Trust Based Routing for Misbehavior Detection in Ad Hoc Networks

Wei Gong\textsuperscript{1,2}, Zhiyang You\textsuperscript{1,2}, Danning Chen\textsuperscript{2}, Xibin Zhao\textsuperscript{2}, Ming Gu\textsuperscript{2}, Kwok-Yan Lam\textsuperscript{2}
\textsuperscript{1}Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
\textsuperscript{2}Key Laboratory of Information System Security of Ministry of Education, School of Software, Tsinghua University, Beijing 100084, China
Email: ph.d@live.cn

Abstract—Node misbehavior due to selfish or malicious intention could significantly degrade the performance of MANET because most existing routing protocols in MANET are aiming at finding most efficiency path. To deal with misbehavior in MANET, an incentive mechanism should be integrated into routing decision-making. In this paper firstly we review existing techniques for secure routing, and then propose to use trust vector model based routing protocols. Each node would evaluate its own trust vector parameters about neighbors through monitoring neighbors’ pattern of traffic in network. At the same time, trust dynamics is included in term of robustness. Then we integrated trust model into Dynamic Source Routing (DSR) and Ad-hoc On Demand Distance Vector (AODV) which are most typical routing protocols in MANET. We evaluate the performance of those trust routing protocols by comparing the simulation results of with and without the proposed trust mechanism. The simulation results demonstrate that modified routing protocols can effectively detect malicious nodes and mitigate their attacks.

Index Terms—MANET; security; trust model

I. INTRODUCTION

The proliferation of mobile devices such as smart phone and laptop, has led to the growth of wireless Mobile Ad hoc Networks (MANET). MANET is formed with wireless mobile nodes without pre-established infrastructure. Each node in MANET is responsible for relaying packets to other nodes. Some packets can be delivered from a source node to a destination node by way of various intermediate nodes, thereby maintaining network connectivity [1] and applicability of MANET depends heavily on cooperation between nodes in such a dynamic environment. Due to openness of MANET, nodes moving in any direction can join or leave the network at any time, and also the transfer medium that is electromagnetic spectrum can be publicly accessed without restriction. In such a context selfish/malicious nodes are more likely to appear. Selfish nodes are characterized by their reluctance to spending resources to cooperate on its behalf. Malicious nodes always attack the network’s availability through common techniques such as flooding, black hole and denial of service (DoS) [2]. Because of the difficulties in MANET such as dynamic network topology and constraint battery resources, security solutions that have been deployed for wired networks are not directly portable to ad hoc networks.

Many secure routing protocols were developed to protect routing protocols from malicious behaviors. The Authenticated Routing for Ad-hoc Networks (ARAN) [3] secure routing protocol is an on-demand routing protocol which relies on the use of digital certificates to identifies and defends against malicious actions in the ad-hoc network. In Ref. [4], Zapata proposed the Secure Ad-hoc On-Demand Distance Vector which is an extension of the AODV routing protocol. Through providing security features like integrity, authentication and non-repudiation, it effectively protects the route discovery mechanism. And this scheme is based on the assumption that each node should have certified public keys of all nodes in ad hoc network. Security-Aware Ad-hoc Routing (SAR) [5] provides a generalized framework for any on-demand secure ad-hoc routing protocol. SAR uses security information to dynamically control the routing selection process according to routing tables. It requires that nodes at the same trust level must share a secret key.

From above discussion, we can see that most current solutions for defending against malicious attacks are based on public key infrastructure (PKI). The prominent problem of such approaches is that all the secure protocols were relied on a central or distributed trusted third party for enforcing cryptographic mechanisms. The requirement of trust authorities is against the basic characteristic of ad hoc networks which are supposed to be improvised and temporal. Furthermore, those solutions also cannot address the problem that the legitimate certified node can still perform routing attacks against other normal nodes without being detected. In addition, the computation and network overhead of PKI mechanism are not very fit for MANET in terms of the network’s overall throughput, availability and robustness. Therefore we need to find an effective mechanism to discover the misbehavior nodes whether they already have trust identity or not and consequently prevent ad hoc network performance from severely deteriorating due to malicious nodes’ attacks.
In the following, we discuss typical malicious behaviors and their effect in MANET. Then we outline some relevant previous work in Section 3. In section 4 we describe our trust model in MANET in detail and integrated trust model into commonly MANET protocols in section 5. The simulations and their analysis are presented in section 6. Finally, section 7 offers concluding remarks.

II. AFFECTS OF MALICIOUS NODES IN MANET

In MANET, a variety of attacks targeting the network layer have been identified and elaborately analyzed by researchers. By disturbing routing protocols, malicious nodes can inject themselves into the path between the source and destination, fabricate packet from source, and absorb network traffic. One of the most typical misbehaviors in MANET is black hole attack [2] which often taken actions in two phases. First, in routing discovery phase, with the intention of dropping packets the malicious node announcement that it has a valid route to the destination node, but this route is fabricated. And then, in data forwarding phase, the attacker absorbs the data packets which are supposed to be forwarding to next-hop node. Therefore, communications of nodes under black hole attack are totally suppressed. An even more subtle form of black hole attacks is that attackers can selectively forward packets; thereby their malicious behaviors are not so easy to be detected. Another type of attack is grey hole which is an extension to the black hole attack. In grey hole attack the attacker could build a grey hole, in which it could drop some specific packets. For instance, it can forward routing packets; but absorb data packets. A more subtle type of similar attack is the wormhole attack [6] in which it creates a tunnel between two colluding malicious nodes which may be linked through a faster wired network. Moreover, rushing attack [7] is a typical fabrication attack. This attack always aims at on demand routing protocols which hold back duplicate packets at each node. The legitimate routing messages could be suppressed by malicious routing messages sent by attackers because nodes may discard legitimate routing messages as duplicate copies.

We did black hole attacks simulations in NS2 simulator [8], using widely used Dynamic Source Routing (DSR) [9]. Fig. 1 shows the results of the network packet delivery ratio (This ratio is data packets successfully arrived at destinations to all data packets delivered from sources) under different number of malicious nodes and different node mobility settings. Compared to the situation without malicious nodes in the figure, we can infer that packet delivery ratio is dramatically reduced as the result of presence of a small portion of malicious nodes in the network. Furthermore, simulation results show that the ad hoc network performance is severely deteriorating along with the move speed of nodes increasing. It can be explained by the notion that the faster malicious node moves, the bigger region it covers.

The above experiments show terrible effects of a relatively small number of malicious nodes on the performance of ad hoc network. Therefore, we should introduce some countermeasures to suppress these misbehaviors.

III. RELATED WORK

In order to protect MANET from malicious behaviors such as black hole attacks, several secure routing protocols have been proposed in recent years [10-14]. Most of those protocols which are trying to secure route discovery process are based on PKI such as digital signature and shared keys. In [13], the author proposed that the source node always checks the routing reply which is based on digital signature to make sure it is from the destination so that it can secure the route. ARIADNE [14] provides assurance that both the source and destination nodes authenticate the messages, and moreover, the intermediate nodes have to insert their own digital signature in route request.

However, PKI-based secure routing protocols have some constrains. Key generation and distribution which are essentials of PKI security are difficult to be implemented within MANET which lacked trust authorities. It is obvious that the securities of those protocols are based on the same assumption: Only if they have chance to in advance configure central or distributed trust authorities, the assurance of security is just in hand. However, the pre-configuration requirement contradicts the nature of MANET which ought to be improvised spontaneously and provisionally. Therefore, the notion of relative security protection may be more suitable for MANET. Here we propose an efficacies mechanism which is based on trust evaluation other than PKI-based approach. In our mechanism each node should collect information from normal actions such as packet sending and receiving to compute its own trust vector about neighbor nodes. This trust vector can be normalized into a single trust value that has been provided as evidence for decision making in routing selection process. In this way we could address these problems or to reduce bad affects of malicious nodes to some extent. The next section proposed the trust model. Unlike most PKI-based schemes discussed before, our model mainly focuses on post route discovery phase: when packets are being transmitted on discovered routes. Of course, if we want to
protect our confidential information that is transmitted over discovered route which ought to be trusted, the use of cryptography is absolutely useful.

IV. TRUST MODEL

Our trust model is an adaptation of Vector Model of Trust [15] and we configure it for use in MANET. The definitions of our trust vector model in MANET are as follows.

A. Trust vector

Trust vector of node A to node B is:

\[ V(A \rightarrow B) = [\lambda E_A, \lambda K_A, \lambda R_A]. \]  

(1)

where \( \lambda E_A, \lambda K_A \) and \( \lambda R_A \) are node A’s evaluation of experience, knowledge and recommendation to node B, respectively.

In order to normalize trust vector, we introduced the trust policy vector which has the same dimension as the trust vector as in [15]. The elements of policy vector are real numbers in the range \([0,1]\) and the sum of all elements is equal to 1.

The normalization of trust vector can be defined as

\[ |V(A \rightarrow B)| = |W_e| \odot |V(A \rightarrow B)| \]

\[ = |W_e, W_k, W_r| \odot [\lambda E_A, \lambda K_A, \lambda R_A] \]

\[ = W_e \odot \lambda E_A + W_k \odot \lambda K_A + W_r \odot \lambda R_A \]

\[ = |W_e, W_k, W_r| \odot \lambda T_B \]

(\( \lambda T_B, \lambda E_A, \lambda K_A, \lambda R_A \) \( \in [0,1], |W_e, W_k, W_r| \in [0,1] \)).

(2)

where \( W_j \) is node A’s trust policy vector composed of three weight elements according to three dimension of trust vector respectively. \( \lambda T_B \) is a single trust value of node A on node B corresponding to the normalized trust vector. There is a big advantage of using a single value. First, a single value is more intuitive and easier to be integrated in routing selection than a vector. Moreover, in evaluating recommendation, this value could be included into recommendation value computation.

B. Evaluating experience

\( \lambda E_A \) is node A’s evaluation to node B by directly monitoring packets communication of node B. This evaluation measures the ability of forwarding packets on node B. If node B is node A’s neighbor, \( \lambda E_A \) can be computed by node A as follows:

\[ \lambda E_A = \frac{P_B}{P_0} = \frac{P_B^{out} - P_B^{off}}{P_B^{out} - P_B^{on}}. \]

(3)

where \( P_0 \) is the number of packets node B had actually forwarded. It should be all out-coming packets from node B denoted as \( P_B^{out} \) exclude those packets which are from source node B to destination node A denoted as \( P_B^{off} \); \( P_B \) is the number of all packets node B responsible for forwarding. It can be computed as all in-coming packets denoted as \( P_B^{on} \) except packet those from source node A to destination B denoted as \( P_B^{on} \).

C. Evaluating knowledge

\( \lambda K_A \) is node A’s evaluation to node B by directly observing MAC layer link quality between node A and node B on physical layer. This parameter is the probability that the data packet will be successfully transmitted between two network nodes [17]. Computation formula is as follows:

\[ \lambda K_A = (1 - p_{B,A}) \times (1 - p_{B,A}). \]

(4)

\( p_{B,A} \) is packet loss probability from node A to node B, while \( p_{B,A} \) is packet loss probability from node B to node A. For example, we let each node broadcast a probe packet every second. Suppose that node A has received 7 probe packets from B in the previous 10 seconds, at the same time B found that it had received 9 probe packets from A in the previous 10 seconds. Thus, the loss rate of packets from A to B is 0.3, while the loss rate of packets from B to A is 0.1. We all know that a successful data transfer in IEEE 802.11 MAC layer involves sending the data packet and receiving a link-layer acknowledgment from the receiver. Thus, the probability that the data packet will be successfully transmitted from A to B in a single attempt is \((1-0.3) \times (1-0.1) = 0.63\).

D. Evaluating recommendation

\( \lambda R_A \) is node A’s evaluation to node B by collecting recommendations about node B from other nodes which should be the neighbor of node B. This is given by the equation

\[ \lambda R_A = \sum_{C_{cap}} \left[ |V(A \rightarrow C)| \times |V(C \rightarrow B)| \right] / \sum_{C_{cap}} |V(A \rightarrow C)|. \]

(5)

where \( C_{cap} \) is a group of recommenders.

E. Trust Dynamics

Suppose that, initially node A have the trust value on node B is at time t1; but after a certain period, node B may travel to another zone which is out of radio range of node A due to nodes mobility in MANET. At time t2, node B happens to back in node A’s radio range again. The trust value should decay during this time gap. Let \( \lambda T_B(t) \) be the trust value of node A to node B at time t1 and \( \lambda T_B(t) \) be the decayed value of the same at time t2. Then the time-dependent trust value is defined as follows

\[ \lambda T_B(t) = \lambda T_B(t_0) \times e^{-\Delta t/k}. \]

(6)

where \( \Delta t = t_2 - t_1 \) and \( k \) is an integer greater than or equal to 1.

V. TRUST ROUTING

In this part, we present the utility of our trust model by demonstrating how to integrate this trust scheme into routing protocols in MANET. Here we select DSR and AODV as representative routing protocols. Though these
protocols are currently under development, they have been constantly improved.

We will briefly introduce these two protocols first. And then we discuss the how to compute each fraction of trust vector for each of these protocols. The last we present the whole trust routing process by examine details of each routing phase.

A. DSR and AODV

The Dynamic Source Routing (DSR) protocol [9] is a typical on-demand routing protocol. The most interesting feature is that all data packets sent in DSR have no need to depend on intermediate nodes to decide which one is next hop; this is because complete routing information is included in the data packets from source node. In routing discovery phase, a node requires a route to a destination, it first broadcasts a ROUTE REQUEST message. Then if the recipient node which has no knowledge about the required destination, it would append its own address to this request packet and rebroadcast this new ROUTE REQUEST message; if this recipient node is the destination or an intermediate node which has a route to the destination in its route caches, it would send a ROUTE REPLY message containing the complete route information from the source to the destination. At last source node should receive these route request feedbacks, but as there may be many ROUTE REPLY messages arrived at source node, so it need to make a decision based on some criteria such as number of hops or latency. In routing maintenance phase, all nodes should add usable routing information from which they have been forwarding or overhearing any packets into their own route caches. A ROUTE ERROR message should be sent to each node which had used this particular route if nodes found some route had been broken.

The Ad-hoc On Demand Distance Vector (AODV) [4], as its name, is a distance vector routing protocol which is designed specifically for mobile ad-hoc networks. It is also a typical routing of on demand protocols in MANET which find the routes only when asked. To some extent, the route discovery and maintenance phases of the AODV are adapted from the DSR and Destination-Sequenced Distance-Vector Routing (DSDV) [16]. One distinctive feature of AODV is that it makes extensive use of sequence numbers in route control packets in order to avoid the problem of routing loops. In routing discovery phase, a source node also broadcasts a ROUTE REQUEST message to find the route which is not know. A ROUTE REQUEST message always includes an identifier, source IP address, destination IP address, hop counter, sequence number and other control flags. The identifier field is used to uniquely symbol the ROUTE REQUEST message; the hop counter records the number of intermediary nodes between the sources to the destination; the sequence number is used to decide which packet is newer than the others. If the intermediary node which received the ROUTE REQUEST message has no idea about the source IP address and message identifier or doesn’t have a newer route to the destination which usually indicated by a larger sequence number, should first add one hop to hop counter and then rebroadcasts this updated route request packet; in a certain interval of time, this intermediary node should keep a reverse route to the source. If the destination node receive the ROUTE REQUEST message or intermediary node has a newer route to the destination, it should send a ROUTE REPLY message to the source node. A ROUTE REPLY message should include the sequence number of destination, source IP address, destination IP address, hop counter and other control flags. All intermediary nodes which receive the ROUTE REPLY message should increments the hop counter and transmit this message on the reverse route lines. In routing maintenance phase, the AODV use periodic HELLO messages to detect status of physical link. If a detection of link break for some active route is made, the detector would send a ROUTE ERROR message to its neighbors so that nodes which had used that particular route could update their own route information.

B. Trust computation in routing

In evaluation of experience of trust vector, we should measure how many out-com ing packets the immediate neighboring node had been sincerely sent. To realize this, we should monitor their participation in the packet forwarding. So we place all nodes in the promiscuous mode all the time whether a node transmits control packets or data packets. When it overhears its immediate neighbor nodes forwarding the packet, it should first checks the integrity of the packet in order to make sure the packet had not been modified by other malicious nodes. Then we out-coming packet counter of this neighbor node should be incremented if it pass integrity test. However if the integrity test fails or the neighbor node refuse to cooperate to forward packets it supposed to, its corresponding forwarding counter would not change. After a period of time, its experience value would be extremely low as a result of malevolent behavior.

In evaluation of knowledge of trust vector, we use link-layer acknowledgements because underlying MAC protocol provides feedback of the successful delivery of the transmitted data packets. Therefore, we can conveniently compute the MAC layer quality.

We all know that the nodes in MANET generally have a transmission range about 100 or 200 meters. This limitation also causes that some nodes cannot directly acquired experience and knowledge of trust vector, but only from recommendation of others. In evaluation of recommendation of trust vector, we should consider how trust value can be easily propagated from one node to the others without much burdensome overhead on network commutation. Here we proposed a proper scheme based on route discovery process. We choose to spread trust value of nodes along with the ROUTE REQUEST message as shown in Fig.2. The node should add its trust value about preceding node from which it received the ROUTE REQUEST message before sending a ROUTE REQUEST message to next hop. By using this scheme, the trust value could be properly spread along with ROUTE REQUEST message. For instance, firstly a ROUTE REQUEST message from node A to node B and node C, then before node B should propagate request
Figure 2. The spread of trust value along with ROUTE REQUEST message to node D and node E, it should append its trust value of node A to this request message. Afterward, node E and node E should go on flood route request, at this time, the node D’s and node E’s trust value of node B are easily spread along with route request. So does node C.

C. Trust integration in routing

In this part, we will discuss how trust model integrated into whole routing process. In DSR and AODV protocols, before starting a new route discovery, firstly it should examine the routing table or cache to check whether there already has a workable route to the destination. If unfortunately no proper route information is available, a route discovery phase is triggered by flooding the ROUTE REQUEST message. Afterward, when we should select which node can be the next hop, a number of criteria can be judged such as number of hops or latency because usually one node that is faster to get destination is preferred. Here in trust routing protocols, old rule should be changed, we assign trust value as cost of nodes. Therefore, when each time a node has some packet to be sent or forwarded, it should examine the routing table or routing cache for all possible paths which can arrive at same destination. Then it makes a comparison of trust value of all next hops in those candidate paths and the most proper path which should be highest trust value is selected. If the next hop is not known by the sender or forwarder before, the least number of hops path to the destination is selected. All nodes should try to minimize cost by selecting neighbor nodes, which must have an trust value greater than or equal to the predefined trust value threshold. If there is no next hop available in candidate paths according to trust level which should be greater than the trust threshold, then a local link repair process is started. So if there have been any data packets which should be forwarded with an improper trustworthy next hop, we would stop forwarding action and buffered it for a certain interval of time in which another route discovery would be started to find an alternate route that should be trustworthy. In case an alternate route is found, the packet would be sent onto that route. Otherwise, a ROUTE ERROR message would be sent to the source node informing it of the link error.

The trust value threshold can be set in numerous ways according to different application environment. In general, a higher trust value threshold means a rigid forwarding policy which must be obeyed by all nodes. In applications where precise throughput and correct forwarding is required, this high threshold is more preferred. Nevertheless, it is possible that a heavy traffic node may be mistakenly diagnosed as malicious if its trust value below the threshold. On the other hand, a lower threshold suffices the requirement to detect nodes which demonstrate sustained malicious behavior. Moreover, packets dropping may be caused by high mobility and fast change in topology whether the nodes is malicious or benevolent. In addition, packet drop can also happen as a result of MAC layer collisions due to congestion in communication. So many packet droppings are influenced by the very nature of MANET such as mobility pattern or traffic load. This claim can be confirmed by the fact that packet delivery ratio always below 100% even when there is no malicious nodes in MANET. Fortunately, even under high mobility situation, this ratio of packet droppings which due to network characteristic can be neglect. So we can still successfully detect those malicious nodes according to a suitable trust value threshold under diverse conditions.

VI. SIMULATION AND ANALYSIS

A. Simulation set-up

In this section, we show results of simulation of two groups, one is DSR with trust vector model (TVDSR) and pure DSR, the other is AODV with trust vector model (TVAODV) and pure AODV. The simulations are done in NS2 simulator (version 2.33). In TABLE I, we summarize the parameters used in the simulations. Also we use parameters $W_k = 0.4, W_k = 0.2, W_k = 0.4, k=1$. 

<table>
<thead>
<tr>
<th>TABLE I. SIMULATION PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total simulation time</td>
</tr>
<tr>
<td>Simulation area</td>
</tr>
<tr>
<td>Total number of nodes</td>
</tr>
<tr>
<td>Radio range</td>
</tr>
<tr>
<td>Maximum speed</td>
</tr>
<tr>
<td>Pause time</td>
</tr>
<tr>
<td>Data payload</td>
</tr>
<tr>
<td>Traffic Type</td>
</tr>
<tr>
<td>Type of Attack</td>
</tr>
<tr>
<td>Maximum connection</td>
</tr>
</tbody>
</table>

© 2010 ACADEMY PUBLISHER
**B. Simulation Metrics**

To evaluate the proposed mechanism, we conclude the performance parameters of ad-hoc network which are most concerned as the following metrics:

**Packet Delivery Ratio**: It is the ratio of data packets successfully arrived at destinations to all data packets delivered from source nodes.

**Average Latency**: It is the mean time (in seconds) taken by data packets from source nodes to destinations.

**Routing Overhead**: It is the ratio of the total number of control packets generated and forwarded to the total number of data packets received.

**C. Analysis**

As shown in Fig. 3, it presents the results for trust vector based DSR (TVDSR) and pure DSR protocol in the presence of different number of malicious nodes. The PDR in the TVDSR protocol is about 20 percent higher than that of pure DSR protocol. The primary reason for this is that the TVDSR does take into account the trust value of the nodes according to their behaviors while pure DSR only prefers faster routes by default. In addition, in TVDSR all nodes not only continuously monitor the behaviors of their neighboring nodes, they also should make routing decision based upon their trust values in order to avoid including any malicious nodes. Nevertheless this decision-making rule can cause an increase in the routing overhead. Also an increment in the average latency compared to pure DSR has been observed. The reason for this may be that the routes in route cache are always optimal in terms of trust values but not the number of hops, because more trustworthy node cannot guarantee that it is more close to source nodes. Moreover, we can see that in presence of same number of malicious node, PDR of both the pure DSR and TVDSR are decreasing with the increment in mobility of nodes. This can be attributed to the higher mobility may cause more packets be absorbed by malicious nodes whose coverage is also enlarged by mobility.

In Fig.4, we can see the comparison results of trust vector based AODV and pure AODV. The results show that PDR of TVAODV protocol are always higher than that of the pure AODV protocol despite the increase in malicious nodes. This is because that the trust model in TVAODV computes trust values for other nodes and so these values could be used to bypass malicious nodes after route discoveries. The TVAODV protocol has a higher route overhead because of giving up fastest paths routing selection rule. Also the most trusted paths are not always the fastest; it is observed that the average latency
in TVAODV increases with the number of malicious nodes.

VII. CONCLUSION AND FUTURE WORK

In this paper, we investigate a trust mechanism in MANET based on Vector of Trust Model which is an abstract trust model. Then we show that this mechanism can be effectively integrated into the DSR and the ADV routing protocols. The design of this mechanism is to successfully detect malicious nodes and then take actions on them. Afterward we did a series of simulations to test the performance of our proposed mechanism in both DSR and AODV. The simulation results have showed that in the presence of malicious nodes in ad hoc network, the performance of TVDSR and TVAODV which are integrated with proposed trust vector evaluation mechanism are better than their standard protocols in terms of packet delivery ratio. Future work includes adapting our model to counteract more malicious attacks in MANET such as wormhole attacks, byzantine attacks, resource consumption attack etc [18], and attacks aiming at trust model itself, for example, fabricating trust recommendations and conspiring to rate each other high scores among malicious nodes, should be also taken into consideration.

REFERENCES


Wei Gong. He was born in 1982, Ph. D. candidate. His research interests include distributed computing, wireless security and mobile ad-hoc networks.

Zhiyang You. He was born in 1983, Ph. D. candidate. His research interests include trust management and availability model of ad-hoc networks.

Danning Chen. He was born in 1986, master candidate. His research interests include wireless security and availability model of ad-hoc networks.

Xibin Zhao. He was born in 1973, Lecturer. His research interests include software engineering, computer system and information security and e-commerce.

Ming Gu. She was born in 1962, professor, Ph. D. supervisor, deputy director of Key Laboratory of Information System Security of Ministry of Education. Her research interests include operating system, middleware technique, formal method, information security and e-commerce.

Kwok-Yan Lam. He was born in 1966, professor, Ph. D. supervisor, director of Key Laboratory of Information System Security of Ministry of Education. His research interests include information security, distribute computing, smartcard system, grid computing and mobile computing.