Chapter 44
On Securing Cloud Storage Using a Homomorphic Framework

Daya Sagar Gupta
Indian Institute of Technology Dhanbad, India

G. P. Biswas
Indian Institute of Technology Dhanbad, India

ABSTRACT

In this chapter, a cloud security mechanism is described in which the computation (addition) of messages securely stored on the cloud is possible. Any user encrypts the secret message using the receiver's public key and stores it. Later on, whenever the stored message is required by an authentic user, he retrieves the encrypted message and decrypts it by using his secret key. However, he can also request the cloud for an addition of encrypted messages. The cloud system only computes the requested addition and sends it to the authentic user; it cannot decrypt the stored encrypted messages on its own. This addition of encrypted messages should be the same as the encryption of the addition of original messages. In this chapter, the authors propose a homomorphic encryption technique in which the above-discussed scenario is possible. The cloud securely computes the addition of the encrypted messages which is ultimately the encryption of the addition of the original messages. The security of the proposed encryption technique depends on the hardness of elliptic curve hard problems.

INTRODUCTION

To secure network communication, there exists two methods of cryptographic systems: symmetric key encryption and asymmetric key encryption. Till 1976, only symmetric-key cryptographic systems were known, in which both sender and receiver of an authentic message use the same key for encryption and decryption respectively. Some structures, like the one-time pad, attained information theoretic security, which indicates that the structure is protected even against adversaries with unlimited computing power.

DOI: 10.4018/978-1-7998-7705-9.ch044

However, information theoretic security includes a high economy in terms of the key size and need randomness. Additionally, symmetric-key cryptography requires substantial key-management.

Diffie & Hellman (1976) proposed an asymmetric key cryptosystem (also known as public-key encryption) known as the Diffie-Hellman (DH) key exchange protocol which could be used to securely exchange a symmetric key between two users. They also presented the conception of a trapdoor one-way function. The property of a one-way function f is that it is easy to calculate *f(x)* from *x*, but hard to calculate *x* from *f(x)*. These trapdoor functions become the backbone of asymmetric key cryptography. The idea of Public-key encryption is to generate a couple of keys that are mathematically associated with each other, comprising of a private key SK and a public key PK. The sender of a message uses the public key PK of the receiver to encrypt his message and the receiver then uses his private key SK to decrypt the incoming message. The public key is known to everyone and the corresponding private key is kept secret. The security of the corresponding private key depends on the hardness of one-way functions because finding the private key from the public key information is as hard as reversing a one-way function. Public key encryption (PKE) provides a key advantage over symmetric cryptography.

PKE plays an important role in the field of information security. This paper uses a PKE technique to ensure the security of the designed protocol. The proposed protocol is based on the homomorphic encryption. The idea of homomorphic encryption was first proposed by Rivest, Adleman & Dertouzous (1978) as a notion of *privacy homomorphism.* A public key encryption technique which includes the homomorphic property: $E(m_1\ o_M\ m_2) = E(m_1)\ o_C\ E(m_2)$ is termed as homomorphic encryption. In general, a PKE has three algorithms: *keyGen* which generates a pair of key (public key and private key), *encrypt* which encrypts the message using the public key and *decrypt* which decrypts the message using the private key. Homomorphic encryption also includes these three conventional algorithms with the inclusion of an efficient algorithm *evaluate* which takes cipher texts $c_1, c_2 \ldots\ldots c_n$ and public key as inputs and produces a valid encryption of some function f on messages $m_1, m_2 \ldots\ldots m_n$ i.e.

$$E(f_M(m_1, m_2 \ldots\ldots m_n)) \leftarrow f_C(c_1, c_2 \ldots\ldots c_n)$$

The proposed scheme is based on the elliptic curve cryptography (ECC). The elliptic curves play a very important role in the field of cryptography. The security of elliptic curve cryptography is much better than that of the RSA cryptosystem. Their scheme deals with the properties of an elliptic curve. The proposed protocol is based on the bilinear property designed for elliptic curves. ECC depends on the difficulty provided by elliptic curve operations like addition operation of the points on elliptic curves. Elliptic curve cryptography is nothing but a kind of PKE with a pair of keys i.e. secret and public keys. On the bilinear map, the Computing Diffie-Hellman Problem (CDHP) is difficult, but the Decision Diffie-Hellman Problem (DDHP) is easy. Miller (1985) and Kblitz (1987) independently proposed the security of elliptic curve cryptosystem algorithm which depends on the discrete logarithm problem of elliptic curves. A number of ECC-based cryptosystems like Nobelis et al. (2012), Gupta & Biswas (2017a), Gupta & Biswas (2017b), Gupta & Biswas (2017c), Munir & Mohammed (2012) etc. are proposed in the literature.

In this paper, the authors are going to present a cryptographic technique which is based on the difficulty of elliptic curve Diffie-Hellman problem and Bilinear Diffie-Hellman problem as discussed in Boneh D & Franklin (2001), Gupta & Biswas (2015a), Gupta & Biswas (2015c) and others. The main work is done to secure the cloud storage. To do so, the authors use the homomorphic encryption technique in their proposed work. To implement the proposed protocol, the authors use four algorithms*: keyGen,*