Future Generation Computer Systems 73 (2017) 22–31

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Efficient and secure identity-based encryption scheme with equality test in cloud computing

Libing Wu [a], Yubo Zhang [a], Kim-Kwang Raymond Choo [b], Debiao He [a,c,*]

[a] State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China
[b] Department of Information Systems and Cyber Security, University of Texas at San Antonio, USA
[c] Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Gulin, China

HIGHLIGHTS

- We propose an identity-based encryption with equality test (IBEET) scheme using bilinear pairing for cloud storage.
- We show that our proposed concrete scheme is provably secure. It satisfies required security properties.
- Detailed performance analysis and experimental result are given.

ARTICLE INFO

Article history:
Received 5 November 2016
Accepted 7 March 2017
Available online 20 March 2017

Keywords:
Identity-based encryption
Equality test
Searchable encryption
Cloud computing
Provable security

ABSTRACT

Efficient searching on encrypted data outsourced to the cloud remains a research challenge. Identity-based encryption with equality test (IBEET) scheme has recently been identified as a viable solution, in which users can delegate a trapdoor to the server and the server then searches on user outsourced encrypted data to determine whether two different ciphertexts are encryptions of the same plaintext. Such schemes are, unfortunately, inefficient particularly for deployment on mobile devices (with limited power/battery life and computing capacity). In this paper, we propose an efficient IBEET scheme with bilinear pairing, which reduces the need for time-consuming HashToPoint function. We then prove the security of our scheme for one-way secure against chosen identity and chosen ciphertext attacks (OW–ID–CCA) in the random oracle model (ROM). The performance evaluation of our scheme demonstrates that in comparison to the scheme of Ma (2016), our scheme achieves a reduction of 36.7% and 39.24% in computation cost during the encryption phase and test phase, respectively, and that our scheme is suitable for (mobile) cloud deployment.

1. Introduction

Due to the increasingly popularity of ubiquitous mobile devices and cloud computing, storing of data (e.g. photos, videos, emails and instant messages) in the cloud has become a trend among individual and organizational users. However, cloud service providers cannot be fully trusted to ensure the availability, confidentiality or integrity of user data outsourced/uploaded to the cloud (e.g. cloud servers may be corrupted and cloud employees may be malicious or "curious") [1,2].

Therefore, it is unrealistic to entrust sensitive data to a cloud server. A common practice is to encrypt the data prior to outsourcing. This, however, complicates searching on the (encrypted) data. Data owner would need to first download and decrypt the encrypted data prior to searching. This is both impractical and expensive (e.g. bandwidth consumption), particularly for large datasets. One viable solution is searchable encryption (SE), which allows untrusted servers to search on encrypted data on the behalf of data owners without the risk of data leakage [1]. Unsurprisingly, practical implementations of SE have been presented in the literature (see [3–13]), and one popular area of focus is keywords search using equality test.

To perform an equality test, an user first generates a trapdoor T_w for the plaintext keyword w, before sharing with the server S. Then, S can determine whether w is identical to w' by checking whether $f(w, T_w)$ is equal to $f(w', T_{w'})$, without learning any information about w or w'. Due to its potential to be used for searching on ciphertext, keywords search schemes with equality test can be applied in various practical application scenarios,

* Correspondence to: Computer School, Wuhan University, Wuhan, China.
E-mail addresses: whuwlb@126.com (L. Wu), cszyb@whu.edu.cn (Y. Zhang), raymond.choo@fulbrightmail.org (K.-K.R. Choo), hedebiao@163.com (D. He).

http://dx.doi.org/10.1016/j.future.2017.03.007
0167-739X/

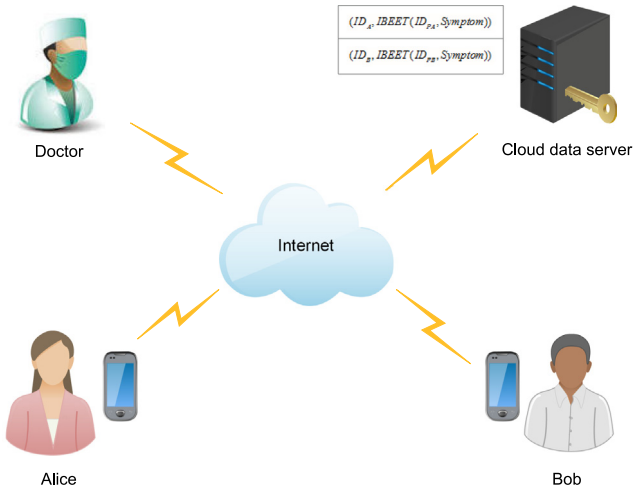

Fig. 1. A typical IBEET application scenario.

such as email classification, and distributed storage. Similarly, public key encryption with keyword search (PKEKS) has been the subject of recent research focus. For example, Yang [14] proposed a public key encryption scheme with equality test (PKEET) based on keyword search, which is designed to determine whether two ciphertext are encryptions of the same plaintext. However, this scheme does not scale well with a large number of users due to certificate management challenges. Thus, integrating with identity-based encryption (IBE) [15,16], Ma [17] proposed an IBEET scheme based on keywords search. To the best of our knowledge, this proposed scheme is the first to combine PKEET with identity-based encryption. However, Mas scheme is inefficient, especially for deployment on resource-limited mobile devices. Therefore, we build on the results reported by Ma [17], and specifically using Sakai–Kasahara IBE scheme [16] instead of Boneh–Franklin IBE [15]. We remark that despite the flaw in Boneh and Franklin's proof reported by Galindo [18], it does not affect the security proof of our scheme as the construction of the PKE in our security proof differs from that of Boneh and Franklin (see Section 5).

A typical IBEET application scenario is illustrated in Fig. 1. Other potential applications of IBEET include mobile healthcare social network (MHSN). In a MHSN, a user (e.g. patient) who wish to reach out or be connected to another user with the same symptoms can search on the data server of the healthcare service provider, without compromising their privacy (e.g. symptoms, location and ethnicity). For example, suppose that *Alice* and *Bob* are two patients having the same symptoms and they wish to find and communicate with each other to share their experience and encourage each other. *Alice* (with identity ID_A) encrypts her private data, say *symptom*, with her doctor's public identity ID_{DA} and then uploads the tuple $(ID_A, IBEET(ID_{DA}, Symptom))$ to the healthcare cloud server. Then, *Alice* generates a trapdoor td_A corresponding to *symptom* and sends it to the server as well. Similarly, *Bob* (with identity ID_B) uploads the tuple $(ID_B, IBEET(ID_{DB}, Symptom))$ and sends a trapdoor td_B to the server. Upon receiving these data, the healthcare cloud server could search and determine that both *Alice* and *Bob* have the same *symptom* because $IBEET(ID_A, Symptom)$ is identical to $IBEET(ID_B, Symptom)$, without knowing what the actual symptom is. Finally, the server sends the search result to *Alice* and *Bob*, which allows both of them to reach out to each other. Such a scheme can be extended to a multiple users setting. For example, one or more patients can generate and send a trapdoor along with the respective search requests to the cloud server and obtain feedback indicating whether there are any patient(s) on the system having the same symptoms.

1.1. Our contributions

Our contributions in this paper are three-fold.

(1) We propose a new IBEET scheme designed for mobile cloud (including MHSN) using bilinear pairing.
(2) We prove that our scheme is OW–ID–CCA secure under the ROM.
(3) We demonstrate that compared with the first published IBEET scheme of Ma [17], our scheme achieves a better performance in both the encryption and test phases. Thus, our scheme is more suitable for mobile cloud deployment.

1.2. Organization

The rest of this paper is organized as follows. Related work and some preliminaries are presented in Sections 2 and 3, respectively. In Section 4, we present the proposed IBEET scheme. The security analysis and performance evaluation are presented in Sections 5 and 6, respectively. We conclude the paper in Section 7.

2. Related work

2.1. Public key encryption with keyword search (PEKS)

The concept of public key encryption with keywork search was proposed by Boneh et al. [4]. Suppose that there are three participants, user Alice, user Bob and a mail server. Bob sends a email to Alice (with public key Alice.com) that encrypted with Alice.com. And Alice sends a trapdoor T to the mail server so that the server could search on the encrypted email for specific keywords. The keyword search does not reveal any information about the message in the email.

2.2. PKEET

The first PKEET was proposed by Yang et al. [14] which allows one to determine whether two ciphertexts encrypted under different public keys contain the same plaintext. A year later in 2011, Tang [19] proposed the FG-PKEET scheme, which combines a fine-grained authorization policy enforcement mechanism with PKEET. Subsequently, Tang [20] proposed an all-or-nothing public key encryption with equality test (AoN-PKEET). This is a coarse-grained authorization mechanism to specify who can perform a plaintext equality test from their ciphertexts. The author also presented a secure personal health record (PHR) application, where patients can outsource their encrypted PHRs to a third-party service provider.

Ma et al. [21] introduced a public key encryption with delegated equality test (PKE-DET) scheme, which allows equality tests to be delegated by the server with trapdoors offered by users. However, Huang et al. [22] pointed out that PKE-DET is inefficient due to too many bilinear map operations involved, and proposed the public key encryption with authorized equality test (PKE-AET) scheme as a solution. In PKE-AET, testers could only perform equality test on specific ciphertexts.

2.3. IBEET

The concept of IBEET was first proposed by Ma et al. [17] in 2016, which combined identity-base cryptosystem with PKEET. This allows equality tests to be carried out on one or more users' ciphertexts. However, Ma et al.'s scheme is inefficient due of the use of HashToPoint operation and a number of bilinear map operations. Both operations are time-consuming, and consequently, the scheme is not suitable for deployment on mobile or other lightweight devices.