Certificate Management in Ad Hoc Networks

Matei Ciobanu Morogan, Sead Muftic
Department of Computer Science, Royal Institute of Technology
[matei, sead] @ dsv.su.se

Abstract

Various types of certificates are basic tools of modern cryptography and networks security. They are used in various protocols, in the form of public key identity certificates, binding a key to its owner or in the form of attribute certificates, being a proof of rights and capabilities of their owner. Management of certificates (creation, distribution, verification, and revocation) is dependent on a certification infrastructure comprising various certification authorities, protocols, and policies.

In this paper we consider usage and management of certificates in open, ad hoc networks. Ad hoc networks differ from fixed, wired networks in several important aspects, one of them being that access to the Internet is not always available. This significantly influences certificate management protocols since online access to various certificate system resources (CA certificates, CRL, etc) is not always available. In this paper we specify security requirements and constraints in such environments and outline some potential solutions for adaptation of certificate management protocols to these new network environments.

1. Introduction

Ad-hoc proximity networks differ from wired local area networks in several important aspects [1]. One aspect is that access to the global network can not always be established, either because of lack of coverage or because of high cost of communications. Furthermore, devices can remain unconnected for extended periods of time. Therefore, transactions and control mechanisms must be as independent from outside resources as possible. Another specific aspect is unpredictability of the network, that is, devices which form it are not preregistered or known in advance. Network is formed dynamically by users and devices in a proximity.

The possible unavailability of access to the global network has an important influence on the usage of public key certificates. Certificates rely on a Public Key Infrastructure [2] consisting of a hierarchy of Certification Authorities (CA:s). Usage, management and validation of certificates usually assumes that a CA is available online. One of the most popular standards for public key certificates is based on in the X.509 standard [3].

Previous work has been done on certification and authentication in ad hoc networks. [4] discusses the problem of certification in ad hoc networks versus the Internet, and offers some solutions to establish trust without using certificates. [5] discusses different protocols for authentication in ad-hoc networks, also without the use of certification, but based on transmitting authentication information through a separate link (IR or contact between the nodes). [6] describes a system for ad hoc networks similar to the PGP certification system ([7]) where certificates are issued, stored and distributed by the users.

In this paper we explore techniques to relax the requirement of constant access to a CA, so that it is possible to use, manage and verify certificates even in situations where access to the Internet is not available.

2. Scenario

We will use the following scenario for exemplification.

Several devices form an ad hoc network in a place where access to the Internet is not available. Some of these devices may have been offline for a longer period of time. The assumption is that all of them have certificates which they have previously obtained from CA:s in the same certification hierarchy.

In the ad hoc network, the devices need to communicate and share information with various security requirements. They have to authenticate each other and to verify each other’s authorization to access the information. Transactions may also take place between the devices, like electronic payment transactions or signing digital contracts.

3. Online operation of certification systems

Prior to using them, certificates need to be obtained from a Certification Authority server. This is done by creating and submitting a certificate request to the CA. The CA verifies the request and issues the certificate, making it available to the requesting client. By means of a shared secret between the client and the server, the client can now fetch the certificate.

Once obtained, the certificates are used in communication with other entities.

When receiving the certificate of a peer, one needs to validate it to ensure its correctness. For this, access to an online CA is needed for several actions:

- obtaining the certificate chain up to a known CA
- verifying that the certificate has not been revoked
- getting updated policy information concerning the usage of the certificate

A certificate chain is needed in the case that the certificate to be verified comes from an unknown CA. In that case, the certificates of all the CA:s in the hierarchy up to a CA that is known and trusted are needed (Figure 1). The top CA of a certification hierarchy is known and trusted by all entities in that hierarchy, so usually a certificate chain contains all certificates up to the top CA.

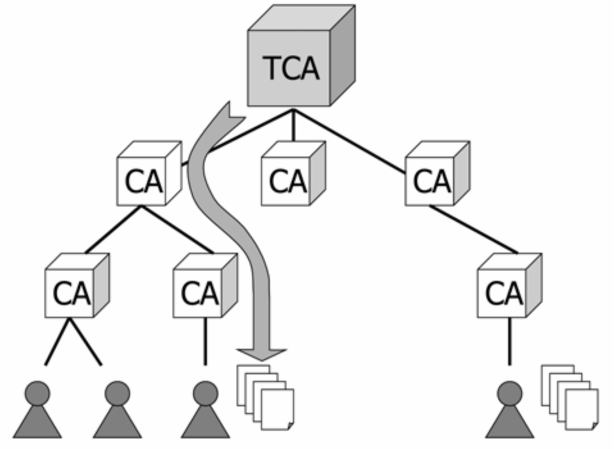


Figure 1. Certificate chain

A certificate may become invalid before its expiration date, for example if the corresponding private key has been revealed [8]. In this case the certificate must be revoked, so it is no longer accepted as valid by anyone, even though it is signed by a trusted CA and is still in its validity period. This is done by adding the certificate to a Certificate Revocation List (CRL) [9]. These lists are distributed through the entire certification hierarchy so that all Certification Authorities (CA) get them. When a client needs to validate a certificate, it contacts a CA to get the latest CRL and then checks if the certificate has been revoked or not [10].

The policy of usage of the certificates may change over time as well. These changes must be communicated to all entities in that certification hierarchy.

4. Solutions for offline operation

Since the devices in an ad hoc network are not always offline, they can submit certificate requests and fetch the issued certificates from the CA servers while being online. The certificate functions that need to be available offline are the functions to validate a peer certificate.

To validate certificates offline, all information listed in section 3 must be available locally.

Certificate chains are not an issue, since they can easily be stored by each entity together with their own certificates. When sending the certificate to another entity, the certificate chain is sent as well. Of course, since a certificate chain can be much larger in size than a single certificate, one can choose to send it only when it is needed, that is, when the peers do not have previous knowledge of the issuing CA.

A more difficult issue is the CRL and the policy information updates. These are updated at regular intervals, and are extremely important to the security of any application based on certificates. One does not want devices that have been offline for a while to accept certificates that have been revoked under the time, or to rely on outdated security policies. We call the CRL and the security policy changes the *certification information updates*.

When online, a certificate can be validated instantly, and the circulation of information regarding the revocation of a certificate or changes in security policy is fairly quick. Therefore, if a certificate is revoked in Sweden, it is probable that it will no longer be usable in Florida by the time its possessor gets there.

When offline, this situation changes. Even if a client has downloaded a CRL the last time it was connected to the Internet, it will expire quite soon, and there is no guarantee that the certificate it receives during a transaction has not been revoked in the meantime. Therefore, there is always a risk in validating certificates without online access. The risk becomes greater as the age of the available CRL increases, as more certificates will be revoked under that time.

For some applications this risk is not acceptable, but for others, the advantage in usability and convenience that is gained by allowing the CRL and security policy information to be somewhat outdated is worth the risk. We call the period that is acceptable, for an application, from the last update of the CRL and security policy, the *grace period*. Ad hoc aware applications that use certificates should therefore specify in their own security policies the grace period that is acceptable for a particular transaction. For example, monetary transactions up to a certain amount may be possible with a grace period of one week, while buying a car would require online access. Also, sharing certain sensitive documents would be possible with a grace period of a few hours, while for less sensitive documents the grace period may be one day.

This requires modifications to the certification module used by the devices, so that an application can request not only the validation of a certificate, but also specify the allowed grace period. The certification module stores the time of the last update and returns a negative response if the interval is longer than the grace period accepted by the application, indicating this reason.

This allows for a certain offline period during which applications can still use and verify certificates, according to their security policies.

5. Channelling update information

By adding a grace period to the policies of applications that use certificates, it becomes possible to use and verify certificates without direct access to a CA server, if the last contact with a CA server was within the grace period of a particular application. However, if an entity has not had direct contact with a CA longer than the grace period, usage of certificates is impossible, even if other entities in the ad hoc network have the latest updates.

This is unnecessary limiting, since the information is there, in the ad hoc network, and we only need a secure way to spread it to the devices that don't have it.

We call this technique *channeling of the update information*. It works in the following way. While online, a user obtains a signed CRL and security policy update from a CA, together with a timestamp (Figure 2). In the ad hoc network, the signed CRL and security policy update can be distributed to other devices, who can verify the signature of the CA to ensure their validity, and check the timestamp to see if they are within the required grace period.

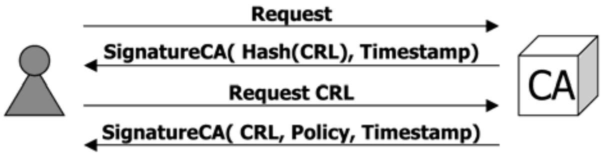

Request
SignatureCA(Hash(CRL), Timestamp)
Request CRL
SignatureCA(CRL, Policy, Timestamp)
CA

Figure 2. Obtaining the CRL from the CA

The first part of the protocol looks like this:

- When in contact with a CA, the certification client asks for the current CRL.
- The CA server responds with the hash of the current CRL together with a timestamp, signed by the CA.
- The client checks the signature of the CRL (which can be a serial number or the hash of the CRL) and compares it to the one it already has. If the CRL differs, it requests the new CRL from the CA server.
- The server, if requested, sends the current CRL to the client. If there is any security policy update, it is also added to the package. This package is cryptographically signed by the CA server or by some other CA in the hierarchy.

By now each client has two items: a signed CRL (or a package containing both CRL and a security policy update), which can be verified by any entity in the certification hierarchy and a timestamp for that CRL, that shows the time when it was downloaded. Note that it is not necessary to download the entire CRL or policy update each time, only when it changes. This scheme can be optimized to further reduce network traffic by using delta-CRLs ([11]), CRL lists that only contain the changes from the base CRL.

This protocol is not loading the CA too much cryptographically, since a new timestamp only needs to be computed at certain intervals and than can be sent to all clients requesting updates.

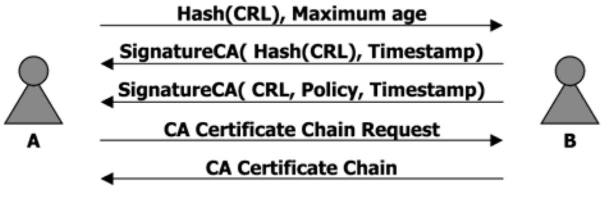

Hash(CRL), Maximum age
SignatureCA(Hash(CRL), Timestamp)
SignatureCA(CRL, Policy, Timestamp)
CA Certificate Chain Request
CA Certificate Chain
A
B

Figure 3. CRL exchange in ad hoc network

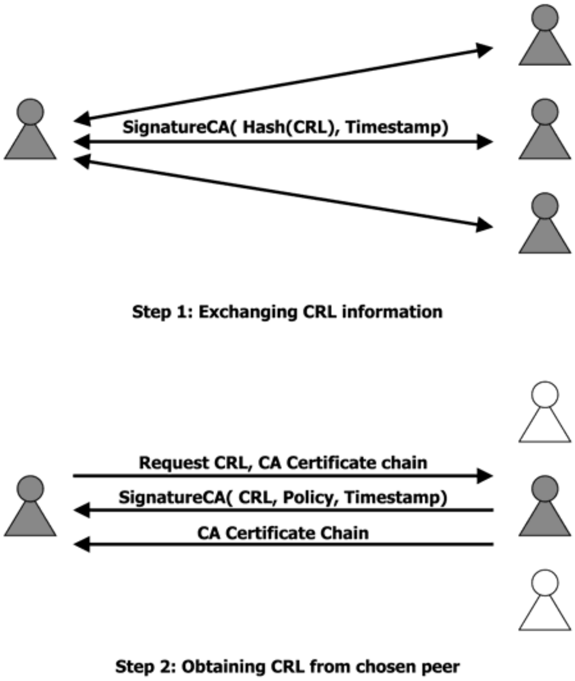
When in an ad hoc network, peers can signal that their latest CRL is not within the grace period for a certain transaction (Figure 3). In that case the following protocol for exchanging the CRL can be performed between the device that does not have the required CRL (A) and the device that may have it (B):

- A sends to B the request stating the maximum age of the CRL, together with the signature of A's current CRL.
- If B has a CRL that is new enough, it compares the signature of A's CRL with the signature of its own. If the CRLs are the same, it sends to A the timestamp of the CRL, signed by the CA. If the CRLs are different, it also sends the CRL itself.
- A can now verify the signatures of the CRL and of the timestamp, and compare the signature of the CRL with the one contained in the timestamp. If the CA that issued the CRL is not known, it requests the corresponding certificate chain from B. Otherwise, the protocol is finished.
- B sends the CA certificate chain to A, if necessary.

The protocol can be adapted for group usage. An initial step is needed to find out which device has the newest CRL. After that, the exchange takes part in a similar manner with the two-party protocol.

One device may attempt to sabotage the process by pretending to have a CRL that is very recent, and then not sending it, thereby blocking other devices that have older, but still usable CRL:s. To prevent this, hashes of the CRL, together with a timestamp and signed by a CA, are used during the initial step (Figure 4). Also, if the CRL can not later be obtained from the device that was chosen during the initial step, the protocol is repeated using the second best result of the initial step.

- Each device informs the others about the CRL it has. This is done by sending a hash of the CRL with a timestamp, signed by a CA. After receiving the timestamps from all other devices, each device decides which one to request.
- Devices compare the hash of the newest CRL with their existing one. If they are the same, the process is finished, as they have a current timestamp for their CRL, signed by a CA. If necessary, they request the certificate chain for the CA that signed the timestamp. If the hashes differ, they request the new CRL, and also, if necessary, the certificate chain of the CA that signed it.
- The device that got chosen during the initial step distributes the CRL to the devices requesting it. If a device requires it, it also adds the certificate chain of the CA that signed the CRL.
- Recipients verify that the signature is correct. If verification does not succeed, they repeat the process with the second best result of the first step.


SignatureCA(Hash(CRL), Timestamp)
Request CRL, CA Certificate chain
SignatureCA(CRL, Policy, Timestamp)
CA Certificate Chain

Step 1: Exchanging CRL information

Step 2: Obtaining CRL from chosen peer

Figure 4. Group protocol to exchange CRL

The group protocol is more efficient than repeating the two-party protocol for each device, since information regarding the CRL is already distributed during the first step. While in the two-party protocol it is the owner of the CRL that compares it with the client's hash and decides whether to send it or not, here the clients obtain that information in the first step, and then only contact the owner in the case their CRL is different.

6. Conclusions

With the adaptations and modifications described above, usage and validation of public key certificates becomes more flexible and robust in ad hoc networks.

Considering the scenario presented in section 2, the devices can verify each other's certificates by using stored CRL:s and certificate chains. The devices that have been offline for a longer period of time can request updated CRL:s and timestamps from the others. Information can then be shared, or transactions performed, if the maximum grace period allowed by each application has not been exceeded.

Some of the constraints of normal certificate usage in ad hoc networks are in this way removed. Also, a balance between security and utility for each application can be specified as a part of the security policy.

The described scenario assumes that certificate chains of all users in an ad hoc network meet at some CA, what implies that all users belong to the same global PKI. The case when users belong to different PKI:s is the subject of current research.

7. References

[1] G.-C. Roman, P. J. McCann and J. Y. Plun, "Mobile UNITY: Reasoning and Specification in Mobile Computing", *ACM TOSEM, VOL. 6, no. 3*, July 1997, pages 250-282.

[2] A. Arsenault and S. Turner, "PKIX Roadmap", *Internet Draft,* IETF PKIX working group, work in progress, October 1999.

[3] R. Housely, W. Ford, W. Polk and D. Sodo, "Internet X.509 Public Key Infrastructure", *Internet Engineering Task Force Draft*, PKIX Working Group, work in progress.

[4] L. Eschenauer, V. D. Gligor and J. Baras, "On Trust Establishment in Mobile Ad-Hoc Networks", *Proc. of the Security Protocols Workshop*, Cambridge, UK, April 2002

[5] D. Balfanz, D.K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in adhoc wireless networks", *In Symposium on Network and Distributed Systems Security (NDSS '02)*, San Diego, California, February 2002

[6] J.-P. Hubaux, L. Buttyftn, and S. Qapkun, "The quest for security in mobile ad hoc networks", *In ACM Symposium on Mobile Ad Hoc Networking and Computing*, Long Beach, CA, USA, October 2001.

[7] P. Zimmerman, *The Official PGP User's Guide*, MIT Press, 1995.

[8] W. Ford and M. S. Baum, *Secure Electronic Commerce,* Prentice Hall PTR, 1997.

[9] U.S. National Institute of Standards and Technology, *A Public Key Infrastructure for U.S. Government unclassified but Sensitive Applications*, September 1995.

[10] C. Adams and S. Lloyd, *Understanding Public-Key Infrastructure: Concepts, Standards and Deployment Considerations*, Macmillan Technical Publishing, 1999.

[11] D. A. Cooper, "A more efficient use of delta-CRLs", *In Proceedings of the 2000 IEEE Symposium on Security and Privacy*, 2000, pages 190—202.