectively permitted to perform equality test operations on the sender's ciphertext with another ciphertext.

4.2 PKEET-AA: scheme description

4.3 Security model for PKEET-AA

In this section, our security model defined as follows:

For the ciphertext security of the PKEET-AA scheme, we define two types of adversaries whose main goal is to reveal information about the encrypted disease records in this fashion:

- ***PKEET-AA Type-1 adversary:***: Such an adversary owns both the assigned server tester's private key and the authorization information cannot decide whether the challenge ciphertext is the encryption of which disease record. To refer to this adversary, we relied on the notion of OW-CCA security.

- ***PKEET-AA Type-2 adversary:*** Such an adversary does not have access to the authorization information and cannot recover the plaintext from the challenge ciphertext. To refer to this adversary, we relied on the notion of IND-CCA security.

For the authorization security of the PKEET-AA scheme, the following security properties are critical:

- The PKEET-AA scheme must achieve the sender's privacy. This property addressed through the notion of anonymous authorization (Anon-Auth).
- The PKEET-AA scheme must allow the senders to hide their viewing selections among multiple assigned server testers. This property addressed through the notion of anonymous multi authorization (Anon-MAuth).

Definition 1 OW-CCA Secure Against PKEET-AA Type-1 adversary [5]. let $\prod$ = (Setup, Key-Gen, Encrypt, Decrypt, Multi-Authorization, Extract, Verification, Test)be a PKEET-AA scheme and $\mathscr{A}$ be a polynomial-time (PPT) adversary. A PKEET-AA scheme is OW-CCA secure if for all adversaries $\mathscr{A}$, $Adv^{OW\ \ CCA,Type\ \ 1}_{\prod,\mathscr{A}}(\lambda) :=| Pr(M' = M^*) |$ is negligible in the security parameter λ.

Definition 2 A PKEET-AA scheme is IND-CCA secure [5] if for all adversaries $\mathscr{A}$, $Adv^{IND\ \ CCA,Type\ \ 2}_{\prod,\mathscr{A}}(\lambda) :=| Pr(\rho' = \rho) - 1/2 |$ is negligible in the security parameter λ.

Definition 3 (Anon-Auth secure). Let $Adv^{Anon\ \ Auth}_{\prod,\mathscr{A}}(\lambda) \overset{def}{=}$

$$\Pr\left[\begin{array}{r} pp \leftarrow Setup(\lambda), (pk_{si}, sk_{si})^n_{i=1} \leftarrow Key\text{-}Gen(pp), \\ ((pk_{i0}, sk_{i0}), (pk_{i1}, sk_{i1})) \leftarrow \mathscr{A}(pk_{si}), \rho \leftarrow \{0,1\}, \\ \hat{Auth_d} \leftarrow Multi\text{-}Authorization(sk_{i\rho}, pk_{si}), \\ \rho' \leftarrow \mathscr{A}^{\mathcal{O}^{MA}}(\hat{Auth_d}) : \rho' = \rho \end{array}\right] - 1/2,$$

A PKEET-AA scheme is Anon−Auth secure if for all adversaries $\mathscr{A}$, $Adv^{Anon\ \ Auth}_{\prod,\mathscr{A}}(\lambda) :=| Pr(\rho' = \rho) |$ is negligible in the security parameter λ.

Definition 4 (Anon−MAuth secure). Let $Adv^{Anon\ \ MAuth}_{\prod,\mathscr{A}}(\lambda) \overset{def}{=}$

$$\Pr\left[\begin{array}{r} pp \leftarrow Setup(\lambda), (pk_i, sk_i)^n_{i=1} \leftarrow Key\text{-}Gen(pp), \\ ((pk_{s0}, sk_{s0}), (pk_{s1}, sk_{s1})) \leftarrow \mathscr{A}(pk_i), \rho \leftarrow \{0,1\}, \\ \hat{Auth_d} \leftarrow Multi\text{-}Authorization(sk_i, pk_{s\rho}), \\ \rho' \leftarrow \mathscr{A}^{\mathcal{O}^{MA}}(\hat{Auth_d}) : \rho' = \rho \end{array}\right] - 1/2,$$

A PKEET-AA scheme is Anon-MAuth secure if for all adversaries $\mathscr{A}$, $Adv^{Anon\ \ MAuth}_{\prod,\mathscr{A}}(\lambda) :=| Pr(\rho' = \rho) |$ is negligible in the security parameter λ.

4.4 Proposed PKEET-AA scheme

In this section, we will provide a detailed construction for PKEET-AA scheme. Figure 2 illustrates the proposed PKEET-AA. The concrete construction as follows:

1. **Setup:**
   Taking a security parameter λ as input, this algorithm produces $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups with the a prime order p. It then outputs the public parameters as follows:
   - Choose g to be a generator for $\mathbb{G}_1$.
   - Determine a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
   - Select cryptographic secure hash functions: $H_1 : \{0,1\}^* \rightarrow \{0,1\}^w$, $H_2 : \mathbb{G}_1 \rightarrow \{0,1\}^{w_1+w_2}$, $H_3 : \mathbb{G}_1 \rightarrow \mathbb{G}_1$, $H_4, H_5 : \mathbb{G}_1 \rightarrow \{0,1\}^w$, $H_6 : \{0,1\}^w \rightarrow \{0,1\}^w$, $H_7 : \{0,1\}^w \times \mathbb{G}_1 \rightarrow Z^*_p$, where w_1, w_2 denotes the bit-length of $\mathbb{G}_1$ and $\mathbb{Z}_p$, respectively.

   The public system parameters are $pp = \{e, \mathbb{G}_1, \mathbb{G}_2, p, g, H_1, H_2, H_3, H_4, H_5, H_6, H_7\}$.
2. **Key-Gen:**
   Taking the public parameters pp, the algorithm selects $\alpha_i, \beta_i \in \mathbb{Z}^*_p$ then outputs the public/private key pair for each participant such as:$(sk_i, pk_i) = ((\alpha_i, \beta_i), (X_i = g^{\alpha_i},\ Y_i = g^{\beta_i}))$.
3. **Encrypt:**
   To encrypt $M_i \in \mathbb{G}_1$ with public key pk_i. It randomly selects $r_{i,1}, r_{i,2} \in \mathbb{Z}^*_p$, then sets the ciphertext $C_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, c_{i,4}, c_{i,5})$ as follows:
   $c_{i,1} = g^{r_{i,1}}$, $c_{i,2} = g^{r_{i,2}}$.
   $c_{i,3} = H_3(M_i) \cdot Y_i^{r_{i,2}}$.
   $c_{i,4} = (M_i \parallel r_{i,1}) \bigoplus H_2(X_i^{r_{i,1}})$.
   $c_{i,5} = H_1(c_{i,1} \parallel c_{i,2} \parallel c_{i,3} \parallel c_{i,4} \parallel M_i \parallel r_{i,1})$.
   Here $\parallel$ represents for concatenation symbol and $\bigoplus$ stands for XOR operation.
4. **Decrypt:**
   To decrypt ciphertext $C_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, c_{i,5})$ with user private key $sk_i = (\alpha_i, \beta_i)$, it works as follows:

   Parse M'_i and $r'_{i,1}$ from
   $M'_i \parallel r'_{i,1} = c_{i,4} \bigoplus H_2(c^{\alpha_i}_{i,1})$.
   Then outputs $M'_i = M_i$ if the following equalities hold:

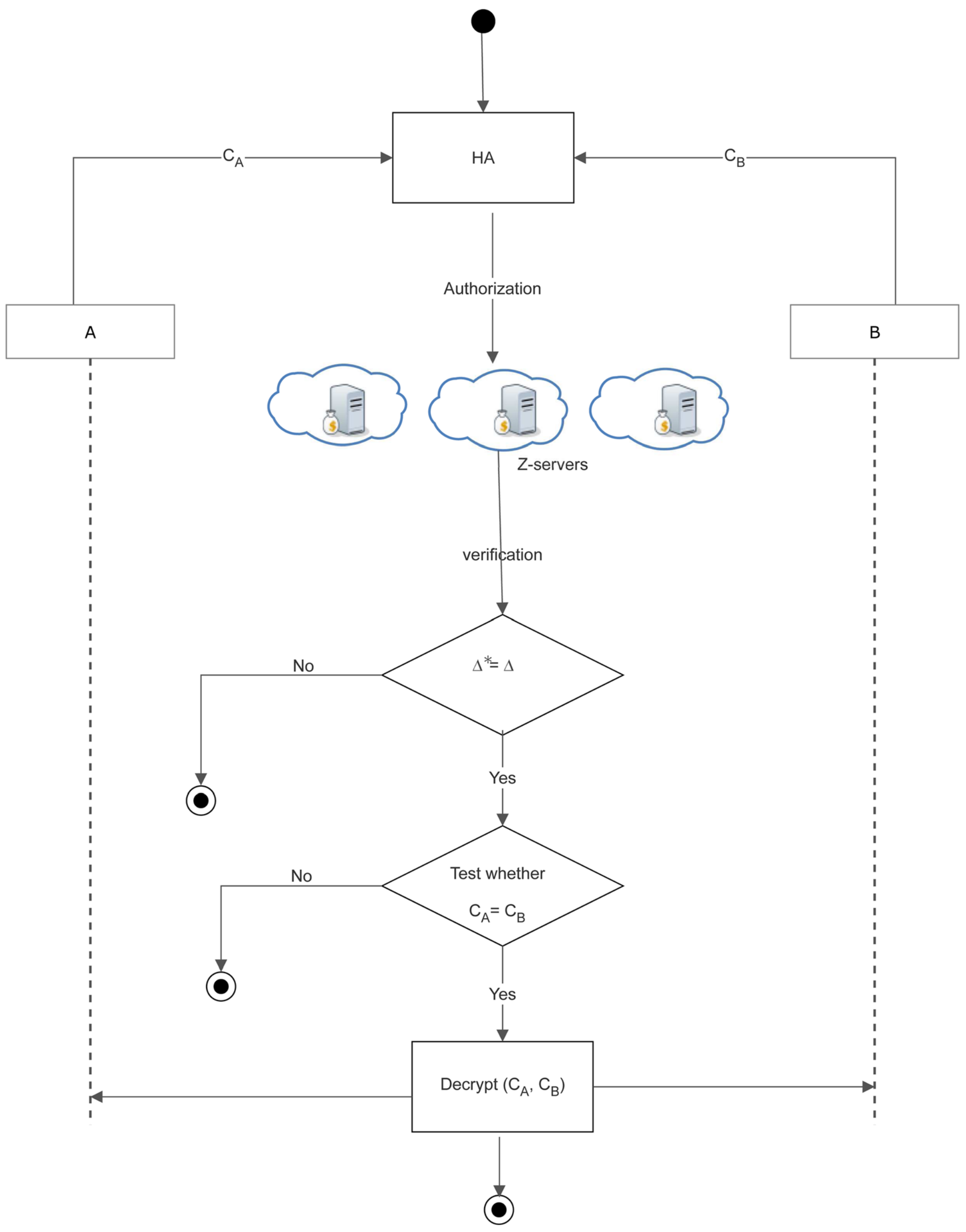

Fig. 2 Proposed PKEET AA scheme

$g^{r'_{i,1}} = c_{i,1}$;
$H_1(c_{i,1} \parallel c_{i,2} \parallel c_{i,3} \parallel c_{i,4} \parallel M'_i \parallel r'_{i,1}) = c_{i,5}$.

5. **Multi-authorization:**

Firstly, the sender U_i compute the disease-warrant w_d. Taking the public parameters pp; his/her private key sk_i, where $sk_i = (\alpha_i, \beta_i)$ and the ciphertext $C_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, c_{i,5})$, then compute $w_d = c_{i,2}^{\beta_i}$.

Secondly, suppose that the health administration (HA) obtains the sender sample matching $C_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, c_{i,5})$ along with its corresponding warrant w_d. The HA generates identical warrant (called a multi-warrant mw_i) to authorize z assigned server testers with public keys $pk_{s_i} = (X_{s_1}, Y_{s_1}), \cdots, (X_{s_z}, Y_{s_z})$, where $z < n$. Without loss of generality, set the assigned server testers public keys as $pk_{s_i} = (X_{s_i}, Y_{s_i}) = (X_{s_1} = g^{\alpha_{s1}}, Y_{s_1} = g^{\beta_{s1}}), \cdots, (X_{s_z} = g^{\alpha_{sz}}, Y_{s_z} = g^{\beta_{sz}})$. For $(i = 1, \cdots, z)$ assigned server testers, it performs the following steps:

- Picks a random number $\psi \in \mathbb{Z}_p^*$.
- Compute $U = g^{\psi}$ and $F_i = (X_{s_i})^{\psi}$.

- Pick an ephemeral value $\vartheta \in \{0,1\}^w$ at random and compute
  $mw_i = H_4(F_i) \parallel (H_5(F_i) \bigoplus \vartheta) = (mw_1, \cdots, mw_z)$.
- Use the ephemeral value ϑ to compute a symmetric key $sk = H_6(\vartheta)$, and generate $V = E_{sk}(w_d)$ and $\Delta = H_7(\vartheta, mw_i, w_d, V, U)$.
- Finally, set $Auth_d = (mw_i, U, V, \Delta)$.

6. **Extract:**
   Taking a ciphertext C_i; the $Auth_d$ value and the assigned server testers private keys $(\alpha_{s_i}, \beta_{s_i})$. The assigned server testers s_i executes the algorithm to extract the multi-warrant mw_i as follows:

   - Compute $F = U^{\alpha_{s_i}}$ and $H_4(F)$.
   - Use $H_4(F)$ to find the corresponding receivers warrant $mw_i \in (mw_1, \cdots, mw_z)$ by computing $mw_i = H_4(F) \parallel R$, where R denotes the remaining string by removing $H_4(F)$ from mw_i.
   - Extract $\vartheta^* = R \bigoplus H_5(F)$.
   - Set a symmetric key $sk^* = H_6(\vartheta^*)$, and compute $w_d{}^* = D_{sk^*}(V) = c_{i,2}^{\beta_i}{}^*$.

7. **Verification:**
   After the assigned server testers s_i extracted the user's warrant $w_d{}^*$. The assigned server testers verifies the validation of the warrant as follows:

   - Compute $\Delta^* = H_7(\vartheta^*, mw_i, w_d{}^*, V, U)$.
   - Check whether Δ^* and Δ are equal. If the equation holds, it outputs 1 as valid; othemwise, it outputs 0.

8. **Test:**
   To decide the test equivalence of two ciphertext $(C_i, C_i^{'})$, given two multi-authorization values $(Auth_d, Auth_d{}^{'}) = ((mw_i, U, V, \Delta), (mw_i^{'}, U^{'}, V^{'}, \Delta^{'}))$, the assigned server testers private keys $(\alpha_{s_i}, \beta_{s_i})$. It then works as follows:

   - Parse C_i as $C_i \leftarrow (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, c_{i,4}, c_{i,5})$.
   - After the assigned server testers s_i extracted and verified the user's warrant w_d (precisely after extracted and verified that the value of $(\Delta^* = \Delta)$. It then works as follows:

     Compute $\chi_i = H_3(M_i) = c_{i,3}/w_d = c_{i,3}/c_{i,2}^{\beta_i}$.

     Apply the same calculation to compute $\chi_i^{'}$ on input $C_i^{'}$ and $w_d{}^{'}$.

     Finally, verify whether $\chi_i^{'} = \chi_i$ or not. When the equation holds, the test algorithm outputs 1 for equivalent; othemwise, it outputs 0 for distinct.

The consistency property of our scheme is proven as follows:

Correctness: The algorithm *Decrypt* can parse $M_i^{'}$ and $r_{i,1}^{'}$ as follows:

$$\begin{aligned} M_i^{'} \parallel r_{i,1}^{'} &= c_{i,4} \bigoplus H_2(c_{i,1}^{\alpha_i}) \\ &= (M_i \parallel r_{i,1}) \bigoplus H_2(X_i^{r_{i,1}}) \bigoplus H_2(g^{r_{i,1}\alpha_i}) \\ &= (M_i \parallel r_{i,1}) \bigoplus H_2(g^{\alpha_i r_{i,1}}) \bigoplus H_2(g^{r_{i,1}\alpha_i}) \\ &= (M_i \parallel r_{i,1}) \end{aligned}$$

Then, it checks both $g^{r_{i,1}^{'}} = c_{i,1}$ and $H_1(c_{i,1} \parallel c_{i,2} \parallel c_{i,3} \parallel c_{i,4} \parallel M_i^{'} \parallel r_{i,1}^{'}) = c_{i,5}$. It is straightforward that the correctness holds along with the decryption algorithm.

Consistency: The algorithm *Test* can compute:

$$\begin{aligned} \chi_i &= c_{i,3}/w_d = c_{i,3}/c_{i,2}^{\beta_i} \\ &= H_3(M_i) \cdot Y_i^{r_{i,2}}/g^{r_{i,2}\beta_i} \\ &= H_3(M_i) \cdot g^{\beta_i{}^{r_{i,2}}}/g^{r_{i,2}\beta_i} \\ &= H_3(M_i) \cdot g^{\beta_i \cdot r_{i,2}}/g^{r_{i,2} \cdot \beta_i} \\ &= H_3(M_i) \end{aligned}$$

and similarly,

$\chi_i^{'} = H_3(M_i^{'})$

Thus, the equality

$\chi_i = \chi_i^{'}$

holds if and only if $M_i = M_i^{'}$, this is because H_3 is a collision resistant function. So the $Test(C_i, Auth_d, C_i^{'}, Auth_d{}^{'})$ algorithm always outputs 1; or 0 otherwise.

5 Security and performance analysis

5.1 Security analysis

In this section, we summaries the security of our PKEET-AA scheme by the following theorems.

Theorem 1 *Our PKEET-AA scheme is OW-CCA secure against Type-1 Adversary and in the random oracle model assuming the CDH problem is intractable.*

Theorem 2 *Our PKEET-AA scheme is IND-CCA secure against Type-2 Adversary and in the random oracle model assuming the DDH and CDH problems are intractable.*

Proof As to this enhanced new primitive, the existing properties in [5] are still holding, and their security proofs remain the same. Therefore, the OW-CCA security against Type-1 Adversary and IND-CCA security against Type-2 Adversary are quite similar to their proof. The only exception for authorization oracle, the modification as below:

Let $\mathscr{A}$ be a PPT adversary against PKEET-AA. We assume that $\mathscr{A}$ makes at most q_{H_i} random oracle queries to $H_i(1 \leq i \leq 6)$, q_D decryption queries, q_{M_A} authorization queries. L_{Hi} and Aut_L denotes for the hash list $\forall(1 \leq i \leq 6)$ and a authorization list, respectively. $\mathscr{B}$ can simulate the Challenger's execution of each hybrid games, controls oracle $\mathcal{O}^{H_i}$, $\mathcal{O}^D$ and $\mathcal{O}^{M_A}$; after that $\mathscr{B}$ takes advantage of $\mathscr{A}$ breaking the hardness problems if $\mathscr{A}$ attaks PKEET-AA.

The multi-authorization oracle $\mathcal{O}^{M_A}$: On receiving multi-authorization request, the authorization oracle work as follows:

1. For the target user t and selected z assigned server public keys (pk_{s_i}), if $C_i \neq \hat{C}_t$, it returns the authorization information $Au\hat{t}h_d = (m\hat{w}_t, \hat{U}, \hat{V}, \hat{\Delta})$.
2. Else, it checks if $(pk_t, pk_{s_i}, \psi, m\hat{w}_t, \hat{U})$ exists in Aut_L, it returns $(m\hat{w}_t, \hat{U}, \hat{V}, \hat{\Delta})$ on authorization list.
3. Else, it picks $\psi \xleftarrow[\mathbb{Z}]{R} p^*$, computes $Au\hat{t}h_d$ as: Without loss of the generality, $\mathscr{B}$ first selects z target server public keys (pk_{s_i}), denoted by $(pk_{s_1} \cdots, pk_{s_z})$, where $(i = 1, \cdots, z)$ and $z < n$. Taking the public parameters pp; an U's private key sk_i for the sender, where $sk_t = (\alpha_t, \beta_t)$ and the ciphertext $\hat{C}_t = (c\hat{_{t,1}}, c\hat{_{t,2}}, c\hat{_{t,3}}, c\hat{_{t,4}}, c\hat{_{t,5}})$, then set $\hat{w_d} \leftarrow c\hat{_{t,2}}^{\beta_t}$.

   For $(i = 1, \cdots, z)$ assigned server testers, it performs the following steps:

   $\mathscr{B}$ picks $\psi \xleftarrow[\mathbb{Z}]{R} p^*$ and execute the authorization algorithm as follows:

   $\hat{U} \leftarrow g^{\psi}$, $\hat{F}_t \leftarrow (X_{s_i})^{\psi}$, $\vartheta \leftarrow \{0, 1\}^w$, $m\hat{w}_t$ $H_4(\hat{F}_t) \parallel (H_5(\hat{F}_t) \bigoplus \vartheta)$, $s\hat{k}$ $H_6(\vartheta)$, $\hat{V} \leftarrow E_{s\hat{k}}(\hat{w_d})$, $\hat{\Delta}$ $H_7(\vartheta, m\hat{w}_t, \hat{w_d}, \hat{V}, \hat{U})$, $Au\hat{t}h_d \leftarrow (m\hat{w}_t, \hat{U}, \hat{V}, \hat{\Delta})$.

   Then records $(m\hat{w}_t, \hat{U}, \hat{V}, \hat{\Delta})$ on authorization list Aut_L, and $Au\hat{t}h_d$ is returned as the authorization information to the adversary $\mathscr{A}$.

This completes the proof of Theorems 1 and 2. □

Theorem 3 *Our PKEET-AA scheme is Anon-Auth secure according to Definition 3 under the DDH assumption.*

Proof We perceive that part of the authorization information (warrant w_d value) is a ciphertext generated by the El Gamal encryption. Thus, it is easy to prove that the PKEET-AA scheme is Anon-Auth secure under the DDH assumption on $\mathbb{G}_1$. Let $\mathscr{A}$ be a PPT adversary against Anon-Auth of the PKEET-AA scheme. We suppose there is an algorithm $\mathscr{B}$ that solves the DDH problem, on input $(g^v, g^u, g^z) \in \mathbb{G}_1$, then $\mathscr{B}$ work as follows:

1. Set the the server's public key $pk_{si} = pk_s = (X_s = g^{\alpha_s}, Y_s = g^{\beta_s})$.
2. Obtain two pair keys of senders t_0 and t_1 from $\mathscr{A}$: $(x_0 = \alpha_0, y_0 = \beta_0)(X_0 = g^{\alpha_0}, Y_0 = g^{\beta_0} = g^v)$ and two pair keys for t_1, via invoke $\mathscr{A}$ on input $pk_s = (X_s = g^{\alpha_s}, Y_s = g^{\beta_s})$ with the same distribution as that in PKEET-AA.
3. Randomly chooses $\rho \in \{0, 1\}$ and considering on $c\hat{_{t,2}} = g^u$, compute $\hat{w_d} = c\hat{_{t,2}}^{\beta_\rho} = g^{u\beta_\rho}$ and then generates $Au\hat{t}h_d$ by taking (pp, pk_s, sk_ρ) as input as follows: $\hat{\psi} \xleftarrow[\mathbb{Z}]{R} p^*$, $\hat{U}$ $g^{\hat{\psi}}$, $\hat{\vartheta} \xleftarrow[\{]{R} 0, 1\}^w$, $\hat{x} \xleftarrow[\{]{R} 0, 1\}^w$, $\hat{x'} \xleftarrow[\{]{R} 0, 1\}^w$, $m\hat{w}$ $\hat{x} \parallel (\hat{x'} \bigoplus \hat{\vartheta})$, sk $H_6(\hat{\vartheta})$, $\hat{V} \leftarrow E_{sk}(g^{a\beta_\rho})$, $\hat{\Delta}$ $H_7(\hat{\vartheta}, m\hat{w}, g^{u\beta_\rho}, \hat{V}, \hat{U})$, $Au\hat{t}h_d$ $(m\hat{w}, \hat{U}, \hat{V}, \hat{\Delta})$. Then, returns $Au\hat{t}h_d$ to $\mathscr{A}$.
4. Guess. The adversary $\mathscr{A}$ responds with its guess $\rho' \in \{0, 1\}$. If $\rho = \rho'$ output 1; othemwise, output 0.

On receiving a multi-authorization request, the authorization oracle for $\mathscr{A}$ simulated as follows.

Given an instance $(Y_0 = g^{\beta_0} = g^v, c\hat{_{t,2}} = g^u, \hat{w_d} = c\hat{_{t,2}}^{\beta_\rho} = g^{u\beta_\rho} = g^z org^{uv})$ of the DDH problem, let us analyze the behavior $\mathscr{B}$ of as below:

1. *Case 1:* $z \xleftarrow[\mathbb{Z}]{R} p$. The warrant value $\hat{w_d}$ constructed as g^z. The view of $\mathscr{A}$ in the simulation by $\mathscr{B}$ is perfect, if and only if $\mathscr{B}$ outputs ρ' of $\mathscr{A}$ is equal to ρ , we have $Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^z) = 1] = 1/2$.
2. *Case 2:* $z = uv$. The warrant value $\hat{w_d}$ constructed as g^{uv}. The view of $\mathscr{A}$ in the simulation by $\mathscr{B}$ is perfect, if and only if $\mathscr{B}$ outputs ρ' of $\mathscr{A}$ is equal to ρ , we have $Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^{uv}) = 1] = \epsilon(\lambda)$.

It is well-known that the DDH problem is hard with exist a negligible function

$$negl(\lambda) \geq \mid Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^z) = 1] - Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^{uv}) = 1] \mid = \mid 1/2 - \epsilon(\lambda) \mid$$

This implies that $\epsilon(\lambda) \leq 1/2 + \text{negl}(\lambda)$, therefore, our proposed PKEET-AA scheme is Anon-Auth secure against $\mathscr{A}$.

□

Theorem 4 *Our PKEET-AA scheme is Anon-MAuth secure according to Definition 4 under the DDH assumption.*

Proof We observe that part of the authorization information (F_i value) is a ciphertext generated by the El Gamal encryption. Thus, it is easy to prove that the PKEET-AA scheme is Anon-MAuth secure under the DDH assumption on $\mathbb{G}_1$. Let $\mathscr{A}$ be a PPT adversary against Anon-Auth of the PKEET-AA scheme. We suppose there is an algorithm $\mathscr{B}$ that solves the DDH problem, on input $(g^v, g^u, g^z) \in \mathbb{G}_1$, then $\mathscr{B}$ work as follows:

1. Invoke $\mathscr{A}$ and obtain a target warrant w_d and $t-1$ server's public key chosen from $S - \{pk_{s1}, pk_{s2}\}$, denoted by $pk_{s2}, \cdots, pk_{st}$
2. Randomly chooses $\rho \in \{0,1\}$ and then generates $\hat{Auth_d}$ as follows:

   (a) Pick $\hat{\psi} \xleftarrow[\mathbb{Z}]{R} p^*$ and set $\hat{U} \leftarrow g^{\hat{\psi}}$.

   (b) Pick a string $\hat{\vartheta} \xleftarrow[\{]{R} 0,1\}^w$.

   (c) Randomly pick $\hat{x_\rho} \xleftarrow[\{]{R} 0,1\}^w$ and compute $m\hat{w}_1 \leftarrow \hat{x_\rho} \parallel (\hat{x_\rho} \bigoplus \hat{\vartheta})$.

   (d) For $i = 2, \ldots, t$ assigned server testers, do the following:

   (e) Compute $sk = H_6(\hat{\vartheta})$, $\hat{V} = E_{sk}(w_d)$, $\hat{\Delta} = H_7(\hat{\vartheta}, m\hat{w}_1, m\hat{w}_2, \cdots, m\hat{w}_t, w_d, \hat{V}, \hat{U})$, $\hat{Auth_d} = (m\hat{w}_1, m\hat{w}_2, \cdots, m\hat{w}_t, \hat{U}, \hat{V}, \hat{\Delta})$. Then, returns $\hat{Auth_d}$ to $\mathscr{A}$.

   i. Invoke $\mathscr{A}$ and obtain $(sk_{si}', pk_{si}') = ((\alpha_{si}', \beta_{si}'), (X_{si}' = g^{\alpha_{si}'}, Y_{si}' = g^{\beta_{si}'}))$, and compute $F_i' = (\hat{U})^{\alpha_{si}'}$.

   ii. Make $H_4(F_i')$ and $H_5(F_i')$ queries to obtain $\hat{x_i} \in \{0,1\}^w$ and $\hat{x_i'} \in \{0,1\}^w$.

   iii. Compute $m\hat{w}_i \quad \hat{x_i} \parallel (\hat{x_i'} \bigoplus \hat{\vartheta})$.

3. Guess. The adversary $\mathscr{A}$ responds with its guess $\rho' \in \{0,1\}$. If $\rho = \rho'$ output 1; othemwise, output 0.

On receiving a multi-authorization request, the authorization oracle for $\mathscr{A}$ simulated as follows.

Given an instance $(Y_{si}' = g^{\beta_{si}'} = g^v, \hat{U} = g^{\hat{\psi}} = g^u, F_i' = (\hat{U})^{\alpha_{si}'} = g^z org^{uv})$ of the DDH problem, let us analyze the behavior $\mathscr{B}$ of as below:

1. *Case 1:* $z \xleftarrow[\mathbb{Z}]{R} p$. The value of F_i' is constructed as g^z. The view of $\mathscr{A}$ in the simulation by $\mathscr{B}$ is perfect, if and only if $\mathscr{B}$ outputs ρ' of $\mathscr{A}$ is equal to ρ , we have $Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^z) = 1] = 1/2$.

2. *Case 2:* $z = uv$. The value of F_i' is constructed as g^{uv}. The view of $\mathscr{A}$ in the simulation by $\mathscr{B}$ is perfect, if and only if $\mathscr{B}$ outputs ρ' of $\mathscr{A}$ is equal to ρ , we have $Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^{uv}) = 1] = \epsilon(\lambda)$.

It is well-known that the DDH problem is hard with exist a negligible function

$$negl(\lambda) \geq \mid Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^z) = 1] - Pr[\mathscr{B}(\mathbb{G}_1, g, g^v, g^u, g^{uv}) = 1] \mid = \mid 1/2 - \epsilon(\lambda) \mid$$

This implies that $\epsilon(\lambda) \leq 1/2 + \text{negl}(\lambda)$, therefore, our proposed PKEET-AA scheme is Anon-MAuth secure against $\mathscr{A}$. □

5.2 Performance analysis

In this section, we evaluate the performance of our proposed PKEET-AA scheme as well as the performance of the schemes mentioned in [4, 6, 12, 13, 15, 21]. The comparison is primed based on the experimental results in [16, 22, 23], for various cryptographic operations using MIRACLE [24] in Intel(R) Xeon(R) E5-26300@2.30 GHz CPU platform processor with memory 1 GB and the Ubuntu 14.04 operating system. From this experimental results, we can notice that the relative running time of one pairing operation T_e is 5.275 m/s, one-way hash function T_h is 0.009 m/s, HashToPoint function T_H is 5.101 m/s, scalar multiplication T_{sm} is 1.97 m/s and exponential operation T_{exp} is 0.331 m/s, where the symmetric key encryption and decryption running times are very close to hash function running time [14, 25]. Hence, we adopted the same running time of one-way hash function for both encryption T_{se} and decryption T_{sd}.

For the sake of convenience, we define the following notations: T_h(the time complexity of one-way hash function); T_H(the time complexity of HashToPoint function); T_e (the time complexity of pairing operation); T_{sm} (the time complexity of scalar multiplication); T_{mul} (the time complexity of pairing-based scalar multiplication); T_{exp} (the time complexity of exponential operation); T_{sd} (the time complexity of symmetric decryption); T_{se} (represents the symmetric encryption).

We compare our scheme's communication efficiency in terms of computation, communication costs, and privacy preservation against its counterpart PKE-DET scheme [4] since the latter is the only scheme in previous works that provide the anonymity feature for data owners. As we can observe in both Table 1 and Fig. 3, PKE-DET [4] has a higher computational cost than our scheme, since their scheme uses too many bilinear map operations. According to Table 2, PKE-DET scheme does not provide a warrant

Table 1 Efficiency comparison

Schemes	Operations			
	Encryption	Decryption	Authorization	Test
FG-PKEET [6]	$4T_{exp}+2\mathrm{T_h}+\mathrm{T_H}=6.443$	$2T_{exp}+2\mathrm{T_h}=0.6\#$	$3T_{exp}=0.993$	$4T_e=21.1$
IBEET-FA [15]	$5T_{exp}+5\mathrm{T_h}++\mathrm{T_H}+2\mathrm{T_e}=17.351$	$2T_h+2\mathrm{T_e}=10.56\#$	$T_{exp}+\mathrm{T_H}=5.432$	$4T_e+2\mathrm{T_h}=21.11\#$
IBEET [12]	$6T_{exp}+2\mathrm{T_H}+2\mathrm{T_e}+\mathrm{T_h}=22.747$	$2T_{exp}+\mathrm{T_h}+\mathrm{T_H}+2\mathrm{T_e}=16.322$	$T_{exp}+\mathrm{T_H}=5.432$	$4T_e+2\mathrm{T_H}=31.302$
IBEET [13]	$2T_{exp}+2\mathrm{T_h}+\mathrm{T_H}=5.7\#1$	$2T_h+\mathrm{T_H}+2\mathrm{T_e}=15.66\times$	$T_h=0.009$	$2T_{exp}+2\mathrm{T_e}+2\mathrm{T_H}=21.414$
IBE-FET [21]	$4T_{sm}+\mathrm{T_{exp}}+2\mathrm{T_e}+2\mathrm{T_h}+2\mathrm{T_H}=2\#.\times\#1$	$5T_{sm}+\mathrm{T_{exp}}+\mathrm{T_H}+\mathrm{T_h}+\mathrm{T_e}=20,566$	$2T_{sm}+\mathrm{T_{exp}}=4.271$	$2T_{sm}+2\mathrm{T_e}=14.4\times$
PKE-DET [4]	$5T_{exp}+\mathrm{T_e}+\mathrm{T_h}+\mathrm{T_H}=12.04$	$4T_{exp}+\mathrm{T_e}+\mathrm{T_h}+\mathrm{T_H}=11.70\times$	$3T_{exp}=0.993$	$2T_{exp}+4\mathrm{T_e}+2\mathrm{T_H}=31.\times64$
PKEET-AA	$4T_{exp}+2\mathrm{T_h}+\mathrm{T_H}=6.443$	$2T_{exp}+2\mathrm{T_h}=0.6\#$	$3T_{exp}+4\mathrm{T_h}+\mathrm{T_{se}}=1.03\#$	$3T_{exp}+4\mathrm{T_h}+\mathrm{T_{sd}}=1.03\#$

verification method, and so our scheme has much better efficiency.

As shown in Table 2, we can observe that the communication cost of scheme [21] increases linearly with the number of messages, while the proposed schemes [4, 9] are fixed value. However, we find that existing the bilinear pairings in the test algorithm lets our proposed scheme is smaller than scheme [21]. Additionally, our scheme's warrant security assumption is stronger than the DLP that uses in scheme [21]. Thus, the proposed PKEET-AA scheme can be considered more secure to ensure the privacy of sender than that of scheme [21].

In the field of healthcare applications in cloud computing, power consumption, computational cost, and communications overhead cause serious problems. According to Table 1, and despite that our authorization algorithm has slightly higher computational cost than those in schemes [4, 6, 13], our test algorithm still has the most lower consumption cost than schemes [4, 6, 12, 13, 15, 21]. Moreover, our decryption algorithm is the same efficient as the scheme [6], which is less computational cost than those of other schemes [4, 12, 13, 15, 21]. Following the first PKEET [3], many schemes suffer from using bilinear pairings in the test algorithms [4, 6, 12, 13, 15, 21]. But the relative computation cost of the pairing is approximately 20 times higher than that of the scalar multiplication over elliptic curve group, which add extra burden to the cloud servers.

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 health care application and feasible to cloud computing.

6 Applications

In the following, several applications, including the anonymous multi-vendor e-marketplace system, e-mail service system, and financial transactions, are presented to demonstrate how our new PKEET approach to implemented in real-world applications. The anonymous multi-vendor e-marketplace system discussed in detail, but the other two are only briefly touched on.

For further clarification, we provide another scenario of our new PKEET approach for an anonymous multi-vendor e-marketplace system, as shown in Fig. 4. There is a *Broker*, that facilitates communications between buyer and seller. There are *k Vendors* that sell their items, and there are several *Customers*, who intend to shop at more than one Vendor. In this scenario, we suppose that the customers' data is encrypted, both at rest (when stored on a local server) and in transit (i.e. when being sent to the broker).

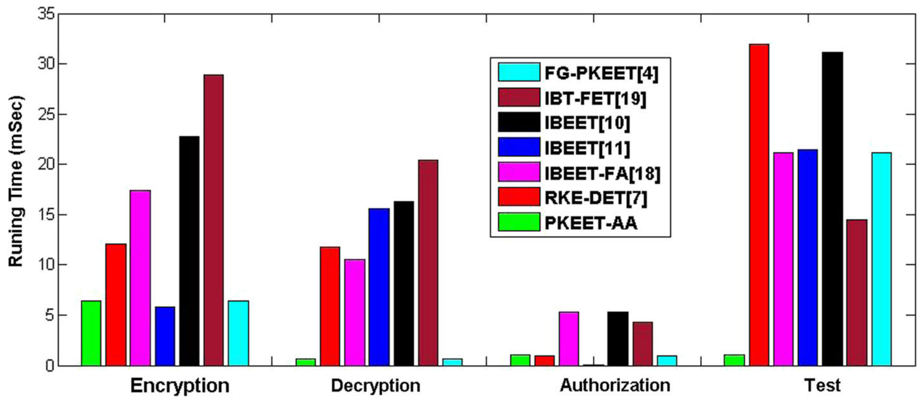

Fig. 3 The efficiency comparison

Table 2 Comparative summary

		PKE DET [4]	PKE AET [9]	IBE FET [21]	PKEET AA
Storage	Public key	$2 \mid \mathbb{G} \mid$	$2 \mid \mathbb{G} \mid$	$(n{+}2) \mid \mathbb{G} \mid$	$2 \mid \mathbb{G} \mid$
	Ciphertext	$4 \mid \mathbb{G} \mid + \mid Z_q \mid$	$3 \mid \mathbb{G} \mid + \mid Z_q \mid$	$(n{+}2) \mid \mathbb{G} \mid + \mid \mathbb{G}_T \mid + \mid Z_q \mid$	$4 \mid \mathbb{G} \mid + \mid Z_q \mid$
	Warrant	$2 \mid \mathbb{G} \mid$	$2 \mid \mathbb{G} \mid$	$(n{+}1) \mid \mathbb{G} \mid$	$\mid \mathbb{G} \mid$
Warrant	Ciphertext	✓	✗	✓	✓
	Data owner privacy	✓	✗	✓	✓
	Assigned server privacy	✗	✗	✗	✓
	Verification	✗	✓	✗	✓
Warrant security	Assumption	DDH		DLP	DDH
Cipher security		OW CCA/IND CCA	OW CCA/IND CCA	OW ID CCA	OW CCA/IND CCA

$\mid \mathbb{G} \mid, \mid \mathbb{G}_T \mid$: the bit length of point in group $\mathbb{G}$ and $\mathbb{G}_T$ respectively, $\mid Z_q \mid$: the bit length of number $\in Z_q$, n: the number of messages

To anonymously test whether two encrypted messages are actually encrypted with the same plane message. In other words, to anonymously test whether the same item is sold out by two different vendors, a customer C_A encrypts his private information M_{C_A} (e.g item, vendor) under the public key pk_{C_A} and the vendor's public key pk_V, and sends the encrypted massage to the broker with warrant w_A. A customer C_B does exactly as C_A did. Upon receiving the two encrypted messages from C_A and C_B, the broker could determine any server tester to search whether M_{C_A} and M_{C_B} have the identical items. However, there is no knowledge of what the real item and Vendor. Afterward, the broker sends the search result to the customer C_A and C_B, allowing them to share their results. *Financial audit logs* contain a very sensitive information about financial transactions. Our PKEET-AA approach allows financial institutions to release audit logs in encrypted format. For instance, suppose that there are fraudsters or unscrupulous merchants repeatedly submit authorization and/or clearing requests for the same financial transactions after having been initially declined. When it does, our PKEET-AA approach allows two auditors from different financial institutions to encrypt their private information (e.g merchant, requested transaction, transaction status= declined). Then, they send this information for the equality test. After the auditors receive the result, they share it and take the necessary legal and financial action.

Electronic mail (email or e-mail) is a method of exchanging messages for confidential personal as well as business needs. To authenticate the sender of an e-mail message at the receiver's client and guarantee that the message can be read-only by the intended recipient, many approaches are used. The most common approach is to encrypt e-mails and store them at mail servers. Our PKEET-AA approach allows e-mail users to anonymously test whether two encrypted e-mail messages are actually encrypted with the same plain message or not, as in previous scenarios.

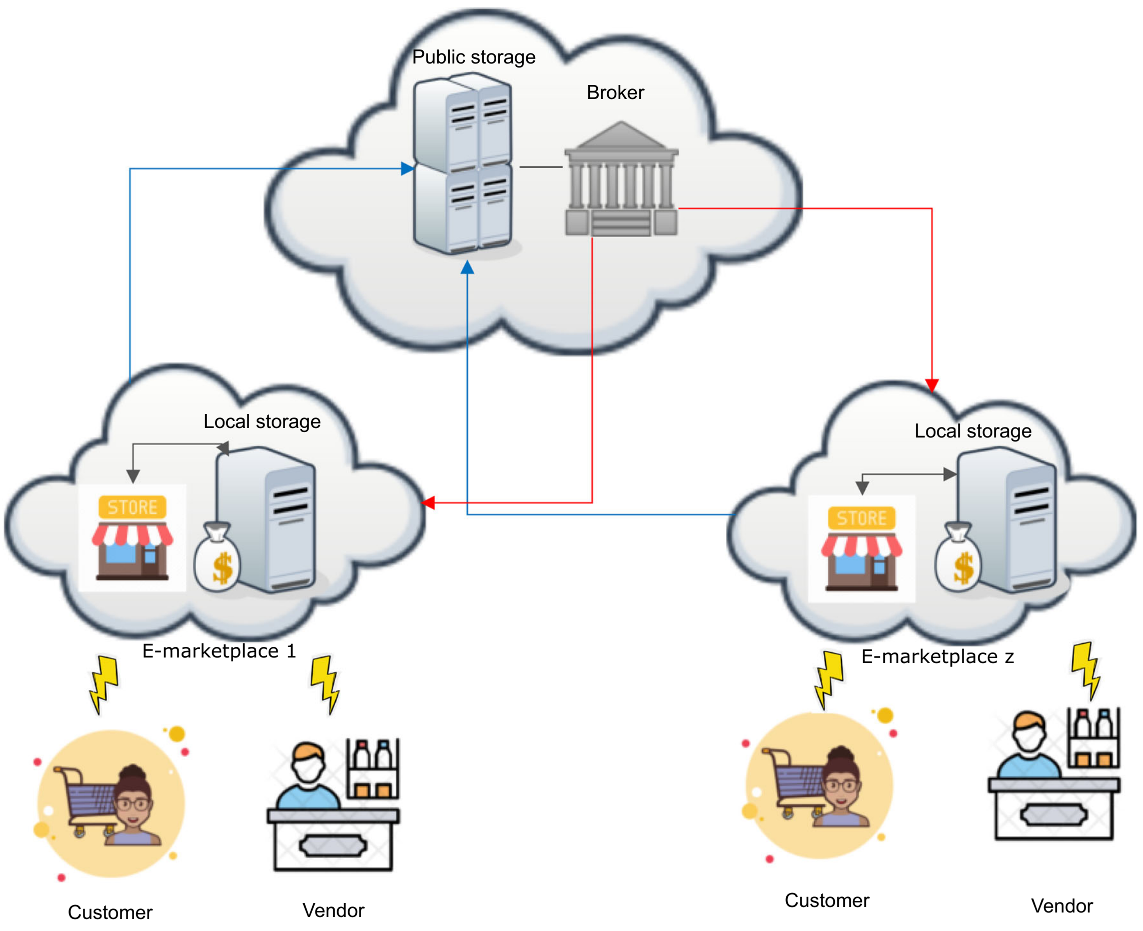

Fig. 4 Scenario 2: multi vendor e marketplace system

7 Conclusion

This paper is a combination of PKEET-FA and V-PKEET schemes by employing the "PKEET" scheme. The concrete construction and the security analysis proved that our proposed PKEET-AA scheme is secure under the security model based on the random oracle model by the mean of the Decisional Diffie Hellman (DDH) assumption. To the best of our knowledge, our scheme is the first work sup-porting an equality test technique in a multi-server sce-nario. Our PKEET-AA scheme based on traditional public-key infrastructure (PKI) setting in which is required semi-trusted infrastructure nodes and is identified to be vulner-able to traffic analysis and some other kind of active attacks. Furthermore, the PKEET-AA can achieve two types of anonymization, which can protect the patients' information by providing a privacy-preserving technique that can protect sensitive information in the test servers from each other. Our scheme is more secure and efficient compared to other proposed approaches that used PKI and IBE cryptosystems. More specifically, our scheme achieves low computational cost during the decryption and equality test processes. Thus, we argue that our scheme is more compatible with cloud applications, including the health-care system. This work's outcomes could serve the implementation of further effective countermeasures rela-ted to cloud computing applications to solve security and privacy issues. Our future work will focus on designing and employing more equality test techniques under different cryptosystems such as identity-based cryptography (IBC) and certificateless cryptography (CL-PKC).

Acknowledgements Funding was provided by National Natural Sci ence Foundation of China (Grant No. 61472066).

References

1. Whitfield, D., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*(6), 644 654.
2. Dan, B., Di Crescenzo, G., Ostrovsky, R., & Persiano, G. (2004). Public key encryption with keyword search. In *International conference on the theory and applications of cryptographic techniques* (pp. 506 522). Berlin, Heidelberg: Springer.

3. Yang, G., Tan, C. H., Huang, Q., & Wong, D. S. (2010). Prob abilistic public key encryption with equality test. CT RSA, CA, USA, March 1 (pp. 119 131). Springer.
4. Tang, Q. (2012). Public key encryption schemes supporting equality test with authorisation of different granularity. *Interna tional Journal of Applied Cryptography*, *2*(4), 304 321.
5. Mohammed, R., Yongjian, L., Fagen, L., Shijie, Z., & Hisham, A. (2019). IBEETRSA: Identity based encryption with equality test over RSA for wireless body area network. *Mobile Networks and Applications*, *25*, 223 233.
6. Ma, S., Zhang, M., Huang, Q., & Yang, B. (2014). Public key encryption with delegated equality test in a multi user setting. *The Computer Journal*,. https://doi.org/10.1093/comjnl/bxu026.
7. Yan, X., Wang, M., Zhong, H., Cui, J., Liu, L., & Franqueira, V. N. L. (2017). Verifiable public key encryption scheme with equality test in 5G networks. *IEEE Access*, *5*, 12702 12713.
8. Tang, Q. (2011). Towards public key encryption scheme sup porting equality test with fine grained authorization. In: *Pro ceedings of 16th Australasian conference information security and privacy. Melbourne, Australia* (Vol. 6812, pp. 389 406).
9. Tang, Q. (2012). Public key encryption supporting plaintext equality test and user specified authorization. *Security and Communication Networks*, *5*(12), 1351 1362.
10. Huang, K., Tso, R., Chen, Y. C., Rahman, S. M. M., Almogren, A., & Alamri, A. (2015). PKE AET: Public key encryption with authorized equality test. *The Computer Journal*,. https://doi.org/10.1093/comjnl/bxv025.
11. Wu, L., Zhang, Y., Choo, K. K. R., & He, D. (2017). Efficient and secure identity based encryption scheme with equality test in cloud computing. *Future Generation Computer Systems*, *73*, 22 31.
12. Ma, S., Huang, Q., Zhang, M., & Yang, B. (2015). Efficient public key encryption with equality test supporting flexible authorization. *IEEE Transactions on Information Forensics and Security*, *10*, 458 470.
13. Boneh, D., & Franklin, M. (2001). Identity based encryption from the Weil pairing. In *Advances in cryptology, CRYPTO 2001* (pp. 213 229). Springer.
14. Andrew, Y. C. (1982). Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)* (pp. 160 164). IEEE.
15. Du, M., Wang, K., Chen, Y., Wang, X., & Sun, Y. (2018). Big data privacy preserving in multi access edge computing for heterogeneous Internet of Things. *IEEE Communications Magazine*, *56*(8), 62 67.
16. MIRACL, Multiprecision Integer and Rational Arithmetic C/C++ Library. https://indigo.ie/mscott/.
17. Ramesh, S., & Bhaskaran, V. M. (2014). An improved remote user authentication scheme with elliptic curve cryptography and smart card without using bilinear pairings. *International Journal of Engineering and Technology (IJET)*, *5*(6), 5140 5154.
18. Wang, D., He, D., Wang, P., & Chu, C. H. (2014). Anonymous two factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Transactions on Dependable and Secure Computing*, *PP*(99), 1 15.
19. Jiang, Q., Ma, J., Li, G., & Yang, L. (2013). An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks. *Wireless Personal Commu nications*, *68*(4), 1477 1491.
20. Kouichi, S., & Shizuya, H. (1995). Relationships among the computational powers of breaking discrete log cryptosystems. In *Advances in cryptology EUROCRYPT'95* (pp. 341 355). Berlin/Heidelberg: Springer.
21. Ming, Y., & Erxiu, W. (2019). Identity based encryption with filtered equality test for smart city applications. *Sensors*, *19*(14), 3046.
22. He, D., & Chen, J. (2013). An efficient certificate less designated verifier signature scheme. *The International Arab Journal of Information Technology*, *10*(4), 389 396.
23. Hea, D., Chen, J., & Zhang, R. (2011). An efficient identity based blind signature scheme without bilinear pairings. *Computers & Electrical Engineering*, *37*(4), 444 450.
24. Rivest, R., Adleman, L., & Dertouzos, M. (1978). On data banks and privacy homomorphisms. *Foundations of Secure Computa tion*, *4*, 169 179.
25. Tushar, S., & Takeshi, K. (2018). Outsourcing private equality tests to the cloud. *Journal of Information Security and Applica tions*, *43*, 83 98.

Hisham Abdalla is a postdoc toral research fellow at School of Information and Software Engineering, University of Electronic Science and Tech nology of China, Chengdu, P.R. China. He received his Ph.D. degree in information security from School of Information and Software Engineering, Univer sity of Electronic Science and Technology of China in 2016. His research interests include cloud computing security, cryptography and digital right management.

Hu Xiong is a professor at School of Information and Software Engineering, Univer sity of Electronic Science and Technology of China, Chengdu, P.R. China. He received his Ph.D. degree from University of Electronic Science and Tech nology of China in 2009. His research interests include infor mation security and crypto graphic protocols.

Abubaker Wahaballa is a post doctoral research fellow at School of Information and Software Engineering, Univer sity of Electronic Science and Technology of China, Chengdu, P.R. China. He received his Ph.D. degree from University of Electronic Science and Tech nology of China. His research interests include information security, cryptography, steganography and DevOps.

Mohammed Ramadan is a post doctoral research fellow at School of Information and Software Engineering, Univer sity of Electronic Science and Technology of China, Chengdu, P.R. China. He received his Ph.D. degree in information security from School of Com puter Science and Engineering, University of Electronic Science and Technology of China in 2016. His research interests include information security, cryptographic protocols, and wireless/mobile networks security.

Zhiguang Qin is a professor at School of Information and Software Engineering, Univer sity of Electronic Science and Technology of China, Chengdu, P.R. China. His research interest include network security, social network. He has published more than 100 papers in international journals and conferences among which more than 50 are indexed by SCI and EI. He has been principal investor of 2 NSF key projects, 2 sub topics of national major projects and 6 national 863 projects.