Mobile Agents Integrity Research

Huanmei Guan[1,2], Huanguo Zhang[2,3], Ping Chen[1], and Yajie Zhou[1]
1 Computer Center, Wuhan University, Wuhan 430072, China
hmguan-wh@163.com
2 College of Computer,Wuhan University, Wuhan 430072, China
liss@whu.edu.cn
3 The State Key Laboratory of Software Engineering, Wuhan University,
Wuhan 430072, China

Abstract. Mobile agents are an important technology in e-commerce systems and offer new possibilities for the e-commerce applications. This paper examines some mobile agent integrity protocols and proposes a new protecting protocol of mobile agent integrity. It can defend most known attacks, provides encryption transmission and route secrecy of mobile agents.

1 Introduction

Mobile agents are an important technology in e-commerce systems and offer new possibilities for the e-commerce applications. They can provide very flexible approach for information gathering on prices and assets available from the hosts they visit. They can create new types of electronic ventures from e-shops, e-auctions to virtual enterprises and e-marketplaces. Such systems are developed for diverse business areas, e.g., contract negotiations, service brokering, stock trading and many others([1]) .

Mobile agent systems have many advantages over traditional distributed computing environments: require less network bandwidth, increase asynchrony among clients and servers, and dynamically update server interfaces, and introduce concurrency and so on ([2]). But certain applications have a need for protection of security of the mobile agents. In the mobile agent systems the agent's code and internal state autonomously migrate between hosts and could be easy changed during the transmission or at a malicious host site. A malicious host may expose, modify, insert, delete or truncate data the agent collected from other previously visited servers to benefit itself ([3, 4]).

The integrity of an agent means that its code and execution state can not be changed by an unauthorized party or such changes should be detectable. The general goal is to protect the results within the chain of partial results from being modified ([5, 6, 7]). To protect integrity some protocols have been proposed in different papers. This paper will examine some protocols and extract general methods from these protocols. As result of this examination the paper will proposes a new integrity protocol for mobile agents. It can defend most known attacks, provides encryption transmission and route secrecy of mobile agents.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 describes the notations and security properties. Section 4 proposes a new integrity protocol for mobile agents. Section 5 gives security analysis of this protocol. Finally, conclusions are drawn in section 6.

2 Related work

Forward Integrity denotes the integrity of the partial results. Yee ([8]) defines the notion of weak forward integrity in the following mode "if a mobile agent visits a sequence of servers S_1, S_2, …, S_n, and the first malicious server is S_m, then none of the partial results generated at servers S_i, where $i < m$, can be forged". In their scheme, an agent and its originator maintained a list of secret keys, or a key generating function. The agent used a key to encapsulate the collected offer and then destroyed the key. However, a malicious host may keep the key or the key generating function. When the agent revisits the host or visits another host conspiring with it, a previous offer or series of offers would be modified, without being detected by the originator.

Karjoth ([9]), et al. proposed a notion of strong forward integrity where an attacker S_m can not forge any of partial results generated at server S_i, where $i < m$, even by colluding with one (or more) other visited server S_j, where $j < i$. In the their scheme, A chain O_0, O_1, O_2, …, O_n is an ordered sequence of encapsulated offers such that each entry of the chain depends on the previous and the next members. This dependency is specified by a chaining relation. Their scheme could resist the modification attack but could not prevent two colluders truncation attack. In this attack, a host with the agent at hand colludes with a previously visited host to discard all entries between the two visits.

Cheng ([10]), et al. proposed a data collection protocol that prevents two colluders truncation attack in a free roaming agent. The protocol is to require an external party, typically the preceding visited host, to co-sign the agent migration. Therefore, two colluders are not sufficient to affect a truncation attack. Their scheme can also be generalized to prevent the L ($L \geq 2$) colluder truncation attack. The co-signing mechanism But it could not prevent more than L colluders truncation attack.

Darren Xu ([11]), et al. proposed a scheme uses "one hop backwards and two hops forwards" chain relation as the protocol core to implement the generally accepted mobile agents security properties. This scheme can defend most known attacks. But if itinerary of mobile agents is protected, it difficult to find the second host forward.

3 Notations and security properties

Table 2. The Notation Used in This Paper

Notations	Meaning
(I_A, C_A, S_A, D_A)	I_A is A's identity, C_A is A's code, S_A is the state of A and D_A is A's data
$S_0 = S_{n+1}$	ID of the originator
$S_i, 1 \leq i \leq n$	ID of the host i
T	ID of the trusted third party
o_0	A secret possessed by host S_0. It can be regarded as a dummy offer and is only known to the originator
$o_i, 1 \leq i \leq n$	An offer from host S_i
$O_i, 0 \leq i \leq n$	An encapsulated offer (cryptographically protected o_i) from host S_i
$O_0, O_1, \ldots, O_n$	The chain of encapsulated offers from the originator and host $S_1, S_2, \ldots, S_n$
h_A	The agent integrity check value
$h_i, 0 \leq i \leq n$	Message integrity check value associated with O_i
$r_i, 0 \leq i \leq n$	A random number generated by host S_i
$(KD_i, KE_i), 0 \leq i \leq n$	A private/public key pair of host S_i
(KD_T, KE_T)	A private/public key pair of T
$Enc_{KE_i}(m)$	A message m asymmetrically encrypted with the public key KE_i of host S_i
$Dec_{KD_i}(m)$	A message m asymmetrically decrypted with the private key KD_i of T
$Sig_{KD_i}(m)$	The signature of host S_i on a message m using its private key KD_i.
$Verif(\sigma, KE_i)$	A signature verification function for signature σ and public key KE_i
$H(m)$	A one-way collision-resistant hash function
$A \rightarrow B: m$	A sending a message m to B

An agent is defined as $A = (I_A, C_A, S_A, D_A)$ where I_A is the identity, C_A is the code, S_A is the state and D_A is the data of the agent. Both I_A and C_A are static while S_A and D_A are variable.

Digital signature and encryption need a working public key infrastructure. Each host S_i has a certified private/public key pair (KD_i, KE_i). The transmission of mobile agents is encrypted. An agent's route information is secret. The main technique is to require a trusted third party.

Assume that an agent has visited an undetermined number m of hosts, $m \leq n$. An agent is captured by an attacker. This attacker possibly is the host S_{m+1}. Some hosts excluding S_m may collude with the attacker. Let i range over 1, …, m. Mobile agents security properties based on the assumptions:

- Verifiable Forward Integrity: The trust third party T can verify the offer o_i by checking whether the chain is valid at O_i.
- Data Confidentiality: Only the originator can extract the offers o_i from the encapsulated offers O_i.
- Non-repudiability: Host S_i cannot deny submitting o_i once it has been received by originator S_0.
- Forward Privacy: None of the identities of the creator of offer o_i can be extracted.
- Strong Forward Integrity: None of the encapsulated offers O_k, where $k \leq m$, can be modified.
- Insertion Resilience: No offer can be inserted at i unless explicitly allowed, i.e., S_{m+1}. It is not possible for S_{m+1} to insert more than one offer even if S_{m+1} collude with some specific L hosts.
- Deletion Resilience: No partial result O_k can be deleted by any S_i, with $k < m$. It is not possible for S_{m+1} to delete more than one offer even if S_{m+1} collude with some specific L hosts.
- Truncation Resilience: Truncation at i is not possible.
- Itinerary Secrecy: Only the originator and the trusted third party T know a mobile agent's migration route. Truncation at i is not possible even if some specific L hosts collude with S_i to carry out the attack.
- Secure Transmission.

4 The Protocol

4.1 Agent at the originator S_0:

S_0: $O_0 = \mathrm{Sig}_{KD_0}(\mathrm{Enc}_{KE_0}(o_0, r_0))$

$$h_0 = \mathrm{Sig}_{KD_0}(H(O_0), \mathrm{Enc}_{KE_T}(S_1))$$

$$h_A = \mathrm{Sig}_{KD_0}(H(I_A \parallel C_A))$$

$S_0 \rightarrow T:\ h_0$

$S_0 \rightarrow S_1:\ \mathrm{Enc}_{KE_1}(I_A \parallel C_A \parallel S_A \parallel D_A), h_A, O_0$

4.2 Agent at host S_1:

S_1: $\mathrm{Dec}_{KE_1}(I_A \parallel C_A \parallel S_A \parallel D_A)$

$$\mathrm{Verif}(h_A, KE_0) \overset{?}{=} \mathrm{true}$$

$$O_1 = Sig_{KD_1}(Enc_{KE_0}(o_1, r_1))$$
$$h_1 = Sig_{KD_1}(H(O_1), Enc_{KE_T}(S_2))$$

$$S_1 \rightarrow T : h_1$$
$$S_1 \rightarrow S_2 : Enc_{KE_2}(I_A \| C_A \| S_A \| D_A), h_A, \{O_0, O_1\}$$

4.3 Agent at host S_i:

S_i: $Dec_{KE_i}(I_A \| C_A \| S_A \| D_A)$

$$Ver(h_A, KE_0) \overset{?}{=} true$$
$$O_i = Sig_{KD_i}(Enc_{KE_0}(o_i, r_i))$$
$$h_i = Sig_{KD_i}(H(O_i), Enc_{KE_T}(S_{i+1}))$$

$$S_i \rightarrow T : h_i$$
$$S_i \rightarrow S_{i+1} : Enc_{KE_{i+1}}(I_A \| C_A \| S_A \| D_A), h_A, \{O_k \mid 0 \le k \le i\}$$

4.4 Agent at host S_n:

S_n: $Dec_{KE_n}(I_A \| C_A \| S_A \| D_A)$

$$Ver(h_A, KE_0) \overset{?}{=} true$$
$$O_n = Sig_{KD_n}(Enc_{KE_0}(o_n, r_n))$$
$$h_n = Sig_{KD_n}(H(O_n), Enc_{KE_T}(S_{n+1}))$$

$$S_n \rightarrow T : h_n$$
$$S_n \rightarrow S_{n+1} : Enc_{KE_{n+1}}(I_A \| C_A \| S_A \| D_A), h_A, \{O_k \mid 0 \le k \le n\}$$

4.5 Agent at host S_{n+1} ($S_{n+1} = S_0$):

S_{n+1}: $Dec_{KE_{n+1}}(I_A \| C_A \| S_A \| D_A)$

$$Ver(h_A, KE_0) \overset{?}{=} true$$

$S_{n+1} \rightarrow T$: $\{h'_k = H(O_k) \mid 0 \le k \le n\}$, T verifies the forward integrity and returns results to host S_{n+1}

4.6 At the trusted third party T:

T: $Verif(h_i, KE_i)$, recover $H(O_i)$, $Enc_{KE_T}(S_{i+1})$

$$\mathrm{Dec}_{Kd_T}(S_{i+1}), \text{recover } S_{i+1}$$

$$\text{Receive } h'_0, h'_1, h'_2, ..., h'_n$$

$$h'_k \overset{?}{=} H(O_k), (0 \le k \le n)$$

To begin the protocol, the originator S_0 randomly generates r_0. Host S_0 encrypts a dummy offer o_0 and r_0 using its own public key KE_0. Host S_0 signs this encrypted value to construct a dummy encapsulated offer O_0. Next, Host S_0 calculates a hash value h_0 from O_0, and encrypts S_1 using T's public key KE_T, and then signs them. Host S_0 also computes a hashed value h_A from I_A and C_A. h_A is the certified agent integrity checksum. Host S_0 encrypts this agent using it's the next host's public key KE_1. Finally, Host S_0 sends h_0 to the trusted third party T and the agent migrates to the first host S_1.

When the agent arrives at host S_i, S_i verifies h_A in order to ensure the identity I_A and code C_A were not modified by any malicious hosts. Host S_i randomly generates r_i. Host S_i encrypts o_i and r_i using the originator's public key KE_0. Host S_i signs this encrypted value to construct an encapsulated offer O_i. Host S_i calculates a hash value h_i from O_i, and encrypts S_{i+1} using T's public key KE_T, and then signs them. Finally, Host S_i sends h_i to the trusted third party T and the agent migrates to host S_{i+1}.

When the agent returns host S_{n+1} ($S_{n+1} = S_0$), S_{n+1} verifies h_A again. Host S_{n+1} computes a hash value h'_k from O_k ($0 \le k \le n$), then sends h'_k to the trusted third party T and requests T to verify the forward integrity.

The trusted third party T receives h_i, recovers $H(O_i)$ and S_{i+1}. The chain of hash $H(O_0)$, $H(O_1)$, $H(O_2)$, …, $H(O_n)$ is an ordered sequence. S_1, S_2, …, S_n is the agent's route information.

T receives h'_k ($0 \le k \le i-1$). It compares h'_k with $H(O_k)$, so as to ensure O_k was not altered. Then T returns results to host S_{n+1}.

5 Security Analysis

Here we analyze how the protocol achieves the security properties.

- Verifiable Forward Integrity: The trust third party T fulfills the forward integrity for each host S_i.
- Data Confidentiality: If the encryption scheme is secure, only the originator S_0 can decrypt $\mathrm{Enc}_{KE_0}(o_i, r_i)$ to extract o_i. The trusted third party T and other hosts cannot gain o_i.
- Non-repudiability: Each host S_i signs its offer o_i by its private key KD_i. If the signature scheme is secure, host S_i cannot repudiate O_i.
- Forward Privacy: The host identity S_i is encrypted using the trust third party T's public key. Only T can extract the identity of S_i. T saves the agent migrate route.
- Strong Forward Integrity: Suppose the attacker leaves O_m intact but changes O_k to O'_k, where $0 \le k \le m-1$. S_{n+1} will calculate h'_k from O'_k and send h'_k

to T. In the trusted third party T, Since $h_k^{'}$ not equal $H(O_k)$, T will report this attack to S_{m+1}.

- Insertion Resilience: Suppose the attacker leaves O_m intact but inserts a $O_k^{'}$ before O_k, where $0 \le k \le m-1$. Following similar reasoning as in the above analysis, S_{n+1} will calculate $h_k^{'}$ from $O_k^{'}$ and send $h_k^{'}$ to the trust third party T. Through comparing $h_k^{'}$ with $H(O_k)$, T will find this change. Therefore no offer can be inserted in the chain of encapsulated offers.
- Deletion Resilience: Suppose the attacker leaves O_m intact but deletes O_k, where $0 \le k \le m-1$. S_{n+1} will calculate $h_k^{'}$ from O_{k+1} and send $h_k^{'}$ to the trust third party T. Through comparing $h_k^{'}$ with $H(O_k)$, T will find this change.
- Truncation Resilience: Suppose the attacker leaves O_m intact but deletes O_k, where $0 \le k \le m-1$. S_{n+1} will calculate $h_k^{'}$ from O_{k+1}. Similarly, T will find this modify. In the other words, if the T is secure, Collusion attack is infructuous.
- Itinerary Secrecy: Only the originator and the trust third party T know a mobile agent's migration route.
- Secure Transmission: The transmission of mobile agents is encrypted.

6 Conclusions

This paper examined some protocols and gives security requirements in mobile agent systems. As result of this examination the paper will proposes a new integrity protocol for mobile agents. It can defend most known attacks, provides encryption transmission and route secrecy of mobile agents.

Acknowledgement

This research is Supported by the National Natural Science Foundation of China (60673071)

References

1. A. Corradi, M. Cremonini, R. Montanari and C. Stefanelli, Mobile Agents Integrity for Electronic Commerce Applications, *Information Systems* Vol. 24, No. 6, pp. 519-533 (1999).
2. D.B. Lange and M. Oshima, Seven Good Reasons for Mobile Agents, *Communications of the ACM*, 1999, 42(3): 88−89.
3. W. Jansen, Countermeasures for Mobile Agent Security, *Computer Communications*, 2000, 23(17): 1667−1676.
4. S. Fünfrocken, Protecting Mobile Web-Commerce Agents with Smartcards, *First International Symposium on Agent Systems and Applications* (ASA'99)/Third International Symposium on Mobile Agents (MA'99), Palm Springs (1999).
5. D. Westhoff, et al, *Protecting a Mobile Agent's Route against Collusions,* Proceedings of SAC'99, San Antonio: Springer-Verlag, 1999: 215−226.
6. M. Giansiracusa, Mobile Agent Protection Mechanisms and the Trusted Agent Proxy Server (TAPS) Architecture,
(2003). http://www.isi.qut.edu.au/research/publications/technical/qut-isrc-tr-2003-010.pdf.

7. P. J. Hardjono and J. Seberry, Fundamentals of Computer Security, *Springer-Verlag*, Berlin (2003).
8. B.S. Yee, *A Sanctuary for Mobile Agents*, Proceedings of Secure Internet Programming: Security Issues for Distributed and Mobile Objects, Berlin: Springer-Verlag, 1999: 261–273.
9. G. Karjoth, N. Asokan, and C. *Gülcü, Protecting the Computation Results of Freeroaming Agents*, Proceedings of the 2nd International Workshop on Mobile Agents (MA '98), Stuttgart: *Springer-Verlag*, 1998: 195−207.
10. J.S.L. Cheng and K.W. Victor, *Defenses against the Truncation of Computation Results of Free-roaming Agents*, Proceedings of the 4th International Conference on Information and Communications Security, Singapore: Springer-Verlag, 2002: 1−12.
11. X. Darren, *An improved free-roaming mobile agent security protocol against colluded trunation attack*, Proceedings of the 30th Annual International Computer Software and Applications Conference (2006).