Secure Routing in MANETs using Key Management

Rakesh Kumar ER

Asst. Prof. & Head
Computer Science and Engineering,
SAMS College of Engineering and Technology, Chennai, INDIA
rakeshkumarer@gmail.com

Abstract* - The Mobile ad hoc networks (MANETs) have attracted much attention due to their mobility and ease of deployment. The wireless and dynamic natures render them more vulnerable to various types of security attacks than the wired networks. The major challenge is to guarantee secure network services. To meet this challenge, certificate revocation is an important integral component to secure network communications. In this paper, we focus on the issue of certificate revocation to isolate attackers from further participating in network activities. General assumption is that nodes in possession of a valid secret key can be trusted. Consequently, a secure and efficient key-management scheme is crucial. Keys are also required for protection of application data. However, the focus here is on network-layer management information. Whereas key management schemes for the upper layers can assume an already running network service, schemes for the protection of the network layer cannot. Keys are a prerequisite to bootstrap a protected network service. For encrypting the messages we are going to implement the RC 4 Algorithm and for routing we will implement the AODV for forwarding the packets to the receiver side.

***Index Terms* – MANETS; RC4; AODV; Key Management; Security Schemes**

I. Introduction

Mobile ad hoc networks have wireless links and work independently of fixed infrastructure. They are self-organizing and self-configuring. The wireless nodes operate both as communication end-points as well as routers, enabling multi-hop wireless communication. The wireless devices imply limited power resources and bandwidth. Network topology may change rapidly due to mobility, interference, physical obstacles on the path, and so forth. Application areas range from conference hall networks to ad hoc networks for emergency and rescue operations and military tactical use.

The wireless and dynamic nature of ad hoc networks leave them more vulnerable to security attacks than their wired counterparts. Passive eavesdropping as well as active message insertions, denial of service, and battery-exhaustion attacks are inherently easy. Security attacks can be launched towards any layer of the protocol stack. Defense mechanisms for the lowest layers call for physical tamper protection and transmission security measures such as spread-spectrum techniques, frequency hopping, and interleaving. Cryptographic techniques are essential for the protection of the higher layers. In wired networks routers are part of an established and controllable infrastructure. The same is not true in ad hoc networks where the nodes act both as routers and communication end points. This makes the network layer more prone to security attacks. There is no guarantee that malicious nodes do not mingle and interfere.

Examples of possible attacks are misdirection and insertion of bogus routing information, black holes (nodes attracting traffic by maliciously advertising shortest path to other nodes), and wormholes (adversary nodes colluding by tunneling packets from one part of the network to another). A primary challenge is to decide which routing information can be trusted. A number of schemes relying on cryptographically signed routing messages have been designed most without detailing key management further. Nevertheless, the possession of cryptographic keys serves as proof of trustworthiness. Consequently, a proper key management service is required. This is to ensure that nodes which are legitimate members of the network and only those are equipped with the necessary keys whenever needed. Whereas key-management services are needed for application layer security as well as for protection of the network layer, this article focuses on the more challenging of the two, namely, providing keys for the network layer. Key management schemes for the application layer can assume an already running network service. Schemes for the network layer routing information cannot. Keys are a prerequisite to bootstrap a protected network service.

II. Related Works

Key management has two important aspects: key distribution, which describes how to disseminate secret information to the principals so that secure communications can be initiated, and key revocation, which describes how to remove secrets that may have been compromised. Key management in sensor networks face constraints of large scale, lack of a priori information about deployment topology, and limitations of sensor node hardware. While key distribution has been studied extensively in recent work, the problem of key and node revocation in sensor networks has received relatively little attention. Yet revocation protocols that function correctly in the presence of active adversaries pretending to be legitimate protocol participants via compromised sensor nodes are essential. In their absence, an adversary could take control of the sensor network’s operation by using compromised nodes which retain their network connectivity for extended periods of time.

In this paper, we present an overview of key distribution methods in sensor networks and their salient features to provide context for understanding key and node revocation. Then we define basic properties that distributed sensor node revocation protocols must satisfy, and present a protocol for distributed node revocation that satisfies these properties under general assumptions and a standard attacker model. As with all networks comprising geographically distributed nodes, communication security in sensor networks requires effective management of cryptographic keys. In contrast to traditional networks, key management in sensor networks is particularly complex due to the large numbers of sensor nodes, the lack of a priori information about the deployment topology of the network, the limited hard are capabilities of the nodes, and the constant exposure of nodes to capture by an active adversary who could obtain key material. Two important aspects of key management are key distribution and key revocation.

Key distribution refers to the task of distributing secret keys between sensor nodes to provide communication secrecy and authenticity. Key revocation refers to the task of securely removing keys that are known to be compromised. If the cryptographic primitives themselves do not expose the secret keys reasonable and common assumption then secret keys can only be exposed by compromising sensor nodes. The problem of sensor node revocation can thus be reduced to that of key revocation; i.e., by revoking all of the keys belonging to a known compromised sensor node, we can effectively remove the node’s presence in the network. In contrast to key distribution, which has been studied extensively in recent work, key revocation received relatively little attention; i.e., with the exception of the centralized revocation scheme proposed no other schemes have been reported to date. Yet, key revocation is as important as key distribution in sensor network key management. A sensor network is generally designed for deployment in open, unmonitored environments exposing nodes to physical attacks. This requires that, in the event of node capture by an adversary, the sensor network have the ability to revoke the cryptographic keys of captured nodes. Otherwise, the entire network’s operation may be compromised by an adversary that surreptitiously controls both the operation and communication of these nodes.

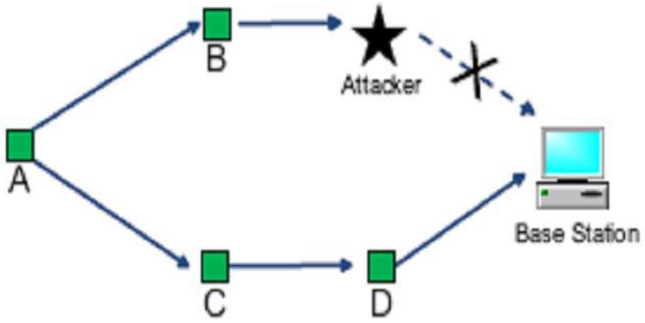
We first review in brief several known methods for key distribution in sensor networks. This forms the background for our main discussion of the problem of distributed key revocation. Distributed node revocation is useful due to its ability to eliminate compromised nodes without requiring a central authority that might become an attractive attack target. Thus, distributed revocation improves re- action time after node capture and overall system resilience. Distributed revocation protocols are more complex than centralized ones due to the fact that any of the nodes executing the protocol may be malicious and attempt to block or subvert the protocol. A distributed revocation protocol is correctly designed, specified, and formally verified in the absence of an active adversary, assurance of correct behavior would still be lacking. For example, captured nodes could circumvent or block protocol operation, or collude among themselves to execute the revocation protocol correctly against legitimate nodes to disconnect them from the network. So far, research in sensor net key management has been missing the following tools: (1) a rigorous specification of distributed-revocation properties that must hold in a sensor network even in the presence of an active adversary, (2) a precise definition of the adversary model, and (3) a distributed key revocation protocol that satisfies those properties in a general sensor-network setting.

The main contributions of this paper are a rigorous definition of distributed revocation properties for sensor networks, a general active adversary model, and a protocol for distributed key revocation that satisfies the specified properties under the defined adversary model. However, distributed key revocation cannot be defined independently of the specific key distribution scheme used in a particular sensor network. This is the case because some key distribution methods are more suitable for specific key revocation methods, while others may prevent key revocation altogether. A secondary contribution of this paper is a succinct overview of key pre-distribution methods and their salient features that affect key revocation and overall sensor-network operation and resiliency.

III. ENHANCED WORK

Attacks can be mounted by a single adversary or collaborative ones. We differentiate between node compromise and disruption attacks. By saying that a node is compromised, we mean that adversaries have complete control over it, including learning or modifying its secret information, changing its intended behavior, and so on. In contrast, disrupting a node means that adversaries can only disrupt communication to that node, e.g., by interfering with wireless signals to and from it, but cannot read the secret information stored on it. Therefore, node disruption attacks are less severe than node compromise attacks. However, we assume that adversaries cannot compromise or disrupt an unlimited number of nodes so that legitimate nodes are always the majority. Nor can they break any of the cryptographic primitives on which we base our design. In addition, we assume static instead of dynamic adversaries.

First of all, it is often difficult to identify an attacker who participates in the network using an id “stolen” from another legal node. For example, it is extremely difficult to detect a few attackers colluding to launch a combined wormhole and sinkhole attack. Additionally, despite the certain inevitable unfairness involved, TrustManager encourages a node to choose another route when its current route frequently fails to deliver data to the base station. Though only those legal neighboring nodes of an attacker might have correctly identified the adversary, our evaluation results indicate that the strategy of switching to a new route without identifying the attacker actually significantly improves the network performance, even with the existence of wormhole and sinkhole attacks.



(Fig:1) An example to illustrate how TrustManager works.

Fig. 1 Gives an example to illustrate this point. In this example, node A, B, C and D are all honest nodes and not compromised. Node A has node B as its current next-hop node while node B has an attacker node as its next-hop node. The attacker drops every packet received and thus any data packet passing node A will not arrive at the base station. After a while, node A discovers that the data packets it forwarded did not get delivered. The TrustManager on node A starts to degrade the trust level of its current next-hop node B although node B is absolutely honest. Once that trust level becomes too low, node A decides to select node C as its new next-hop node. In this way node A identifies a better and successful route (A - C - D - base). In spite of the sacrifice of node B's trust level, the network performs better. Further, concerning the stability of routing path, once a valid node identifies a trustworthy honest neighbor as its next-hop node, it tends to keep that next-hop selection without considering other seemingly attractive nodes such as a fake base station. That tendency is caused by both the preference to maintain stable routes and the preference to highly trustable nodes.

IV. Problem Formulation

Mobile ad hoc networks have received increasing attention in recent years due to their mobility feature, dynamic topology, and ease of deployment. A mobile ad hoc network is a self-organized wireless network which consists of mobile devices, such as laptops, cellophanes, and Personal Digital Assistants, which can freely move in the network. In addition to mobility, mobile devices cooperate and forward packets for each other to extend the limited wireless transmission range of each node by multihop relaying, which is used for various applications, e.g., disaster relief, military operation, and emergency communications. Security is one crucial requirement for these network services. Implementing security is therefore of prime importance in such networks. Provisioning protected communications between mobile nodes in a hostile environment, in which a malicious attacker can launch attacks to disrupt network security, is a primary concern.

Certificate management is a widely used mechanism which serves as a means of conveying trust in a public key infrastructure to secure applications and network services. A complete security solution for certificate management should encompass three components: prevention, detection, and revocation. Tremendous amount of research effort has been made in these areas, such as certificate distribution attack detection and certificate revocation. Certification is a prerequisite to secure network communications. It is embodied as a data structure in which the public key is bound to an attribute by the digital signature of the issuer, and can be used to verify that a public key belongs to an individual and to prevent tampering and forging in mobile ad hoc networks. Many research efforts have been dedicated to mitigate malicious attacks on the network. Any attack should be identified as soon as possible. Certificate revocation is an important task of enlisting and removing the certificates of nodes who have been detected to launch attacks on the neighborhood. In other words, if a node is compromised or misbehaved, it should be removed from the network and cut off from all its activities immediately. In our research, we focus on the fundamental security problem of certificate revocation to provide secure communications in MANETs.

Algorithm Implementation Steps

RC4

- A symmetric key encryption algorithm Invented by Ron Rivest.
- Normally uses 64 bit and 128 bit key sizes.
- Most popular implementation is in WEP for 802.11 wireless networks and in SSL.
- Cryptographically very strong yet very easy to implement.
- Consists of 2 parts: Key Scheduling Algorithm (KSA) & Pseudo-Random Generation Algorithm
- Using a secret key generate the RC4 keystream using the KSA and PRGA.
- Read the file and xor each byte of the file with the corresponding keystream byte.
- Write this encrypted output to a file.
- Transmit file over an insecure channel.

AODV Routing Protocol

The Ad-hoc On-demand Distance Vector (AODV) routing protocol is a routing protocol used for dynamic wireless networks where nodes can enter and leave the network at will. To find a route to a particular destination node, the source node broadcasts a RREQ to its immediate neighbors. If one of these neighbors has a route to the destination, then it replies back with a RREP. Otherwise the neighbors in turn rebroadcast the request. This continues until the RREQ hits the final destination or a node with a route to the destination. At that point a chain of RREP messages is sent back and the original source node finally has a route to the destination.

We proved that AODV protocol never produces routing loops by proving that a combination of sequence numbers and hop counts is monotonic along a route. This means that there can't be any loop in the routing table. The proof was done completely automatically and our algorithm was able to generate all the predicates needed. The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, selfstarting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hopcount, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

V. CONCLUSION

Nodes with low trust values will be excluded by the routing algorithm. Therefore, secure routing path can be established in malicious environments. Based on the proposed scheme, more accurate trust can be obtained by considering different types of packets, indirect observation from one-hop neighbors and other important factors such as buffers of queues and states of wireless connections, which may cause dropping packets in friendly nodes. The results of MANET routing scenario positively support the effectiveness and performance of our scheme, which improves throughput and packet delivery ratio considerably, with slightly increased average end-to-end delay and overhead of messages. Particularly, we have proposed a new incentive method to release and restore the legitimate nodes, and to improve the number of available normal nodes in the network. In doing so, we have sufficient nodes to ensure the efficiency of quick revocation.

VI. REFERENCES

[1] S. Corson and J. Macker, “Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations,” IETF RFC 2501, Jan. 1999.
[2] F. R. Yu, Cognitive Radio Mobile Ad Hoc Networks. New York: Springer, 2011.
[3] J. Loo, J. Lloret, and J. H. Ortiz, Mobile Ad Hoc Networks: Current Status and Future Trends. CRC Press, 2011.
[4] Q. Guan, F. R. Yu, S. Jiang, and V. Leung, “Joint topology control and authentication design in mobile ad hoc networks with cooperative communications,” IEEE Trans. Veh. Tech., vol. 61, pp. 2674 –2685, July 2012.
[5] F. R. Yu, H. Tang, S. Bu, and D. Zheng, “Security and quality of service (QoS) co-design in cooperative mobile ad hoc networks,” EURASIP J. Wireless Commun. Networking, vol. 2013, pp. 188–190, July 2013.
[6] Y. Wang, F. R. Yu, H. Tang, and M. Huang, “A mean field game theoretic approach for security enhancements in mobile ad hoc networks,” IEEE Trans. Wireless Commun., vol. 13, pp. 1616–1627, March 2014.
[7] J. Chapin and V. W. Chan, “The next 10 years of DoD wireless networking research,” in Proc. IEEE Milcom’11, (Baltimore, MD, USA), Nov. 2011.
[8] S. Bu, F. R. Yu, P. Liu, P. Manson, and H. Tang, “Distributed combined authentication and intrusion detection with data fusion in high-security mobile ad hoc networks,” IEEE Trans. Veh. Tech., vol. 60, pp. 1025 –1036, Mar. 2011.

[9] C. Adjih, D. Raffo, and P. Muhlethaler, “Attacks against OLSR: distributed key management for security,” in Proc. 2nd OLSR Workshop, (Domaine de Voluceau, France), Dec. 2005.
[10] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Securing mobile ad hoc networks with certificate less public keys,” IEEE Trans. Dependable and Secure Computing, vol. 3, pp. 386–399, Oct.–Dec. 2006.

AUTHOR DETAILS

Rakesh Kumar ER was born in Kanyakumari District, Tamil Nadu, India in 1985. He obtained his B.Sc., M.Sc. M.E. M.Phil. Degrees in Computer Science in the years 2005, 2007, 2010 and 2012, M.B.A Degree in Human Resources in the year of 2013 respectively. He has more than 7 years of teaching experience. He has presented 5 research papers in various national and international conferences. He has also published more than 5 research papers in reputed national and international journals. He has guided several UG and PG students for their project work. His area of interest is Network Security and Wireless Sensor Networks. Currently, he is with SAMS College of Engg. & Tech, Chennai, India, as Asst. Prof and Head of the Department of Computer Science and Engineering.