Homomorphic Encryption Scheme & Its Application for Mobile Agent Security

Sachin Upadhye[1], P.G. Khot[2],

Research Scholer[1] , P.G.T.D., Deptt. Of Statistics, RTM Nagpur University, Nagpur,India
Professor[2], P.G.T.D., Deptt. Of Statistics, RTM Nagpur University, Nagpur

***Abstract*—** Mobile agents (MA) are autonomous software entities that are able to migrate across heterogeneous network execution environments. Protection of Mobile agents is one of the most difficult problems in the area of mobile agent's security. A security issues with mobile agents have not been solved and are becoming obstacles for the application of mobile agents. Four threat categories are identified for mobile agent: threats stemming from an agent attacking an agent platform, an agent platform attacking an agent, an agent attacking another agent on the agent platform, and other entities attacking the agent system Homomorphic encryption are a technique in which the encrypted mobile codes can be executed directly on different platforms without decryption.
In this paper section introduce the basic part of cryptosystem. In next section explain theory related Homomorphic Encryption. Section III contain the application view of Homomorphic Encryption used for mobile agent security and last conclude the paper.

***Keywords*—** Homomorphic encryption, mobile agent security, composite function

I. INTRODUCTION

A mobile agent is a software object that is not bound to the stem where it begins its execution. It has the unique ability transport itself from one system in a network to another. The ability to travel allows a mobile agent to move to a system at contains an object with which the agent wants to interact and then to take advantage of being in the same host or to work as the object. Mobile agents reduce network traffic, overcome network latency, encapsulate protocols, execute asynchronously and autonomously, adapt dynamically, naturally heterogeneous are robust and fault-tolerant.
The original idea of moving cryptography comes from calculating encrypted mobile agents directly, but as the homomorphic encryption scheme which supporting the idea of moving cryptography can't be found, so moving cryptography can't be used in practice. This is a compound method, organized by composite function and homomorphic encryption scheme. Both codes and data can be encrypted using this method, and the encrypted program can be executed directly without decryption. This method is an extension of moving cryptography put forward by Sander and Tschudin, which preserve many advantages and get rid of many drawbacks of original cryptography [3-6].

A. Encryption and Decryption

Encryption is the conversion of data into a form, called cipher text that cannot be easily understood by unauthorized people and decryption is the process of converting encrypted data back into its original form, so that the authorized recipient can understand it. According to Kerckoffs' principle [1], [2], security must rely upon the secrecy of the scheme, but not on the obfuscation of the code. A cryptography scheme is assumed to be publically known whereas the secret piece of information such as key is responsible for the secrecy of the scheme. According to key management, encryption schemes are of two types: Symmetric and Asymmetric encryption schemes.

B. Symmetric Encryption

An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message is called as Symmetric Encryption. Symmetric-key systems are faster, but their main drawback is that two parties wishing to communicate have to exchange the key in a secure way. In addition, scalability is problem as the number of users increase in the network. Due to its secret nature, symmetric-key cryptography is sometimes referred as secret-key cryptography.

C. Asymmetric Encryption

An encryption scheme is called asymmetric encryption if it uses two keys instead of one key as in symmetric encryption. One key encrypts the data and the other decrypts. It is also changeably

II. HOMOMORPHIC ENCRYPTION

During the last few years, homomorphic encryption techniques have been studied extensively and have found application in many different cryptographic protocols operating over open and untrusted networks. Untrusted networks are given only an encrypted version of the data. The network will perform computation on this encrypted data. To ensure that the encrypted data is really being processed securely was addressed by Rivest [7] through homomorphic encryption. However, this scheme has security flaws as pointed out by Brickell and Yacobi [8]. Ever since such schemes have been improved and implemented for practical purposes as in the case of secret sharing scheme, threshold scheme, electronic auction, commitment scheme, oblivious transfer, anonymity, privacy, electronic voting, multiparty computation,

zero knowledge proof, watermarking and fingerprinting [9], protection of mobile agent and mix-net. The scheme put forward in this paper is based on theory of three address code, homomorphic encryption scheme (HES) and composite function (FnC).

A. Three Address Code

Most original programs will be translated into executing objective codes using compiler. There are several phases before creating objective codes. Explicit middle forms will be created after grammatical analysis and semantic analysis, three address codes is one of the middle forms [10]. Three address codes are description of a series of strings,

e.g. x:=y op z

Here, x, y, z are names of constants or variables, op is a random operator. Usually, three addresses will be included in three address codes, two for operands, and one for result. So, original expression may change into following expressions:

$t_1 := y * z$

$t_2 := x + t_1$

t_1 and t_2 are temporary variables created by compiler.

B. Addition-multiplication homomorphic

Addition –multiplication homomorphic (AMH) is a subset of secret homomorphic. It is defined by Sander and Tschudin as following forms: Suppose R and S make a ring, then there is a encryption function E: R → S .

(a) Addition homomorphic means there is a valid algorithm PLUS to calculate E(x+y) according to E(x) and E(y), but don't need to know the concrete size of x and y.

(b) Multiplication homomorphic means there is a valid algorithm MULT to calculate E (xy) according to E(x) and E(y), but don't need to know the concrete size of x and y. Addition homomorphic and multiplication homomorphic keep back addition and multiplication separately [11-15], both secrecy homomorphic and addition-multiplication homomorphic may guarantee the security of arithmetic operation on encrypted data, and needn't to decrypt the data.

C. Composite Function

Composite function is defined as follows: it is consisted of output of h(x) and input of g(x), and shown as f(x) = g . h or f(x) = g(h(x)) in math, h(x) is the hidden original function. The agent host which owns function must choose a conversion matrix g(x) to create a composite function f(x). Compare f(x) with encrypted function h(x), f(x) is a different function. So, security and integrity of data get guarantee [16-18]. Because the result of composite function f(x) is encrypted, malicious host don't know the result of function. The owner of function (that is the owner of mobile agent) gets the encrypted result through function g(x).

Alice is the owner of agent and have function h(x), she wants to calculate the input x of Bob, but she won't want expose herself function, so she choose a function g(x), and create a function f(x), then send it to Bob. Bob calculates result through f(x) function using his input x, and send result to Alice. Bob can't calculate function h(x), because what he can see is just f(x). Only Alice can get the real result of h(x), through adding f(x) into inverse function, that is

$$h(x) = g^{-1}(f(x)).$$

D. Homomorphic Encryption Algorithm

KeyGen(λ)

1. Input the security parameter
2. Output a tuple (*sk*,*pk*) consisting of the secret key *sk* and public key *pk*

Encrypt (*pk* , π)

1. Input a public key *pk* and plaintext π
2. Output cipher text φ

Decrypt (*sk* , φ)

1. Input a secret key sk and cipher text φ
2. Output the corresponding plaintext π

Evaluate (*pk*, C, φ)

1. Input a public key *pk* a circuit C with t input (of the set C of allowed circuits) and a set φ of t cipher text $\varphi 1, \varphi 2, \ldots . \varphi t$
2. Output a cipher text φ

Therefore, a homomorphic encryption scheme consists of all algorithms of a conventional public key encryption scheme and an extra one. The correctness condition for the conventional part of a homomorphic encryption scheme is identical to that of a (non-homomorphic) public key encryption scheme. The additional algorithm Evaluate is supposed to do the following:

If φi is a cipher text correspond to the plaintext πi for i = 1 ...t and φ = ($\varphi 1, \ldots \varphi t$) , then Evaluate (pk, C, φ) shall return a cipher text φ corresponding to the plaintext C(π1,..... πt) circuits a ci C with t inputs. A homomorphic encryption scheme i said to correctly evaluate C (a set of circuits), if the correctness condition on the algorithm Evaluate from above hold for all circuits $C \in C$.

III. Application of Homomorphic Encryption in Mobile Agent

A. Mixed-Multiplicative Homomorphic Encryption Scheme (MMH)

Here we discuss about the security approach presented in [22] which we implemented. This approach focuses on extending the mobile cryptography approach, proposed in [23] in terms of privacy and integrity, and explore its usefulness and effectiveness in protecting mobile agents. To extend mobile cryptography, in [22], composite functions and additive-multiplicative homomorphic are considered to encrypt mobile agents. Homomorphic Encryption Scheme (HES) enables direct computation on encrypted data without decryption. Properties of HES that are needed to secure mobile agents are [22]:

additively homomorphic:
Computing E(x+y) from E(x) and E(y) without revealing x and y

multiplicatively homomorphic:
Computing E(xy) from E(x) and E(y) without revealing x and y

mixed-multiplicatively homomorphic:
Computing E(xy) from E(x) and y without revealing x.

The mobile agent encrypted with HES will be able to run on any host without decryption. Also, the HES encrypted agent will generate encrypted results, which will be decrypted by the agent owner. This will improve the overall security of the mobile agents. Computation on encrypted data protects the data from the untrusted hosts. But, the challenge is to find encryption schemes for arbitrary functions. We can find encrypting transformations for specific function classes such as polynomials and rational functions [22]. Also, an important observation made in [22] is that for computing with encrypted polynomial it is not necessary to have both the additive and multiplicative property of an encrypted function: it is sufficient that the encryption supports addition and "mixed multiplication" [22].

B. MMH (Mixed Multiplicative Homomorphic) Cryptosystem

MMH cryptosystem presented in [9] uses a large number, n, such that $n = p \times q$ where p and q are large prime numbers.
Let

$Zp = \{ x \mid x \leq p \}$ be the set of original plaintext messages

$Zn = \{ x \mid x < n \}$ be the set of cipher text message and

$Qp = \{a \mid a$ is not an element of $Zp \}$

be a set of encryption clues. The types of operations defined are addition and multiplication on Zp. The encryption and decryption algorithms are as follows:

Encryption: Given x is an element of Zp, pick a random number a in Qp such that $x = a \bmod p$. Compute the encrypted value $y = Ep(x) = a \bmod n$. (This can be accomplished by picking a random r and creating $a = x + rp$.)

Decryption: Given $y = Ep(x)$ is an element of Zn, use the key p to recover $x = Dp(y) = y \bmod p$. This cryptosystem is additively, multiplicatively, and mixed-multiplicatively homomorphic.

Example (Multiplication): Let $p = 17$, $q = 13$, $n = 221 = p \times q$ and the values, $x1 = 8$ and $E(8) = 59$ and $x2 = 2$ where $E(2) = 36$. $(59 \times 36) \bmod 221 = 135$ Decrypting 135 yields, $16 = 135 \bmod 17$,
which is the same as the unencrypted multiplication result
$x1 \times x2 = 8 \times 2 = 16$.

A mixed-multiplicative homomorphic allows encryption of a plaintext message without any knowledge of the cryptosystem including the keys and encryption algorithm. An advantage of this approach is that the encryption can be done in real-time, because the encryption of the plaintext, y, requires only a single invocation of the encryption function. One possible application of the mixed-multiplicative homomorphic encryption scheme is multi-party computation, where each party does not want to reveal its data to the other participants. A mixed-multiplicative homomorphic encryption scheme will allow each participant to encrypt inputs to a program, and perform the direct computation on the encrypted data. This scheme is protected against the cipher text-only attack due to the difficulty in factoring of a large prime number. But, it needs to be protected against the following attacks [24].

Known-Plaintext Attack: Cryptanalyst knows a plaintext cipher text pair (x, y). Since
$y = E(x) = (x + rp) \bmod n$, $rp \bmod n = E(x) - x \bmod n$.
So, p must be gcd(rp, n).

Integrity Attack: Since decryption is performed modulo p, any unencrypted number $x < p$ will be deciphered as itself. So, an encrypted value can be replaced with a chosen value and claim it to be encrypted

Automatic encryption of Remote input: By definition of the MMH, the remote input x, can be automatically encrypted by a malicious host by multiplying x by E(1) assuming if the agent owner provides E(1). No need to know the encryption algorithm

IV. Conclusion :-

In this paper we have shown the how the homomorphic encryption is useful for mobile agent security. First we explain the homomorphic encryption and then by using MMH (Mixed Multiplicative Homomorphic) Cryptosystem scheme, the computation is done on the encrypted data itself without decryption thus providing security. This scheme provides a new method to encrypt information

without any secret key. MA encrypted by MMH can execute tasks on other hosts of network without decryption, thus saves executing time, and it is effective to defend attacks of malicious hosts.

REFERENCES

[1] Kerckhoffs, A., (1883). "La cryptographie militaire (part i)", Journal des Sciences Militaires, Vol. 9, no. 1, pp. 5–38.
[2] Kerckhoffs, A., (1883). "La cryptography militaries (part ii)", Journal des Sciences Militaires, Vol. 9, no2pp161 191
[3] T. Sander and C.Tschudin. Protecting Mobile Agents against Malicious Hosts. Mobile Agents and Security, LNCS 1419, Berlin: Springer-Verlag 1998, 44-60.
[4] Ching Lin, Vijay Varadharajan. MobileTrust: a trust enhanced security architecture for mobile agent systems [J], International Journal of Information Security,2010,9(3): 153 – 178.
[5] Xiaogang Wang, Darren Xu, Junzhou Luo. A Free-Roaming Mobile Agent Security Protocol Based on Anonymous Onion Routing and k Anonymous Hops Backwards[A], Proceedings of the 5th international conference on Autonomic and Trusted Computing. Special Session Papers: Routing and Reliable Systems,LNCS, Jun.2008, Oslo, Norway, 588 – 602.
[6] Woei-Jiunn Tsaur, Chien-Hao Ho. A mobile agent protected scheme using pairing-based cryptosystems[J]. International Journal of Mobile Communications,2005,3(2):183-196.
[7] Rivest R., Adleman L. and Dertouzos M., (1978), "On data banks and privacy homomorphics", Foundations of Secure Computation, Academic Press, pp. 169 – 177.
[8] Brickell, E. and Yacobi, Y., (1987). "On privacy homomorphics", Advances in Cryptology (EUROCRYPT '87), volume 304 of Lecture Notes in Computer Science, Springer, New York, USA, pp. 117 – 126.
[9] Rappe, D., (2004), Homomorphic Cryptosystems and their Applications, Ph.D. thesis, University of DortmundDortmundGermany
[10] Ruchuan Wang, Xiaolong Xu. Research of MA security mechanism [J]. Chinese Journal of Computers,2002,25(12) : 1294-1301.
[11] Xiang Tan,Minqing Gu,Congming Bao. Mechanism for Mobile Agent Data Protection[J]. Journal of Software,2005,16(3): 477 - 484.
[12] Xiaoping Wu, Honggen Xing, Zhidong Shen. Research of MA security application model based on distributed confidence level [J]. Computer Engineering and Science,2010, 32(6):19-22.
[13] Rossilawati Sulaiman, Xu Huang, Dharmendra Sharma. E-health Services with Secure Mobile Agent[A], Proceedings of the 2009 Seventh Annual Communication Networks and Services Research Conference, Communications Networks and Services Research Conference, IEEE computer society,May.2009,270-277
[14] Monia Loulou, Mohamed Jmaiel, Mohamed Mosbah. Dynamic security framework for mobile agent systems: specification, verification and enforcement[J], International Journal of Information and Computer Security,2009,3(3/4):321-336.
[15] Christopher Colby, Karl Crary, Robert Harper, Peter Lee, Frank P fenning. Automated techniques for provably safe mobile code[J], Theoretical Computer Science,2003,290(2):1175-1199.
[16] Christopher Colby, Karl Crary, Robert Harper, Peter Lee, Frank P fenning. Automated techniques for provably safe mobile code [J], Theoretical Computer Science, 2003, 290(2):1175-1199.
[17] Carles Garrigues, Nikos Migas. Protecting mobile agents from external replay attacks[J]. Journal of Systems and Software. 2009, 82(2):197-206.
[18] D M Hein, R Toegl An Autonomous Attestation Token to Secure Mobile Agents in Disaster Response. LNICST 17, 2009, pp:46-57.
[19] Tomas Sander and Christian F. Tschudin, Towards Mobile Cryptography. Technical Report 97-049, International Computer Science Institute, Berkeley. 1997. http://wwwicsiberkeleyedu/~sander/publications/tr 97-049ps,
[20] Tomas Sander and Christian F. Tschudin, Protecting Mobile Agents Against Malicious Hosts, in Vigna, Giovanni (Ed.): Mobile Agents and Security, Springer-verlag, 1998.
[21] T.Sander and C. Tschudin. On software protection via function hiding. In Information hiding, pages 111-123, 1998],
[22] Hyungjick Lee and Jim Alves-Foss and Scott Harrison, The use of Encrypted functions for Mobile Agent Security from the Proceedings of the 37th Hawaii International Conference on System Sciences 2004
[23] Makoto Yokoo, Koutarou Suzuki. Secure Multi-agent Dynamic programming based on Homomorphic Encryption and its Application to Combinatorial Auctions.
[24] Hyungjick Lee and Jim Alves-Foss and Scott Harrison, The use of Encrypted functions for Mobile Agent Security from the Proceedings of the 37th Hawaii International Conference on System Sciences 2004