A Parallel Dispatch Model with Secure and Robust Routes for Mobile Agents

Yan Wang, Kian-Lee Tan, and Xiaolin Pang

Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
{ywang, tankl, pangxiao}@comp.nus.edu.sg

Abstract. For mobile agents to be widely accepted in a distributed environment like the Internet, performance and security issues on their use have to be addressed. In this paper, we first present a parallel dispatch model with secure dispatch route structures. This model facilitates efficient dispatching of agents in a hierarchical manner, and ensures route security by exposing minimal route information to hosts. To further enhance route robustness, we also propose a mechanism with substitute routes that can bypass temporarily unreachable hosts, dispatch agents to substitute hosts before attempting the failed hosts again. Finally, a model for distributing the load of decrypting substitute routes is presented. We also present results of both analytical and empirical studies to evaluate different models.

1 Introduction

For mobile agent technologies to be accepted, performance and security issues on their use have to be addressed. First, deploying a large number of agents may cause significant overhead when dispatching them. Efficient dispatch methods are desirable. Second, when a mobile agent arrives at a host for execution, the code and data will be exposed to the host and the resources at the host may also be exposed to the mobile agent. Thus, security mechanisms should be set up to protect mobile agents from malicious hosts and vise versa. Particularly in EC environments, since a lot of e-shops selling the same product should be visited to respond to a customer's request, and they are competitive, it is important to protect the routes of a mobile agent if it should visit a list of hosts (e-shops) or if it should dispatch some mobile agents to other hosts. If a malicious host knows the route information, it may tamper with it so that its competitors that may offer better prices or services will not be visited.

In this paper, we first present a binary dispatch model that can hierarchically and efficiently dispatch n mobile agents in parallel with a complexity of $O(log_2 n)$. Based on it, we present a secure dispatch route structure where the agent at a dispatch layer only exposes the addresses of its child host to the current host. Thus, we preserve the efficiency of the binary dispatch model while ensuring route security. In addition, we propose a mechanism with encrypted substitute routes to facilitate robustness without sacrificing security and efficiency. It can bypass temporarily unreachable hosts by dispatching agents to substitute hosts, and try failed hosts again at the end of the whole

K. Bauknecht, A M. Tjoa, G. Quirchmayr (Eds.): EC-Web 2002, LNCS 2455, pp. 129–138, 2002.

dispatch process. Finally, a model for distributing the load of decrypting substitute routes is presented.

In this paper, we employ well-known public-key encryption algorithm and signature scheme [1, 2]. In the following, we assume that there exists a secure environment including the generation, certification and distribution of public keys and each host can know the authentic public keys of other hosts.

2 Basic Binary Dispatch Model and Its Secure Route Structure

In this paper, we assume a *master agent* A_0 running at home host H_0 is responsible for dispatching other agents. We call an agent a *Worker Agent* (WA) if its sole responsibility is to perform simple tasks, e.g., accessing local data on a host. If a WA also dispatches other agents besides its local data-accessing task, it is called a *Primary Worker Agent* (PWA) [3].

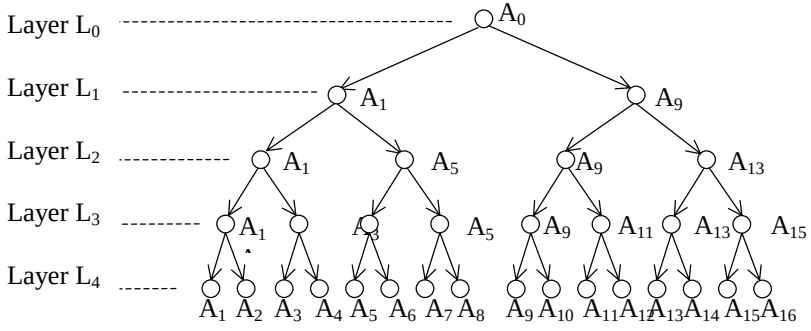


Fig. 1. Dispatch tree with 16 WAs

Here, we briefly introduce the basic binary dispatch model. As shown in the *dispatch tree* in Fig. 1, master agent A_0 has to dispatch 16 agents to 16 hosts (e.g. agent A_i to host H_i). Now, 16 mobile agents can be divided into 2 groups led by two PWAs, say A_1 and A_9. After A_1 is dispatched to H_1 it will dispatch A_5 and distribute 4 members to it. After that A_1 will transit to the same layer (i.e., L_2) as A_5, which is called a *virtual dispatch* costing no time. Now A_1 has 4 members only. Following the same process, A_1 dispatches A_3 and A_2 successively. Meanwhile, A_0 dispatches A_9 to H_9 to activate all agents in another branch in parallel. At last, A_1 becomes a WA and starts its local data-accessing task at H_1. As a whole, the model benefits from the parallel dispatches by different PWAs at different hosts. When there are $n=2^h$ mobile agents and T is the average time for dispatching a mobile agent, $(h+1)T$ will be the time for dispatching n mobile agents. So, the dispatch complexity will be $O(\log_2 n)$.

To ensure route security, we applied cryptographic technique to binary dispatch model. A basic definition of route structure is as follows:

(1) For a PWA at current host CH, $r(CH)=P_{CH}[isPWA, ip(RH), r_L(CH), r_R(CH), S_{H0}(isPWA, ip(PH), ip(CH), ip(RH), r_L(CH), r_R(CH), t)]$ (I)

(2) For a WA at current host CH, $r(CH)=P_{CH}[isWA, ip(H_0), S_{H0}(isWA, ip(PH), ip(CH), ip(H_0), t)]$

where

– r(CH) denotes the route at the current host, CH, where the agent should reside;
– isPWA or isWA is the token showing the current state of the agent;
– ip(H) denotes the IP address of host H; RH denotes the right child host of current host; PH denotes the parent host of current host;
– $r_L(CH)$ and $r_R(CH)$ denote the encrypted route for the left and right children respectively;
– $P_{CH}[M]$ denotes the message M is encrypted by the public key P_{CH} of the current host CH; $S_{H0}(D)$ denotes the signature signed on document D by host H_0 using its secret key S_{H0};
– and *t* is the timestamp at which the route is generated. *t* is unique for all routes within a dispatch tree.

Starting the binary dispatch process with secure routes, the agent A_0 dispatches two PWAs to different hosts, each being encapsulated with an encrypted route for future dispatch tasks. When an agent has successfully arrived current host CH, the carried route r(CH) can be decrypted with the secret key of CH so that the agent can know:

(1) it is a PWA or a WA. It is used to determine the next task of the agent;
(2) the signature signed at host H_0: S_{H0}(isPWA, ip(PH), ip(CH), ip(RH), $r_L(CH)$, $r_R(CH)$, t) for a PWA, or S_{H0}(isWA, ip(PH), ip(CH), ip(H0), t) for a WA.

If it is a PWA, it will also know

(1) the address ip(RH) of the right child host RH;
(2) the encrypted route $r_R(CH)$ for its right child agent, which can only be decrypted by the right child host
(3) the encrypted route $r_L(CH)$ for the left dispatch (virtual dispatch).

If it is a WA, it will know the address of H_0, ip(H_0), the home host where A_0 is residing. With this address, the WA can send its result to A_0.

Clearly, in this model, at any layer, only the address of the right child host is exposed to current host so that the right dispatch can be performed. For a PWA, if it has $m=2^k$ members altogether, only *k* addresses are exposed to the host.

For any route, since all information included in the route appears in the signature, any tamper attack will not success. Also the wrong dispatch attack and replay attack can be found by the destination host. Meanwhile, with nested structure, the dispatch skip attack will not success. More discussions on security threats can be found in [4].

3 Robustness Enhanced Extension

So far we have presented a security enhanced dispatch model for mobile agents. However, each PWA only knows the right child host RH where its right child agent is to be dispatched at a certain layer. As such, should the right host be unreachable, the right dispatch branch cannot be deployed and all the members grouped in this agent will thereby not be activated.

In [5] Li proposed a robust model for serial migration of agents and the route robustness is enhanced by dividing a route, say {ip(H_1), ip(H_2), …, ip(H_n)}, into two parts, say {ip(H_1), …, ip(H_i)} and {ip(H_{i+1}), …, ip(H_n)}. They are distributed to two agents A_1 and A_2 respectively. A_1 and A_2 are in partner relationship. Each agent residing at any host en route knows the addresses of the next destination and an alternative host. But the latter is encrypted by the public key of its partner agent. In case the

migration cannot be performed, the encrypted address will be sent to the partner agent for decrypting. With its assistance, the agent can continue its migration.

The problem of Li's model is that since both A_1 and A_2 are dynamically migrating, when one needs the other's assistance, locating each other will be costly for both time and system resources. Meanwhile, the model is serial so it is not efficient. But the idea using the mutual assistance of two agents to enhance the robustness is good and can be easily used in our model, where the two first PWAs in the left and right branches can do it better.

To provide one substitute route, the route structure (I) can be extended as follows:

(1) For a PWA at current host CH, $r(CH)=P_{CH}[isPWA, ip(RH), r_L(CH), r_R(CH), r_R'(CH), S_{H0}(isPWA, ip(PH), ip(CH), ip(RH), r_L(CH), r_R(CH), r_R'(CH), t)]$,
where $r_R'(CH)=P_{APWA}[ip(SH), r(SH), S_{H0}(ip(SH), r(SH), t)]$ is the substitute route for the right branch of host CH, SH is the substitute host. (II)

(2) For a WA at CH, $r(CH)=P_{CH}[isWA, ip(PH), ip(H_0), S_{H0}(isWA, ip(PH), ip(CH), ip(H_0), t)]$

$r_R'(CH)$ is encrypted by the public key of the first PWA in another branch of the whole dispatch tree, which here is termed as *Assistant PWA* (APWA). For example, in Fig. 1, A_1 is the first PWA in left branch so it is the APWA for the right branch following A_9. A_9 is the APWA for the left branch following A_1.

Now suppose A_1 is the first PWA in the left dispatch sub-tree. A_m is the right one. If the current host CH is the descendant of A_1, then $r_R'(CH)$ is encrypted by the public key of A_m, say P_{Am}. Otherwise, if CH is in the right dispatch sub-tree from the root node, $r_R'(CH)$ is encrypted by P_{A1}. If the dispatch failure occurs when A_{CH} is dispatching A_{RH} to right host RH, and A_{CH} is in the left sub-tree, A_{CH} should report it to A_m attaching the substitute route $r_R'(CH)$.

$$msg=P_{Hm}[ip(CH), ip(RH), r_R'(CH), S_{CH}(ip(CH), ip(RH), r_R'(CH), t_1)] \quad (1)$$

where t_1 is the time when msg (1) is generated.

When A_m gets such a message, it will

Step 1: Detect whether RH is unreachable. If it is true, then go to step 2, otherwise go to step 3

Step 2: A_m will decrypt $r_R'(CH)$, $r=S_{Hm}[r_R'(CH)]=[ip(SH), r(SH), S_{H0}(ip(SH), r(SH), t)]$, and send r to A_{CH} through a message

$$msg=P_{CH}[ip(SH), r(SH), S_{H0}(ip(SH), r(SH), t), S_{Hm}(ip(SH), r(SH), S_{H0}(ip(SH), r(SH), t), t_2)] \quad (2)$$

Stop.

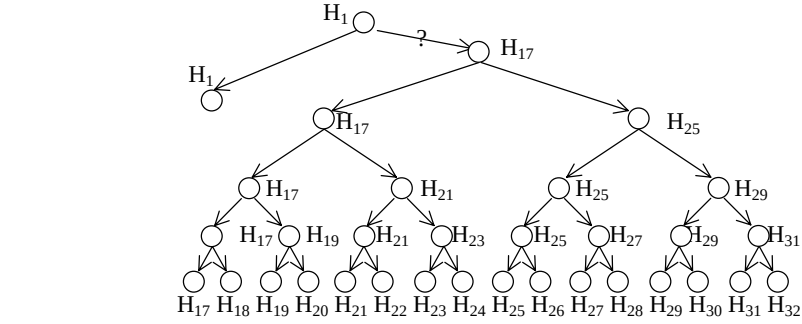
Step 3: If RH is in the correct state, A_m will tell A_{CH} about it and record the request in a database. Stop.

In msg (2), the second signature is generated by H_m and t_2 is the corresponding timestamp; SH is the substitute host.

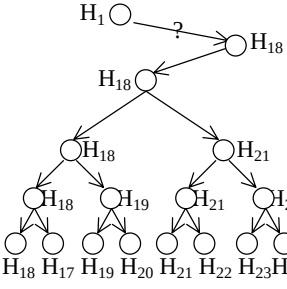
In this way by route structure (II), a PWA will have a substitute route for the dispatch of its right child agent. Once the original dispatch is not successful, with the assistance of its APWA, it can have another destination to dispatch.

What we should address is that the substitute host is originally included in the members for the right dispatch branch. Taking the dispatch tree in Fig. 2 as an example, if the dispatch failure occurs when A_1 at host H_1 is dispatching A_{17} to H_{17}, A_1 can

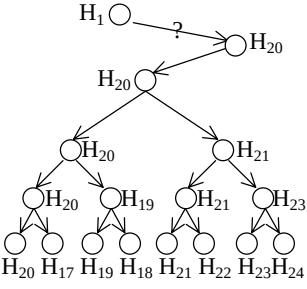
get a substitute route with the assistance PWA A_{33} at H_{33}. To generate the substitute route, choosing H_{18} to be the substitute host is better. By exchanging the positions of H_{17} and H_{18} as shown in Fig. 2(b), though H_{18} becomes the root of the branch with H_{17} to H_{32}, most sub-branches under H_{18} is kept unchanged. This is very important and can reduce the complexity to generate a new substitute route.



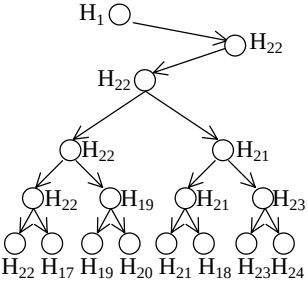
(a) original sequence



(b) if H_{17} is not reachable, H_{18} becomes a substitute



(c) H_{20} is the second substitute



(d) H_{22} is the third substitute

Fig. 2. Examples of substitute routes

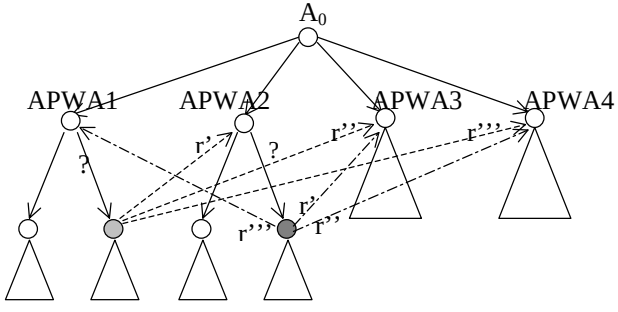


Fig. 3. A model with 4 branches and 3 substitute routes

Following the same idea, the second and the third substitute routes can be generated as shown in Fig. 2(c) and 2(d), where H_{20} can be the second substitute and H_{22} can be the 3rd one. An originally unreachable host should be put to be a leaf node so that the failure of the second dispatch attempt can be made without increasing more load of the APWA for route decryption.

As shown in Fig. 3, when there exist 3 substitute routes, 4 APWAs can be considered to partition the burden for decrypting substitute routes. In each branch following an APWA, the dispatch is performed in binary way. Each substitute route is encrypted by public keys of hosts where different APWAs reside. For instance, when a dispatch failure occurs in the first branch, the first substitute route is sent to APWA2 for decryption. The second substitute route will be sent to APWA3. Likewise the 3rd substitute route can only be decrypted by APWA4. Similarly, the first substitute route in the branch of APWA2 should be sent to APWA3 for decryption and so on. In this way, the burdens of decryption for APWAs are partitioned. The whole dispatch efficiency is not significantly decreased while the robustness is enhanced.

4 Complexity Analysis and Experimental Study

4.1 Complexity Analysis

In this section we compare our model with two existing secure models.

Westhoff's model in [6] adopted a fully serial migration providing secure route structure without any robustness mechanism. Suppose the visited hosts are H_1, H_2, …, H_n, the route is:

$$r(H_i)=P_{Hi}[ip(H_{i+1}), r(H_{i+1}), S_{H0}(ip(_{Hi}), ip(H_{i+1}), r(H_{i+1}), t)] \ (1 \le i < n)$$
$$r(H_n)=P_{Hn}[EoR, S_{H0}(ip(_{Hn-1}), ip(H_n), t] \quad \text{(i)}$$

where S_{H0} is the secret key of home host H_0 and EoR is the token meaning the end of the route.

Obviously the migration complexity is *O(n)* if there are *n* hosts to be visited.

Li's model [5] mentioned in Section 3 ensures both security and robustness. In Li's model, as the addresses of *n* hosts are distributed to two agents, say {ip(H_1), …, ip(H_m)} and {ip(H_{m+1}), …, ip(H_n)}, the nested route structure is:

$$r(H_i)=P_{Hi}[ip(H_{i+1}), r(H_{i+1}), r(H_i)', S_{H0}(ip(H_{i+1}), r(H_{i+1}), r(H_i)', t)] \quad \text{(ii)}$$

where $r(H_i)'=P_{AA}[ip(H_{i+2}), r(H_{i+2}), r(H_{i+2})', S_{H0}(ip(H_{i+1}), r(H_{i+2}), r(H_{i+2})', t)]$ is the substitute route where H_{i+2} is the new destination if H_{i+1} is not reachable. P_{AA} is the public key of the assistant agent.

The whole migration time can be theoretically half of the first model. However the time complexity is *O(n).*

Theorem 1: Disregarding the time spent on local data access, the time complexity of migration of Westhoff's model and Li's model for visiting *n* hosts is *O(n).*

In comparison, in our model the dispatch efficiency is greatly improved.

Theorem 2: If *n (n≥2)* WAs are dispatched by binary dispatch model, $h=log_2 n$ *(h≥1)* is an integer and the height of the dispatch tree, *t* is the time for dispatching a PWA or a WA, then the total dispatch time for *n* WAs is $T=(h+1)t$ and the time complexity is $O(log_2 n)$.

With regard to the complexity for generating routes, three models have different performances. Based on nested secure structure, which helps to prevent route tampering or deleting attacks and detects them as early as possible, assuming that the time to encrypt a route of arbitrary-length is a constant, the complexity for generating routes can be analyzed as follows.

Theorem 3: The time complexity for generating routes of Westhoff's model is *O(n).*

For Westhoff's model, the route with *n* addresses can be generated after the route with *n-1* addresses has been generated. So, the complexity is $T(n)=O(n)$ where $T(n)=T(n-1)+C$ *and* $T(1)=C$. *C* is a constant and the time of encrypting a route.

Theorem 4: The time complexity for generating a route with 1 substitute route of Li's model is $O(n)$.

In Li's model, suppose the hosts in predefined sequence are $\{H_1, \ldots, H_i, H_{i+1}, H_{i+2}, \ldots, H_n\}$, if host H_{i+1} is not reachable, H_{i+2} will become the next destination from H_i and H_{i+1} will never be visited for this journey. Consequently, from route structure (ii), when generating $r(H_i)$, both $r(H_{i+1})$ and $r(H_i)'$ should be generated first. $r(H_i)'=P_{AA}[ip(H_{i+2}), r(H_{i+2}), r(H_{i+2})', S_{H0}(ip(H_{i+1}), r(H_{i+2}), r(H_{i+2})', t)]$, it is a substitute route with the addresses of $H_{i+2}, H_{i+3}, \ldots, H_n$ in sequence. Note $r(H_{i+1})=P_{Hi+1}[ip(H_{i+2}), r(H_{i+2}), r(H_{i+2})', S_{H0}(ip(H_{i+1}), r(H_{i+2}), r(H_{i+2})', t)]$. The difference of two routes is that they are encrypted by different public keys. Therefore when generating $r(H_i)'$, $r(H_{i+2})$ and $r(H_{i+2})'$ exist already and the cost for generating $r(H_i)'$ is constant *C* only. Hereby the route generation complexity is $T(n)=T(n-1)+2C$ *and* $T(1)=C$. And $T(n)$ is $O(n)$. Likewise, the time complexity for generating 3 substitute routes is the same where $T(n)=T(n-1)+4C$.

However, if a failed host is used for a second attempt in Li's model, the complexity for generating routes will become extremely bad since the sequence of hosts in a substitute route has been changed and the route should be generated and encrypted again.

If host H_{i+1} is not reachable from H_i, when H_{i+1} is put as the last destination for the second attempt, the sequence of hosts in the substitute route will be $\{H_{i+2}, H_{i+3}, \ldots, H_n, H_{i+1}\}$. In such a case, when a migration route includes 1 substitute route, the time complexity will be $T(n)=2T(n-1)+C$ and $T(n)$ is $O(2^n)$. Likewise, when there are 3 substitute routes, the time complexity will be $T(n)=4T(n-1)+C$ and $T(n)$ is $O(4^n)$.

Theorem 5: The time complexity for generating routes with 1 or 3 substitute routes of Li's model making the 2nd attempt to the failed hosts are $O(2^n)$ *and* $O(4^n)$ respectively.

Theorem 6: In the secure binary dispatch model, the complexity for generating routes without substitute route is $O(n)$.

For our model, the complexity for generating routes without substitute route is $O(n)$, where

$$\begin{cases} T(n)=2T(n/2)\ (n=2^k)\ // & \text{2 routes are generated for left branch and right branch, each has } n/2 \text{ addresses} \\ T(i)=2T(i/2)+C\ (i=2^h,\ 2^{k-1}\le i \le 2)\ // & \text{if r(CH) has } i \text{ addresses, each of its } r_L \text{ and } r_R \text{ has } i/2 \text{ addresses} \\ T(1)=C & \end{cases}$$

Theorem 7: In the robust binary dispatch model, the complexity for generating routes with 1 or 3 substitute route is $O(n\log_2 n)$.

When generating the first substitute route for a branch, only a few steps should be taken in the left sub-branch of this branch. Considering the case in Fig. 2(b), when H_{17} and H_{18} are exchanged, the branches with the root of H_{19}, H_{21} and H_{25} are all not changed. The number of the steps is the height *h* of the sub-branch. And hereby $T(n)$ is $O(n\log_2 n)$, *where* $T(n)=2T(n/2)+C$, $T(i) \le 2T(i/2)+(h+1)C\ (n=2^k,\ i=2^{h+1},\ 2^{k-1}\le i \le 2)$ *and* $T(1)=C$.

Similarly, the step numbers for generating the second substitute route and the third one are all *(2h-1)*. The time complexity for generating a route with 3 substitute routes and 4 branches is $O(n\log_2 n)$, where $T(n)=4T(n/4)+C$, and $T(i) \leq 2T(i/2)+(5h-1)C$ ($n=2^k$, $i=2^{h+1}$, $2 \leq i \leq 2^{k-2}$) and $T(1)=C$.

4.2 Experiments

In Section 4.1, for simplicity, the analysis is based on the assumption that the encryption time of a message of any length is a constant. To further study the performance of the different models, we conducted 3 experiments on a cluster of PCs connected to a LAN with 100Mbytes/s network cards PCs running Window NT, JDK [7], IBM Aglets 1.0.3 [8]. For route generations, experiments are based on a PC of Pentium III 700 MHz CPU and 128 Mbytes RAM. For serial migration and binary dispatch, the experiment is put on a cluster of PCs. Each PC has a Pentium 200MMX CPU and 64 Mbytes RAM. All programs run on the top of the Tahiti servers from the ASDK [8, 9] and JDK from Sun Microsystems [7].

Note that all encrypted routes adopt nested structure. To encrypt a route, we use the RSA algorithm [2] with the key length of 1024 bit. Hash function MD5 is used to generate a hash value with fixed-length of 128 bytes.

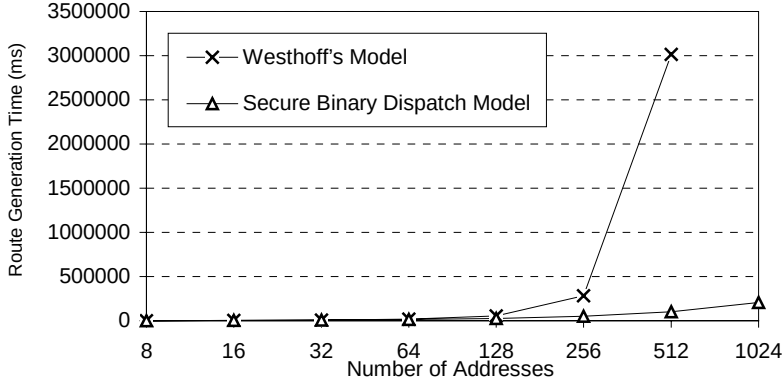


Fig. 4 Route generation time for Westhoff's model and binary dispatch model

In experiment 1, we first compare the route generation time of Westhoff's model and our secure binary dispatch model. All results are shown in Fig. 4. When the number of addresses is fewer than 128, the 2 models deliver similar performances. When the number becomes 256 or more, the binary dispatch model begins to outperform the serial model.

For Westhoff's model, each time after encryption, the route's length is increased at least with a length of an IP address and a signature. For example, when there are 512 addresses, the Westhoff's model performs 512 encryptions. As we measure, it uses 284 seconds to complete the first 256 encryptions and 2731 more seconds for the last 256 encryptions. The total time is 3015 seconds. For the binary dispatch model, it completes all encryptions in 101 seconds, and takes 37 seconds for 512 leaf nodes. But

when generating the route with 1024 addresses, the program of the Westhoff's model ran out of memory after the 771th address is added where the heap size is set up to 1200 Mbytes and it has reached the maximum.

In experiment 2, we compare the generation time for routes with one substitute route. For Li's model, we implemented the case of skipping a failed host. The results shown in Fig. 5 illustrates that though time complexities of the two models analyzed in Section 4.1 are different (i.e. $O(n)$ vs. $O(nlog_2n)$), their performances are very close to each other when the number of addresses is not greater than 256. But when the there are 512 addresses or more, the binary dispatch model begins to outperform.

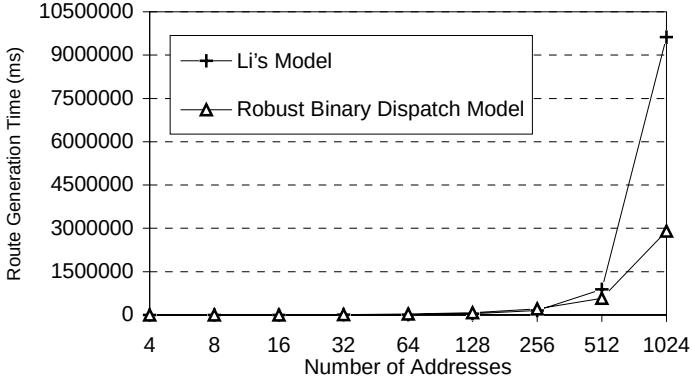


Fig. 5 Comparison of the time for generating a route with 1 substitute route

In experiment 3, we tested up to 64 hosts to compare the migration/dispatch time of different models ignoring any robustness mechanism. In the implementation, a mobile agent will not access any local data so that the measured time is used for migration or dispatch only. The results are shown in Fig. 6.

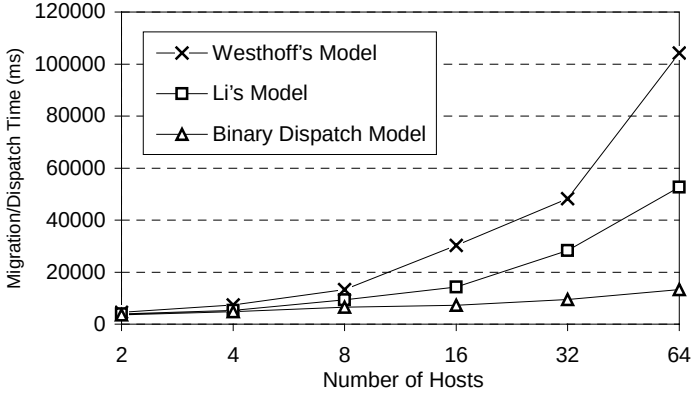


Fig. 6 Comparison of migration/dispatch time

When the number of visited hosts is no more than 8, the performance differences are not significant. With the increase of the number of hosts, the migration time of any

serial migration model increases very fast. In comparison, the dispatch time for binary dispatch model increases fairly slowly. When having 64 hosts, the binary dispatch model can get 74.9% and 87.3% savings respectively in comparison to Li's model and Westhoff's model.

5 Conclusions

In this paper we have proposed a binary dispatch model of mobile agents with secure routes and robustness mechanisms. It utilizes the automation and autonomy of mobile agents and the corresponding code is simple. Besides the high efficiency from binary dispatch, the secure mechanism provides the capability to protect mobile agents from malicious hosts. Meanwhile, the robustness mechanism enables the fault-tolerance without any loss on security. Additionally, for practical applications, mobile agents having tasks of the same type and having physically close destinations can be put in the same group encapsulated with pre-encrypted route structures.

Acknowledgement. This work is supported by the NSTB/MOE funded project on Strategic Program on Computer Security (R-252-000-015-112/303).

References

1. Wayner P., Digital Copyright Protection, SP Professional, Boston, USA (1997)
2. Rivest R.L., Shamir A., Adleman L., A Method for Obtaining Digital Signatures and Public-key Cryptosystems, Communications of the ACM (1978)
3. Wang Y., Dispatching Multiple Mobile Agents in Parallel for Visiting E-Shops, Proc. of 3rd International Conference on Mobile Data Management (MDM2002), IEEE Computer Society Press, Singapore 2002 (61-68)
4. Wang Y. and Tan K. L., A Secure Model for the Parallel Dispatch of Mobile Agents, Proc. of Third International Conference on Information and Communications Security (ICICS2001), LNCS Vol. 2229, Springer-Verlag, Xi'an, China, 2001 (386-397)
5. Li T., Seng C.K. and Lam K.Y., A Secure Route Structure for Information Gathering, 2000 Pacific Rim International Conference on AI, 2000
6. Westhoff D., Schneider M., Unger C. and Kenderali F., Methods for Protecting a Mobile Agent's Route, Proceedings of the Second International Information Security Workshop (ISW'99), LNCS 1729, Springer Verlag, 1999 (57-71)
7. URL: http://java.sun.com/products/
8. Lange D., and Oshima M. Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley Press, Massachusetts, USA (1998)
9. URL: http://www.trl.ibm.co.jp/aglets/