Secure Data Transmission through Trusted Node in Mantes using AODV Routing Algorithm: SATEM

Dr. Indumathi .J[1], Anish A[2]

Department of Information Science and Technology,
College of Engineering, Anna University, Chennai, Tamilnadu, India

Abstract: To ensure fair and secure communication in Mobile Ad hoc Networks (MANETs), the applications running in these networks must be regulated by proper communication policies. However, enforcing policies in MANETs is challenging because they lack the infrastructure and trusted entities encountered in traditional distributed systems. This paper presents the design and implementation of a *policy enforcing mechanism based on trusted execution monitor built on top of the Trusted Platform Module*. Under this mechanism, each application or protocol has an associated policy. Two instances of an application running on different nodes may engage in communication only if these nodes enforce the same set of policies for both the application and the underlying protocols used by the application. In this way, nodes can form trusted application centric networks. Before allowing a node to join such a network, SATEM (Service-aware Trusted Execution Monitor) verifies its trustworthiness of enforcing the required set of policies. If any of them is compromised, SATEM disconnects the node from the network and SHA (Secure Hash Algorithm) algorithm for secure transmission. We demonstrate the fastidiousness of our solution through security analysis, and its low overhead through performance evaluation of the applications.

Keywords: Wireless Ad hoc and Sensor Networks, Mobile Ad hoc Networks, Service-aware Trusted Execution Monitor, Self-configuring, Survivable Adaptive Radio Networks.

I. Introduction

Wireless Ad hoc and Sensor Networks (WASN) are becoming an important platform in several domains, including military warfare and command and control of civilian critical infrastructure. They are especially attractive in scenarios where it is infeasible or expensive to deploy significant networking infrastructure. Examples in the military domain include monitoring of friendly and enemy forces, equipment and ammunition monitoring, targeting, and nuclear, biological, and chemical attack detection [C K Toh, (2007)]. Consider a military network scenario where more powerful and less energy-constrained ad hoc nodes may be carried by soldiers or in vehicles, while a large number of low cost and low-energy sensor nodes with limited energy resources may be distributed over the battlefield [Issa Khalil, et. al, (2010)]. This network setup can guide a troop of soldiers to move through the battlefield by detecting and locating enemy tanks and troops. The soldiers can use information collected by the sensor nodes to strategically position to minimize any possible causality. Examples in the civilian domain include habitat monitoring, animal tracking, forest fire detection, disaster relief and rescue, oil industry management, and traffic control and monitoring [Issa Khalil, et. al, (2010)].

MANET is a special class of ad hoc network. The concept of ad-hoc networks which was founded in early 70's and it has three generations [Ali Bazghandi, et. al, (2011)].

In *First generation* they were called *PRNET (Packet Radio Networks)*. In conjunction with ALOHA (Areal Locations of Hazardous Atmospheres) and CSMA (Carrier Sense Medium Access), approaches for medium access control and a kind of distance-vector routing PRNET were used on a trial basis to provide different networking capabilities in a combat environment.

The second generation of ad-hoc networks emerged in 1980s, when the ad-hoc network systems were further enhanced and implemented as a part of the *SURAN (Survivable Adaptive Radio Networks)* program. This provided a packet-switched network to the mobile battlefield in an environment without infrastructure. This program proved to be beneficial in improving the radios' performance by making them smaller, cheaper, and resilient to electronic attacks.

In the 1990s, *the third generation* of commercial ad-hoc networks arrived with notebook computers and other viable communications equipment. At the same time, the idea of a collection of mobile nodes was proposed at several research conferences.

The IEEE 802.11 subcommittee had adopted the term "ad-hoc networks" and the research community had started to look into the possibility of deploying ad-hoc networks in other areas of application.

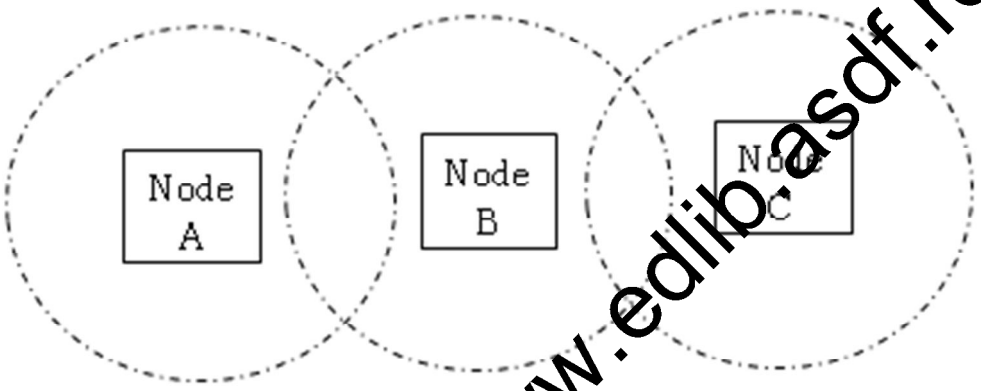


Figure 1 Mobile Ad Hoc Network

A mobile ad-hoc network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless links shown in Figure 1. Ad hoc is a Latin word and it means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The *primary challenge* in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet [L. Zhou, et. al, (1999)].

MANET's are a kind of wireless ad hoc networks that usually has a routable networking environment on top of a Link Layer ad hoc network. The growth of usage of laptops and 802.11/Wi-Fi wireless networking has made MANETs a popular research topic since the mid-1990s. Many academic papers evaluate protocols and their abilities, assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other [Kannan Govindan, et. al, (2011)]. Different protocols are then evaluated based on measure such as the packet drop rate, the overhead introduced by the routing protocol, end-to-end packet delays, network throughput etc.,[Kannan Govindan, et. al, (2011)]. So, finding appropriate routing protocol for mobile ad hoc network is a challenging task and also the routing protocol with less packet drop is a tedious task for the researchers because MANETs is high mobility.

This paper presents the design and implementation of a policy mechanism SATEM for secure data transmission in mobile ad hoc network with reduced packet loss, SATAM works based on a kernel-level trusted execution monitor. Under this mechanism, each MANET node follow the policy mechanism then only the node will be added in the network otherwise the node will be discarded. Since an application may depend on other applications, our policy enforcing mechanism creates a trusted node. The AODV routing algorithm is used to perform in secure routing in mobility nodes and the data security handled by SHA algorithm.

2. Trusted Node using SATEM

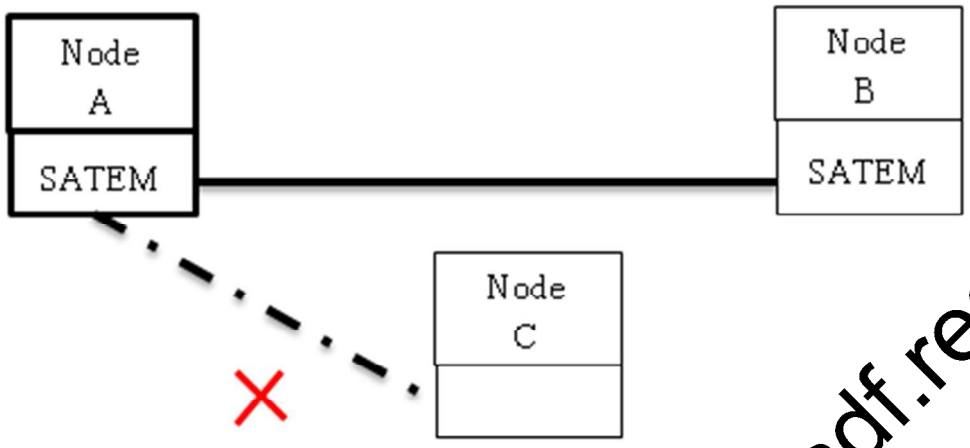


Figure 2 Connecting Trusted Node

Identifying trusted node is a type of security management system for computers and networks. SATEM gathers and analyzes information from various areas within a computer, or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). SATEM uses vulnerability assessment (sometimes referred to as scanning), which is a technology developed to assess the security of a computer system or network.

Unselfish Sharing: Each node simultaneously posts queries, answers queries, receives responses, and forwards queries for others. To benefit all nodes in the network, it is vital to ensure that enough nodes respond to and relay the queries posted by others. Similar concerns exist in other applications such as a P2P file sharing network, where sufficient file providers are desired. To achieve these goals, each node must abide by a policy, like the following before joining the network. In SATEM, the first step is to establish the trusted computing base that includes the trusted agent and the entire OS kernel. This process involves a trusted boot, in which each component in the boot sequence, starting from the TPM, measures the integrity of the next one before handing over the control.

Request to join the network: Node A sends a join request to Node B by specifying the application identity (e.g., the IP address and port number) and receives a request for a guarantee of trusted enforcement of the tier policy.

Deliver the Acknowledgement: If node B is in the communication range then the Node B send the Acknowledgement to the Node A.

Evaluate the policy: Once Node A received the acknowledgement first authenticates and verifies the integrity of the commitments and attestation. Then, it verifies the system commitment, the enforcement, and the boot attestation in the SATEM report against the local trust policy before accepting Node B to the tier. From the boot attestation, the member node learns that the requesting node has been booted into a trusted SATEM kernel. Knowing the system commitment convinces the member node that the kernel of the requesting node will not load untrusted modules, which protects the trusted agent from being tampered with. Knowing the enforcement convinces it that the enforcer software execution stack on the requesting node is trusted because the trusted agent will enforce the commitment to prevent untrusted code from being loaded by the enforcer.

Grant permission to join: The SATEM finds if node B is a trusted node then node A send the request to join the network and if node B accepts then the communication will take place.

3. Secure Routing using AODV

The reactive routing protocol which eliminates broad storm problem is Ad-Hoc On- Demand Distance Vector (AODV) routing protocol which builds on the DSDV algorithm. The AODV is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors (C K Toh) of AODV classify it as a pure on-demand route acquisition system, as nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges [C K Toh, (2007)].

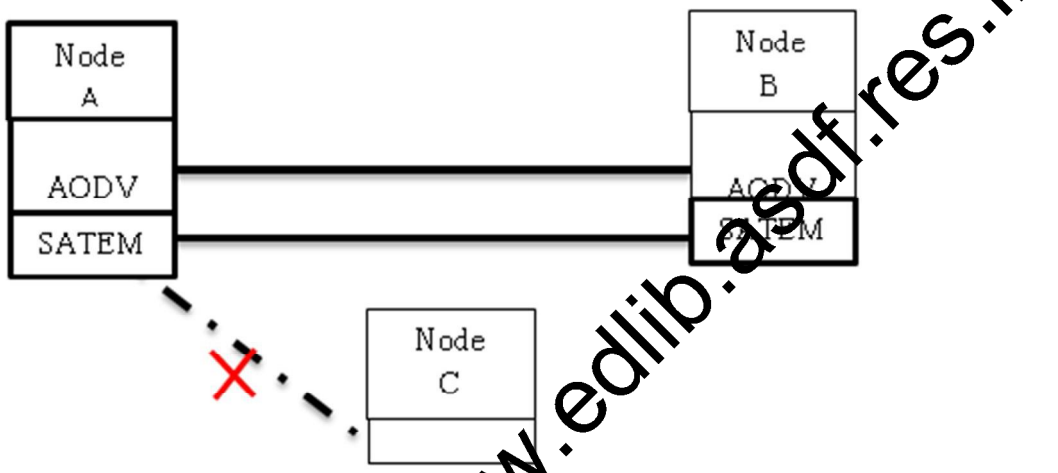


Figure 3 Establishing Trusted Routing Path

The AODV Routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and Dynamic Source Routing (DSR) stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the Route Request packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single Route Request.

The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the DestSeqNum of the current packet received is greater or equal than the last DestSeqNum stored at the node with smaller hop count. A Route Request carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcSeqNum), the destination sequence number (DestSeqNum), the broadcast identifier (BcastID), and the time to live (TTL) field. DestSeqNum indicates the freshness of the route that is accepted by the source. When an intermediate node receives a Route Request, it either forwards it or prepares a Route Reply if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the Route Request packet.

If a Route Request is received multiple times, which is indicated by the BcastID-SrcID pair, the duplicate copies are discarded. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send Route Reply packets to the source. Every intermediate node, while forwarding a Route Request, enters the previous node address and its BcastID. A timer is used to delete this entry in case a Route Reply is not received before the timer expires. This helps in storing an active path at the

intermediate node as AODV does not employ source routing of data packets. When a node receives a Route Reply packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop toward the destination.

The AODV Protocol eliminates broad storm problem using weighted persistence scheme. The Packets are rebroadcasted with the probabilistic approach. The number of rebroadcasts are reduced therefore broad cast success rate is increased.

4. Secure Data Transmitting using SHA

MANET has no centralized infrastructure or administrator, so key management is a challenging task in MANET. Key management includes key generation, key distribution and key maintenance. Key management protocol can be divided into two categories Private Key Management and Public Key Management. Private Key management protocol establishes private key or secret key that is used in symmetric-key cryptography. The public key management protocol provides a pair of keys (private/public) used for asymmetric key cryptography.

Symmetric-key cryptography is more efficient than asymmetric key cryptography however it needs a shared secret key between two communicating nodes. We need to set up n. (n-1)/2 shared secret keys if n is the size of network. Every node must have a mechanism to securely store the shared secret for each other nodes in the network. Since nodes in the ad-hoc network are resource constrained, key setup is an expensive operation. A variety of mechanisms can be used to set up shared secret key between two nodes. For example, shared secret keys can be preloaded between all the interested parties before the start of communication possibly through physical contact.

A trusted third party also known as key-distribution center (KDC) can be used. Key distribution center first shares a secret key with each node and then sets up secret key between two parties. If public key infrastructure (PKI) is present, the key can be encrypted with each participant's public key and transported to them. The two communicating party can create a secret key between themselves using symmetric key agreement schemes. The most common popular key agreement schemes use SHA.SHA1 outputs a 160bit digest of any sized file or input. In construction it is similar to the previous MD4 and MD5 hash functions, in fact sharing some of the initial hash values. It uses a 512 bit block size and has a maximum message size of 2^{64}-1bits.

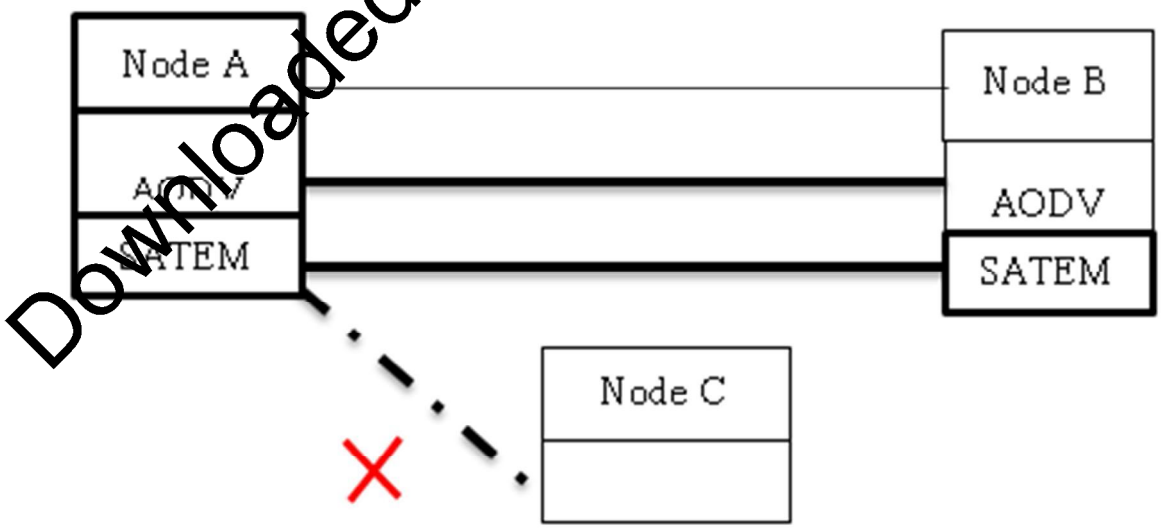


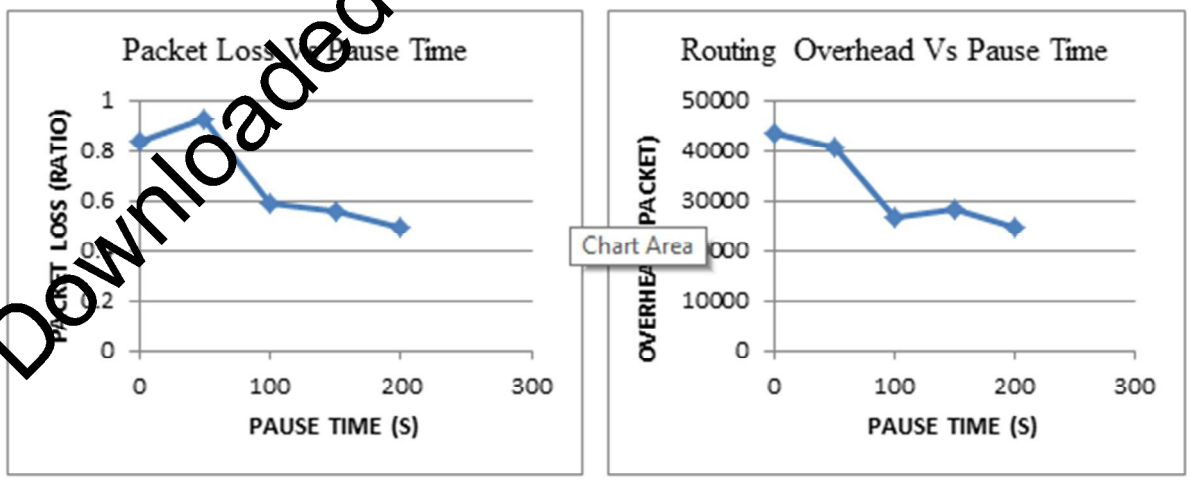
Figure 4 Secure Data Transmission Via Trusted Node

SHA1 Algorithm

- Padding
  - Pad the message with a single one followed by zeroes until the final block has 448 bits.
  - Append the size of the original message as an unsigned 64 bit integer.
- Initialize the 5 hash blocks (h0,h1,h2,h3,h4) to the specific constants defined in the SHA1 Standard.
- Hash (for each 512bit Block)
  - Allocate an 80 word array for the message schedule
    - set the first 16 words to be the 512bit block split into 16 words.
    - the rest of the words are generated using the following algorithm word [i3] XOR word [i8] XOR word [i14] XOR word [i16] then rotated 1 bit to the left.
  - Loop 80 times doing the following.
    - Calculate SHA function () and the constant K (these are based on the current count Number.
    - e=d
    - d=c
    - c=b (rotated left 30)
    - b=a
    - a = a (rotated left 5) + SHA function () + e + k + word[i]
  - Add a, b, c, d and e to the hash output.
- Output the concatenation (h0, h1, h2, h3, h4) which is the message digest.

5. Simulation Evaluation

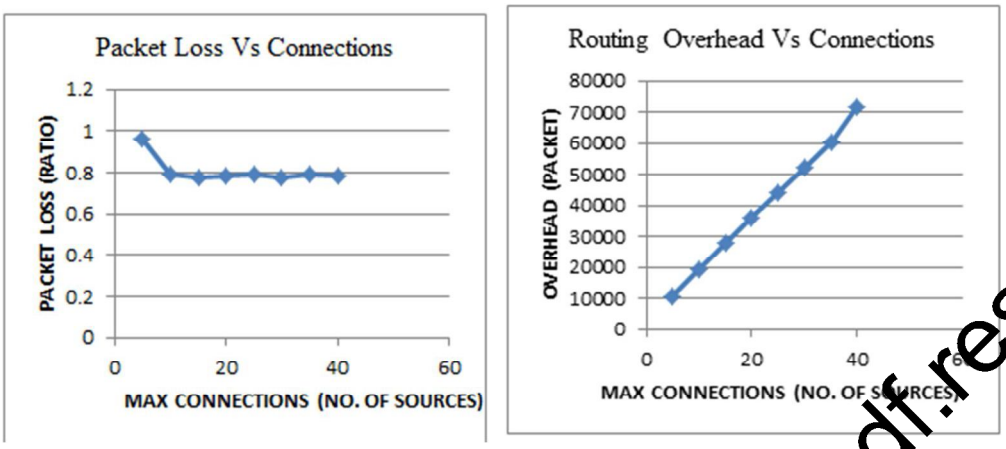
We conduct a series of experiments to evaluate the performance and packet loss during the data transmission. This can be done using NS-2 simulation and using three parameter to evaluate pause time, number of connections and number of nodes. Each parameter have constants they are, for pause time number of node (50), speed(25m/s), maximum connection (25), for number of connection number of nodes(50), speed(25m/s), pause time(0) and for the number of nodes pause time(0s), maximum speed (10m/s), maximum connections (10). Then to create the MANET environment, we use the following constants as simulation area (1000*800), CBR traffic type is used for traffic, packet size (512bytes), packet rates (4 packet/s), and maximum connection (25).



5 (a) 5 (b)

Figure 5 Varying Pause Time (0, 100, 200, 300, 400), (a) Packet Loss Vs Pause time, (b) Routing Overhead Vs Pause Time

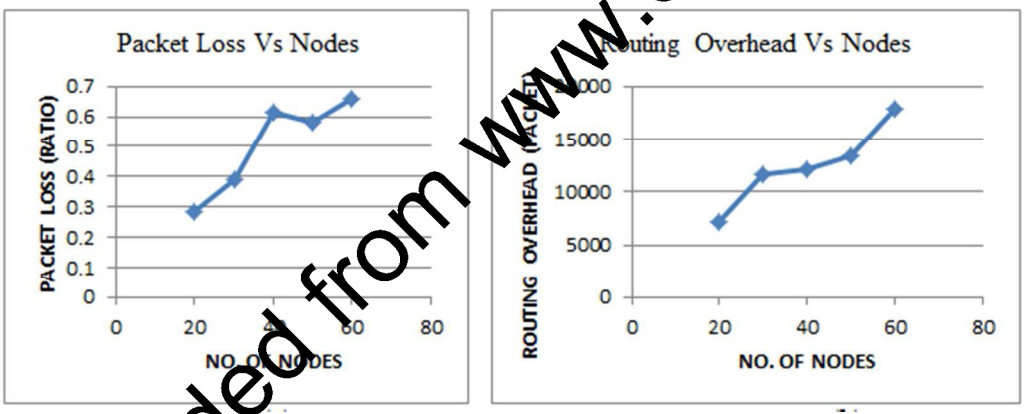
By varying the pause time, the packet loss and the routing overhead will reduced shown in Figure 5.1.



6 (a) 6 (b)

Figure 6. Varying Maximum Connections (5-40), (a) Packet Loss Vs Max Connections, (b) Routing Overhead Vs Max Connections

By varying the maximum connections, the packet loss will be reduced but the routing overhead will gradually increase shown in Figure 5.2



7 (a) 7 (b)

Figure 7 Varying No Of Nodes (5-40), (a) Packet Loss Vs Number of Nodes (b) Routing Overhead Vs Number of Nodes

By varying the number of nodes in the network, the packet loss and routing overhead will be gradually increase shown in Figure 5.3.

According to the simulation result, we have a better routing overhead and reduced packet loss based on three different parameter.

6. Result and Discussion

In the simulation, we used the NS-2 simulator to evaluate how the overhead and reduce the packet loss in creating the trusted multi-tier network and enforcing the policies varies in complex MANETs with different mobility scenarios. Here, SATEM only ensures that a protected service cannot load untrusted code from the

disk. Here the main problem is unable to tackle attacks, like buffer overflow, that can cause the protected service to run arbitrary code without changing its disk image.
SATEM only mitigates the problem in two aspects. First, SATEM may reveal the code that has known buffer overflow vulnerabilities by attesting it to the user. Hence, the user can avoid trusting the vulnerable code. Second, in the case of a successful buffer overflow attack, the attacker runs her own code on the service stack without being caught by SATEM.

7. Conclusion

This paper presented a policy enforcement mechanism based on SATEM for MANETs to enforce secure communication in mobile ad hoc network. Under this mechanism, each MANET application has its own policy. All nodes support and enforcing its policy form a trusted application centric network. Only trusted nodes are allowed to join the network. Moreover, communication between them is regulated by the policies each tier. To ensure trusted policy enforcement, we augment each node with a trusted kernel agent. The nodes are communicated based on the path selected by the secure routing protocol, and data SHA secured the data. We evaluated the method through a prototype based on an IEEE 802.11 ad hoc network and through network simulations. The results demonstrate the feasibility of the proposed method (secure data transmission using SATEM) as well as its low overhead.

References

1. Bagga, W., Crosta, S., Michiardi, P., & Molva, R. (2007). Establishment of ad-hoc communities through policy-based cryptography. *Electronic Notes in Theoretical Computer Science*, *171*(1), 107-120.
2. Bazghandi, A. (2011). Ad Hoc Protocols Via Multi Agent Based Tools. *arXiv preprint arXiv:1110.5173, Vol. 8, Issue 4, No 2.*
3. Capkun, S., Buttyán, L., & Hubaux, J. P. (2003). Self-organized public-key management for mobile ad hoc networks. *Mobile Computing, IEEE Transactions on*, *2*(1), 52-64.
4. Govindan, K., & Mohapatra, P. (2012). Trust computations and trust dynamics in mobile Adhoc networks: a survey. *Communications Surveys & Tutorials, IEEE*, *14*(2), 279-298.
5. Kauer, B. (2007, August). OSLO: Improving the Security of Trusted Computing. In *USENIX Security*.
6. Khalil, I., & Bagchi, S. (2011). Stealthy attacks in wireless ad hoc networks: detection and countermeasure. *Mobile Computing, IEEE Transactions on*,*10*(8), 1096-1112.
7. McCune, J. M., Jaeger, T., Berger, S., Caceres, R., & Sailer, R. (2006, December). Shamon: A system for distributed mandatory access control. In *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual* (pp. 23-32). IEEE.
8. Shi, E., Perrig, A., & Van Doorn, L. (2005, May). Bind: A fine-grained attestation service for secure distributed systems. In *Security and Privacy, 2005 IEEE Symposium on* (pp. 154-168). IEEE.
9. Toh, C. K. (2001). *Ad hoc mobile wireless networks: protocols and systems*. Pearson Education.
10. Xu, G., Borcea, C., & Iftode, L. (2006, October). Satem: A Service-aware Attestation Method Toward Trusted Service Transaction. In *the Proceedings of IEEE Symposium on Reliable Distributed Systems (SRDS)* (pp. 321-336).
11. Xu, G., Borcea, C., & Iftode, L. (2007, October). Trusted application-centric ad-hoc networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on* (pp. 1-10). IEEE.
12. Xu, G., Borcea, C., & Iftode, L. (2011). A policy enforcing mechanism for trusted ad hoc networks. *Dependable and Secure Computing, IEEE Transactions on*,*8*(3), 321-336.
13. Zhou, L., & Haas, Z. J. (1999). Securing ad hoc networks. *Network, IEEE*,*13*(6), 24-30.