International Journal of Scientific Engineering and Technology ISSN : 2277-1581)
Volume No.2, Issue No.9, pp : 884-891 1 Sept. 2013

Malicious Attacks in Ad Hoc Networks - Detection & Protection

Gurnitika Kaur, Khyati Marwah
Department of Computer Science & Engineering
Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib
gurnitika@gmail.com, er.khyati@gmail.com

Abstract

Ad hoc networks provide a new wireless networking mechanism for mobile communication. Ad hoc networks do not have any fixed infrastructure as is the case for conventional mobile wireless networks. In ad hoc networks, various network nodes communicate with each other to keep the network connected. Ad hoc networks, being a recent technological development, are mainly being used in space, military and other security-sensitive operations, but their unique properties are fast extending their applications to industrial and commercial fields also. The biggest challenge that we face in design and application of these networks is that these are quite prone to security attacks. In the present work, we try to unravel the various attacks an ad hoc network can encounter and how to protect the network from these. We identify the new challenges and opportunities posed by this new networking environment and explore new approaches to make secure the communication secure. A mechanism is evolved where we capitalize on the inherent redundancy in ad hoc networks — multiple routes between nodes — to defend data from any malicious attacks. Replication and coding techniques are also used to enhance the security of the network.

1 Introduction

Ad hoc networks provide a new wireless networking mechanism for mobile communication. Ad hoc networks do not have any fixed infrastructure as is the case for conventional mobile wireless networks. Mobile nodes that are within each other's radio range communicate directly via wireless links, while those that are far apart rely on other nodes to relay messages as routers. The movement in the nodes in an ad hoc network causes frequent changes of the network topology. Figure 1 shows such an example: initially, nodes A and D have a direct link between them. When D moves out of A's radio range, the link is broken. However, the network is still connected, because A can reach D through C, E, and F.

Ad hoc networks are still mainly being used in defense operations. For example, military units (e.g., soldiers, tanks, or planes), equipped with wireless communication devices, could form an ad hoc network when they roam in a battlefield. Ad hoc networks can also be used for emergency, law enforcement, and rescue missions. Since an ad hoc network can be deployed rapidly with relatively low cost, it becomes an attractive option for commercial uses such as sensor networks or virtual classrooms.

1.1 Security 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, authentication, and non-repudiation.

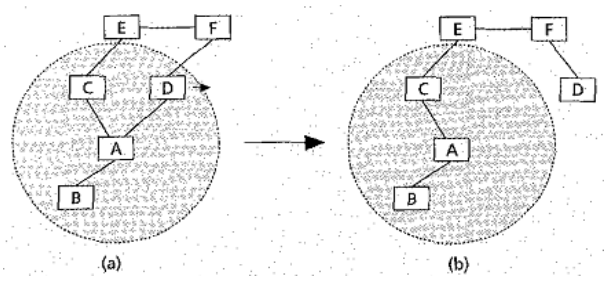


Figure 1: Topology change in ad hoc networks

Figure 1 shows 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).

i. Availability: It 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.

ii. Confidentiality: It ensures that certain information is never disclosed to unauthorized entities. Network transmission of sensitive information, such as strategic or tactical military information, requires confidentiality. Leakage of such information to enemies could have devastating consequences. Routing information must also remain confidential in certain cases, because the information might be valuable for enemies to identify and to locate their targets in a battlefield.

iii. Integrity: It guarantees that a message being transferred is never corrupted. A message could be corrupted because of benign failures, such as radio propagation impairment, or because of malicious attacks on the network.

iv. Authentication: It enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes.

v. Non-repudiation: It ensures that the origin of a message cannot deny having sent the message. Nonrepudiation is useful for detection and isolation of compromised nodes. When a node A receives an erroneous message from a node B, non-repudiation allows A to accuse B using this message and to convince other nodes that B is compromised.

There are other security goals (e.g., authorization) that are of concern to certain applications, but we will not pursue these issues in this paper.

1.2 Challenges

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. Therefore, to achieve high survivability, ad hoc networks should have adistributed architecture with no central entities. Introducing any central entity into our security solution could lead to significant vulnerability that is, if this centralized entity is compromised, then the entire network is subverted.

Thirdly, an ad hoc network is dynamic because of frequent changes in both its topology and its membership (i.e., nodes frequently join and leave the network). Trust relationship among nodes also changes, for example, when certain nodes are detected as being compromised. Unlike other wireless mobile networks, such as mobile IP [21], nodes in an ad hoc network may dynamically become affiliated with administrative domains. Any security solution with a static configuration would not suffice. It is desirable for our security mechanisms to adapt on-the-fly to these changes.

Finally, an ad hoc network may consist of hundreds or even thousands of nodes. Security mechanisms should be scalable to handle such a large network.

1.3 Scope 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., [6, 17]). 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. We will focus our discussion in Section 3 on how to establish such a key management service that is appropriate for ad hoc networks. We present related work in Section 4 and conclude in Section 5.

2. Secure Routing

To achieve availability, routing protocols should be robust against both dynamically changing topology and malicious attacks. Routing protocols [30, 25, 43, 32, 49, 16, 23, 35] proposed for ad hoc networks cope well with the dynamically changing topology. However, none of them, to our knowledge, have accommodated mechanisms to defend against malicious attacks. Routing protocols for ad hoc networks are still under active research. There is no single standard routing protocol. Therefore, we aim to capture the common security threats and to provide guidelines to secure routing protocols.

The nodes in an ad hoc network also function as routers that discover and maintain routes to other nodes in the network. The primary goal of adhoc network routing protocol is to establish a correct and efficient route between a pair of nodes so that messages may be delivered in a timely manner. If routing can be misdirected, the entire network can be paralyzed. Thus, routing security plays an important role in the security of the whole network. Here, we first briefly introduce some currently proposed routing protocols for adhoc networks.

2.1. Routing protocols of adhoc networks

Many different routing protocols have been developed for adhoc networks. They can be classified into two categories:

Table-driven: Table driven routing protocols essentially use proactive schemes. They attempt to maintain consistent up-to-date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information, and any changes in network topology need to be reflected by propagating updates throughout the network in order to maintain a consistent network view.

On demand: A different approach from tabledriven routing is source-initiated on-demand routing. This type of routing creates routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined.

Three main routing protocols for adhoc network are destination-sequenced distance-vector routing protocol (DSDV), AODV, and Dynamic Source Routing protocol (DSR). DSDV is a table-driven routing protocol based on the classical Bellman-Ford routing mechanism. In this routing protocol, each mobile node in the system maintains a routing table in which all the possible destinations and the number of hops to them in the network are recorded. AODV builds on the DSDV algorithm described above and is an improvement since it typically minimizes the number of required broadcasts by creating routes on a demand basis, as opposed to maintaining a complete list of routes as in DSDV. It is an ondemand route acquisition system, since nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges. DSR is different from AODV in the sense that each mobile node keeps track of the routes of which it is aware in a route cache. Upon receiving a search request for path, it consults with its route cache to see if it contains the required information. This protocol uses more memory while reducing the route discovery delay in the system.

Effective operation of an adhoc network is dependent on maintaining appropriate routing information in a distributed fashion. But no security is considered in currently proposed routing protocols, which makes the routing protocol an easy target for attackers.

3. Key Management Service

Cryptographic schemes such as digital signatures are often employed to protect both routing info as well as data. Public key systems are generally espoused because of its upper hand in key distribution. In public key infrastructure each node has a public/private key pair. Public keys distributed to other nodes, while private keys are kept to nodes themselves and that too confidentially. Third party (trusted) called Certification Authority (CA) is used for key management.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 online to reflect the current bindings, since the bindings could change overtime. Public key should be revoked if the owner node is no longer trusted or is out of network. A single key management service for an Ad-hoc network is probably not a good idea, since it's likely to become Achilles' heel of the network. If CA is down/unavailable nodes cannot get the current public keys of other nodes to establish secure connection. Also if a CA is compromised, the attacker can sign any erroneous certificates with the private key. Naive replication of CA can make the network more vulnerable, since compromising of a single replica can cause the system to fail. Hence it's more prudent to distribute the trust to a set of nodes by letting these nodes share the key management responsibility.

The security in networking is in many cases dependent on

proper key management. Key management consists of various services, of which each is vital for the security of the networking systems. The services must provide solutions to be able to answer the following questions:

i. **Trust model**: it must be determined how much different elements in the network can trust each other. The environment and area of application of the network greatly affects the required trust model. Consequently, the trust relationships between network elements affects the way the key management system is constructed in network.

ii. **Cryptosystems**: available for the key management: in some cases only public- or symmetric key mechanisms can be applied, while in other contexts Elliptic Curve Cryptosystems (ECC) are available. While public-key cryptography offers more convenience (e.g. by well-known digital signature schemes), public-key cryptosystems are significantly slower than their secret-key counterparts when similar level of security is needed. On the contrary, secret-key systems offer less functionality and suffer more from problems in e.g. key distribution. ECC cryptosystems are a newer field of cryptography in terms of implementations, but they are already in use widely, for instance in smart card systems.

iii. **Key creation**: it must be determined which parties are allowed to generate keys to themselves or other parties and what kind of keys.

iv. **Key storage**: in ad-hoc networks there may not be a centralized storage for keys. Neither there may be replicated storage available for fault tolerance. In ad-hoc networks any network element may have to store its own key and possibly keys of other elements as well. Moreover, in some proposals such as in [19], shared secrets are applied to distribute the parts of keys to several nodes. In such systems the compromising of a single node does not yet compromise the secret keys.

v. **Key distribution**: the key management service must ensure that the generated keys are securely distributed to their owners. Any key that must be kept secret has to be distributed so that confidentiality, authenticity and integrity are not violated. For instance whenever symmetric keys are applied, both or all of the parties involved must receive the key securely. In public-key cryptography the key distribution mechanism must guarantee that private keys are delivered only to authorized parties. The distribution of public keys need not preserve confidentiality, but the integrity and authenticity of the keys must still be ensured.

3.1. Threshold Cryptography

Threshold Cryptography is the art of chopping a secret into little bits. Only by possessing more than a threshold number of bits of the secret can the secret be determined.

For example if I had four copies of a key and cut each key into three pieces, distributing one piece each to twelve people then it would probably take about five people to use the key. The minimum threshold in this case would be three, and the maximum number of servers to contact would be 9. That's with random distribution. With algorithmic, non-random distribution based on name (e.g. the four lowest in the alphabet get piece1, the next four get piece2, etc.), this can be reduced to exactly three successful contacts necessary to find the whole key.

Algorithms exist to break any secret up such that at least and exactly M out of N holders of pieces of the secret must give approval (and their partial secret or key) in order to compute the total secret (e.g. 3 of 5, 3 of 12, 5 of 12, etc.). Removing probability has a cost, though... a secret must be broken into C(N,M-1) pieces and each holder carries (N-M+1)/N parts of the whole key... so '3 of 12' is more expensive per-node than '5 of 12'. (These numbers come from the pigeonhole principle and constraints: any piece 'pK' must be found on N-M+1 holders so that access to a full M secret-holders guarantees 'pK' will be known, whilst access to M-1 computers must guarantee that there is at least one piece not found, so 'pK' must NOT be with the other M-1 computers. The minimum number of component 'pK' elements to do this is (N) Choose (M-1). Individual pK elements can be made artificially large in order to subvert guessing of one or two missing pieces; the combinator function needn't be straightforward append. However, The computation and storage cost of this approach is high, and it may do well to combine it with some straightforward split-and-distribute as listed above; e.g. splitting the 'require 5 of 7 pieces' to 'more than 7' people is the natural extension to splitting 'require 3 of 3 pieces' to '12 people'. The combined effect can avoid the massive costs of splitting and storing, say '5 of 20' parts (C(20,4) unique parts, every node holding ~16/20ths of total secret, vs. C(7,4) parts with each node holding 3/7ths of total secret). The main advantage of mixing in this algorithmic division approach is in achieving better guarantees as to redundancy and survivability while simultaneously increasing the number of users one must access to possess the whole secret. E.g. for the other approach, to require 5 users would require splitting the key into 5 pieces and divvying that up among, say, 15 people; it would take access to 5 people to gain the secret, and the secret could be lost by losing 3 people. Splitting it to 5 of 7 first, then dividing the 7 chunks among 14 people results in 2 different people having a copy of any given chunk, and the

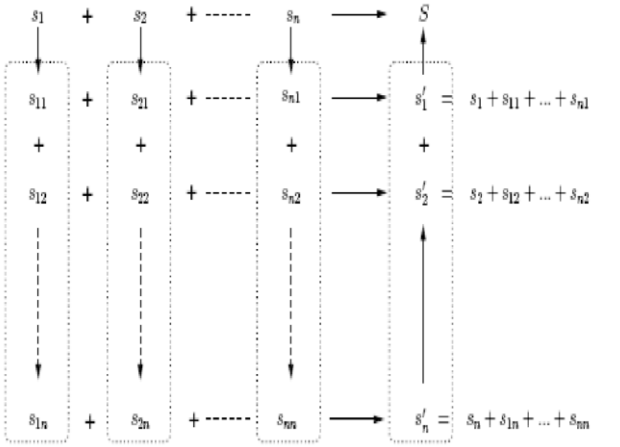
secret won't be lost before losing 6 people (losing three whole chunks).

As a security measure, Threshold Cryptography requires that many systems must be compromised prior to taking control of a secret, inherently including resistance to snooping or abuse by any super users of the computation resource (who would have the ability to do so if the secret were wholly on one system). It also provides inherent redundancy of the secret... e.g. if you can guarantee that it takes at least and at most 5 of 12 secret-holders to build the secret, you can guarantee that a failure of up to 7 systems is tolerable without failure. With a probabilistic split, you can easily calculate a percentage chance that the data is unavailable for each loss of node... and, with intelligent split of components, you can guarantee that at least some count of nodes must be lost before the data has any chance of being lost.

3.2. Proactive Secret Sharing

Threshold cryptography provides the way to convey a shared key to a node without the aid of key infrastructure and is suitable for a secret key sharing in a MANET. However, given t or more shares in an (n, t) threshold cryptography scheme, the secret S can be found. Without the share refresh and with infinite time span it is not hard for malicious nodes to compromise at least t shareholder nodes and finally obtain the secret key.

To make each share refresh without disclosure of any share or a secret key itself, Proactive Secret Sharing (PSS) can be employed with threshold cryptography as an additional component. It allows to refresh all shares by generating a new set of shares for the same secret key from the old shares without reconstructing the secret key.



[A]

In the PSS implementation, each shareholder randomly generates own sub-shares e.g., (si1, si2, . . . , sin) on node i), and each sub-share is mutually exchanged to refresh own share. More precisely, the PSS procedure shown in Fig.1 can be performed in the following steps:

1) Let (s1, s2, . .,sn) be an (n, t) sharing of the secret key Sof the service, with node ihaving si.

2) Node i (i∈ {1 . . . n}) randomly generates si's sub shares (si1, si2, . . . , sin) for an (n, t) sharing.

3) Every sub-share sij (j ∈ {1 . . . n}) is distributed to node jthrough secure link.

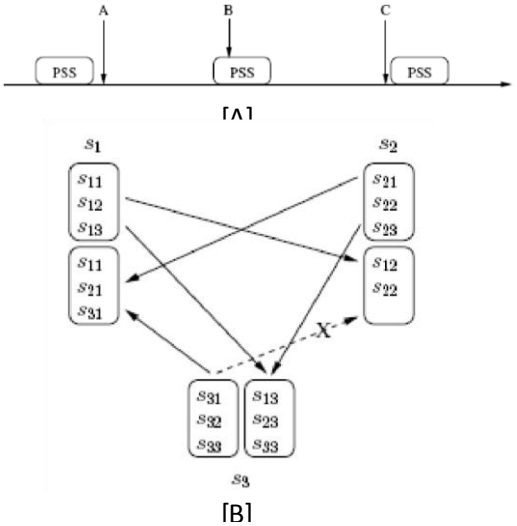
When node jgets the sub-shares (s1j, s2j, . . . ,snj), it computes a new share from these sub-shares and its old share with an equation

$$s'_j = s_j + \sum_{i=1}^{n} s_{ij}$$

After each PSS, all the shares will be changed, so that old shares become useless. In such case, since it is impossible to obtain new share from old share, a malicious node must collect at least tshares during the time between two executions of the PSS, which obviously makes his job more difficult.

3.2.1 PSS Scheme

PSS is a periodic or on-demand protocol. Any share holder node (master node) in the network can trigger the PSS procedure at any time. Here, objective exists in this condition. If an arbitrary share holder starts a PSS procedure without any timing coordination with other shareholders, or if each shareholder tries to refresh its own share using some concurrent scheme, where all shareholders just perform communication and computation in parallel, inconsistency may happen, and a new node may receive inconsistent shares and then cannot generate the secret key.



[B]

Let us see Fig.2 [A] showing periodic PSS procedures. In an (n, t) sharing of S, if a new node requests shares and can get tshares between the last and the next PSS procedures (e.g. A), the node will generate Swithout any trouble. However, if a new node requests to get shares during PSS (e.g. B), shareholders should make the node wait for their PSS and give their new shares

Afterward. Or, if a new node tries to get shares just before PSS is started (e.g. C), each shareholder should put the PSS on hold until the node successfully get current tshares. These matters imply that a shareholder (master node) needs to know whena new share can be used as own share and whenthe new share can be given to a new node. As well as such timing issue, there is another concern related to untraceable sub-share transmission. The considerable story is, while a shareholder can understand whether it has gotten sub-shares generated by other shareholders successfully, it does not know other shareholders also have gotten necessary sub-shares successfully. Or, if a shareholder does not get sub-shares from corresponding shareholders due to loss of connectivity with them, it should not keep waiting for the reply from the shareholders for a long span of time. This situation will be especially happened in a MANET environment in which mobile nodes may leave from the network or turn off itself without any notification. Let us see Fig.2 [B]. Three shareholders (s1, s2, and s3) start their PSS procedure and exchange each sub-share. For instance, because of the link failure, s3 cannot give its sub-share to s2. The worst phenomenon here is *s*1 does not know that s2 and s3 had the share transmission problem and hence may start giving its new share to a new node, which is an inconsistent share to other old shares. In this case, shareholders should keep own old shares until all shareholders complete their PSS (for a short period) or change the (n, t) sharing to an (n − 1, t − 1) sharing (with eliminating the left share holder) along their pre-determined policy.

3.3Asynchrony

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 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.

4. Related Work

4.1 Secure routing

Secure routing in networks such as the Internet has been extensively studied. 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 considered. 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.

Perlma 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.

4.2 Replicated secure services

The concept of distributing trust to a group of servers is investigated by Reiter [39]. This is the foundation of the Rampart toolkit [38]. Reiter and others [40] have successfully used the toolkit in building a replicated key management service Ω, which also employs threshold cryptography. One drawback of Rampart is that it may remove correct but slow servers from the group. Such removal renders the system at least temporarily more vulnerable. Membership changes are also expensive. For these reasons, Rampart is more suitable for tightly coupled networks than for ad hoc networks.

Gong [14] applies trust distribution to Key Distribution Center (KDC), the central entity responsible for key management in a secret key infrastructure. In his solution, a group of servers jointly act as a KDC with each server sharing a unique secret key with each client.

In [29], Malkhi and Reiter present Phalanx, a data repository service that tolerate Byzantine failures in an asynchronous system. The essence of Phalanx is a Byzantine quorum system [28]. In a Byzantine quorum system, servers are grouped into quorums satisfying a certain intersection property. The service supports read and write operations and guarantees that a read operation always returns the result of the last completed write operation. Instead of requiring each correct server to perform each operation, the service performs each operation on only a quorum of servers. However, this weak consistency among the servers suffices to achieve the guarantee of the service because of the intersection property of Byzantine quorum systems.

In [2], Castro and Liskov extend the replicated state-machine approach [41] to achieve Byzantine fault tolerance. They use a three-phase protocol to mask away disruptive behavior of compromised servers. A small portion of servers may be left behind, but can recover by communicating with other servers.

None of the systems provide mechanisms to defeat mobile adversaries and to achieve scalable adaptability. The latter two solutions do not consider how a secret (a private key) is shared among the replicas. However, they are useful in building highly secure services in ad hoc networks. For example, we could use Byzantine quorum systems to secure a location database [15] for an ad hoc network.

4.3 Security 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 accommodate different authentication schemes. Our key management service is a prerequisite for such a security architecture.

5 Conclusion

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 anad 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.

6. References

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

ii. *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.*

iii. *Y. Desmedt. Threshold cryptography. European Transactions on Telecommunications, 5(4):449-457,July-August 1994.*

iv. *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.

v. 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.

vi. 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.

vii. 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.

viii. M. J. Fischer, N. A. Lynch, and M. S. Peterson. Impossibility of distributed consensus with one faultyprocessor. Journal of the ACM, 32(2):374-382, April 1985.

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

x. 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 Barbara, CA USA, August 17-21, 1997, Proceedings, volume 1294 of Lecture Notes in Computer Science, pages 440-454. Springer, 1997.

xi. M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson. The digital distributed systems security architecture. 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).

xii. 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.

xiii. 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.

xiv. L. Gong. Increasing availability and security of an authentication service. IEEE Journal on SelectedAreas in Communications, 11(5):657-662, June 1993.

xv. Z. J. Haas and B. Liang. Ad hoc mobility management using quorum systems. IEEE/ACM Transactionson Networking, 1999.

xvi. Z. J. Haas and M. Perlman. The performance of query control schemes for zone routing protocol. InSIGCOMM'98, June 1998.

xvii. A. A. Hassan, W. E. Stark, and J. E. Hershey. Frequency-hopped spread spectrum in the presence of aflower partial-band jammer. Transactions on Communications, 41(7):1125-1131, July 1993.

xviii. R. Hauser, T. Przygienda, and G. Tsudik. Lowering security overhead in link state routing. ComputerNetworks, 31(8):885-894, April 1999.

xix. A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. Proactive public-key and signature schemes. In Proceedings of the 4th Annual Conference on Computer Communications Security, pages 100-110, Zurich, Switzerland, April 1-4, 1997. ACM.

xx. A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. In D. Coppersmith, editor, Advances in Cryptology—Crypto'95, the 15th Annual International Cryptology Conference, Santa Barbara, CA USA, August 27-31, 1995, Proceedings, volume 963 of Lecture Notes in Computer Science, pages 457-469. Springer, 1995.

xxi. J. Ioannidis, D. Duchamp, and J. M. Gerald Q. IP based protocols for mobile internetworking. ACM SIGCOMM Computer Communication Review (SIGCOMM'91), 21(4):235-245, September 1991.

xxii. S. Jacobs and M. S. Corson. MANET authentication architecture. Internet Draft (draft-jacobs-imep-auth-arch-01.txt), February 1999.