

SWAP framework

Part II: framework architecture

Kjell Jørgen Hole



Last updated June 10, 2008

Outline

1. Choice of public-key certificates
2. Framework architecture
3. Initial identification
4. Secure data storage
5. Certificate revocation

2

Public-key certificates

(choice of authentication method)

3

What identifiers to use?

- Let an entity be a smart phone, the individual owning the phone, or an application server
- All entities are associated with at least one locally unique **identifier** (or ID)
 - locally unique IDs are possible because the user community is restricted to “individuals allowed to access this application”
- Note that it is possible to revoke an entity’s access to the application server since the entity has a unique ID

4

Examples of unique IDs

- SSNs
- E-mail addresses
- patient IDs
- Account numbers
- IP addresses
- Phone numbers
- Web addresses

5

Choice of certificates (1)

- An organization utilizing the framework is likely to use its own (locally) unique IDs
- Most likely, the organization wants to control the creation of certificates using established IDs
- As a result, the framework may use **ID certificates** that associate IDs with public keys

6

Choice of certificates (2)

- Even though the IDs are unique, people have a tendency to select wrong IDs when they want to communicate with other people
- Can instead use **authorization certificates** that associate authorizations with public keys
 - if keyholders are to be hold accountable for their actions, these certificates must still contain names, but the names need only be meaningful to humans
 - it is also possible to bind the keyholders to their keys using a database

7

Choice of certificate (3)

- We choose ID certificates because
 - we want to keep users responsible for their actions
 - there is Java support for the much used X.509 certificates
- The framework will utilize three types of certificates
 - CA certificates
 - (application) server certificates
 - user certificates

8

Framework architecture

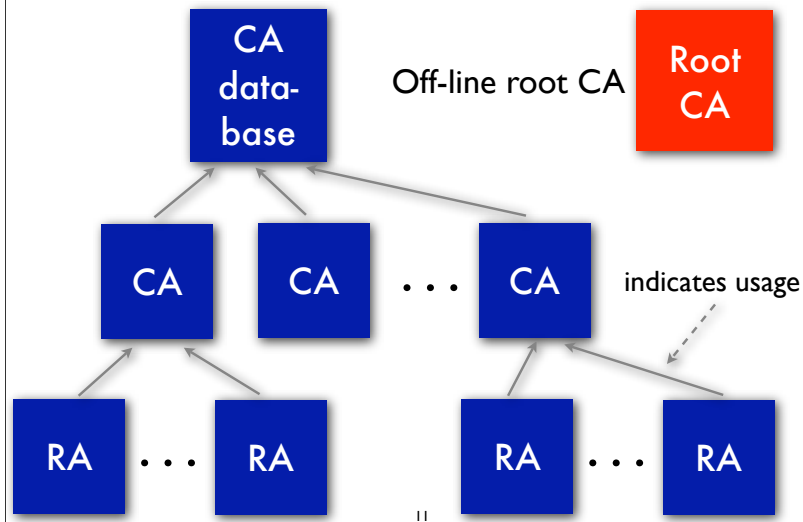
9

RA/CA architecture (1)

- **RA** (Registration Authority) and **CA** (Certification Authority)
- RA and the CA utilize cryptographic modules to store keys
- **CA database** contains public-key certificates and revocation information
- **remark:** in addition, the application server is likely to have its own database

10

RA/CA architecture (2)



11

RA/CA architecture (3)

- One off-line root CA
- Root CA issues certificates to one or more on-line CAs
- Each on-line CA can support multiple RAs
 - need protocol for secure communication between an RA and a CA
- All CAs store issued certificates and revocation information in the same internal CA database

12

Physical security (1)

- A CA is always placed in a secure environment
- An RA placed in a non-secure environment is vulnerable to attacks
 - an RA connected to the Internet is particularly troublesome
- The SWAP framework requires the same level of physical security for both on-line CAs and RAs
 - since physical security is expensive, it is likely that the total number of on-line CAs and RAs will be small

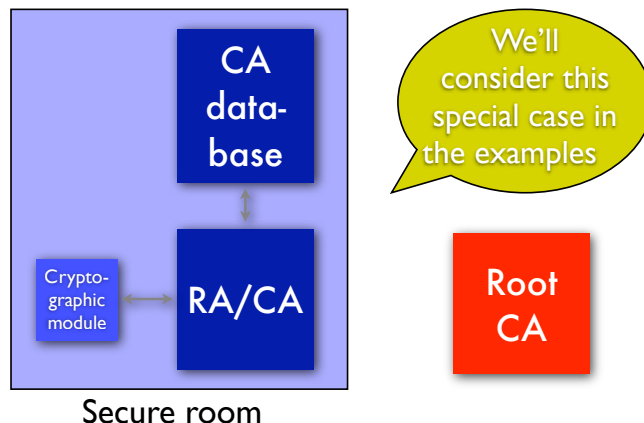
13

Physical security (2)

- A small number of CAs and RAs also limits the number of attack points
- One CA and one or more RAs may be placed in the same secure room
- It is possible to have a combined RA/CA in the case where only one CA and one RA are needed
 - can remove protocol for secure communication between CA and RA

14

Special case: combined RA/CA



15

CA tasks

- Off-line root CA issues certificates to on-line CAs
- An on-line CA must
 - issue public-key certificates to new application servers and smart-phone clients
 - renew certificates
 - revoke certificates
- CA database maintains public-key certificates and revocation information

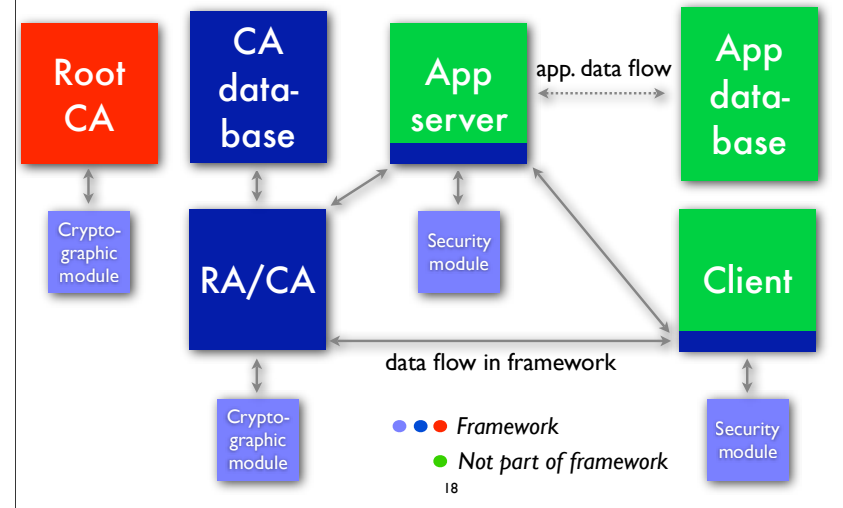
16

New root certificate

- The root CA has a self-signed certificate
- When the root certificate expires, the root CA issues a new self-signed certificate
- The new certificate is signed with the old root key
- The other entities must be able to download the new root certificate

17

Complete architecture



Framework on app. server

- Verify certificate from a smart-phone client
 - a. verify that a received certificate has been issued by a trusted CA
 - b. CA's digital signature is valid
 - c. the certificate is within its established validity period
 - d. the certificate has not been revoked
 - e. the certificate is used according to the policy
 - f. ID verification (?)
- Authenticate client
- Provide secure storage of keys

19

Framework on client

- Each user installs a security module on his phone
 - this module may be a tamper-resistant smart card
- The module generates public-private key pairs
 - the private key never leaves the module
 - malicious software must not be able to read keys
- The module also contains root CA's certificate
- Access to module is protected via a long PIN or a password

20

Client tasks

- Generate cryptographic keys
- Provide secure storage of keys
- Verify certificates from app. server and other clients
 - a. verify that a received certificate has been issued by a trusted CA
 - b. CA's digital signature is valid
 - c. the certificate is within its established validity period
 - d. the certificate has not been revoked
 - e. the certificate is used according to the policy
 - f. domain name verification
- authenticate app. server or client

21

Certificate validation problem?

- The local time on the client may well be wrong

Simplification

- It is possible to view both the application server and a smart-phone client as a “general client” to the central PKI framework
- We'll call this general client a **PKI member**
- This simplified view shows that it is possible to define a general **PKI client** that can run on both the application server and a smart-phone client
 - implementations are likely to be different on servers and phones

23

Initial identification

24

Initial user identification

- Must choose **off-line** or **Internet-based** identification
- We choose off-line identification since on-line authentication is more open to attack
- A new user shows up in person at an RA
- The RA first verifies the information presented by the individual:
 - driver's license
 - passport
 - employee badge with picture

25

Security module

- A security module is then assigned to the individual
 - the module is connected to an RA
- The module generates a pair of public-private keys
- The key pair is stored on the security module
- The module sends the public key to the RA
- The RA sends a certificate request to a CA

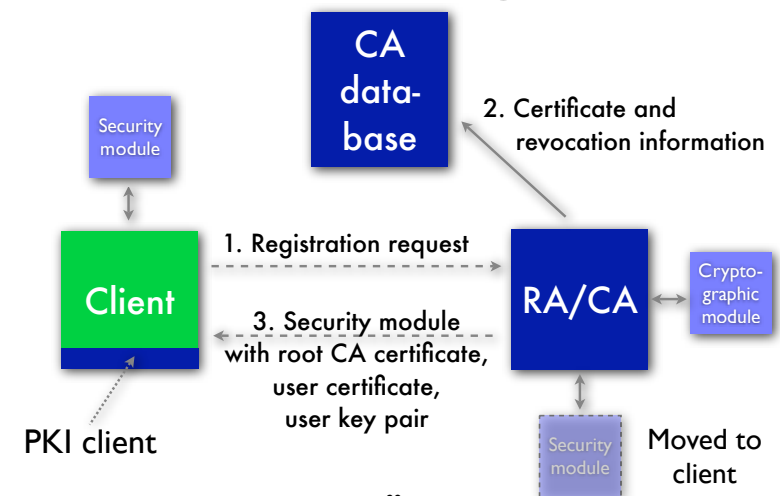
26

Generation of certificate

- CA generates user certificate and sends it to RA
- User certificate is stored on security module
- User certificate and other information are stored in central CA database
- User installs module on smart phone

27

Illustration of registration



28

Initial app. server identification

- Assume that the application server can use the same security module as the clients
- In this case, the operator of the application server can follow the outlined procedure for the clients to identify the server
 - the information needed by the RA may differ from the client case

29

Certificate update (1)

- It is suggested that certificate updates should occur once 70% to 80% of the current key lifetime has been exhausted
- When a (user or server) certificate is close to its expiration date, the entity must generate a new public-private key pair and request a new certificate from a CA
- The certificate request is signed with the old private key
- The CA issues a new certificate and sends it to the entity

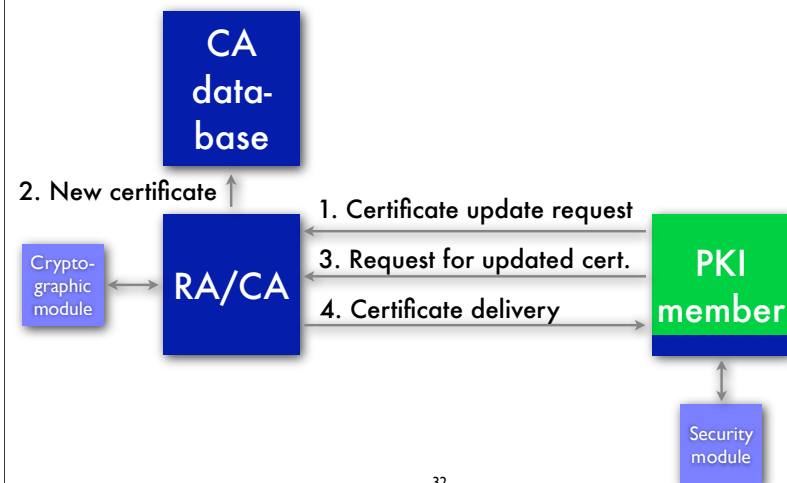
30

Certificate update (2)

- Old certificate is revoked from the moment the new certificate is issued
- **Remark:** If a user certificate has expired or been revoked, then the user must show up in person at an RA and go through the initial identification procedure again
- **Remark:** When a server certificate has expired or been revoked, the owner (or an authorized operator) of the server must show up in person at an RA and reinitialize the security module as in the previous case

31

Illustration of cert. update



32

Secure data storage

Better solution needed

33

Security module and local encryption

- Encryption of client data may be based on a secret key obtained from the PIN or password used to unlock the security module
- This way it is still possible to decrypt a local file even if the security module malfunctions
- If the module is destroyed, the customer must obtain a new module at an RA
 - new public-private key pairs must be generated

34

New PIN

- It should be possible for the user to change PIN
- Since the secret key depends on the PIN, a new key is generated when the PIN is changed
- In this case it is necessary to decrypt all stored data with the the old key (based on the old PIN), and to encrypt the data with the new key (based on the new PIN)

35

Certificate revocation

36

Need good revocation technique for thin clients

- Client description:
 - online
 - limited storage space
 - limited processing power
 - insecure OS

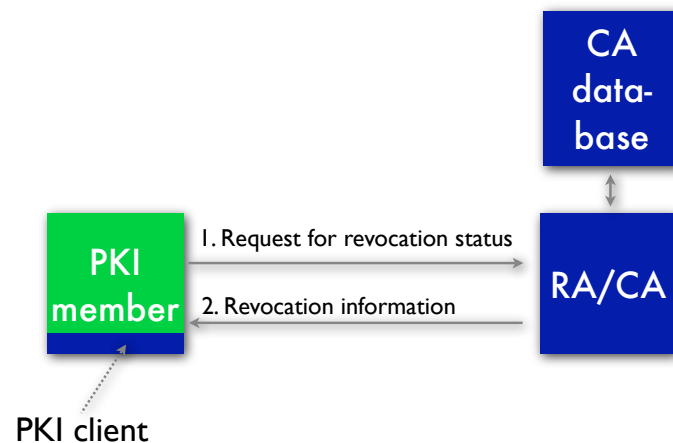
37

Revocation solution

- The PKI client automatically contacts a CA to verify the revocation status of a received certificate and disables the connection if the cert. is revoked
 - separate, secure protocol is needed
 - instantaneous, no CRL propagation delay
 - consistently applied, need not worry about whether application checks for revocation
 - scales to support many smart-phone clients
 - not robust against DoS (Denial-of-Service) attacks

38

Revocation status



39

Revocation process

- Only the owner of a certificate and the administrator of the application server are allowed to revoke a certificate
- A digital signature is used to determine whether a revocation request should be carried out
- Separate procedure is needed to allow the framework administrator to revoke certificates belonging to clients that have broken a contractual agreement

40

Certificate revocation in *ad hoc* network

- Certificate revocation may be less important in *ad hoc* networks because two-way authentication is carried out by users talking directly to each other
 - assuming short-range Bluetooth communication
- Revocation checking is not possible when a client has no contact with a CA
- A client with access to a CA can ask for the status of a certificate

41

Sources

- C. Adams and S. Lloyd, *Understanding PKI*, 2nd Edition, Addison Wesley, 2003
- R. Clarke, "Conventional Public Key Infrastructure: An Artifact Ill-Fitted to the Needs of the Information Society," www.anu.edu.au/people/Roger.Clarke/II/PKIMisFit.html
- P. Gutmann, "PKI: It's Not Dead, Just Resting," *IEEE Computer*, August 2002, pp. 41–49
- National Research Council, *Who Goes There?* National Academies Press, 2003
- L. F. Cranor and S. Garfinkel, editors, *Security and Usability*, O'Reilly, 2005
- BSK, Norsk BankID sertifikatpolicy for banklagrede kvalifiserte sertifikater til personlunder, v. 1.1, desember, 2005
- S. Bibb and J. Kourdi, *Trust Matters*, Palgrave Macmillan, 2004

42

