International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 12 4180 - 4183

Somewhat Homomorphic Encryption Technique with its Key Management Protocol

Samyak Shah.
Computer department,
K. J. Somaiya College of Engineering,
Mumbai, India.
samyak.shah@somaiya.edu

Yash Shah.
Computer department,
K. J. Somaiya College of Engineering,
Mumbai, India.
yash.shah@somaiya.edu

Janika Kotak.
Computer department,
K. J. Somaiya College of Engineering,
Mumbai, India.
janika.kotak@somaiya.edu

Abstract-- Cloud computing has been contemplated as the architecture of various Business organizations, providing easy access to vast data storage and applications services. Most of the cloud service providers encrypt the data only on the network , while some even store the data in encrypted format. This means anyone with access to the cloud servers (cloud service providers) can appropriate it. Even if the data is encrypted during storage, keys are often stored along with your data .Thus an end-to-end encryption scheme has been proposed as a promising solution to data storage on cloud ,in order to perform computations on the encrypted data and thereby store the key securely. Somewhat Homomorphic Encryption is a fully homomorphic encryption technique which is compact, semantically secure with significantly smaller public key and is capable of encrypting integer plaintexts rather than single bits, with comparatively lower expansion and computational complexities.

Keywords-*Cloud computing, Cryptography, Homomorphic Key Management (HKM), Homomorphic encryption, Somewhat Homomorphic encryption(SHE).*

*****

I. Introduction

Cloud Computing has the potential to revolutionize a large part of the IT industry, making software even more service-effective and transforming the way computer hardware is designed and purchased. But there are many security issues related with storage on cloud. Security issues for cloud computing are as given below:

Availability – Data availability is one of the significant issues of organizations for which safety is topmost priority. Availability concerns also extend to the want to drift to another Cloud service provider.

Data Security – Data security risk is mainly from loss of physical, public and rational control of data. Other risks included in [12] are data leakage and breach, distributed denial of service and loss of cryptographic keys. The inability to fully differentiate data or segregate clients can lead to exposure of confidential data . Data containing medical details of clients, data of defence organisations can raise issues about authorization, authentication and access controls.

Third-Party Control – This is probably the significant reason of concern in the cloud. With the growing value of corporate information, third party access can lead to a possible loss of intellectual property and trade secrets. Also there is a problem that a cloud service provider abuses admission rights to client information. The fear of corporate spying and data clash also roots from third party control. . A situation can also arise in which the user becomes confined to a particular seller.

Privacy and Legal Issues – Data in the cloud which is usually distributed worldwide raises concerns about jurisdiction, data exposure and confidentiality .[11]

Normally in Encryption Techniques, the data is converted into cipher text which is not easily understood by unauthorized people and is stored either in plaintext or cipher text form on the cloud based on the encryption technique used & decryption converts the cipher text back into its original plaintext. For every query fired, the required data is first sent to the user machine from the cloud where decryption takes place and then computations are performed and then again encryption has to be performed to store the data back on cloud. This procedure is explained in Figure 1:

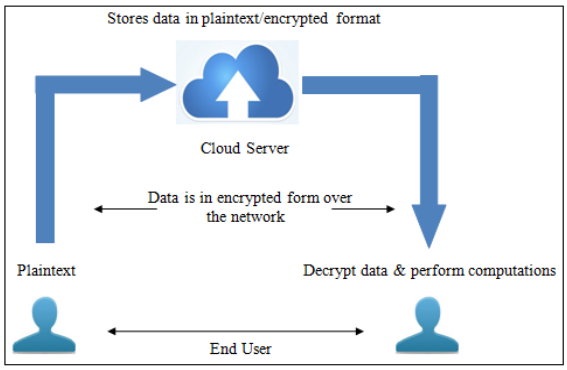

Figure 1: Existing System for Encryption on Cloud

The best way to secure cloud storage therefore would be to use homomorphic encryption technique. Homomorphic

 ISSN: 2321-8169
4180 - 4183

Encryption technique allows processing on data in its encrypted format. This means, instead of decrypting the data every time to perform a query, this technique will allow to perform query on the encrypted data itself and then the results can be decrypted by the user. In this way, the data remains encrypted during transfer, storage and computation as well.[4]

Somewhat Homomorphic Encryption (SHE) technique is a fully homomorphic encryption technique. A partially homomorphic technique includes only one kind of operation, addition or multiplication while fully homomorphic technique includes multiple types of operations.

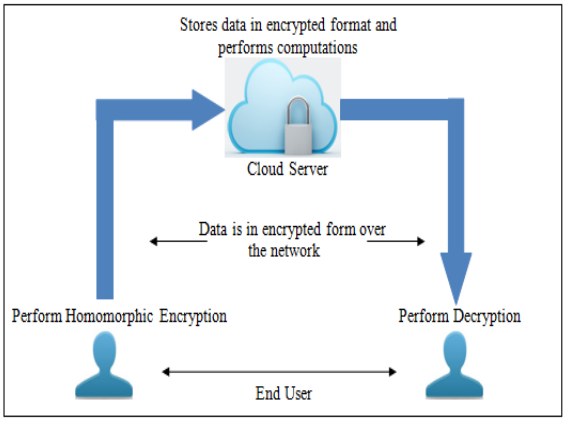

Figure 2: Proposed System for Encryption on Cloud

II. SHE Algorithm

The soft-oh notation $f(n) = \tilde{O}(g(n))$ is used to represent $f(n) = O(g(n)\lg^k g(n))$ for some k, ignoring the logarithmic factors and any other smaller additive complexities. The parameter p is used to denote the size of the plaintext integer to be encrypted, which may be taken as $O(n)$. In view of the correctness, homomorphism and security of the scheme, the suggested theoretical parameter setting is :

m= $\tilde{O}(n)$ = size of the plaintext
p = $\tilde{O}(n^3)$ = size of the primary secret key integer P,
r = $\tilde{O}(n^2)$ = size of the secondary secret key integer R,
g = $\tilde{O}(n^4)$ = size of the public key integers X_1,
s = 3n = size of the noise integer N used for encryption,
r'= n = size of the noise integer *R'* used to obtain the public key integers.

With this, the public key homomorphic encryption scheme over larger message space contains four steps as follows:

KeyGen ():
Choose a p-bit integer P, which is random odd , as primary secret key. Choose a r-bit random prime integers r. Choose two g-bit random integers Q_0, Q_1. Choose a r'-bit random integers R'. Compute $X_0 = PQ_0$, $X_1 = PQ_1 + RR_1$.Output secret key, sk = (p, r) and public key, pk = (X_0 , X_1).

Encrypt (pk, m):
Choose a m-plaintext integer M to be encrypted. Choose a s-bit random integer N_1 & N_2 .Compute $X_2 = [N_1X_1]$ mod X_0. Compute the cipher text as $C = (M + N_2X_2)$ mod X_0. Output the cipher text C.

Decrypt (sk , C):
Compute M = (C mod P) mod R and output M.

Evaluate (pk, f, ($c_1 \ldots c_k$)):
Let, f is the multivariate polynomial with k variables. Given k cipher texts ($C_1 \ldots C_k$) corresponding to the plaintext integers ($M_1 \ldots M_k$),compute f over them performing each addition and multiplication operations in f modulo X_0, and output the resulting ciphertext c. i.e.,for any two ciphertexts C_1, C_2 each multiplication and addition in f are performed as,mul (C_1, C_2) = (C_1C_2) mod X_0 and add (C_1, C_2) = ($C_1 + C_2$) mod X_0.The resulting ciphertext C is decrypted using Decrypt algorithm.[3]

Example computation:
Let us consider the scheme with a numerical example corresponding to the security parameter n = 5. It is to be noted that for the convenience in calculation and to show the homomorphic property of the scheme, we take s = n instead of 3n as specified in the scheme (only for this example).
$p = 5$, $e' = 25$, $e = 125$, $s = 5$, $r' = 5$, and $g = 625$

Let us take a 125-bit integer
P=34191175214920477472831036883018780493 and R be a 25-bit odd integer, R=17040277. Then the secret key will be SK = (P, R).

To generate public key , choose two 625 -bit random integers,
Q0=78057208462146889402840085484358807227420956857612105574377792736988362687959872449998141921395822603809622853162980661866764452795741519482656871056277432921710499781579113120408005421913, and

Q1=10761047832872489897585060713077815282324295777238945066360806446309701487731058483452855768864391924472340906031874204918829336230966581431408835145911351561851982048601285013016513803416 1.
Choose a R' 5-bit random integer '11'.

Compute,
$X_0=PQ_0=$
2668867691316837684159067587686169593178310818081559023606565261701096473702414547805432063490428056991486889160445703501791793070364255125568503477070919512896578056519500957484175084463864411078369492220913710120149199143109 , and

$X_1=PQ_1+RR'=$
3679328719498835931229921019579923558488532638878700298679754505189980543473606279815727298686097044135071652696675453884940765391509247158964090032066616987636108995027914869847603810784518211914786968437945520423857781864420.
Public key PK = (X_0, X_1)

***Encrypt*L**: Consider two 5-bit, signed, plaintext integers, say M_1 = **14**, M_2 = **15**. To encrypt M_1, we choose N_1 = 13, which is a 5-bit random integer,and get

$X_2=[N_1X_1]X_0=$
2460522601098626475284824263874123176319640398036600481525199118551107012215834324912109803581984604900654369329203940973769467893427875931868611306660389120027589974531377030787873104313041766559949221937758693467614778804607.

Now, choose N_2 = 14, a 5-bit random even integer. Then, we get

$C_1=[M+N_2X_2]X_0=$
2420904119578718444078728642003689350335235755533698458074004519302340486592705975104352488262647784711318500683506731611271033663619201539338516568394413525627322965205266941220122446816211798898855200477657187104816513547204 .

Similarly, to encrypt M_2,choose a 5-bit random integer N_1 = 13, and get

$X_2=[N_1X_1]X_0=$
2460522601098626475284824263874123176319640398036600481525199118551107012215834324912109803581984604900654369329203940973769467893427875931868611306660389120027589974531377030787873104313041766559949221937758693467614778804607,

Again, choose a 5-bit random even integer N_2 = 14, and compute

$C_2=[M+N_2X_2]X_0=$
2420904119578718444078728642003689350335235755533698458074004519302340486592705975104352488262647784711318500683506731611271033663619201539338516568394413525627322965205266941220122446816211798898855200477657187104816513547205

***Decrypt*L** : To decrypt compute (C_1 mod P) mod R =14. Similarly, (C_2 mod P) mod R =15.

***Evaluate*L**: Consider the function, $f(x,y) = x^3 + xy + y^3$. Now, we take $x = M_1 =$ **14**, and $y = M_2 =$ **15**. The plaintext resulting from the evaluation of the circuit with these values is,$M = f(M_1, M_2) = f($**14**,**15**$) = (-14)3 + (14 \times 15) + 15^3 =$ 6329.

Now, evaluation of the circuit over the ciphertexts C_1 and C_2 gives,
$C=f(C_1,C_2)=$
1087549790066933934307026398344412004888907195139189253150071224079731001254863194382787460665323394749850956060219594452519912300290449540476594634888988899791400578017264566426681673535567834417897279355541036027193401926029.

Decrypting the cipher text we get the value of evaluated function as, M=6329.

III. Key Management Protocol

The somewhat homomorphic encryption technique uses both the public and the private keys for the encryption and decryption respectively. Inorder to be able to do so, the user needs to store the key because it is practically not possible to remember the 64-bit key. Thus, the key has to be stored in a way that the attacker does not get access to them and thereby be able to view the user data. Inorder to launch any attack on the user data, the attacker needs to know the key to be able to decrypt the data and obtain it in the plaintext format. Although the attacker manages to get the public key, it should not be able to obtain the private key and thus we have to make use of the key management protocol.
The Key Management Protocol functions as given below:

The SHE algorithm runs on the user machine for the encryption and decryption of the data. While a user creates an account, a private key is generated for that particular user which will then be used for any further computations. Inorder to perform computations this private key is mapped with the password of the user which is obtained on every login and thus it is used for the purpose of decryption whereas encryption is done using the public key which varies every time. This private key is stored with the key management server in an encrypted format. ElGamal encryption scheme can be used for the purpose of encrypting the keys[7] . This method therefore helps in storing the keys securely but this security of key could be hampered if the attacker manages to obtain the file where the passwords are stored and therefore it is mandatory to store passwords in a way which will not allow the user to launch any attack. Also with the use of the hashing function using SHA technique[14] to hash the passwords and using salt along with the hash function to increase the noise, dictionary attack on the passwords can be avoided.

Let *p* be a given password. Then we generate a random salt value *s* and compute *y = h(p, s)* and store the pair *(s, y)* in the password file. Note that the salt *s* is not secret. To verify an entered password *z* ,we retrieve *(s, y)* from the password file, compute *h(z, s)*, and compare this result with the stored value *y*. [16] It is known that hashing is irreversible, and it is not possible for the attacker to obtain the password; which indirectly keeps the key safe. With the help of hash function, it becomes mandatory for user to remember the password or atleast try to remember it with the help of the password hint, without which the user will not be able to access the account and also all the data stored in it. This is a limitation in the key management protocol which does not allow the password to be changed and also provides no space for the user to forget the password.

The HKM protocol has a minimalistic and simple design. This has several advantages:

- It decreases the probability of execution mistakes.
- It allows clean modelling of protection and makes the method open to to analysis.
- It opens up the likelihood for improving the protocol with stronger security assurances.

IV. Conclusion

Thus Somewhat Homomorphic Encryption can be used with key management protocol which will make the algorithm more secure. It will be very difficult to crack the contents even if the attackers get access to the cloud or the files where the keys are stored. Also, the proposed method makes the processing of the encrypted data practical for suitable applications. It will be possible to achieve functionally loaded, usable and proficient computations on the encrypted data without hampering the assurance of privacy. Thus, scope and promises of homomorphic cryptography in cloud computing environments cannot be ignored. Researchers all over the world are taking great interest in recent years to develop homomorphisms that can be deployed practically.

V. Acknowledgement

The authors gratefully acknowledge the contributions of Prof. Manish Potey and thank him for his endless support and motivation, the college for providing the necessary infrastructure and platform for doing this research.

VI. References

[1] Ming Li, Shucheng Yu, Kui Ren, Wenjing Lou and Y.Thomas Hou, “Toward Privacy- Assured and Searchable Cloud Data Storage Services”, IEEE Network July/August 2013, pg 56-62.

[2] Maha TEBAA , Abdellatif EL GHAZI, “Homomorphic Encryption method applied to cloud computing”, IEEE 2012,pg 86-89.

[3] Y Govinda Ramaiah , G Vijaya Kumari. “Efficient Public Key Homomorphic Encryption Over Integer Plaintexts”, 2012 IEEE ,pp 123-128.

[4] Craig Gentry “Computing Arbitary Functions of Encrypted Data”, Communications of ACM,Vol.53 No.3, March 2010,pp 97-105.

[5] V.Miller. “Uses of Elliptic Curve in Cryptography”. CRYPT’85, LNCS218 ,pp417- 426,1986.

[6] R.L.Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public Key Cryptosystems”. Communications of ACM, Vol21.No.2,1978

[7] Alen Rosen, “Analysis of the Porticor Homomorphic Key Management Protocol” Oct 18, 2012 , Available : http://www.porticor.com/wp-content/uploads/2014/03/Porticor-homomorphic-KM-analysis.pdf

[8] Kristin Lauter , Michael Naehrig , Vinod Vaikuntanathan , “Can Homomorphic Encryption be Practical?” http://research.microsoft.com/pubs/148825/ccs2011_submission_412.pdf

[9] Cipher Cloud, ”Why Cloud Protection Ebook” Available : CipherCloud Online, http://www.ciphercloud.com/why-you-need-cloud-infomation-protection /

[10] CipherCloud “ciphercloud information protection” Available :CipherCloud Online, http://pages.ciphercloud.com/Guide-to-Cloud-Data-Protection.html

[11] Aderemi A. Atayero, Oluwaseyi Feyisetan, “Security issues in Cloud Computing : The Potentials of Homomorphic Encrpytion”,Journal of Emerging Trends in Computing and Information Sciences, Vol. 2, NO. 10, October 2011,pg 546- 552.

[12] Catteddu, D. and Hogben, G. Cloud Computing: benefits, risks and recommendations for information security. Technical Report. European Network and Information Security Agency, 2009

[13] D Eastlake 3rd, P Jones , “US secure hash algorithm 1 (SHA1)” Available : CipherCloud Online, http://www.hjp.at/doc/rfc/rfc3174.html

[14] William Stallings, “Cryptography and Network Security :Principles and Practice”, 5th Edition, Pearson education.

[15] Behrouz Forouzan , Debdeep Mukhopadhyay , “ Cryptography and Network Security”, 2nd Edition, Tata McGraw-Hill Education.

[16] Deven N. Shah, “Mark Stamp’s Information Security: Principles and Practice”, 2nd Edition, Wiley Education