Privacy-Preserving Protocol in Multi-User Cloud

Somayeh Sobati Moghadam
Hakim Sabzevari University
Computer and Electrical Engineering Faculty
Hakim Sabzevari University
Sabzevar, Khorasan Razavi, Iran

ABSTRACT

Outsourcing data in the cloud environment has recently gained special attention. However, data privacy remains one of the top concerns for users in cloud outsourcing scenarios. In this paper, we propose a privacy preserving query protocol based on proxy re-encryption schemes. The proposed protocol protects data privacy of all users in a multi-user cloud setting. Data privacy is preserved against honest but curious adversary model.

General Terms

Data Privacy, Cloud Computing

Keywords

Privacy-Preserving Protocol, Data Privacy, Cloud Computing, Multi-User

Nowadays, cloud computing is bombing due to a large number of benefits such as availability, scalability, and elasticity [6]. As a result, individuals, businesses, and organizations outsource data in to the cloud. One of the most notable out-sourcing services is database outsourcing where organizations outsource the data storage and management to Cloud Service Provider (CSP) [17]. The most challenge in data outsourcing is storing sensitive data such as business secrets, credit card numbers or other sensitive personal information on the CSP's. Data confidentiality and privacy is the most concern because organization or individuals do not want to reveal their private data for various legal and competitive reasons [13].
Encryption provides privacy and confidentiality, and avoid any data revealing by semi-trusted CSP or other users and outsiders [16]. Considering the real world scenarios where each data owner (personal or business data owner), encrypts his data with his key, as a result, outsourced data are encrypted with different keys, which makes impossible to compute over encrypted data [4].

Multi User Cloud Outsourcing: In many real world cloud outsourcing there are *multi-users*, who wish to encrypt data and store them at the CSP [15]. In a single-user cryptographic systems, access control is straightforward, while in group sharing in cryptographic systems, users must rendezvous with data owner to obtain necessary privileges [2]. Traditional access control in such scenarios needs interception of the user for each reply from the server to filter out encrypted data that the final user cannot access [6] because it could not be delegated to the CSP who is not fully trusted. Additionally, the CSP could bypass access control to gain access to the stored data [1], which compromises data privacy. A Naive solution is to share a secret key among all users, which is not desirable because the most real world scenarios are non-interactive [4]. The approaches like Lopez et al. [12] are impractical while they need to execute a light multi party computation to decrypt the results. Additionally, user revocation is another problem, which is not scalable and need to re-execute a round of key distribution. On the other hand, each data owner encrypts his data with his key which are generated independently rather than generated from a common secret key [14].

Our Contribution: In this paper, we propose a novel privacy preserving protocol that anables a group of users to outsource encrypted data as well as computation tasks completely to a cloud service provider (CSP). Encrypted data are re-encrypted by a trusted proxy server and stored at the CSP's. Computation are computed by the CSP over re-encrypted ciphertexts without revealing the underlying plaintexts. The result of computations are sent to the proxy server and are re-encrypted in a such a way that can be decrypted by the user who has access to the private key.
The reminder of this paper is organized as follows. Section 1 presents system model, security model, and our goals. Some preliminaries are introduced in Section 2. The proposed protocol is described in Section 3. Finally, Section 4 concludes the paper along with the scope of future researches.

1. MODELS AND ASSUMPTIONS

1.1 System Model

Considering a scenario where n data owners want to outsource their sensitive data as well as the query processing on their combined data to a cloud environment. We assume that the system is composed of the following parties:
(i) *Users:* All users are *honest-but-curious* They wish to compute over encrypted shared data and get the final results without disclosing the privacy of other user's data, (ii) A *Cloud Service Provider (CSP)*: is a honest-but-curious cloud provider, although honest but it try to legally infer more about private data of any user, and (iii) A trusted *Proxy:* interact between users and the CSP in order to pre/post processing. Having outsourced data, The main goal of protocol is to enable any authorized user to perform outsource data and query over the combined encrypted data in a privacy preserv-

ing manner [?]. All data user can query encrypted data and get the results from his/her data that already been stored in the cloud along with all shared data from other users. They register with the proxy by providing some kind of information [11].

1.2 Security Model

In our setting we consider *honest-but-curious* CSP, which means the CSP follows the designed protocol correctly but may attempt to infer more about data [10]. The CSP does not collude with users or the proxy server. All communications between the CSP and the proxy server are considered to be secured via secure protocols such qs SSL [9]. All users and the proxy server are trusted [5]. The users do not collude between them or with the proxy server [8].

1.3 Goals

The main goal is achieving data privacy on shared data. Specifically, we want to enable the users to compute over encrypted shared data stored at the CSP [7]. The proposed protocol prevents the CSP from being able to learn plaintexts from stored data. Computations are processed over encrypted data without decryption.

2. PRELIMINARIES

In this section we introduce the cryptographic primitives which are used in our protocol.

2.0.1 Proxy Re-encryption. The notation of proxy re-encryption introduced by Blaze Bleumer, and Strauss (BBS) in [3]. The objective here is to transform a ciphertext of Alice to ciphertext of Bob without revealing decryption keys or clear text. The BBS approach introduces a *re-encryption* key $RK_{A\to B}$, which allows a trusted *proxy server* to re-encrypt a ciphertext from the secret key sk_A to the secret key sk_B without learning the plaintext. The BBS scheme is based on ElGamal and uses a group $\mathbb{G}$ of prime order q, with generator g. The BBS scheme consists of four algorithms $\Pi_{BBS} = \{KG, Enc, ReEnc, Dec\}$ which are defined as follows:

—KG: **Key Generation**

Choose a random value a from $\mathbb{Z}_q^*$, $sk_A = a$ and $pk_A = g^a$

Choose a random value b from $\mathbb{Z}_q^*$, $sk_B = b$ and $pk_B = g^b$

Set $Rk_{A\to B} = b/a \quad mod \quad q$

—Enc: **Encryption**

Choose a random value r from $\mathbb{Z}_q^*$

For a plaintext value m

—Compute $\alpha_m = g^r.m$

—Compute $\beta_m = g^{ar}$

—set $C_A = Enc(m) = (\alpha_m, \beta_m)$

—$ReEnc$: **Re-encryption**

For C_A set $ReEnc(C_A) = (g^r.m, \quad (g^{ar})^{Rk_{A\to B}})$

$$= (g^r.m, \quad (g^{ar})^{b/a})$$

$$= (g^r.m, \quad g^{br})$$

$$= C_B$$

—Dec: **Decryption**

For ciphertext $C_A = (\alpha_m, \beta_m)$ compute $\dfrac{\alpha_m}{(\beta_m)^1/a} = \dfrac{g^r.m}{(g^{ar})^{1/a}}$

As a result, Bob can decrypt ciphertexts on behalf of Alice [11]. The proxy function could be: *unidirectional* or *bidirectional*. In *bidirectional* proxy function both Alice and Bob can decrypt the ciphertext of each other using the *same bidirectional* proxy function while in *unidirectional* they need to use completely different functions. The primitive of uni/bidirectional encryption can be considered as a specific case of threshold cryptography, which a threshold is defined to decrypt a ciphertext. Each user encrypts his data and sends them to the CSP, after each time he can send queries over all data consist of other data owners 's data and his data, without disclosing data privacy.

3. PRIVACY PRESERVING QUERYING PROTOCOL (PPQP)

3.1 Mail Idea

The problem of privacy preserving aggregation over encrypted data in an outsourced environment was addressed in [12], however, such solutions was proposed under a single -user setting. In this paper, we propose an efficient and novel $PrivacyPreservinfProtocol(PPP)$ that enables a group of data owner to outsource their sensitive data to a Cloud Service Provider (CSP). In the proposed protocol, the sensitive data of users should never be revealed to other users or the CSP. Moreover, the proposed protocol incurs lightweight computation overhead at the user's. Since the purpose of data outsourcing is to shift the computation to cloud as much as possible, hence all computations are computed at the CSP's.

3.2 Different Phases

The proposed protocol consists of the following phases.

3.2.1 Initialization Phase. The proxy enhances key generation algorithm and generates a set of public and private key for each user and sends to the users. Once at initialization, the proxy server also generates a key pairs (pk_{CSP}, sk_{CSP}) for encryption data at the CSP side, while pk_{CSP} is public key and sk_{CSP} is secret key. The secret key sk_{CSP} remains at the proxy and never reveals to any users or the CSP and pk_{CSP} is send to all users.

3.2.2 Querying Phase. When a user has a query, Q, first he encrypts all constants in the query by his public key and generates Q_i (we show this step by $Q \xrightarrow{U_i} Q_i$). The user sends Q_i to the proxy server. The proxy server re-encrypts Q_i and generates Q_{CSP} ($Q_i \xrightarrow{proxy} Q_{CSP}$) and sends it to the CSP. Upon receiving Q_{CSP}, the CSP executes and the results sends the results to the proxy.

Remember that the results are encrypted under pk_{CSP} (we show it as $e_{CSP}(Res)$, i.e., the results encrypted by the CSP key), thus the user cannot decrypt them. The CSP is unable to re-encrypt the results because he doesn't access to sk_{CSP}. The results are sent to the proxy for re-encrypting into user ciphertext $e_{U_i}(Res)$ ($e_{CSP}(Res) \xrightarrow{proxy} e_{U_i}(Res)$) such that the user could decrypt and see the results [11].

3.3 The Proposed Protocol

Consider a set of users $\langle U_1, ..., U_n \rangle$ have a set of sensitive data $\langle T_1, ..., T_n \rangle$ where T_i belongs to the user U_i. The users wish to share data and compute some computation over data without learning sensitive data by the other users. All data would be encrypted with a unique key, which the secret key is unknown for the users. The proposed protocol is called *Privacy Preserving Querying Protocol* $PPQP$, which is defined as follow:

Privacy Preserving Querying Protocol $PPQP$:

Initialization Phase:

—Choose a random value b from $\mathbb{Z}_q^*$

—Set $sk_{CSP} = b$ and $pk_{CSP} = g^b$

—For $i = 1, \ldots, n$:

—Choose a random value a_i from $\mathbb{Z}_q^*$

—Set $sk_i = a_i$ and $pk_i = g^{a_i}$

—Set $Rk_{U_i \longrightarrow CSP} = b/a_i \mod q$

—Send sk_i, pk_i, and $Rk_{U_i \longrightarrow CSP}$ to the user U_i

Sending data:

User U_i:

—Choose a random value r from $\mathbb{Z}_q^*$

—For plaintext value T_i compute $\alpha_i = g^r.T_i$ and $\beta_i = g^{a_i r}$

—Set $C_{U_i} = (\alpha_i, \beta_i)$

—send C_{U_i} to the proxy server

Proxy:

—Compute $\beta_{CSP} = (\beta_i)^R k_{U_i \longrightarrow CSP} = (g^{a_i r})^{b/a_i} = g^{br}$

—Set $C_{U_{CSP}} = (\alpha_{CSP}, \beta_{CSP}) = (\alpha_i, \beta_{CSP})$

—Send $C_{U_{CSP}}$ to the CSP

Querying data:

User U_i:

The user encrypts each plaintext value in the query Q like the "*sending data- user*" process and generates Q_i. Then the user sends Q_i to the proxy.

Proxy:

The proxy server encrypts all values in Q_i like the "*sending data- proxy*" phase and generates Q_{CSP}. Then the proxy sends Q_{CSP} to the CSP.

Decryption:

The CSP computes Q_{CSP} and sends back the encrypted results, $enc_{CSP}(Res) = (\alpha_{CSP}, \beta_{CSP})$ to the proxy.

Proxy:

—Compute $\beta_i = (\beta_{CSP})^{Rk_{CSP \longrightarrow U_i}} = (g^{br})^{a_i/b}$

—Set $\alpha_i = \alpha_{CSP}$

—Send α_i, β_i to the user U_i

User U_i:

—Compute $\dfrac{\alpha_i}{(\beta_i)^{1/a_i}}$

3.4 Discussion

The encryption scheme used in PPQP is an extension of ElGamal public key encryption. Hence, the security of PPQP offers the same, i.e., PPQP is secure against *Chosen Plaintext Attack (CPA)*. Informally, a cryptosystem is CPA secure if an adversary cannot distinguish between the encryption of two plaintexts m_1 and m_2. The CSP or an outsider attacker who has access to encrypted data cannot learn plaintexts without access to secret keys and re-encryption keys used by the proxy server.

Computational overhead is analysed in terms of the number of modular exponentiation (we denote it by c_e). The PPQP incurs $(2c_e)$ for encryption at the user's, c_e for re-encryption at the proxy's, and c_e for decryption at the user's. Considering a plaintext m of size $|m|$, the proposed protocol introduces ciphertexts of size $2|\mathbb{Z}_q^*|$.

4. CONCLUSION

I this paper, we proposed a privacy preserving protocol, which allows the users to securely outsource data and execute computation at a CSP. The proposed protocol provides data privacy using secure cryptosystem. A trusted proxy server re-encrypts ciphertexts in such a way that computations can be carried out over encrypted data. Ciphertexts can only be decrypted by the user who has access to the private key.
As future work, we plan to implement the proposed protocol and analyse the efficiency and overhead of the protocol. Moreover, we aim at reducing computational and storage overhead at both the user's and the proxy server's.

5. REFERENCES

[1] Muhammad Rizwan Asghar, Giovanni Russello, Bruno Crispo, and Mihaela Ion. Supporting complex queries and ac-

cess policies for multi-user encrypted databases. In *Proceedings of the 2013 ACM Workshop on Cloud Computing Security Workshop*, CCSW '13, pages 77–88, New York, NY, USA, 2013. ACM.

[2] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, February 2006.

[3] Matt Blaze and Martin Strauss. Atomic proxy cryptography. Technical report, Proc. EuroCrypt '97, 1998.

[4] Fangquan Cheng, Qian Wang, Qianwen Zhang, and Zhiyong Peng. Highly efficient indexing for privacy-preserving multi-keyword query over encrypted cloud data. In *Web-Age Information Management*, pages 348–359. Springer, 2014.

[5] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Keep a few: Outsourcing data while maintaining confidentiality. In *Computer Security - ESORICS 2009, 14th European Symposium on Research in Computer Security, Saint-Malo, France*, pages 440–455, 2009.

[6] Ernesto Damiani, S. De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Key management for multi-user encrypted databases. In *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*, StorageSS '05, pages 74–83, New York, NY, USA, 2005. ACM.

[7] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Encryption policies for regulating access to outsourced data. *ACM Trans. Database Syst.*, 35(2), 2010.

[8] Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, Gerardo Pelosi, and Pierangela Samarati. Efficient and private access to outsourced data. In *2011 International Conference on Distributed Computing Systems, ICDCS 2011, Minneapolis, Minnesota, USA, June 20-24, 2011*, pages 710–719, 2011.

[9] Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, and Pierangela Samarati. privacy of outsourced data. In *Digital Privacy: Theory, Technologies, and Practices*, pages 382–405, 2007.

[10] Sara Foresti. *Preserving privacy in data outsourcing*, volume 99. Springer Science & Business Media, 2010.

[11] Anca Ivan and Yevgeniy Dodis. Proxy cryptography revisited. In *in Proceedings of the Network and Distributed System Security Symposium (NDSS*, 2003.

[12] Adriana Lopez-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. Cryptology ePrint Archive, Report 2013/094, 2013. `http://eprint.iacr.org/2013/094`.

[13] Raluca A. Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles 2011, SOSP 2011, Cascais, Portugal, October 23-26, 2011*, pages 85–100, 2011.

[14] Raluca A. Popa and Nickolai Zeldovich. Multi-key searchable encryption. *IACR Cryptology ePrint Archive*, 2013:508, 2013.

[15] Raluca Ada Popa and Nickolai Zeldovich. Multi-key searchable encryption. Cryptology ePrint Archive, Report 2013/508, 2013. `http://eprint.iacr.org/`.

[16] Stephen Tu, M. Frans Kaashoek, Samuel Madden, and Nickolai Zeldovich. Processing analytical queries over encrypted data. *Proceedings of the VLDB Endowment, PVLDB*, 6(5):289–300, 2013.

[17] Li Xiong, Subramanyam Chitti, and Ling Liu. Preserving data privacy in outsourcing data aggregation services. *ACM Trans. Internet Technol.*, 7(3), August 2007.