Certificate-based Key Management for Secure Communications in Ad Hoc Networks

Christian Schwingenschlögl and Stephan Eichler[1]

University of Technology in Munich, Institute of Communication Networks, Munich, Germany.
e-mail: {schwinge,eichler}@lkn.ei.tum.de

Abstract: **Security in mobile ad hoc networks has evolved to a very active research field over the past few years. Not only these networks are usually operating in an open environment that does not offer any physical protection, they also have to provide self-organizing, highly distributed security functions in order to stay operational when nodes leave the network and to stay secure in the presence of adversaries and compromised nodes. To make things worse, security mechanisms for ad hoc networks always have to deal with an inherent contradiction: in addition to the highly distributed operation a "small footprint" of these protocols is required as high communication overhead heavily effects the already volatile scalability properties of ad hoc networks. In our paper, we present a new protocol LKN-ASF (LKN Ad Hoc Security Framework), that can be used in infrastructure-supported ad hoc networks. As quite a lot of application scenarios consider at least temporary connectivity from the ad hoc network to the Internet or vice versa, LKN-ASF resolves both problems: secure and performant operation in a highly distributed environment as well as the setup of a public key infrastructure that is a prerequisite to most security solutions proposed so far.**

1 Introduction

Ad hoc networks typically operate in an especially hostile environment. Compared to computers on the Internet, the physical link itself is relatively vulnerable. In addition, the necessary distributed realization of network functions make it easy to mount devastating attacks with relatively little effort. Recently proposed attacks make routing effectively impossible even without having to compromise a single host on the network. The high vulnerability of these networks has been realized and a lot of contributions have been made so far in order to increase security. We give a brief overview about this work in section 2. Mostly, secure routing is considered. As we will see later in section 2, all protocols either have to rely on a public key infrastructure or have to perform some secure key distribution mechanism before operation. In section 4, we present our protocol LKN-ASF (LKN Ad Hoc Security Framework). LKN-ASF can be deployed in ad hoc networks having at least temporary access to gateway nodes. New nodes can be added or removed at any time during operation and strong authentication as well as secure bindings between a node id and its functionality are provided via X.509 certificates. If necessary, certificates can also be revoked during operation, excluding the corresponding node from all further network operations.

[1]This work has been partially supported by the Bayerische Forschungsstiftung and BMW AG within the FORSOFT/SOFTNET project.

2 Secure Communication in Ad hoc Networks - A Review

2.1 Threats and Attacks

Ad hoc networks are especially vulnerable to most kinds of already known attacks. Their distributed nature also enables completely new types of attacks (or makes known attacks much more effective). [7] gives an overview and a classification of these different attacks with a focus on sensor networks. Most of the described attacks, however, can be applied directly or with slight modifications also in ad hoc networks. This section gives a brief summary of the different attack-classes. It is not intended to give a complete listing but to show the principal threats ad hoc networks are facing and that very few malicious nodes can disable normal network operation.

Spoofed, altered, or replayed routing information As normally every node in an ad hoc network acts as router, any participating node (insider) can mount such an attack. Effects include routing loops, attraction or distraction of network traffic, generation of false error message, partitioning of the network, etc.

Selective Forwarding Malicious nodes drop messages instead of forwarding them further along its path. More subtle forms of this attack that make detection very hard if not impossible include the selective dropping of certain packets only. This attack works best if the attacker is directly included in the forwarding path. Malicious nodes can achieve this e.g. using the sinkhole attack or, especially also with routing protocols that utilize multiple routes to the destination.

Sinkhole Attacks This type of attack typically works by making a malicious node looking especially attractive to surrounding nodes with respect to the routing algorithm. The goal is to lure as much traffic as possible from a particular area through a compromised node, creating a sinkhole with the adversary at the center. Despite this is a general attack applicable to a wide range of networks, it is especially effective in networks with a special communication pattern like sensor networks or surroundings of Internet gateways.

Sybil Attacks This class of attacks can seriously undermine the effectiveness of diverse fault-tolerant and distributed schemes created to improve performance, reliability and security in ad hoc and peer-to-peer networks. In a Sybil attack, a single node presents multiple identities to other nodes in the network. [3] describes these attacks that can also be applied to location aware systems in more detail.

Wormholes Wormhole attacks [5] typically involve at least two cooperating adversaries[2]. An adversary tunnels messages from one part of the network usually via a low-latency out-of-bound channel to another part of the network where they are replayed. These attacks can, among others, be used to distort routing, create sinkholes and to exploit routing race conditions as well as work even in the presence of authenticated and encrypted routing information.

HELLO flood attacks This type of attack has been introduced in [7]. It is applicable against all protocols that use HELLO messages to form neighborship-relations among the nodes. A malicious node can send, record or replay HELLO-messages into the network with high transmission power and therefore convince every node in the network that he is its neighbor. This attack leaves the network in confusion as most nodes simply send their packets into oblivion.

Acknowledgment spoofing This attack is applicable against protocols that use link layer acknowledgments. Spoofed link layer acknowledgments can convince nodes that a weak link is strong or disabled nodes are alive. Among others, this can be used to mount selective forwarding attacks.

2.2 Countermeasures

A good overview about different threat models and attacks can be found in [7]. Although it is focused on sensor networks, most of the techniques described are applicable directly or with slight modifications in different classes of ad hoc networks. Some traffic patterns described here for sensor networks might also be profitable in some scenarios of highly mobile vehicle-to-vehicle networks[3], introducing the same vulnerabilities also to this class of networks. Some ideas to secure parts of the ad hoc communication are presented, however, the paper concludes that no secure routing protocol exists so far and leaves this as an open problem. Besides cryptography itself, the necessity of careful protocol design is highlighted because the limited applicability of end-to-end security mechanisms.

A *Lightweight Hop-by-Hop Authentication Protocol* for ad hoc networks is proposed in [17]. It uses hop-by-hop authentication for verifying the authenticity of packets transmitted in the network and a one-way key chain (TESLA [12]) for packet authentication and for reducing the overhead for establishing trust among nodes. The authentication of packets helps especially to defend against traffic inserted by external (i.e. not part of the network) malicious nodes as these packets are not routed any more. For bootstrapping, a working PKI is assumed as the last value in the key chain, $K(0)$, has to be signed with the senders private key so that anybody who knows its public key can verify the signature and hence the authenticity of $K(0)$. Furthermore, the receivers are required to be loosely synchronized with the sender - the upper bound of the synchronization error between any two nodes has to be known. TESLA itself includes that receivers cannot authenticate packets immediately on its arrival. A variation allows an immediate authentication, however, requires packet buffering at the sender side.

A security-aware ad hoc routing algorithm is presented and its applicability to AODV is shown in [15]. The goal is to characterize and explicitly represent the trust values and trust relationships associated with ad hoc nodes and to use these values to make routing decisions. Only nodes that provide the required level of security can generate or propagate route requests, updates, or replies. If one or more routes exist that satisfy the required security attributes, the shortest of these routes is found. However, a nontrivial prerequisite is required for this protocol, namely an appropriate key management scheme for ad hoc networks.

In [13], a protocol called "authenticated routing for ad hoc networks (ARAN)" is presented. Also some specific attacks against DSR (Dynamic Source Routing) and AODV (Ad Hoc On Demand Distance Vector) are described. ARAN introduces authentication, message integrity and non-repudiation into the routing protocol - protecting against malicious actions by third parties and peers in certain ad hoc environments. The security of this protocol is based on certificates, a preliminary certification process that guarantees end-to-end authentication has to be performed before ARAN can start working.

A route discovery protocol which guarantees that fabricated, compromised, or replayed route replies would either be rejected or never reach back the querying node is presented in [10]. The protocol is capable of operating without the existence of an on-line certification authority, however, a security association (SA) between the source node and the destination node is assumed. Such trust relationships typically are instantiated via public key cryptography.

In [9], a new proposal for anonymous routing is presented. The approach, that guarantees route anonymity and location privacy, combines broadcast with trapdoor information. This idea is used on top of the well-known MIX design, introduced by Chaum to achieve an anonymous email solution [2]. Using the proposed protocol (ANODR, ANonymous On Demand Routing), an enemy can neither link network members' identities with their locations, nor follow a packet flow to its source and destination. Ad hoc routing is dissociated from the network member's identity. ANODR also ensures there is no single point of compromise. Location privacy of participating nodes is not compromised by a single node intrusion - an on-demand route is traceable only if all forwarding nodes on that route are intruded.

Wormhole attacks and its implications on some routing algorithms are introduced in [6]. Basically, the defense is based on so-called packet leashes. Geographic and temporal leashes are also considered in [6]. Geo-

[2] however, also the special case where only one malicious node tunnels messages between two regular nodes often makes sense

[3] especially in-network processing, aggregation, duplicate elimination, etc.

graphical leashes are used to ensure that the recipient of a packet is within a certain distance from a sender, temporal leashes allow to ensure an upper bound on packet lifetime. The TIK protocol presented in this paper implements temporal leashes. One necessary prerequisite for TIK is that a node can obtain an authenticated key for any other node.

A routing protocol that functions in the presence of byzantine failures is presented in [1]. This includes nodes that drop, modify, or misroute packets in an attempt to disrupt the routing service. Links are assigned different weights according to its stability and an on-demand route discovery protocol is used to find a least weight path to the destination. The protocol consists of the following three phases:
Route discovery with fault avoidance: A least weight path from the source to the destination is found using flooding and a faulty link weight list.
Byzantine fault detection: Faulty links on the path from source to destination are detected. Be n the length of the path, the technique proposed in [1] finds a faulty link after log n faults have occurred *Link weight management:* Links are penalized with high weights if faults occur and are rehabilitated over time. This information is used in the route discovery phase to avoid faulty paths.

Pairwise shared keys that are established on-demand are required by the protocol. A PKI, either distributed or Certificate Authority (CA) based is assumed for this purpose.

Ariadne, a secure on-demand routing protocol for ad hoc networks is presented in [4]. It prevents attackers or compromised nodes[4] from tampering with uncompromised routes consisting of uncompromised nodes and also prevents some types of Denial-of-Service attacks. The protocol only uses highly efficient symmetric cryptographic primitives, a simulation-based performance evaluation is given that compares Ariadne with two versions of DSR: the standard DSR protocol and, for fairness reasons, a version of DSR with all optimizations not present in Ariadne disabled. Different types of authentication mechanisms supported by Ariadne require different setup preparations:
Pairwise shared secret keys: A mechanism has to be in place to set up the necessary $n(n+1)/2$ keys in a network with n nodes.
TESLA [12]: Shared secret keys between communicating nodes and one authentic public TESLA key for each node have to be set up.
Digital Signatures: Authentic public keys, one for each node have to be distributed.

2.3 Summary Contermeasures

The countermeasures proposed above help to defend against specific threats against ad hoc networks. In practical implementations, usually some sort of trust distribution has to be done before the network can be used securely. The main problems of these protocols operating stand-alone include either the scalability (no new nodes can be included in the network during operation) or the vulnerability against certain attacks (e.g. sybil attacks against threshold schemes). To the best of our knowledge there seems to be no way yet to get both, scalability and security in a fully distributed stand-alone ad hoc network.

With LKN-ASF , we realized a scalable, certificate-based security framework to be used in any ad hoc network with at least temporary access to gateways (see 3). We have shown the possibility to combine LKN-ASF with the AODV routing protocol and present a related performance evaluation in section 5. In addition to that, the LKN-ASF framework can be used to provide an infrastructure to implement most of the countermeasures found in the literature in a scalable way.

3 Infrastructure Supported Ad hoc Networks

Envisioned applications for Ad hoc networks typically not only consider completely stand-alone Ad hoc operation. More often, at least temporary connectivity to Internet gateways [14] or a user-transparent combination of different network interfaces [8] are considered to combine the advantages of different mobile network technologies. This strategy allows to provide the user with his familiar network experience (e.g. frequently visited web-pages, email, personal profiles, etc.). This concept also allows the usage of scalable and efficient security protocols as shown in our paper. A schematic overview of the assumed network topology is depicted in figure 1a.

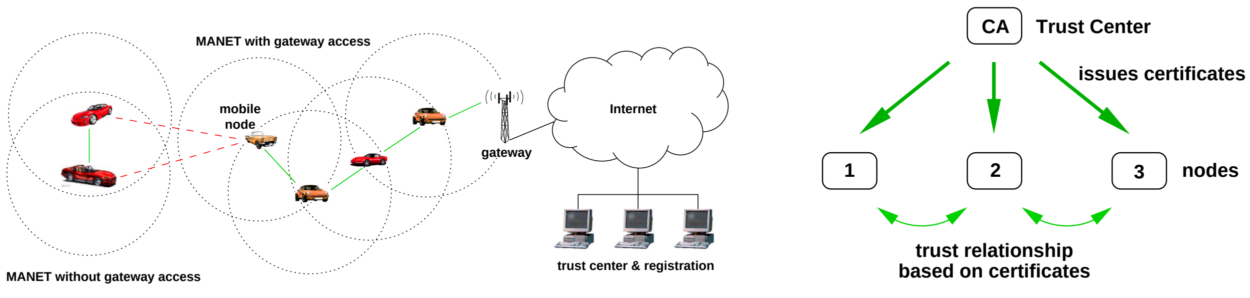
4 LKN-ASF : LKN Ad Hoc Security Framework

LKN-ASF is designed for infrastructure supported ad hoc networks and relies on the access to gateway nodes at least from time to time. After the protocol description in this section, the performance test results are presented in section 5. For the definition of the protocol the network infrastructure shown in figure 1a was used as a starting point. Hence, the nodes of the ad hoc network have access to gateway nodes from time to time. This is an important characteristic of the ad hoc environment to be able to use the protocol with all its features.

4.1 Trust Center

A public key infrastructure (PKI) with a trust center – the certification authority CA – is used to introduce trust within the network. The basic setup of the PKI is shown in figure 1b. To be able to communicate a node has to be registered at the trust center. The node obtains a certificate for its public key during the registration process. The certificate is signed with the key of the trust center. The CA has to check if node and key belong together and if the node is trustworthy before issuing the signed certificate. Every subscriber within the network knows the public key of the trust center and can check the validity of any public key certificate issued by the trust center. Therefore, any two nodes can exchange and validate their public keys without having access to any other node or gateway. If the certificates are valid the nodes can trust each other and establish a secure connection.

[4]the protocol provides security against one compromised node and arbitrary active attackers



(a) Full network configuration

(b) Public Key Infrastructure

Figure 1: LKN-ASF System Overview and Trust Configuration

In figure 1b the trust center consists of the CA only. In large networks a single CA would be overloaded with the requests from all nodes. Therefore, registration authorities (RA) are introduced. The RAs handle requests from the nodes for the CA. The identity verification of the subscriber and the check whether this node really created the key is done by the RA. The CA only has to issue the certificates, thus it is not overloaded anymore. Therefore, the trust center consists of the CA and one or several RAs depending on the size of the network. An overview about different PKI trust models can be found in [11].

4.2 Certificates

Our protocol is used to exchange certificates. These certificates are used to verify the trustability of communication partners within the network. Every node needs at least one public key certificate to be able to communicate within the network.

Public Key Certificates Public key encryption is used to provide security mechanisms to the nodes of a network using the protocol. Hence, every participating node needs a private and public keypair. The trust center issues a public key certificate to every node after the association of node and key has been validated. Within the Internet the X.509 standard for certificates exists. This standard is used for the certificates in LKN-ASF to support common protocols also relying on the X.509 standard.

Attribute Certificates Besides the public key certificates also attribute certificates can be used with LKN-ASF . These special certificates can only be used in combination with a valid public key certificate and provide additional information about the subscriber. This feature is especially important in Ad hoc networks as the sole identity of a node is not always significant. Attribute certificates can e.g. be used to certify that a node is a valid gateway. Hence, it is harder for untrustworthy or corrupt nodes to act as gateway and run related attacks. In addition, these certificates can be used to define user groups and therefore enforce access rights or related services within the network.

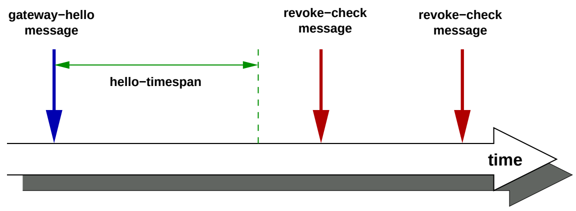


Figure 2: Demanding revocation messages

4.3 Certificate Caching

In order to work efficiently in highly mobile networks (e.g. vehicle-to-vehicle communication), LKN-ASF uses caching of exchanged certificates. Since mobile nodes have a limited memory capacity only a certain number of certificates can be cached. Hence, a previously exchanged certificate can be reused as long as it is still saved in the certificate cache. The size of the cache can influence the performance of the protocol, especially if a number of connections have to be handled simultaneously. Our implementation currently uses simple LRU (least recently used) caching of certificates, however, we are planning to conduct more simulations to examine the effects of different caching strategies.

4.4 Certificate Revocation

The security of a PKI can only be maintained if compromised nodes can be excluded. Hence, it must be possible to revoke certificates owned by formerly trustworthy nodes. This certificate revocation is done by the CA which issues a revocation message. A mobile ad hoc network is not fully connected at all times. Therefore, revocation messages can't be reliably sent to all nodes at the time of creation. LKN-ASF has to ensure the reliable and timely distribution of the revocation messages. Therefore, revocation messages have an individual packet ID and contain the number of previously issued revocation messages. Using this information nodes receiving a revocation message can determine if messages are missing and demand these messages from either neighboring nodes or a gateway.

As long as a node has access to a gateway no checks on missed revocation messages are necessary. In figure 2 the demand process for missed revocation messages is shown. Only if no gateway-hello messages are received anymore the revoke-check messages are broadcast to neighboring nodes to demand missed messages. If a node reestablishes a connection to a gateway it demands all missed revocation messages at once.

4.5 Certificate Renewal

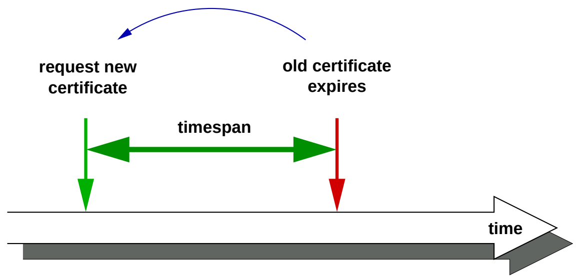


Figure 3: Process of certificate renewal

For security reasons it is not recommendable to issue certificates without expiration time. The validity of a public key should be limited to a certain time period to decrease the possibility that a key is compromised by an attacker. Hence, the certificates issued within the network have to be renewed from time to time. Depending on the size of the encryption keys the timespan should be chosen between several months and several years. A new certificate can only be issued by the trust center. Hence, the nodes can not request a new certificate at any time, a connection to a gateway is required. At the time of expiration a connection to a gateway might not exist, leaving the node without a valid certificate. Therefore, the renewal process has to be started within a reasonable timeframe before the expiration of the current certificate. Otherwise the node has no valid certificate, thus it can no longer communicate within the network.

In figure 3 the certificate renewal process is shown. The timespan depends on the frequency of gateway contacts as well as on the timespan of validity. A node having gateway access nearly at all times can choose a short renewal timespan.

5 Performance Evaluation

In the following section the results of the protocol evaluation simulations are presented. The security protocol was implemented with SDL (Specification and Description Language) to validate its functionality as well as within GloMoSim (Global Mobile Simulation Systems Library) for performance evaluation. GloMoSim has been developed at the University of California and was first introduced in [16].

5.1 Implementation in GloMoSim

LKN-ASF was implemented in a way that allows it to be used together with all other protocols presently available in GloMoSim. For routing performance evaluation, we used it in combination with the AODV (Ad Hoc On Demand Distance Vector) routing protocol. This made it possible to implement a secure handshake which is used to exchange certificates and ensure a secure connection that is set up in the RREQ phase. Valid routes are only found if a secure connection exists between source and destination node. Since LKN-ASF does not rely on protocol functions of GloMoSim, it can easily be used together with other routing protocols or even be integrated into the application layer.

For the gateway discovery, a simple realization has been chosen. The gateway nodes send out hello-messages with a certain frequency which can be defined in the configuration file of the simulator. A node receiving a hello-message assumes gateway access. In the time interval defined in the configuration file at least one hello-message has to be received. Otherwise the gateway access is no longer valid. This time interval can be set as a multiple of the hello-message frequency. Other gateway discovery methods can be found in the literature (e.g. [14] and can be integrated easily in the existing simulation environment.

5.2 Simulation Settings

The LKN-ASF implementation in GloMoSim allows to change several parameters like the time interval for gateway hello-messages, the cache size, expiration time of certificates as well as the validity of certain certificates. Further it is possible to define revoke events for certificates. Another parameter, mainly usefull for debugging, can be set to prevent that invalid GloMoSim packets are discarded.

During the different simulations some parameters have not been altered to get comparable results. The parameter defining an invalid certificate has been tested but has not been used in the simulation results presented. The cache size was set to five certificates in all simulations except in the scenario with 200 nodes. In this large scenario the cache size was set to fifteen certificates. The time interval used to check gateway access was set to three times the frequency of the gateway-hello messages which was set to 30 seconds. The check time for new revocation messages was set to five minutes.

For the first simulations (basic protocol functionality, communication between node and gateway, cache size variations, network load) the simulation area has been set to 1500x1500m, transmission range 220m and manual node-placement has been done manually to get a connected setting. No node mobility has been simulated. For the ”high node density” scenario, the simulation area has been set to 2700x1500m and 200 nodes (range 220m) have been placed randomly. Node mobility was simulated using the random waypoint mobility model (maximum speed 50 km/h, pause time 60sec).

5.3 Simulation Results

In this section the simulation results gained with LKN-ASF are presented. Several different simulations have been carried out to evaluate the protocol in different scenarios.

Basic Protocol Functionality One fundamental simulation was the check of functionality. In this simulation two nodes exchange one Megabyte of data. The results can be seen in table 1. Nodes 1 and 2 communicate without LKN-ASF while Nodes 3 and 4 communicate with LKN-ASF activated. The result shows that the protocol

Simulationdata	Node 1	Node 2	Node 3	Node 4
Begin of transmission	200.005 s	200.006 s	200.015 s	200.017 s
End of transmission	208.490 s	208.615 s	208.587 s	208.720 s
Duration	8.485 s	8.609 s	8.572 s	8.703 s
Throughput	942780 bps	929273 bps	933281 bps	919137 bps
Packets sent	1961	1077	1974	1086
Packets received	1077	1961	1086	1974
Packet overhead	–	–	13	13

Table 1: Comparison of normal communication and secure communication

works and does not affect the data transmission in a significant way. The effects on throughput and duration of the transmission are within the expected range.

The values shown in table 1 are valid for a direct data transmission. For multihop transmissions the results have been accordingly. In some circumstances, the secure data transmission did even show better performance as its insecure counterpart because more stable routes have been found during the somewhat longer connection setup time.

Communication Between Node And Gateway The usage of attribute certificates is possible with no significant additional cost. Simulations to check this included communication between a node and a gateway and showed results basically identical to the results presented in section 5.3. The only difference is that in addition to the public key certificate one or more attribute certificates have to be exchanged. This only slightly affects the duration of the connection, the values are comparable to the ones in table 1.

	Node 1	Node 2	Node 3	Node 4	Node 5
Cache size $\geq$ 5					
Connection establishment	18.554 s	31.403 s	31.176 s	18.664 s	18.669 s
Connection clearing	49.477 s	68.700 s	67.620 s	50.726 s	50.204 s
Duration	30.893 s	37.297 s	36.444 s	32.062 s	31.535 s
Cache size 4					
Connection establishment	18.554 s	31.403 s	33.613 s	18.664 s	18.669 s
Connection clearing	44.214 s	199.237 s	63.907 s	44.596 s	44.582 s
Duration	25.660 s	167.834 s	30.294 s	25.932 s	25.913 s
Cache size 3					
Connection establishment	18.554 s	—	39.341 s	18.664 s	18.669 s
Connection clearing	164.084 s	—	63.931 s	40.599 s	38.187 s
Duration	145.530 s	—	24.590 s	21.935 s	19.518 s
Cache size 2					
Connection establishment	18.554 s	33.891 s	33.523 s	18.664 s	18.669 s
Connection clearing	172.590 s	63.609 s	178.474 s	35.049 s	∞
Duration	154.036 s	29.718 s	144.951 s	16.385 s	∞
Cache size 1					
Connection establishment	18.554 s	31.027 s	30.726 s	21.354 s	18.663 s
Connection clearing	170.030 s	182.259 s	40.731 s	63.332 s	28.640 s
Duration	151.476 s	151.232 s	10.005 s	41.978 s	9.977 s

Table 2: Connection times from scenarios with changing cache sizes

Cache Size Variations The cache size can have an impact on connection times and duration. In several simulations this impact has been investigated. Five nodes communicate with one particular node at the same time and try to exchange one Megabyte of data. In table 2 the connection times as well as the duration are shown. If the cache size is five or bigger all connections should be processed at the same time. The results in the first line of the table show that all transmissions have a similar duration and are processed almost simultaneously. As expected, as soon as the size of the cache is smaller than the number of simultaneous transmissions the connection times are heavily affected. With a cache size of four,

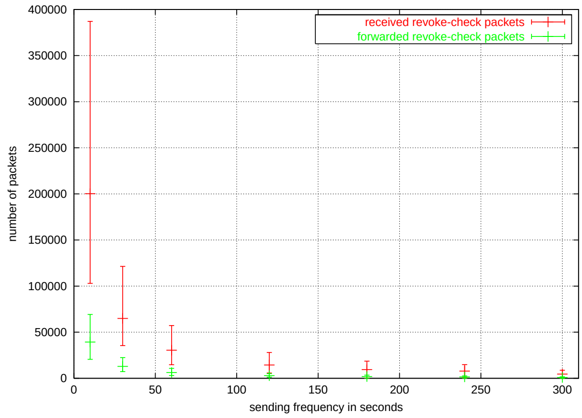


Figure 4: Number of received and forwarded revoke-check messages during 20 minutes against the sending frequency

one transmission has a duration of 167.8s, thus the transmission takes more than five times longer compared to the other four transmissions. A similar outcome results from a cache size of three. In this case, one transmission doesn't start at all. Three transmissions are handled normally and one transmission lasts longer again.

These simulations show that even with a cache size too small to handle all demanded transmissions simultaneously, most data transmissions can be completed successfully. Some of the transmissions need more time[5] than the average but are still completed in our scenario. Hence, communication within the network is possible even if the cache size is too small to hold all necessary certificates simultaneously.

Network Load Due to Hello-Messages and Revoke-Check Messages Using LKN-ASF leads to extra traffic within the network. This extra traffic results from the constantly sent hello- as well as revoke-check messages. Depending on the sending frequency, the number of messages traveling the network can be very high. In figure 4 the influence of a small sending interval on the number of revoke-check messages is shown. The simulations demonstrated that a sending interval smaller than 30 seconds for hello-messages do not significantly improve gateway discovery times but instead result in an unacceptably high network load. Similar results have been obtained for revoke-check messages. An interval smaller than 60 seconds for revoke-check messages is not recommended. However, this should not be critical as in most systems revoke lists are typically distributed only around once a day.

A possible enhancement would be a hybrid system - revoke requests are only sent out at certain times when new revoke lists are expected to be distributed and active notification if new revocation information becomes available in between.

Scenario With High Node Density In this simulation a scenario with mobile nodes and higher node density

[5]This can be critical especially in highly mobile networks.

has been analyzed. 190 nodes and ten gateways were used to simulate 35 data transmissions within 30 minutes of simulation time. The simulation has been run in two configurations - one using LKN-ASF and one using plain AODV without any additional security protocols. Results of these two scenarios have been very similar. The scenario using LKN-ASF showed only slightly worse transmission quality and throughput parameters. At the same time the nodes in this scenario had to handle around 200000 extra messages which affected transmission quality.

6 Conclusion and Outlook

In this paper we have given an overview about attacks especially effective in ad hoc networks and also given a review of currently proposed countermeasures. With LKN-ASF we presented a scalable and efficient framework for key management in this environment that can be used to support the countermeasures presented. First performance evaluations have been presented using LKN-ASF on top of the AODV routing protocol. We are planning to conduct further simulations with different network scenarios concerning number of nodes and mobility- as well as communication patterns. Based on these results, further performance enhancements involving caching mechanisms and route setup will be examined.

REFERENCES

[1] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *ACM Workshop on Wireless Security (WiSe)*, 2002.

[2] D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, (24):84–88, 1981.

[3] John R. Douceur. The Sybil Attack. In *1st International Workshop on Peer-to-Peer Systems (IPTPS)*, March 2002.

[4] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *The 8th ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2002.

[5] Y.-C. Hu, A. Perrig, and D.B. Johnson. Wormhole Detection in Wireless Ad Hoc Networks. Technical report, Rice University, Dept. of Computer Science, June 2002.

[6] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2003.

[7] Chris Karlof and David Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[8] Wolfgang Kellerer, Christian Bettstetter, Christian Schwingenschlögl, Peter Sties, Karl-Ernst-Steinberg, and Hans-Jörg Vögel. (Auto)Mobile Communication in a Heterogeneous and Converged World. *IEEE Personal Communications Magazine*, 8(6):41–47, December 2001.

[9] Jiejun Kong, Xiaoyan Hong, and Mario Gerla. An ANonymous On Demand Routing Protocol with Untraceable Routes for Mobile Ad-hoc Networks. Technical Report TR-030020, UCLA Computer Science Department, April 2003.

[10] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure Routing for Mobile Ad hoc Networks. In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, January 2002.

[11] Radia Perlman. An Overview of PKI Trust Models. *IEEE Network*, pages 38–43, December 1999.

[12] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, May 2000.

[13] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks. In *Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP)*, November 2002.

[14] Jin Xi and Christian Bettstetter. Wireless Multi-Hop Internet Access: Gateway Discovery, Routing, and Addressing. In *Proceedings of International Conference on Third Generation Wireless and Beyond (3Gwireless)*, May 2002.

[15] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-Aware Ad-Hoc Routing for Wireless Networks. Technical Report UIUCDCS-R-2001-2241, UILU-ENG-2001-1748, Department of Computer Science, University of Illinois at Urbana-Champaign, August 2001.

[16] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.

[17] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia. LHAP: A Lightweight Hop-by-Hop Authentication Protocol For Ad-Hoc Networks. Technical report, Center for Secure Information Systems, George Manson University, Fairfax, VA and Department of Information and Computer Science, Univ. of California at Irvine, 2003.