UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Study of MA protection based on Homomorphic Encryption and Composite Function Technology

Wu Jiehong[1] Yin Hang[1]
Engineering Training Center
Shenyang Aerospace University
Shenyang , China
wujiehong@sau.edu.cn

Zhang Po[2] Shi Xiangbin[2]
Computer College
Shenyang Aerospace University
Shenyang , China
wujiehong@sau.edu.cn

Abstract- **Mobile agents (MA) are autonomous software entities that are able to migrate across heterogeneous network execution environments. Mobility and autonomy compensate the deficiencies of distributed technology pretty well. But the security issues with mobile agents have not been solved and are becoming obstacles for the application of mobile agents. Homomorphic encryption is a technique in which the encrypted mobile codes can be executed directly on different platforms without decryption. This paper presents a protection scheme of mobile agent (AMHCFES) in network management application, which combining protect method of homomorphic encryption and composite function technology. The correctness and security proof of AMHCFES protection scheme are given in this paper, and malicious hosts can be avoided effectively using this scheme**

Keywords- mobile agent; homomorphic encryption; composite functio; active protection

I. Introduction

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.[1-5] This paper gives a practical scheme to realize the ideas of moving cryptography. 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[6-9].

II. Theory Related

The scheme put forward in this paper is based on theory of three address code, homomorphic encryption scheme (HES) and composite function(FnC).

2.1 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, one for result. So, original expression may changed into following expressions:

t1 : = y * z

t2 : = x + t1

t1 and t2 are temporary variables created by compiler.

2.2 Addition-multiplication homomorphic(AMH)

Addition –multiplication homomorphic 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 $E : R \rightarrow S$.

(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.

2.3 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). Figure 1 is as follows.

This work is supported by natural science foundation of China (61170185), natural science foundation of Liaoning province (20101082) ,and scientific research project of Liaoning education department(L2011030)

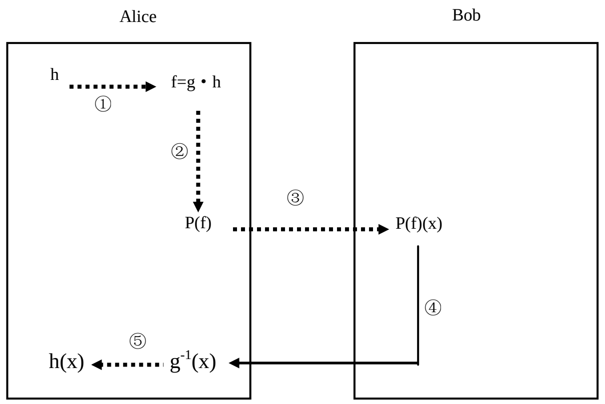

Fig.1 Composite Function

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

III. Addition Multiplication Homomorphic and Composite Function Encryption Scheme(AMHCFES)

First, data and state information of three address codes will be encrypted in this scheme, then three address codes operating code will be encrypted through composite function. Concrete operation steps are as follows:

Three address codes operands will be encrypted using addition-multiplication homomorphic encryption scheme.

Take away the operands dependance problem aroused by addition-multiplication homomorphic encryption scheme(take away encrypted data by addition-multiplication homomorphic)

Three address codes operating code statements will be encrypted using composite function technology.

Solve the operating code dependance problem aroused by composite function technology.

At first, original three address codes will be got from compiler in this scheme, and sensitive data and state information of mobile agents will be encrypted using AMH, then three address codes operating code will be encrypted using composite function technology after first encryption. Double encryption problems of operands and operating code will encounter during the first and second encryption, that will arouse the incorrect encryption of mobile agent. So AMHCFES will find all operands and operating code statements which have been double encrypted, and take away such statements. Double encryption problems will be solved to every encryption process. Encrypted three address codes will be created in AMHCFES, this codes execute the same task as original three address codes and get the same effect. But encrypted three address codes are hard for malicious host to read and modify mobile agents' code, data and state information.

3.1 AMHCFES Idea

The process description of encryption and decryption is given in Fig.2 . When encrypting, AMH is used to encrypt data, a simple function, that is $g(x) = x^3 + 1$ is used to encrypt operating codes. Then the encrypted mobile agent is sent to other hosts in network directly and results are returned to the original host. When decrypting, the reverse function $g^{-1}(x) = \sqrt[3]{x-1}$ is used first, then AMH scheme is used to get the real return results.

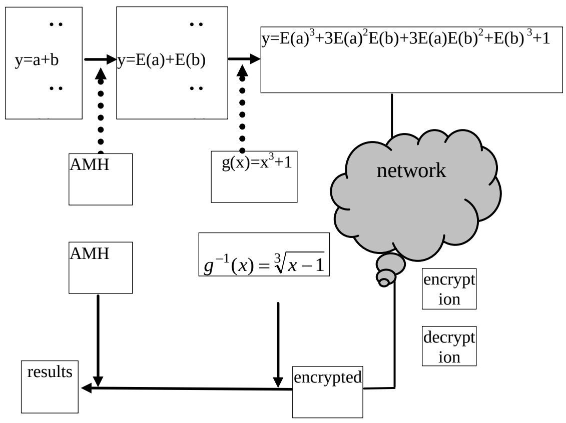

Fig.2 Encryption and Decryption Procedure

3.2 AMHCFES Algorithm

A great number n is used in the algorithm, which makes $n=p \times q$, p and q are prime numbers in this expression. Set $Z_p = \{x | x \le p\}$ as original plaintext information set, and set $Z_n = \{x | \ x < n\}$ as ciphertext information set, $Q_p = \{a | \ a \notin Z_p\}$ is the clue set of encryption.

Addition and Multiplication operating type are defined on Z_p.

The encryption and decryption algorithms are as follows:

(1) Encryption algorithm: To given $x \in Z_p$ choosing a random a in Q_p, and making x = a mod p. Encrypted result is calculated as y = Ep(x) = a mod n.(This is also can be done by choosing a random r and creating a expression a = x +rp.)

(2) Decryption algorithm: To given $y = E_p(x) \in Z_n$, using cryptographic key p to recover original value x = Dp(y) = y mod p.

A plain information x can be encrypted into more kinds of ciphertext in this cryptosystem. So, although $E_1(x) \neq E_2(x)$ but D(E1(x)) = D(E2(x)).

3.2.1 The Correctness Proof of Algorithm

Here, theorem 3.1 is given to show the correctness of algorithm, and the proof process will be given.

Theorem 3.1: To all $x \in Z_p$, existing Dp (Ep(x)) = x.

Proof: let $y = E_p(x)$ and random a is used to encrypt information, exists

a mod n = y (3.1)

Then divide n by p, equality (3.1) meaning

y mod p = (a mod n) mod p = x (3.2)

Done.

The proof of Algorithm AMHCFES's addition and multiplication properties which based on mod n is as follows:

Theorem 3.2: To all s and t in Zp , existing D(E(s)t) = D(E(st)).

Proof: Let's calculate E(s) t and E(s t) first.

(1) E(s) t : In order to encrypt s, let's choose a value a_1, make s = a_1 mod p, that is obtain the expression $a_1 = k_1 p + s$, as $y_1 = a_1 \bmod n$,so $a_1 = k_2 n + y_1$,and then get the expression $k_1 p + s = k_2 n + y_{11}$ .Thus, y_1 is obtained.

$$y_1 = k_1 p - k_2 n + s = (k_1 - k_2 q) p + s \tag{3.3}$$

As $E(s) = y_1, E(s)t = y_1 t$, get

$$y_1 t = (tk_1 - tk_2 q) p + ts$$

Obtaining the equality through decryption:

$$D(E(s)t) = D(y_1 t) = y_1 t \bmod p = st \tag{3.4}$$

(2) E(st): In order to encrypt st, let's choose a value a_2, make st = a_2 mod p, that is $a_2 = k_3 p + st$, so st is encrypted as $y_2 = a_2 \bmod n$, that is $a_2 = k_4 n + y_2$, Obtaining y_2 through calculation: $y_2 = (k_3 - k_4 q) p + st$

Obtaining equality through decryption:

$$D(E(st)) = D(y_2) = y_2 \bmod p = st \tag{3.5}$$

(3) D(E(s)t) = D(E(t)) : Obtaining the results from equality (4) and (5) :

$$y_1 t \bmod p = y_2 \bmod p = st$$

Which means, existing D(E(s)t) = D(E(t)) based on mod p. Done.

3.2.2 Examples of Algorithm AMHCFES

Any input random t can be encrypted automatically using this algorithm[19]. The auto encryption property of algorithm AMHCFES is shown by following example.

E.g. supposing p=101, q=71, then n = pq =7171. For the same reason, supposing host of mobile agent provides E(1) =203. Malicious host hope the input vnumber 8 encrypted, then letthe encrypted number 8 multiply E(1), and obtain ciphertext,that is E(8) =1624. In order to test the equality, choosing A=15966, then 15966 mod 7171 = 1624.

Attention : As A$\in$Qn..={ A$|$(A $\notin$ Zn) $\cap$ (A $\geq$ n)} , then 1624 mod 101 = 8.

3.3 Verification of AMHCFES Security

This scheme is decrypted by calculating x = y mod p, and the security of this algorithm may be tested by arguments on safety[20-21].

Ciphertext Attack: as y$\in$ Zq , if cryptanalysts want to find number A$\in$ Qn , they don't need to get p, but they need p to calculate a mod p = x. So, it is difficult for cryptanalysts to find p in module n, the same as factoring to n, if they know cipher only. So it is hard to find initial values only knowing cipher.

Plaintext Attack: if cryptanalysts know a pair of plaintext-cipher(x, y), then they create a data set of t, that is Ai$\in$ Qn, i = 1,⋯,t, so Ai = y mod n. To every i, there is Ai = x mod p, so p$|$(Ai – x), p = gcdti=1(Ai – x). But it is hard to let such thing happens.

Integrity Attack: Since module p is needed by all to execute decryption, any open data such as x < p can be used to decrypt data. So malicious host may choose a value to replace any encrypted data[22-24]. But such choice is blind as module p isn't known. It is hard to find initial values.

IV. CONCLUSION

This scheme provides a new method to encrypt information without any secret key. MA encrypted by AMHCFES can execute tasks on other hosts of network without decryption. It is effective to defend attacks of malicious hosts.

4.1 Shortcoming

There are some restrictions and assumptions in this scheme, which restrict application scope of the scheme.

Assumption 1: This scheme is used for Integer since AMH is based on loop theory.

Assumption 2: Control structure of MA codes can not be encrypted by composite function, because such operators are included in control structure, e.g. logical expression with other type operators in if statements, other operators here are: logical operator, Boolean operator, assignment operator, etc.

4.2 Improvement

Further study is needed to improve scheme, the work will be done in future is as follows:

Data set handled in this scheme is integer because of loop theory assumption. In order to further expand application scope of MA protection, researching how to expand data set to other data types is the study focus afterwards.

In MA encryption algorithm, type of function call are limited to some basic input output statements, so valid calling method of user defined function and system function will be studied.

REFERENCES

[1] D. Lange, O. Mitsuru. Seven good reasons for mobile agents [J]. Communications of the ACM, 1999, 42(3): 88-89.

[2] U. Topaloglu, C. Bayrak. Secure mobile agent execution in virtual environment[J]. Autonomous Agents and Multi-Agent Systems,2008,16(1): 1 – 12.

[3] Jason C. Hung, Han-Bin Chang, Hsuan-Pu Chang, Yu-Hsin Cheng, Kuo-Yen Lo. Evolution of ubiquitous autonomous agents[J], International Journal of Ad Hoc and Ubiquitous Computing,2009,4(6):334-343.

[4] Stefan Kraxberger,Peter Danner, Daniel Hein. Secure multi-agent system for multi-hop environments[A], Proceedings of the 5th international conference on Mathematical methods, models and architectures for computer network security, Security of multi-agent systems and software protection[C],LNCS, Sep. 2010,. St. Petersburg, Russia, 270 – 283.

[5] Richard Ssekibuule.Mobile Agent Security Against Malicious Platforms[J],Cybernetics and Systems,2010,41(7):522-534.

[6] T. Sander and C.Tschudin. Protecting Mobile Agents Against Malicious Hosts. Mobile Agents and Security, LNCS 1419, Berlin: Springer-Verlag 1998, 44-60.

[7] 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.

[8] 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.

[9] 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.

[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 Pfenning.Automated techniques for provably safe mobile code[J], Theoretical Computer Science,2003,290(2):1175-1199.

[16] Carles Garrigues, Nikos Migas. Protecting mobile agents from external replay attacks[J]. Journal of Systems and Software. 2009, 82(2):197-206.

[17] D M Hein, R Toegl. An Autonomous Attestation Token to Secure Mobile Agents in Disaster Response. LNICST 17,2009, pp:46-57.

[18] Joan Tomas-Buliart, Marcel Fernandez. Protection of Mobile Agents Execution Using a Modified Self-Validating Branch-based Software Watermarking with External Sentinel. Critical Information Infrastructures Security[C]. LNCS 5508, Berlin: Springer- Heidelberg, 2009, 287-294.

[19] S Venkatesan, C Chellappan, P Dhavachelvan. Advanced mobile agent security models for code integrity and malicious availability check[J]. Journal of Network and Computer Applications.2010,33(6):661-671.

[20] G. Vigna. Cryptographic traces for mobile agents[G]//LNCS 1419: Proceedings of Mobile Agents and Security .Berlin:Springer,1998:137-153

[21] F. Hohl. Time limited blackbox security: Protecting mobile agents from malicious host [A]. Mobile agent and security [C]. LNCS 1419, Springer-Verlag, 1998, 92-113.

[22] Weiwei Song, Zhen Ye, Lei Yue. Dynamic trust model and application in mobile agent environment [J]. Journal of Hefei University of Technology(Natural Science),2009,32(1):73-77.

[23] Dengguo Feng, Yu Qin. Research of proving method in credible calculating environment [J]. Chinese Journal of Computers,2008,31(9):1640-1652

[24] Apostolos P. Fournaris .Trust Ensuring Crisis Management Hardware Module[J], Information Security Journal: A Global Perspective,2010,19(2):74-83..