Open access Journal International journal of Emerging Trends in Science and Technology

Securing the Cloud Computing Systems with Multi-Keys Using Homomorphic Encryption Method

Authors

Ananth Thangarajan
M.E.
Email: *ana.scorp@gmail.com*

Dr. P. Balakumar
M.E.
Email: *psbalakumar@gmail.com*

Abstract -This paper describes to improve the security in cloud system when end-user communicates with the cloud server. In order to make a connection between end-user and the Cloud Server, first end-user or Cloud Server make sure that they are communicating with right counterpart. Multi-Key encryption concept is used to encrypt the request message or data which is sent from end-user or also from Cloud Server. Request message sent by end-user to cloud server should matches with the response received from by end-user from cloud server. Multi-key concept takes more number of iteration when compared to single-key concept when trying to validate the requested message. Multi-Key concept is more secured than single-key.
Keywords: Deterministic, Non-Deterministic, Multi-Keys, Linear Equation, Matrix

1. Introduction

Cloud computing is the provision of dynamically scalable and often virtualized resources as a service over the internet. Cloud computing represent major changes in storing information and run applications. Instead of hosting an applications or storing a data in the personal computer everything is hosted in the cloud machine. Data which is available in the Cloud has been accessed via the internet.

End-user wants to store their confidential data into the cloud machine, in order to store and access data in the cloud server, first step is to make a secure connection between end-user and cloud machine. Then user can access those data from internet and also from different location across globe. If the user wants to access their data, user has to make a secure connection to the cloud server and then user will send a request message to the cloud server and cloud server will validate the request messages which are received from the end-user. Both the end-user and the cloud server will do the validation and then secure connection will make only when the validation is successful on both sides.

2. Architecture

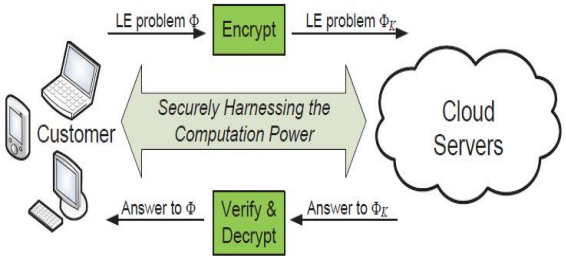

Figure 1. Architecture

End-user has to send a message to cloud server and vice-versa. Message should be encrypted with end-user keys. End-user will send an encrypted message to cloud server and cloud server will process the request message and stored in cloud server's local variable for verification purpose and then cloud server will again encrypt the encrypted message and send back to end-user. End-user will decypt the received

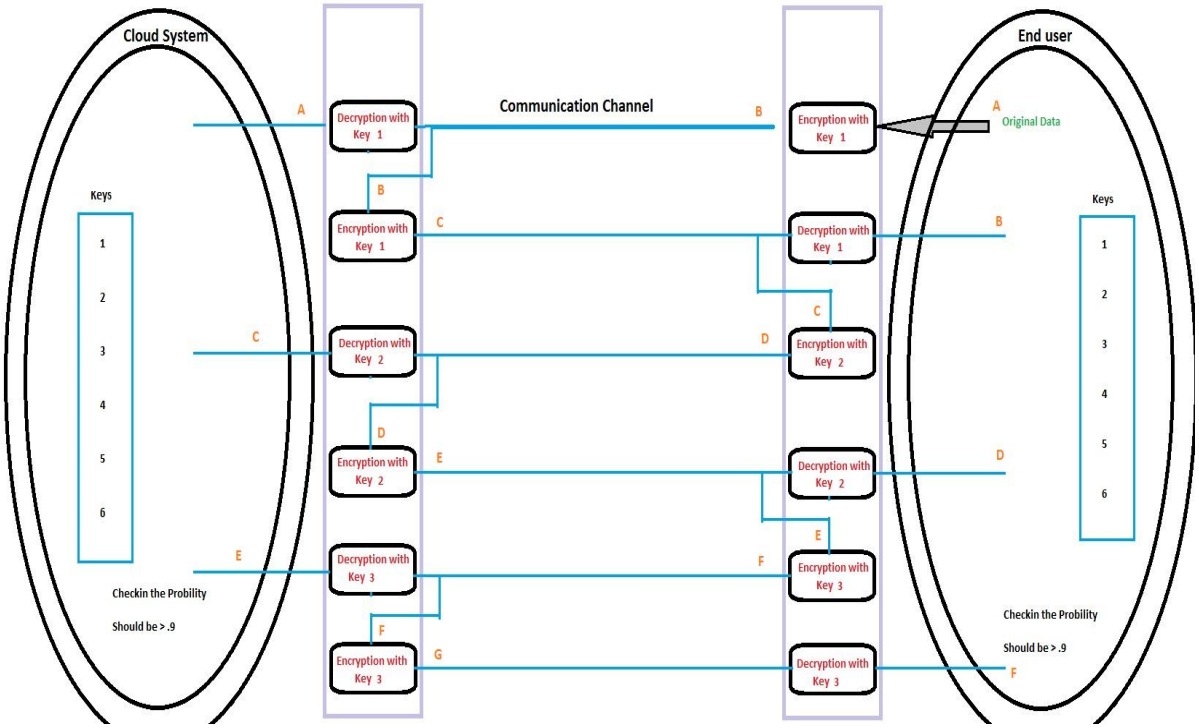

Figure 2. Flow Diagram

message and will verify the received message with the outgoing message. Both received and the outgoing message should be same. End-user will verify the processed message (double encryption with the same key from end-user and the cloud server) received from the cloud server and then secure connection will establish between customer and the cloud server.

3. Key generation

End-user will generate a set of keys and send to cloud server or cloud server will generate a set of keys and send to end-user. Once the key is generated and the generated keys should be shared between the end-user and the cloud server. Then end-user has to send a request message to make a connection with cloud server and vice-versa. End-user will encrypt the request message Encrypt(m1) with the Key1 and send to cloud server through communication channel. Cloud server will decrypt the message sent by end-user [Decrypt(Encrypt(m1))] with Key1 and also will encrypt the same encrypted message (before decryption in cloud server side) which is received from the end-user with the Key1 which is available in cloud server and send back the encrypted message to end-user for validation. End-user will decrypt the message which is received from cloud server and compared with the encrypted message which is sent by end-user and should match otherwise someone has interrupted the communication channel or someone has modified the data which is sent through communication channel.

Lets take "A" is the request message to be send from end-user and this message is encrypted with Key1, so the encrypted message will be Encrypt(A,Key1)→B. Message B is sent through communication channel and this has been decrypted with Key1 in the cloud server Decrypt(B,Key1)→A and this message will be stored in cloud server. Again message B is encrypted with Key1 Encrypt(B,key1)→C [Encrypt(Encrypt(A,Key1), Key1)] and C

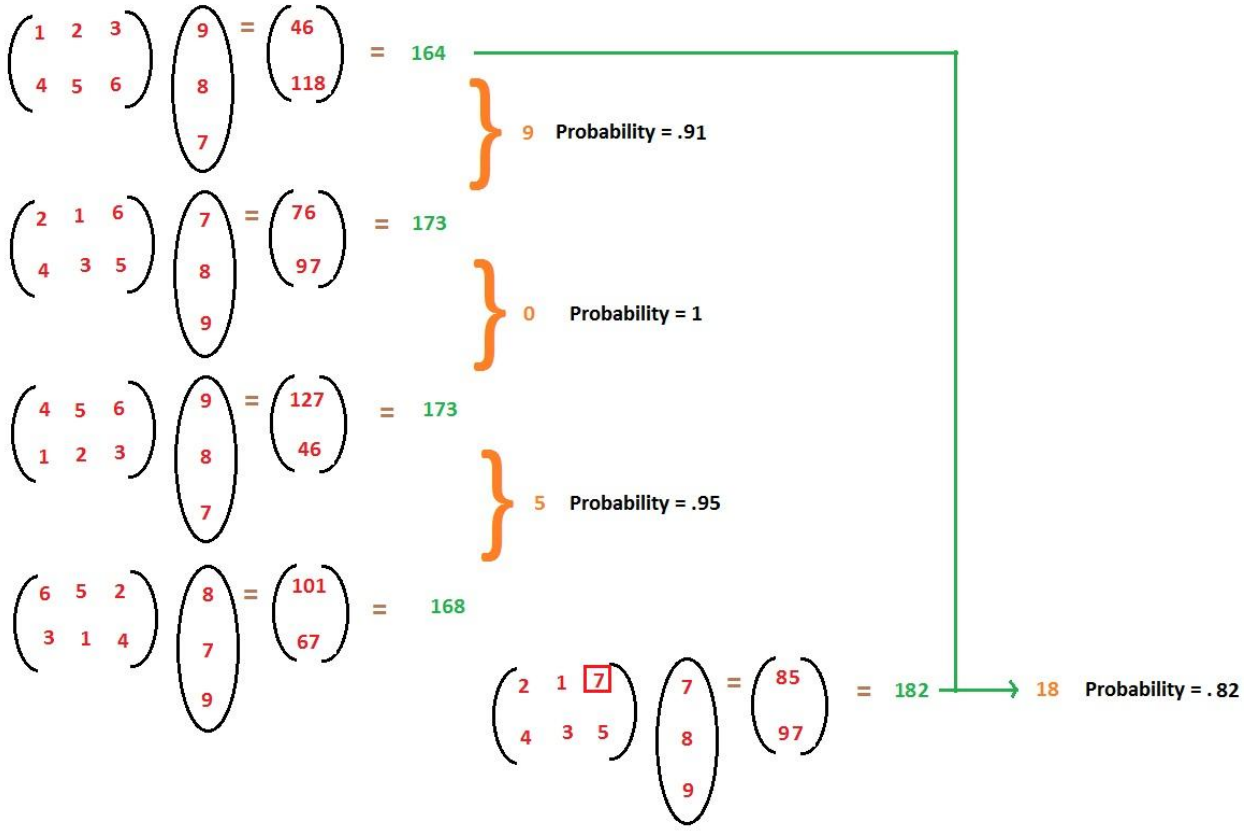

Figure 3. Solving Linear Equation

will be sent to end-user and message C will be decrypted with Key1 Decrypt(C,Key1)→B. Output of end-user is B and also the message received from the cloud server is also B, so input message should matches with the output message. Hence the message is communicated securely and no has interrupted the message in the communication channel

4. Solving Linear Equation

This matrix calculation is used to find out the probability of the output and the input message which is communicated between end-user and the cloud server. Let's take [1 2 3 4 5 6] is the generated keys which is shared between end-user and the cloud server and [9 8 7] is the request message or data which is sent to the cloud server for validation or store data in the cloud server. Once the Keys are generated and need to share between the end-user and the cloud server. Sharing of the keys should be through communication channel. There may be a chance of altering or hacking the secret keys. Suppose the message [9 8 7] is encrypted with the Key1, its giving the some encrypted value and which should be stored in variable and again in the second iteration if we encrypt the same encrypted message with the Key1 which is available in cloud server and send to end-user, it gives the different value and stored in the variable. Here we should find the difference in the variables which is generated in the first and second iterations and the difference should be greater than .9. Suppose, if the key is altered (Key 6 is modified to 7 as shown in the below Fig.3) and encrypt the same message and will produce some result and the difference should be always less than .9 (refer the below Fig.3). Same process will be manipulated on both end-user and the cloud server. If the probability value is less than .9, then the end-user or cloud server will send a request message to generate a new set of keys for a secure communication. Again the same process will be carried out to validate the input and the output message for the secure communication.

5. Homomorphic Encryption

A. Functionality:

An encryption scheme has three algorithms KeyGen, Encrypt and Decrypt of which must be efficient. Polynomial is a security parameter that specifies the length of the keys. In secret-key, encryption scheme, KeyGen uses to generate a key that is used in both Encrypt and Decrypt, first to map a original message to an encrypted message, and then to map the encrypted message back to the original message. In public-key encryption scheme, KeyGen uses to generate two keys a public encryption key, which may be made available to everyone and a secret decryption key. As an asymmetric encryption scheme, one can find Alice's public key as a security lock, which she can constructs and distributes, used to lock without a key. Anyone can place their message inside her security box (encrypt), and mail it via a public channel to Alice, but only Alice can able to unlock the box using the secret key (decrypt).

Full Homomorphic encryption is to handle both Deterministic and Non-Deterministic and has a compact ciphertexts and also Evaluate the encryption and decryption in an efficient way. If an encryption scheme is deterministic i.e., if there is only one encrypted text which has encrypts a given message then it cannot be semantically secure. An attacker can easily tell whether ciphertext encrypts m0 (message) by running c0 Encrypt(pk,m0) and seeing if c and c0 are same. A secure encryption scheme must be probabilistic i.e., there must be many encrypted text with the given message, and the encryption must choose randomly according to random distribution. Also the message which we need to send or receive data should be in shuffled order. One can prove the (conditional) one-wayness or semantic security of an encryption scheme by reducing a hard problem to breaking the encryption scheme. For example, if there is an efficient algorithm that breaks the encryption algorithm, then this algorithm can be iterated multiple times like a subroutine to a large numbers.

B. Deterministic Method:

In this method, we are sending the encrypted message in a sequence of message. For eg. "I am in Namakkal" is the message which I need to send to the cloud server. From the end-user, first we will to encrypt the "I" and send to the cloud server through the communication channel and then encrypt the "am" and send to the cloud server and then "in" and then "Namakkal". Hacker can easily identify the sequence, if they identify the sequence "I am __ Namakkal", then hacker can easily identify the message "in". Then hacker can easily frame the sentence.

Example: 1+2+3+4+5
3 + 3+4+5
6 + 4+5
10 +5 = 15

C. Non-Deterministic Method:

In this method, we are sending the encrypted message in a sequence of message. For Eg. "I am in Namakkal" is the message which I need to send to the cloud server. From the end-user, first we will encrypt the "Namakkal" and send to the cloud server through the communication channel and then encrypt the "in" and send to the cloud server and then "am" and then "I". Hacker cannot easily identify the sequence.

Example: 1+2+3+4+5
1+ 5 +4+5
1+ 5 + 9
10+5 = 15

We have formulated the problem in outsourcing large amount of data in the systems with the help of LE via iterative methods and provide mechanism to achieving privacy, cheating resilience and efficiency.

6. Cloud Storage

In order to improve the security we can store the data in cloud system and that data should be in encrypted format. Even if the person who is managing the cloud system cannot able to read that message which is available in cloud server. Stored message should be authenticated, then only end-user can able to read or write the message into the cloud server. For Eg. Alice owns a jewel shop. She has precious materials like gold, silver, etc. that she wants her workers to design rings. But she doesn't trust her workers and she assumes that they will steal her valuable things, if they have an opportunity. In other words, she wants her workers to finish pieces without giving them any access to the materials.

Here is her plan. She uses a transparent box called glove-box, secured by a lock. She places the raw precious materials inside the secured box, locks it with her secret key and gives that secured box to the worker. Using the glove box, the worker will assemble the ring inside the box. Since the glove box is im-penetrable, the worker cannot able to get the precious materials without her permission and he might as well return the box to Alice, with the finished piece inside. Alice will unlock the box with her secret key and can able to extracts the ring. Then worker processes the precious materials into a finished piece, without having access to the materials.

The locked impenetrable box which has raw precious materials inside which represents an encryption of the initial data m1…mt, which can be accessed only with the secret key. The gloves represent the homomorphism or malleability of the encryption scheme, which allows the original raw data to be manipulated in the secured box. The completed furnished ring which is inside the box represents the encryption of original message, the desired function of the initial data. Lack of access to the secured box is represented by lack of physical access. Of course, Alice's jewel shop is only an analogy. It does not represent the homomorphic encryption. We discuss some flaws in the algorithm, after we describe homomorphic encryption more formally. Despite its flaws, we return to the algorithm. Since it motivates good questions and represents some aspects of our solution quite well most notably bootstrapping.

7. Limitations

Here we are using Deterministic Method using Multi-Key. Arranging the received message is easy in Deterministic method. But, in Non-Deterministic with multi-key, the data will be sent in different sequence of message as discussed in section 5.2. This Multi-key concept requires more iteration to retrieve the original message from the encrypted message. Increase in the number of keys leads to increase in the more number of iteration and also to improve the security and also decrease the chance of intruder to decrypt the message.

The physical analogy in real time scenario represents some aspects of homomorphic encryption method. For example, the physical analogy suggests that original messages that are encrypted separately are in different "encryption boxes" and cannot interact with each other. Of course, these interactions are precisely the purpose of homomorphic encryption. In order to fix this analogy, one may imagine that the gloveboxes have a one-way insertion slot like the mail bins used by the post office. Then original messages can be added to the same glovebox or encryption box as they arrive.

Another flaw is that the output f(m1…mt) may have significantly fewer bits than m1…mt, whereas in the analogy (absent significant nuclear activity inside the glovebox) the conservation of mass dictates that the box will have at least as much material as inside the glovebox when the worker is done as when he started. Finally, in Alice's jewel shop even though a worker cannot access the materials from a locked or secured glovebox. But he can able to tell that whether or not a box contains a certain set of materials. But he cannot identify the importance of the material

which is available inside the glovebox i.e., gloveboxes or secured box do not provide "semantic security".

8. Comparative study

In Homomorphic Deterministic method arranging the encrypted data is easy when compared to Homomorphic Non-Deterministic method, because when the received message is in the same order as in the similar sequence that we has sent in the sender side. But in Non-Deterministic method rearranging the received message need separate algorithm to update the timestamp in the sending message which is used to rearrange the message in the received message.

9. Key Characteristic

Agility improves with user ability to re-provision technological infrastructure resources in an efficient manner.

Cost is claimed to be reduced and delivering an operational expenditure in a public cloud. An infrastructure is typically provided by a third-party and does not need to be purchased and update the local machine frequently. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation. The cost of savings depends on the type of activities supported and the type of infrastructure available in-house.

Virtualization technology allows servers and storage devices to be shared and utilization has to be increased and applications can be easily migrated from one physical server to another server.

Centralization is to keep the data and the services in a centralized location and we can access from all over the globe.

Utilization and efficiency improvements for systems that are often only 10–20% utilized.

Reliability can be improved by using multiple redundant sites which makes well designed cloud computing suitable for business continuity and disaster recovery.

Performance is monitored and consistently improved and loosely coupled architectures are constructed using web services as the system interface.

Security could be improved by centralizing the data, focusing the security features and etc., but concerns can persist about loss of control over certain sensitive data and the lack of security for stored kernels. Security is often as good as or may be better than other traditional security systems, because providers are able to devote resources to solving security issues that many customers cannot afford. The complexity of the security has to be increased when data is distributed over a wider area or greater number of devices. In addition, user access to security audit logs may be difficult or impossible.

Maintaining the cloud computing applications is easier because they do not need to be installed on each user's computer and can be accessed from different places.

10. Conclusion

This paper describes about to improve the security issue in Cloud Server when end-user trying to access the cloud server data from across the globe. Existing system is implemented using the single key using homomorphic encryption method to perform the data encryption and also for validation Matrix-Vector formulation has been used. Accessing the cloud data from public place is prone to change or modify the original message. So in order to improve the security obviously we need to increase the keys to encrypt and also increase the computation to validate the message. So in this paper we are describing about to use Multi-Keys using Homomorphic encryption method to improve the security in Cloud System.

As we are using Multi-Keys to improve the security in the Cloud System, so number of iteration will increase to validate the received message. In the future we need to find out the algorithm to reduce the number of iteration while validating the received message. And also we need to improve the efficiency of the system to speed up data retrieval from the Cloud System.

11. References

[1] Harnessing the Cloud for Securely Outsourcing Large-Scale Systems of Linear Equations, Cong Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, Jia Wang, Member, IEEE, and Qian Wang, Member, IEEE – 2013

[2] C. Gentry, "Computing Arbitrary Functions of Encrypted Data," Comm. ACM, vol. 53, no. 3, pp. 97-105, 2010.

[3] K. Forsman, W. Gropp, L. Kettunen, D. Levine, and J. Salonen, "Solution of Dense Systems of Linear Equations Arising from Integral-Equation Formulations," IEEE Antennas and Propagation Magazine, vol. 37, no. 6, pp. 96-100, Dec. 1995.

[4] A. Edelman, "Large Dense Numerical Linear Algebra in 1993: The Parallel Computing Influence," Int'l J. High Performance Computing Applications, vol. 7, no. 2, pp. 113-128, 1993.

[5] B. Carpentieri, "Sparse Preconditioners for Dense Linear Systems from Electromagnetic Applications," PhD dissertation, CERFACS, Toulouse, France, 2002.

[6] R. Cramer and I. Damga°rd, "Secure Distributed Linear Algebra in a Constant Number of Rounds," CRYPTO: Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology, 2001.

[7] P. Mohassel and E. Weinreb, "Efficient Secure Linear Algebra in the Presence of Covert or Computationally Unbounded Adversaries," CRYPTO: Proc. 28th Ann. Int'l Cryptology Conf., pp. 481-496, 2008.

[8] J.R. Troncoso-Pastoriza, P. Comesan˜ a, and F. Pe´rez-Gonza´lez, "Secure Direct and Iterative Protocols for Solving Systems of Linear Equations," Proc. First Int'l Workshop Signal Processing in the EncryptEd Domain (SPEED), pp. 122-141, 2009.

[9] W. Du and M.J. Atallah, "Privacy-Preserving Cooperative Scientific Computations," Proc. IEEE 14th Computer Security Foundations Workshop (CSFW), pp. 273-294, 2001

[10]Dr. P. Balakumar M.E. Ph.D., Professor, Head of the Department, Department of Computer Science and Engineering, Mahendra Institute of Technilogy, Thiruchengode.

[11]Ananth Thangarajan M.E., Department of Conputer Science and Engineering, Mahendra Institute of Technology, Thiruchengode.