Chapter 19
An End to End Cloud Computing Privacy Framework Using Blind Processing

Youssef Gahi
Ibn Tofail University, Kenitra, Morocco

Imane El Alaoui
Ibn Tofail University, Kenitra, Morocco

Mouhcine Guennoun
Cisco Systems, Toronto, Canada

ABSTRACT

Database-as-a-service (DBaaS) is a trend allowing organizations to outsource their databases and computations to external parties. However, despite the many advantages provided by this service in terms of cost reduction and efficiency, DBaaS raises many security issues regarding data privacy and access control. The protection of privacy has been addressed by several research contributions proposing efficient solutions such as encrypted databases and blind queries over encrypted data, called blind processing. In this latter context, almost all proposed schemes consider an architecture of a single user (the data owner) that requests the database server for encrypted records while he is the only one capable of decrypting. From a practical perspective, a database system is set up to support not only a single user but multiple users initiating multiple queries. However, managing various accesses to an encrypted database introduces several challenges by itself, like key sharing, key revocation, and data re-encryption. In this article, we propose a simple and efficient blind processing protocol that allows multiple users to query the same encrypted data and decrypt the retrieved results without getting access to the secret key.

DOI: 10.4018/978-1-7998-8954-0.ch019

INTRODUCTION

The cloud computing technology is a remote system that offers various remote services, such as Database-as-a-service, to allow companies and end-users to outsource their data and computations easily. Database-as-a-service (DBaaS) provides organizations with unlimited data storage. It is a cost-effective, cloud-based service that is characterized by easy deployment and higher availability. However, it also introduces new challenges to protecting data privacy. The amount of sensitive information stored in the cloud systems is increasing very quickly, and this information has to be protected from malicious access and processing. Although the cloud architecture has set up a set of techniques to preserve the confidentiality of data or to control access to them, the sensitive data need to be protected efficiently to have sufficient control over who can access these data. Homomorphic encryption schemes (HESs) have been proposed to allow executing operations over encrypted data without the need to decrypt them to get sensitive records.

HESs allow performing the logical operations XOR and AND over encrypted bit values without decryption (Gentry, 2010). Based on this type of scheme, it is possible to create processing systems and architectures through which the one can outsource only encrypted data and ask untrusted parties (maybe a cloud provider or a cloud server) to execute blind operations on its behalf without revealing data content. Applications based on this concept bring a high added value to the remote computation concept and blind processing techniques (Boneh et al., 2013; Gahi, Guennoun, and El-Khatib, 2011; Gahi et al., 2011; Gahi et al., 2012a; Gahi et al., 2012b; Raykova et al., 2012).

While data encryption and HES-based applications can add an important security layer, they also introduce several challenges that should be addressed. These challenges are related to key management and access policies. Almost all HES-based schemes only consider the case of a single user (the data owner) attempting to retrieve (encrypted) records from a database server (with the assumption that he/she is the only one capable of decrypting those records). In practice, however, a database system is usually queried by multiple users. Furthermore, current HES-based schemes assume that all users have the same access rights to the shared data.

The straightforward approach of transforming a single-user scheme to a multi-user one by sharing the decryption key among all users is a simple solution that suffers from a significant shortcoming. Clearly, with such an approach, it is quite challenging to determine the actual data requester since all requesters are using the same key. Moreover, if the transformation approach is to rely on a third party or key management systems, new problems related to key sharing, key distribution, and key revocation (which sometimes requires costly data re-encryption) will arise.

In this paper, we propose a secure processing architecture that allows multiple users to use various encryption/decryption keys to operate on the same encrypted, remote database. Our proposal is based on both the fully homomorphic encryption scheme and a blind decryption concept that we detail in this contribution. Our system is constituted of three entities, namely, a data owner, one or more clients, and a remote server. The data owner encrypts data using a public key and stores the encrypted records in a remote database (DBaaS). Then, clients hide the database queries and send them to the remote server for blind processing. The server receives encrypted queries and blindly checks the access rights to retrieve/modify the desired records. If the client has access rights, then the server will execute a blind query and extract encrypted items. These items will be transferred to a blind decryptor that blindly re-crypt, using a different public key, an already encrypted text (Gahi et al., 2016). This way, the client can successfully