Information Sciences 373 (2016) 419–440

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Semi-generic construction of public key encryption and identity-based encryption with equality test

Hyung Tae Lee[a], San Ling[a], Jae Hong Seo[b,*], Huaxiong Wang[a]

[a] Division of Mathematical Sciences, School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore
[b] Department of Mathematics, Myongji University, Republic of Korea

ARTICLE INFO

Article history:
Received 17 March 2016
Revised 29 August 2016
Accepted 5 September 2016
Available online 6 September 2016

Keywords:
Public key encryption
Identity-based encryption
Equality test
Random oracle model

ABSTRACT

Public key encryption with equality test (PKEET), which was first introduced by Yang et al. (CT-RSA, 2010), has various applications including facilitating keyword search on encrypted data and partitioning encrypted data on the cloud. It can be also applied to manage personal health records on the internet. For these reasons, there have been improvements on earlier PKEET schemes in terms of performance and functionality.

We present a *semi-generic* method for PKEET constructions, assuming only the existence of IND-CCA2 secure traditional public key encryption (PKE) schemes, the hardness of Computational Diffie-Hellman (CDH) problems, and random oracles. Our approach has several advantages; it enables us to understand requirements for the equality test functionality more clearly. Furthermore, our approach is quite general, in that if we change the underlying PKE scheme with the identity-based encryption (IBE) scheme (and we assume the hardness of Bilinear Diffie-Hellman problems instead of CDH), then we obtain the first IBE scheme with equality test (IBEET) satisfying analogous security arguments to those of PKEET. Although an IBEET construction was recently proposed, but we note that it satisfies only weak security requirements.

1. Introduction

Public key encryption with equality test (PKEET), which was first introduced by Yang et al. [19], is a public key encryption (PKE) scheme that supports the capability for testing equality between ciphertexts using different public keys as well as the same public key. This property can be applied to various scenarios in practice. In particular, it is very useful for managing outsourced databases in a secure way.

Let us consider the following scenario to elucidate an advantage of using PKEET in secure outsourced database applications. Suppose that each user stores his/her emails with an email service provider. In this case, we face with two seemingly conflicting requirements, protecting data privacy and managing stored data efficiently. For the former, storing emails in the encrypted form seems necessary. But, it precludes operations over stored data without decrypting it, in particular, keyword search over stored emails. To support it, the email service provider makes senders append encrypted keywords to an en-

* Corresponding author.
E-mail addresses: hyungtaelee@ntu.edu.sg (H.T. Lee), lingsan@ntu.edu.sg (S. Ling), jaehongseo@mju.ac.kr, jhsbhs@gmail.com (J.H. Seo), hxwang@ntu.edu.sg (H. Wang).

http://dx.doi.org/10.1016/j.ins.2016.09.013
0020-0255/

crypted email, and may check encrypted keywords to response to user's keyword search queries or to filter out spam emails. However, traditional PKE schemes do not allow such operations to be performed over encrypted data.

Fully homomorphic encryption (FHE) [8] could be a suitable candidate to resolve the above issue, but the service provider cannot also check the result of operations without decrypting it. Searchable encryption [3] or deterministic encryption [2] could also be utilized. However, these primitives are basically designed to perform tests on ciphertexts generated by the same public key. Hence, the email service provider in the above scenario has to generate a token for each user in the system to monitor stored emails. On the other hand, a goal of PKEET is to enable one who has trapdoors to check equality among ciphertexts generated by different public keys as well as the same public key, so that the email service provider can perform tests on ciphertexts regardless of exploited public keys.

Furthermore, as suggested by Tang [18], PKEET can also be applied to emerging computing scenarios, e.g., internet-based private health record (PHR) applications [15,18]. In a PHR system, each patient may obtain his/her data from various sources: prescription results from a doctor, treatment from a hospital, test results from a laboratory, and so on. The patient receives such data as encrypted using his/her own public key, and stores them with the service provider. When he/she wants to match his/her data with that of others in order to get some help, he/she requests the service provider to search for them over encrypted data using different public keys. Due to its various applications as above, many researchers have developed PKEET schemes [9,11–13,16–18] for the purpose of achieving better performance and providing different levels of authorities for equality testing.

1.1. Our contribution

We provide a semi-generic PKEET construction that exploits traditional PKE schemes having sufficiently large plaintext spaces. Our PKEET system model follows that of Tang's all-or-nothing PKEET scheme [18]. In this scheme, each user issues a trapdoor to a specified tester and the authority to test the equality of all of his/her ciphertexts. Thus, the tester who has knowledge of the two user's trapdoors, is able to check the equality of ciphertexts using their public keys. We note that this model can also be regarded as a PKEET scheme supporting flexible authorization, where we only consider the authorization for equality test on all receiver's ciphertexts (so called Type-1 authorization in [12]).

In Section 1 of [18], Tang initially attempted to construct a generic PKEET scheme by defining an encryption algorithm for a message m by:

$$(\mathsf{C}_1, \mathsf{C}_2) = (\mathsf{PKE1}(pk_1, m), \mathsf{PKE2}(pk_2, \mathcal{H}_1(m)))$$

where PKE1 and PKE2 are traditional PKE schemes and $\mathcal{H}_1$ is a cryptographic hash function. Then, each user issues the secret key sk_2 for PKE2 to the tester and he/she can check their equality by decrypting the C_2's for both ciphertexts and then comparing $\mathcal{H}_1(m)$ values. Immediately, however, Tang demonstrated that the above formulation could not achieve the IND-CCA2 security [18] because an adversary could query

$$(\mathsf{C}_1^*, \mathsf{C}_2) = (\mathsf{PKE1}(pk_1, m_b), \mathsf{PKE2}(pk_2, \mathcal{H}_1(m_\beta)))$$

to the decryption oracle by guessing $b \in \{0, 1\}$, chosen by the challenger as β, and then generate the second component C_2 themselves where $(\mathsf{C}_1^*, \mathsf{C}_2^*)$ is the challenge ciphertext.

We resolve the above problem by providing a way to prevent decryption queries of the obtained ciphertexts by modifying the challenge ciphertext shown above. Our solution is as follows: Let $\mathbb{G}$ be a cyclic group with a generator g of prime order p, and $y = g^x$ is an additional public key for a randomly chosen element $x \in \mathbb{Z}_p^*$. In the encryption algorithm, r is randomly selected from $\mathbb{Z}_p^*$ and is used to compute g^r. Then, the algorithm attaches g^r to the message m and its hash value $\mathcal{H}_1(m)$. Their ciphertexts are generated using traditional PKE schemes. In addition, the algorithm provides the hash value of generated ciphertexts by attaching y^r. That is, our encryption algorithm for a message m is defined by

$$\mathsf{PKEET.Enc}(pk, m) = (\mathsf{PKE1}(pk_1, m\|g^r), \mathsf{PKE2}(pk_2, \mathcal{H}_1(m)\|g^r), \mathcal{H}_2(\mathsf{C}_1, \mathsf{C}_2, y^r))$$

where $\mathsf{C}_1 = \mathsf{PKE1}(pk_1, m\|g^r)$, $\mathsf{C}_2 = \mathsf{PKE2}(pk_2, \mathcal{H}_1(m)\|g^r)$, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are cryptographic hash functions. Informally speaking, an adversary must know y^r to generate a valid ciphertext by modifying the challenge ciphertext, where y^r is the solution to the Computational Diffie-Hellman (CDH) problem for the instance (g, g^r, y). We demonstrate that our semi-generic construction achieves one-wayness under adaptive chosen ciphertext attack (OW-CCA2) security against Type-I adversaries, who have a trapdoor information for the equality test, and the indistinguishability under adaptive chosen ciphertext attack (IND-CCA2) security against Type-II adversaries, who do not have that information. (See Section 2.1 for the details of types of adversaries.) Those are shown assuming that the exploited PKE schemes are IND-CCA2 secure and the CDH assumption holds in the random oracle model. Moreover, we attempt to interpret Tang's all-or-nothing PKEET scheme [18], which coincides with our system model, in the view of our semi-generic construction.

Our construction can be easily extended to the identity-based setting by replacing PKE schemes and the CDH assumption with traditional identity-based encryption (IBE) schemes and the bilinear Diffie-Hellman (BDH) assumption, respectively. Thus, our modified encryption algorithm for IBE with equality test (IBEET) is defined as follows: Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and set a public parameter $g_1 = g^s$ for a randomly chosen element s from $\mathbb{Z}_p^*$. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map defined over $\mathbb{G}$ and $\mathbb{G}_T$. We define an encryption algorithm of our semi-generic IBEET construction with an identity ID and a message m by

$$\mathsf{IBEET.Enc}(pp, \mathsf{ID}, m)$$