Securing Your Data in Agent-Based P2P Systems

Xiaolin Pang[1] Barbara Catania[2] Kian–Lee Tan [{1,3}]
[1]Department of Computer Science, National University of Singapore,Singapore 117543
[2]Department of Computer and Information Sciences, University of Genoa,Genoa, Italy, 16146
[3]SMA, National University of Singapore, E4-04-10, 4 Engineering Drive 3, Singapore 117576
{pangxiao, tankl} @comp.nus.edu.sg, catania@disi.unige.it

Abstract

Peer-to-peer (P2P) technology can be naturally integrated with mobile agent technology in Internet applications, taking advantage of the autonomy, mobility, and efficiency of mobile agents in accessing and processing data. In this paper, we address the problem of protecting critical information in agent-based P2P Internet applications under two different scenarios. First, we assume the route of a mobile agent in the P2P system is fixed. Under this assumption, we propose the usage of an efficient parallel dispatch model where the agent's route is signcrypted at the first step and dispatched to each new peer to collect information. Then, we assume the route is not specified and we propose the usage of a modified multi-signcryption scheme to guarantee protection. Based on this second approach, a mobile agent determines the next peer to communicate with independently and information is collected dynamically in one round of visiting a group of peers. Security issues under the two proposed models are then discussed.

1 Introduction

Peer-to-peer (P2P) technology, also known as peer computing, is an emerging paradigm that is now viewed as a potential technology that could provide a decentralized infrastructure for information sharing. Unlike the traditional client-server model, where there is typically a single or small cluster of servers and many clients [1], each node is treated as a peer in a P2P system and each peer can both consume as well as provide data and/or services. In addition, each peer may join and leave the P2P network at any time, resulting in a truly dynamic and ad-hoc environment [2].

The P2P technology can be naturally integrated with mobile agent technology for applications on the Internet, taking advantages of the autonomy, mobility, and efficiency of mobile agents in accessing and processing data. Indeed, while P2P provides a dynamic distributed infrastructure, mobile agents guarantee the agility and mobility to applications. A mobile agent can act on behalf of its owner to migrate through the distributed P2P network, access data, perform computations, and send results back to its owner. This approach reduces the traffic volume caused by broadcasting and redundant processing, thus reducing the overall query [9] and update cost [10]. Electronic commerce applications are an example in which P2P and mobile agent technologies could be successfully exploited [3,4,5].

While the P2P domain might seem exciting and promising, measures of trust and security must be applied to each peer to establish a secure connection for secure computing in such a distributed environment. Firstly, the connection between peers must be secure, which requires at least the capability of each peer to identify the other participant in the connection. Secondly, the sensitive data managed or exchanged via applications must be protected. The trust and security properties are established by using various techniques such as authentication of peers communicating with each other or with any other entity involved in the P2P application, authorization of certain entities to do some action or access some information, encryption of sensitive information flowing between peers over an unsecured network.

Security is an even more important issue when the critical data is carried by a mobile agent [6,7,8]. Indeed, while agents can be used to extract data for query purposes, the agents are prone to attack and hence the security of data in the agent is of prime concern in P2P database applications using agent technology.

In this paper, after motivating the usage of mobile agents in P2P systems, we examine data security problems in agent-based P2P environment, particularly in the context of BestPeer [2,3], an agent-based P2P system designed to serve as a platform on which P2P

applications can be easily and efficiently developed, and we propose two security models, with different degree of dynamism and security. In the first approach, based on an efficient parallel dispatch model, the route of a mobile agent in the P2P system is fixed; in particular, it is signcrypted [11,12] at the first step and dispatched to each of the new peer to collect information. In the second approach, a mobile agent collects information dynamically in one round of visiting a group of peers; in this case, the route of a mobile agent is not specified and the next peer to communicate with is determined dynamically based on some statistical or historical information.

The rest of this paper is organized as follows. Section 2 briefly introduces the BestPeer P2P system whereas Section 3 motivates the usage of mobile agents in BestPeer. Section 4 presents two approaches for protecting information in agent-based P2P systems such as BestPeer, based on two different models. Security issues are then discussed in Section 5. Finally, Section 6 presents some conclusions and outlines future work.

2 BestPeer – A P2P Based Data Management System

BestPeer [2,3] is a generic P2P system designed to serve as a platform on which P2P applications can be easily and efficiently developed, integrating two powerful technologies: mobile agents and P2P technologies. While P2P technology provides resource-sharing capabilities among nodes, mobile agent technology further extends these functionalities. In particular, since agents can carry both code and data, they can effectively perform any kind of functions. With mobile agents, BestPeer provides more than just files and raw data, but also processed and meaningful information. For example, in BestPeer, an agent can be sent to a peer which has the data file of content to "digest" and to generate reports for the requester.

In a BestPeer network, there are two types of entities: a large number of computers (nodes or peers), and a relatively fewer number of Location Independent Global names Lookup (LIGLO) servers. Each participating peer runs a Java-based software that enables it to communicate or share resources with any other peers in the BestPeer network, thus realizing a P2P distributed object management and sharing environment. Each peer is essentially an object management system and retains its autonomy: it determines its degree of participation, i.e., which objects/services to share with other nodes, amount of resources to share, and access control. Figure 1 illustrates the major components of BestPeer. Each peer contains an object management system, a global dictionary, and an agent manager for managing and executing mobile agents.

In BestPeer networks, LIGLO servers are used to uniquely identify peers whose IP addresses may change due to dynamic IP addressing and as a result of frequent connections to and disconnection from the BestPeer network. Through the LIGLO servers, a peer knows exactly who the other peers are; otherwise, the same peer with a different IP address each time it joins the network may be considered as a 'new' participant. Like existing P2P systems, each BestPeer node maintains addresses of a set of nodes that it can directly reach. It also maintains meta-data of objects/services provided by its neighbour peers. If a request can be satisfied locally at a node, it is done; if it can be satisfied by some of its neighbour peers, it is routed to them; otherwise, the request is routed to all neighbour's neighbour peers, which in turn may route it to their neighbours, and so on.

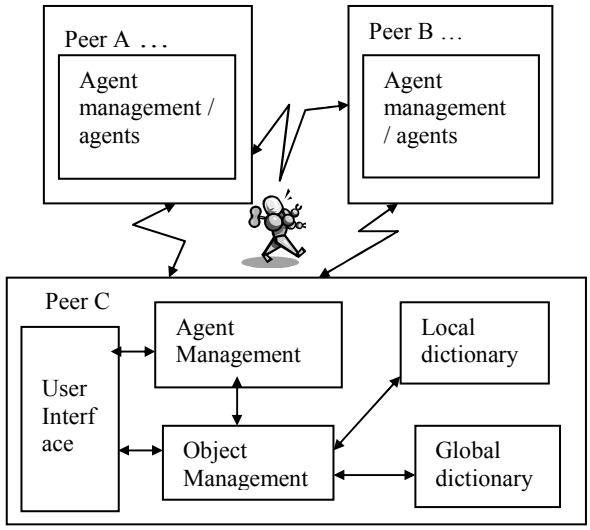

Figure 1. BestPeer Architecture

3. Efficiency of Mobile Agents in P2P Systems

The usage of mobile agents in P2P systems such as BestPeer is motivated mainly by efficiency reasons. Suppose a query must be executed on some dataset and assume that such query requires the execution of some functions that are not supported by the local DBMS. As such, the operation cannot be pushed down to the DBMS. Instead, the data have to be first retrieved, and the operation has to be performed on the data before the answers to the query can be obtained. Such process can be supported either by a message-based protocol or an agent-based protocol, as shown in Figures 2 and 3. In the first case, since the message-based protocol is a data-shipping strategy, remote data are transferred to the query node to be processed there. In the second case, since the agent-based approach is a code-shipping

strategy, the agent carries the processing code to the remote peer and performs remote execution. Only (partial) answers produced by the agent are then returned.

In order to validate the benefit of using mobile agents, we compared the query processing performance of an information retrieval system based on agents against the performance of a system based on message passing and information pulling. To this purpose, a query agent has been implemented on top of the BestPeer P2P system [2,3]. By assuming the query requires only one remote access, the whole query process is divided into three phases: (i) the message (message-passing protocol) or the agent (agent-based approach) is sent to the remote peer; (ii) the remote peer processes the request; (iii) the remote peer returns the result to the originator. The answer size is set to be 0.1% of the whole data set. The total response time includes the cost of data transfer, i.e., message, code and data, and processing time. From the obtained results, shown in Figure 4, it follows that the completion time of the message-based protocol increases exponentially when the data size increases. The overhead of the data-shipping results in a longer response time performance. As a result, when the size of data to be transferred across the network increases, the mobile agent-based protocol is superior. In general, when the size of the answer requested by a query increases , which typically is the case in most commercial applications, the agent based approach provides significant saving in communication costs. However, while agents are a good means in extracting data for the query, the agents are prone to attack and hence the security of data in the agent is of prime concern in database applications using agent technology.

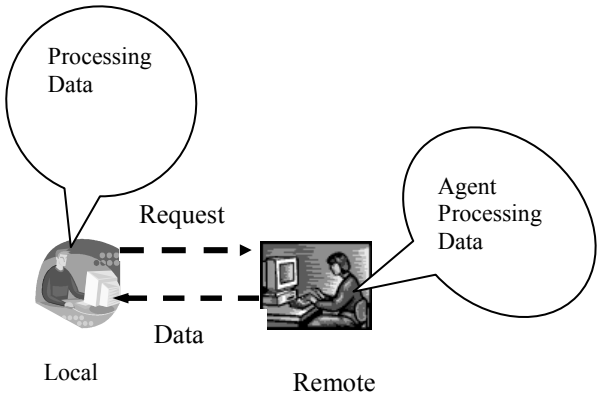

Figure 2: Message-Based Protocol

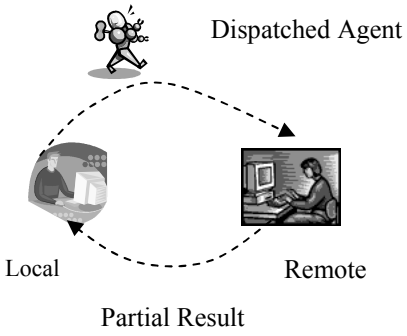

Figure 3: Agent-Based Protocol

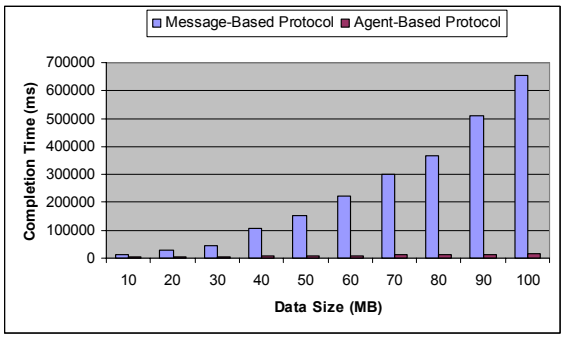

Figure 4: Completion Time vs. Data Size

4. Protecting Information in Agent-based P2P Systems

To authenticate the action of a mobile agent in the P2P system and protect the collected data, we assume each peer owns a public/private key pair and the corresponding key pair certificate, and uses the asymmetric key pair to perform signcryption operation [11,12], which has a significantly less computation cost than traditional "signature-then-encryption" technology [13].

In the following, we present two models for protecting information in agent-based P2P systems such as BestPeer, using different approaches for collecting information among peers. One is a parallel dispatch model, where a pool of mobile agents is securely dispatched to a group of peers in a hierarchical way, starting from an initial peer. Each agent collects information from each peer and sends back the secured information to the initial peer. The other is a serial dispatch model, where an autonomous mobile agent is created by the original peer and dispatched to visit each new peer dynamically in one round. After finishing the task of collecting information, the mobile agent returns back the result within a multi-signcrypted message.

4.1 Protecting Information based on Parallel Dispatch Model

We first present a parallel dispatch model for mobile agents [16], then we focus on protecting the sensitive information carried by mobile agents based on the considered model.

In the following, for simplicity, our discussion is restricted to a binary dispatch model where an agent in a peer can dispatch exactly two other agents, resulting in a binary tree structure. However, the model can be easily generalized to dispatch multiple (more than two) agents. As shown in Figure 5, A_{MA} (a master agent in one certain peer) is willing to communicate with other peers to obtain information. An agent working on behalf of a peer is responsible for dispatching PWAs (Primary Worker Agents) and distributing tasks to them. A PWA is a special WA (Worker Agent) that should dispatch other mobile agents. A WA is only responsible for locally accessing information and sending back data/answers to the master agent. A PWA can also have the task of performing information access depending on the application.

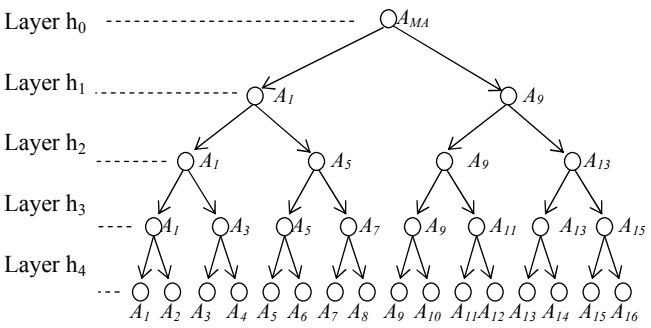

Figure 5. Dispatch Tree with 16 PWAs

4.1.1 Initialization of Mobile Agent

Under the proposed approach, the route of an agent is signcrypted by the public key of a peer CP, to which the agent should be dispatched, and private key of MA. The *Signcrypted Initiation Data* (SID) describes the signcrypted information of mobile agents to be dispatched. It comprises the description of mobile agent's tasks, including dispatching tasks or data-accessing task or both. For a PWA, the SID includes the IP address of its right child if it is a PWA and SIDs for the right and left child (itself). The signature in SID signed by MA can be used for checking integrity and preventing forgery and attacks. Note that in a P2P system, the dynamic IP address must be resolved dynamically and determined based on Network Address Translation (NAT) [1]. In BestPeer, LIGLO servers can be used for this purpose.

More formally, the SID structure for a mobile agent A to be dispatched at peer CP under the binary dispatch model, signcrypted by the public key of CP and private key of MA, is the following:

- If A is a PWA and the right child is a PWA, $SID(CP)=ES_{CP}$[PWA, Token, ip(RP), SID_L, SID_R, ip(PP), ip(CP), Certificate, Code, t], where Token equals PWA.
- If A is a PWA, the right and the left child are WAs, $SID(CP)=ES_{CP}$[PWA, Token, ip(LP), ip(RP), SID_L, SID_R, ip(PP), ip(CP), Certificate, Code, t], where Token equals WA.
- If A is a WA, $SID(CP)=ES_{CP}$[WA, ip(PP), ip(MA), ip(CP), Certificate, Code, t].

In SID(CP) description, ip(P) denotes the IP address of peer P; LP, RP and PP denote the left child's peer, the right child's peer, and the parent peer respectively; Token denotes the child agent is PWA or WA; SID_L and SID_R denote the signcryped SID for the left and right children respectively; ES_{CP}[M] denotes the message M is signcrypted by the public key of the current peer CP (P_{CP}) and the private key of original peer MA (S_{MA}). Code is the code of the agent for dispatch task or data access task. Finally, *t* is the timestamp, unique for all routes within a dispatch tree, at which the signcryption is generated whereas Certificate is the key-certificate of the original peer MA. The addresses of PP and CP are used for verification whereas the certificate is used for authentication.

4.1.2 Information Gathering procedure

Starting the binary dispatch process, the agent A_{MA} dispatches two PWAs to different peers, each being encapsulated with a SID for future dispatch tasks. When an agent A has successfully arrived at the current peer CP, the carried route SID(CP) can be decrypted and verified with the private key of CP and the public key of MA so that the agent can know:

- whether it is a PWA or WA;
- the signature signed at MA that can be used for integrity verification.

A PWA will also know:

- the address ip(RP) of the right child peer RP and whether it is a PWA or a WA;
- the signcrypted initiation data SID_R for the right child agent, which can only be decrypted and verified by the right child peer;
- the signcrypted initiation data SID_L for the left dispatch.

A WA will also know:

- the address of MA, ip(MA), the home where A_{MA} is residing. With this address, the WA can send the collected information to A_{MA}.

Clearly, under this model, at any layer, only the address of the right child agent is exposed to the current peer so that the right dispatch can be completed. For a PWA, if it has at most $m=2^k$ members, only k addresses of its members are exposed to the peer.

Suppose the dispatching of a PWA A at a peer CP is successful. Then, peer CP sends the following singcrypted message msg to PP:

$msg=ES_{PP}[Entity_{CP},ip(CP),t]$

where $Entity_{CP}$ is the full entity of the dispatched agent A at CP including its passport, code, and data, and t is the timestamp when CP receives the agent successfully. Once getting such a message, peer PP will keep message msg in its database as a successful dispatch record.

At this point, if Token equals PWA, a new dispatching at peer RP is performed, encapsulating SID_R and the process is repeated. Then, A should try to complete its virtual left dispatch; to this purpose, SID_L is decrypted and verified by using the private key of CP and the public key of MA.

On the other hand, if Token equals WA, A has to dispatch two WAs to peers ip(RP) and ip(LP), encapsulating SID_R and SID_L, respectively. When a WA is dispatched to peer CP, it will collect information and send back the result, denoted by Result(CP), obtained at time t_2, to agent A_{MA}, signcrypted by the private key of CP and public key of MA:

$$msg=ES_{MA}[ip(CP), Result(CP), t_2]$$

4.2 Protecting Information based on Serial Dispatch Model

Mitomi and Miyaji's multisigncryption scheme [17], an extension of signcryption scheme for multi-entities performing together the signcryption operation on a message, was proposed with the novel properties of message flexibility, order flexibility and order verifiability, with the aim to protect sensitive information from disclosing to other entity except the original peer. While the proposed approach guarantees a high degree of flexibility, it does not seem to be secure with respect to *exclude attacks*. In an exclude attack, one or more malicious peers try to exclude some of the other peers after these peers have partially signcrypted the information they provided. In the following, we modify the proposed multi-signcryption scheme in order not only to prevent peer exclude attacks but also to protect the secrecy of the collected information from being disclosed to other peers except the original one. Under the modified scheme, a mobile agent carries the request signcrypted by the original peer and then visits the other peers to collect information dynamically. Each new peer modifies the message (i.e. adds the information) and performs signcryption operation on the modified part to secure the provided information. The route of mobile agent is not specified beforehand.

4.2.1 Initialization of Participants

Let p, q be sufficiently large primes with $p = 2q+1$, and let $g \in Z_p^*$ have order q. Each signer peer $P_0, P_1, ..., P_n$ generates a pair of asymmetric key pairs (x_i, y_i) ,where $x_i \in Z_p^*$ and $y_i = g^{x_i} \bmod p$, and publishes the public key y_i along with its identity information Id_i through a certificate authority.

In the following, req (initially corresponding to information m_0) represents the original peer's request used to collect information from other peers, h, h_1, and h_2 are appropriate hash functions, and "||" denotes concatenation. Moreover, encryption/decryption functions are denoted by $(E(K_i, m_i) / D(K_i, C_i))$, where E denotes encryption operation, D denotes decryption operation, and C_i is a cipher text. h_2 is used to generate a symmetric key K_i to execute encryption/decryption function (E/D).

4.2.2 Dynamic Information Gathering procedure

In the following, we present a procedure for dynamically gathering secret information by mobile agents, based on the multi-signcryption scheme.

Preparation of the Mobile Agent

Original Peer (P_0) selects a random number $k_0 \in Z_q^*$ and computes: $R_0 = y_0^{k_0} \bmod p$, $r_0 = (h(req \,||\, Id))^{-1} \cdot R_0 \bmod q$ and $s_0 = (x_0 r_0 + y_1) \cdot k_0^{-1} \bmod q$. Then, the original peer gives the signature on req, i.e., (req, Id_0, s_0, r_0), to the mobile agent. After initialization, the mobile agent migrates to other peers to search for information autonomously.

Note that, differently from the original multi-signcryption Scheme [17], the public key of P_0 (i.e. y_0) is added at the first step when generating the signcrypted request. This means that only P_0 can unsigncrypt the message and get the collected information provided by each other peer.

Execution of the Mobile Agent

(1) When peer P_1 receives (req, Id_0, s_0, r_0) from P_0, it provides information m_1 on the request req, using the multi-signcryption scheme. To this purpose, it chooses a random number $k_1 \in Z_q^*$ and computes:

$R_1 = y_0^{k_1} \bmod p$,

$r_1 = (h_1(m_1 \,||\, Id_1)r_0)^{-1} \cdot R_1 \bmod q$,

$s_1 = (x_1 r_1 + y_2) \cdot k_1^{-1} \bmod q$.

At the same time, a secret key K_1 is generated as $K_1 = h_2(h_1(m_1 \| Id_1))$ and $(m_1 \| Id_1)$ is encrypted by using function E as $C_1 = E_{K_1}(m_1 \| Id_1)$. Then, P_1 sends the mobile agent with $(Id_0, Id_1 \| s_0, s_1 \| req_0, C_1 \| r_1)$ to the next peer.

(2) When peer P_i $(1 < i \le n)$ receives the partial signcrypted result $(Id_0, Id_1, ..., Id_{i-1} \| s_0, s_1, ..., s_{i-1} \| req, C_1, ..., C_{i-1} \| r_{i-1})$ from P_{i-1}, it reads the original peer's request description req and selects a modification m_i of the message. Then P_i performs signcryption by choosing a random number $k_i \in Z_q^*$ and computes:

$R_i = y_0^{k_i} \bmod p$, $r_i = (h_1(req \| Id_i) r_{i-1})^{-1} \cdot R_i \bmod q$ and $s_i = (x_i r_i + y_{i+1}) \cdot k_i^{-1} \bmod q$

At the same time, secret key K_i is generated as: $K_i = h_2(h_1(m_i \| Id_i))$ and $(m_i \| Id_i)$ is encrypted by using function E as $C_i = E_{K_i}(m_i \| Id_i)$. Then, P_i send $(Id_0, Id_1, ..., Id_i \| s_0\ s_1, ..., s_i \| req_c, C_1, ..., C_i \| r_i)$ to the next peer P_{i+1}.

(3) The last peer P_n uses the partial multi-signcryption information on req received from P_{n-1} to perform multi-signcryption operation by $(P_0, P_1, ..., P_n)$ as $(Id_0, s_0, req_0), ..., (Id_i, s_i, C_i), ..., (Id_n, s_n, C_n)$.

Note that, differently from the original proposal, to protect the agent from exclude attacks, the public key of the next peer to be visited is added in the generation of the signcrypted message for secure autonomous agents, by providing the verification of the signer's order, which means that the order of peers agent visits will be fixed after an agent returns back to the original peer [18].

Multi-unsigncryption on message *req*

(1) For $i = n, n-1, ..., 2$, the original peer computes:

$R_i' = y_0^{s_i^{-1} \cdot y_{i+1}} \cdot y_i^{x_0 r_i s_i^{-1}} \bmod p$, $T_i = R_i' \cdot r_i^{-1} \bmod q$ and $K_i = h_2(T_i)$, by using P_i's public key y_i. Then, it decrypts m_i and Id_i by $(m_i' \| Id_i') = D_{K_i}(C_i)$.

If $Id_i' = Id_i$ holds, then it accepts the signature and recovers r_{i-1} by computing: $r_{i-1} = T_i \cdot (h(m_i \| Id_i))^{-1} \bmod q$.

(2) For $i = 1$, it computes:

$R_1' = g^{s_1^{-1} \cdot y_2} \cdot y_0^{x_0 \cdot r_1 \cdot s_1^{-1}} \bmod p$, $T_1 = R_1' \cdot r_1^{-1} \bmod q$, $K_1 = h_2(T_1)$ verifies and decrypts m_1 and Id_1 by $(m_1' \| Id_1') = D_{K_1}(C_1)$.

If $h_1(m_1' \| Id_1') = T_1$ holds, then it accepts the signature.

At the end, the original peer decrypts and verifies the information and the signature provided by each new peer.

5 Analysis and Security Issues

In this section, the parallel dispatch model and the serial dispatch model are first compared, then security issues in both models are discussed in detail.

5.1 Comparison of the Parallel Dispatch Model and Serial Dispatch Model

In Section 4, we have proposed two models for mobile agents to collect information in P2P systems. While one model dispatches agents serially, the other exploits parallelism to dispatch agents simultaneously. The dispatch time complexity in the parallel dispatch model is $O(\log_2 n)$ whereas the complexity of the serial dispatch model is *O(n)*. Thus, the parallel dispatch model is more efficient than the serial one. Moreover, in the parallel model, since each work agent has different starting and ending time for the data accessing task, the returned results can hardly cause the master agent to become a bottleneck. The main drawback of this approach is however due to the fact that peers that agents will visit should be predefined and the route of the agents should be signcrypted at the first step. On the other hand, in the proposed serial model, a mobile agent is more autonomous than in the parallel model, it visits a pool of peers dynamically, and determines the next peer to be visited independently. However, the original peer can evaluate the collected information only after the mobile agent returns back the collected information at the end of the overall process.

5.2 Security Issues in the Binary Dispatch Model

(1) Preventing a PWA from Dispatching a Child Agent

During the period of dispatching a child agent, a malicious peer may peek into the code of the agent and cause it to skip the dispatch process at certain layer after the route is unsigncrypted. Note that skipping a peer

would mean skipping all other addresses that may be triggered by that peer. In the worst case, assuming peer P_1 is the malicious one, as shown in Figure 5, if the dispatch of A_5 from H_1 is not in fact performed, those agents in the group including A_5 to A_8 will not be activated. This means that the successful interception of the dispatch of a PWA will affect all the agents dispatched from that PWA.

This attack can be easily detected in this model. As an example, consider again Figure 5. If P_1 makes A_1 skip the process of dispatching agent A_5, agent A_{MA} cannot receive any messages from agents A_5, A_6, A_7 or A_8. A_{MA} will ask peers P_1 and P_5 to show whether the predefined dispatch has been performed. Apparently, if the dispatch has been carried out, P_1 will receive the confirmation message with the signature $ES_1[Entity_5, ip(P_5), t]$ from P_5. P_1 cannot forge this signature without P_5's private key. So, no matter what P_1 claims, the attack can be detected.

If the skipped dispatch is for a WA, for example A_7 does not dispatch A_8, it can also be detected since P_7 cannot show a correct signature from H_8 to show the dispatch is successful.

(2) Route Skip Attack

There is yet another case that can be handled in this model. Consider a partial dispatch route: PWA A_i at peer P_i dispatches A_j to P_j and P_j dispatches A_k to P_k, or there are more PWAs between A_i and A_k. In this model, a SID for a PWA includes the signcrypted route for its right child agent, which can only be unsigncrypted at the child's peer. Thus, the case described above that A_i directly dispatches A_k is not likely to take place without the involvement of A_j. That is why the SID is a nested structure. In the worst case, even if P_i can successfully predict that P_k is its descendent in the dispatch route and makes A_i dispatch a forged agent to P_k, the attack will not be successful either since the signature by MA encapsulated in SID clearly shows where the agent should come from and which peer should be its destination. Thus, the forged agent will be detected by the destination peer. Furthermore, the signature is also required to be included in the returned result for the verification by A_{MA}. Therefore, since forging the signature is impossible, this kind of attack cannot arise.

(3) Tampering a PWA to Dispatch an Agent to a Wrong Peer

If a malicious peer knows a peer where an agent will be dispatched from it, and the remote peer may probably offer a better service than itself, it may tamper the address so that the agent can be dispatched to another peer that is known not to be able to provide a competitive offer. The tamper can be done just after the signcrypted route is unsigncrypted. However, when an agent is dispatched to a wrong peer, its signcrypted route will not be correctly unsigncrypted there. Without the correct route, the verification process cannot be undertaken. Alternatively, even if the destination peer can get the correctly unsigncrypted route, the route will show that is a wrong destination since the address of the destination peer is included in the signature in the route generated by MA that cannot be tampered with. Thus, in both situations, the attack can be detected by the destination peer and the agent will be returned to the sender. Meanwhile, this error will be recorded by the destination peer for future investigation.

(4)Collusion Attack

Suppose that, in a normal sequence, peer P_i should dispatch an agent to P_j. If P_i and P_k are in a collusion tie, the agent could be dispatched to P_k. In this way, P_i and P_k make an attempt to skip the visit to P_j who is their competitor and send their own services instead. However, P_k can hardly forge the signature by P_j that should be included in the message returned to A_{MA}. In such a case, the counterfeited message can be detected when it is returned and this will cause an investigation against P_k and P_i. Since P_j will report that no such agent has ever been dispatched to it and P_i cannot show the correct dispatch record, which should include the signature by P_j, the attack can be identified.

5.3 Security Issues in the Serial Dispatch Model

(1) Protection of the Private Key of the Original Peer

In the proposed multi-signature and multi-signcryption scheme for secure mobile agents, the private key of the original peer is used in the first step of the signature generation on the request information. Moreover, agents do not carry the private key and any peer providing additional information only modifies the message part by performing the signcryption operation. Thus, to forge the signature and perform encryption of the request, the private key of the original peer must be obtained, but it is kept by the original peer and the scheme's secrecy is based on the discrete logarithm problem[15]. Thus, this type of attack cannot arise.

(2) Protection of Exclude Attack

Based on the assumption of non-conspiracy among peers, to protect agents from exclude attacks, similarly to what has been proposed by Kotzanikolaous, Burmester and Chrissikopoulos [18], the public key of the next peer to be visited has been added in the generation of the signcrypted message for secure autonomous agents, by providing the verification of the signer's order, which means that the order of peers agent visits will be fixed after agent returns back to the original peer [18].

(3) Prevention of the Disclosure of Collected Information

In the multi-signcryption scheme, the information provided by each peer is encrypted by a session key (i.e., $K_i = h_2(h_1(m_i \| Id_i))$). Any other party, no matter whether it gets the partial or whole multi-signcryption message, can verify the multi-signature and unsigncrypt

the information with P_i's public key y_i, unsigncrypts m_i and Id_i. Therefore, the information cannot be protected.

In our modified multi-signcryption scheme, the original peer's public key is included in the first step of signature generation. After the agent is dispatched to collect information, each peer generates the partial signature and encrypts the information. If any verifier, except the original peer and the peer that provides the information, wants to verify and unsigncrypt the information, it must get the private key of the original peer or of the peer that provides the information. This is almost impossible based on the assumption of non-conspiracy among peers.

6. Conclusions

In this paper, after motivating the usage of mobile agents in P2P systems for information retrieval tasks, we propose two models -parallel dispatch model and serial dispatch model - for mobile agents to collect information in P2P systems and we discussed how information can be protected under both models . To detect the attacks and protect agents from malicious peers, the route of agents in the parallel model should be predefined and signcrypted by the original peer before visiting new peers. On the other hand, in the proposed serial model, a mobile agent is more autonomous than in the parallel model, since it visits a pool of peers dynamically and determines the next peer to be visited independently. The drawback is that the original peer can evaluate the collected information only after the mobile agent returns back the collected information at the end of the overall processing. In future work, we plan to extend and combine the two proposed models in order to get maximum benefits in using mobile agents technology in the BestPeer system, making agents working on behalf of their owner more autonomously.

Acknowledgement

Kian-Lee Tan and Xiaolin Pang are partially supported by the NSTB/MOE research project (R-252-000-015-012/303).

References

[1] D.S.Milojicic, V.Kalogeraki, R.Luk ose, K.Nagaraja, J.P- ruyne, B.Richard, S.Rollins and Z.Xu , " Peer-to-Peer Computing", HP Labs Technical Reports, 2002.

[2] W.S.Ng, B.C.Ooi, K.L.Tan and A.Zhou, " PeerDB, A P2P-based System for Distributed Data Sharing", Proceedings of the 19th International Conference on Data Engineering, Bangalore, India, March 2003.

[3] P. Kalnis, W.S. Ng, B. C. Ooi, D. Papadias and K.L. Tan, "An Adaptive Peer-to-Peer Network for Distributed Caching of OLAP Results", Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD), Wisconsin, 2002.

[4] S. Sohn, and K. J. Yoo, "An Architecture of Electronic Market Applying Mobile Agent technology", Proceeding of 3rd IEEE Symposium on Computers and Communications (ISCC '98), Athens, Greece, pp. 359-364,1998.

[5] D.Lange and M. Oshima, "Mobile Agents with java: The Aglet API", appears in Mobility: Process, Computers, and Agents (edited by Milojicic, D., Douglis, F. and Wheeler, R.), Addison-Wesley Press, Reading, Massachusetts, USA, pp. 495-512,1999.

[6] D.Lange and M.Oshima "Mobile Agents with java: The Aglet API", appears in Mobility: Process, Computers, and Agents (edited by Milojicic, D., Douglis, F. and Wheeler, R.), Addison-Wesley Press, Reading, Massachusetts, USA, pp. 495-512, 1999.

[7] D.Lange and M. Oshima, " Programming and Deploying Java Mobile Agents with Aglets", Addison-Wesley Press, Massachusetts, USA, 1998.

[8] S.Papastavrou, G. Samaras and E. Pitoura, "Mobile agents for World Wide Web distributed database access", IEEE Transactions on Knowledge and Data Engineering, , Vol. 12, Issue 5 , pp. 802 –820,2000.

[9] S. Das, K. Shuster, and C. Wu, "ACQUIRE: Agent-based Complex QUery and Information Retrieval Engine", Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, 2002.

[10] D. Milojicic, " Mobile Agent Applications", IEEE Concurrency, Vol 7, Issue 3, pp. 80-90, 1999.

[11] Y. Zheng, "Signcryption and Its Applications in Efficient Public Key Solutions", Information Security Workshop (ISW '97), Springer-Verlag, LNCS 1397, pp.291—312, 1998.

[12] Y. Zheng, "Digital Signcryption or How to Achieve Cost (Signature & Encryption)<<Cost (Signature)+Cost (Encryption), in Advances in Cryptology-CRYPO'97, vol 1294, Springer-Verlag, , pp.165-179, 1997.

[13] A. Menezes, P. van Oorschot, and S. Vanstone, "Signature Schemes", Handbook of Applied Cryptography, CRC Press, 1996.

[14] P. Wayner, "Digital Copyright Protection", SP Professional, Boston, USA, 1997.

[15] A. Menezes, P. Oorschot and S. Vanstone, "Handbook of Applied Cryptography", CRC Press,1996.

[16] Y.Wang, K.L. Tan, X.L. Pang, "A Parallel Dispatch Model with Secure and Robust Routes for Mobile Agents", Proceedings of EC-WEB 2002 Conference, Springer-Verlag, LNCS Vol. 2455, pp 129-138, 2002.

[17] S.Mitomi and A.Miyaji, "A multisignature Scheme with Message Flexibility, Order Flexibility and Order Verifiability", Proceedings of 5th ACISP, LNCS #1841, Springer, pp. 298-312, 2000.

[18] P. Kotzanikolaous, M. Burmester and V.Chrissikopoulos, "Dynamic Multi-signatures for Secure Autonomous Agents", IEEE 2001, pp. 587-591, 2001.