Security in Ad Hoc Networks

Kamal, Assistant Professor, MDU Rohtak, dkamal@brcm.edu.in

Dinesh Kumar, Assistant Professor, MDU Rohtak, kdinesh@brcm.edu.in

Kanishka Raheja, M.Tech Scholar, MDU Rohtak, kanishka.raheja@gmail.com

In ad hoc networks the communicating nodes do not necessarily rely on a fixed infrastructure, which sets new challenges for the necessary security architecture they apply. In addition, as ad hoc networks are often designed for specific environments and may have to operate with full availability even in difficult conditions, security solutions applied in more traditional networks may not directly be suitable for pro- tecting them. A short literature study over papers on ad hoc networking shows that many of the new generation ad hoc networking proposals are not yet able to address the security problems and they face. Environment-specific implications on the re- quired approaches in implementing security in such dynamically changing networks have not yet fully realized.

I. INTRODUCTION

Wireless networks [34] consist of a number of nodes which communicate with each other over a wireless channel which have various types of networks: sensor network, ad hoc mobile networks, cellular networks and satellite networks. Wireless sensor networks consist of small nodes with sensing, computation and wireless communications capabilities. Many routing protocols have been specifically designed for WSNs where energy awareness is the key issue. Routing protocols in WSNs [41] differ depending on the application and network architecture. Ad-hoc networks are a new paradigm of wireless communication for mobile hosts where node mobility causes frequent changes in topology. Ad hoc networks are self-configurable and autonomous systems consisting of routers and hosts, which are able to support movablity and organize themselves arbitrarily. This means that the topology of the ad hoc network changes dynamically and unpredictably. Moreover, the ad hoc network can be either constructed or destructed quickly and autonomously without any administrative server or infrastructure. Without support from the fixed infrastructure, it is undoubtedly arduous for people to distinguish the insider and outsider of the wireless network. That is to say, it is not easy for us to tell apart the legal and the illegal participants in wireless systems.

1.1Security goals

Security is an important issue for ad hoc networks, especially for those security-sensitive applications. To secure an ad hoc network, we consider the following attributes: availability, confidentiality, integrity, authen- tication, and non-repudiation.

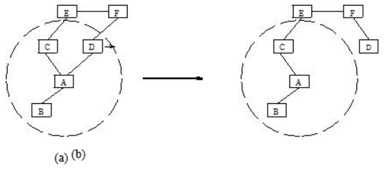


Figure 1: Topology change in ad hoc networks: nodes A, B, C, D, E, and F constitute an ad hoc network. The circle represents the radio range of node A. The network initially has the topology in (a). When node D moves out of the radio range of A, the network topology changes to the one in (b).

Availability ensures the survivability of network services despite denial of service attacks. A denial of service attack could be launched at any layer of an ad hoc network. On the physical and media access control layers, an adversary could employ jamming to interfere with communication on physical channels. On the network layer, an adversary could disrupt the routing protocol and disconnect the network. On the higher layers, an adversary could bring down high-level services. One such target is the key management service, an essential service for any security framework.

1.2Challenges

The salient features of ad hoc networks pose both challenges and opportunities in achieving these security goals.

First, use of wireless links renders an ad hoc network susceptible to link attacks ranging from passive eavesdropping to active impersonation, message replay, and message distortion. Eavesdropping might give an adversary access to secret information, violating confidentiality. Active attacks might allow the adversary to delete messages, to inject erroneous messages, to modify messages, and to impersonate a node, thus violating availability, integrity, authentication, and non-repudiation.

Secondly, nodes, roaming in a hostile environment (e.g., a battlefield) with relatively poor physical protection, have non-negligible probability of being compromised. Therefore, we should not only consider malicious attacks from outside a network, but also take into account the attacks launched from within the network by compromised nodes.

1.3Scope and roadmap

Traditional security mechanisms, such as authentication protocols, digital signature, and encryption, still play important roles in achieving confidentiality, integrity, authentication, and non-repudiation of communication in ad hoc networks. However, these mechanisms are not sufficient by themselves.

We further rely on the following two principles. First, we take advantage of redundancies in the network topology (i.e., multiple routes between nodes) to achieve availability. The second principle is distribution of trust. Although no single node is trustworthy in an ad hoc network because of low physical security and availability, we can distribute trust to an aggregation of nodes. Assuming that any t + 1 nodes will unlikely be all compromised, consensus of at least t + 1 nodes is trustworthy.

In this paper, we will not address denial of service attacks towards the physical and data link layers. Certain physical layer countermeasures such as spread spectrum have been extensively studied (e.g., [44, 6, 42, 17, 37]). However, we do focus on how to defend against denial of service attacks towards routing protocols in Section 2.

All key-based cryptographic schemes (e.g., digital signature) demand a key management service, which is responsible for keeping track of bindings between keys and nodes and for assisting the establishment of mutual trust and secure communication between nodes.

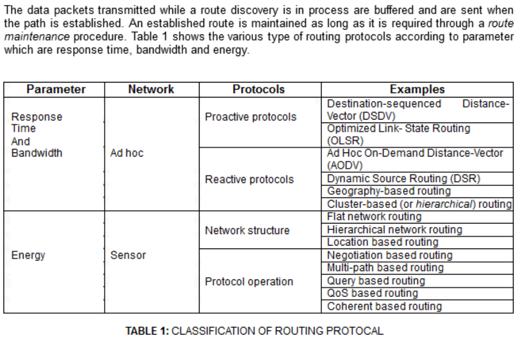
ROUTING PROTOCOL AND ITS CHALLENGE IN AD HOC NETWORK

ROUTING PROTOCOLS

Routing in mobile ad hoc networks faces additional problems and challenges [22], [30] when compared to routing in traditional wired networks with fixed infrastructure. There are several well known protocols in the literature that have been specifically developed to cope with the limitations imposed by ad hoc networking environments. Most of the existing routing protocols follow two different design approaches to confront the inherent Characteristics of ad hoc networks: the table- driven and the source-initiated on-demand approaches.

Table-driven ad hoc routing protocols maintain at all times routing information regarding the connectivity of every node to all other nodes that participate in the network. Also known as proactive, [49] these protocols allow every node to have a clear and consistent view of the network topology by propagating periodic updates [27]. An alternative approach to that followed by table-driven protocols is the source-initiated on-demand routing. According to this approach, a route is created only when the source node requires a route to a specific destination. A route is acquired by the initiation of a route discovery function by the source node.

The data packets transmitted while a route discovery is in process are buffered and are sent when the path is established. An established route is maintained as long as it is required through a route maintenance procedure. Table 1 shows the various type of routing protocols according to parameter which are response time, bandwidth and energy.



The data packets transmitted while a route discovery is in process are buffered and are sent when the path is established. An established route is maintained as long as it is required through a *route maintenance* procedure. Table 1 shows the various type of routing protocols according to parameter which are response time, bandwidth and energy.

Parameter	Network	Protocols	Examples
Response Time And Bandwidth	Ad hoc	Proactive protocols	Destination-sequenced Distance-Vector (DSDV)
			Optimized Link- State Routing (OLSR)
		Reactive protocols	Ad Hoc On-Demand Distance-Vector (AODV)
			Dynamic Source Routing (DSR)
			Geography-based routing
			Cluster-based (or *hierarchical*) routing
Energy	Sensor	Network structure	Flat network routing
			Hierarchical network routing
			Location based routing
		Protocol operation	Negotiation based routing
			Multi-path based routing
			Query based routing
			QoS based routing
			Coherent based routing

TABLE 1: CLASSIFICATION OF ROUTING PROTOCAL

3Key Management Service

We employ cryptographic schemes, such as digital signatures, to protect both routing information and data traffic. Use of such schemes usually requires a key management service.

We adopt a public key infrastructure because of its superiority in distributing keys and in achieving integrity and non-repudiation. Efficient secret key schemes are used to secure further communication after nodes authenticate each other and establish a shared secret session key.

In a public key infrastructure, each node has a public/private key pair. Public keys can be distributed to other nodes, while private keys should be kept confidential to individual nodes. There is a trusted entity called Certification Authority (CA) [11, 47, 26] for key management. The CA has a public/private key pair, with its public key known to every node, and signs certificates binding public keys to nodes.

The trusted CA has to stay on-line to reflect the current bindings, because the bindings could change over time: a public key should be revoked if the owner node is no longer trusted or is out of the network; a node may refresh its key pair periodically to reduce the chance of a successful brute-force attack on its private key.

It is problematic to establish a key management service using a single CA in ad hoc networks. The CA, responsible for the security of the entire network, is a vulnerable point of the network: if the CA is unavailable, nodes cannot get the current public keys of other nodes or to establish secure communication with others. If the CA is compromised and leaks its private key to an adversary, the adversary can then sign any erroneous certificate using this private key to impersonate any node or to revoke any certificate.

A standard approach to improve availability of a service is replication. But a naive replication of the CA makes the service more vulnerable: compromise of any single replica, which possesses the service private key, could lead to collapse of the entire system. To solve this problem, we distribute the trust to a set of nodes by letting these nodes share the key management responsibility.

B. SECURITY CHALLENGES IN AD HOC NETWORKS

Use of wireless links renders an Ad hoc network susceptible to link attacks ranging from passive eavesdropping to active impersonation, message replay and message distortion [9],[10],[52].Eavesdropping might give an attacker access to secret information thus violating confidentiality. Active attacks could range from deleting messages, injecting erroneous messages; impersonate a node etc thus violating availability, integrity, authentication and non- repudiation. Nodes roaming freely in a hostile environment with relatively poor physical protection have non-negligible probability of being compromised. Hence, we need to consider malicious attacks not only from outside but also from within the network from compromised nodes. Thus following are the ways by which security can be breached. [56]

Vulnerability of Channels: As in any wireless network, messages can be eavesdropped and fake messages can be injected into the network without the difficulty of having physical access to network components.

Vulnerability of nodes: Since the network nodes usually do not reside in physically protected places, such as locked rooms, they can more easily be captured and fall under the control of an attacker.

Absence of Infrastructure: Ad hoc networks are supposed to operate independently of any fixed infrastructure. This makes the classical security solutions based on certification authorities and on-line servers inapplicable.

Dynamically Changing Topology: In mobile ad hoc networks, the permanent changes

of topology require sophisticated routing protocols, the security of which is an additional challenge. A particular difficulty is that incorrect routing information can be generated by compromised nodes or as a result of some topology changes and it is hard to distinguish between the two cases.

III. SECURITY MODEL

In this section we first discuss security goals attacks and thus secure routing protocol which are following:

A. SECURITY GOALS FOR AD HOC

Availability: Ensures survivability despite Denial Of Service (DOS) attacks. On physical and media access control layer attacker can use jamming techniques to interfere with communication on physical channel. On network layer the attacker can disrupt the routing protocol. On higher layers, the attacker could bring down high level services e.g.: key management service.

Confidentiality: Ensures certain information is never disclosed to unauthorized entities.

Integrity: Message being transmitted is never corrupted.

Authentication: Enables a node to ensure the identity of the peer node it is communicating with. Without which an attacker would impersonate a node, thus gaining unauthorized access to resource and sensitive information and interfering with operation of other nodes.

Non-repudiation: Ensures that the origin of a message cannot deny having sent the message.

Non-impersonation: No one else can pretend to be another authorized member to learn any useful information.

Attacks using fabrication: Generation of false routing messages is termed as fabrication messages. Such attacks are difficult to detect.

B. ATTACK ON AD HOC NETWORK

There are various types of attacks on ad hoc network which are describing following:

Location Disclosure: Location disclosure is an attack that targets the privacy requirements of an ad hoc network. Through the use of traffic analysis techniques [20], or with simpler probing and monitoring approaches, an attacker is able to discover the location of a node, or even the structure of the entire network.

Black Hole: In a black hole attack a malicious node injects false route replies to the route requests it receives, advertising itself as having the shortest path to a destination[26]. These fake replies can be fabricated to divert network traffic through the malicious node for eavesdropping, or simply to attract all traffic to it in order to perform a denial of service attack by dropping the received packets.

Replay: An attacker that performs a replay attack injects into the network routing traffic that has been captured previously. This attack usually targets the freshness of routes, but can also be used to undermine poorly designed security solutions.

Wormhole: The wormhole attack is one of the most powerful presented here since it involves the cooperation between two malicious nodes that participate in the network [53]. One attacker, e.g. node A, captures routing traffic at one point of the network and tunnels them to another point in the network, to node B, for example, that shares a private communication link with A. Node B then selectively injects tunneled traffic back into the network. The connectivity of the nodes that have established routes over the wormhole link is completely under the control of the two colluding attackers. The solution to the wormhole attack is packet leashes.

Blackmail: This attack is relevant against routing protocols that use mechanisms for the identification of malicious nodes and propagate messages that try to blacklist the offender

[58].An attacker may fabricate such reporting messages and try to isolate legitimate nodes from the network. The security property of non-repudiation can prove to be useful in such cases since it binds a node to the messages it generated.

Denial of Service: Denial of service attacks aim at the complete disruption of the routing function and therefore the entire operation of the ad hoc network [15]. Specific instances of denial of service attacks include the routing table overflow and the sleep deprivation torture.. In a routing table overflow attack the malicious node floods the network with bogus route creation packets in order to consume the resources of the participating nodes and disrupt the establishment of legitimate routes. The sleep deprivation torture attack aims at the consumption of batteries of a specific node by constantly keeping it engaged in routing decisions.

Routing Table Poisoning: Routing protocols maintain tables that hold information regarding routes of the network. In poisoning attacks the malicious nodes generate and send fabricated signaling traffic, or modify legitimate messages from other nodes, in order to create false entries in the tables of the participating nodes [15]. For example, an attacker can send routing updates that do not correspond to actual changes in the topology of the ad hoc network. Routing table poisoning attacks can result in the selection of non-optimal routes, the creation of routing loops, bottlenecks, and even portioning certain parts of the network.

Rushing Attack: Rushing attack is that results in denial-of-service when used against all previous on-demand ad hoc network routing protocols [55]. For example, DSR, AODV, and secure protocols based on them, such as Ariadne, ARAN, and SAODV, are unable to discover routes longer than two hops when subject to this attack. develop Rushing Attack Prevention (RAP), a generic defense against the rushing attack for on-demand protocols that can be applied to any existing on-demand routing protocol to allow that protocol to resist the rushing attack.

Breaking the neighbor relationship: An intelligent filter is placed by an intruder on a communication link between two ISs(Information system) could modify or change information in the routing updates or even intercept traffic belonging to any data session.

Masquerading: During the neighbor acquisition process, a outside intruder could masquerade an nonexistent or existing IS by attaching itself to communication link and illegally joining in the routing protocol do main by compromising authentication system. The threat of masquerading is almost the same as that of a compromised IS.

Passive Listening and traffic analysis: The intruder could passively gather exposed routing information. Such a attack can not effect the operation of routing protocol, but it is a breach of user trust to routing the protocol. Thus, sensitive routing information should be protected. However, the confidentiality of user data is not the responsibility of routing protocol

3.1System model

Our key management service is applicable to an asynchronous ad hoc network; that is, a network with no bound on message-delivery and message-processing times. We also assume that the underlying network layer

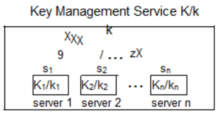


Figure 2: The configuration of a key management service: the key management service consists of n servers. The service, as a whole, has a public/private key pair K/k. The public key K is known to all nodes in the network, whereas the private key k is divided into n shares s_1, s_2, . . . , s_n, one share for each server. Each server i also has a public/private key pair K_i/k_i and knows the public keys of all nodes.

provides reliable linksi. The service, as a whole, has a public/private key pair. All nodes in the system know the public key of the service and trust any certificates signed using the corresponding private key. Nodes, as clients, can submit query requests to get other clients' public keys or submit update requests to change their own public keys.

Internally, our key management service, with an (n, t+1) configuration (n ≥ 3t+1), consists of n special nodes, which we call servers, present within an ad hoc network. Each server also has its own key pair and stores the public keys of all the nodes in the network. In particular, each server knows the public keys of other servers. Thus, servers can establish secure links among them. We assume that the adversary can compromise up to t servers in any period of time with a certain durationii.

If a server is compromised, then the adversary has access to all the secret information stored on the server. A compromised server might be unavailable or exhibit Byzantine behavior (i.e., it can deviate arbitrarily from its protocols). We also assume that the adversary lacks the computational power to break the cryptographic schemes we employ.

The service is correct if the following two conditions hold:

•(Robustness) The service is always able to process query and update requests from clients. Every query always returns the last updated public key associated with the requested client, assuming no concurrent updates on this entry.

•(Confidentiality) The private key of the service is never disclosed to an adversary. Thus, an adversary is never able to issue certificates, signed by the service private key, for erroneous bindings.

3.3Proactive security and adaptability

Besides threshold signature, our key management service also employs share refreshing to tolerate mobile adversariesiv and to adapt its configuration to changes in the network.

Mobile adversaries are first proposed by Ostrovsky and Yung [31] to characterize adversaries that tem- porarily compromise a server and then move on to the next victim (e.g., in form of viruses injected into a network). Under this adversary model, an adversary might be able to compromise all the servers over a long period of time. Even if the compromised servers are detected and excluded from the service, the adversary could still gather more than t shares of the private key from compromised servers over time. This would allow the adversary to generate any valid certificates signed by the private key.

Proactive schemes [24, 20, 19, 10, 9] are proposed as a countermeasure to mobile adversaries. A proactive threshold cryptography scheme uses share refreshing, which enables servers to compute new shares from old ones in collaboration without disclosing the service private key to any server. The new shares constitute a new (n, t + 1) sharing of the service private key. After refreshing, servers remove the old shares and use the new ones to generate partial signatures. Because the new shares are independent of the old ones, the ivNote that the term mobile here is different from that of mobile networks.

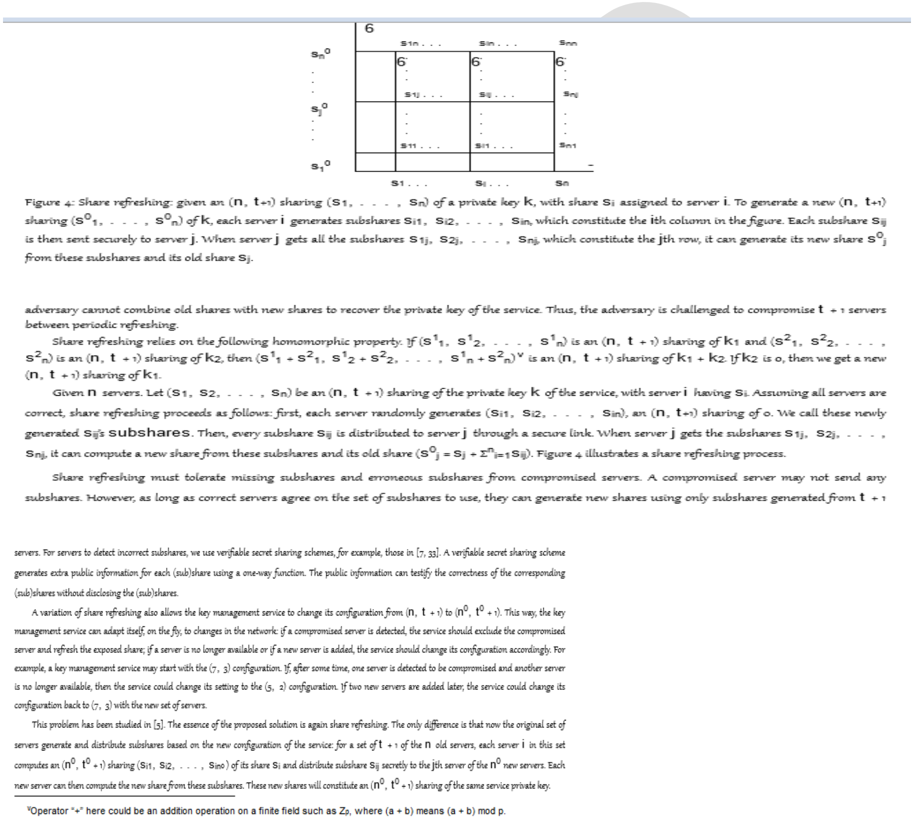

6
s_{1n} . . .
s_{in} . . .
s_{nn}
s_n^0
6
6
6
s_{1j} . . .
s_{ij} . . .
s_{nj}
s_j^0
s_{11} . . .
s_{i1} . . .
s_{n1}
s_1^0
s_1 . . .
s_i . . .
s_n

Figure 4: Share refreshing: given an $(n, t+1)$ sharing $(s_1, \dots, s_n)$ of a private key k, with share s_i assigned to server i. To generate a new $(n, t+1)$ sharing $(s^0_1, \dots, s^0_n)$ of k, each server i generates subshares $s_{i1}, s_{i2}, \dots, s_{in}$, which constitute the ith column in the figure. Each subshare s_{ij} is then sent securely to server j. When server j gets all the subshares $s_{1j}, s_{2j}, \dots, s_{nj}$, which constitute the jth row, it can generate its new share s^0_j from these subshares and its old share s_j.

adversary cannot combine old shares with new shares to recover the private key of the service. Thus, the adversary is challenged to compromise $t + 1$ servers between periodic refreshing.

Share refreshing relies on the following homomorphic property. If $(s^1_1, s^1_2, \dots, s^1_n)$ is an $(n, t + 1)$ sharing of k_1 and $(s^2_1, s^2_2, \dots, s^2_n)$ is an $(n, t + 1)$ sharing of k_2, then $(s^1_1 + s^2_1, s^1_2 + s^2_2, \dots, s^1_n + s^2_n)^v$ is an $(n, t + 1)$ sharing of $k_1 + k_2$. If k_2 is 0, then we get a new $(n, t + 1)$ sharing of k_1.

Given n servers. Let $(s_1, s_2, \dots, s_n)$ be an $(n, t + 1)$ sharing of the private key k of the service, with server i having s_i. Assuming all servers are correct, share refreshing proceeds as follows: first, each server randomly generates $(s_{i1}, s_{i2}, \dots, s_{in})$, an $(n, t+1)$ sharing of 0. We call these newly generated s_{ij}'s subshares. Then, every subshare s_{ij} is distributed to server j through a secure link. When server j gets the subshares $s_{1j}, s_{2j}, \dots, s_{nj}$, it can compute a new share from these subshares and its old share ($s^0_j = s_j + \Sigma^n_{i=1} s_{ij}$). Figure 4 illustrates a share refreshing process.

Share refreshing must tolerate missing subshares and erroneous subshares from compromised servers. A compromised server may not send any subshares. However, as long as correct servers agree on the set of subshares to use, they can generate new shares using only subshares generated from $t + 1$

servers. For servers to detect incorrect subshares, we use verifiable secret sharing schemes, for example, those in [7, 33]. A verifiable secret sharing scheme generates extra public information for each (sub)share using a one-way function. The public information can testify the correctness of the corresponding (sub)shares without disclosing the (sub)shares.

A variation of share refreshing also allows the key management service to change its configuration from $(n, t + 1)$ to $(n^0, t^0 + 1)$. This way, the key management service can adapt itself, on the fly, to changes in the network: if a compromised server is detected, the service should exclude the compromised server and refresh the exposed share; if a server is no longer available or if a new server is added, the service should change its configuration accordingly. For example, a key management service may start with the (7, 3) configuration. If, after some time, one server is detected to be compromised and another server is no longer available, then the service could change its setting to the (5, 2) configuration. If two new servers are added later, the service could change its configuration back to (7, 3) with the new set of servers.

This problem has been studied in [5]. The essence of the proposed solution is again share refreshing. The only difference is that now the original set of servers generate and distribute subshares based on the new configuration of the service: for a set of $t + 1$ of the n old servers, each server i in this set computes an $(n^0, t^0 + 1)$ sharing $(s_{i1}, s_{i2}, \dots, s_{in^0})$ of its share s_i and distribute subshare s_{ij} secretly to the jth server of the n^0 new servers. Each new server can then compute the new share from these subshares. These new shares will constitute an $(n^0, t^0 + 1)$ sharing of the same service private key.

[v] Operator "+" here could be an addition operation on a finite field such as Z_p, where (a + b) means (a + b) mod p.

3.4Asynchrony

Existing threshold cryptography and proactive threshold cryptography schemes assume a synchronous system (i.e., there is a bound on message-delivery and message-processing times). This assumption is not necessarily valid in an ad hoc network, considering the low reliability of wireless links and poor connectivity among nodes. In fact, any synchrony assumption is a vulnerability in the system: the adversary can launch denial of service attacks to slow down a node or to disconnect a node for a long enough period of time to invalidate the synchrony assumption. Consequently, protocols based on the synchrony assumption are inadequate.

To reduce such vulnerability, our key management service works in an asynchronous setting. Designing such protocols is hard; some problems may even be impossible to solve [8]. The main difficulty lies in the fact that, in an asynchronous system, we cannot distinguish a compromised server from a correct but slow one.

One basic idea underlying our design is the notion of weak consistency: we do not require that the correct servers be consistent after each operation; instead, we require enough correct servers to be up-to-date. For example, in share refreshing, without any synchrony assumption, a server is no longer able to distribute the subshares to all correct servers using a reliable broadcast channel. However, we only require subshares to be distributed to a quorum of servers. This suffices, as long as correct servers in such a quorum can jointly provide or compute all the subshares that are distributed. This way, correct servers not having certain subshare(s) could recover its subshare(s) from other correct servers.

Another important mechanism is the use of multiple signatures for correct servers to detect and to reject erroneous messages sent by compromised servers. That is, we require that certain messages be accompanied with enough signatures from servers. If a message contains digital signatures from a certain number (say, t + 1) of servers testifying its validity, at least one correct server must have provided one signature, thus establishing the validity of the message.

We have implemented a prototype of such a key management service. The preliminary results have shown its feasibility. Due to the length restriction of this paper, we are unable to provide a detailed description of this service. Full papers describing the key management service and its underlying proactive secret sharing protocol in asynchronous system are in preparation.

4Related Work

4.1Secure routing

Secure routing in networks such as the Internet has been extensively studied [36, 27, 30, 45, 46, 18]. Many proposed approaches are also applicable to secure routing in ad hoc networks. To deal with external attacks, standard schemes such as digital signatures to protect information authenticity and integrity have been con- sidered. For example, Sirios and Kent [45] propose the use of a keyed one-way hash function with windowed sequence number for data integrity in point-to-point communication and the use of digital signatures to protect messages sent to multiple destinations.

Perlman [36] studies how to protect routing information from compromised routers in the context of Byzantine robustness. The study analyzes the theoretical feasibility of maintaining network connectivity under such assumptions. Kumar [27] recognizes the problem of compromised routers as a hard problem, but provides no solution. Other works [30, 45, 46] give only partial solutions. The basic idea underlying these solutions is to detect inconsistency using redundant information and to isolate compromised routers. For example, in [46], where methods to secure distance-vector routing protocols are proposed, extra information of a predecessor in a path to a destination is added into each entry in the routing table. Using this piece of information, a path-traversal technique (by following the predecessor link) can be used to verify the correctness of a path. Such mechanisms usually come with a high cost and are avoided (e.g., in [30]) because routers on networks such as the Internet are usually well protected and rarely compromised.

.3Security in ad hoc networks

In [22], an authentication architecture for mobile ad hoc networks is proposed. The proposed scheme details the formats of messages, together with protocols that achieve authentication. The architecture can accom- modate different authentication schemes. Our key management service is a prerequisite for such a security architecture.

5Conclusion

In this paper, we have analyzed the security threats an ad hoc network faces and presented the security objectives that need to be achieved. On one hand, the security-sensitive applications of ad hoc networks require high degree of security; on the other hand, ad hoc networks are inherently vulnerable to security attacks. Therefore, security mechanisms are indispensable for ad hoc networks. The idiosyncrasy of ad hoc networks poses both challenges and opportunities for these mechanisms.

This paper focuses on how to secure routing and how to establish a secure key management service in an ad hoc networking environment. These two issues are essential to achieving our security goals. Besides the standard security mechanisms, we take advantage of the redundancies in ad hoc network topology and use diversity coding on multiple routes to tolerate both benign and Byzantine failures. To build a highly available and highly secure key management service, we propose to use threshold cryptography to distribute trust among a set of servers. Furthermore, our key management service employs share refreshing to achieve proactive security and to adapt to changes in the network in a scalable way. Finally, by relaxing the consistency requirement on the servers, our service does not rely on synchrony assumptions. Such assumptions could lead to vulnerability. A prototype of the key management service has been implemented, which shows its feasibility.

The paper represents the first step of our research to analyze the security threats, to understand the security requirements for ad hoc networks, and to identify existing techniques, as well as to propose new mechanisms to secure ad hoc networks. More work needs to be done to deploy these security mechanisms in an ad hoc network and to investigate the impact of these security mechanisms on the network performance.

References

[1]E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo. Diversity coding for transparent self-healing and fault-tolerant communication networks. IEEE Transactions on Communications, 41(11):1677–1686, November 1993.

[2]M. Castro and B. Liskov. Practical Byzantine fault tolerance. In Proceedings of the 3rd USENIX Symposium on Operating System Design and Implementation (OSDI’99), pages 173–186, New Orleans, LA USA, February 22–25, 1999. USENIX Association, IEEE TCOS, and ACM SIGOPS.

[3]Y. Desmedt. Threshold cryptography. European Transactions on Telecommunications, 5(4):449–457, July–August 1994.

[4]Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, Advances in Cryptology— Crypto'89, the 9th Annual International Cryptology Conference, Santa Barbara, CA USA, August 20–24, 1989, Proceedings, volume 435 of Lecture Notes in Computer Science, pages 307–315. Springer, 1990.

[5]Y. Desmedt and S. Jajodia. Redistributing secret shares to new access structures and its applications. Technical Report ISSE TR-97-01, George Mason University, July 1997.

[6]A. Ephremides, J. E. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. Proceedings of the IEEE, 75(1):56–73, January 1987.

[7]P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th Annual Symposium on the Foundations of Computer Science, pages 427–437. IEEE, October 12–14, 1987.

[8]M. J. Fischer, N. A. Lynch, and M. S. Peterson. Impossibility of distributed consensus with one faulty processor. Journal of the ACM, 32(2):374–382, April 1985.

[9]Y. Frankel, P. Gemmel, P. MacKenzie, and M. Yung. Optimal resilience proactive public-key cryp- tosystems. In Proceedings of the 38th Symposium on Foundations of Computer Science, pages 384–393, Miami Beach, FL USA, October 20–22, 1997. IEEE.

[10]Y. Frankel, P. Gemmell, P. MacKenzie, and M. Yung. Proactive RSA. In B. S. Kaliski Jr., editor,

Advances in Cryptology—Crypto'97, the 17th Annual International Cryptology Conference, Santa Bar- bara, CA USA, August 17–21, 1997, Proceedings, volume 1294 of Lecture Notes in Computer Science, pages 440–454. Springer, 1997.

[11]M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson. The digital distributed systems security archi- tecture. In Proceedings of the 12th National Computer Security Conference, pages 305–319, Baltimore, MD USA, October 10–13, 1989. National Institute of Standards and Technology (NIST), National Computer Security Center (NCSC).

[12]R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust and efficient sharing of RSA functions. In N. Koblitz, editor, Advances in Cryptology—Crypto'96, the 16th Annual International Cryptology Conference, Santa Barbara, CA USA, August 18–22, 1996, Proceedings, volume 1109 of Lecture Notes in Computer Science, pages 157–172. Springer, 1996.

[13]R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In U. M. Maurer, editor, Advances in Cryptology—Eurocrypt'96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12–16, 1996, Proceedings, volume 1233 of Lecture Notes in Computer Science, pages 354–371. Springer, 1996.