Security of Mobile ad hoc Networks

Roland Gémesi*  Balázs Ivády*  László Zömbik#
gemesi@alpha.ttt.bme.hu  ivady@alpha.ttt.bme.hu  laszlo.zombik@eth.ericsson.se
* Department of Telecommunications and Telematics, BUTE
# Ericsson Hungary, ResearchLab

1. Introduction

In contrast with traditional networks, ad hoc networks do not require any previously built infrastructure, they are distributed and fully self-organized systems. In such a network there should not need special roles for control and management.

Mobile ad hoc networks have the following special attributes. Having dynamic topology means that the architecture of the network is not static, so routes can be valid just for a short time. Moving or disappearing nodes should not disturb the operation of the services. Usually nodes are small handheld devices with a limited capacity of resources (like battery power, CPU or memory). Wireless links usually have much lower bandwidth and these resources are often shared and limited. Moreover links and devices are more vulnerable than in fixed, wired networks.

Additional threats exist for mobile ad hoc environments due to their distributed conception. In the rest of this paper we will examine these networks from a security point of view, and further previously presented ideas will be introduced to make these systems secure.

In section 2 we introduce routing types of ad hoc networks and we show two present protocols. Security goals and different types of threats are discussed in section 3. Section 4 gives a cryptographic overview about the mechanisms needed in ad hoc networks. Security issues of routing mechanisms are discussed in section 5 and possible solutions are introduced to achieve a secure solution.

2. Routing mechanisms in ad hoc networks

In a multi-hop environment packets have to be routed to find their destinations; they should be transferred through a correct route. In a regular (non-ad hoc) network dedicated points (gateways, servers) have information about the architecture of the network, so they can determine which direction the packet should be forwarded. Ad hoc networks have no fixed infrastructure, so there is no centralised knowledge or role. All nodes therefore should participate in the routing process; this should be distributed system from routing and other signalling point of view.

2.1. Types of routing

There are two main groups of routing protocols in the mobile area: pro-active and reactive.

In a pro-active protocol, routes are constantly being tracked and stored. The main advantage of these protocols is that the possible routes are continuously known. However, these can cause a high quantity of data to be stored and a lot of packages have to be transmitted through the network. On the contrary, reactive protocols only try to find a route, when it is needed. This may occur higher delay at the beginning of
the communication, because routes have to be discovered at this time. These are also known as on-demand protocols. Hybrid protocols combine the above two methods.

In the next two sections two reactive protocols will be introduced. [bevprot]

2.2. Dynamic Source Routing (DSR)

In the Dynamic Source Routing (DSR) the sender strictly determines the full path of packets. The header of packets contain the complete sequence of hops, from which intermediate nodes will be able to figure out the address of the next hop.

Sender maintains a route cache, and it always checks first their database in order to find a path towards the destination. If a route does not exist, sender initiates a route request. This is a broadcast message that spreads through the network and locates the target host. If the destination is reachable, it will send back an answer message that will include the list of intermediate nodes of the path. If the path that the sender tries to use is broken, then an error message is being generated as the detection of failure. Then the sender should try to request a route again. [perfan]

In networks with low mobility this protocol can be quite effective, because the entries in the cache tables will stay usable. A disadvantage is that all packets have to contain all the intermediate node addresses in the route, which can mean significant overhead in the signalling and in the sender’s memory.

2.3. Ad hoc On-Demand Distance Vector Routing (AODV)

AODV is also a reactive protocol, which uses a broadcast route discovery mechanism similar to DSR, but instead of source routing, AODV relies on the dynamically established route table entries at intermediate nodes. Endpoints not required to know the whole path, since the intermediate nodes know only the correct direction and as a result they form the actual route. Loop freedom is reached with the application of sequence numbers.

When a node starts to communicate, it initiates a Path Discovery process, which is a broadcasted route request (RREQ) message containing the destination’s and originator’s address. Each node receiving a RREQ message sets up a reverse route back to the source, increments the hop-count and rebroadcasts the request. The destination replies to the first incoming RREQ with a route reply (RREP) message on that way to the originator. This reply message runs back through the selected reverse route and sets the forward route in the nodes. (Figure 1.) Each route entries have a specific lifetime, consequently unused reverse route entries will be deleted. When there is no traffic in a link, nodes can perform local connectivity management. This means broadcasting Hello messages only to the neighbours to check connectivity.

When a link error is detected a route error (RERR) message is generated with the list of unreachable destinations and sent back to the sources. In AODV it is possible to repair a link locally, which is called local repair. If the local repair process succeeds, the endpoints will not be informed about the action.

AODV is a scalable protocol, it offers low processing and memory overhead, low network utilization and it can determine and maintain routes effectively. This protocol can handle networks with higher mobility. [aodvdraft]
3. Security issues of ad hoc networks

3.1. Security goals

Communication networks, which have proper security provides the following security services for reliable and secure information transport.

First of all availability provides that the network and services remain accessible and usable despite of different attacks or malfunctions. Authentication guarantees a node to ensure the source of data or identity of peer communicates with. Without this, a node could not make sure of the origin of the other side. Confidentiality means the secrecy of data, so that information never discloses to any unauthorized entities. Neither user data nor system signalling should be allowed for eavesdroppers or for not permitted parties. Integrity ensures that the information is correctly transferred, it is unaltered and errorless. Such modifications can be made by natural conditions or malicious attacks. Non-repudiation is a protection against false denial of involvement in a communication. Without this, nobody should take responsibility for his acts. Several cryptographic algorithms needs key management service to administer keys. Other services might be needed, like as authorization, which can grant a system entity to access a system resource.

3.2. Threats in mobile ad hoc environments

Wired networks with fixed infrastructure have many security threats, however mobile Ad hoc networks reach further questions. In a distributed network, without any infrastructure (fixed routers, centralised points, static neighbours) communicating peers have to rely on the whole network, therefore proper security hardly can be established. The most important questions are highlighted in this subsection.

Denial of Service Attacks

Denial of Service Attacks (DoS) block service functioning. These type of attacks decrease service availability, prevent of authorised access to a system or generate high delays.

The adversary could scramble the physical medium, which causes interference and large number of information loss. This can make the communication channel almost unusable. Moreover it is hard to identify the data of the attacker from the natural noise of the medium. Radio channel is a shared and limited resource of mobile communication, so everyone should keep the access rules. Flooding the medium with bad messages or neglecting the access rules can prevent nodes to access the medium. If a node play destructively communication become impossible.

Mobile devices have relatively small capacity of CPU, memory and battery. Sending large amount of CPU consuming queries overloads the processor, which can prevent other operations for a while. Battery usage also should be optimised in order to increase the availability time of a service. Therefore in unused time periods processor and radio receivers are always attempted to change to power sleeping (sleep) mode. However, incoming requests must be still processed. A malicious node performing an energy exhaustion attack can exploit this fact.

Routing information is created collectively, but malicious nodes could broadcast wrong routes. They could send the others corrupt and outdated information or even could alter message paths. They can cause much longer paths or can overload nodes.
Other possible malicious behaviour is node selfishness, taking advantages of other nodes’ collaboration but not taking part in. These nodes may be tempted to not relay packets in order to save their own resources. (e.g. Battery) [questfor]

**Impersonation**

A malicious node can simulate that it is the node the other party wanted to communicate with. However using authentication process, a verification step can reveal the identification of the communicating parties. This step is quite difficult because of the lack of any centralized entity that stores the needed information about peers.

Other type of impersonation is man in the middle attack, when a malicious node become involved in the route between the two peers.

**Confidentiality Violation**

By the usage of radio medium, the physical area of the shared medium is extended therefore eavesdropping is available. Passive eavesdropping is a special kind of attack where a malicious node only listens on the shared medium to catch others’ communication.

Active eavesdroppers can intercept and selectively modify data. It can be man in the middle attack or routing manipulation based attack.

Even communication is encrypted code breaking attacks are possible, mainly when weak encryption algorithms are used.

**Message alteration**

Malicious node can influence the message has been sent to the destination. It could insert new message to confuse the nodes or alter the meaning of the information. The removal of important information is also dangerous, this could be reached for example by route manipulation.

**Message repudiation**

It means denial by a system entity that it was involved in a communication. If message can be denied than later an investigation cannot decide that the node itself or other malicious entity behaved as logged.

**Anonymity violation**

Malicious nodes involved in the communication process are able to collect some information about the neighbouring nodes due to the access medium is shared and identification is readable. Therefore identification or localisation of the target host became easy.

When nodes have authentication credentials, they can identify themselves, so it is possible to collect information about the other nodes. These identifications can be disclosed to eavesdroppers too.

**Physical tampering**

If an unauthorized person gets access to the equipment, hardware or software modification could be made. An adversary can modify the operation of device, can install viruses, Trojans, etc.

**Lack of centralised trusted element**

In a mobile ad hoc environment it is assumed that there is no trusted centralised secure network element. The advantage of this is that the system is not vulnerable when a node with centralised functionality brakes down or became compromised. However this raises a lot of problems, because authentication procedure in distributed system cannot easily handled since there is no fully trusted network element.

**Routing manipulation**
Ad hoc networks rely on the routing information, which are solicited by the nodes themselves. An adversary could propagate false information to increase delays or to route the packets through a specific node.

4. Security mechanisms

Traditional security mechanisms, such as authentication, digital signature and encryption can provide good security properties, but most of them require key management service. This is often achieved using a centralised entity. In an ad hoc network we cannot rely on a node with centralised responsibility, these tasks should be distributed. Moreover this architecture should consist redundancy, because nodes disappear or they lost connection and even they can be compromised.

4.1. Public Key Infrastructure (PKI)

In a public key infrastructure (PKI) each node has a public/private key pair. A message encrypted with the public key is only decodeable with the corresponding secret key. From the public key the corresponding private key cannot be derived.

The goal of Certificate Authority (CA) is to sign certificates so to bind public keys to nodes. CA should stay on-line to reflect the current bindings. It should track changes and it can revoke certificates if the node is no longer trusted.

PKI is a main element of certificate-based authentication, because it provides the CA to ensure the identity of the nodes.

4.1.1. Threshold cryptography

A solution for certificate authority in ad hoc environment is distribution of trust using threshold cryptography. A (n, t+1) threshold cryptography scheme allows n parties to share the ability to perform a cryptographic operation, so that any t+1 parties can perform this operation jointly, whereas it is infeasible for at most t parties to do so, even by collusion.

In this case n servers of the key management service share the ability to sign certificates. This scheme can tolerate t compromised servers by dividing the private key k of the service into n shares (s1, s2, ..., sn) assigning one share to each server. We call this composition an (n, t+1) sharing of k. After this each server can generate a part of the signature, which are combined. This certificate still adequate even if t parties are compromised.

Besides the advantage of threshold signature, this kind of key management service is easily able to refresh the participants’ shares. This is very effective against mobile adversaries that compromises a server and then moves on the next victim. This kind of attacker can take several servers, but by the end of periodical share refresh the obtained node loses the ability of service. During a share refresh a new threshold configuration can be created, which is a good occasion to adapt network changes. Moreover share refreshing is not a complicated operation, so once created share configurations can be adapted for a long time. When too many participants are compromised, an entirely new share should be created.

Threshold cryptography has a serious problem, it assumes synchrony of nodes. In reality it is not the case, because a node could disconnect for a period or a denial of service attack could slow it down, while the others would do a share refresh. When it tries to attach to the service, it will not be able to participate the communication because a new threshold configuration would be applied. [secadhoc]

The establishment of distributed service causes another issue. When only few nodes exist in the network, it is undecided which one is authorized to initiate the shared
authority. Moreover, once network splits into distinct parts, these parts may operate as different authorities.

### 4.1.2. Self organizing Public key infrastructure

Trusted third parties are a very troublesome part of certificate-based authentication. Therefore it would be beneficial to avoid them and build a self-organized structure. There are some certificate-based systems (e.g. Pretty Good Privacy (PGP)), in which certificates are issued by the users themselves and distribution of certificates is managed by a self-organized structure.

Each user maintains a local certificate repository that contains a limited number of certificates. When user \( u \) wants to obtain the public key of user \( v \), they merge their local certificate repositories and \( u \) tries to find an appropriate certificate chain from \( u \) to \( v \) in the merged repository (Figure 2). For the construction of the local repositories several algorithms exists to provide that almost any pair of users can find certificate chains, even if the size of the local repositories is small compared to the total number of users of the network.

This approach realizes a self-organized certificate system.

### 4.1.2. ID-Based cryptography

The basic idea of identity-based cryptoschemes is that information used for identification (node address, user name, etc.) could act as a public key, so that CA is no longer needed to bind keys and identities. However the strengthen of PKI is that the secret and public keys cannot be derived from each other. In this case, only a central entity is able to derive the secret pair of a public key (identity). This operation is needed only once, at the registration of users. After this, participants are able to authenticate each other in a non-interactive manner, without communicating with a third party.

In a real scenario compromised nodes have to reregister themselves, so the registering central entity should stay available. Serious problem is that this central entity will know all the registered secret keys too, so it became a vulnerable part of the system. Moreover the registration process needs huge resources and lasts a lot of time. [questfor]

### 4.2. Common key architecture

A group of nodes that previously identified each other wants to form a secure part of the network; therefore they should establish an encrypted session. Symmetric key architectures are usually used for this purpose, because of the relatively low processing overhead of the encoding and decoding mechanisms. This requires the participants to have common (shared) secret key, which they use for both encrypting...
and decrypting. By the application of symmetric keys, secure broadcast and multicast communication is also possible. Several key establishment protocols exist.

### 4.2.1. Diffie-Hellman (DH) key exchange

The Diffie-Hellman (DH) key-exchange algorithm is suitable to securely share a common secret between two parties. Securely means that a third party who can also hear their communication (e.g. eavesdropper) must not be able to generate that key.

The two parties (A and B) agree on a cyclic finite group G of order q, and a generator \( \alpha \) for the group. After that, both of them choose a secret exponent e. A computes its public key \( \alpha^A \) in group G and sends it to B. Likewise, B sends \( \alpha^B \) to A. Now, both parties know their own exponent and can raise other parties’ public key to it, producing the shared key \( \alpha^{AB} \). But an eavesdropper, who has not heard the exponents, cannot figure out the key.

In a group, more than two participants should be able to agree on a common key. Some solutions for extending the DH key exchange [keyest] to multiparty key agreement is introduced.

### 4.2.2. GDH.2

GDH is the acronym of generalized Diffie-Hellman. Nodes form a chain and the first member starts a DH like message to the next one, who extends it with additional information and forwards. This process goes till the last node. This node is now able to generate the common key, and also key pieces that needed by each participant. These pieces are all broadcasted, and a third party (eavesdropper) cannot generate the common secret from these. Finally the common key is successfully established. (Figure 3.)

![Figure 3. GDH.2 common key generation](image)

The last (broadcaster) node should play a central role, which is not advantageous in an ad hoc environment. Moreover the relatively large amount of broadcasted data results significant overhead.

### 4.2.3. Hypercube and Octopus

The main idea for establish a common secure key between more than two participants in Hypercube is to form pairs of nodes and establish the common secret using DH protocol. Then form pairs of pairs and perform key exchange by all participants, and so on. (figure 4.) The problem is that the number of members must be \( 2^n \) which cannot be supposed.

An extension to this problem is the Octopus protocol. This forms a hypercube kernel of the network and extends it with tentacles. First these tentacles do a DH key exchange with their respective central nodes. After that the central nodes perform the Hypercube key exchange and
inform the tentacle nodes about the new key. However Hypercube kernel plays a central role and the involvement of a new participant is difficult.

4.3. GKMP (Group Key Management Protocol)

This protocol provides management functions of symmetric keys for a set of nodes. Key generation concept used by GKMP is a cooperative generation between two entities, such as Diffie-Hellman key exchange. After generating the group key, GKMP distributes it to qualified GKMP entities. It also allows new members to join, member deletion and rekeying the group. GKMP provides a peer-to-peer review process; entities have permission certificates (PC) as part of the keying process. Therefore each entity can verify the permissions of any other GKMP entity but can modify none. GKMP supports compromise recovery and the list of compromised nodes is disseminated through the network and Compromise Recovery List (CRL) is stored in each host.

This protocol attempts to delegate as many functions to the group as possible, so it tries not to rely on any centralized entity. However some functions, such as granting privileges, creating and distributing key, creating group and rekey messages should still be centralized. [gkmp]

We can conclude, that achieving a secure communication in the sense of confidentiality is not a difficult problem, because there are methods to establish common secret (DH, GDH.2, Hypercube, Octopus) among participants. This secret can be used in a common key architecture, so communication will not be available for non-authorized parties.

A much more serious problem is the proper authentication of the participants. In ad hoc environment, we cannot rely on a centralized Certificate Authority that could ensure certificates. We could see some possible solutions (Threshold Cryptography, Self organizing PKI, ID-Based PKI), although those are effective only in special circumstances.

5. Security issues in routing mechanisms

Security properties of ad hoc networks depends on the used routing mechanism, although in present routing protocols security is not included. An attacker may become involved in packet forwarding, moreover it is sufficient to run an intermediate node close and force a DoS attack using the shared radio channel. The scramble of the attacker is mostly indistinguishable from the natural loss caused by the noise of the channel. The goal is to establish a reliable channel, where trustiness can be a QoS parameter of the route.

Consequently a routing protocol should be able to guarantee some level of security of a path and to force packets to travel through this route. Using distributed mechanisms or having a node with additional routing information (e.g. at the source) the path can be determined.
5.1. Onion routing

In onion routing, between the peers the messages traversed securely. This is realized with a public key system. The sender collects the public keys of the intermediate nodes and encrypts the sent message with all of them. Each intermediate node decrypts it with its secret key, removes the outermost lock. At the destination the encryption can disappear only if each intermediate node has used its secret key for decoding (Figure 5).

This kind of routing security is applicable mostly when the sender knows the path. (e.g. DSR).

5.2. Security-aware routing

Traditionally, routing protocols try to find an optimal route. The metric for optimality is distance usually measured in hops. To improve the quality of security of an ad hoc route “Security Aware ad-hoc Routing” (SAR) has been presented, which incorporates security levels of nodes.

Different techniques exist to measure or specify the quality of security of a route. Each host has the attributes such as “trust level” and “security level”. These attributes are used by the routing algorithm, and only nodes that provide the required level can participate the routing protocol. However, these levels should be immutable, so that a node with a lower level can neither change it’s own nor the requested level.

In SAR the sender who initiates a route discovery, embeds the needed security attributes into the request. Intermediate nodes forward it only if it has the proper security attributes, otherwise it should be dropped. If the destination receives a request with proper security attributes, consequently an end-to-end path with the required security attributes can be found, and a security-aware route could be established. SAR can be implemented based on any on-demand ad hoc routing protocol with suitable modification. [secaware]

These protocol modifications result in changes the nature of discovered routes. The route discovered by SAR may not be the shortest one in the terms of hop count, although SAR is able to find a route with a quantifiable guarantee of security. If one or more routes that satisfy the required properties exist, SAR will find the shortest (optimal) such route. However SAR may fail even if a network is connected, but the required security attributes cannot be provided.

A problem with SAR is that the specific levels should be authenticated; nodes should not identify their own attributes. We could see that authentication in ad hoc networks is not a simple process. On top of that comprehensible levels reach new threats, because those levels are often related with the importance of nodes.

5.3. Watchdog, Pathrater

Using static parameters of participants is not perfect approach, because they may change over time. (e.g. Compromitted nodes)
Watchdog method examines behaviour of participant continuously. Due to the shared radio channel, each node can listen for the sent packets of their neighbours, consequently a malicious node can be detected easily by its direct neighbour. This procedure does not cause additional network traffic, although it may make a mistake (e.g. Asymmetric link, collusion) or can be mislead (e.g. Directional aerial).

After detecting malicious nodes, another goal is to exclude them from communication. In the Pathrater method each node maintains a rating for every other node it knows. Ratings of nodes on actively used paths are continuously incremented; while decremented on misbehaving paths. With the help of such information it is possible to select a well behaving path.

5.4. Selfishness

Previously examined secure routing procedures tries to exclude misbehaving parties from packet forwarding, although their messages are forwarded without complaint. This type of Selfishness behaviour can be hopeful for example to save battery resources, although in quantities it occurs the network to stop functioning. Forcing cooperation can be achieved by a virtual currency, which is called ‘nuglets’. Participants buy services from each other, while selfish nodes run out of nuglets.

Punishment of untrustworthy nodes should provide that over time only good behaviour pay off. This means that misbehaving nodes should be totally refused by the network.

6. Conclusions

We could see that ad hoc networks has many serious threats. We introduced some possible solutions towards the possibility of secure communication in such a distributed environment.
References: