Revocation of User Certificates in a Military Ad Hoc Network

J. Jormakka and H. Jormakka

***Abstract*— This paper presents a scheme for revoking certificates in a medium-small size semi-ad hoc military network, but the solution can be applied in the civilian side e.g. by police and crisis management. It describes the functionalities of a protocol to handle certificates, a set of policy rules in a node for handling certificates and an analysis how the proposed mechanisms can mitigate attacks on the certificate revocation solution. The mechanisms allows communication between the nodes on a lower security level even if the latest certificate revocation list is not available; protects against false revocations of certificates; and implements a mechanism for lowering trust levels of certificates.**

***Keywords*— Ad hoc networks, Certificates distribution and revocation, Military.**

I. INTRODUCTION

A mobile ad hoc network is a wireless network where nodes transfer data to each other without the help of a base station. Usually data is passed through other nodes hop-by-hop. The paper describes a certificate revocation scheme of a military communication network for command posts and brigade headquarters. The network operates in semi-ad hoc mode, i.e., as an ad hoc network that is often connected to a fixed network.

Certificate revocation is the mechanism by which a Certification Authority (CA) announces that a certificate it has issued is no longer valid, even though its validity time has not expired. Certificate revocation is necessary if the private key corresponding to the public key in the certificate is suspected to be compromised, or for other reasons, e.g., if the user changes affiliation or name.

The certificate revocation mechanism in the ITU-T X.509 Recommendation uses Certificate Revocation Lists (CRL): the CA sends periodically new CRLs and puts them to the X.500 directory. Users can recover the list from Certificate Revocation List Distribution Points (CRL DP), a 1993 addition to the 1988 version of the X.500 Recommendation. The CRL mechanism is commonly used with other directory solutions.

Certificate Revocation Lists have a number of problems. One is the scalability of the mechanism in a very large network. Various solutions have been proposed in the literature. The network that is studied in this paper is so small that the scalability problem does not arise. Another problem with CRLs is that there is some time delay between the compromise of the certificate, e.g., loss of the private key, and the revocation of the certificate. Thus, there is always some time for authentication fraud. A problem that is characteristic to Mobile Semi-Ad Hoc Networks is the unavailability of the CRL: if the CA is reached through the fixed network and a wireless user is not sufficiently often connected to the fixed network, he cannot always have the latest CRL. Therefore he should not trust a certificate of another user. However, communication between users in the ad hoc network may be even more essential than reliable authentication. New mechanisms are required allowing communication between users on as high security level as can be offered and enabling sufficiently secure and efficient revocation of certificates in a semi-ad hoc network.

J. Jormakka, Aalto University, Espoo, Finland, Comnet, (adjunct prof.) jorma.o.jormakka@gmail.com

H. Jormakka, Technical Research Centre of Finland (VTT), Espoo, Finland, henrykasj@gmail.com

II. RELATED WORK

Certificate revocation is one of the known weaknesses in public key cryptography and a large number of research papers have been written on the subject. Much research has been directed to the scalability of the CRL mechanism by improving data structures, see e.g. [1], [2], [3], [4], [5]. The X.500 Recommendations already contain some options, like Delta CRLs. In a large network CRL distribution poses scalability challenges since the CRLs are typically very large. More general performance issues of large networks have also been treated, like in [6] and [7].

There are rather few proposals for certificate revocation in wireless ad hoc networks that address the problem that CRLs are not always available. This situation is most compelling in military ad hoc networks where connection to the fixed network that usually holds CLRs is often unavailable, and there exists a determined adversary, who tries to take advantage of the situation. In crisis and emergence response operations one usually may assume that the connections work and adversaries either to not exist or are not competent. In civilian ad hoc networks there often does not exist compelling reasons to secure the networks against imposters. However, in a military network one must secure communications. There are not that many alternatives if CRLs cannot be obtained. Either the protocol does not need certificate revocations – but then it requires renewals or other similar costly operations – or certificates are revoked by one or more participants in the ad hoc network.

Li et al in [8] describe a scheme for wireless ad hoc networks where each node keeps up the validity of its certificate using a One-Way Hash Chain. Other nodes can request the node to send a certificate with updated validity information. The method dispenses with certificate revocations, but it relies on keeping, in addition to the private key, another private secret in the mobile node. The authors propose keeping it in an USB-Key. This method can be useful in a network where the device may be lost but users carry the

USB-Key with them and this key is not lost. In a military ad hoc network the method does not give good security: if a node is lost it often means that not only the user, but also all he carries with him, is under adversary control. URSA in [9] is also a method, which dispenses with certificate revocation. It accomplishes this by requiring tickets. The drawback is a considerable traffic in renewing tickets. In Chinni et al. [10] certificate revocation is very shortly discussed in the context of renewing a certificate: the local environment is checked and if the node has not misbehaved or marked as convicted, it is granted a certificate. This description is too terse to be a method that can be implemented but it seems to suggest some kind of a vote.

Several authors have thought that if many users are needed for revoking a certificate, then certificate revocation must be slow. Thus, they have devised alternatives where only one user is needed to revoke a certificate. Naturally, this user may be mistaken or malicious, therefore the proposals require some cost to the revoking user, or assume that only trusted users can revoke certificates. As example of the first alternative is the suicide method in [11]. If a user revokes a certificate, he at the same time revokes his own certificate. In a military network this solution is not acceptable as the revoking soldier loses his communication capabilities and cannot fill his tasks. Two examples of the second alternative are [12] and [13]. In both cases only one user is needed to revoke a certificate, but the revoking user must be trustable. This does not work as we cannot know who is trustable.

Thus, a working solution for a military ad hoc network needs some kind of voting. Arboit et al in [14] present a revocation scheme based on a reputation protocol. It is voting but it focuses on network nodes independently collecting information of bad behavior from all other nodes. Because of this, the method will not revoke certificates sufficiently fast in order to protect military operations. The method presented in this paper is also based on the voting method. We require voting since one person can and does make mistakes and it is difficult to decide if a colleague should be excluded from the network. It is possible that his identity is stolen or that he is a spy, but this is never immediately clear. Thus more than one vote should be needed, but the votes are not accumulated over time as in [14]. Instead, if some suspicions arise, the soldier noticing them contacts physically another, usually higher ranking, member of the network, and they collectively exclude the potential threat.

Kitada et al [15] propose a Public Key Infrastructure (PKI) system where a node collects the certificates that it needs on demand. They also want to dispense with CRLs. The knowledge of revoked certificates is maintained by each node locally by asking each node that has issued a certificate if the certificate is valid. This solution is unsuitable to semi-ad hoc networks: if the issuer is in the wireless network it may be compromised, while if the issuer is the CA which is reachable through the fixed network, the connection may be broken. Morogan and Muftic [16] propose a validity time, the *grace period,* for CRL and a mechanism *channeling of the update information* by which ad hoc network nodes can obtain the latest CRL from each other. The mechanism for distribution of certificates is similar to the one presented in part *B.* of section 4 of this paper.

III. USE OF CERTIFICATES IN MANETS

The results presented in this paper were obtained in a project, where a medium-small size military mobile ad hoc network intended for the purposes of a command post or a brigade-level headquarter was designed and a mobile ad hoc node was implemented using, whenever possible, commercially available hardware and software. As a security solution in a military network has military specific requirements and cannot be directly adopted from civilian solutions, revocation of certificates was one of the parts that were designed specifically for that type of network.

The security requirements are stricter than in the civilian side since the adversary is better motivated and more capable than in many other usage scenarios, and the network gives access to classified material. Additionally, connections may be one-way only, i.e., existence of a jamming device, variations of signal power levels, or some radio propagation effects allow transmission to one direction only. Though the usual operation of the network requires high bandwidth bidirectional links, such as 802.11g or 802.16, the security solutions should be able to work on one-way connections if needed. As most protocols require responses, a very low bandwidth backward channel can be assumed to exist.

In many wireless ad hoc networks, both in the civilian and military sectors, there is a constraint on computing power imposed by small battery powered devices. This constraint is not critical in the intended network where users are usually in armored vehicles and the power source is provided by generators of these vehicles.

A stand-alone mobile ad hoc network can use whatever suitable authentication method between the users of the network nodes, but usually the ad hoc network is a part of a larger network and is in fact a semi-ad hoc network. The mobile ad hoc network considered in this paper is a typical military ad hoc network, which means semi-ad hoc network as the services to be used mostly reside in the fixed part and the network must provide organization-wide connectivity. Figure 1 shows the location of the brigade headquarter network between the fixed network and the tactical network (wireless military network for combat net radios).

The prevalent way of authentication in the wired network is based on the Public Key Infrastructure (PKI). The users in the mobile ad hoc network communicate with the services and users of the fixed network. This is why the authentication method in the mobile ad hoc network is most naturally also based on certificates. As the network is owned by one organization there is only one CA and all users have the CA certificate. Joint operations will become more common in the future and multiple CAs must be supported. The implied changes to handling certificate revocations are not large; mainly, a certificate path is needed.

We can assume that the CA can be trusted as CA compromise is expected to be very difficult. The mobile ad hoc network users can reach the CA through the fixed network.

In the fixed network user certificates, CA certificates and certificate revocation lists are stored in a directory. In the wireless ad hoc network public key cryptography should preferably be used without a directory because it cannot be

assumed to be always available. A combination of the following two methods dispenses with the directory.

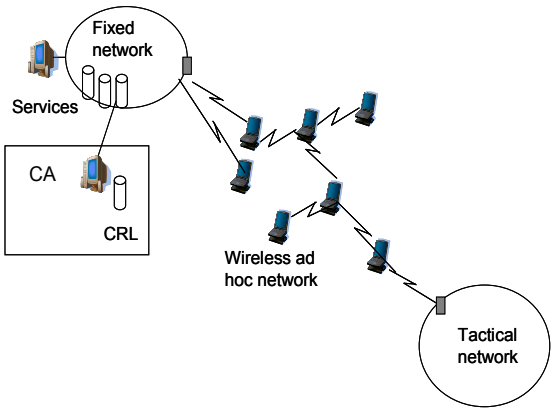

Figure. 1. Command-post/brigade headquarter ad hoc network. The terminals access services in the fixed network and take part in group decision making. The applications are database applications with graphical interfaces, voice and possibly video.

A. Storing certificates of all users locally

Storing certificates of other users locally is a possibility if the network is not too large. The memory of a smart card is not sufficient for certificates of all users even in a small network. Let us assume that only the certificates of the user himself, those of the services and the CA certificate(s) are stored on a smart card. Certificates of other users can be stored on a USB plug-in, Compact Disc (CD) or hard disc. Let us assume that the way the network is used is that activities are divided into operations which have a well-defined starting time and usually also a well-defined finishing time. At the beginning of each operation valid certificates are distributed to all users e.g. with a USB plug-in or a CD. During the time of the operation certificate revocations and new user certificates are added to the hard disc. A certificate of a new user can be obtained from the CA .

B. Sending own certificate

The user who wants to be authenticated sends his certificate during the authentication phase. The receiver checks the validity of the user certificate by checking the signature of the CA and the validity time of the certificate.

An adversary trying to impersonate a user may give a revoked certificate which is still valid. Therefore, either the validity time should be very short or users need certificate revocation lists in order to authenticate other users in a secure way. In the first case the certificates must be renewed during the operation. This may be impossible as connectivity to the fixed network can be broken. We will assume that the validity time of the certificate sent by a user has not expired during the operation time frame, thus certificate revocations are necessary.

IV. ATTACKS AGAINST CERTIFICATES

Public key cryptography has a quite high security level, but there are many potential threats to certificates.

- Attack 1: Registering a certificate on somebody else's name. The CA is assumed to be capable of validating the user before giving a certificate.
- Attack 2: Changing a certificate to a crafted certificate. Certificates are signed by the CA making this practically impossible.
- Attack 3: Pretending to be a CA and falsifying certificates of users. All trust is lost if the certificate path includes a CA that does not correctly check the identities of users to whom it issues user certificates. In practice, an adversary must be able to insert a false CA into a certificate path. In the 1988 version of the X.500 directory this was at least theoretically possible since user certificates and CA certificates had the same structure and an adversary might cause a user certificate to be taken as a CA certificate. A user gets a user certificate if a CA trusts that the user is whom he claims to be, whereas a CA Certificate is only granted to a party who is trusted to be able and willing to behave as a CA, e.g., to check the identity of other users. In X.500 versions since 1993 the possibility of this attack can be removed since there is a field that can be used to distinguish between certificate-types.
- Attack 4: Blocking CRL updates or removing them from the directory. This attack is solved in X.509 by requiring that CRLs are sent periodically on known times (i.e., CRL indicates the time the next CRL will be received). Then the user knows if he has the latest CRL. The update period for CRL may be too long for a particular application and there may be times when an immediate revocation is needed. The CA can issue a new CRL at any time but it cannot be assumed that a user will receive the CRL or know of its existence. Thus, the same revocation must be also in the periodically sent CRLs.
- Attack 5: Registering a valid user certificate with a name that is misleadingly similar to a name of a valid user. This mechanism is a form of social engineering: a user mistakenly takes another user for somebody else and authentication also succeeds: the cryptographic application authenticates him to be what his name literally states. This kind of acting is difficult if the CA is making intelligent decisions of what names users can have. There are cases when this type of misleading may still be possible. A user with the same name may be mistaken to be another user sending e.g., from the home terminal. If a company has a good security policy, such cases can be minimized.
- Attack 6: Threatening, bribing, blackmailing or in other way persuading a valid user to behave in the way the adversary desires. This threat cannot be removed by technical means.
- Attack 7: Stealing the private key and using it before the theft is noticed. This is always possible and cannot be well protected against.
- Attack 8: Hacking, social engineering, sending malicious code, or in other ways breaking into network nodes and stealing or modifying information.
- Attack 9: Announcing the certificate of a valid user compromised. This can cause a denial of service attack to the valid user.
- Attack 10: Downloading CRLs in order to cause a denial of service attack to the network.

- Attack 11: Checking revocation of a certificate in order to learn something of the state of a valid user. If lost private keys are revoked, this gives information of what nodes are known to be lost. In a military network this can be important information.
- Attack 12: Gaining access to the system with stolen credentials although loss of the private key was noticed. This may occur if certificate revocation has not been spoted.

Additionally, the following problems may occur in connection with a public key cryptosystem:

- Problem 1: Refusing communication with a valid user since the latest CRL is not available and one side of the communication cannot verify the validity of the public key. In time-critical activities this is an annoying problem.
- Problem 2: CRLs can be long and distribution of CRLs may cause excessive load to a mobile network.
- Problem 3: The CA may become a bottleneck.

Some of the above attacks are sufficiently well solved by the Public Key Infrastructure system in the fixed network (Attacks 1, 2, 3 and 4). However, we must still keep these attacks in mind when proposing any modification to the handling of certificates. For instance, a mechanism sending revocations on demand must consider Attack 4, while sending a validity statement of a certificate on demand can be quite secure. Some attacks are unsolvable or non-technical and cannot be dealt with (Attacks 5, 6, 7 and 8). We look for a solution mitigating or removing four attacks (9, 10, 11 and 12) and the three problems. As there are attacks that cannot be removed, the goal is not perfect security but rendering the mentioned attacks at least as difficult as the main remaining threats.

V. REVOCATION POLICY ALTERNATIVES

A typical service provided in a military network accesses sensitive data stored in data warehouses. It is most often a database application and the access paradigm is publish/subscribe. There are multiple copies of the same data and we can assume that the load on one data warehouse does not become a limiting factor and a user can obtain the service he needs from some of the data warehouses. Partial local copies of the data are made in the wireless network. Services are not mobile, i.e., moving from one computer platform to another, mainly for security reasons.

In a typical service usage scenario a user authenticates to a service with strong credentials using a certificate and a private key stored on a smart card and that the card is protected by a password. The services could be protected by an additional password that have to be memorized and that is one per user to all the services. The motivation for the password is to protect the services in case a node and a smart card are both captured, and the password (PIN code) to the smart card is available, or e.g., the lock on the smart card is removed by a suitable application of a voltage level.

It is common that a node or the whole wireless network is not connected to the fixed network. Usually this only means technical communication problems but in case of a military network it may mean that the smart card, possibly the whole node, is captured and a private key may be compromised.

There are at least three possible approaches to certificate revocation:

1. *If the wireless network is not connected to the fixed network do not usually revoke certificates corresponding to a lost smart card. Let the CA, or a node authorized by the CA, the possibility to revoke a specific certificate*

In this policy the assumption is that it is very difficult for the adversary to obtain the private key even though the valid user has lost his smart card. However, it should be able to revoke a specific certificate if there is reason to suspect that a private key is compromised.

2. *Use a shared and not electronically stored password for authentication as an additional mechanism to strong credentials*

Password authentication is here used in a way similar to a PIN code. If the password can be recovered from lost card, the mechanism does not add any strength to security. Therefore the password must not be stored. It is difficult to remember many passwords and therefore it should be shared by the users of the wireless network.

3. *Always revoke all certificates corresponding to a lost smart card.*

This is a natural policy: usage of a lost smart card indicates that authentication with the credentials is not from a valid user, or at least one should verify the user.

The advantage of the first policy is that revocation is done by the CA in the same way as in the fixed network, i.e., when the ad hoc network has connectivity and there are no military ad hoc network specific problems. In the intended application the material is highly sensitive and the risk that a private key is broken from stolen equipment or obtained from a captured user is too large and overcomes the problems of certificate revocation. This policy must be discarded.

The second policy does not need CRLs and access cannot be gained by using a stolen node or smart card. In a very small military network the users could be considered being able to keep the shared secret. However, there is a disadvantage - one more password has to be memorized. This means that the mechanism is too weak to protect sensitive data. We must discard this solution also.

The selected solution is the last policy. In that case certificate revocations can be common and occur when the network is disconnected from the CA. This implies that there is a need for a mechanism revoking certificates also when the CA is not reachable. The protocol enabling certificates revocation is briefly presented in the next section.

VI. CERTIFICATE REVOCATION PROTOCOL

The protocol supports four functionalities each containing some protocol actions. Only the main ideas of the functionalities are presented in this paper.

User credentials are considered lost if they are announced lost to the CA. This means that the network is not polling users in order to check whether their credentials are lost, instead there is normal communication between users and if a user thinks that credentials of

another user are misused, he issues a *Doubt* message to the CA. The *Doubt* mechanism is described later. The CA makes the decision to consider credentials lost and sends a revocation of the certificate.

A. Verification and distribution of certificates

The first functionality that must be supported enables users communication even if the fixed network is not available. Although most user certificates are available in a local storage, CRL is not always up to date. Unencrypted communication is not an acceptable solution. Functionality *A*. comprises of three protocol actions and has the necessary security level.

Partial-authentication

In case when two users start communication and one of the nodes notices that it does not have the most current CRL, it concludes that it can make only partial authentication. Communication is possible with partial authentication (PA), but in that case a particular *PA security policy* is applied by both sides restricting the message types and services. Messages between parties using partial authentication are marked with a special *partial-authentication* flag. This makes it possible for other nodes to notice if partially authenticated communication is carried through them.

Send-own-certificate

A user can add its certificate to the message if it expects that the other side of the communication does not have it. This mechanism consumes bandwidth considerably and should be used only by users who enter a new, rarely used by him network, or if they have received a new certificate.

Distribute-certificate

A user may distribute a certificate to the whole mobile ad hoc network. An efficient distribution mechanism is assumed to exist in the network as the typical service in the network is group collaboration, e.g., planning military operations. Distribution can be made using a distribution node and one-to-one connections or by multicast. Multicast in mobile ad hoc networks is usually difficult, but in the application the nodes are typically not moving while taking part in group collaboration. The distribution mechanism is not assumed to be reliable, i.e., not all nodes always receive the messages.

B. Distribution of certificate revocations

The second functionality provides mechanism for distributing certificate revocations. CA, the party that revokes certificates, distributes CRLs to the data warehouses periodically, but also has the possibility to issue and distribute a CRL at any time. To prevent congestion the users of the wireless network do not fetch the CRL from the CA. Instead, each service located in any of the data warehouses of the fixed network after mutual authentication can provide it on a user's request. As the services in any case contain sensitive data, we may assume that they are well protected and trustable. If not, security has already been lost. Because of the military character of the network the users are grouped into units which are commonly working together. Therefore it is practical that a user requests a validity statement of the certificates of a group. The service requested forms a message:

Message := { group-id, start-time, (all-valid | (revoked, certificate)| (group-validity, bit-string))}*

The group-id is a two byte id of the group to which belongs the party whose certificate validity is being checked. If needed, the list of the group members can be obtained from the warehouse server. The two byte field *start-time* is the agreed starting time of the operation. It is measured in seconds, starting from 0. As the nodes know the validity duration, the ending time is not coded. The third field in this message has three alternatives: *all-valid=00*, *revoked=01*, *group-validity=10*. For optimization reasons the option *revoked* is used if the number of revoked certificates is less than eight. Otherwise the last option (*group-validity)* is used. After each *revoked* comes an identifier for a user whose certificate is revoked. The identifier is 6 bits, which gives one byte with the prefix *revoked*. In case where more than eight certificates were revoked, the *group-validity* option is used. After *group-validity* follows a bit string where each bit corresponds to a group member: a revoked certificate is coded as bit one and each valid certificate as bit zero. The group of the brigade headquarter network has less than 64 members. The group validity bit string together with the two bites (10) is eight bytes long. *Message* is padded to full bytes with a bit string of ones.

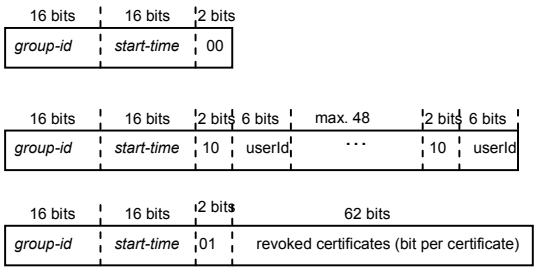

Figure 2. The structure of the distribution of certificate revocation message.

CRL-update

The service sends the *Message* to a user. The message is protected by the shared session key, so the user trusts the message because it trusts the service. This protocol action produces very small messages and can effectively cope with Problem 2.

CRL-update-signed

The service signs the *Message* and sends it to the user. This protocol action produces larger messages, but there is the advantage that the user can pass the message to other users. If the bandwidth of the network allows, this action is the preferred one.

Announce-CRL-DP

If a node in the network has the latest signed CRL (from *Message*) for a member of a group and it notices that communication marked with the flag *partially authenticated* is passing through it, the node sends an *Announce-CRL-DP*

message to the parties in the communication. This message informs the nodes that an up-to-date *Message* can be obtained from this node.

Request-CRL
A user requests *Message* from a service or from another user with this request. A user receiving this request first checks that the requester is announced valid in the CRL it has from the *Message*, then authenticates the user, and if everything is verified, sends a *Message* signed by a service. The requesting user does not need to trust the user who sends the *Message*, only to trust the service.

Send-own-CRL
A user may send *Message* signed by a service. Sending *Message* that contains a validity proof of the user certificate removes the problem of revoked certificates in the *Send-own-certificate* action.

C. Revoking certificates

The third functionality is revoking certificates. The usual way of revoking certificates is that the CA issues a CRL where the certificate is revoked.

Revoke
The *Revoke* action is a command for revoking a particular certificate at any time. It can be given by the CA or by another user to whom the rights of the CA has been transferred. The issuer of *Revoke* distributes the message, in the ad hoc network it is distributed to the whole network.

Doubt
Announcing revocations is a problem in any system using certificate revocation lists. A user who has lost his credentials cannot be authenticated in a strong way before he obtains new certificate. In an ad hoc network he most probably announces the loss through the same network as usually no external communication network is available. As he must revoke his certificates without credentials, he must access as another user. This means that an adversary can equally well try to revoke certificates of any valid user and in this way block the user from the network. The tasks are typically time-critical so even temporary denial of service situations must be avoided. The *Doubt* mechanism is designed so that blocking valid users is difficult.

Any node can announce to the network that it suspects that another node is not trustable. *Doubt* is a one-way protocol requiring sending the message doubt $\{b, k_b\}$ where b is a user id and k_b is the user's public key. There are three principal reasons for a node to send the doubt message. One is the loss of a user's private keys. In such case the user must access some other node and have a valid user of the node issue a *Doubt* on his certificate. Another reason is when a node monitoring traffic passing through notices that another user makes several failed attempts to access services. Services are protected by passwords and if a user accesses the services and becomes refused several times there is good reason to suspect a compromised node. A third reason to issue a *Doubt* is when a group of users have agreed that some node is compromised and decide to force the believe level on that user's certificate to zero in order to exclude the user from the network.

Clear
The nodes keep a believe level for each certificate. Receiving a *Doubt* message lowers the believe level of a certificate. Receiving *Clear* initializes the believe level of a certificate. *Clear* can be issued by the CA or by a user authorized by the CA.

Check-up-question
There are cases when a user has to prove his identity by answering check.-up questions. One of such cases is when the user has lost his smart card and tries to revoke his certificate by sending *Doubt and* the CA cannot authenticate him in the usual way. Another one is when an adversary tries to invalidate a certificate of a valid user. There is no especially good method for solving this problem. The usual way is to store some questions of personal information with answers to an announcement centre (here, the CA) and require a correct answer for revoking the certificate. Personal information is rather easily obtained and temporary PIN codes known to the user and the CA may be slightly stronger. The certificate could be revoked by the user giving this PIN and his name to the CA. The information can only be used once.

D. Authorization

The network has a trusted entity, the CA. As the CA is not always available and as the solution is intended to a military network, a trusted entity capable of revoking certificates, issuing new certificates and clearing doubts is needed. The CA is authorizing an entity to act as a trusted entity by issuing *Transfer-of-rights*.

Transfer-of-rights
The CA can transfer rights to another user. It is outside to scope of the technical solution to guarantee that the user is trustworthy. *Transfer-of-rights* is not distributed to the network. If a user is transferred the rights of the CA and wants to do an operation with CA rights, it must include the transfer message structure to the message so that the other nodes know it is authorized.

VII. CERTIFICATE REVOCATION POLICY RULES

In this section we will derive rules and conclusions concerning the *Doubt* protocol. The presented below activity consist of operations which have a definite starting time and a known duration. At the starting time all parameters are initialized.

The policy language used here is a modified and simplified version of the formalism presented in [17] for public key systems in such a way that it can support the above *Doubt* protocol using a *believe* set that expresses the level of user's trust on other users certificates.

We assume that there is only one CA; called ca. It is directly trusted by all users and all certificates are signed by ca. This assumption reflects the situation that the network is owned by one organization and it is not very large. As organizations today often enter into joint activities, in the

future this assumption will have to be relaxed and certification paths must be allowed.

Expressions

- The expression "*a* `says` *S*" means that the user *a* sends an electronically signed message stating that the expression *S* is true.
- The expression "*a* `transfer` *b*" means that the user *a* transfers rights to the user *b*.
- The expression "*a* `doubts` *S*" means that the user a has sent a message *Doubt* for the expression *S*.
- The expression "*a* `revoke` *S*" means that the user a has sent a CRL revoking the expression *S*.
- The expression "`trust` $\{b,k_b\}$" means that the user trusts the public key k_b to belong to the user *b*.
- The expression "`clear` $\{b,k_b\}$" restores the trusts on the expression that the public key k_b belongs to the user *b*.

Each node *a* keeps a table of values of the type $believe_a[b]$, where each of the values expresses the level that user a believes in validity of user *b* belonging to certain group B. Because of the specific nature of military ad hoc network we assume that the set B covers a unit to which belongs the user's group, but the solution could be generalized by extending the table dynamically whenever a new member accesses the network. Each of the $believe_a[b]$ is initialized to a small number *M* at the time the operation is started. Each node a keeps also a queue $queue_a(b)$ of identities of users who have submitted *Doubt* messages for *b*. The size of the queue is the number of users allowed to lower the believe level of a certificate to zero. This number is expected to be small (2-5), so the memory requirements are not too large. The queue is initialized to zero. The queue is a First-In-First-Out queue. POP takes the first identity to be serviced from the queue and PUSH puts an identity to the end of the queue.

The policy rules are read from up down, that is whatever rule is first filled, its conclusion is taken. The policy rules in each node are as follows:

R1: *ca* says transfer *a*, *a* says S $\Rightarrow$ *ca* says *S*
R2: *ca* says revoke $\{b,k_b\}$ $\Rightarrow$ $believe_a[b]=0$
R3 *a* says doubt $\{b,k_b\}$ $\wedge$ $a \in queue_a(b)$
$\Rightarrow$ GO TO R5:
R4: *a* says doubt $\{b,k_b\}$ $\wedge$ *ca* says $\{b,k_b\}$
$\Rightarrow$ $believe_a[b]--$
PUSH *a* $queue_a(b)$
R5: *ca* says $\{b,k_b\}$ $\wedge$ $believe_a[b]>0$
$\Rightarrow$ trust $\{b,k_b\}$
R6: *service* says $\{b,k_b\}$ $\wedge$ $believe_a[b]>0$
$\Rightarrow$ trust $\{b,k_b\}$
R7: *ca* says clear $\{b,k_b\}$ $\Rightarrow$
$believe_a[b]=M$
WHILE ($\exists c \in queue_a(b)$) POP $queue_a(b)$

VIII. DISCUSSIONS

The *Doubt* mechanism can realize to some extent the idea of the RUMOR protocol proposed by [18]. Using the RUMOR protocol any node may announce its doubt that a node is compromised. This protocol is only a communication mechanism and does not specify how a node concludes that it sends a RUMOR and what a node receiving a RUMOR should do. The *Doubt* mechanism has a smaller scope and is more precisely defined. Any node can send the *Doubt* message indicating that the binding between the user and the public key is in doubt. A node, which receives a *Doubt* message will lower the believe level of this certificate.

There are a number of issues that must be considered in the *Doubt* protocol. Let us assume that there is a compromised host in the network. If it can send a sufficient number of *Doubt* messages and in this way force the believe level on a valid certificate to zero, it can create a denial of service to a valid user. If on the other hand, there is only one valid user that knows that a certificate should be revoked, for instance a valid user who has lost its private key, his announcement should be acted on. A cryptographic shared secret is a too heavy mechanism for this purpose. A reasonable compromise has been achieved with the $queue_a(b)$ mechanism. It is a modification of the simple CHOKe mechanism that has been proposed in [19] for limiting UDP flows.

Announce-CRL-DP has a similar purpose as the *channeling of the update information* in [16]. The main difference is how a node which has an up-to-date CRL notices that it should give the CRL to other nodes. In a military network it is occasionally necessary to restrict communication to the minimum. In this low activity mode of the network, each node monitors traffic passing through it and if it notices that a connection has the *partial-authentication* flag set, it sends *Announce-CRL-DP* to the parties in the communication. This mechanism saves bandwidth and makes detection of the network less likely.

IX. ANALYSES

Let us see how the proposed solution mitigates the problems and attacks listed in section 4. It is not possible to completely remove these threats, only to make the attacks more difficult than some other attacks, such as social engineering, hacking and malicious code. No probability measure can be assigned to such attacks: therefore we will create a measure by a tactical argument. Let us say that an attack is a *serious threat* if a form of the attack that is likely to work can be designed. There usually are errors in the code and with a finite amount of work an adversary can find a successful attack. Many people can be misled and it is possible to find users that are likely to fall on a well-designed social engineering attack. Thus, these attacks make a serious threat. Let us say that an attack is a *minor threat* if using it requires much of good luck. The assumption is that a professional attacker prefers to plan less random attacks using mechanisms that require less of a good luck. The goal is to show that the revocation scheme renders attacks 9-12 and the problems 1-3 to minor threats.

*Proposition 1***.** Attack 9 is a minor threat.

Argument: If an adversary tries to invalidate a certificate of a valid user, he issues a *Doubt* message. There are the following alternatives:

1. The *Doubt* message reaches the CA. The CA sends a *Check-up-question* to both the adversary and to the valid user. The valid user answers correctly. If the adversary answers incorrectly, he cannot invalidate the certificate. If the adversary answers correctly, the issue is investigated further according to the CA policy. In this case the certificate of a valid user is revoked only by good luck.
2. The connection to the fixed network is broken. The adversary must convince *k* other users to send a *Doubt* message in order to force the believe level of a valid user's certificate to zero (the length of the believe queue is in this case *k)*. If *k* is selected sufficiently large, this can be considered to require too much luck.

*Proposition 2***:** Attacks 10 and 11 are minor threats.
Argument: User authentication, with checking of certificate revocations, is required before a request of certificate revocations is accepted by a service or another user. This means that the system must already be compromised if Attacks 10 or 11 succeed.

*Proposition 3***:** Attack 12 is a minor threat.
Argument: The mechanisms for distributing certificate revocations mitigate this attack.

If the CA decides that a certificate is not trustable, it issues a CLR and sends it to services. Revocation is made by the CA if the network is connected to the fixed network. Otherwise, a user which has been transferred CA rights can revoke certificates. It is also possible for a set of users to lower the believe level of a certificate to zero by the *Doubt* mechanism.

If a user notices that his certificate should be revoked, for instance, has lost the private key, he issues *Doubt* on his certificate. If the CA receives *Doubt*, it will pose a *Check-up-question* to the sender of the *Doubt* message and to the user with the certificate to be revoked. If the user can answer the *Check-up-question* correctly, and the user of a compromised node does not answer at all, or answers incorrectly, the CA revokes the certificate. Thus an adversary has a high risk that the revocation of a lost certificate is distributed.

It may be argued that proposition 3 is not quite filled. CA rights are transferred to a single person and thus compromising this person has fatal consequences. This is true, but the military nature of operations implies that the commanders always have an important role in each operation. It is more important to grant a single person CA rights than to protect the network against attacks.

Proposition 4: Problem 1 is a minor threat.
Argument: Functionality *A*. (section 6) allows communication to the extent that the policy rules for partial authentication allow. Serious lack of communication requires an unfortunate combination of events.

Proposition 5. Problems 2 and 3 are minor threats.
Argument: We can argue that the protocol in section 6 can be applied in a way that requires the minimum possible amount of transfer of data from the fixed network to wireless network for certificate revocation purposes.

Due to sensitivity of the material in the services, revocation of certificates is necessary. In the presented method obtaining information that a certificate is revoked requires in minimum one bit of information per certificate (see Fig. 2). Compressed form of this information is the smallest amount of data and the proposed compression method is nearly optimal (it is octet-aligned for efficiency). Encrypting it with a symmetric crypto-algorithm is needed for security purposes and it does not increase the size of data. Therefore *CRL-update* contains for practical reasons the smallest possible amount of data: Problem 2 is then minimized to the extent that it can be. Additionally, as in the designed method load is distributed to services, Problem 3 is not a threat.

X. RUNNING ON UNIDIRECTIONAL LINKS

Let us briefly discuss running the revocation protocol on unidirectional links, as was desired in section 3. In that case all the protocol messages needed for certificate revocation must either be one-way or require answers that can be given on a low bandwidth connection in the reverse direction.

A simple protocol between two terminals *A* and *B* is presented below. It protects against: man-in-the-middle attack, replay, eavesdropping, impersonating *A* and *B*, as well as capturing *A* and *B*.

Let us assume that *A* can reach *B*, but not vice versa. Both *A* and *B* share predefined knowledge - tables of numbered keys (passwords), called *local-keys*. Let us introduce the following notations: K_B is the public key of *B*, $[X]_k$ means that data *X* is encrypted with the key *k*, $signed_A\{X\}$ denotes data *X* with electronic signature made by *A*. Let $cert_A$ denote the certificate of *A* and *key1|key2* mean a bitwise concatenation of two keys, *key1* and *key2*.

In order to send data to *B*, *A* has to authenticate itself. *A* sends to *B* the following data: *Message* signed by a service *S, (*optionally*)* its own certificate, and a signed message encrypted with the public keys of *B*. The message contains a seed for the session key, called here *session-key1*. The session key is obtained by a bitwise concatenation of *session-key1* and one of the numbered keys from the shared table. The message also contains a serial number:

$$M_{auth} = signed_S(Message), (cert_A),$$
$$[signed_A\{session\text{-}key1,\ local\text{-}key\text{-}number,\ serial\text{-}number_A\}]_{KB}$$

$$local\text{-}key = local\text{-}key\text{-}table[local\text{-}key\text{-}number]$$
$$session\text{-}key = session\text{-}key1|local\text{-}key.$$

When *B* receives the message, it must use its private key in order to open it. Thus, *A* can trust that either the message was received by *B*, or the receiver cannot open the message. To obtain the session key the user needs a shared key pointed by the *local-key-number* in the message. The key concatenated with *session-key1* gives the session key to be used in one way communication where *A* sends to *B* messages of the form:

$$M_{data}=[data]_{session\text{-}key}.$$

As there is no return channel, or it has very low bandwidth, strong forward error correction is needed for all messages.

The use of tables with passwords provides access control, as the tables of passwords are protected by the access rights of their users. It also protects against using a stolen or lost terminal. The serial number prevents replays. *A* can either keep track of all users *B* and keep a counter for every *B*, or it can use one counter for all sent packets. In either case, *B* should not receive multiple times a packet with the same serial number from *A*. Both *A* and *B* share the same CA and the public key of the CA and usually the certificates of *A* and *B* are locally stored in both *A* and *B*, thus *B* can verify the certificate. *B* can check if the certificate of *A* is revoked from the *Message*. If needed, *A* also sends its certificate.

Often some low bit channel from *B* to *A* is available, it can be HF radio, some covert channel or the actual signal channel which sometimes works. This low bit channel can be used by *A* for verifying any information, such as that *B* obtained the message. *A* poses a question and gets the answer in some simple code. An example is a code where *A* encrypts with the session key an integer and *B* must reply if the integer is odd or even:

$A \rightarrow B$, $[number]_{session\text{-}key}$
$B \rightarrow A$, $[answer]_{session\text{-}key}$,

$answer = number \bmod 2$.

Correct answers to N questions can be given by chance with probability 0.5^N. For high security N to should be very large, but in this case N=3 or N=5 might suffice: the probability of recognizing the enemy correctly before shooting is only between 80% and 99%, therefore the needed confidence can be of the same range.

This simple protocol for unidirectional links solves revocation of certificates with *Message* from *B.*, section 6. A useful subset of the actions in the protocol in 6 can run on top of this protocol as most of the actions are one-way.

XI. Conclusion

The paper proposes a scheme for certificate revocation in a mobile military ad hoc network. The intended application is from the military side but some of the mechanisms may have wider use. The outlined protocol for revocation of certificates fills the needs of the intended application and gives a sufficient security level for practical purposes.

The solution can be applied in the civilian side e.g. by police and crisis management. There are some conditions that the proposed mechanism assumes. The activity should be organized in operations which start at specific times because the solution synchronizes shared secrets at the start time. If such times are not available, difficulties in synchronizing shared secrets lower the security level of check-up questions and other similar mechanisms. The adversary also should plan his actions in a cost-effective way, rather than the ad hoc way of script kiddies. If this kind of an assumption cannot be made, the division of threats to major and minor threats is obviously not useful.

Acknowledgments The work presented in the paper was supported by the anonymous.

References

[1] M. Naor and K. Nissim, Certificate Revocation and Certificate Update, *IEEE J. on Selected Areas Comm.,* Vol. 18, No. 4, pp. 561-570, 2000.

[2] P. Kocher, On certificate revocation and validation," in Financial Cryptography-FC'98, *Lecture Notes in Computer Science*, Berlin, Springer-Verlag, pp. 172-177, 1998.

[3] S. Micali, Efficient certificate revocation, Tech. Memo MIT/LCS/TM-542b, 1996.

[4] M.E. Nowatkowski and H.L. Owen, Certificate Revocation List Distribution in VANETs Using Most Pieces Broadcast, *Proc. IEEE SoutheastCon 2010*, pp. 238-241, 18-21. March 2010.

[5] J. J. Haas, Y-C. Hu, and K. P. Laberteaux, Efficient Certificate Revocation List Organization and Distribution, *IEEE J. on Selected Areas Comm.* Vol. 29, No. 3, March 2011.

[6] C. B. Popescu, B. Crispo, and A. S. Tanenbaum, A Certificate Revocation Scheme for a Large-Scale Highly Replicated Distributed System, *Proc. 8th IEEE International Symposium on Computers and Communication* (ISCC´03), 2003.

[7] B-H. Li, Y-B. Hou, and Y-L. Zhao, A Scalable Scheme for Certificate Revocation, *Proc. 4th International Conf. on Machine Learning and Cybernetics,* Guangzhou, 18-21, pp. 3852-3856, Aug. 2005.

[8] J. Li, Y. Zhu, H. Pan, and S. Liu, A Distributed Certificate Scheme Based on One-Way Hash Chain for Wireless Ad Hoc Networks, *Mobile Technology, Applications and Systems, 2nd International Conference*, 2005.

[9] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, *IEEE/ACM Tr. on Networking*, Vol. 12, No. 6, Dec. 2004.

[10] S. Chinni, J. Thomas, G. Ghinea, and Z. Shen, Trust model for certificate revocation in ad hoc networks, *Ad Hoc networks*, No. 6, pp. 441-457, 2008.

[11] J. Clulow and T. More, Suicide for the Common Good. a New Strategy for Credential revocation in Self-Organizing Systems, *AMCSIGOPS Operating Systems reviews*, vol. 40,no. 3, pp. 18-21, Jul. 2007.

[12] W. Liu, H. Nishiyama, N. Ansari, and H. Kato, A Study on Certificate revocation in Mobile Ad Hoc Networks, *Proc. IEEE ICC 2011*. 2011.

[13] K. K. Chauhan, and S. Tapaswi, A Secure Key Management System in Group Structured Mobile Ad Hoc Networks, *Proc. WCNIS,* pp. 307-311, 25-27 June 2010.

[14] G. Arboit, C. Crepeau, C.R. Davis, and M. Maheswaran, A localized certificate revocation scheme for mobile ad hoc networks, *Ad Hoc Networks*, No. 6, pp. 17-31, 2008.

[15] Y. Kitada, A. Watanabe, and I. Sasase, On demand distributed public key management for wireless ad hoc networks, *Communication, Computers and Signal Processing*, 2005. PA CRI M. 2005 IEEE Pacific Rim. Conf. 24-26, pp. 454-457, Aug. 2005

[16] M. C, Morogan and S. Muftic, Certificate Management in Ad Hoc Networks, IEEE Database, 2002.

[17] R. Kohlas and U. Mauer, Reasoning About Public-Key Certification: On Bindings Between Entities and Public Keys, *IEEE J. on Selected Areas Comm.*, Vol. 18, No. 4, pp. 551-560, 2000.

[18] C. Candolin and H. H. Kari, Distributing incomplete trust in wireless ad hoc networks, *Proc. IEEE SoutheastCon*, pp. 68-73, 2003.

[19] P. Pan; B. Prabhakar and K. Psounis, CHOKe: A stateless AQM scheme for approximating fair bandwidth allocation, *Proc. IEEE INFOCOM*, Mar. 2000.

Jorma Jormakka (M'98) received the Ph.D. degree in mathematics 1988 from the University of Helsinki. Currently he is an adjunct professor at the Aalto University and at the National Defence University. In the years 2000-2010 he was the professor of command and control systems at the National Defence University. During 2000-2004 he was professor of information technology at the Helsinki University of Technology, and in 1997-1999 he was professor of telecommunications at the Lappeenranta University of Technology.

Henryka Jormakka obtained her Master and PhD degrees in Mathematics from University of Lodz, Poland. She has worked as a research professor of information technology at Technical Research Centre of Finland and at Lappeenranta University of Technology. Her research interests include telecommunication protocols, service architecture, mobility management, agent technology, middleware platforms and security.