ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 6, June 2017

An Efficient Privacy-Preserving Communicating with Multiple Keys Using Assessment toolkit

[1] Swetha K B, [2] Harshitha M P, [3] Manoj M, [4] VickySamson, [5] Shikar joshi
[1]Assistant Professor, [2][3][4][5]Student of 8th semester
[1][2][3][4] Department of Electronics and Communication Engineering, R R Institute of Technology, Chikkabanavara, Bengaluru-560090

***Abstract*— In this paper, we propose a toolkit for efficient and privacy-preserving outsourced calculation under multiple encrypted keys (EPOM). Using EPOM, a large amount of users can safely outsource their information to a cloud server for storage. Moreover, encrypted data which belongs to different users can be processed without compromising on the security of the individual user's (original) data and the final computed results. To avoid the associated key management cost and private key exposure risk in EPOM, we use a distributed two-trapdoor public-key cryptosystem, the core cryptographic primitive. We also present the toolkit to make sure that the commonly used integer operations can be securely handled across different encrypted domains. We then prove that the proposed EPOM achieves the goal of secure integer number processing without resulting in privacy leakage of data to unauthorized parties. Last, we demonstrate the utility and the efficiency of EPOM using simulations.**

Index terms— Privacy-preserving, homomorphic encryption, outsourced computation, multiple keys.

I. INTRODUCTION

CLOUD computing due to support real-time and massive support real-time and massive storing and processing of data, is increasingly used in domains such as Inter-net of Things (IoT), e-commerce, and scientific research. It is, therefore, unsurprising that cloud computing is considered a viable solution to address the demands due to a significant increase in storage media and the number of digital and Internet-connected devices (e.g. Internet of Things and medical devices). For example, in 2011, the U.S Federal Government adopted a Cloud First policy which requires government agency's Chief Information Officers to implement a cloud-based service whenever there was a secure, reliable, and cost-effective option. Despite the benefits afforded by the use of cloud computing, data security and privacy remain areas of ongoing focus. For example, in the final US Government Cloud Computing Technology Roadmap published by the National Institute of Standards and Technology (NIST), security and privacy are considered one of the high-priority requirements, and a number of dedicated cloud computing research labs, such as and, have been established in recent years.

In attempts to conserve resources, reduce operational costs, and maintain efficiency, cloud service providers often store data belonging to multiple users on the same server (i.e. multi-tenancy). Therefore, different users should be distributed with an individual key (i.e. multiple keys, a.k.a, and multikey), to avoid multi-tenancy related attacks (e.g. a users private data viewed by other unauthorized users). One application of the multikey setting is e-healthcare cloud, where patients can transmit and store their health related information (e.g. patient's heart rate, blood pressure and glucose levels) on the hospitals cloud servers. This will facilitate diagnosis of the patient's physical condition based on the information. It is, however, important to ensure the security and privacy of patient's health and other personally identifiable information (PII), such as health status. The privacy of decision making model used is also considered by the e-health service provider as a trade secret. One way to achieve the security and privacy of the data is to issue all users (e.g. patients and service provider) different (unique) keys. In addition, an e-health service provider uses patient's health and PII (encrypted under different keys) in their training decision model. For example, historical medical data are used to train Nave Bayesian classifier in Clinical Decision Support System (CDSS). However, achieving secure calculation over the data under multiple keys without comprising the privacy of individual data remains a hard problem.
In this paper, we propose an Efficient Privacy-preserving Outsourced calculation framework with multiple keys

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 6, June 2017

(EPOM) to address the above-mentioned challenge. We regard the contributions of this paper to be four-fold, namely:

A. Our proposed EPOM is designed to allow different data providers to outsource their data (e.g. data belonging to users from different data providers) to the cloud server for secure storage and processing.

B. We construct a new cryptographic primitive, Distributed Two Trapdoors Public-Key Cryptosystem (DT-PKC), which is deployed in EPOM to split a strong private key into different shares. This will allow us to reduce the risk of private key leakage and private key management cost in a multi-key setting.

C. We build a privacy-preserving outsourced calculation toolkit of integer numbers with multiple keys. The toolkit consists of commonly used elementary operations, such as multiplication, division, comparison, sorting, sign bit acquisition, equivalence testing and greatest common divisor. The extension of the toolkit can also securely store and process real numbers.

D. We then demonstrate the utility of EPOM using a purposefully built simulator in Java, which demonstrates that our proposal can effectively and securely outsources the storage and process of data in a multiple keys setting.

ARCHITECTURE:

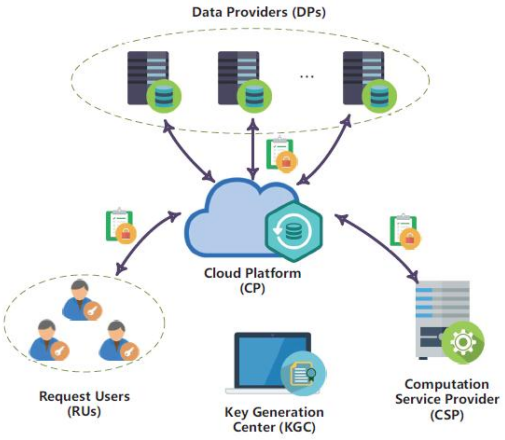

Fig.1

SYSTEM ANALYSIS

A. existing system:

Single key and multiple keys fully homomorphic cryptosystem in the existing scheme are rather inefficient, in terms of computation and storage. In the near future, if an efficient multi-key fully homomorphic cryptosystem exists; we can remove the CSP from the system which will also result in a more elegant system.

B. proposed system:

Proposed a new efficient and privacy preserving outsourced calculation framework with multiple keys. The framework is designed to allow different data providers to securely outsource their data with their own public key, and for a cloud server to process the multi-key encryption data on-the-fly. Proposed DT-PKC, the length of N will affect the running time of the proposed Cryptosystem. We observe that both run time and communication overhead of the basic algorithms increase with N. This is because run time of the basic operations (modular multiplication and exponent) increases as N increases, resulting in the transmission of more bits. For the toolkits of integer protocols, two factors will affect the performance.

II. PRELIMINARY

Let (xμ−1, • • •, x0) denotes the binary representation of x, where x0 and xμ−1 are the least and most significant bits, respectively. The goal of SBD is to convert the encryption of x into the encryption of the individual bits of x , without disclosing any information regarding x to both parties. More formally, we define the SBD protocol as follows:

([xμ−1] pk, • • •, [x0] pk) ← SBD ([x] pk).

In this section, we outline the notations used in the paper. We also define the Additive Homomorphic Cryptosystem (AHC) and Secure Bit-Decomposition (SBD) Protocol, which are the building blocks in the proposed EPOM.

Throughout the paper, we use pki and ski to denote the public key and weak private key of party, respectively. Pk denotes the joint public key (see Section IV for the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 6, June 2017

construction), SK denotes the system strong private key, and SK (1) and SK (2) denote the partial strong private keys. Furthermore, we denote [x] pki as the encrypted data x under pki, Dsk (•) as the decryption algorithm using SK, L(x) as the bit-length of x, and |x | to represent the absolute value of x.

A. additive homomorphic cryptosystem

Suppose [m1] pk and [m2] pk are two additive homomor-phic cipher texts under the same public key pk in an additive homomorphic cryptosystem. The additive homomorphic cryptosystem (e.g. Paillier cryptosystem and Benaloh cryptosystem [16]) has the additive homomorphism property:
Dsk ([m1] pk • [m2] pk) = m1 + m2.

B. secure bit-decomposition protocol (sbd)

Suppose that there are two parties in the protocol, Alice and Bob. Bob holds the AHC encrypted value [x] pk, where $0 \leq x < 2\mu$ and μ is the domain size of x in bits. We also remark that x is known to neither Alice nor Bob.

III. SYSTEM MODEL & PRIVACY REQUIREMENT

In this section, we formalize the EPOM system model; outline the problem statement, and define the attack model.

A. system model

In our system, we mainly focus on how the cloud server responds to user request in a privacy-preserving manner. The system comprises a Key Generation Center (KGC), a Cloud Platform (CP), a Computation Service Provider (CSP), Data Providers (DPs) and Request Users (RUs) see Fig. 1.

1. KGC: The trusted KGC is tasked with the distribution and management of both public and private keys in the system.

2. DP: Generally, a DP will use its public key to encrypt some data, before storing the encrypted data with a CP.

3. CP: A CP has Unlimited data storage space, and stores and manages data outsourced from all registered RUs. A CP also stores all intermediate and final results in encrypted form. Furthermore, a CP is able to perform certain calculations over encrypted data.

4. CSP: A CSP provides online computation services to users. The CSP is also able to partial decrypt cipher texts sent by the CP, perform certain calculations over the partial decrypted data, and then re-encrypt the calculated results.

5. RUs: The goal of a RU is to request a CP to perform some calculations over the encrypted data under multiple keys.

After the calculation has been performed, the result can be decrypted by RU upon successful authentication.

B. problem statement

1) Secure Outsourced Storage: As the cloud storage service is often provided by third-party servers who may be untrusted or semi-trusted, it is important for DP to outsource the data to the cloud without compromising its own privacy.

2) Secure Processing Toolkit for Integer: In order to achieve data processing on-the-by, the encrypted integer calculation toolkit needs to be built to support commonly used integer number operations over the plaintext. For example, additions, multiplications and divisions should be achievable by operating on two encrypted numbers.

3) Secure Processing under Multiple Keys: In order to support outsourced data processing across different parties, a multi-key data calculation mechanism (e.g. comparison of encrypted numbers under different public keys) needs to be constructed. Moreover, as the final result contains information belonging to different parties, fine-grained authentication mechanisms should be designed to guarantee the privacy of individual DP.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 6, June 2017

C. attack model

In our attack model, we consider the KGC to be a trusted entity, which generates the public and private keys for the system. On the other hand, RUs, DPs, CP and CSP are curious-but-honest parties, which strictly follow the proto-col. However, RUs, DPs, CP and CSP are also interested to learn data belonging to other parties. Therefore, we introduce an active adversary A∗ in our model. The goal of A∗ is to decrypt the challenge DPOs original cipher text and the challenge RUos encrypted final results with the following capabilities:

1) A∗ may eaves drop all communications to obtain the encrypted data.

2) A∗ may compromise the CP to guess the plaintext value of all cipher texts outsourced from the DPs (including the challenge DPs), and all cipher text sent from the CSP by executing an interactive protocol.

3) A∗ may compromise the CSP to guess the plaintext value of all cipher texts sent from the CP by executing an interactive protocol.

4) A∗ may compromise one or more RUs and DPs, with the exception of the challenge RU or challenge DP, to obtain access to their decryption capabilities, and guess all cipher texts belonging to the challenge RU or challenge DP.

The adversary A∗ is, however, restricted from compromising (1) both the CSP and the CP concurrently, (2) the challenge DP, and (3) the challenge RU. We remark that such restrictions are typical in adversary models used in cryptographic protocols.

IV. PRIVACY PRESERVING INTEGER CALCULATIONTOOLKIT FOR MULTIPLE KEYS

After introducing the underlying algorithms in DT-PKC, we will now present the secure sub-protocols as the toolkit for processing integers, namely: Secure Addition Protocol across Domains (SAD), Secure Multiplication Protocol across Domains (SMD), Secure Sign Bit Acquisition Protocol (SSBA) Secure Less Than Protocol (SLT), Secure Maximum and Minimum Sorting Protocol (SMMS), Secure Equivalent Testing Protocol (SEQ), Secure Division Protocol (SDIV) and Secure Greatest Common Divisor Protocol (SGCD). We assume that both CP and CSP will be involved in the sub-protocol, as the CP holds a partial strong private key SK (1), and the CSP has the remaining partial strong private key SK (2) and public key pk_. Note that both x, y involved in the above sub-protocols are integer (i.e. x, y can be positive, negative or zero); therefore, we restrict |x | and |y | to be in the range of [0, R1], where L (R1) < L (N)/8. If a larger plaintext range is needed, we can simply use a larger N. A larger N implies a broader plaintext range, and therefore, a higher level of security. However, this will affect the efficiency of DT-PKC.

ALGORITHM

A. encryption algorithm:

A mathematical procedure for performing encryption on data. Through the use of an algorithm, information is made into meaningless cipher text and requires the use of a key to transform the data back into its original form.

B. decryption algorithm:

Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand. This term could be used to describe a method of un-encrypting the data manually or with un-encrypting the data using the proper codes or keys.

C. polynomial time algorithm:

An algorithm that is guaranteed to terminate within a number of steps which is a polynomial function of the size of the problem. See also computational time complexity. Search the data without loss of time to provide out stream for the process.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 6, June 2017

V. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of EPOM.
Experiment analysis
The computation cost and communication overhead of the proposed EPOM were evaluated using a custom simulator built in Java, and the evaluations were performed on a personal computer (PC) with 3.6 GHz eight-core processor and 12 GB RAM memory.

A. Basic Crypto Primitive & Protocols' Performance:

We first evaluate the performance of our basic cryptographic primitive and toolkit for integer on our PC tested. See Tables II and III, respectively. We denote N as 1024 bits to achieve 80-bit security levels. We then use a Smartphone with eight-core processor (4×Cortex-A17 + 4×Cortex-A7) and 2 GB RAM memory to evaluate the performance of the basic crypto primitive see Table II. The evaluations demonstrated that the algorithms in DT-PKC are suitable for both PC and Smartphone deployments. Note that the toolkit for integer number calculations is designed for outsourced computation; therefore, they are only evaluated in the PC tested.

B. Factors Affecting Protocols' Performance:

For our pro-posed DT-PKC, the length of N will affect the running time of the proposed cryptosystem. We observe that both run time and communication overhead of the basic algorithms increase with N. This is because run time of the basic operations (modular multiplication and exponent) increases as N increases, resulting in the transmission of more bits. For the toolkit of integer protocols, two factors will affect the performance, namely: (i) length of N (for all protocols), and (ii) domain size of the plaintext (for SBD, SDIV, and SGCD), we observe that both computational and communication costs of all protocols increase with N, as the protocols rely on the basic DT-PKCurcing.

TABLE I
THE PERFORMANCE OF DT-PKC (1000-TIME ON AVERAGE, 80-BIT SECURITY LEVEL)

Algorithm	Enc	CR	PDec	WDec	PSDec1	PSDec2	PWDec1	PWDec2
PC Run Time	16.408 ms	16.275 ms	8.361 ms	8.432 ms	23.135 ms	23.248 ms	8.257 ms	8.799 ms
Smartphone Run Time	89.671 ms	90.643 ms	47.043 ms	50.651 ms	135.130 ms	130.712 ms	45.675 ms	57.240 ms

TABLE II
PERFORMANCE OF SUB-PROTOCOL (1000-TIMES FOR AVERAGE, 80-BITS SECURITY LEVEL)

Protocol	CP compute.	CSP compute.	Commu.
SAD	124.913 ms	61.420 ms	1.998 KB
SMD	340.479 ms	141.226 ms	4.491 KB
SBD (10-bits)	0.969 s	0.396 s	14.997 KB
SSBA	459.936 ms	185.559 ms	5.741 KB
SLT	192.226 ms	96.237 ms	3.244 KB
SEQ	1.006 s	0.439 s	15.485 KB
SMMS	1.054 s	0.459 s	16.219 KB
SDIV (10-bits)	9.263 s	3.742 s	132.039 KB
SGCD (10-bits)	102.675 s	42.899 s	1.446 MB

And basic operations. We observe that SBD, and the computational cost and the communication overhead in both SDIV and SGCD increase with the plaintext bit length. This is due to the increase in encrypted data which consumes more computation and communication resources. Next, we present the theoretical analysis for EPOM.

VI. RELATED WORK

With the constant evolution of cloud and related technologies, more users choose to encrypt before outsource their own data to cloud servers for storage. However, it is important to ensure the security and privacy of outsourced data. While homomorphic encryption technique allows searching of encrypted data, it is not yet practical to do so. More specifically, Gentry constructed the first fully homomor-phic encryption scheme based on lattice-based cryptography to support an arbitrary number of addition and multiplication operations. Since the seminal work of Gentry in 2009, a number of single-key fully homomorphic encryption schemes and multi-key fully homomorphic encryption schemes had been proposed. However, one of the biggest drawbacks of fully homomorphic cryptosystems is complexity in both computation (including encryption and decryption) and storage (including both public/private key size and cipher text size). It is not yet practical to implement fully homomorphic cryptosystem in the real-world.
Partial homomorphic encryptions (including additive and multiplicative homomorphic encryption) are often considered the next best solution. However, partial

homomorphic encryptions can only handle one kind of homomorphic operation with arbitrary times. Additive homomorphic encryption scheme, such as Paillier cryptosystem and Benaloh cryptonymsytem, allows other parties to securely perform some additive homomorphic calculations over the cipher text. Multiplicative homomorphic encryption scheme, such as unpadded RSA cryptosystem and El-Gamal cryptosystem, allows some multiplication over the plaintext. In recent years, some cryptosystems attempt to provide for both additive and multiplicative operations. However, these systems generally achieve only limited numbers of homomorphic operations.

For example, the BGN cryptosystem can only support limited numbers of additive homomorphic operations and only one multiplicative homomorphic operation.

VII. CONCLUSION

In this paper, we proposed a new efficient and privacy-preserving outsourced calculation framework with multiple keys. The framework is designed to allow different data providers to securely outsource their data with their own public key, and for a cloud server to process the multi-key encryption data on-the-ßy. To ensure that the scheme can be deployed in a real-world application, we proposed a new cryptographic primitive, Distributed Two Trapdoors Public-Key Cryptosystem (DT-PKC), to reduce both key management cost and private key exposure risk. We also built toolkit to per-form privacy preserving calculations to handle commonly used integer operations in a privacy preserving way. Our evaluations demonstrated that our framework (and the underlying building blocks) is sufficiently efficient for a real-world deployment.

ACKNOWLEDGMENT

The authors thank the associate editor and the anonymous reviewers for their constructive and generous feedback.

REFERENCES

[1] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, toward efficient and privacy-preserving computing in big data era IEEE Network., vol. 28, no. 4, pp. 46Ð50, Jul/Aug. 2014.

[2] Q. Do, B. Martini, and K.-K. R. Choo, ÒA forensically sound adversary model for mobile devices PLoS ONE, vol. 10, no. 9, p. e0138449, 2015.

[3] X. Liu, B. Qin, R. Deng, and Y. Li, An efficient privacy-preserving outsourced computation over public data IEEE Trans. Service Computer, to be published.

[4] Cloud Computing, accessed on 2016. [Offline]. Available: https:// en. wikipedia. org/ wiki/ Cloud_c omputing

[5] University of Melbourne. The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, accessed on 2015. [Online]. Available: http://www.cloudbus.org/

[6] D. Catalano and D. Fiore, Boosting linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data IACR Cryptol. EPrint Archive, Oct. 2014. [Online]. Available: http://eprint.iacr.org/2014/813

[7] Z. Brakerski and V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) LWE,Ó SIAM J. Computer., vol. 43, no. 2, pp. 831Ð871, 2014.

[8] P. Mukherjee and D. Wichs, Two round multiparty computation via multi-key FHE,Ó IACR Cryptol. EPrint Archive, Apr. 2015. [Online]. Available: http://eprint.iacr.org/2015/345