Cloud Computing Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem

Reem Alattas & Khaled Elleithy
Computer Science & Engineering Department
University of Bridgeport
ralataas@bridgeport.edu & elleithy@bridgeport.edu

***Abstract*—Although cloud computing is growing rapidly, a key challenge is to build confidence that the cloud can handle data securely. Data is migrated to the cloud after encryption. However, this data must be decrypted before carrying out any calculations; which can be considered as a security breach. Homomorphic encryption solved this problem by allowing different operations to be conducted on encrypted data and the result will come out encrypted as well. In this paper, we propose the application of Algebraic Homomorphic Encryption Scheme based on Fermat's Little Theorem on cloud computing for better security.**

***Index Terms*—Cloud computing, homomorphic encryption, security, algebra homomorphism.**

Introduction

Cloud computing opens up a new world of opportunities, but mixed in with these opportunities are numerous security challenges that need to be considered and addressed. Among these challenges are availability, third party control, and data security. Data in the cloud is usually globally distributed which raises concerns about jurisdiction, data exposure, and privacy. If all data stored in the cloud was encrypted, that would effectively solve many issues. However, a user would be unable to leverage the power of the cloud to carry out computation on data without first decrypting it, or shipping it entirely back to the user for computation. The cloud provider thus has to decrypt the data first, perform the computation then send the result to the user.

Homomorphic encryption schemes allow the transformation of cipher-text C(m) of message m, to cipher-text C(f(m)) of a computation/function of message m, without disclosing the message. Therefore, the user could carry out any arbitrary computation on the hosted data without the cloud provider intervention.

In this paper, we propose applying Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem (AHEF) on cloud computing to solve the data security and third party control issues. AHEF is based on the concept of fully homomorphism and Fermat's little theorem.

This paper structure is as follows: related work and approaches are discussed in section II. Then, section III gives a brief overview of homomorphic encryption and introduces the application of AHEF on cloud computing. The scheme of the new methodology is described in Section IV. Finally, we give a short summary of our contributions in section V.

Related Work

In 1978, Ronald Rivest, Leonard Adleman and Michael Dertouzos introduced for the first time the concept of Homomorphic encryption. Since then, little progress has been made for almost 30 years. The encryption system of Shafi Goldwasser and Silvio Micali, that was proposed in 1982, was an additive Homomorphic encryption, but it could encrypt only a single bit. In the same notion, Pascal Paillier proposed a provable security encryption system in 1999 that was also an additive Homomorphic encryption. Few years later, in 2005, Dan Boneh, Eu-Jin Goh and Kobi Nissim invented a security system that can perform an unlimited number of additions but only one multiplication.

Most recently, Craig Gentry proposed the first fully homomorphic encryption scheme in 2009. That system evaluates an arbitrary number of additions and multiplications; and thus computes a function of any type on the encrypted data.

The application of fully homomorphic encryption is an important brick in cloud computing security. Generally, we could outsource the calculations on confidential data to the cloud, while keeping the secret key to decrypt the result of calculation.

HOMOMORPHIC ENCRYPTION

The proposed algebraic homomorphic encryption scheme is based on the concept of fully homomorphism, and uses a subset of it. It is also based on Fermat's little theorem and Fraction Module.

Fermat's little theorem is one of the four number theorems. It states that if p is a prime number, then for any integer a, the number $a^p - a$ is an integer multiple of p. In the notation of modular arithmetic, this is expressed as

$$a^p \equiv a (mod\, p)$$

If a is not divisible by p, Fermat's little theorem is equivalent to the statement that $a^{p-1} - 1$ is an integer multiple of p:

$$a^{p-1} \equiv 1 (mod\, p)$$

Fraction Module is simply a new operation. When discussing homomorphic encryption in this paper, we call this operation similar module operation, and use the symbol *smod* to present it.

Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem (AHEF)

Xiang and Cui came up with the Algebraic Homomorphism Encryption Scheme based on Fermat's Little Theorem (AHEF), which can be described as follows:

1) Select two large secure primes p and q. Let $N = pq$, such that p and q are secret, and N is public.
2) A rational number x can be expressed as the fraction form: $x=x_a/x_b$, such that the numerator x_a is an integer, and the denominator is a positive integer.
3) Select a random integer r. The encryption algorithm is $E(x)$, and the encrypted cipher text is:
$c=E(x)=fmod((x_a/x_b)^{r(p-1)+1}, N)$.
4) Decryption algorithm is $D(\ )$, such that $x = D(c) = fmod(c, p)$.

A fully homomorphic encryption scheme, such as AHEF, must respect both addition and multiplication operations as shown below.

Multiplicative Homomorphism: Let x and y be rational numbers, then AHEF meets the multiplicative homomorphism, i.e.
$E(xy) = fmod(E(x)E(y), N)$, or
$xy = D(E(x)E(y)) = fmod(E(x)E(y),p)$.

Additive Homomorphism: Let x and y be rational numbers, then AHEF meets additive homomorphism, i.e.
$E(x+y) = fmod(E(x)+E(y),N)$, or
$x+y = D(E(x)+E(y)) = fmod(E(x)+E(y),p)$.

A simple example to verify the nature of algebraic homomorphism of AHEF is given below.

Selecting $p = 173$, $q = 199$, then $N = pq = 34427$.

Let $x = 2.4$ and $y = -1.75$. Now, we will express x and y as fractions: $x = \frac{12}{5}, y = -\frac{7}{4}$

Then, we will randomly select $r_x = 17$, $r_y = 26$. AHEF can be used to encrypt x and y:

$$E(x) = fmod\left(\left(\frac{12}{5}\right)^{r_x(p-1)+1}, N\right) = \frac{28730}{18170}$$

$$E(y) = fmod\left(\left(\frac{-7}{4}\right)^{r_y(p-1)+1}, N\right) = \frac{-28379}{13671}$$

Multiplicative Homomorphism:

$$D(E(x)E(y)) = fmod(E(x)E(y), p)$$
$$= fmod(\frac{28730}{18170} \times \frac{-28379}{13671}, p)$$
$$= fmod(\frac{-815328670}{248402070}, p)$$
$$= \frac{smod(-815328670{,}173)}{smod(248402070{,}173)}$$
$$= \frac{-84}{20} = xy$$

Additive homomorphism:

$$D\big(E(x)+E(y)\big)=fmod(E(x)+E(y),p)$$

$$=fmod(\frac{28730}{18170}+\frac{-28379}{13671},p)$$

$$=\frac{smod(smod(28730\times 13671,p)+smod(18170),p),p)}{smod(18170\times 13671,p)}$$

$$=\frac{smod(smod(392767830,173)+smod(-515646430,173),173)}{smod(248402070,173)}$$

$$=\frac{smod(48+(-35),173)}{20}$$

$$=\frac{13}{20}=x+y$$

The security of AHEF algorithm is based on the difficulty of dividing by a large integer. Due to the random number being used in the encryption process, for the same plaintext *x*, the two encrypted results are not the same, i.e. *E1(x)* $\neq$ *E2(x)*, but *D(E1 (x)) = D(E2 (x))*.This feature guarantees that users can not infer the original data through statistical laws. More security properties can be found in [1].

AHEF SCHEME

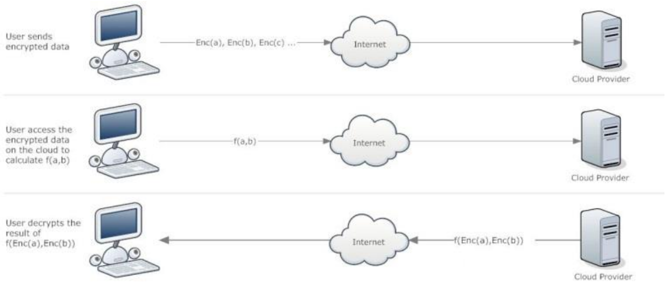
User sends encrypted data
Enc(a), Enc(b), Enc(c) ...
Internet
Cloud Provider
User access the encrypted data on the cloud to calculate f(a,b)
f(a,b)
Internet
Cloud Provider
User decrypts the result of f(Enc(a),Enc(b))
Internet
f(Enc(a),Enc(b))
Cloud Provider

Figure 1. AHEF Applied to Cloud Computing

As shown in figure 1, the process will start by sending the encrypted data to the cloud provider. The user can access the encrypted data on the cloud. Moreover, she can do calculations on that encrypted data, get the encrypted result. Then, decrypt the result on premise for better security.

SUMMARY

In this paper, AHEF algorithm was applied to cloud computing in order to carry out different calculations on encrypted data without decryption. The obtained result is encrypted as well and can be decrypted securely on premise.

REFERENCES

[1] Xiang Guangli; Cui Zhuxiao; , "The Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem," Communication Systems and Network Technologies (CSNT), 2012 International Conference on , vol., no., pp.978-981, 11-13 May 2012

[2] Tebaa, M.; El Hajji, S.; El Ghazi, A.; , "Homomorphic encryption method applied to Cloud Computing," Network Security and Systems (JNS2), 2012 National Days of , vol., no., pp.86-89, 20-21 April 2012

[3] Brenner, M.; Wiebelitz, J.; von Voigt, G.; Smith, M.; , "Secret program execution in the cloud applying homomorphic encryption," Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on , vol., no., pp.114-119, May 31 2011-June 3 2011

[4] 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.

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

[6] Craig Gentry, A Fully Homomorphic Encryption Scheme, 2009.

[7] 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.

[8] 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.

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

[10] Domingo-Ferrer J , Herrera-Joancomart i J. A new privacy homomorphism and applications [J]. Information Processing Letters, 1996, 60 (5) : 277-282.

[11] T. Sander and C. Tschudin. Towards mobile cryptography. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, 1998. IEEE Computer Society Press.

[12] N. Karnik. Security in Mobile Agent Systems. PhD thesis, Department of Computer Science and Engineering. University of Minnesota, 1998.

[13] Yao A.C. How to generate and exchange secrets[C].The 27th IEEE Symp on Foundations of Computer Science(FOCS) , Toronto,Canada:IEEE,1986:162-167

[14] Chen L.and Gao C.M.Public Key Homomorphism Based on Modified ElGamal in Real Domain[A].2008 International Conference on Computer Science and Software Engineering[C].Wuhan, Hubei, China: IEEE Computer Society,2008:802-805

[15] Xing G.L.,Chen X.M.,and Zhu P.,et al.A Method of Homomorphic Encryption[J]. Wuhan University Journal of Natural Sciences, 2006, 11(1):181-184.

[16] Zhu P.,He Y.X.,and Xiang G.L. Homomorphic encryption scheme of the rational[A]. 2006 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2006[C].Piscata way: IEEE Computer Society,2007:1-4

[17] Fontaine C.and Galand F.A Survey of Homomorphic Encryption for Nonspecialists[J]. EURASIP Journal on Information Security,2007,Vol.2007:1-9

[18] M. Ajtai. Generating hard instances of lattice problems (extended abstract). STOC ’96, pp. 99–108.

[19] M. Ajtai and C. Dwork. A public key cryptosystem with worst-case / average-case equivalence. STOC ’97, pp. 284–293.

[20] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. Eurocrypt ’02, pp. 83–107.

[21] F. Armknecht and A.-R. Sadeghi. A new approach for algebraically homomorphic encryption. Eprint 2008/422.

[22] L. Babai. On Lov´asz’s lattice reduction and the nearest lattice point problem. Combinatorica 6 (1986), 1–14.

[23] D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC1. STOC ’86, pp. 1–5.

[24] D. Beaver. Minimal-latency secure function evaluation. Eurocrypt ’00, pp. 335–350.

[25] J. Benaloh. Verifiable secret-ballot elections. Ph.D. thesis, Yale Univ., Dept. of Comp. Sci., 1988.

[26] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. SAC ’02, pp. 62–75.

[27] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. Eurocrypt ’98, pp. 127–144.

[28] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. TCC ’05, pp. 325–341.

[29] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-Secure Encryption from Decision Diffie-Hellman. Crypto ’08, pp. 108–125.

[30] D. Boneh and R. Lipton. Searching for Elements in Black-Box Fields and Applications. Crypto ’96, pp. 283–297.