

Contents

1	Foreword	2
1.1	About the Author and the Document	2
1.2	About the Papers that are presented	2
2	Cluster-Based Security Architecture	3
2.1	Basic Ideas	3
2.2	Threshold Cryptography and Clustering	3
2.3	Conceptual Building Blocks	4
2.3.1	Network-Wide Distributed Certification Authority	4
2.3.2	Intra-Cluster Security	5
2.3.3	Access Control through Authorization Certificates	5
2.4	Detailed Examples	6
2.4.1	Log-On Procedure	6
2.4.2	Merging a Cluster into a Network	7
2.4.3	Merging Two Networks	7
2.4.4	Adaptable Complexity	8
3	IP-Address Handoff	9
3.1	Basic Terms and Ideas	9
3.1.1	Terms and Pre-Requisites	9
3.1.2	Motivation	9
3.1.3	Related Works	10
3.2	Solutions to Broken Routing Fabrics	10
3.3	Solutions to Broken On-Going Communications	10
3.3.1	Assumptions	11
3.3.2	Route Rebuilding	11
3.3.3	Communication Preservation	11
3.3.4	Challenges to Key Management	13
4	Summary	14

1 Foreword

1.1 About the Author and the Document

This document was written by Fabian Meyer for the 'Advanced Topics in Mobile Communications' (AToMiC) seminar at the University of Göttingen, in the summer semester 2004. For more information on the seminar, please contact Dr. Xiaoming Fu or Professor Dieter Hogrefe who are responsible for this seminar.

Fabian Meyer (fmeyer@cs.uni-goettingen.de)

1.2 About the Papers that are presented

This document outlines the ideas presented in two papers from Infocom 2004.

The first paper is called 'A Cluster-Based Security Architecture for Ad Hoc Networks' by M. Bechler, H.-J. Hof, D. Kraft, F. Pählke and L. Wolf.

The second paper is about 'IP Address Handoff in the MANET' and was written by H. Zhou, M.W. Mutka and L.M. Ni.

2 Cluster-Based Security Architecture

2.1 Basic Ideas

The first paper proposes a Cluster-Based Security Architecture for Mobile Ad Hoc Networks (MANET). This approach is chosen because it effectively circumvents a number of problems. Problem number one is that if a central authority exists, it can easily become the target of an attack. Problem number two is that pre-shared keys are virtually impossible in ad-hoc networks, because the nodes are very likely unknown to each other beforehand. A third problem is that encryption without authentication is useless, because you cannot be sure of your communication partners identity.

The paper of Bechler et.al. proposes a de-centralized certification authority (CA) as solution to these problems. Furthermore their approach aims to provide the means for quick adaption to changes in the network, scalability to support a large number of nodes and a fine-grained access control. De-Centralization is accomplished by using clustering and threshold cryptography.

2.2 Threshold Cryptography and Clustering

Threshold cryptography is a way of spreading a mutual secret over a number of different entities. The idea is that a trusted dealer divides a secret 'D' into 'n' parts. These 'n' parts are constructed so that knowledge of 'k' parts ($k \geq n$) allows the reconstruction of the whole secret. This is called a '(k,n) threshold scheme'.

The sharing of the secrets should be verifiable. This is ensured by using the right construction algorithm. This algorithm ensures that each node can verify both, the secret itself, as well as a share of the secret. This is called 'verifiable secret sharing'.

For further security, the approach in the paper of Bechler et.al. uses Proactive-Secret-Sharing (PSS). This means that the secret shares are exchanged periodically without changing the secret itself. This ensures that a malicious node that managed to get a share of the secret cannot pose as the rightful owner of the share for a long time. After the shares are updated, the malicious node does not know which node now has the share it holds. This makes it practically useless for the malicious node.

Clustering means that a MANET is partitioned into several clusters. Each cluster has a distinguished node called cluster head (CH) and gateways (GW) to manage communication with adjacent clusters.

Cluster-Heads perform several tasks for the local network, like authorization for access to resources, admittance into the network, maintaining a list of all nodes and their status in the cluster, maintaining a list of all GWs and the adjacent clusters. But also all CHs in a MANET form a so-called Cluster-Head-Network. This network has all the secret shares and works as the

de-centralized authority.

Gateways can be chosen in two different ways. The first is that the permission to become a gateway has to be granted by the CH. The second is less complicated, but also less secure. In the second way, each node that comes into contact with another cluster simply becomes a GW. The problem of the later is that malicious nodes can easily have traffic transferred through them, giving an easy opportunity for an attack.

The clusters organize and coordinate inter and intra cluster using beacons. Beacons are sent by all CHs and all GWs. While the CH send CH-beacons (CHb) containing the public key of itself and the CH-Network, a list of all nodes in its cluster and their status (guest, member, GW, ...) and information about all the local GWs and adjacent clusters. The GWs send GW-beacons (GWb) which informs all nodes in the cluster of the adjacent clusters it is being GW for. This is actually redundant, since the same information is maintained and broadcasted through the CH. Bechler et.al. nevertheless consider it useful and propose it in their work.

This approach to clustering is independent of the routing protocol that is being used. It can work with routing protocols that provide means for clustering as well as with protocols that do not. When it is used with a protocol that offers clustering some synergy effects arise. One of these effects is secure routing. If the routing protocol provides clustering, it is possible to ensure secure routing by choosing the nodes that may forward network packets. The sender of a packet can set an option that determines if the package may or may not traverse insecure nodes (non-members for example). The only disadvantage is, that each cluster-node needs to keep two routing tables, one for secure routing and one for normal routing.

2.3 Conceptual Building Blocks

The paper distinguishes between three conceptual building blocks. These blocks are:

- Network-Wide Distributed Certification Authority
- Symmetric Encryption for Secure Communication on Intra-Cluster Links
- Access Control through Authorization Certificates (AuthCert)

2.3.1 Network-Wide Distributed Certification Authority

The first block is the network-wide CA. The paper proposes a CA that is distributed over the whole network. The two biggest advantages are that there is no central point that can be attacked and that a distributed CA offers far more availability than a central authority. With a distributed CA it is not necessary to reach one particular node, if one is unreachable another node that is part of the distributed CA can be used.

The distributed CA is built using the network that is formed by all the cluster heads as the CA. Each cluster head has a share of the secret key. This secret key is called the 'network key'. By using different network keys, it is possible to have more than one network in the same area, they just need to use different network keys. If there is more than one network, these networks may be merged but that introduces another set of problems and will be dealt with later.

Each cluster head (CH) has the option to choose a successor if need be. This can be useful if the current CH leaves the area (in case of wireless access, that means decreased availability) or leaves the cluster. If a CH chooses a successor, all current states and the share of the network

key are transferred to the new successor. The old CH notifies the CH-network and the local cluster of the change. Any updates of the network key share will then be sent to the new CH. When a CH fails without choosing a successor first some issues arise. When the failing CH is the only CH of the whole network (which is only true in very small MANETs) it is not a big problem. All nodes lose their states and a new cluster is built. But if there is an existing network, the process is much more complicated. The problems are that the new formed cluster starts as in an untrusted state to the existing network. The new CH and its new cluster has to be re-authenticated and re-accepted into the network they belonged to a short time ago. It can happen that the new CH and the new cluster are not trusted by the CH-network. In that case the new cluster has to be dissolved completely and all the nodes have to try getting into existing clusters of the existing network.

2.3.2 Intra-Cluster Security

Intra-Cluster security is another issue that has to be taken care of. Bechler et.al. propose the use of an encryption using a symmetric key that is known to all members of a cluster. This key is sent to the nodes when they are admitted into the cluster as full members. To make sure that eavesdroppers can not get the key these messages are encrypted using a public/private key encryption between the new node and the CH. All cluster-internal traffic is encrypted with the symmetric key. This provides a minimum level of security as eavesdroppers outside the cluster can not see the source and destination address of caught packets. This approach is meant to be integrable with 802.11 or Bluetooth mechanisms.

2.3.3 Access Control through Authorization Certificates

The third conceptual block is about access control using authorization certificates (AuthCert). The basic idea is that the access to some services or resources are restricted. If a node needs one of the resources or services it has to get an AuthCert from the CA. Some examples for resources and services are gateways, FTP-servers, printers, mail-servers and many others. Only full members of the network can get access to these.

To become a full member a new node first has to join a cluster. It will start as a guest with no rights. In order to become a full member a node has to be authenticated. In the approach of Bechler et.al. a guest node has to gather 'Warrant Certificates'. The certificates are issued by nodes in the cluster that have been given the rights to warrant. The cluster nodes that warrant for a new node have to check the authenticity (or 'trustworthiness') on their own.

The paper does not propose a technical solution how this is done, but gives some examples what could be done. They discern two different approaches. The first approach is to do authentication on technical level, for example the warranting node connects to the guest via cable or IRDA connection to verify its identity. The second approach is authentication outside the technical level. In a conference this could mean that the owners of the two nodes talk to each other for authentication. When vehicular MANETs are used Bechler et.al. propose number plate recognition to verify a node's identity.

Regardless of the way authentication is done, the result is that a guest node gathers a number of warrants. If the number of warrants is sufficient it is allowed into the network. If the number of warrants is more than the minimum the new node may automatically be granted additional rights (like the use of a GW or access to a FTP-server).

When the guest node has been authenticated, it gets its public key signed by the CA (using the

network key) and becomes a full member of the cluster.

The AuthCerts that are necessary to access further resources and services (as mentioned above) can be issued by the controlling entities. The controlling entities can also grant the privilege to grant access to the service to other nodes. If this approach is impractical for a network a simpler method can be used alternatively. One simpler method is a general table that categorizes nodes and grants access to certain service based on a nodes status.

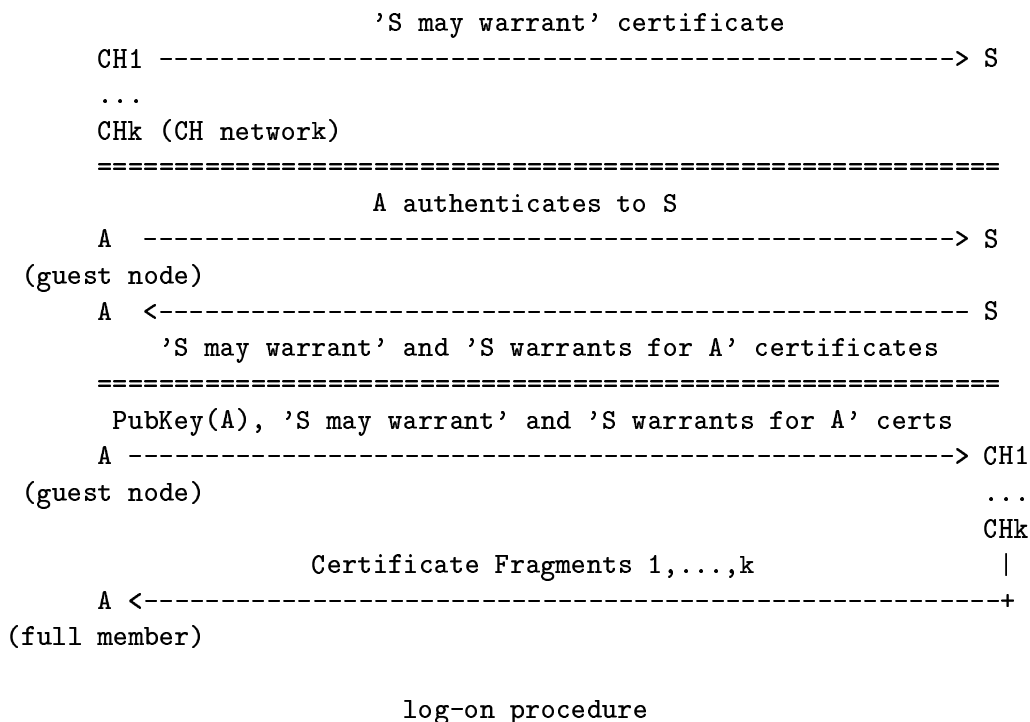
<i>User or provider group</i>	<i>Credential</i>
all nodes	none
all full members	secret symmetric cluster key or certified public node key
specific nodes	authorization certificate
directly trusted node	any of the above or pre-shared key

2.4 Detailed Examples

In this section the following topics are covered:

- Log-On Procedure
- Merging a Cluster into a Network
- Merging Two Networks
- Adaptable Complexity

2.4.1 Log-On Procedure



The figure shows the log-on procedure as it has been described before. At some point the node 'S' is granted the privilege to warrant for other nodes. This is done by the CH-network by issuing an 'S may warrant' certificate. This certificate is sent to node 'S'. The authenticity of this 'warrant certificate' can be verified by every CH because it was signed with the network key.

At some point later a new node wants to be admitted into the cluster. It needs to gather warrants from different nodes. When it has enough warrants it will be admitted into the network. In the figure node 'A' authenticates itself to node 'S'. After the authentication is verified, node 'S' sends two certificates to 'A'. The first certificate is the 'S may warrant' certificate. This is done so that the CH can check if the warranting node really is authorized to do so. Otherwise it would be fairly easy for an attacker to just fake warrants to be admitted into the cluster. The second certificate is the 'S warrants for A' certificate which is issued by 'S' on successful authentication of 'A'.

When 'A' has gathered enough warrants from nodes in the network, it sends its own public key and all the warrant and 'may warrant' certificates to the CH-network. After the CH-network has checked all the certificates it signs the public key of node 'A' with the network key and transmits the secret symmetric key for secure intra-cluster communication to node 'A'. The reason why only fragments are transmitted is that the CH-network is decentralized and as such each node responds on its own. It is not necessary that all CHs reply, but more than one. Each of these CHs send a part of the certificate.

2.4.2 Merging a Cluster into a Network

If a cluster wants to be merged into a different network, the CH of that cluster needs to get warrants from the nodes of the network the cluster wants to join. The procedure is not much different from the joining of a single node into a network. The difference is that the CH of the joining cluster needs to get more warrants so its status is trustworthy enough to be a CH. If it gathered enough warrants it becomes part of the CH-network and receives a share of the network key.

If the CH of the joining cluster can not gather enough warrants it has to pass its CH status to another node of its own cluster that is more trusted in the new network and can be accepted as part of the CH-network.

If no node in the joining cluster can gather enough warrants (meaning is trusted enough), the cluster has to be dissolved and all the nodes have to join existing clusters of that network.

2.4.3 Merging Two Networks

If two networks should be merged it is difficult and costly to do so. The problems arise from the fact that two network keys can not be mixed. That means that one of the two networks has to drop its key and get shares of the other one. All certificates that have been issued with the dropped key will have to be re-issued. Furthermore there will most probably be an adjustment to the (k,n) -threshold scheme necessary. This means that the key that is being kept will have to be parted into more shares so that the new CHs can all get one.

Apart from these problems, there is also the problem how to decide which key should be dropped. The best way is to base the decision on how many certificates have been issued with the two keys. The one with fewer certificates will be dropped to minimize the number of re-issued certificates. Even if this sounds easy now, the decision can be difficult. It might be that both networks issued

the same number of certificates. Or other factors might force the dropping of the key that has issued more certificates. In any case the merge of two previously independent networks is very costly in terms of bandwidth and computing time.

2.4.4 Adaptable Complexity

In order for their approach to be feasible even for small devices, Bechler et.al. designed for an adaptable complexity. The most costly part of their approach is encryption. Like in other areas there is always a trade-off between complexity and security. In this case the proposal is to use different levels of security. Bechler et.al. name four security levels:

1. no encryption
2. secret cluster-key (for intra-cluster traffic)
3. public keys for nodes (directly exchanged)
4. public keys for nodes (using CA)

The decision which security level is used is made per-case. That means that less powerful devices like PDAs can use level 1 or 2 while laptops use level 4 in the same network. The drawback is, that communication is impossible if no consensus about the used security level can be found. If a laptop is configured to accept only level 4 security, a PDA which is only capable of level 2 can not connect to it over the network.

3 IP-Address Handoff

3.1 Basic Terms and Ideas

3.1.1 Terms and Pre-Requisites

The second paper deals with the problem of IP-address handoff on Mobile Ad Hoc Networks (MANET). To eliminate misunderstandings the terminology will be explained in the understanding they are used within the scope the paper.

By MANET a temporary, wireless network of mobile nodes is meant in this context. Further assumptions are that a MANET does not have an infrastructure and is IP-based. This means that nodes have to be configured with a free IP-address before they can receive unicast messages. It can also happen that the IP-address changes during a nodes participation in the MANET. There are many different events that can cause the change of an address. Here are just a few examples:

- merging two parts of the network
- merging two independent MANETs
- merging a MANET with a LAN
- use of a hierarchical addressing scheme with subnets in the MANET

With these prerequisites in mind, we can discuss the motivation and ideas presented in the paper of Zhou et.al.

3.1.2 Motivation

The motivation for this work was to find solutions to two major problems. The first is the problem of broken routing fabrics. Broken routing fabrics cause an overhead in network load for the time when packets are routed falsely and overhead in time for fixing the wrong routing.

The second problem is more complicated to overcome and represents the main focus point of the paper, it is about broken on-going communications. What happens when a communication-partner is forced to change its address during an existing communication session? The communication breaks down. This is very problematic for a number of reasons.

Here are a few examples to illustrate the problems caused by broken on-going communications. When real-time media is used a breakdown of the data-stream is not acceptable. When one node receives a video broadcast it does not want the stream to break down just because it has to change its address. In the case of an address-change on the server-side it may not be possible

to use 'active resuming' because the server might not know the addresses of all clients. Last but not least security issues arise. For example if one node is using voice over ip telephony (VoIP) and changes its address while another node that is also using VoIP changes its address and is assigned the old address of the first node. The second node would suddenly be receiving the VoIP stream for the first node.

3.1.3 Related Works

Some works relate to the problems described above. Tunneling can be used to circumvent some of the problems but tunneling introduces a denial of service (DoS) problem (which will be covered in detail later). Another related work is done with MobileIP. In MobileIP a home agent (HA) which resides in a node's home network, forwards packets for the home address to the temporary address. The problem is that a MANET is typically not connected to the internet. That means that the HA is not accessible in the MANET. Therefore a different solution has to be found.

3.2 Solutions to Broken Routing Fabrics

First of all, it is necessary to state that for the scope of the paper, it is assumed that ad hoc on demand vector routing (AODV) is used as routing protocol. AODV is a reactive routing protocol. That means that it is only propagating routes when changes occur. Otherwise it is a simple table driven routing. The idea to prevent broken routing fabrics is that a node notifies its neighbors of an upcoming address change by using a 'route shift packet'. This packet contains the old and the new IP-addresses of the node that is changing its address. This packet is only sent to the neighboring nodes. This is ensured by setting the time to live (TTL) to 1.

One security issue that arises through these 'route shift packets' is that an attacker might spoof the old address and get all of that node's traffic redirected to itself. To anticipate such an attack, authentication has to be used. Because authentication through a CA causes too much overhead a 'cookie' approach is proposed by Zhou et.al. Every node generates a random number associated to its IP-address. The node sends a hash of this random number in all of its broadcast and routing messages. Every node that receives such a hash value, stores it in association with the IP. When a node needs to authenticate itself (for example in the route shift packet), it sends the original random number. Every node that received the hash can verify that the original random number is the source of the hash and verify the authenticity of the sending node.

3.3 Solutions to Broken On-Going Communications

This is the main focus of the work of Zhou et.al. How can communications stay active while one, or both nodes change their addresses. First it starts off with some assumptions on which the solutions are based. It continues with a simple solution for route rebuilding before it focuses on communication preservation. This part closes with a final note on challenges to key management.

3.3.1 Assumptions

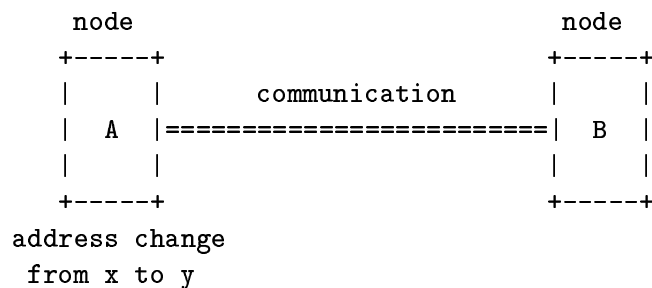
One major assumption is that the IP-layer supports more than 1 address per node. When talking about address changes, the new address is the primary and the old address is the secondary address. The primary address is used on all outgoing packets while the secondary address ensures that packets to the old address still reach the node (at least for a transition time). The networks HELLO messages are extended so that they contain both addresses. One last thing is that the node must not reply to routing requests (RREQ) sent to the secondary address.

3.3.2 Route Rebuilding

Route rebuilding is achieved through the use of 'gratuitous Route Reply' packet (gRREP). This gRREP is sent to all active and recent communication partners to ensure everyone notices the change and adapts accordingly. The gRREP packet also updates all the nodes on the path.

3.3.3 Communication Preservation

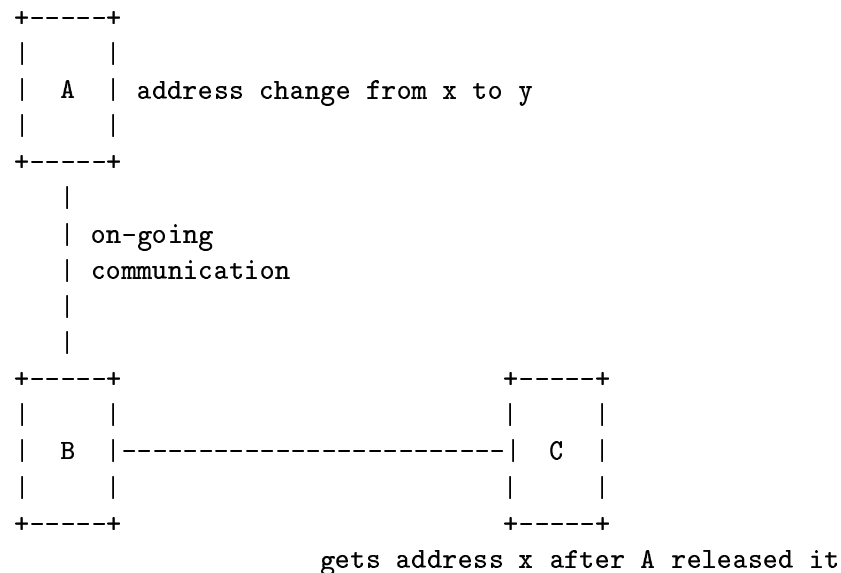
Why is it a problem when an IP-address changes during communication? The problem is that the checksums in the transport layer of the end-hosts are calculated using the addresses. When an address changes, the checksums will be wrong and the packet will be discarded as defective. The solution proposed by Zhou et.al. is to use an adapted network address translation (NAT) mechanism.



Zhou et.al. propose the use of NAT to conserve the communication between A and B while A changes its IP-address. The following table illustrates how NAT is used in both nodes.

Node	Incoming	Outgoing
A	new address y changed to x (for correct verification)	old source x changed to y
B	new address y changed to x (for correct verification)	old destination x changed to y

By using NAT in this way, the transport layer of both nodes does not notice the change of the address. The communication is still possible without problems. The advantage over tunneling is that it saves the overhead of a second IP-header in all the packets and that only one address has to be changed in each NAT. Furthermore the previously mentioned DoS problem does not arise.



Assume node A and B communicate. Node A has to change its address from x to y. Shortly after A released the address x, a new node C is assigned that address. This is no problem when using the adapted NAT, but when tunneling is used all traffic that is destined for C will be forwarded to A, flooding it with useless data. The drawback of NAT is that communication between A and C will not be possible as long as A still has x as secondary address, because it will consider all packets for C as local packets.

Zhou et.al. propose some further enhancements to NAT. Their paper describes an enhancement to use sequence and port numbers to discern connections. This enhancement is not necessary in the node that changes its address. Only the communication partner has to use an enhanced NAT table.

<i>old remote addr.</i>	<i>new remote addr.</i>	<i>local port</i>	<i>remote port</i>	<i>remote seq. num.</i>	<i>next remote seq. num.</i>
x	y	80	2030	228743	22884312
..

If a packet with the wrong sequence number is received by node B it will not be processed by the NAT. This way it is possible for node B and C to communicate, even while A and B communicate using the same address x with NAT.

The table shown above has to be installed at some point shortly after A changed its addresses. To notify node B of the change Zhou et.al. introduce an 'address change message' (ACM). This ACM message triggers the installation of the NAT entry. It has to be sent by A before any data packets are sent after the change has occurred. The data that has to be sent, can be buffered

until the ACM was sent. If no packets need to be transmitted to B, A may wait until B sends a packet before it sends the ACM. Because of possible security risks the ACM has to be verifiable (signed with As private key). To save overhead ACMs can be combined with the previously discussed gRREP messages.

The NAT table entries also have to be deleted at a later point. This is done when one of the following events trigger the deletion. When TCP communication is used the TCP FIN flag on a data packet triggers the deletion of the according entry. Because UDP does not support flags, a timeout is used for UDP connections.

3.3.4 Challenges to Key Management

The problem for key management arises because a nodes key can still be bound to an IP-address it does not have anymore. When the node requests its key it will be denied its own key, because it is still bound to its old address. The proposed solution is to use the same 'cookie' approach that is used in the 'Solutions to Broken Routing Fabrics' section. The node will generate a random number for each of its IP-addresses and sent a hash of that number to the CA. If the node changed its IP-address it can sent the original number in the request and the CA can verify it using the stored hash.

4 Summary

Both papers present interesting ideas for some of the problems that MANETs have. The feasibility of these approaches still has to be proven, but the concepts are interesting. The first paper lacks disappointingly on how the mentioned authentication methods are to be implemented. Especially the verification on non-technical level seems like a nice idea that is not possible in real life. The problem is that authentication through 'talking users' appears to be very unpractical, especially when the MANET is at a big conference with hundreds or thousands of people. Who would want to spend all his time authorizing people for the network use?

The second paper is very interesting, because it uses well-established methods to solve new problems. The problem is that the assumptions may reduce its usability to the few cases that fit the assumptions perfectly. In real life all these assumptions and prerequisites might not be met in most MANETs.

In summary I would say that both ideas are interesting and should be kept in mind. In the current state they are experiments and not feasible for real life use. Both parts should be tested thoroughly and afterwards enhanced to support a more realistic environment.