International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 8 (2014), pp. 811-816
© International Research Publications House
http://www. irphouse.com

A Fully Homomorphic Encryption Implementation on Cloud Computing

Shashank Bajpai and Padmija Srivastava

Cloud Computing Research Team,
Center for Development of Advanced Computing [C-DAC], Hyderabad,
Ministry of Communications and Information Technology, Government of India,
shashankb@cdac.in, padmijas@cdac.in
www.cdachyd.in

ABSTRACT

Cloud Computing has been the most promising innovation in the computing world in past decade. Its usage is still hindered by the security concerns related with critical data. The encryption of remotely stored data has been the most widely used technique to bridge this security gap. The speculated vast usage of Cloud Computing solutions for data storage and with Big Data Analytics gaining strong foothold; the security on cloud is still at big risk. Fully Homomorphic Encryption is a good basis to enhance the security measures of un-trusted systems or applications that stores and manipulates sensitive data. The model is proposed on cloud computing which accepts encrypted inputs and then perform blind processing to satisfy the user query without being aware of its content, whereby the retrieved encrypted data can only be decrypted by the user who initiates the request. This allows clients to rely on the services offered by remote applications without risking their privacy.

Key Points: Cloud Computing, Security, Fully Homomorphic Encryption, Implementation on Cloud, Future ScopeText

1. Cloud Computing

The information technology model for computing, which is composed of all the IT components (hardware, software, networking, and services) that are necessary to enable development and delivery of cloud services via the Internet or a private network. The prominent actors in Cloud Computing are Cloud Provider and Cloud User. Cloud Provider is the enterprise vendoring cloud services. A Cloud User can vary from organisations, educational institutes to individuals utilising the cloud services. This definition has no notion of security for data in the cloud computing even if it's a very new. There is a necessity for security, confidentiality and visibility with respect to the current cloud providers. Provide Infrastructure as Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS) is not sufficient if the provider of the Cloud does not guaranty a better security and confidentiality of customer data. Cloud providers like: IBM, Google and Amazon use the virtualization in their Cloud platform, and in the same machine can coexist the storage space and treatment virtualized which belong to the concurrent enterprises. The aspect of security and confidentiality must intervene to protect the data from each of the enterprises.

2. Cloud Security

At present both in Public Cloud and Private Cloud; security ensures to encrypt the data stored. Also it is very easy to have secure transmission from a local machine to a cloud data store. The stored data being encrypted and the channel of data transmission well secured with key exchanges. But actually performing computations on that data stored in the cloud requires decrypting it first; this makes critical data available to the cloud provider. Data Mining and other Data Analysis onto the Encrypted Database is a far distant thing to achieve by using encryption standards available. The proposal here is to encrypt data before sending to the cloud providers. Thereby to enable a cloud computing vendor to perform computations on clients' data at their request, such as analyzing sales patterns, without exposing the original data. To achieve this it is also necessary to hold the cryptosystems based on Homomorphic Encryption either a Fully Homomorphic Encryption (FHE) or Somewhat Homomorphic Encryption (SHE).

3. Homomorphic Cryptosystems

They are ones where mathematical operations on the ciphertext have regular effects on the plaintext. A very simple demonstration of the mathematical consistency required: A user sends a request to add the numbers 1 and 2, which are encrypted to become the numbers 33 and 54, respectively. The server in the cloud processes the sum as 87, which is downloaded from the cloud and decrypted to the final answer, 3.A normal symmetric cipher -- DES, AES is not homomorphic. The RSA algorithm is homomorphic but only with respect to multiplication.

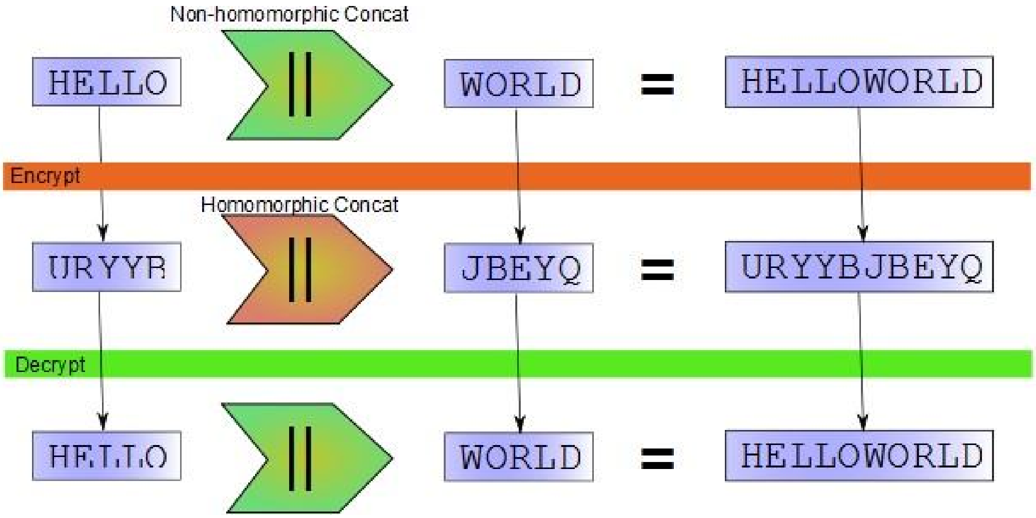


Figure 1. *A string concat example of Homomorphic Encryption*

At first, the notion of processing data without having access to it may seem paradoxical, even logically impossible.To convince you that there is no fallacy, and to give you some intuition about the solution, let us consider an analogous problem in the physical world. Sita owns a jewelry store. She has raw precious materials gold, diamonds, silver, etc. She wants her workers to assemble into intricately designed rings and necklaces. But she distrusts her workers and assumes that they will steal her jewels if given the opportunity. In other words, she wants her workers to process the materials into finished pieces, without giving them access to the materials. For that she uses a transparent impenetrable glovebox, secured by a lock for which only she has the key. She puts the raw precious materials inside the box, locks it, and gives it to a worker. Using the gloves, the worker assembles the ring or necklace inside the box. Since the box is impenetrable, the worker cannot get to the precious materials, and ures he might as well return the box to Sita, with the finished piece inside. Sita unlocks the box with her key and extracts the ring or necklace. In short, the worker processes the raw materials into a finished piece, without having true access to the materials. Of course, Sita's jewelry store is only an analogy.

4. Implementation of FHE

In 2009 Craig Gentry of IBM has proposed the first encryption system “Fully Homomorphic” that evaluates an arbitrary number of additions and multiplications and thus calculate any type of function on encrypted data. The internal working of this adds another layer of encryption every few steps and uses an encrypted key to unlock the inner layer of scrambling. This decryption "refreshes" the data without exposing it, allowing an infinite number of computations on the same.

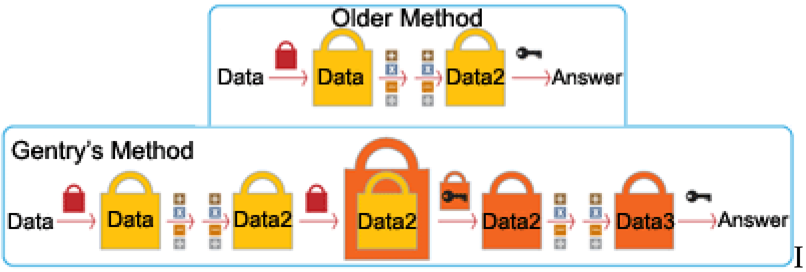


Figure 2. *Craig Gentry implementation of FHE.*

5. FHE on Cloud

The application of fully Homomorphic encryption is an important brick in Cloud Computing Security; more generally, outsourcing of the calculations on confidential data to the Cloud server is possible, keeping the secret key that can decrypt the result of calculation. In our implementation, we analyze the performance of existing homomorphic encryption cryptosystems, and are working on a virtual platform as a Cloud server, a VPN network that links the Cloud with the customer, and then simulating different scenarios. For example a Database-Server communicating with Client using FHE Cryptosystem is as shown in the figure below.

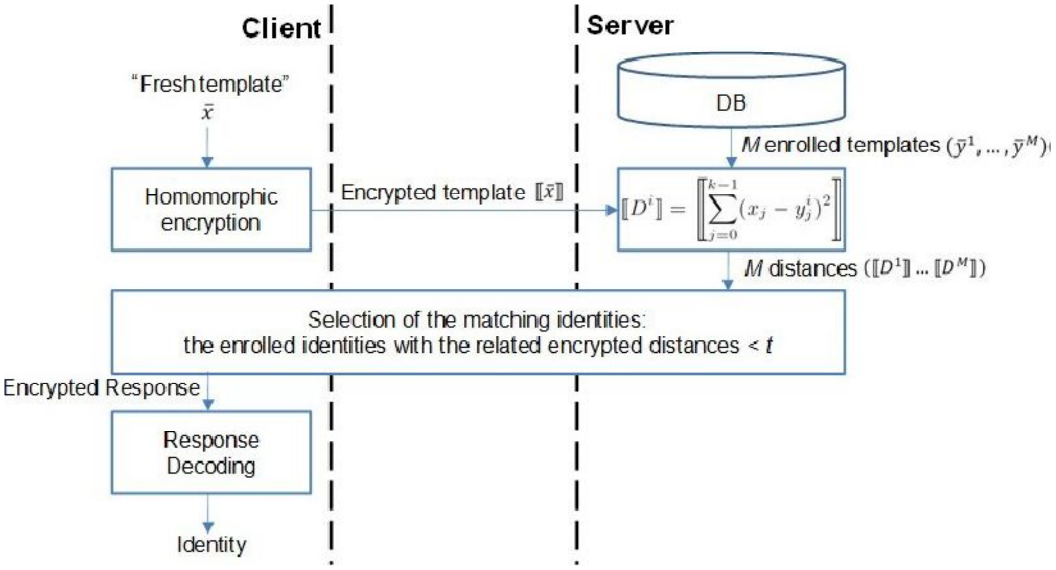
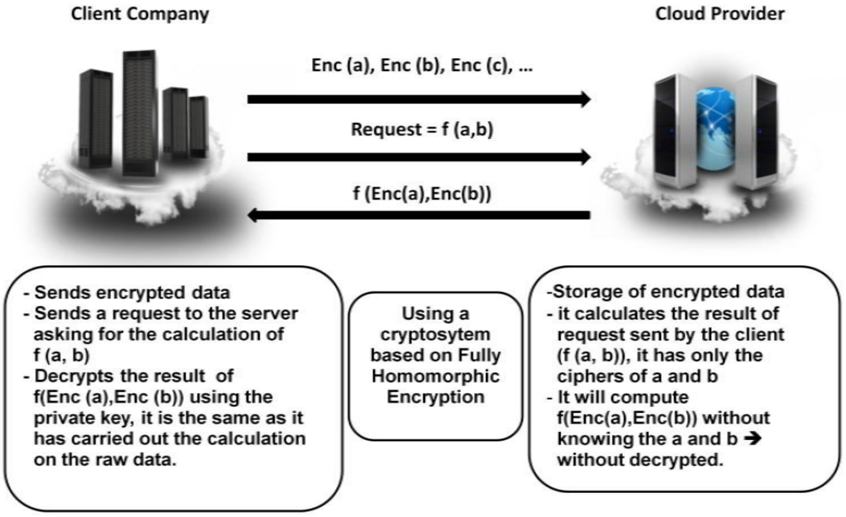

Client
Server
"Fresh template"
$\bar{x}$
DB
M enrolled templates $(\bar{y}^1, \ldots, \bar{y}^M)$
Homomorphic encryption
Encrypted template $[\![\bar{x}]\!]$
$[\![D^i]\!] = \left[\!\left[\sum_{j=0}^{k-1} (x_j - y_j^i)^2\right]\!\right]$
M distances $([\![D^1]\!] \ldots [\![D^M]\!])$
Selection of the matching identities:
the enrolled identities with the related encrypted distances < t
Encrypted Response
Response Decoding
Identity

Figure 3. *A Database-Sever & Client implementing Homomorphic Encyption*

On similar lines; the Cloud Computing scenario can be illustrated as below:


Client Company
Cloud Provider
Enc (a), Enc (b), Enc (c), ...
Request = f (a,b)
f (Enc(a),Enc(b))
- Sends encrypted data
- Sends a request to the server asking for the calculation of f (a, b)
- Decrypts the result of f(Enc (a),Enc (b)) using the private key, it is the same as it has carried out the calculation on the raw data.
Using a cryptosytem based on Fully Homomorphic Encryption
-Storage of encrypted data
- it calculates the result of request sent by the client (f (a, b)), it has only the ciphers of a and b
- It will compute f(Enc(a),Enc(b)) without knowing the a and b ➔ without decrypted.

6. Challenges of FHE on Cloud

The double layer of encryption causes the system runs too slowly for practical use.

We are working on optimizing the same for specific applications such as searching databases for records reduce the time complexity. Also to trust a very new encryption scheme for confidentiality is not feasible and it requires considerable (~10 yrs) of usage exposure. A team from MIT's Computer Science and Artificial Intelligence Laboratory, who worked in conjunction with the University of Toronto and Microsoft Research, sought to combine multiple schemes to solve these challenges. The system starts with homomorphic encryption, with a decryption algorithm embedded in a garbled circuit which is itself protected by attribute-based encryption this ensures the process stays encrypted.

7. CONCLUSION

Security of cloud computing based on fully homomorphic encryption is a new concept of security which is to enable to provide the results of calculations on encrypted data without knowing the raw entries on which the calculation was carried out respecting the confidentiality of data. Our work is based on the application of fully Homomorphic encryption to the security of Cloud Computing: a) Analyze and improve the existing cryptosystem to allow servers to perform various operations requested by the client. b) Improve the complexity of the homomorphic encryption algorithms and study the response time to requests according to the length of the public key.

8.REFERENCES

[1] Craig Gentry, A Fully Homomorphic Encryption Scheme, 2009 http://crypto.stanford.edu/craig/craig-thesis.pdf

[2] Understanding Homomorphic Encryption - http://en.wikipedia.org/wiki/Homomorphic_encryption

[3] FHE implementation with garbled circuit - http://eprint.iacr.org/2010/145.pdf

[4] New encryption method promises end-to-end cloud security, by Kevin McCaney Jun 13, 2013 - http://gcn.com/Articles/2013/06/13/Encryption-end-to-end-cloud-security.aspx?Page=1

[5] Homomorphic Encryption Applied to the Cloud Computing Security by Maha TEBAA, Saïd EL Hajji, Abdellatif EL Ghazi - http://www.iaeng.org/publication/WCE2012/WCE2012_pp536-539.pdf

[6] Vic (J.R.) Winkler, "Securing the Cloud, Cloud Computer Security, Techniques and Tactics", Elsevie r

[7] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic , volume 1592, 1999

[8] Julien Bringe and al. An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication, Springer-Verlag , 2007. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2):120-126, 1978. Computer Science, pages 223-238. Springer, 1999.

[9] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 469-472, 1985.

[10] WiebBosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. J. Symbolic Comput., 24(3-4): 235 -265, 1997. Computational algebra and number theory, London, 1993.

[11] Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. On Data Banks and Privacy Homomorphisms, chapter On Data Banks and Privacy Homomorphisms, pages 169-180. Academic Press, 1978.

[12] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Theory of Crypto gr aphy Conference, TCC'2005, volume 3378 of Lecture Notes in Computer Science, pages 325-341. Springer, 2005.