Generating CA-authenticated public keys in ad hoc networks

Demo

Gina Kounga
DoCoMo-EuroLabs
Landsberger Strasse 312
80687 Munich, Germany

Thomas Walter
DoCoMo-EuroLabs
Landsberger Strasse 312
80687 Munich, Germany

Christian Schaefer
DoCoMo-EuroLabs
Landsberger Strasse 312
80687 Munich, Germany

tsurname@docomolab-euro.com

1. GOALS OF THE DEMO
The demo is a proof of concept of an entity authentication solution defined in [1] that permits nodes in ad hoc networks to generate, on-demand, public/private key pairs whose validity can be verified based on a unique certificate issued by a Certification Authority (CA) during registration in the fixed network. Registration only occurs once and the certificate issued by the CA does not contain any public key but binds an identity to a hash code. This avoids the need to manage the revocation of the certificate in the ad hoc network. In the demo, the entity authentication solution is used by a buying/selling application defined in [2][3] that permits users, in ad hoc networks, to buy or sell digital multimedia resources anytime, anywhere and from anybody. Subsequently, we briefly present the authentication solution and the buying/selling application.

2. Entity authentication solution
The authentication solution comprises the steps of entity registration, public/private key pair generation and public key validation. These steps are further detailed hereafter.

2.1 Registration
In the fixed network, for instance, when a user registers at his network provider, he requests a certificate from a CA as follows. He contacts a CA with his mobile device A in order to obtain a reliable copy of the system parameters: the two one way hash functions \( h \) and \( f \) as well as the large prime \( g \), \( K_{CA} \) the CA’s public key and an integer \( L \). Upon registration or whenever connected to a fixed network A’s mobile device synchronizes its clock; this is important for reasons that become obvious later. When A receives these parameters, it chooses a secret \( s \). Note that \( s \) is not stored by A; instead, it is generated from a strong passphrase. Then A generates the check value \( V \) :

\[
v = h(g \prod_{j=0}^{m} f^{j}(s))
\]

Where \( f^{j}(s) \) means that the cryptographic one way hash function \( f \) is applied \( j \) times on \( s \).

A then sends its identity \( ID_{A} \) and the check value \( V \) to the CA, in order to obtain a certificate that binds these two values. The CA next verifies that A really owns \( ID_{A} \) and that no certificate has been issued that already contains \( V \). If all the verifications succeed, then CA sends to A the certificate \( Cert_{A} \) which includes the mentioned system parameters but as well \( ID_{A} \) and \( V \). After A has received the certificate, it divides time into interval \( T_{0} \) of equal length \( L \). The first time interval \( T_{0} \) starts at the issue time \( t_{0} \) of \( Cert_{A} \). The solution works such that for each time interval \( T_{i} \) a new public/private key pair is generated as discussed in the following sections. Obviously, the smaller \( L \) is the smaller is the risk that an attacker uses a revoked public/private key pair in the ad hoc network that is linked to \( Cert_{A} \).

2.2 Public/private key pair generation
In time interval \( T_{i}, 0 \leq i < m \), A uses as private key \( K_{m-1-i} \), generated as follows:

\[
K_{m-1-i} = \prod_{j=0}^{m-1-i} f^{j}(s)
\]

and as corresponding public key \( g^{K_{m-1-i}} \). The key pair expires automatically, when the next time interval \( T_{i+1} \) after at most \( L \) time units is reached.

2.3 Public key validation
To enable a device B to obtain a verified copy of the public key \( g^{K_{m-1-i}} \), A sends to B:

\[
\begin{align*}
& g^{K_{m-1-i}} , \\
& f^{m-i}(s) \text{ and} \\
& Cert_{A}.
\end{align*}
\]

B determines \( i \) from its local time, \( t_{0} \) and \( L \) both contained in \( Cert_{A} \). Then B computes:

\[
f^{j}(s), \text{for } m-i < j \leq m
\]
Then, B computes $v^*$ as follows:

$$v^* = (g^{k_{m-i-1}}) \prod_{j=m-i}^{m} f_{j}(x)$$

$$= (g^{k_{m-i-1}}) \prod_{j=0}^{m} f_{j}(x)$$

Then B verifies that $v = h(v^*)$ where $v$ is contained in CertA.

If yes, then, B considers $g^{k_{m-i-1}}$ as valid and can use it to verify some digital signatures during $T_{di}$ or to establish some secure communication channels with A. These details are omitted due to limited space.

3. Buying/Selling multimedia resource in ad hoc networks

In the demo, the CA is emulated. Certificates are generated from public parameters stored on the devices. However, the public/private key pairs used are generated and validated as defined previously.

The process that permits B – for buyer— to buy a previously requested multimedia resource to a node S – for seller— in ad hoc networks is represented below in Figure 1.

Without implementing DRM features, for example the encryption of the resource, the previous process has been implemented entirely on constrained devices (Nokia Communicator 9300/9500i in Java) and appears in the demonstration as shown in Figures 2 to 6. The implementation confirmed that the authentication solution performs on constrained devices. Performance figures are included in [1] and are shown on the poster that goes with the demonstration.

4. SETUP TIME AND FACILITIES NEEDED

The demo requires 3 power plugs (220 V) and some space for a A0 poster (59.4cm x 84cm). The set up time for the demo is 30 minutes.

5. EXTRA INFORMATION

Please refer to the “References” section.

6. References

