Dissertation Report

A Collaborative Security Approach for Preventing Attacks from Malicious Nodes in Two Layered Network

SUBMITTED BY:-

NEDUMPEEDIKAYIL BABU ASHOK

SCHOOL OF COMPUTER ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

June 13, 2012
Table of Contents

ABSTRACT .................................................................................................................. v

ACKNOWLEDGEMENT ............................................................................................. vii

LIST OF FIGURES .................................................................................................. vi

LIST OF TABLES ..................................................................................................... viii

CHAPTER 1 INTRODUCTION ............................................................................. 1
  1.1 Background ...................................................................................................... 1
  1.2 Objective of the project .................................................................................. 5
  1.3 Rationale of the project .................................................................................. 5
  1.4 Scope of the project ....................................................................................... 5

CHAPTER 2 RELATED WORK ........................................................................... 7
  2.1 Reputation management for distributed service oriented architectures ........ 7
    2.1.1 System architecture ................................................................................ 7
  2.2 Trust and reputation management in peer to peer network ....................... 10
    2.2.1 Trust and reputation mechanism .......................................................... 11
  2.3 Effective acquaintance management based on Bayesian learning for
distributed intrusion detection network .......................................................... 12
  2.4 Collaborative security architecture for black hole attack prevention in
mobile ad hoc network ...................................................................................... 17
  2.5 Sequoia – a robust communication architecture for collaborative security
monitoring system .............................................................................................. 20
CHAPTER 3 TRUST AND REPUTATION MANAGEMENT IN TWO LAYERED NETWORKS

3.1 Basic interactions in a two layered network

3.2 Trustworthiness of a node

3.3 Trustworthiness of a super node

  3.3.1 Service based trust

  3.3.2 Recommendation based trust

3.4 Super node’s trust on other node

  3.4.1 Direct interaction

  3.4.2 Indirect interaction

3.5 Watch dog

CHAPTER 4 SIMULATION

4.1 Intra-community interactions

  4.1.1 Detection and isolation of malicious nodes in a community

  4.1.2 Detection and isolation of both malicious super node and member
      nodes in a community

4.2 Inter-community interactions

  4.2.1 Interaction between 2 communities in a two layered architecture

  4.2.2 Interaction between 3 communities in a two layered architecture

4.3 Comparison of performance of a two layered architecture with and without
    watch dog
CHAPTER 5 CONCLUSIONS AND FUTURE WORKS..............................................56

5.1 Limitations ..........................................................................................56

5.2 Future Works.....................................................................................57

Reference ..................................................................................................58
ABSTRACT

Central management of security is not efficient enough to detect the sophisticated threats and attacks from the Internet in the present cyber world. An alternative trend is to use collaborative security approaches, where nodes may use specific knowledge both local and acquired from other nodes to make security related decisions. A good collaborative security system can improve the security of the whole group.

Two layered architecture was proposed to avoid message overhead in large networks, with less number of nodes involved in a network transaction when compared to single layer architecture. In a two layered architecture, a large network is divided into smaller sub networks or local networks. Within each sub network, one node is selected as the super node based on its capability and connectivity. The top layer consists of interconnected super nodes from each sub network and the bottom layer consists of all the member nodes under each super node.

In a two layered network the nodes acquire services and files from unknown nodes in the network. It is important for the nodes in a two layered network to model, update and share their trust with other nodes to prevent attacks by malicious nodes. Reputation and trust management is an efficient way to find a good node to interact with. When various nodes collaborate, factors related to trust, robustness, efficiency and stability of the network need to be addressed.

The aim of this project is to develop a novel super node based trust and reputation mechanism with the help of watch dogs for two layered network. This approach
relates network security aspects to human behaviour. Each node in the network will accept service from provider nodes which they trust. Trust is based on consumer node’s personal experience with the service provider node or provider node’s reputation in the community or a combination of both. Malicious node providing bad service will have low reputation and trust values in the network and gradually those nodes will be isolated from the network.

Our approach utilises Bayesian network to model trust, weighted average to aggregate reputation, Chernoff Bound theorem to calculate personalized trust and watch dog to monitor the reputation of super nodes in the community. Experiments were carried out to validate the new approach based on two layered network simulated using Java NetBeans IDE. Comprehensive test cases involving varying number of malicious nodes in single community network were tested followed by multi-community network. The simulation results confirm that our approach can effectively prevent attack caused by malicious nodes and achieve better effectiveness compared to the network that does not use watch dogs.
ACKNOWLEDGEMENT

Author would like to express his sincere gratitude to,

Assistant Professor Zhang Jie, Project Supervisor

A/P Zhang Jie has provided the author with guidance, advices and teaching throughout the course of the project. He has given the author immense freedom to explore his own ideas and work around with that. The supervision, support and encouragement he gave had truly helped in the completion of the project.

The author would also like to thank Mr. Hoang Tri Tam, Research Student, School of Computer Engineering, Nanyang Technological University for his support and kind patience in using the software for this project.
LIST OF FIGURES

Figure 1: The Overlay Network of Collaborative Nodes .................................................. 2
Figure 2: Single layered architecture .............................................................................. 4
Figure 3: Two layered architecture .................................................................................. 4
Figure 4: Community based two layered network ......................................................... 25
Figure 5: Number of bad transaction in a community with one malicious node ............ 37
Figure 6: Super node’s trust on the malicious node in the community ............................. 37
Figure 7: Number of bad transaction in a community with 40 malicious member nodes ..................................................................................................................... 38
Figure 8: Super node’s trust on malicious nodes in a community with 40 malicious member nodes .................................................................................................................. 38
Figure 9: Member node’s trust on malicious nodes in a community with 40 malicious member nodes .................................................................................................................. 38
Figure 10: Number of bad transaction in a community with 80 malicious member nodes ............................................................................................................................ 39
Figure 11: Member nodes trust on malicious member node in a community with 80 malicious member nodes ................................................................................................. 39
Figure 12: Super node's trust in malicious member node in a community with 80 malicious nodes ...................................................................................................................... 40
Figure 13: Number of bad transaction in a community with malicious super node ....... 41
Figure 14: Average trust value of malicious super node in the community ..................... 42
Figure 15: Number of times watch dog is triggered in the community where super node is malicious .......................................................................................................................... 42
Figure 16: Number of bad transaction in a community with malicious super node and 50 malicious member nodes ............................................................................................ 43
Figure 17: Trust of the super node in the community with malicious super node and 50 malicious member node

Figure 18: Number of times watch dog was triggered in the community with malicious super node and 50 malicious member nodes

Figure 19: Trust of SN1 on SN2

Figure 20: Trust of SN2 on SN1

Figure 21: Number of interaction from C2 to C1

Figure 22: Trust of SN1 on SN2 in a 2 layered network

Figure 23: Trust of SN2 on SN1

Figure 24: Number of interaction between C1 and C2 in the two layered network

Figure 25: Trust of SN1 on SN2 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 26: Trust SN1 on SN3 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 27: Trust of SN2 on SN1 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 28: Trust of SN2 on SN3 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 29: Trust of SN2 on SN1 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 30: Trust of SN3 on SN1 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 31: Number of bad transaction from C2

Figure 32: Performance of network with and without watch dog
LIST OF TABLES

Table 1: Trust values used for example ................................................................. 29
Table 2: Trust values used for example ................................................................. 33
Table 3: Initial parameter used for simulation ..................................................... 36
Table 4: Initial parameter for simulation of malicious super node in a community ... 41
Table 5: Initial parameter for inter-community simulation ................................. 45
Table 6: Initial setting used for the simulation of two layered architecture with 3 communities interacting each other ............................................................... 49
Table 7: Initial parameter for simulation of performance of a network with watch dog 54
CHAPTER 1
INTRODUCTION

1.1 Background

Intrusion, the term used to collectively represents the attack caused by worms, viruses, spywares, spamware, malicious nodes, insider threats, denial-of-service (DoS), distributed-denial-of-service (DDoS) [2] etc. In the recent years, intrusion has become a growing concern due to the increasing complexity to detect the potential damages caused by viruses. Recent studies in finance and banking sector pertaining to insider intruders indicates that over 25% of the fraud reported by users actually involved customers and other co-workers [3-7].

Collaboration, the term implies “something created by working jointly with another or others”. Collaborative security means several nodes working jointly to detect invasion or malicious nodes. The nodes share their knowledge and experience about the malicious nodes within the group, which consequently assist in improving the performance of the whole group.

As a promising tool for solving complex security tasks, collaborative monitoring is better explained in the example detailed below. Employees in an organization are provided with unique access cards with distinctive access facilities within it. Monitoring each employee and their access points are truly difficult with limited infrastructures and security officers. An intruder with a fake/random access card, in this case, is better detected by a user/employee who is familiar and regularly working on that facility, than by an existing monitoring infrastructure [4, 8].
Intrusion has become a severe issue in recent years due to its impending substantial damage combined with its hard-to-detect complexity. To protect nodes from malicious intrusion, Intrusion Detection Systems (IDSes) were designed to monitor traffic and computer activities. On the detection of suspicious activity, IDSes will raise alert to the network administrator. Since an IDS of any single vendor, does not have a comprehensive knowledge of all intrusions, they can be easily compromised by a novel or unfamiliar threats. Collaboration among nodes in IDS can overcome this weakness, by sharing collective knowledge and experience among peer nodes.

![Figure 1: The Overlay Network of Collaborative Nodes](image)

Figure 1 describes a Collaborative Intrusion Detection Network System (CIDS) in which several nodes are connected to each other in a peer-to-peer manner. In the event a node detects suspicious behaviour from an executable file, but lacks confidence to make a decision, it will consult with the other nodes to which it is connected and those nodes will share their experience and knowledge which will help the node to make a decision. The whole network is thus benefited by this collaboration.

Collaboration among nodes in IDS can predictably increase the security by having each node benefit from the collaborative knowledge and experience shared by peer nodes, predominantly in peer to peer (P2P) systems which enable files to be hosted by unknown
users. This suggests that a number of trust and reputation issues are to be addressed in P2P networks in order to help users discover the desired authentic data.

A Reputation system thus emerged as a substitution for enforcement principle used in e-commerce. Reputation systems collect and aggregate feedback about behaviour and help nodes to make decisions about whom to trust. Past experience with a remote partner is projected into the future, giving a measurement of their trustworthiness. In the physical world, capturing and aggregating feedback is neither easy nor cheap, but in a network it is cheap, easy and evidently efficient.

Reputation system consists of a propagation mechanism that allows the nodes to obtain reputation value. There are two available approaches for reputation propagation: centralised and decentralised approaches. In a centralised approach, the reputation value is stored in a central server and nodes get the reputation value from it by forwarding their query to the central server. Centralized system is more vulnerable to one point attack[9-11]. In a decentralized approach each node retains and manages reputation of the other nodes, which they interact with. Every time a need arises, the user can ask other nodes for the desired values.

A key issue that ascends in a peer to peer network is that, the messages and files from a source has to pass through several intermediate nodes before it reaches the destination because all the nodes are not directly connected. This issue is resolved by the use of two layered or super node architecture, wherein, a large network is divided into smaller sub networks or local networks. Within each sub network, one node is selected as the super node based on its capability and connectivity. The two layered architecture comprises of a top layer of interconnected super nodes from each sub network and the bottom layer of all the member nodes under each super node. During a message or file transfer, the
number of intermediate nodes involved in a transaction will be less in a two layered architecture when compared to that of a single layered architecture.

Consider a one layered architecture as shown in Figure 2. If node ‘A’ wants to send a message to node ‘G’, it has to pass through nodes ‘B’, ‘C’ and ‘F’ before it reaches the destination node ‘G’.

![Figure 2: Single layered architecture](image)

Consider the same scenario in a two layered architecture with ‘D’ and ‘E’ as super nodes as shown in the Figure 3. In this case the message from the source node ‘A’ has to pass through super nodes ‘D’ and ‘E’ before it reaches the destination node ‘G’. Only two intermediate nodes are involved there by increasing the performance of the network.

![Figure 3: Two layered architecture](image)

In a CIDS the message from a source has to pass through several intermediate nodes before it reaches the destination, since all the nodes are not connected directly. A two layered architecture significantly reduces this message overhead caused due to test and feedback messages. Two layered network consists of several clusters called
community/domain, each community with a super node that controls and monitor the activities and interactions of all other member nodes in that community.

1.2 Objective of the project

The project addresses the issues caused by malicious nodes in a two layered network. A member node or even super node can be malicious. Thus a two layered network will be greatly benefited by the use of collaboration that allows the nodes to gather, share and deliver information, thereby isolating the malicious node and increasing the performance of the entire network. This project introduces a novel community based collaborative security approach for two layered networks.

1.3 Rationale of the project

The current two layered collaborative approach presumes the super node to be always trustworthy [13][14][19][30]. Previous approaches in network monitoring did not incorporate super nodes into their monitoring mechanism. The approach introduced in this project incorporates super nodes into the monitoring mechanism, where the activities and trust of super node is continuously monitored by its member nodes and other super nodes in the network. On the detection of malicious activity by a super node, watch dog will be triggered which will aggregate the reputation of the super node in the community. Low reputation of a super node will result in the re-election of super node in that community. The project aims to develop a novel super node based trust and reputation mechanism without centralized control for two layered networks with the help of watch dogs.

1.4 Scope of the project

The scope of the project includes development of an approach which allows nodes in a two layered network to interact with one another in a healthy way. That is the
likeminded nodes will interact themselves isolating the malicious nodes. A node will receive information and services from nodes of the same or higher trust level where as it will provide information and service only to nodes with same or lower trust level thereby isolating the malicious node from providing services. Experiments were carried out to validate this approach under a simulated environment created by using Java Net Beans IDE. Comprehensive test cases involving varying number of malicious nodes in single community network were tested followed by multi-community network. The experimental results confirm that the new trust and reputation management approach developed with the help of watch dogs can effectively prevent attack caused by malicious node and achieve better effectiveness compared to the system that does not use watch dogs.

The report provides a brief introduction of the current collaborative security approaches used in two layered networks and some other related works. The new system of approach is then explained with the assistance of various examples followed by simulation results. Conclusions on this new approach including its possible limitations and few suggestions for future improvements are provided in the final chapter.
CHAPTER 2
RELATED WORK

A robust two layered architecture has to address the issues like how nodes in a local network evaluate the trust of other nodes and how to brush off the compromised node form the network. The rest of the chapter explain some of the related works in this topic

2.1 Reputation management for distributed service oriented architectures

Reputation management approach proposes a system capable of both managing the service provider’s reputation and of estimating real Quality of Service (QoS) values in a Distributed Service Oriented Architecture (D-SOA) and is well suited where trusting authority is implicitly distributed [12].

2.1.1 System architecture

D-SOA is a two level hierarchical network. Top level subsystem constitutes of a set of Super nodes forming an overlay network which act as a distributed trusting authority. The low level consists of the service exchange subsystem and its components called nodes. Nodes are service consumers and providers. The task of the super node is to monitor service exchange activity occurring at the low level and update information both on QoS and provider’s reputation for supporting service selection. Physically the network is partitioned into small clusters called domains. Each cluster consists of a super node which monitors the activities of all nodes belonging to its domain [13, 14].

Each Super node maintains a reputation value for every member nodes in its domain and a reputation value for any neighbouring super node in the overlay network.

In a service transaction there are mainly 4 nodes involved.

- Consumer Node (CN): the node which receives service.
• Provider Node (PN): the node which provide service to consumer node

• Seeker Super Node (SSN): The super node that is the cluster head of the domain where consumer node belongs.

• Manager Super Node (MSN): The super node that is the cluster head of the domain where Provider node belongs[14].

The event flow for a typical transaction can be explained as follows

1. When CN requires a service, it will send query to its SSN.
2. SSN will forward the query from the CN to its neighbour SN in the overlay network.
3. When a Super node receives a query, it performs a local search for the service provided by nodes in its domain.
4. Each node replies to the local query.
5. The super node replies to the SSN with the list of service matching the query along with QoS values for each service.
6. SSN will aggregate all the list of query replies from all MSN and will forward to CN.
7. The CN will select a service out from the received list using reinforcement learning model and after consuming the service it will send the feedback to the SSN. SSN will update its reputation based on the feedback.
8. SSN will forward the feedback from the CN to the MSN and MSN will update the reputation of PN based on this feedback [14].

Q - learning is used for QoS estimate. The average utility of performing an action $\alpha$ in a state $s_t$ is referred as $Q ( s_t, \alpha_t )$. $Q ( s_t, \alpha_t )$ is updated based on the past estimate and the reward $r_{t+1}$ obtained from the interaction, according to the following equation [15]
\[ Q(s_t, a_t) \leftarrow (1 - \alpha) Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_\alpha Q(s_{t+1}, \alpha) \right] \] (1)

Where \( Q(s_t, a_t) \), the current estimate of the utility is obtained by performing the action \( a_t \) in the state \( s_t \), \( s_{t+1} \) is the new state, \( r_{t+1} \) is the reward obtained and \( \max_\alpha Q(s_{t+1}, \alpha) \) is the maximum reward obtainable in the new state. The parameters \( \alpha \) and \( \gamma \) represent the learning rate and the discount factor ranging in \([0, 1]\) respectively. These two parameters are used to control the learning mechanism.

The consumer node based on the \( Q(s_t, a_t) \) values; perform the comparison using reinforcement comparison technique. Each action can be selected with a probability \( \pi \) directly related to its estimated average rewards, computed using \([14]\),

\[
\pi_t(a, s) = P_r\{ a_t = a \mid s_t = s \} = \frac{e^{Q(s_t, a_t)/\tau}}{\sum_\alpha e^{Q(s_t, a_t)/\tau}} \] (2)

Where parameter \( \tau \) is the Boltzmann distribution, which makes the actions quite equiprobable.

After receiving replies from MSN (Step 6) the SSN aggregate the replies using \([16]\),

\[
Q_{o_sacc} = Q_{o_sadv} \times \frac{rep}{rep_{max}} \] (3)

Where \( rep \) = reputation value of MSN providing service

\( rep_{max} = \) maximum reputation value of all neighbour super nodes

\( Q_{o_sadv} = \) QoS advertised by the MSN

\( Q_{o_sacc} = \) Filtered QoS by SSN
D-SOA provides an architecture which overcomes all the limitation of centralized system. D-SOA helps consumer in the selection of the best service. D-SOA system is based on a decentralized reinforcement learning approach [17] that allow consumer node to learn the best service in order to maximize the perceived QoS. The architecture also has a reward mechanism to motivate provider nodes for their honest behaviour. The approach is designed on the assumption that super node cannot be compromised. It also has a decentralized architecture which makes it vulnerable to antisocial behaviour [18].

2.2 Trust and reputation management in peer to peer network

P2P networks are networks where peers co-operate to perform a function in a decentralized manner. Each peer in the network can act as both consumers and providers of resources and services. Since peers are heterogeneous, some nodes might be benevolent where as some other might be buggy or malicious in providing service. Mechanisms of trust and reputation can be used to help peers to distinguish between good and bad partners.

In a P2P network trust can be defined as a peer’s belief in another’s capabilities, honesty and reliability based on its own experiences, where as reputation is the peer’s belief in another peer based on recommendations received from other peers. Trust and reputation are dynamic in nature because it may increase or decrease with experience and also decay with time. Trust and reputation are context specific. A service rated as good by one peer may be bad to another peer. Even in the same context different peers may have different opinion about the service, so it can be considered as multi faceted[19].
2.2.1 Trust and reputation mechanism

Each peer in the network keeps two kind of trust in another peer. The first one is the trust on the other peer’s capability in providing service and the second is the trust on the peer’s reliability in providing recommendations about other peer’s. Reliability can be considered as the union of truthfulness and similarity. Truthfulness means how truthful the peer is in telling its information and similarity is the measure of resemblance between one peer and another in preferences and ways of judging issues. Since peers are heterogeneous, they may have different preferences, evaluation criteria and judge issues [19, 23].

To explain the approach let’s consider file sharing application as an example. In a file sharing application, there is: a provider, a consumer and referees. The peer receiving the service is called consumer, the peer providing the file is called file provider and the peer who giving recommendation is called referees. A peer will develop two kinds of trusts, one based on the capability in providing file and other the trust in peer’s reliability in making recommendations. The approach assumes the entire peer to be truthful in telling their evaluation[19].

A search request for a file in P2P will result in a long list of providers. Once the peer receives a list of file providers for a given search, it will arrange the list according to its trust. Then the peer chooses the file provider with the highest trust. If the file provider is trust worthy according to the peer’s past experience it will download the file from the provider else it will select another file provider. If the peer has no previous interaction with the file provider it will ask other peers to make recommendations. So based on the personal trust and the reputation, the peer will decide whether to accept the service from the file provider or not.
After each interaction the peer updates its trust in the file provider based on his experience. If the experience is satisfying it will increase the trust or decrease the trust. If the decision of interaction was based on other peer’s recommendations, the peer will also update the trust on other peers that gives the recommendation. That is the peer will increase the trust on the peer which recommended correctly and decrease the trust of those who made wrong recommendations.

After interaction with a file provider the peer will update its trust on file provider and also will update the trust on the referee-peer using the formula [19]:

$$tr_{ij}^n = \alpha tr_{ij}^o + (1 - \alpha) e_\alpha$$

Where $tr_{ij}^n$ denotes the new trust value that the $i^{th}$ peer has in $j^{th}$ after update and $tr_{ij}^o$ is the previous trust value. $\alpha$ is the learning rate which can be a real number between 0 and 1. $e_\alpha$ is the new evidence, which can be -1 or 1. In the case of recommendation if there is a mismatch between recommendation and actual experience with the file provider, the evidence is negative, so $e_\alpha = -1$ and if there is no mismatch it will be positive that is $e_\alpha = 1$. Peers also can gossip with each other to find other peers reliability.

2.3 Effective acquaintance management based on Bayesian learning for distributed intrusion detection network

Intrusion Detection system (IDS) as the name implies is a monitoring system which will monitor on network traffic and computer activities and raise alarm to administrator in case of intrusion. Basically of two types: host based Intrusion Detection Systems (HIDS) or Network Intrusion Detection Systems (NIDS) based on their targets. Signature based IDS or anomaly based IDS according to their detection methodologies.

NIDS monitors the network traffic and computer activities and compare it with a known data if any abnormality is found it will raise alarm[20]. HIDS capture intrusion by
comparing observable intrusion data such as log files[21, 22]. Signature based IDS check for matching signature of the file with the database if it matches with malicious signature in the database, alarm is triggered. Anomaly generates alarm if traffic volume or computer activity is beyond a predefined value.

HIDS alone can be easily compromised. Collaboration techniques can improve the overall intrusion detection efficiency. So let’s look into Collaborative IDN, which is an overlay network which provides collaboration for HIDS. It can be centralized or distributed.

Problem faced by collaboration are, malicious insider can mislead whole HIDS and collaborator are of different expertise and capabilities. Selection of appropriate collaborator from a large number of nodes is a major challenge. To address this challenge “Acquaintance Selection” algorithm is implemented.

HIDS will make use of Bayesian learning technique to identify experts and novice node based on past experience with them. It is done by evaluating false positive and false negative rate of the node and dishonest collaborator are identified and removed. It also suggest Bayesian decision making model for HIDS to minimize cost of the making false decision. The proposed algorithm can select collaborators dynamically in any contest setting to achieve high efficiency at low cost.

A probation list is also maintained. So a new collaborator will be in probation list for certain days before their feedback is considered. A probation list node after staying in the probation list for a threshold amount of time (tp) is called mature node. Only mature node is selected to acquaintance list.
In CIDN each node is connected to collaborator in peer to peer manner. The list of collaborator a particular node is connected with; is the acquaintance list of that particular node.

When a node in HIDS detects suspicious activity but not confident enough to raise an alarm it will send consultation text to the nodes in the acquaintance list. The consultation request contains details of the suspicious files. Then all the feedbacks from collaborators are collected and final decision is made. But it is important for the node to maintain a healthy acquaintance list in order to make accurate decision.

To test the trust worthiness of nodes in acquaintance list, the node will send a consultation test message to the collaborators in the acquaintance list. Based on the feedback from each collaborator and comparing it with the known result; trust of each collaborator node is evaluated. Further extending on this concept, the nodes in the probation list is also tested. For a suspicious file, consultation request will be send to nodes in acquaintance list and probation list. Result from the acquaintance list is compared with that from the nodes in probation list to measure the trust of nodes in probation list.

Acquaintance management algorithm updates acquaintance list periodically by recruiting new nodes and remove dishonest nodes. More acquaintance gives more accuracy and lower risk of being compromised. But having more nodes in acquaintance list causes high maintenance cost. So adding apt node to acquaintance list is important.

*Acquaintance Selection Algorithm:* The algorithm looks for node which brings lowest overall cost and stores it. And stored node is added to acquaintance list if current acquaintance length is lower than minimum value or if the cost of reduction is greater than zero and current length is not reached the maximum length[1,16].
Acquaintance Management Algorithm: A node has acquaintance list and probation list. It will send test message to nodes in both list and update the estimated false positive and true positive rate of them. The algorithm searches for optimal collaborators which are matured and honest (stayed in probation list for some time duration). Once such nodes are identified, acquaintance selection algorithm is used to find optimal node. Once collaborators are identified, request is sent out for node to join acquaintance list. If the corresponding node responds within a fixed time it will be added to the list. The list is updated after fixed amount of time [1,16].

Threats and Defence: The collaboration management itself can be target of attack and it can be compromised. Some of the common threats and defence against those threats are explained below

Sybil Attack: This attack occurs when a malicious node in the system creates a large amount of fake identities. This fake identity created is then used to gain influence over the network by providing false diagnosis. For example, fake node will gain trust from some honest nodes and then will recommend false diagnosis which cause false negative decisions from the victims. To protect the HIDS against Sybil attack, authorization mechanism can be integrated which make creation of fake identification difficult. In addition collaboration required HIDS to allocate resources to answer diagnosis request from their collaborators, which is costly to do with many fake identities [24] [25].

Newcomer Attack: Newcomer attack occurs when a malicious node can easily register as a new user. For example if a malicious node creates a new ID, it will allows that node to erase its previous bad history. However in the new model discussed above, it is not easy for newcomer to launch attack since a newcomer requires probation period for all newcomers. It takes fairly long time for a new node to prove its trustworthiness and pass
probation period. Only after the new node passes probation then only their feedback is considered by other nodes in the network. Then if a node wants to erase its bad history by registering new ID, it has to go through the probation period before it can make influence on the system [26] [25].

Betrayal Attack: It happens when a trusted node suddenly turns into malicious node and send false diagnosis in the network. The performance of the whole system is degraded by this kind of attack. But the current algorithm enables fast learning of malicious behaviour and the dynamic acquaintance algorithm will remove the malicious node from its acquaintance list at the same time the probation period make it difficult for such removed nodes to come back to the network easily.

Collusion Attack: Acquaintance selection algorithm employs random selection of potential collaborators; it will be unlikely for multiple peers to be selected together by one peer as collaborator.

Evaluation of the new algorithm: First experiment where conducted to find the apt test message rate. For various values of test message rate 2/day, 10/day and 50/day is tested and 10/day seems optimal. Increase in the test message rate is directly proportional to convergence time for false positive. And convergence time is directly proportional to false negative value.

Second stimulation was to analyze cost and number of collaborators. The stimulation result shows that introduction of more collaborators will decrease the risk cost. However it is also noticed that after an optimal point adding a new collaborator will increase the cost. This is because when the number of collaborators becomes large enough, the cost saving by adding more collaborators becomes small and the maintenance cost become
more significant. So the group with higher detection accuracy will have lower optimal costs and smaller number of collaborators to reach that optimal cost.

Next area explored was efficiency of feedback aggregation. Bayesian and threshold based feedback aggregation were tested. Result shows that threshold based decision largely depends on the chosen threshold value. But Bayesian decision prevail threshold under all settings.

When testing brute force algorithm with acquaintance list algorithm it turns out that brute force has slightly better cost efficiency. When it comes to the case of running time for smaller candidate list (<11) brute force is better. As the candidate list becomes larger, greedy algorithm is more efficient. Considering convergence, simulation results show that initially HIDs will collaborate with random nodes but after time passes the collaboration will be with nodes which are of similar detection accuracy levels. Stability test shows that collaboration with nodes with similar expertise level is stable than those with different expertise level.

It shows that the new algorithm discussed above helps the nodes to find small number of best collaborators with which they can interact and thereby minimizing the false intrusion decision cost and the maintenance cost. Algorithm to find the optimal length of acquaintance list based on detection accuracy remains open for further work.

2.4 Collaborative security architecture for black hole attack prevention in mobile ad hoc network

An ad hoc network is an infrastructure-less group of wireless mobile nodes. These nodes cooperate among themselves by forwarding data packet. These will allow nodes to communicate beyond their wireless transmission range. All nodes in the network are free to move around.
Black hole is one of the major problems faced in the ad hoc wireless networks. Mobile ad-hoc network is dynamic in nature, in such a dynamic network malicious and selfish node can have an adverse effect in the network. Even a small number of malicious nodes can degrade the network throughput.

Usually in an ad hoc network protocol, any intermediate node in the network could respond to a route request packet if it has a fresh enough route to the destination. During the formation of network the protocol assumes all nodes in network as trusted nodes, but in reality that’s not the case. This will allow malicious node to crash the network easily by replying to the route request. The reply from malicious node will be faster than that from a normal node because malicious node does not have to check its routing table to reply to the route request.

Watch dog is an existing concept against black hole attack. This technique is used to improve the throughput in an ad-hoc network. Watch dog is a node in the network itself. Each watch dog will be assigned the duty to monitor the activity of a particular node. In a watch dog method, the watch dog buffers the data packets which were recently sent and track the misbehaving node. The watch dog watches the node to which data packet is send and it makes sure the packet is forwarded to next node. The data packet received by a node has to be forwarded within time out period [27] [28]. This method has some draw backs like a compromised node may provide false report about other node.

Keeping in mind the drawbacks of watch dog method, an extension was recommended. Nodes in the network were classified into three categories: trusted, watch dog, and ordinary nodes. In the current watch dog concept; every node will be assigned as a watch dog for some other node which makes room for false reporting. In the new method only trusted nodes are selected on a random basis to be watch dog. Nodes during the formation of the network are considered as trusted nodes and only trusted nodes are
selected as watch dogs. Every node joining the network has to prove its trust worthiness to be promoted as trusted node. The algorithm assumes that the nodes present in the beginning of the network are trusted nodes. Another assumption is that trusted node doesn’t show malicious and selfish behaviour. Watch dogs are selected from trusted nodes only for a given period of time. After that particular time period, another set of watch dogs are selected. The watch dog selection depends on the node energy, node storage capacity available and nodes computing power. The watchdog selected will have additional duty of monitoring the nodes in the network for malicious behaviour for a fixed period of time.

Two thresholds are maintained for all nodes: Suspected_Threshold and Acceptance_Threshold. If any node crosses the Suspected_Threshold it will be declared as malicious. Acceptance_Threshold is the measure of good behaviour of a node. Over a specified period of time if a node crosses this threshold is considered trusted node. Usually Acceptance_Threshold is set to a high value in order to increase the quality of trusted nodes.

The simulation of the algorithm gives promising result in defending black hole attack and it also increases the throughput of the network. In contrary to these advantages, message overhead in the network increases due to the introduction of additional packets. The increase in overhead can be reduced by increasing the time period between the two watchdog selection routines [29].

2.5 Sequoia – a robust communication architecture for collaborative security monitoring system

Sequoia is a self organized, resilient, adaptive, high quality and reliable communication architecture. Highly secure communication is achieved in Sequoia by controlling a set of mechanisms for trust management of participating nodes and trust-based routing.

The design of sequoia is based on open proxy blacklist and worm defence. It consists of three protocols.

The Monitor Neighbour Discovery Protocol (MND): In this topology every monitor node is connected to nearby nodes as its neighbour. A new node joins the network by contacting a known landmark node and obtains its coordinates. The monitor node then chooses its neighbour based on round trip measurement. Local gossiping is used to further optimize the neighbour list[30].

Distributed Dominator Selection Protocol (DDS): A flat neighbourhood overlay is created by MND. Now the second level of hierarchy is created. A monitor node which meets some minimum requirement regarding the trust and performance is then selected as dominators, which are in the top level of the hierarchy. The nodes in lower level are called dominees. Dominators are elected by nodes in the lower level. So dominator acts as a hub with dominee nodes connected to it. The two layered architecture prevents the malicious nodes from forwarding security information to others[30].

Communication Path Discovery Protocol (CPD): It helps to discover multiple paths between sender node and destination node considering efficiency and security parameters. Trusted path are found using a distributed protocol which maximizes the trust rating of a path[30].
If dominee ‘A’ wants to send a data packet to dominee ‘B’ in a different group, ‘A’ will first transmit the data to dominator of that group and the data is then transferred through a route consisting of only dominators and finally it reaches the dominator of the group containing ‘B’. From that dominator, data is passed to the destination. Dominator’s performance and trustworthiness are examined to form the communication path [30].

Sequoia uses a self organized topology without centralized control to support rich, fault tolerant and secure communication. The Sequoia does not address the issue of an instance where dominee is compromised.

**Summary**

This chapter briefly explains several existing techniques used in two layered network to prevent attack caused by malicious nodes. First the report explains “Reputation Management for Distributed Service-Oriented Architectures” which helps consumer nodes in the selection of the best services among the huge multitude provided by the network. This work proposes a new decentralized methodology for trust management where malicious service provider that declares QoS higher than the real value will be penalized.

Then the report expounds on “Trust and Reputation Model in Peer-to-Peer Networks”, which uses Bayesian network to model trust and reputation. Bayesian networks provide a flexible method to model differentiated trust and combine several aspects of trust.

“Sequoia”, a protocol based architecture was explored under section 5 of the chapter. Sequoia architecture comprises of three protocols through which monitors (dominators) self organize into a two-level hierarchy on which scalable, fast and trustworthy message delivery can be achieved. Sequoia provides a mechanism where the message from the provider node to the consumer node passes only through trustworthy dominators.
All the three approaches do not consider the possibility of compromised super node. Super node plays a crucial role in two layered network, a malicious or compromised super node will have adverse effect on network performance. So it is really critical to monitor the super nodes in the network.

Report further looked into “Collaborative Security Architecture for Black Hole Attack Prevention in Mobile Ad Hoc Networks” where watch dogs are used to monitor every transaction in the network. The use of watch dogs in the network will increase the message overhead in the network due to the introduction of additional packets.

“Effective Acquaintance Management for Collaborative Intrusion Detection Networks” was examined under section 3 of the chapter, which focus on acquaintance management where each node select and maintain a list of collaborators which they trust. The node will consult about intrusion among the nodes in his acquaintance list. The simulation results show that the new algorithm discussed above helps the nodes to find small number of best collaborators with which they can interact and thereby minimizing the false intrusion decision cost and the maintenance cost. Algorithm to find the optimal length of acquaintance list based on detection accuracy remains open for further work.
CHAPTER 3

TRUST AND REPUTATION MANAGEMENT IN TWO LAYERED NETWORKS

One of the major issues in peer to peer network is that the message and file from a source has to pass through several intermediate nodes before it reaches the destination since all the nodes are not directly connected. But it is not necessary that all the intermediate nodes are trustworthy and message overhead in the network will increase.

The solution for the above mentioned issue is the use of two layered or super node architecture. A large network is divided into smaller sub networks or local networks. Initially in a sub network, a node is selected as leader or super node based on its capability and connectivity. So the top layer of the architecture consists of the entire super node connected to each other and the bottom layer consists of all the normal nodes. In a subnet, the nodes maintain a reputation value of other nodes based on their personal experience. Whenever there is a re-election of super node, they will consider this reputation value to elect the new super node.

Super node plays an important role in the two layered network. Super node acts as a middle man between all indirect interactions. So if the super node is compromised it will affect the whole community. Super node maintains a trust value on every node in the network and other super nodes which it is connected to. When a member node wants a service, it issues service request and it is passed to all its neighbouring nodes. Super node is connected to all nodes in the domain, so all nodes will have a super node as one of its neighbouring node.

The node requesting a service is called consumer node and the node providing service to consumer node is called service provider or service providing node. If any node in the network provides the service requested by the consumer node, the consumer node will
accept the service from the node based on its personal experience. If the consumer node doesn’t have enough experience with the service providing node it will ask the super node for reputation value of the service providing node. Based on the received reputation information, the consumer node can then model the trustworthiness of the service providing node. Any node after receiving service from another node, will update its trust on the provider node. In case of indirect transaction it will also send a feedback to the super node based on the quality of the service. Super node maintains trust values of all the member nodes under it, along with other super nodes in its neighbour list. All nodes will maintain a trust value on the node which they have interacted. When it comes to the intra community interaction, the interaction will be based on the reputation of the super node of the community in which the provider node belongs to [31]. The network is open so that any node can join or leave the community at any time. So when a node does not get enough services from a community it is free to leave the community and join another community.

3.1 Basic interactions in a two layered network

A two layered network is shown on the figure 4. In a two layered architecture, depending on their mode of interaction they can be classified as, inter-community and intra-community interactions. Intra-community interaction is when two nodes in the same domain interact with each other whereas when a node in one community interacts with node in other community it is called inter-community interaction.
Depending on the number of intermediate nodes involved in an interaction, it can be classified into

- Direct Interaction
- Indirect Interaction

When a node gets service or file from other node without any intermediate node in between, it is considered as a direct interaction. The receiver node which receives the service and the node providing the service will be directly connected. In the figure 4, node ‘A’ receiving a service from super node ‘SN1’ is an example for direct interaction.

Indirect interaction is when a node receives service from other node to which it is not directly connected or the transaction passes through intermediate nodes. In figure 4, consider the case when node ‘A’ wants a service which is provided by node ‘H’. Node ‘A’ and ‘H’ are not directly connected, so node ‘H’ will send the service to its super node SN2. Super node will then forward the service to the super node ‘SN1’ from there service will be send to node ‘A’. This is an example for indirect interaction.

3.2 Trustworthiness of a node

When consumer node ‘C’ want to accept service from other node ‘S’, it will first use its own experience. After each time of using the service, ‘C’ will updates its trust on ‘S’
based on the quality of service provided by node ‘S’. The quality of service provided by ‘S’ can be “satisfying” or “not satisfying”, which is used to update the trust of node ‘C’ on ‘S’. After interaction the consumer node will update its trust on the provider node according to the following reinforcement learning formula [18]

\[ T_c(s) = \alpha_s T^E_c(s) + (1 - \alpha_s) e(s) \]  

(5)

Where \( T_c(s) \) denotes the trust value of the provider node after the update based on the consumer nodes personal experience with the service. The consumer node will also forward the \( e(s) \) value to super node of the community it belongs to; \( T^E_c(s) \) denotes the trust value before update; \( \alpha_s \in (0,1) \) is the learning rate for service; \( e(s) \) is the evaluation of the interaction represented by either ‘1’ for “satisfying” or ‘0’ for “not satisfying”.

\[ \Delta T_c(s) = \left| T_c(s) - T^E_c(s) \right| \]  

(6)

Where \( \Delta T_c \) the difference in trust on the provider node is before and after update by the consumer node, \( \Delta T_c(s) \) is inversely proportional to the learning curve value. That is, higher the learning curve value lower will be \( \Delta T_c(s) \) and vice versa.

If the consumer node ‘C’ does not have enough personal experience with ‘S’, it will ask the super node ‘SN’ for reputation about the provider node ‘S’. The super node will then ask reputation opinion about node ‘S’ from all the nodes under super node. The super node will provide reputation about the service, which will be a value in the interval \([0,1]\) where 0 means reputation of the node ‘S’ is bad and 1 means node ‘S’ has a very good reputation. The super node will aggregate the reputation value according to the following weighted average formula [32]

\[ R_{SN}(s) = \frac{\sum^n_{i} T_{SN}(n_i) R_{n_i}(s) + 1 \cdot R_{sn}(s)}{\sum^n_{i} T_{SN}(n_i) + 1} \]  

(7)
Where $T_{SN}(n_t)$ is the super node’s trust in the node $n_t$, $R_{n_t}(s)$ is the reputation opinion about the node ‘S’ provided by $n_t$ and $R_{sn}(s)$ is the super node reputation about ‘S’.

The consumer node will calculate its reputation on provider node using

$$R_C(s) = R_{SN}(s) T_C(sn) \tag{8}$$

Where $R_C(s)$ is the reputation about provider node by consumer node; $R_{SN}(s)$ is the aggregated reputation opinion about the node ‘S’ provided by SN; $T_C(sn)$ is the trust value of consumer node on super node.

The decision whether to trust the provider node ‘S’, depends on the personal experience of the consumer node and the aggregated reputation opinion of the super node. The trust on the provider node is calculated as follows [33]

$$T_C(s) = \omega T_C^1(s) + (1 - \omega) R_C(s) \tag{9}$$

Where $T_C^1(s)$ represent the old trust value of the consumer node about the provider node; $\omega$ represents the weight assigned to $T_C^1(s)$. If the trust calculated is below the threshold trust requirement of the consumer node it won’t accept the service from the provider node. After receiving the service, depending on the quality of service the trust value is again recalculated using equation 1.

$\omega$ is calculated based on the number of interactions between the consumer node ‘C’ and the provider node ‘S’. Chernoff Bound theorem [32-36] is used to calculate the minimum number of interaction required for ‘C’ to be confident about the trust value it has on ‘S’.

$$N_{min} = -\frac{1}{2e^2} \ln \frac{1 - \gamma}{2} \tag{10}$$
Where \( \varepsilon \) is the maximal level of error that can be accepted by ‘C’ and \( \gamma \) is the confidence measure.

If total number of interaction \( N_{all} \) is larger than or equal to \( N_{min} \), consumer node ‘C’ has enough personal experience with provider node, so the trust will be calculated purely based on its personal experience and the aggregated reputation value \( R_{sp}(s) \) will be ignored. Otherwise, the consumer node will also use the aggregated result along with its personal experience. \( \omega \) is calculated using [32, 33, 36]

\[
\omega = \begin{cases} 
\frac{N_{all}}{N_{min}} & \text{if } N_{all} < N_{min} \\
1 & \text{otherwise}
\end{cases}
\]  

(11)

When the consumer node has enough experience with provider node, \( \omega \) will be 1 and the aggregated reputation opinion \( R_C(s) \) is ignored. When the consumer node does not have enough interaction with provider node, \( \omega \) will be less than 1 and the aggregated reputation \( R_C(s) \) of the service also contribute in finding the trust value of the provider node.

**Example**

In the network shown in Figure 4, consider a scenario where node ‘A’ receives service from node ‘D’ and node ‘A’ has already two interactions with D.

Assume \( \alpha_s = 0.8, \varepsilon = 0.6, \gamma = 0.7 \) and the trust values and reputation value on the domain is given in table 1.

<table>
<thead>
<tr>
<th>( T_A(SN1) )</th>
<th>0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_B(D) / R_B(D) )</td>
<td>0.7</td>
</tr>
<tr>
<td>( T_C(D) / R_C(D) )</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Table 1: Trust values used for example

<table>
<thead>
<tr>
<th>$T_{SN1}(D)/R_{SN}(D)$</th>
<th>0.73</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_A(D)$</td>
<td>0.7</td>
</tr>
<tr>
<td>$T_{SN1}(B)$</td>
<td>0.6</td>
</tr>
<tr>
<td>$T_{SN1}(C)$</td>
<td>0.5</td>
</tr>
<tr>
<td>$T_{threshold}(A)$</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Aggregated reputation opinion of the super node about node ‘D’ is

$$R_{SN}(D) = \frac{0.6 \times 0.7 + 0.5 \times 0.6 + 1 \times 0.73}{0.6 + 0.5 + 1} = 0.69$$

The consumer node reputation opinion of the consumer node ‘A’ about node ‘D’ is

$$R_A(D) = 0.69 \times 0.95 = 0.66$$

Minimum number of interaction required by ‘A’ to be confident about the trust value it has on ‘D’ is given by

$$N_{\text{min}} = -\frac{1}{2 + (0.6)^2} \ln \frac{1 - 0.7}{2} = 2.63 = 3$$

The trust of node ‘A’ on node ‘D’ is given by

$$T_A(D) = 0.33 \times 0.7 + (1 - 0.33) \times 0.66 = 0.67 \quad \left( \omega = \frac{N_{\text{all}}}{N_{\text{min}}} = \frac{1}{3} = 0.33 \right)$$

After receiving service, if the service is good trust value will increase. The new trust value will be

$$T_A(D) = (0.8 \times 0.67) + ((1 - 0.8) \times 1) = 0.736$$

$$\Delta \ T_A(D) = 0.066$$

And if service is bad, trust value of A on I will decrease. The value will be

$$T_a(i) = (0.8 \times 0.67) + ((1 - 0.8) \times 0) = 0.536$$
\[ \Delta T_A(D) = 0.134 \]

In the same scenario explained above, if \( \alpha_s \) is increased to 0.9 then the \( \Delta T_a(i) \) value will decrease as follows

\[ T_a(i) = (0.9 \times 0.67) + ((1 - 0.9) 1) = 0.703 \]

\[ \Delta T_a(i) = 0.033 \]

And if service is bad, the new trust value will be

\[ T_a(i) = (0.9 \times 0.67) + ((1 - 0.9) 0) = 0.603 \]

\[ \Delta T_a(i) = 0.067 \]

### 3.3 Trustworthiness of a super node

#### 3.3.1 Service based trust

In a local community super node is modelled as a node that provides services to other nodes in the community. Since super node is more powerful than member nodes it will have large number of interaction than the other nodes. Whenever super node provides a service, the consumer node which is the node receiving the service will update the trust on the super node using the formula

\[ T_c(sn_i) = \alpha_s T_c^1(sn_i) + (1 - \alpha_s) e(s) \]

Where \( T_c(sn_i) \) denotes the trust value of the consumer node on the super node after the update based on the consumer nodes personal experience with the service; \( T_c^1(sn_i) \) denotes the trust value before update; \( \alpha_s \in (0, 1) \) is the learning rate for service and \( e(sn_i) \) is the evaluation of the interaction represented by either 1 for “satisfying” or 0 for “not satisfying”.

30
3.3.2 Recommendation based trust

When super node makes a correct recommendation to a consumer node and if this recommendation turns to be correct, the consumer node will update its trust on super node using the formula

\[ T_c(sn_i) = \alpha_r T'_c(sn_i) + (1 - \alpha_r) e(r) \]  

(13)

Where \( T_c(sn_i) \) denotes the trust value of the consumer node on the super node after the update based on the consumer node’s personal experience with the recommendation; \( T'_c(sn_i) \) denotes the trust value before update; \( \alpha_r \in (0,1) \) is the learning rate for recommendation and \( e(r) \) is the evaluation of the recommendation represented by either 1 for “correct recommendation” or 0 for “incorrect recommendation”. The initial value of a member node’s trust in a super node may be set to 0.5, which means that the super node is neither trustworthy nor untrustworthy.

3.4 Super node’s trust on other node

3.4.1 Direct interaction

A super node will have its entire member node as its neighbouring node along with all other super node it is connected to. After direct interaction with its neighbouring node, super node will update super node’s trust in that neighbouring node using the formula explained in equation 1.

3.4.2 Indirect interaction

3.4.2.1 Intra-community interaction

When a consumer node ‘C’ receives service from a provider node ‘S’, after receiving the service the consumer node will update its trust on the provider node ‘S’ and will send feedback to the super node ‘\( SN_i \)’ based on its personal experience. The super node once
receives the feedback from the consumer node, will update its trust value about the
provider node using the formula

\[ T_{SNi} (s) = T_{SN_i}^l(s) + [(1 - \alpha_f) (e(f) - T_{SN_i}^l(s))] \ T_{SN_i}^l(c) \]  \hspace{1cm} (14)

Where \( T_{SNi} (s) \) is the super node’s trust on provider node ‘S’ before update; \( T_{SN_i}^l(s) \) is the super node trust on provider node before update; \( \alpha_f \in (0,1) \) is the learning rate for feedback; \( e(f) \) is the evaluation of the feedback represented by either 1 for “positive feedback” or 0 for “negative feedback” and \( T_{SN_i}^l(c) \) is the super node’s trust on the consumer node that is providing the feedback.

### 3.4.2.2 Inter-community interaction

When a consumer node ‘C’ in one community wants some service, it will send service request to its super node, called seeker super node ‘SSN’. ‘SSN’ then will look for any reputed node on its community for providing the service. If it cannot find any node then it will forward service query to other super nodes to which it is connected. When a super node receives a query, it performs a local search for the service requested by ‘SSN’ among nodes in its community. Each node that provides that particular service will reply to the query. Super node will reply to the ‘SSN’ with ID of the node providing the service along with its reputation. If more than one node provides the same service, super node will reply to the ‘SSN’ with a list of ID’s along with its reputation value on those nodes offering the service. The queried super node is called manager super node ‘MSN’, which is the super node of the community hosting the service provider node ‘S’. The ‘SSN’ will receive a certain number of ‘MSN’ lists of services from all its neighbouring super nodes. ‘SSN’ will filter the ‘MSN’ list it receives using its trust value on each community. The aggregation is done using a similar way as explained in equation 2 and the result is forwarded to the consumer node. Consumer node then by using its personal
experience and the reputation opinion forwarded by the ‘SSN’ will decide whether to accept the service or not. Once it receives the service it will send feedback to the ‘SSN’, ‘SSN’ will forward the feedback to the ‘MSN’ and it will also update its trust on ‘MSN’ in a similar way as explained in equation 7. ‘MSN’ after getting the feedback will update its trust on the provider node ‘S’ using equation 7.

Example

In the network shown in Figure 4, consider a scenario where node ‘A’ receives service from node ‘I’.

Assume $\alpha_s = 0.8$, $\alpha_f = 0.95$, $\alpha_f = 0.95$ and the trust values and reputation value on the network is given in table 2.

<table>
<thead>
<tr>
<th>T_A (SN1)</th>
<th>0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_SN1 (SN3)</td>
<td>0.8</td>
</tr>
<tr>
<td>T_SN3 (I) / R_SN3 (I)</td>
<td>0.9</td>
</tr>
<tr>
<td>T_threshold (A)</td>
<td>0.6</td>
</tr>
<tr>
<td>T_SN1 (A)</td>
<td>0.85</td>
</tr>
<tr>
<td>T_SN3 (SN1)</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 2: Trust values used for example

$R_{SN1} (I) = T_{SN1} (SN3) \times R_{SN3} (I) = 0.8 \times 0.9 = 0.72$

$T_A (I) = R_{SN1} (I) \times T_A (SN1) = 0.72 \times 0.95 = 0.684$

After receiving service if the service is good, the trust values will increase. The new trust values will be

$T_A (I) = (0.8 \times 0.684) + ((1-0.8)1) = 0.75$

$T_A (SN1) = (0.95 \times 0.95) + ((1-0.95)1) = 0.9525$
\[ T_{SN1}(SN3) = 0.8 + [( 1 – 0.95) (1 – 0.8)] (0.85) = 0.8085 \]
\[ T_{SN3}(I) = 0.9 + [( 1 – 0.95) (1 – 0.9)] (0.8) = 0.904 \]

And if the service is bad, trust values will decrease. The new trust values will be

\[ T_A(I) = (0.8 \times 0.684) + ((1-0.8)0) = 0.5472 \]
\[ T_A(SN1) = (0.95 \times 0.95) + ((1-0.95)0) = 0.9025 \]
\[ T_{SN1}(SN3) = 0.8 + [( 1 – 0.95) (0 – 0.8)] (0.85) = 0.766 \]
\[ T_{SN3}(I) = 0.9 + [( 1 – 0.95) (0 – 0.9)] (0.8) = 0.864 \]

3.5 Watch dog

Watch dog is an existing concept used in mobile ad hoc networks to improve the throughput. Watch dog is a node in the network itself. In our approach for security enhancement, three or more watch dogs are assigned randomly from the member nodes. Each watch dog will be assigned the duty to monitor the trust of member nodes on super node. Watch dog will be re-selected randomly from the pool of member nodes at fixed time intervals. Every member node in the community is keeping a trust value on super node, if the trust value of any member node on super node goes beyond a predefined threshold value it will trigger the watch dog. The watch dog will aggregate the entire member node’s trust on the super node in a community based on the formula

\[ T_w(sn) = \frac{\sum^n_i T_w(n_i) R_{n_i}(sn)}{\sum^n_i T_w(n_i)} \quad (15) \]

Where \( T_w(sn) \) is the watch dog’s trust in the super node \( sn \); \( T_w(n_i) \) is the watch dog’s trust in the member node \( n_i \) and \( R_{n_i}(sn) \) is the reputation opinion about the node ‘SN’ provided by \( n_i \).
After aggregation the result of the watch dogs are compared and decision is made through voting. If the aggregated trust values on the super node are below the threshold value, the watch dog will raise alarm to the network administrator to change the super node.

**Summary**

This chapter introduced the new approach used for detection and isolation of malicious node. The new approach integrates Bayesian network to model trust, weighted average to aggregate reputation, Chernoff Bound theorem to calculate personalized trust and watch dog to monitor the reputation of Super Node in the community. Bayesian network is used to model trust along with reinforcement learning. The peer’s needs are different in different situations and the trust is multi-faceted. Bayesian network provide flexible way to model trust. How trust is modelled during direct, indirect, inter and intra community interaction and the mathematical formulae used to build the approach are well explained in this chapter with examples.
CHAPTER 4
SIMULATION

4.1 Intra-community interactions

4.1.1 Detection and isolation of malicious nodes in a community

A community with a super node and 29 member nodes were simulated; one member node is made malicious to verify the behaviour of the system. The simulation was run for 20 days, with malicious node having 10 random transactions per day. The initial parameters used for the simulation is shown in table 3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Trust of Super node on member nodes , $T_{SN}(N_i)$</td>
<td>0.5</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Initial trust of member node on super node, $T_{N_i}(SN)$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Threshold value of a member node to accept the service, $T_{\text{threshold}}(N_i)$</td>
<td>0.3</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>$\alpha_s$</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>$\alpha_c$</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>$\alpha_f$</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$N_{\text{min}}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3: Initial parameter used for simulation
Figure 5: Number of bad transaction in a community with one malicious node

Figure 6: Super node’s trust on the malicious node in the community

Figure 5 shows the number of bad transaction in the network. The graph represents the bad transaction at the beginning of the simulation due to the malicious node. After few days of simulation, the malicious node was isolated in the network. Figure 6 shows the average trust value of other nodes on the malicious node, which is decreasing after every transaction.

Simulation was repeated using the same initial parameters, but the number of nodes in the community was increased to 100 with 40 malicious nodes. The number of interaction for every member node was set to 10 interactions per day and the super node with 20 interactions per day. The simulation was run for 50 days and the result is as follows
Figure 7: Number of bad transaction in a community with 40 malicious member nodes

Figure 8: Super node’s trust on malicious nodes in a community with 40 malicious member nodes

Figure 9: Member node’s trust on malicious nodes in a community with 40 malicious member nodes
Figure 7 shows that even though there are some malicious activity in the network in the beginning due to the 40 malicious nodes, as time passes the network successfully isolates the malicious node with the help of super node. As a result the number of bad transactions gets reduced. Figure 8 and 9 represents super node’s and member node’s average trust on malicious nodes respectively, both the values decreases over time.

The same simulation was repeated by increasing the number of malicious nodes from 40 to 80 and the results are as follows

![Figure 10: Number of bad transaction in a community with 80 malicious member nodes](image1)

![Figure 11: Member nodes trust on malicious member node in a community with 80 malicious member nodes](image2)
The result shows that even if the majority of the nodes in the community are malicious, the network is still capable of isolating them. Figure 10 shows the number of bad transactions over a day, Figure 11 and 12 shows the member node’s and super node’s average trust on the malicious nodes respectively. The simulation result assures the detection and isolation of malicious member nodes in a community.

4.1.2 Detection and isolation of both malicious super node and member nodes in a community

To study the behaviour of malicious super node in the network, simulation was performed with the initial parameter in the table 4 with malicious super node and all the member nodes being trust worthy.
<table>
<thead>
<tr>
<th>$\alpha_s$</th>
<th>0.8</th>
<th>0.8</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_r$</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>$\alpha_f$</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$N_{min}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Minimum threshold trust value a super node need to maintain, $T_{threshold}$(SN) | 0.25 | 0.5 | 0.25 |

Threshold trust value of a member node on super node, below which watchdog will be triggered, $T_{threshold}$(Watch dog) | 0.15 | 0.3 | 0.15 |

Table 4 : Initial parameter for simulation of malicious super node in a community

The simulation results are as follows

Figure 13 : Number of bad transaction in a community with malicious super node
Figure 13 shows the number of bad transaction in the network, figure 14 shows the average of trust value of the super node in the network and figure 15 shows the number of times watch dog was triggered during the simulation. Initially there were bad transactions in the network due to the malicious super node. Because of the malicious activity, super node losses reputation in the community. Watch dog was triggered whenever a member node become suspicious about the super node and once the overall reputation of the super node in the community falls below the threshold value watch dog will call for re-election and the malicious super node is replaced by a trust worthy super
node. As a result the performance of the network is increased and the number of malicious transactions in the network reduces as shown in Figure 13.

The same parameters were simulated with a malicious super node along with 50 malicious member nodes and the result is given below.

![Number of bad transaction in a community with malicious super node and 50 malicious member nodes](image1.png)

*Figure 16: Number of bad transaction in a community with malicious super node and 50 malicious member nodes*

![Trust of the super node in the community with malicious super node and 50 malicious member nodes](image2.png)

*Figure 17: Trust of the super node in the community with malicious super node and 50 malicious member nodes*
Figure 18: Number of times watchdog was triggered in the community with malicious super node and 50 malicious member nodes

Figure 16 shows the number of bad transaction in the network, figure 17 shows the average trust value of the super node in the network and figure 18 shows the number of times watchdog was triggered during the simulation. The result shows that even if half of the nodes in the community are malicious, the network is still able to isolate the malicious member nodes. Malicious super node was detected and replaced with new super node.

4.2 Inter-community interactions

4.2.1 Interaction between 2 communities in a two layered architecture

A community (C1) with a trust worthy super node (SN1) and 100 trust-worthy member nodes along with another community (C2) with a malicious super node (SN2), 20 trust-worthy member nodes and 80 malicious member nodes was created in the simulation environment. Both the community forms a two layered architecture. The simulation was run for 50 days with each community having 10 inter-community transactions. The initial setting is shown in table 5
<table>
<thead>
<tr>
<th>Parameter</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial trust of Super node on member nodes, $T_{SN}(N_i)$</td>
<td>0.5</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Initial trust of member node on super node, $T_{Ni}(SN)$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Threshold trust value of a member node to accept the service, $T_{\text{threshold}}(N_i)$</td>
<td>0.4</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>$\alpha_s$</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>$\alpha_r$</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>$\alpha_f$</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$N_{\text{min}}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Minimum threshold trust value a super node need to maintain, $T_{\text{threshold}}(SN)$</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>Threshold trust value of a member node on super node, below which watchdog will be triggered, $T_{\text{threshold}}(\text{Watch dog})$</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Default trust between super nodes</td>
<td>0.95</td>
<td>0.6</td>
<td>0.8</td>
</tr>
</tbody>
</table>

*Table 5: Initial parameter for inter-community simulation*

The simulation results are as follows
Figure 19: Trust of SN1 on SN2

Figure 20: Trust of SN2 on SN1

Figure 21: Number of interaction from C2 to C1
Figure 19 shows the trust of SN1 on SN2 over time, since SN2 is malicious the trust of SN1 on SN2 decreases over time. Figure 20 shows the SN1 trust on SN2, since C2 is a good community, the trust of SN1 on SN2 keeps increasing. Figure 21 represents the number of interaction from C2 to C1, since C1 provides bad service; C2 will isolate C1 after few days. In a community where super node and majority of the nodes are malicious that community will be isolated from the network as shown by simulation result above. The minority good nodes in the community will not get enough service from the super node or other community member nodes, so they will leave the community and join another community.

The same stimulation was re-run with reducing the number of malicious nodes in the C1 from 80 to 40, super node was kept malicious and the results are as follows

![Figure 22: Trust of SN1 on SN2 in a 2 layered network](image)
Figure 22 show the trust of SN1 on SN2 during the simulation. As SN1 is malicious it will decrease the trust value on SN2 even though SN2 is providing good service. After few days SN1 will be replaced by a new super node and from there onwards trust of SN1 on SN2 will start increasing as shown in the figure 22. Figure 23 shows the trust of SN2 on SN1. As SN1 is malicious it will be providing bad service to C2 which will result in low trust value of SN2 on SN1 initially. After few days SN1 loses its reputation in the community because of its malicious behaviour and will be replaced by a new super node. The new super node will provide good recommendation and service to community C2, so trust of SN2 on SN1 will increase.
4.2.2 Interaction between 3 communities in a two layered architecture

A two layered architecture with three communities C1, C2 and C3 was created in the simulation environment. C2 contains malicious super node (SN2), 80 malicious member nodes and 20 good nodes where as C1 and C3 contain trust worthy super node (SN1 and SN3 respectively) and 100 good member nodes. The initial parameter used is given in table 5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial trust of Super node on member nodes</td>
<td>0.5</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>, $T_{SN}(N_i)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial trust of member node on super node, $T_{Ni}(SN)$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Threshold trust value of a member node to accept the service, $T_{threshold}(N_i)$</td>
<td>0.4</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>$\alpha_s$</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>$\alpha_r$</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>$\alpha_f$</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$N_{min}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Minimum threshold trust value a super node need to maintain, $T_{threshold}(SN)$</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>Threshold trust value of a member node on super node, below which watchdog will be triggered, $T_{threshold}(Watch \ dog)$</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Default trust between super nodes</td>
<td>0.95</td>
<td>0.6</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 6: Initial setting used for the simulation of two layered architecture with 3 communities interacting each other
Figure 25 and 26 show the trust of SN1 on SN2 and SN3 respectively during simulation. C2 contains a bad super node and 80 malicious member nodes and is considered as a bad community. So SN1’s trust on SN2 is decreasing over time as shown in figure 25 and since C3 is a good community providing good service SN1’s trust on SN3 is increasing.
Figure 27: Trust of SN2 on SN1 during the simulation of the two layered architecture with 3 communities interacting each other.

Figure 28: Trust of SN2 on SN3 during the simulation of the two layered architecture with 3 communities interacting each other.

Figure 27 and 28 shows the trust of SN2 on SN1 and SN3 respectively. Since C1 and C3 are good communities, SN2’s trust on SN1 and SN3 is increasing.
Figure 29: Trust of SN2 on SN1 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 30: Trust of SN3 on SN2 during the simulation of the two layered architecture with 3 communities interacting each other

Figure 29 and 30 shows the trust of SN2 on SN1 and SN3 respectively. SN3’s trust on SN1 is expected to increases over time and where as SN3’s trust on SN2 should decreases over time. The expected result matches with the simulation results.
In the beginning of the network formation, the three communities will interact with each other. C2 will be isolated by C1 and C3 because of the malicious behaviour of C2. After few days C1 and C3 will not accept any transaction from C2 because of its low reputation as shown in figure 31.

4.3 Comparison of performance of a two layered architecture with and without watch dog

To compare the performance of the network with and without watch dog simulation where carried out. Two communities (C1 and C2) where formed with each having a super node (SN1 and SN2) and 100 member nodes. Every day each community will have 75 transactions each between them. C2 contains malicious super node and 40 malicious member nodes and the simulation is run for 50 days with initial parameter shown in table 7

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial trust of Super node on member nodes, $T_{SN}$ ($N_i$)</td>
<td>0.5</td>
</tr>
<tr>
<td>Initial trust of member node on super node, $T_{Ni}(SN)$</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Threshold trust value of a member node to accept the service, $T_{\text{threshold}}(N_i)$  & 0.3 \\
$\alpha_s$  & 0.8 \\
$\alpha_t$  & 0.9 \\
$\alpha_f$  & 0.95 \\
$N_{\text{min}}$  & 5 \\
Minimum threshold trust value a super node need to maintain, $T_{\text{threshold}}(SN)$  & 0.25 \\
Threshold trust value of a member node on super node, below which watchdog will be triggered, $T_{\text{threshold}}(\text{Watch dog})$  & 0.15 \\
Default trust between super nodes  & 0.95 \\

Table 7: Initial parameter for simulation of performance of a network with watch dog

Figure 32: Performance of network with and without watch dog

Figure 32 shows the number of transaction between C1 and C2. It is clear from the graph that the use of watch dog will increase the performance of the system and reduces the
system down time. The watch dog along with the member node detect and replace the malicious super node thereby increasing the performance of the overall network.

Summary

Simulation starts with study of intra-community interaction. One community consisting of a good super node and one malicious node was tested, followed by increasing the number of malicious member nodes and finally making the super node malicious. Performance as in number of bad transaction in the network was monitored along with the reputation of malicious node in the network. The simulation results confirm that the new approach is able to detect and isolate malicious node in the network.

Simulation was expanded to inter-community interaction. Two layered network with two communities were created in the simulation environment and interaction between the two communities under various test cases where examined. The evaluation of the approach using simulation shows that new approach is effective in inter-community level also. To further evaluate the approach, a simulation environment of two layered network with three communities with two communities being good and one community with 80 malicious member nodes, 20 good member nodes and a malicious super node was simulated. The result shows that the trust of two good communities on the malicious community decreases and the interaction from the malicious community decreases over time.

The performance of a two layered network with and without watch dog was compared and the result shows that use of watch dog to monitor super node adds to the performance in terms of number of interactions in the network.
CHAPTER 5
CONCLUSIONS AND FUTURE WORKS

The prime objective of the project is to develop a collaborative security approach for two layered networks. A novel approach was developed to protect the two layered network from attacks caused by malicious nodes. The new approach utilises the information from various nodes to prevent the intrusion along with a watch dog. Simulations were carried out to verify the newly developed approach. The unique feature of the new approach is the monitoring of super node in the community by the member nodes. When a super node turns malicious, it will be detected and warning is generated to the network administrator with the help of watch dogs in the network thereby increasing the performance of the network. All the existing trust and reputation mechanisms used in the two layered architecture presumes the super node to be trustworthy or impeccable which is not true in real network. The new approach can be applied to other applications, like e-commerce, recommender systems, web services, virtual organisation[37], cloud computing[38] and P2P distributed computing. The proposed approach has the advantage of effectively detecting malicious node and maintaining the computational load low. Major computation in the network uses the power and resource of the super node, thereby keeping the resources of member nodes intact.

5.1 Limitations

One of the major limitations of the new approach is that, it requires certain interaction before detecting the malicious node. Computational overhead for aggregation of reputation and trust can also be considered as a drawback, but the approach tried to utilize the resource and extra power of the super node to reduce the computational load on the member node. The current watch dog selection is on a random basis, a more robust watchdog selection mechanism will further increase the security of the network.
5.2 Future Works

A practical reward mechanism can be introduced to the approach which will encourage super node and other nodes to provide quality service, recommendation and feedback. Super node election mechanism can be combined with this new approach, which will increase the throughput of the network. Instead of random selection of watch dogs, other approaches can be explored. Primitive security techniques like; path and history based trust, co-signing, probation period for newly joined node in the community etc. can be integrated to refine the approach. Considering contextual information for recommendation and feedback will enhance the approach by having more comprehensive understanding of the nodes in the community, but it will increase network overhead due to the storage of additional information.
Reference


