PKIX Certificate Status in Hybrid MANETs

Jose L. Muñoz, Oscar Esparza, Carlos Gañán, Javier Parra-Arnau

Universitat Politècnica de Catalunya (Departament Enginyeria Telemàtica)[**]
1-3 Jordi Girona, C3 08034 Barcelona (Spain)
{jose.munoz,oscar.esparza,carlos.ganan,javier.parra}@entel.upc.es

Abstract. Certificate status validation is a hard problem in general but it is particularly complex in Mobile Ad-hoc Networks (MANETs) because we require solutions to manage both the lack of fixed infrastructure inside the MANET and the possible absence of connectivity to trusted authorities when the certification validation has to be performed. In this sense, certificate acquisition is usually assumed as an initialization phase. However, certificate validation is a critical operation since the node needs to check the validity of certificates in real-time, that is, when a particular certificate is going to be used. In such MANET environments, it may happen that the node is placed in a part of the network that is disconnected from the source of status data at the moment the status checking is required. Proposals in the literature suggest the use of caching mechanisms so that the node itself or a neighbour node has some status checking material (typically on-line status responses or lists of revoked certificates). However, to the best of our knowledge the only criterion to evaluate the cached (obsolete) material is the time. In this paper, we analyse how to deploy a certificate status checking PKI service for hybrid MANET and we propose a new criterion based on risk to evaluate cached status data that is much more appropriate and absolute than time because it takes into account the revocation process.

Keywords: Certification, Public Key Infrastructure, Revocation, Hybrid MANET, Risk.

1 Introduction

MANETs (Mobile Ad-hoc Networks) are cooperative networks that allow wireless nodes to establish spontaneous communications. As stated in [1], such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multi-hop topologies which are likely composed of relatively bandwidth constrained wireless links. MANETs may operate in

[**] This work is funded by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004), SECCONET (TSI2005-07293-C02-01), ITACA (TSI2007-65393-C02-02), P2PSEC (TEC2008-06663-C03-01) and, by the Government of Catalonia under grant 2005 SGR 01015 to consolidated research groups.

isolation (stand-alone), or they may have gateways to fixed networks. In this last case, the MANET is called "hybrid". Hybrid MANETs are expected to be deployed as an extension to the traditional infrastructure networks. Also notice that the hybrid behaviour can be temporary due to the situation in which an ad-hoc network may be sometimes stand-alone and sometimes connected to the Internet e.g. a subway network in which a MANET user is connected to the Internet while being at the station and disconnected while traveling. The Hybrid MANET scenario is the one considered in this paper.

On the other hand, trust and security are basic requirements to support business applications in this scenario. The public key scheme is the preferred underlying mechanism to provide security services. In a public key scheme, each participant has two keys: a public key (i.e. known by everybody) and a private key (i.e. secret). The announcement of the public key is performed using a signed document called Public Key Certificate (PKC) or simply "certificate“ that binds the participant with her public key. The entity that signs the certificate is called "certificate issuer" or "Certification Authority" (CA). In the literature, there are several ways of managing security and trust in MANETs based on public key cryptography. These approaches basically differ in the degree of decentralization of the mechanisms deployed for issuing, publishing and revoking the certificates (these approaches are reviewed in further detail in the next section).

In decentralized architectures such as [2] and [3] the nodes inside the ad-hoc network participate in the certification process. On the other hand, in the centralized architecture the certification process is fully controlled by an external CA that is a Trusted Third Party (TTP). In this case the CA digitally signs certificates, ensuring that a particular public key belongs to a certain user and the overall certification process is performed according to a standard and publicly available policy. Each scheme has its application scenario: decentralized approaches are suitable for autonomous MANETs or hybrid MANETs that do not require a centralized enforced certification mechanism while the centralized approach is suitable for hybrid MANETs in which inter-operability with currently deployed centralized public key infrastructures (PKIs) is required.

The problem of using a centralized approach is that current PKIs are designed for wired and well-connected networks, so adopting PKIs for hybrid MANETs is not an easy task. Mobile users are expected to move across different networks. When the user is in a network with connection to the PKI, she can use all the PKI services such as get a certificate, launch a status query, etc. However, users may be disconnected from the PKI when they require a real-time PKI service. In this sense, the certificate status checking is a critical service because applications must decide, at the time of usage, whether a certificate is acceptable or not to perform an action. Proposals in the literature suggest the use of caching mechanisms to let the node itself or a neighbour node to store status checking material (typically on-line status responses or lists of revoked certificates). However, to the best of our knowledge the only criterion to evaluate the cached (obsolete) material is the time. In this paper we propose and formulate a new criterion based on risk to evaluate cached status checking data that is much more appropriate and absolute than time because it takes into account the revocation process. The rest of the paper is organized as follows: Section 2 presents an analysis of the main certification approaches for MANET. Section 3 discusses the main issues that have to be solved in order to adapt current PKI status checking mechanisms to MANET. In Section 4, we present our proposal to evaluate cached status data and, finally, we conclude in Section 5.

2 Certificate Management schemes for MANET

In general, certificate management schemes can be classified as:

- Decentralized. The nodes of the MANET participate either fully or partially in the certification process (see Figure 1.b).
- Centralized. Authorities outside the MANET control the certification process according to a global policy (see Figure 1.a).

In the fully decentralized PKI schemes for MANET, like Capkun et al. [3, 4], the nodes of the MANET themselves issue, publish and revoke the certificates. The certificate management is autonomous and self-organized because there is no need for any trusted authority or fixed server and all the nodes have the same role. In this system, like in PGP (Pretty Good Privacy) [5], each user is her own issuer. Certificates are stored and distributed by the nodes in a fully self-organized manner. Each certificate is issued with a limited validity period and it contains its issuing and expiration times. Before a certificate expires, the owner can issue an updated version of the certificate, which contains an extended expiration time. Authors call this updated version the certificate update. Each node periodically issues certificate updates, as long as the owner considers that the user-key bindings contained in the certificate are correct. Trust is achieved via chains of certificates. The nodes build trust paths certifying from one node to another, as in a friendship circle, forming an authentication ring to achieve the trust relationships with other nodes of the MANET. A decentralized trust management model for pervasive computing environments is presented in [6], where authors overcome the challenges posed by dynamic open environments, making use of the autonomy and cooperative behaviour of the entities.

Another group of public key schemes for MANET is based on threshold cryptography [2]. The idea behind these schemes is to distribute certification duties amongst network nodes. A (k, n) threshold scheme allows the signing private key to be split into n shares such that any k nodes could combine and recover the signing key for a certain threshold $k < n$, whereas $k-1$ or fewer nodes are unable to do so. In this manner, the signing key can be partitioned into n shares and distributed to n nodes using the previous cryptographic technique. For instance, any k of n nodes could then collaborate to sign and issue valid digital certificates or issue status data; whereas a coalition of $k-1$ or fewer nodes would not be able to do so. Notice that this scheme is partially decentralized because it requires an initialization phase in which a centralized authority assigns the role to the n nodes that will act as servers for certificate management. Partially decentralized schemes were first proposed by Zhou and Haas in [7]. This work inspired a practical system called COCA [8] in which a threshold cryptography scheme is implemented for infrastructure-based networks. On the other hand, another system called MOCA [9] extends this idea to ad-hoc networks. In this scheme security is improved by selecting powerful nodes as Certificate Authority servers.

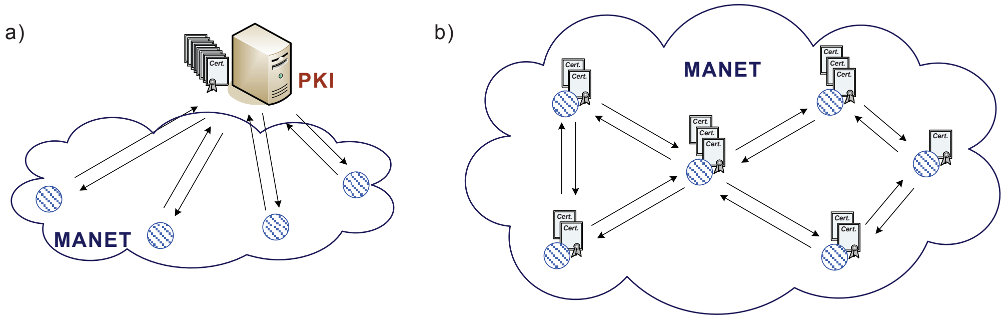

Fig. 1. Centralized and decentralized schemes

Finally, an external public key infrastructure can also be used for the hybrid scenario. In this case, centralized trusted authorities issue, publish and distribute the status (valid/revoked) of certificates according to a well-defined standard methodology. In the Internet, the PKIX [10] is the currently working public key infrastructure. However, PKIX is mostly designed for wired and well-connected networks and adapting the PKIX to the hybrid scenario is a challenging task because MANET nodes are expected to move across different networks, sometimes with on-line connection to the PKIX services and sometimes not. When the user is in a network with connection to the PKI, she can use all the PKI services such as getting a certificate, launching a status query, etc. However, users may be disconnected from the PKIX when they require real-time PKIX services. We discuss the problem of adapting PKI to MANET in more detail in the next section.

3 Adapting PKIX to MANET

The local validity of the certificates in the decentralized approaches may restrict their usability in the hybrid scenario. In this sense, the PKIX approach is suitable for hybrid MANETs that require support for mobility maintaining a centralized enforced certification mechanism and also interoperability with currently deployed PKIs. However, the original design of the PKIX assumes that the user can access at any time to the entities of the infrastructure which is true for wired well-connected networks but not for our scenario.

The first problem that we have to face is the certificate acquisition. A permanent connection of the client to the infrastructure cannot be assumed so the solution is to choose relatively long validity periods for the certificates. The idea is that the user has to pass an initial certification process before she can start operating in the MANET. Once the user has its credential, she can operate in the hybrid scenario without further interaction with the PKI (at least interaction is not required for a quite long time). This way of issuing the certificates can be assumed as an initialization phase equivalent to the initialization phase of the partially decentralized scheme in which the shares are delivered.

On the other hand, a certificate might be revoked (invalidated) prior to its expiration. Among other causes, a certificate may be revoked because of the loss or compromise of the associated private key, in response to a change in the owner's access rights, a change in the relationship with the issuer or as a precaution against cryptanalysis. The revocation policies determine how the status of the certificates is distributed to the end users. So the PKI is responsible for the certificates not only at the issuing time but also during all the certificate's life-time.

The problem is that PKIX explicit revocation systems were designed for wired and well-connected networks in which repositories and responders have a well-known network address and are always available to users. However, MANETs are dynamic environments in which network topology changes randomly and in which mobile users continuously join and leave the network. Therefore, new mechanisms are necessary to distribute explicit status data in MANETs. Proposals in the literature suggest the use of caching mechanisms to address these problems.

Caching schemes allow to manage arbitrary disconnections between the users and the sources of the status data service. Disconnections are alleviated by storing copies of status data (lists of revoked certificates or on-line responses) in the nodes of the ad-hoc network. These copies are obtained when connection to the infrastructure is available. In general, an ad-hoc caching scheme for any service has four different kinds of nodes [11]: server-nodes, client-nodes, caching-nodes and intermediate-nodes (see Figure 2). For the status checking service:

- *Server-nodes*. These nodes have "always updated data“ to offer the status checking service. The server-node has a permanent connection to the certification infrastructure in order to have always fresh status information. Typically, a server-node is an Access Point connected to both to a MANET and to the fixed network.
- *Client-nodes*. These nodes require the status checking service. A *service discovery* mechanism has to be provided to the client so that she can find a node in the network that provides the service.
- *Caching-nodes*. These nodes have cached data and therefore they may also provide the status checking service. A client-node in the absence of connectivity to a server-node or because of performance issues can connect with a close caching-node to obtain the service with cached status data (perhaps quite obsolete data).
- *Intermediate-nodes*. These nodes forward the packets among client and server nodes. They may also store the path to a service provider (whether a server-node or a caching-node) together with service parameters such as data size, the service expected Time-To-Live (TTL), number of hops to reach the provider etc.

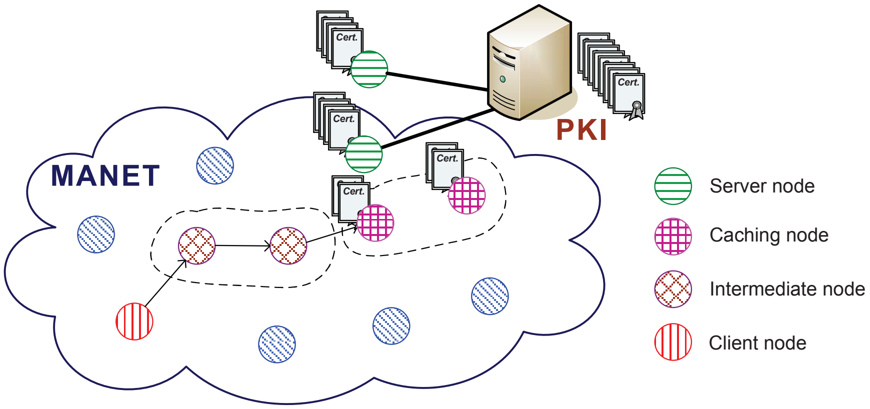

Fig. 2. Four different kinds of nodes in caching schemes.

In the literature we can find some proposals that apply the previous ideas to adapt the PKI status checking standards CRL [12, 13] and OCSP [14] to the MANET. A CRL is a black list with the identifiers of revoked certificates. The integrity and authenticity of the CRL is provided by an appended digital signature. On the other hand, OCSP is a protocol to make the status of certificates available through a request/response mechanism. The OCSP server is called responder and provides signed responses to clients. Next, we give our point of view about this adaptation and we briefly review some remarkable works about this in the literature.

In the case of CRL, server-nodes are nodes that can maintain a stable connection to PKI repositories in order to get the most updated CRL. A caching-node is a node that is willing to collaborate in the certificate status checking service and that has enough cache capacity to store a CRL copy. The caching-node responds to the status requests of client-nodes in the MANET. Notice that a client-node that acquires a valid CRL copy can become a new caching-node. Furthermore, a caching-node that moves to another MANET can collaborate in the new network to provide the service. In this sense, user's mobility helps the status checking service. In [15], the authors investigate the feasibility of using flooding to distribute CRL information in MANETs by simulation. They conclude that the two major factors for flooding to work smoothly are the number of nodes and the communication range. In [16] a MANET cooperative mechanism for certificate validation is presented in order to overcome both the lack of infrastructure and the limited capabilities of the nodes. This solution is based on an extended-CRL where the repositories can build an efficient structure through an authenticated hash tree.

Regarding OCSP, server-nodes are responders. We can consider that there are only responders placed in the PKI (fixed-responders) or we can consider the possibility of having responders implemented in a mobile node that can be part of a MANET (mobile-responders). Despite this possibility, we discourage the use of mobile-responders because they are server-nodes and as such they are supposed to have updated status data. A server-node for certificate status checking must have connectivity with PKI repositories or fixed-responders to get updated status data but this connectivity is not always guaranteed in a MANET. On the other hand, a responder is a trusted authority so it has a private key that has to protect against intruders. In our view, it makes no sense having a server-node that is exposed to attacks and that may not have useful data. Furthermore, in general, increasing the number of trusted authorities in a system is not desirable, the less number of trusted authorities, the less is the probability of having a private key compromised. Besides, if mobile-responders are used, it is necessary to define a mechanism to trust them which is not trivial. With respect caching-nodes, they store OCSP responses issued by server-nodes and distribute them to client-nodes when they detect a request that fulfils freshness requirements. In [17, 18], there is a complete proposal called ADOPT (Ad-hoc Distributed OCSP for Trust) that describes a caching scheme for OCSP in MANET.

4 Evaluation of cached status data based on Risk

As explained in the previous section, caching and discovery mechanisms are necessary to manage the situation in which a user is not able to reach a PKI status data server. When a disconnection happens, the client-node uses service discovery to find a caching node. Then, the node obtains a cached version of available status data and finally, the node decides what to do with the data. In this sense, the CA issues status data bounded by two time-stamps:

- *thisUpdate.* Instant at which status data have been issued.
- *nextUpdate.* Instant at which updated status data are expected to be issued.

Let us define T_s as the issuing interval of status data (1).

$$T_s = nextUpdate - thisUpdate \tag{1}$$

As data in status responses are time-stamped, users can get an idea about how fresh is the status of a certificate by looking at the *thisUpdate* parameter of the response and, finally a user can take a decision about whether operate or not with a certain certificate. According to [19] the time is the only criterion to help the user to take this decision and to the best of our knowledge this is the only criterion proposed in the literature. However, this is a poor criterion that can be enhanced. In this section, we propose other parameter rather than time to take this decision.

First of all, let us illustrate why time is a poor parameter for our purposes. For instance, consider a status response issued a couple of hours ago. We may wonder: *is it fresh or not?* The answer is obviously that *"it depends"*. Two hours may not be considered a long time if there are a couple of revoked certificates every month but this period can be considered quite long if there are two new revoked certificates per hour. Moreover, a scenario with millions of issued non-expired certificates is not the same as a scenario that has hundreds of certificates. In the former, a couple of new revoked certificates is not so relevant while in the latter a couple of new revocations is quite important. As a conclusion, we need a parameter that considers all these aspects. For this purpose, we define a risk function that aids the user to decide whether to trust or not a certificate. We formally define the function *risk* $(r(t))$ as the *probability of considering a certificate as a valid one when the real status known by the PKI is revoked at time t.*

To find an analytical expression for the risk function we first need to analyse the certificate issuing process. Certificates are issued with a validity period T_c. Obviously $T_c >> T_s$, for instance T_c can be a year while the period of status data issuing can be an hour. The number of *non-expired certificates* $(N(t))$ -including revoked and non revoked certificates- is a stochastic process whose mean value at instant t depends on the certificate issue and certificate expiration processes. It is assumed that the elapsed time since issuing until expiration (T_c) is a constant value for all certificates. Therefore, the expiration process is the same as the issuance process elapsed T_c time units. This process is defined by the certificate issue rate λ_c, which matches with the certificate expiration rate. Hence the mean value of *non-expired certificates* in steady state is the mean quantity of issued certificates before the expiration process begins.

$$E[N(t)] = N = \lambda_C T_C, \quad t > T_C \tag{2}$$

On the other hand, there is a group of *revoked non-expired certificates*, that is to say, certificates that have a valid validity period but that have been revoked prior to the expiration date and, therefore they are included in the black list. The subset of *revoked non-expired certificates* is included in the set of *non-expired certificates* and the cardinality of that set, $R(t)$, is a stochastic process that it is typically modelled [20] as a fraction or percentage ($p(t)$) of the non-expired certificates (3).

$$R(t) = p(t)N(t) \quad with\ p(t) \leq 1 \tag{3}$$

Assuming that both processes are independent and using expected values:

$$E[R(t)] = E[p(t)]E[N(t)] \tag{4}$$

$$R = pN \tag{5}$$

We further model the expected percentage of revoked certificates as directly proportional to the certification time T_c (6).

$$p = p'T_c \tag{6}$$

This means that larger certification periods will imply more percentage of revoked certificates. On the other hand, smaller certification periods mean less probability of a certificate being revoked during its life-time and therefore low percentage of revoked certificates. Then, the mean value of the *revoked non-expired certificates* can be expressed as:

$$R = p'\lambda_c T_c^2 \tag{7}$$

We have modelled the issuing and revoking processes of the overall system. However, our goal is to model the risk from the point of view of the user, that is to say, we want to find the probability of considering a certificate as a valid one when the real status known by the PKI is revoked.

Let us assume, without loss of generality, that at instant $t_0 = thisUpdate$ a user gets the current black list of revoked certificates from the PKI. Using this list, the user can split the set of *non-expired certificates* into *revoked certificates* and *not revoked certificates*.

Next, we need to define the subset of *operative certificates* as the group of *non-expired certificates* for which the last status known by a user is *not revoked*. Notice that the PKI may know that a certificate considered operative by a user is in fact revoked. However, due to the MANET conditions it is impossible to communicate this situation to the user.

Now, let us assume that the user is not able to connect to the infrastructure any more. As time goes by the set of *operative certificates* will include revoked certificates and the user will need to take decisions about using an operative certificate assuming a certain risk. The *risk function* $r(t)$ can be evaluated as the ratio between the number of *unknown revoked operative certificates* ($R'(t)$) and the number of *operative certificates* ($N'(t)$) as shown in equation (8).

$$r(t) = \frac{E[R'(t)]}{E[N'(t)]} \tag{8}$$

$N'(t)$ (*number of operative certificates*) can be defined as the number of certificates that were not included in the last black list obtained by the user (were not revoked before t_0) and that they have not expired at t. Included in the set of *operative certificates* there is the subset of *unknown revoked operative certificates.* The cardinality of this subset $R'(t)$ is the number of *operative certificates* that are revoked at instant t, that is, they are revoked but this fact is unknown to the user.

At $t_0 = \mathit{thisUpdate}$ the set of *operative certificates* is the same that the set of *not revoked certificates* and, since the user has the same information that the PKI so there is no risk ($r(t_0) = 0$). Besides

$$E[N'(t_0)] = (1 - p)N \tag{9}$$

$$E[R'(t_0)] = 0 \tag{10}$$

At the instant $t_0 + T_C$ all the certificates included in the black list will be expired. This means that all *non expired certificates* will be *operative*, and any revoked certificate will be unknown to the user. The *risk* at this moment can be expressed as (11).

$$r(t_0 + T_C) = \frac{E[R'(t_0 + T_C)]}{E[N'(t_0 + T_C)]} = \frac{E[R(t_0)]}{E[N(t_0)]} = p \tag{11}$$

To evaluate the function risk between t_0 and $t_0 + T_C$ we have to observe the processes $N'(t)$ and $R'(t)$ in this interval. After t_0 the variation of the number of *operative certificates* ($N'(t)$) depends on these factors:

- Increases because of the new issues.
- Decreases because of the expiration of operative certificates issued before instant t_0 (the certificates issued later do not expire in the considered interval).

The issuance rate is λ_c that is the same as the expiration rate. But notice that not all expirations concern to *operative certificates*. A fraction p of the expirations corresponds to *revoked non expired certificates*, and the other fraction $1 - p$ corresponds to *operative certificates*. Then the expiration rate of *operative certificates* is $(1 - p)\lambda_c$ (see Figure 3).

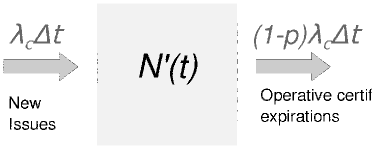

Fig. 3. Evolution of operative certificates

Considering the evolution of the set of *operative certificates* we can evaluate its expected cardinal (12).

$$E[N'(t)] = E[N'(t_0)] + \lambda_C(t - t_0) - (1 - p)\lambda_C(t - t_0) \tag{12}$$

Using (9) we obtain.

$$E[N'(t)] = (1 - p)N + p\lambda_C(t - t_0) \tag{13}$$

Finally, we need an expression for the set of *revoked operative certificates*. This set is the intersection of the set of *operative certificates* and the set of revoked certificates as shown in the Figure 4.

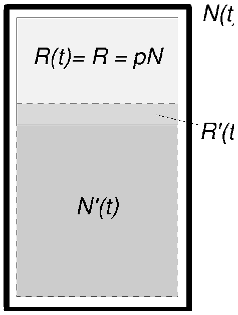

Fig. 4. Sets of certificates

Hence we can express the cardinality of these sets using the following expression.

$$N(t) = R(t) + N'(t) - R'(t) \tag{14}$$

Therefore,

$$R'(t) = R(t) + N'(t) - N(t) \tag{15}$$

We obtain the expected value of the number of revoked operative certificates using (15), (2), (5) and (13).

$$E[R'(t)] = p\lambda_C(t - t_0) \tag{16}$$

To obtain the *risk* function we use the expressions (13), (16) and the expression of its definition (8).

$$r(t) = \frac{p(t - t_0)}{(1 - p)T_c + p(t - t_0)} \tag{17}$$

The previous expression is valid for instants of time $t \;\; \epsilon \;\; t_0 \leq t \leq t_0 + T_c$ and fulfils with the expected results of expressions (10) and (11). Notice that the risk function allows a user to compute the probability of considering a non-expired certificate as non-revoked when the real status known by the PKI is revoked.

On the other hand, it is remarkable that unlike time which is a relative parameter, the risk function gives the user an absolute parameter to aid her taking the decision of trusting or not a particular certificate. This decision must be taken when the user is disconnected from the infrastructure and therefore it is taking into consideration cached (obsolete) status data.

Finally, the risk function should be used as follows:

- In first place, the CA signs the status data with the two standard time-stamps (*thisUpdate* and *nextUpdate*) but it also adds the current parameter p. The CA can calculate this parameter because it knows the current number of issued non-expired certificates and the current number of non-expired revoked certificates.
- When the user has to evaluate status data, she knows T_c as this is the certification period included in her certificate.
- Then, the user obtains p from the status data.

- Next, the user can compute the risk at current time t by replacing t_0 with *thisUpdate* in the risk function.
- Finally, the user can take a decision about a target certificate with the risk value computed.

5 Conclusions

Decentralized certification architectures for MANET such as self-organized PKIs and PKIs based on threshold cryptography generally provide certificate validation mechanisms inside the MANET. However, local validity of the certificates and inter-operability with currently deployed PKIs may restrict their usability in an hybrid MANET scenario. If a centralized certification infrastructure such as PKIX is used, then certificate validation becomes one of the main problems. This is because users need to ensure at the time of usage that the certificate they are relying upon has not been revoked but at the same time trusted servers of PKIX may be unavailable. Besides, standard status checking mechanisms of the fixed network are not directly usable because they are designed for always connected users.

In this sense, caching schemes allow to manage arbitrary disconnections between the users and the sources of the status data service. Disconnections are alleviated by storing copies of status data (lists of revoked certificates or on-line responses) in the nodes of the ad-hoc network. These copies are obtained when connection to the infrastructure is available. On the other hand, a service discovery mechanism is necessary to find the nodes that have cached material. In this paper, we have reviewed and analysed all these issues for adapting the standard PKIX status checking mechanisms to hybrid MANET.

Despite the caching scheme allows the users to obtain status data during disconnections, the cached status data is likely to be outdated. When using cached status data a node could operate with a revoked certificate considering it is a valid one. In this paper, we have presented a novel scheme which provides users within the MANET with an absolute criterion to determine whether to use or not a target certificate when updated status data is not available. By taking into account information about the revocation process, users can calculate a *risk* function in order to estimate whether a certificate has been revoked while there is no connection to a status checking server. Finally, it is also worth to mention that this new criterion can be applied to other networks than hybrid MANETs if these networks are based on an off-line explicit revocation scheme.

Abbreviations

ADOPT Ad-hoc Distributed OCSP for Trust.
CA Certification Authority.
COCA Cornell On-line Certification Authority.
CRL Certificate Revocation List.
MANET Mobile Ad-hoc Network.
MOCA Mobile Certificate Authority.
OCSP On-line Certificate Status Protocol.
PGP Pretty Good Privacy.
PKI Public Key Infrastructure.
PKIX Public Key Infrastructure (X.509).
TTL Time-To-Live.
TTP Trusted Third Party.

References

1. S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501 (Informational), January 1999.
2. Y. Desmedt and Y. Frankel. Threshold cryptosystems. in advances in cryptology—crypto'89. In *the Ninth Annual International Cryptology Conference*, volume 435 of *LNCS*, pages 307–315. Springer-Verlag, 1989.
3. S. Capkun, L. Buttyan, and J.P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2003.
4. J-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC'01)*, 2001.
5. J. Zsako. PGP Authentication for RIPE Database Updates. RFC 2726 (Proposed Standard), December 1999.
6. F. Almenárez, A. Marín, C. Campo, and C. García. Managing ad-hoc trust relationships in pervasive environments. In *Proceedings of the Workshop on Security and Privacy in Pervasive Computing SPPC*, 2004.
7. L. Zhou and Z.J. Haas. Securing ad hoc networks. *IEEE Networks*, 13(6):24–30, 1999.
8. L. Zhou, F.B. Schneider, and R.V. Renesse. Coca: A secure distributed on-line certification authority. *ACM Transactions on Computer Systems*, 20(4):329–368, 2002.
9. S. Yi and R. Kravets. Moca: Mobile certificate authority for wireless ad hoc networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP'O2)*, 2002.
10. Pkix chapter of the ietf. www.ietf.org/html.charters/pkix-charter.html.
11. L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1):77–89, 2006.
12. R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459 (Proposed Standard), January 1999. Obsoleted by RFC 3280.

13. S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820 (Proposed Standard), June 2004.
14. M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard), June 1999.
15. H. W. Go, P. Y. Chan, Y. Dong, A. F. Sui, S. M. Yiu, Lucas C. K. Hui, and Victor O. K. Li. Performance evaluation on crl distribution using flooding in mobile ad hoc networks (manets). In *ACM Southeast Regional Conference archive. Proceedings of the 43rd annual southeast regional conference*, volume 2, pages 75–80, Kennesaw, Georgia, 2005.
16. J. Forné, J. L. Muñoz, O. Esparza, and F. Hinarejos. Certificate status validation in mobile ad hoc networks. *IEEE Wireless Communications*, 16(11):55–62, 2009.
17. G. F. Marias, K. Papapanagiotou, V. Tsetsos, O. Sekkas, and P. Georgiadis. Integrating a trust framework with a distributed certificate validation scheme for manets. *Wireless Communications and Networking*, 1155(10):1–18, 2006.
18. G. F. Marias, K. Papapanagiotou, V. Tsetsos, O. Sekkas, and P. Georgiadis. Integrating a trust framework with a distributed certificate validation scheme for manets. *EURASIP Journal on Wireless Communications and Networking*, 2006(2):1–18, 2006.
19. A. Deacon and R. Hurst. The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments. RFC 5019 (Proposed Standard), September 2007.
20. A. Arnes. Public key certificate revocation schemes, February 2000. Queen's University. Ontario, Canada. Master Thesis.