International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 6, December 2016, pp. 2994~3000
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i6.11972

A Cloud Based Secure Voting System using Homomorphic Encryption for Android Platform

Manish Ranjan, Ayub Hussain Mondal, Monjul Saikia
Department of Computer Science and Engineering, North Eastern Regional Institute of Science and Technology, Nirjuli, Arunachal Pradesh, India

Article Info

Article history:

Received Jul 26, 2016
Revised Nov 8, 2016
Accepted Nov 22, 2016

Keyword:

Android plateform
Cloud computing
Cryptography
Homomorphic cryptosystem
Security and privacy

ABSTRACT

Cloud based service provider are at its top of its services for various applications, as their services are very much reachable from anywhere anytime in current days. It is responsibility of the company that the Cloud storage is owned and maintained by themselves keeping the data available and accessible, and the physical environment protected and running. Could storage provider seem to be uncertain of confidentiality in many cases, as we need to limit ourselves on trust to a third party. Keeping our sensitive data ready to access any time anywhere with preventing any information leakage is a challenging task. Cryptography in this scenario plays an important role, providing security for information to protect valuable information resources on intranets, Internet and the cloud. In addition, Homomorphic cryptosystem is a form of Cryptography where some specific computation can be performed over the cipher text producing a resultant cipher text which, when decrypted, equals the result of operations carry out on the plaintext. With help of this unique property of homomorphism cryptography we proposed a system to keep sensitive information in encrypted form in the cloud storage/service provider and used those data as whenever we require. The scheme proposed here is designed for a secure online voting system on Android platform and voted information is encrypted and stored those in the cloud.

Corresponding Author:

Monjul Saikia,
Departement of Computer Science and Engineering,
North Eastern Regional Institute of Science and Technology,
Nirjuli-791109, Arunachal Pradesh, India.
Email: monjuls@gmail.com

1. INTRODUCTION

The meaning of Encryption contains in the word itself, the base "encrypt" can be seen as "en" and "crypt". The meaning of "en" is "to make", and the word "crypt" means hidden or secret. Therefore the meaning of "encrypt" is "to make hidden or secret" [1].

In cryptography, encryption is the process of encoding messages or information in such a manner that only authorized user can interpret it. Hence the scheme of encryption is widely used for secrete communication between two parties. In an encryption scheme a large pseudo-random number called encryption key is generated by an algorithm which is to be used for computation of the cipher text. Later the receiver uses the same key to decrypt to extract the actual message being transmitted (in case of symmetric key encryption). A user without having the same key used for encryption cannot decrypt correctly.

The objective of modern cryptography can be classified in four categories:

a. Confidentiality: The information is secrete and cannot be recognize by unintended person.
b. Integrity: No modification can be done on the information send in between the sender and the intended receiver.

c. Non-repudiation: The sender cannot refuse the sending of the information once it is encrypted and passed to the receiver.
d. Authentication: Identity of the sender and receiver or the origin of the source and the destination is confirmed by each other.

Recent cryptography uses complex mathematical theory and computational logic making it hard to break such algorithm. Algorithms and procedures to break such algorithmic logic is also under the study of broad topic of cryptanalysis [2].

In a cloud based environment it seems that the demand for privacy of digital data and algorithms that handles the process of privacy have increased exponentially. Probability of suspicious attacks involved within or outside a third party service provider that may manipulate and destruct the stored sensitive information. Technology for guarantee security of private data in cloud is current topic of research. Although secure storage can be achieve by means of using cryptography algorithm, still serious problem arise when computation over the stored data is required. In recent days, development of suitable algorithms to perform computation over cipher data is a hot topic of research that falls under broad area of homomorphic cryptosystems [3].

2. HOMOMORPHIC CRYPTOSYSTEM

The concept of homomorphic encryption was first proposed in 1978 by Ronald Rivest, Leonard Adleman and Michael Dertouzos [4]. Over past 30 years a modest growth has been seen. In 1982 Goldwasser and Mical proposed an encryption scheme which has accomplished a notable level of security [5]. They proposed an additive homomorphic encryption which able to encrypt only one bit. In the year 1999 Pailler [6] had used the same thought to design an encryption system that can perform addition over cipher text, also called an additive homomorphic encryption. Dan Boneh et., al., [7] in 2005, invented a system of verifiable homomorphic encryption scheme, with which can perform a number of additions with only one multiplication at a time. In 2009 Craig Gentry [8] of IBM has proposed a scheme that can compute an arbitrary number of addition and multiplication over cipher text. The encryption system is called a "fully homomorphic encryption". The application of this scheme has been long waited due to the problem of complex computation, which later feasible after development of RSA. A solution proved harder to pin down for more than 30 years, the practical application of fully homomorphic encryption was uncertain whether it was even feasible. During this period, Boneh [9] showed the best result, with unlimited number of addition and at most one multiplication. In 2009 Craig Gentry from IBM showed the first fully homomorphic encryption using lattice-based cryptography.

3. HOMOMORPHIC CRYPTOSYSTEM CLASSIFICATION

Homomorphic Cryptosystem is mainly classified into two broad categories namely partially homomorphic cryptosystems and fully homomorphic Cryptosystem.Different encryption scheme under each category are as shown in Figure 1.

3.1. Partially Homomorphic Cryptosystem

A cryptosystem is said to be partially homomorphic [3] if it can perform either a addition or multiplication, but not both at the same time. Clearly, this is limiting the scheme of partial homomorphic encryption to certain applications, although the efficiency partial homomorphic encryption schemes is high enough for practical applications. Most partial homomorphic encryption schemes only support one type of operation, e.g. Paillier's algorithm supports addition whereas RSA supports only multiplications.

3.2. Fully Homomorphic Cryptosystem

A cryptosystem where arbitrary computation over ciphertexts can be performed is called a fully homomorphic encryption (FHE) [4]. A fully homomorphic encryption can be used for design of programs where various functionality are involved. Such a program can be run by an untrusted third party without revealing its original inputs and internal state to produce desired output. The discovery of the fully homomorphic cryptosystem would have various practical applications in the outsourcing of private computations as well as in the context of cloud computing.

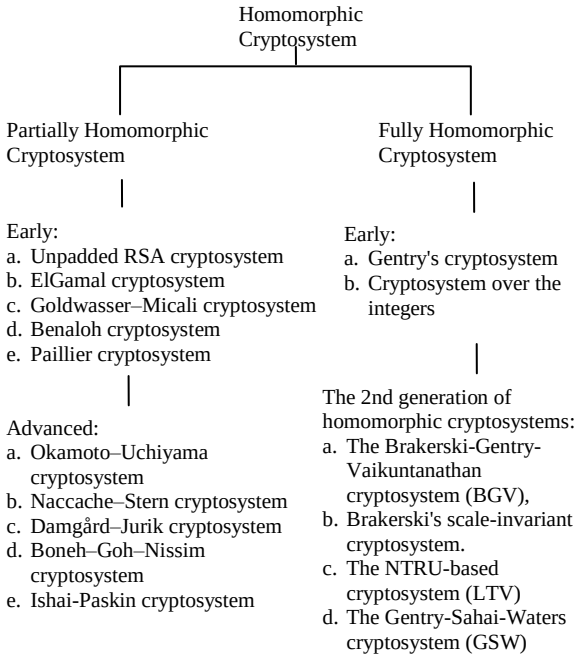


Figure 1. Homomorphic Cryptosystem Classification

4. PAILLIER'S HOMOMORPHIC CRYPTOSYSTEM AND THEIR HOMOMORPHIC PROPERTIES

Paillier's cryptosystem; invented by French researcher Pascal Paillier in 1999 is an algorithm for partially homomorphic cryptosystem [10]. It is indeed a public key cryptography where asymmetric keys are used. Users need to agree with a public key prior to the encryption process. A private key is kept secret, and used at the destination for decryption purpose. The recipient public key is used for the encryption purpose. The keys are generated with the help of some mathematical theories, restricting derivation of the private key from the public key.

The Paillier Cryptosystem scheme has three phases:

a. Key Generation.
b. Encryption.
c. Decryption.

4.1. Key Generation

1. Two large prime numbers p and q randomly are randomly chosen, which are independent of each other, such that $\gcd(pq, (p-1)(q-1)) = 1$.
2. Compute $n = pq$ and= $lcm(p-1, q-1)$.
3. Select random integer g where $g \in \mathbb{Z}^*_{n^2}$
4. Ensure n divides the order of g by checking:

$\mu = (L(g^{\lambda} mod n^2))^{-1} mod\ n$, where function L is defined as $L(u) = \frac{u-1}{n}$

a. The public (encryption) key is (n, g).
b. The private (decryption) key is (λ, μ).

4.2. Encryption

If the public key pair is (n, g), then the encryption of a message m is

$c = g^m r^n\ mod\ n^2$

for a random r where $r \in \mathbb{Z}^*_n$.

4.3. Decryption

If c is the cipher text to decrypt, where $c \in \mathbb{Z}_{n^2}^*$, we compute the plaintext message as:

$$m = L\left(c^{\lambda} \bmod n^2\right).\mu \bmod n$$

4.4. The Homomorphic Properties:

a. Homomorphic addition: The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts,

$$D(E(m_1, r_1) \,.\, E(m_2, r_2) \bmod n^2) = m_1 + m_2$$

b. Homomorphic multiplication: An encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts,

$$D(E(m_1, r_1)^{m_2} \bmod n^2) = m_1 m_2 \bmod n$$

$$D(E(m_2, r_2)^{m_1} \bmod n^2) = m_1 m_2 \bmod n$$

More generally, an encrypted plaintext raised to a constant *k* will decrypt to the product of the plaintext and the constant,

$$D(E(m_1, r_1)^{k} \bmod n^2) = k m_1 \bmod n$$

These special homomorphic properties are best suitable in the proposed secure voting scheme.

5. THE PROPSED VOTING SYSTEM MODEL

The proposed online voting system is designed to provide high security with minimum cost with paperless work in the process of voting. Paillier's encryption technique is used to encrypt the casted votes before storing it in the web server. The most valuable feature of Homomorphic encryption discussed earlier performs the operations on stored data to find out final result of voting without even decrypting those data. This way we can prevent a user from knowing whatever information being stored on the web server providing very high security as well as minimizing the cost by designing a user friendly Android interface for casting votes.

The proposed model has five phases:

a. Candidate Addition phase: In this phase the list contestant are to be uploaded to the server.
b. Voter Registration and validation phase: The voter registers themselves for voting purpose and administrator verifies the voter and gives access permission for voting.
c. Sharing pass code and public key phase: With help of a key generation algorithm public and private keys are generated, administrator then shares the public key with voters by a separate communication medium.
d. Voting process (with encryption) phase: Whenever voter goes for voting, the whole process goes through various phases of homomorphic encryption process.
e. Counting phase: Administrator finally checks the total votes against each of the candidates and declare the results.

It is a client server based voting system which has one administrator who can control the whole system. Administrator is the one who has the privilege to add the candidates as shown in Figure 2, to whom one can vote, generate password for user and calculate the result. A user / voter needs to register before voting. Administrator verifies his/her authenticity then only provides a password through choosing a secure channel. After getting the password the voter now can login to the actual voting system and can vote for any candidate of his/her choice.

Figure 3 shows how admin distributes the public key pair to the voters who have registered. This distribution need to be in a separate channel. After entering to the system a voter need to choose one candidate of his choice and then the encryption process begins at this stage. The cipher text of the casted votes will be stored in the web server database (Shown in Figure 4).

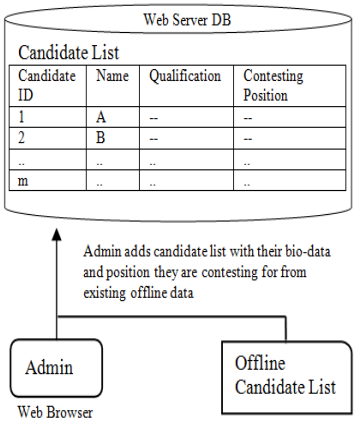


Figure 2. Adding Candidate to the System

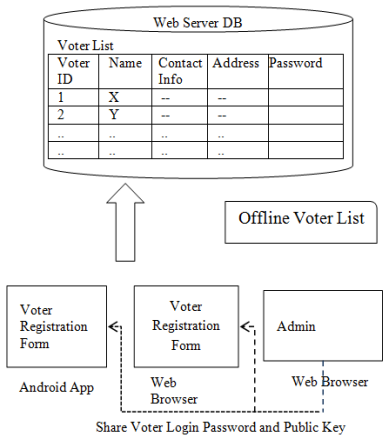


Figure 3. Android and Web Based Registration

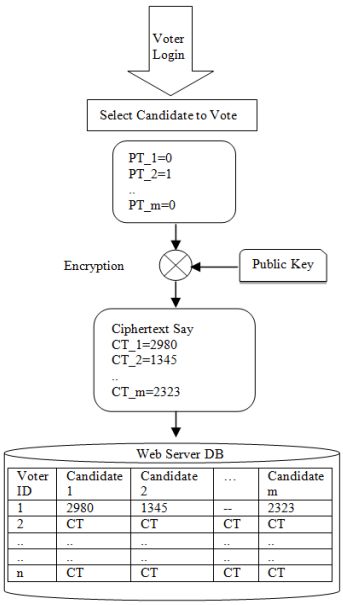
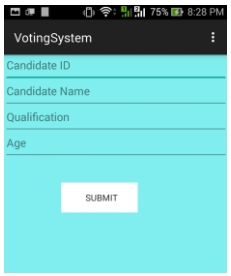


Figure 4. Voting Process

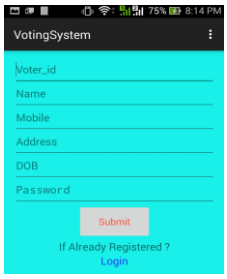
Figure 5 shows different Android interface for the voting system. Total number of vote a candidate obtains:

```
Total_A=A[1]*A[2]*----------*A[n];
A=Decrypt (Total_A)//Total number of votes obtain by A
Total_B=B[1]*B[2]*----------*B[n];
B=Decrypt (Total_B)//Total number of votes obtain by C
Total_C=C[1]*C[2]*----------*C[n];
C=Decrypt (Total_C)//Total number of votes obtain by C
```

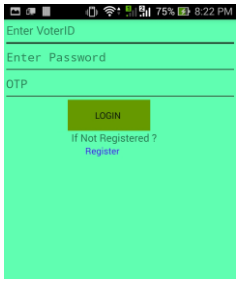
The casted votes of the voters which are saved in the web server database / cloud storage in the encrypted form are multiplied [Homomorphic addition property] and the decrypt function is applied to obtain the final results.



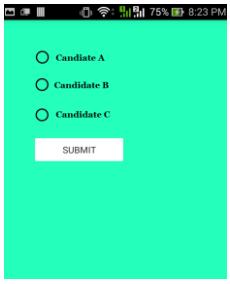
(a) Candidate Addition



(b) Voter Registration



(c) Voter Login Interface



(d) Voting Interface

Figure 5. The Android Interfaces for Voting System

6. CONCLUSION

In this paper we proposed a secure voting scheme applicable to Android platform using Paillier's Algorithm of Homomorphic Cryptosystem. We discuss the model with the prospective of computation over cipher text and computing the voting results with the help of their Homomorphism properties. It is observed that the propose model works efficiently in a web server based model, having Android based interface for casting votes and a cloud based web server for storing voted information in encrypted form. The said model can be extended to make suitable for various applications like Cloud Computing, Untrusted web servers, Secret Sharing scheme, Protection of mobile agents, Election schemes etc.

ACKNOWLEDGEMENTS

This research was supported by TEQIP II, NERIST. We thank our colleagues from department of CSE, NERIST who provided insight and expertise that greatly assisted the research. We thank Prof., Md., Anwar Hussain of ECE Department, NERIST for his guidance and comments that greatly improved the manuscript.

REFERENCES

[1] S. Lian, "Multimedia Content Encryption: Techniques and Application," *CRC Press*, ISBN 987-1-4200-6527-5.
[2] J. A. Buchmann, "Introduction to Cryptography (2nd ed.)," *Springer*, ISBN 0-387-20756-2.
[3] D. Rappe, "Homomorphic Cryptosystems and their Applications," *Doctoral Dissertation, University of Dortmund, Dortmund, Germany*.
[4] R. L. Rivest, *et al.*, "On data banks and privacy homomorphisms- In Foundations of Secure Computation," 1978.
[5] S. Goldwasser, *et al.*, "Why and How to Establish a Private Code on a Public Network," *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS'82)*, Chicago, Illinois, pp. 134-144, 1982.
[6] M. Tebaa, *et al.*, "Homomorphic Encryption Applied to the Cloud Computing Security," *Proceedings of the World Congress on Engineering 2012*, July 4 - 6, London, U.K., vol. I, 2012.
[7] D. Boneh, *et al.*, "Evaluating 2-DNF formulas on ciphertexts," *In Theory of Cryptography Conference, TCC'2005, Lecture Notes in Computer Science*, vol. 3378, pp. 325-341, 2005.
[8] C. Gentry, "A Fully Homomorphic Encryption Scheme," 2009.
[9] C. A. Melchor, *et al.*, "Additively homomorphic encryption with d-operand multiplications," *Cryptology ePrint Archive, Report 2008/378*, http://eprint.iacr.org/, 2008.
[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *In Advances in cryptology -EUROCRYPT'99*, Springer Berlin Heidelberg, pp. 223-238, 1999.

BIOGRAPHIES OF AUTHORS

Manish Ranjan is a student of Bachelor of Technology in North Eastern Regional Institute of Science & Technology Deemed University, Arunachal Pradesh, India. He has very good knowledge in the field of information security. His major areas of interests are Cloud computing, Android development studio, Cryptography, web services etc.

Ayub Hussain Mondal is a student of Bachelor of Technology in North Eastern Regional Institute of Science & Technology, Deemed University, Arunachal Pradesh, India. He has very good knowledge in the field of information security. His major research areas are Cloud computing, Cryptography, web services etc.

Monjul Saikia is Assistant Professor in North Eastern Regional Institute of Science and Technology, Deemed University, Arunachal Pradesh, India. He is a member of various famous and reputed Technical and Research organizations such as Computer Society of IEEE, India (India), IEI (India), ISTE India etc. Currently he is pursuing his Ph.D. in department of ECE, NERIST. He has many contributions in the field of Information Technology and the areas of information security, cryptography, VLSI, multimedia signal processing etc.