In: Sixth International Conference on Digital Information Management (ICDIM 2011),
Melbourne, Australia, 26 September 2011, http://dx.doi.org/10.1109/ICDIM.2011.6093328

BatCave: Adding Security to the BATMAN Protocol

Anne Gabrielle Bowitz*, Espen Grannes Graarud†, Lawrie Brown‡ and Martin Gilje Jaatun§
*ITEM, NTNU, Trondheim, Norway
†Watchcom Security Group, Oslo, Norway
Email: espen.graarud@watchcom.no
‡SEIT, UNSW@ADFA, Canberra, Australia
Email: Lawrie.Brown@adfa.edu.au
§SINTEF ICT, Norway
Email: martin.g.jaatun@sintef.no

***Abstract*—The Better Approach To Mobile Ad-hoc Networking (BATMAN) protocol is intended as a replacement for protocols such as OLSR, but just like most such efforts, BATMAN has no built-in security features. In this paper we describe security extensions to BATMAN that control network participation and prevent unauthorized nodes from influencing network routing.**

I. Introduction

This work developed from a perceived need to implement a secure ad hoc network that might be used in emergency services, disaster assistance, and military applications. Such a network needs to be established quickly, and without the need for existing fixed infrastructure. However it also requires controls to limit access to the network, in order to protect it from intruders or unwanted bystanders. We propose extensions to a suitable ad hoc network routing protocol, BATMAN, so that routing advertisements will only be accepted from authorised stations on the network. We propose the use of proxy certificates, which each client wishing to access the network will generate, and which are signed by one of the suitably authorised stations tasked with creating and managing the network. We assume these stations will be located with suitable emergency services command units that the network is being created to support.

The remainder of this paper is structured as follows: In section II we present related work, and in section III we highlight limitations in the current state of the art. Section IV introduces the BATMAN protocol, and section V outlines requirements that we pose to a secure ad hoc network solution. Section VI describes our solution, while sections VII and VIII present simulation results and experiences from our protype implementation, respectively. We discuss our contribution in section IX and offer conclusions in section X.

II. Related Work on Ad Hoc Network Security

Our proposals evolved from work on developing a secure restricted ad-hoc network for use by emergency services or disaster response personnel [1], [2]. In such a network, access must be managed, but be provided for members of multiple authorities which might not have online access to verify their identity. They focused on the design and implementation of the needed extensions to the OLSR ad hoc network routing protocol. However they only made a brief mention of the use of a public-key infrastructure to identify mobile clients and to authorise their access to some restricted ad-hoc network. They suggested that clients in a region would be pre-configured with certificates that could be used to automatically grant them access. They also noted that there needs to be some means of granting access to mobile devices that are not known, for personnal from out of region or from other services without peering arrangements. They suggested that such devices can be issued short-lived certificates, with limited rights, to grant them access. However details of this were left mostly unspecified.

In other related work, [3] outlines some issues with using X.509 certificates in such ad-hoc networks, particular problems relating to certificate validation and revocation. They propose the use of cached data and risk criterion to address these problems, assuming that some nodes can be intermittently connected, and so could access and cache data. This is unlikely to occur in the emergency or disaster scenarios we discuss. Thus the use of conventional long-lived certificates appears problematic, when immediate checking for certificate revocation is not possible. Short-lived X.509 certificates are proposed as a suitable mobile authentication method for low power or otherwise resource limited devices in [4], [5]. They suggest some reasons for choosing such certificates, which are conventional X.509 certificates but with a much shorter lifetime of hours to days. These include a desire to avoid the cost and overhead of certificate validation and revocation. And then to allow the use of less computationally intensive algorithms and key sizes than may be required in conventional X.509 certificates with lifetimes, and hence need for sufficient strength against attack, for periods of months to years.

III. Addressing Limitations in the Existing Work

Our proposed ad hoc network security extensions address some issues with the prior work noted above. First was the choice of ad hoc network routing protocol to modify. Although OLSR is an Internet standard, several papers have suggested thats its performance in practical trials is less than desired [6], [7]. Of the other protocols tested, it appears that BATMAN provided the best overall performance. We present further details on this choice in the next section.

Next was the choice of types of certificates to use to manage controlled admission to the network. The existing proposals involve using a mix of conventional and short-lived certificates, with the latter being generated in the field as required to support admission of stations without existing, verifiable, conventional certificates. However this means the stations issuing these need to support some certificate authority (CA) functionality, and have CA certificates available to sign these newly created certificates (short-lived or otherwise). Normal client stations would not normally have these.

We propose instead the use of proxy certificates, which are X.509 certificates with specific proxy extensions, that are signed either by another, conventional client certificate, or by a proxy certificate (PC), as we detail later in section VI-A. Hence any client station can potentially act a certificate issuer, able to grant access to other stations. The problem then becomes one of distributing knowledge of which stations have that authority, which we address as part of our protocol extensions. They can also be created with shorter lifetimes, and smaller key sizes, to better suit lower resourced mobile stations. Note with our proposed use of proxy certificates, they become an access token or capability used to gain access to a service, in this case the ad hoc network. This is very much the opposite sense to current use of these certificates, which are used by clients to delegate some of their access rights to a server, particularly in the grid computing domain [8].

Another problem not explicitly addressed in the previous work, is just what controls or restrictions were placed on the process of issuing certificates to grant access to the network. They identify the need to support differing categories of stations needing access. Some may be automatically recognized and trusted because they possess a conventional client certificate issued by a CA known to the proxy issuing client, most likely because both stations belong to the same service or administrative structure. In this case it would be reasonable to automatically issue the proxy certificate and grant network access without any human intervention. Other clients may not be immediately recognized, since they belong to other services, are volunteers, or just not previous known. In such cases it would seem reasonable to require manual verification that the client should be granted access before issuing a proxy certificate to them.

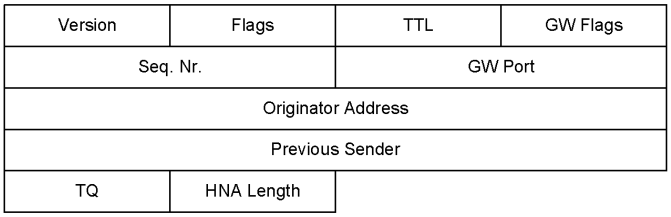
A further advantage in the use of proxy certificates is that they support the specification of restrictions on their use. We propose using this mechanism to assign different rights to different classes of clients. This could be used to indicate which clients are delegated the right to also issue proxy certificates granting access to other stations to the existing network. It also could be used to indicate that some stations should only be end-systems, and not used to relay traffic. Since X.509 certificates are widely recognized, it would also be possible to use the issued proxy certificates to authorise and authenticate the client's use of specific upper-layer applications.

IV. B.A.T.M.A.N.

BATMAN [9] ("Better Approach To Mobile Ad hoc Networking") is an increasingly popular routing protocol for wireless ad hoc networks, which was developed with an aim to replace the Optimized Link State Routing Protocol (OLSR) [10]. OLSR is a pro-active routing protocol, which means that participating nodes regularly exchange routing information with each other. According to the BATMAN developers, the problem with OLSR is that every node in the network calculates the whole routing path, which is a complex way to do it. Not only is it difficult to make sure all nodes have the same information at the same time, it also needs (relatively) much storage and computation time. If nodes sit on different routing information this concept leads to routing loops and heavy route flapping. The result is many patches to the protocol that defies the protocol standard in order to make it more suitable [10].

In BATMAN, each node should only know the next hop, i.e., the link-local neighbor that is the path between itself and the destination. BATMAN calculates the next hop of the optimal route by comparing the number of routing messages it has received from each node and who was the last sender.

The routing messages sent in BATMAN are called OGM. Figure 1 shows its packet format with all header fields. The OGM format has changed since the older BATMAN version III draft [9] was published. There is no official publication of the new version IV packet format, that we use, as of yet. Details of this revised packet format can be found in the project's internal documentation [1].



<table>
<tr><td>Version</td><td>Flags</td><td>TTL</td><td>GW Flags</td></tr>
<tr><td colspan="2">Seq. Nr.</td><td colspan="2">GW Port</td></tr>
<tr><td colspan="4">Originator Address</td></tr>
<tr><td colspan="4">Previous Sender</td></tr>
<tr><td>TQ</td><td>HNA Length</td><td></td><td></td></tr>
</table>

Fig. 1: BATMAN's OGM packet format.

The real workhorse of the packet is the "Originator Address" field which carries a host address of the node 'A' that broadcasted the OGM. When a node 'B' receives this message it checks if the originator address and source address of the IP header are the same - if so the two nodes are direct neighbors. B then forwards the OGM only changing the "TTL" and "Previous Sender" fields. All OGM inside the BATMAN network are broadcasted and rebroadcasted until the TTL has dropped to zero, or until they receive an OGM they have previously sent themselves.

This way all OGM will be received and rebroadcasted by all nodes in the network and all nodes will learn the existence of each other and which nodes are the first hop between them and the other nodes, i.e. the first leg of the path. All nodes and their first hops in their paths are stored in a list called an "Originator List".

When a node which has already received and forwarded an OGM receives the same OGM from another node at a later point - it drops that packet so the network will not get flooded

[1] http://gitorious.org/batman-adv-doc/

by forwarding the same OGM until its TTL is zero. This is also necessary in order to prevent routing loops.

V. Requirements

Ad hoc networks have some desired characteristics such as quick and inexpensive setup and being independent of communication infrastructure, but they also introduce great challenges regarding security.

A. Scenario

The design and implementation presented in this paper is mostly based on an emergency situation scenario, in which communication infrastructure is unavailable. If there is a major emergency situation such as an earthquake or tsunami, it is likely that parts or the entire communication infrastructure at the scene is destroyed or temporarily down. The remaining communication lines will then probably be congested, such that little communication actually goes through.

In this situation, it is of great importance that Emergency Personnel, such as Paramedics, Firemen, Policemen and the Military, are able to communicate efficiently and therefore independently of the public communication infrastructure. They need this network in order to manage the the operation, and therefore availability is probably the most important trait of this network. Secondly, they should be able to trust the communication on the network – i.e., messages sent are from whom they claim they to be.

Also, being able to authorize new actors on the scene, such as Red Cross, can be critical to the operation. These new actors will probably not have the necessary authentication tokens, i.e. certificates, required by the authentication scheme in the network.

B. List of Requirements

Based on the scenario above these requirements can be extracted and made into general requirements that needs to be addressed by the system design. The work presented here is based on several sources, most prevalent being the research from the OASIS project [2] [11] [1] and Winjum et al. [12].

R1 A node must be authorized in order to get full rights in a network [13], [14]

R2 A node without a recognized authentication token should be able to become authorized if necessary

R3 Networks need a master node which handles access control

R4 Access control (after initial authentication) should not rely on centralized nodes

R5 Different networks should be able to collaborate [12]

R6 Only master nodes can decide access policies of users/nodes

R7 Nodes must not be able to alter access policies they are ruled by

An early study produced security requirements of ad hoc networks demanding that the routing logic must not be spoofed or altered to produce different behavior [13]. This means authorization is required (R1) before someone can partake in routing logic. The OASIS project [2] specifically considered a situation where e.g. NGOs contribute to a rescue operation, which means they need to somehow acquire credentials (R2), but this must be administered by some authority (R3). R4 highlights the need for authenticated nodes to function autonomously. A desire for seamless radio coverage over the area gives us R5. R6 comes from the fact that it is not possble to determine access policies prior to network setup, and R7 states the rather obvious, in that nodes that could alter the access policy would violate R6.

VI. Security Solution Overview

The system design requires nodes to be authenticated and trusted before being allowed into the network. Each node also has to verify their identity periodically, or they are dropped from the network. We present the solution briefly in the following; for more details see Graarud [15].

The network setup starts with an out-of-band authentication where a master node, hereafter referred to as a Service Proxy (SP), verifies new nodes. How this is done can be up to the application, but let us assume that the actors carrying their communication devices, hereafter nodes, physically meets the SP at the scene and exchange their public key fingerprints.

When a new node is discovered by the SP using regular routing announcements as part of the pro-active routing protocol, the SP will invite the new node to a handshake to establish a trust between the two nodes. The new node will receive the SP's certificate, and will after verifying the fingerprint request a proxy certificate for itself. After verifying the node's fingerprint, the SP will issue a proxy certificate with (possibly) the rights to participate in building the MANET by broadcasting its own and re-broadcasting other trusted nodes' routing announcements.

A. Further Proxy Certificates Advantages

A Proxy Certificate (PC) is used to delegate rights on behalf of the issuer. That means that the issuer, i.e. the SP, can choose to delegate all or a subset of its rights to the receiver of the Proxy Certificate. This may be useful in a situation where the nodes are unable to properly authenticate themselves with any pre-existing conventional X.509 certificate, when the SP on the scene has no means to verify their certificates. This can be true if their certificates are issued by an unknown root certificate (CA), or if there is no online access to enable verification of the certificate path.

The SP may also be interested in giving the node rights the node would not usually have on this specific scene, depending on the situation. This is easier to achieve when the SP can delegate its own rights.

An important feature of the PC is that the SP can delegate different kind of rights, as long as it is a subset of its own rights, to different nodes. There are countless of different rights that can be useful, given the situation they are used in, but here is a few possible rights/privileges to give the reader an understanding of the possibilities they give:

- Announce itself - let the MANET know of your existence

- Re-broadcast other nodes announcements - reshape the network topology
- Announce a gateway - give the MANET access to another network
- Use the gateway - allow you to communicate outside the MANET
- Send and receive messages with a defined application - full application rights
- Only receive messages from a defined application - limited application rights

If you are setting up a MANET on the scene of a disaster to assist emergency personnel, you could have some actors be able to organize the effort by sending orders/commands to the other actors, while some actors only are allowed to receive the orders. In this situation it might be of great importance to know that only verified nodes are able to give commands, but the importance of getting this information available outweighs the need to verify the nodes/actors receiving this information.

B. Post-Authentication Operation

After being issued with a Proxy Certificate (PC) the newly authenticated node will periodically "broadcast" - unicast to each neighbor - a message containing an ephemeral key and corresponding Initialization Vector (IV), a pseudo-randomly generated nonce, and a digital signature over this message. The ephemeral key is encrypted with the neighbor's public key (hence multiple unicasts instead of an actual broadcast), but the digital signature is generated based on the unencrypted key and the other contents of the message, and is thus identical for all neighbors.

After sending this signed "broadcast" to each neighbor, the node and its neighbors will generate a keystream from the ephemeral key, IV, and nonce. The node will then append two new bytes from this keystream to each routing announcement, and re-broadcasts of neighbors' announcements, sent from this point forward with a sequence number for the recipient to be able to match this "extract" with the keystream at an offset given by the sequence number. The two bytes will then in effect be a one-time password similar to that used by some online banking applications. If this one-time password value is absent or incorrect, the announcement will be dropped and regarded as a spoofing message.

Whenever a routing announcement is re-broadcasted by another trusted node, that node will first replace the sequence number and one-time password that it has verified with the next two bytes of its own key stream. This means that every node only checks its direct neighbor for authentication, which is a design choice. This proposal assumes that since every node is verified by the SP in the first place, all nodes in the network will be able to trust each other, which also means they will trust their neighbors to properly verify their neighbors again.

In order for trusted nodes to learn of newly trusted nodes existence, the SP regularly broadcasts lists containing the id, address and public key of each trusted node in the network. This needs to be done, as before learning about a new node the other trusted nodes will not accept any messages from this node. This means the new node will not be able to exchange its own PC with other nodes directly - only through the SP.

The list, hereafter Authentication List (AL), also adds some web-of-trust like capabilities. The list is signed by the SP, which means the integrity of the list is guaranteed by the SP. This means that if the SP should go offline, e.g. it could be out of range, other trusted nodes in the MANET can continue to broadcast the AL on behalf of the SP - to ensure all nodes in the network know each other. This can be especially important when the network grows large and become fully or partially separated and nodes in one part may not have learnt of the existence of newly trusted nodes yet. It also applies to trusted nodes who have been offline while new nodes have been verified, then re-enter the network while the SP is offline.

VII. Simulations

We have implemented both standard BATMAN and the version with our security enhancements in the network simulation package ns3 [16].

Figure 2 presents Packet Delivery Ratio (PDR) and packet delay results from the simulations running Secure BATMAN, BATMAN and DSDV with 10 nodes and 10 traffic flows.

As seen from Figure 2a, the PDR values of all three routing protocols all well above 80%. Interestingly, Secure BATMAN's PDR values also stay at approximately the same level as the two other protocols. At pause time zero, which is equivalent to continuous node movement, all three protocols show their best behavior with the highest PDR values. This is probably due to the fact that they all are ad hoc network protocol tailored for networks with high node mobility.

When looking at the end-to-end latency in Figure 2b it is surprisingly the Secure BATMAN protocol which has the best results.

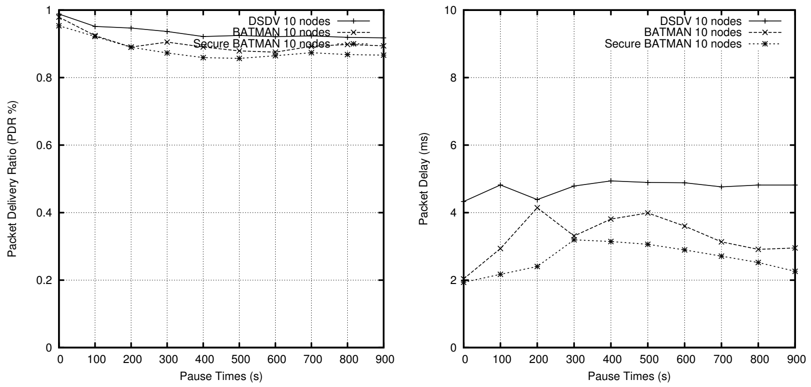
VIII. Prototype

We have implemented our proposed protocol changes by modifying the BATMAN code distributed with a recent Ubuntu Linux distribution.

A. Initialization Phase

Figure 3 presents neighbor discovery results for both the original (Fig. 3a) and modified (Fig 3b) version of BATMAN. The two graphs shows the time in seconds on the y-axis and the trial/run number on the x-axis. The two colored lines on the graphs show the results from first neighbor discovery until the first neighbor is added to routing table (green line - marked with "x") and until both nodes are added to the routing table (red line - marked with "+").

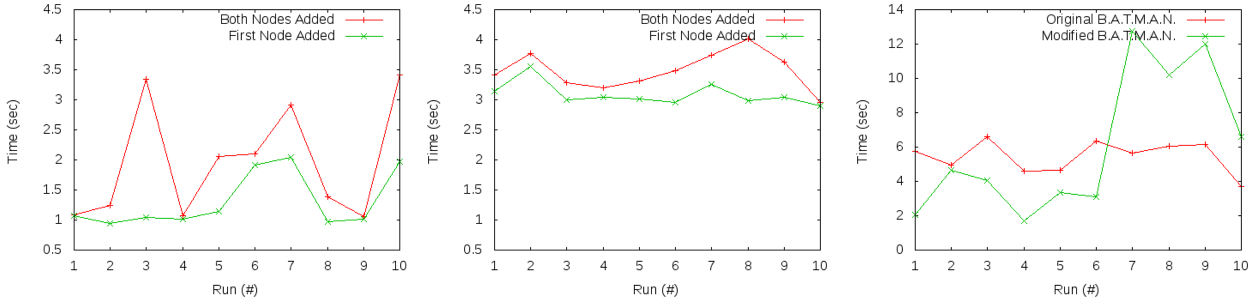
The results from the original protocol, shown in Figure 3a, shows high variance in the time needed to add one and two nodes to the routing table. For 7 out of 10 "first nodes" the time needed is relatively equal, being about one second. For both nodes to be added however, there are much more variance - variying from the best possible time, i.e. equal to adding one node, and up above 3 times longer than adding one node. '



(a) PDR with varying pause times.

(b) Packet delay with varying pause times.

Fig. 2: Simulations results from BATMAN, Secure BATMAN and DSDV (10 nodes and 10 source and sink pairs)



(a) Original BATMAN

(b) Modified BATMAN

(c) Routing path convergence time

Fig. 3: Neighbor discovery for original and secure BATMAN, and routing path convergence

Figure 3b shows the results from the modified version proposed in this thesis. These results indicate that the behaviour of the modified version seems to correlate with the behaviour expected from the hypothesis. A seemingly constant of about two seconds seems to be added to the process of adding both nodes to the routing table.

Another interesting observation is that the time variance seems to be much less than that of the original version. This might be because the authentication handshake and the keystream sharing happens in a separate thread from the regular BATMAN operations, meaning the BATMAN protocol continously receives routing announcements to process while the Authentication Module (AM) handles its part. The idea being that while the AM thread runs, the BATMAN thread "gets ready" to do its part of the job.

B. Route Convergence

The results of the second test are shown in Figure 3c. In this figure, the axes are the same as in the figures above: y-axis shows the time in seconds, and the x-axis shows the trial run. The red line shows the performance of the original implementation, while the green line shows the modified.

As indicated earlier, this test's results are somewhat unclear. While the results using the original implementation seems relatively uniform, with only about 1 second variance, the results from the modified implementation are highly irregular.

Looking through the logs from this test one thing become apparent. With different hardware on the different nodes in the network, their wireless cards send at different levels of transmission power, meaning that while one node can receive packets from a "stronger node", the packets sent might not be received by the other nodes.

The BATMAN protocol messages (routing announcements) are sent quite often, depending on the number of re-broadcasts being sent, meaning the time from when a node is within transmitting range and until its broadcasts are received by nodes within its transmitting range will be quite short. The AM messages however, was mostly tested in an ideal environment where most packets were received, so this was not properly accounted for. Therefore, if a routing announcement from a "stronger node" is received by a "weaker node", the weaker node might send its keystream material without the other node receiving it.

Re-transmitting mechanisms based on guessing that the receiving node has not received the AM messages are in place, but as the mechanism wait until it believes the other node has not received, instead of knowing it instantly. This can of course be managed adding ACK'ing to each AM message, which was not added initially because of the wish to minimize overhead. This however, might have to be re-evaluated.

Another thing to notice is how multiple trial runs using the modified version actually performed better than the original version. This is impossible to explain talking about the design and implementations themselves, but is probably most accuratly explained in the terms of external environment.

IX. Discussion

The proposed system design uses a novel solution to continuously verify routing announcements received from one's neighbors. For this system to be used on typical mobile devices with all their constraints, limitations on computing power, battery lifetime, and saturation in the wireless network must be acknowledged.

Because all nodes in a MANET using a pro-active routing protocol broadcast their routing announcements and forward all received routing announcements, the network traffic will increase exponentially to the amount of nodes in the network and how closely bound they are. Therefore all routing announcements need to be as small as possible. A typical signature is usually one or two orders of magnitude larger than a regular routing announcement, so by adding a signature to the routing announcement - most of the data sent in the network would be signature data. This is far from ideal.

The first solution that one would think of would be to only sign a very few of the announcements, periodically. This however, would be totally disastrous. This would have no protection against spoofing attacks whatsoever, as an attacker could wait for a legitimate node to send a signed announcement and then send his own fake announcements spoofed with the legitimate node's address.

The solution proposed in this paper solves the problem in a different manner. Since each node and its neighbors generate a key stream that can be used to verify messages from that node, only messages with a correct, previously unused, "one-time password" will be accepted and forwarded by any neighbor. Since the keystream has to be renewed periodically, any node not possessing the correct proxy certificate will be dropped from the network upon renewal.

This scheme is fully based on trust. You trust that each of your trusted nodes will only send you its own annoucement (correctly) and rebroadcast only its trusted nodes' announcements without modification. If for some reason a trusted node should behave maliciously, this scheme will not detect this, and allow the trusted node to potentially disrupt the network.

X. Conclusion

We have presented a security extension to the BATMAN ad hoc routing protocol which handles controlled network admission and prevent unauthorized nodes from influencing routing decisions in the network. Our ns-3 simulations indicate that the security mechanisms do not place an undue burden on the network nodes, and our protoype implementation confirms that although further refinements are desirable, BatCave represents a viable security solution for ad hoc networks.

Acknowledgements

This work was started as part of a European Commission Erasmus Mundus visiting scholar grant, and is based on MSc thesis work for the department of Telematics at the Norwegian Univeristy of Science and Technology (NTNU).

References

[1] A. A. Nyre, M. G. Jaatun, and I. A. Tøndel, "A secure MANET routing protocol for first responders," in *Security and Communication Networks (IWSCN), 2009 Proceedings of the 1st International Workshop on*. IEEE, 2009.

[2] I. S. Svagård (editor), "Information security for field workers in crisis situations," SINTEF ICT, http://www.oasis-fp6.org/documents/OASIS_SP24_DDD_253_security_SIN_1_0_pub.pdf, Tech. Rep., 2008.

[3] J. L. Muñoz, O. Esparza, C. Gañán, and J. Parra-Arnau, "PKIX Certificate Status in Hybrid MANETs," in *Proceedings of the 3rd IFIP WG 11.2 International Workshop on Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks*, ser. WISTP '09, 2009, pp. 153–166. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03944-7_12

[4] P. K. Sharma, "Short-Lived Certificates as a Mobile Authentication Method," Master's thesis, Technical University of Denmark (DTU), 2009. [Online]. Available: http://orbit.dtu.dk/getResource?recordId=245323&objectId=1&versionId=2

[5] M. Pitkanen and H. Mikkonen, "Initalizing mobile user's identity from federated security infrastructure," in *Proceedings of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM-08)*, 2008, pp. 390–394. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/UBICOMM.2008.64

[6] M. Reineri, C. Casetti, and C.-F. Chiasserini, "Routing protocols for mesh networks with mobility support," in *Proceedings of the 6th international conference on Symposium on Wireless Communication Systems*, 2009, pp. 71–75. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5285344

[7] M. Abolhasan, B. Hagelstein, and J. C.-P. Wang, "Real-world performance of current proactive multi-hop mesh protocols," in *15th Asia-Pacific Conference on Communications (APCC09)*, Oct. 2009, pp. 44–47. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5375690

[8] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist, "X.509 Proxy Certificates for Dynamic Delegation," in *Proceedings of the 3rd Annual PKI R&D Workshop, Gaithersburg MD, USA*, 2004.

[9] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better Approach To Ad-Hoc Networking (B.A.T.M.A.N)," Last accessed December 19, 2010, http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00.

[10] Open Mesh, "*Why starting B.A.T.M.A.N.?*" Last accessed december 19, 2010, http://www.open-mesh.org/wiki/why-starting-batman.

[11] I. A. Tøndel, M. G. Jaatun, and A. A. Nyre, "Security requirements for MANETs used in emergency and rescue operations," in *Security and Communication Networks (IWSCN), 2009 Proceedings of the 1st International Workshop on*. IEEE, 2009.

[12] E. Winjum, P. Spilling, and Ø. Kure, "Ad Hoc networks used in emergency networks : the Trust Metric Routing approach," FFI Rapport 2005/04015, Tech. Rep., 2006. [Online]. Available: http://rapporter.ffi.no/rapporter/2005/04015.pdf

[13] B. Dahill, B. Levine, E. Royer, and C. Shields, "A secure routing protocol for ad hoc networks," *Electrical Engineering and Computer Science, University of Michigan, Tech. Rep. UM-CS-2001-037*, 2001.

[14] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Network Protocols, 10th IEEE International Conference on*, 2002, pp. 78–87.

[15] E. G. Graarud, "Implementing a Secure Ad Hoc Network," Master's thesis, Norwegian University of Science and Technology (NTNU), 2011. [Online]. Available: https://github.com/espengra/secure-ad-hoc-network-doc

[16] A. G. Bowitz, "Simulation of a Secure Ad Hoc Network," Master's thesis, Norwegian University of Science and Technology (NTNU), 2011. [Online]. Available: http://sislab.no/batcave.html