International Journal Of Computer Science And Applications Vol. 1, No. 3, December 2008
ISSN 0974-1003

Security Enhancement For ARAN (Authenticated Routing For Ad Hoc Networks)

Suresh Kumar Mandilli
Dept of Computer science, Birla Institute of Technology Mesra, Ranchi
sureshmandilli@gmail.com

V. S. Shankar Sriram
Dept of Computer science, Birla Institute of Technology Mesra, Ranchi
sriram@bitmesra.ac.in

G. Sahoo
Dept of Computer science, Birla Institute of Technology Mesra, Ranchi
drgsahoo@yahoo.com

ABSTRACT
Mobile Ad Hoc Network (MANET) is a collection of communication devices or nodes that wish to communicate without any fixed infrastructure and pre-determined organization of available links. The nodes in MANET themselves are responsible for dynamically discovering other nodes to communicate. Although the ongoing trend is to adopt ad hoc networks for commercial uses due to their certain unique properties, the main challenge is the vulnerability to security attacks. Due to the broad field of applications, a general security model can still not be found in any literature. All introduced protocols for ad hoc networks are based on different assumptions and security requirements, and are consequently suited for specific applications only. We surveyed different existing security threats from selfish nodes and their disturbance to mobile ad hoc networks. Also we have chosen Authenticated Routing for Ad hoc Networks (ARAN) secure routing protocol for analysis and identified ARAN is not capable of handling attacks from the selfish nodes. In this paper, we propose a trust-based scheme to integrate with the ARAN protocol for stimulating cooperation among selfish nodes in mobile ad hoc networks. Our enhancement scheme provides incentive for mobile nodes to cooperate and report actions honestly with out any requirement of hardware. Furthermore, we present a formal model for our proposed scheme.

Categories
Wireless Technologies, Network protocols, Management and Security

General Terms
Security, Performance

Keywords
Trusted Server, Credit Clearance Service (CCS) and Certificate Authority.

1 INTRODUCTION
Now-a-days, Mobile ad hoc network (MANET) is one of the recent active fields and has received marvelous attention because of their self-configuration and self-maintenance capabilities [1]. In general, the wireless *MANET* is particularly vulnerable due to its fundamental characteristics of open medium, dynamic topology, and absence of central authorities, distribution cooperation and constrained capability [2].

Although mobile ad hoc networks have several advantages over the traditional wired networks, they have a unique set of challenges. Firstly, MANETs face challenges in secure communication. Secondly, mobile nodes without adequate protection are easy to compromise. Thirdly, static configuration may not be adequate for the dynamically changing topology in terms of security solution. Finally, lack of cooperation and constrained capability which is common in wireless *MANET* makes anomalies that are hard to distinguish from normalcy.

2 PROBLEM STATEMENT
In this paper, we adopted ARAN protocol which introduces authentication, message integrity, and non-repudiation and defends almost against all security attacks in MANETs. However, it does not account for selfish nodes whether by detecting or isolating them from the network. So we decided to survey about the different types of cooperation enforcement schemes in mobile ad hoc networks to design and integrate a trust-based scheme with the ARAN routing protocol to make it capable of defending itself against both malicious and authenticated selfish nodes.

3 RELATED WORK
In this section, we present the overall architecture and the intuitions behind our design.

3.1 System architecture
Figure.1 shows the overall architecture of our system, which consists of Trusted Server, provides the Credit Clearance Service (CCS), also acts as a Certificate Authority (CA) and a collection of mobile nodes..

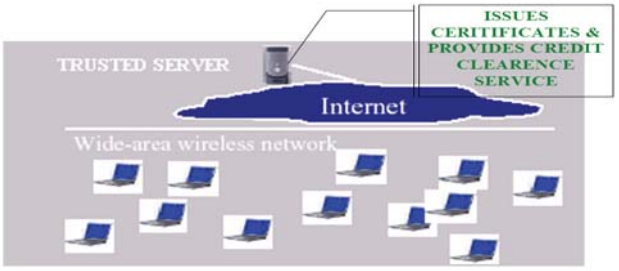

Figure 1: System Architecture

The nodes are equipped with network interfaces that allow them to send and receive messages through a *wireless overlay* network. To identify each node, the Trusted Server issues a certificate to each node as proposed in ARAN mechanism [6]. When a node sends its own messages, the node (or the destination, see later) will lose credit (or virtual money) to the network because other nodes incur a cost to forward the messages. On the other hand, when a node forwards others' messages, it should gain credit and therefore be able to send its messages later. However, the preferred and dominant way to get more credit is by forwarding others' messages. In order to get credit for forwarding others' messages, a node needs to report to the CCS of Trusted Server. In order to save bandwidth and storage, instead of requiring the whole message as a report, our system uses small *receipts*. Such receipts are derived from the content of the messages but do not expose the exact content of the messages. Thus, although we require that the CCS be trusted in terms of maintaining credit balance, the nodes do not need to trust the CCS in terms of message confidentiality.

3.2 Who pays whom?

There are two reasons for charging only the sender. First, charging the destination may allow other nodes to launch a denial-of-service attack on the destination by sending it a large amount of traffic. Even sharing the cost between the sender and the destination could have a similar problem, because the sender could collude with the intermediate nodes, who could secretly return the sender's payment back, so that only the destination pays for the traffic. On the other hand, if only the sender is charged, a node will not have incentive to send useless messages. Second, if the destination benefits from the content of a message and thus should pay for it, the sender can get compensation from the destination. Given these reasons, only the sender will be charged in our system.

A closely related question is who will receive credit for forwarding a message. Ideally, any node who has ever tried to forward a message should be compensated because forwarding a message will incur a cost to the node, no matter it is successful or not. However, a forwarded message may be corrupted on the link, and there is no way to verify that the forwarding action does occur. Given this decision, the credit that a node receives will depend on whether or not its forwarding action is successful — forwarding is successful if and only if the next node on the path receives the message.

3.3 Objectives of the payment scheme

The second basic question is about the objective of the payment scheme. After all, the objectives of our payment scheme are to prevent cheating actions and to provide incentive for the nodes to cooperate. Given such objectives, our system does not target *balanced payment*; that is, we do not require that the total charge to the sender be equal to the total credit received by other nodes for a message. In fact, in order to prevent one type of cheating actions, our CCS charges the sender more than it gives to the other nodes. In order to offset long-term net outflow of credit from the mobile nodes to the CCS, if it is a large network, the CCS periodically returns the credit back to the mobile nodes uniformly; otherwise, the CCS periodically gives each mobile node a fixed amount of credit. Note that this return will not enable any cheating action or reduce the incentive of the nodes to forward others' messages.

3.4 Cheating actions in the receipt-submission

Since the mobile nodes are selfish, without a proper payment scheme, they may not forward others' messages or they may try to cheat the system, if the cheating can maximize their welfare. In particular, a selfish node can exhibit one of the following three selfish actions:

1) After receiving a message, the node saves a receipt but does not forward the message.

2) The node has received a message but does not report the receipt.

3) The node does not receive a message but falsely claims that it has received the message.

Note that any of the selfish actions above can be further complicated by collusion of two or more nodes. We next progressively determine the requirements on our system in order to prevent the above actions.

3.5 Motivating nodes to forward messages

In order to motivate a selfish node to forward others' messages, the CCS should give more credit to a node who forwards a message than to a node who does not forward a message. A basic scheme to achieve this objective is as follows. Assuming R as the message receipt which is the part of the original message, α as the credit given to the node which intimates the message receipt (R) and forwards the original message to the next node ,β as the credit given to the node if it only intimates the message receipt (R) to the server. First, the CCS determines the last node on the path that has ever received the message. Then the CCS asks the sender to pay β to this node, and α to each of its predecessors, where $\beta < \alpha$. Note that the CCS does not ask the sender to pay anything to the successors of the last node. We illustrate the above by an example which is represented in figure.2 which shows the payment representation by the respective nodes. In this example, only the first three intermediate nodes submit their receipts. Therefore, nodes 1 and 2 will each receive a payment of α, and node 3 a payment of β. Since node 4 and the destination do not submit any receipt, they do not receive any credit. The sender pays a total of $2\alpha + \beta$ from its initial amount (X).

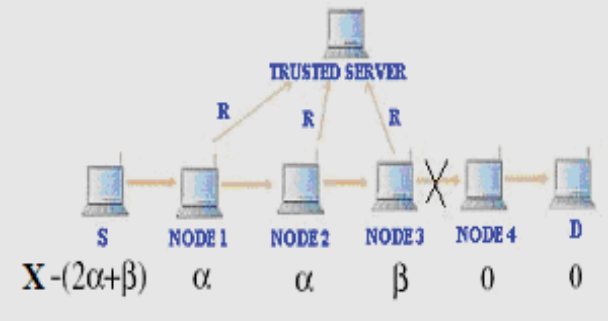

Figure 2: Illustration of our payment scheme

3.6 Motivating nodes to report their receipts

Obviously, each single node having received a message is motivated to report its receipt, if β is greater than its cost of submitting a receipt, which, as we discussed previously, should be low since a receipt is generally small. Unfortunately, there is still a collusion that can work against the above design. As an example, the last node (or in the general case, the last *k* nodes) ever received the message can collude with the sender. In particular, if the last node does not report its receipt, the sender saves α while the last node loses β. However, if the sender gives the last node a behind the scene compensation of β + ϵ, where ϵ > 0, the last node will be better-off while the sender still enjoys a net gain of $\alpha - (\beta + \epsilon)$. Thus, the colluding group gets a net benefit of about $\alpha - \beta$.

In order to prevent this cheating action, the CCS charges the sender an extra amount of credit if the destination does not report the receipt of a message. This extra charge goes to the CCS instead of any nodes. The overall charge to the sender (including payments to other nodes and the extra charge) should be $k\beta$ less than the charge to the sender when the message arrives at the destination, where *k* is the number of nodes not submitting receipts. Given such extra charge, even a colluding group cannot benefit from this cheating action. Again consider the example in Figure 2. Figure 3 shows the revised amount paid by the sender, which is equal to $(4\alpha + \beta)-2\beta$.

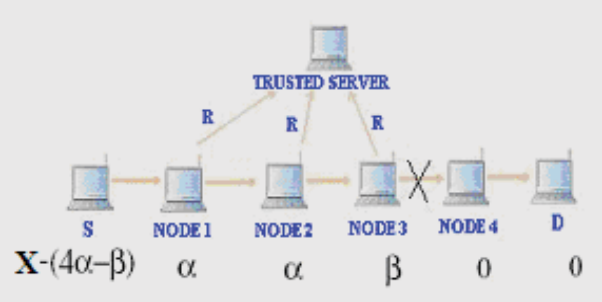

Figure 3: Illustration of our payment scheme

3.7 Preventing false receipts

Instead of forwarding the whole message, an intermediate node can forward only the receipt of a message to its successor, which is sufficient for getting credit. Moreover, the intermediate node can even wait until it has a fast connection to the successor to forward the false receipt, thus further saving resource usage. The key to prevent such attack depends on the destination. We distinguish two cases: 1) the destination colludes with the intermediate nodes. For this case, we argue that the intermediate nodes and the destination should be paid as if no cheating had happened, because after all, the message is for the destination and the destination does submit a receipt for the message, indicating that it has received the message. 2) the destination does not collude with the intermediate nodes. In this case, if the intermediate nodes forward only the receipt of a message instead of the whole message, then the destination will not be able to receive a valid message payload, and therefore will not submit a receipt for the message. Based on this observation, we can prevent the potential cheating action of the intermediate nodes by greatly reducing the amount of credit given to the intermediate nodes, if the message is not reported to be received by the destination. With such reduction of credit, the cheating nodes cannot get enough credit even to cover the minimum expense needed for this type of cheating, *i.e.*, the cost of forwarding a receipt. To be more exact, if the destination does not report a receipt of a message, we multiply the credit paid to each node by γ, where γ< 1. Figure shows the revised amount of credit received by each node. In particular, comparing Figure 4 with Figure 3, due to this revision, we reduce the charge to the sender by γβ instead of β, for each node on the path who does not report a receipt.

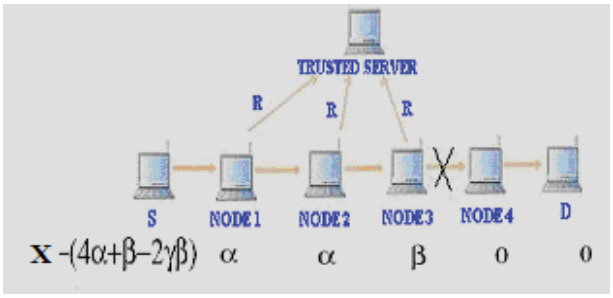

Figure 4: Illustration of our payment scheme

4 FORMAL MODEL

Consider a network that has d + 1 nodes, n0, n1,…, nd, from the sender to the destination. Let Mi be the information held by node ni that is unknown to the CCS.

4.1 Authenticated Route Discovery

The source trusts the destination to select the return path. The source node, A, begins route instantiation to destination D by broadcasting to its neighbors a route discovery packet (RDP):

A->brdcast: [RDP, IPD, NA] KA-,certA

4.2 Authenticated Route Setup

By receiving the RDP, the destination unicasts a *Route Reply (RREP)* packet back along the reverse path to the source. Let the first node that receives the RREP sent by D to be node C:

D->C: [RREP, IPA, NA] KD-,certD

4.3 Certification Process

Trusted Server (T) issues a certificate, whose public key is known to all valid nodes. Keys are pre-generated and exchanged through an existing out of band relationship between T and each node. Before joining the ad hoc network, each node must request a certificate from T. Each node receives exactly one certificate after securely authenticating their identity to T. So a node A receives a certificate from T as follows:

T ->A:certA = [IPA,KA+,t,e]KT-

The certificate contains the IP address of A, the public key of A (KA+), a timestamp t of when the certificate was created (t) and a time (e) at which the certificate expires. These variables are concatenated and signed by T (KT-).

4.4 Information

For i > 0, && if node ni has ever received message m Then

Mi = TRUE

Else

Mi = FALSE

Obviously, the sender n0 and the set of nodes that have ever received message m constitute a prefix of the path. Therefore,

Mi = TRUE if $0 < i \leq e$

FALSE if $e < i \leq d$,

Where e is the index of the last node that has ever received message m.

4.5 Actions

Each node, ni (i > 0), has two possible actions:

1. Reporting that it has ever received message m (by submitting a valid receipt), or
2. Withholding its report.

We denote the action of ni by Ai. Then Ai is either TRUE or FALSE. The only exception is n0, which has no choice of action. We define A0 = TRUE, for completeness of our model.

4.6 Cost of Actions

We denote the cost of ni's action by Ui. As discussed before, in general, the cost of sending a receipt to the CCS is very low. However, if node ni does not receive message m but can successfully claim that it has received the message, then a colluding node must have forwarded ni a copy of the receipt. Let δ be the cost of forwarding a receipt from one mobile node to another node. Then the colluding node incurs a cost of δ and ni must compensate the colluding node with δ. Counting this cost on ni, we have

If Mi = FALSE and Ai = TRUE

Ui = δ

Else

Ui =0

4.7 Computing payments

We assume that p = (n0, n1, . . . , ne, . . . , nd), where ne is the last node on path p that submits a valid receipt with sequence number seq. Then the CCS charges C from node n0, and pays Pi to node ni,

$C = ((d - 1)\,\alpha + \beta - (d - e)\,\gamma\beta)$.

where system's payment to ni (i > 0) is

Pi = α if $i < e = d$

β if $i = e = d$

γα if $i < e < d$

γβ if $i = e < d$.

For n0, the charge of C can be viewed as a negative payment

$P0 = -C = -((d - 1)\,\alpha + \beta - (d - e)\,\gamma\beta)$

When the CCS computes payment, a ROUTE REQUEST is rejected if any signature in the message is invalid. Furthermore, if a ROUTE REQUEST submitted by a node is a part of another ROUTE REQUEST submitted by the same node, then the former message is rejected.

5 CONCLUSION AND FUTURE WORK

We proposed Trusted-Server based scheme, built on top of ARAN secure routing protocol, to provide incentive to mobile nodes to cooperate. Thus, the proposed design proves to be more efficient and more secure than ARAN secure routing protocol in defending against both malicious and authenticated selfish nodes. Our system determines payments and we showed that our system motivates each node to report its behavior honestly, even when a collection of the selfish nodes collude. We would like to extend this model using mobile agents in place of Trusted Server as our future work.

6 REFERENCES

[1] H. Yang, H. Luo, F. Ye, S. Lu, L. Zhang, Security in mobile ad hoc networks: challenges andsolutions," In proc. IEE Wireless Communication, UCLA, Los Angeles, CA, USA; volume- 11, Page(s):38- 47, ISSN: 1536-1284

[2] W. Arbaugh, N. Shankar, and Y.C. Wan. Your 802.11 wireless network has no clothes. Technical report, Dept. of Computer Science, University of Maryland, March 2001.

[3] E. Royer and C. Toh. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications, April 1999, pages 46-55.

[4] M. Ilyas and R. Dorf. The handbook of ad hoc wireless networks. The Electrical Engineering Handbook Series archive. Publisher CRC Press, 2003.

[5] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking, September 2002, pages 12-23.

[6] K. Sanzgiri, B. Dahill, B. Levine, E. Royer and C. Shields. A Secure Routing Protocol for Ad hoc Networks. Proceedings of the tenth IEEE International Conference on Network Protocols, November 2002, pages 78-87.

[7] Y. Hu, D. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks. In Fourth IEEE Workshop on Mobile Computing Systems and Applications, June 2002, pages 3-13.

[8] P. Papadimitratos and Z. Haas. Secure data transmission in mobile ad hoc networks. Proceedings of WiSe, September 2003, pages 41-50

[9] P. Papadimitratos, Z. Haas and P. Samar. The Secure Routing Protocol (SRP) for Ad Hoc Networks. Internet-Draft, draft-papadimitratos-securerouting- protocol-00.txt, December 2002.

[10] M. Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In Proceedings of the ACM Workshop on Wireless Security, September 2002, pages 1-10.