A Profile Of PKCS #11 V2.11 For Mobile Devices

Magnus Nyström

PKCS Workshop 2002

Motivation

- Given that mobile devices (PDAs, SmartPhones, Cellular phones, pagers, etc.) :
  1. increasingly needs access to security services such as
     - Digital rights management
     - Secure communication
     - Transaction security,
  2. runs on a wide variety of OS platforms,
  3. to a high degree supports or will support cryptographic tokens such as
     - smart cards
     - secure multimedia cards…

Motivation

- …there is an increased need to offer a standardized interface to tokens in this environment
- Why not just PKCS #11 v2.11?
  - Too large
  - Too flexible

A profile of PKCS #11 seems suitable!

Previous Work

- PKCS #11 v2.10 Conformance Profile Specification
  - Contains 4 profiles:
    - RSA acceleration (clearly not suited)
    - D-H acceleration (clearly not suited)
    - RSA asymmetric client (too constrained)
    - Large Application Profile (lacks certain features)

Proposal

- Make a profile based on PKCS #11 v2.11
  - Fastest way forward, will (hopefully) be possible to incorporate PKCS #11 advancements later on

Profile: Mechanisms

- CKM_RSA_KEY_PAIR_GEN, CKM_RSA_PKCS, CKM_RSA_X509 (MUST support at least 1024-bit keys), CKM_MD5_RSA_PKCS, CKM_SHA1_RSA_PKCS, CKM_SHA_1_HMAC, CKM_CMS_SIG, CKM_SHA_1, CKM_MD5, and possibly a selection of symmetric mechanisms - RC4, 3DES, RC5 (?), AES, and - possibly - TLS/SSL mechanisms (not used frequently by browsers).
- Comparison with LAP:
  - LAP only supports CKM_RSA_X509 and CKM_RSA_PKCS
- My view is that LAP is too constrained (but the list above may well be too long)

Profile: Functions

- Basically the same as is in LAP with the following additions:
  - C_SeedRandom(), C_GetRandom()
    - Reason: Tokens used by these devices normally supports this operation
  - C_VerifyInit, C_Verify()
    - Reason: As above

Profile: Thread Handling

- Suggest that library must be able to handle option 4 in Section 6.5.2 of PKCS #11 v2.11:
  - Library must be prepared to us native OS primitives for concurrent accesses or application-supplied callback primitives
- Reason:
  - Most flexible solution, yet should be achievable on all platforms

Profile: Sessions

- My suggestion is similar as for the LAP: 10 simultaneous R/O, 1 R/W.

Profile: Templates

- Similar to LAP, but:
  - Changed “MUST” to “MUST be able to”, in a number of places, e.g. some applications may prefer to generate short-lived keys in software and not have them as token objects.

For Discussion

RSA SECURITY

- Compound Call:
  - Find slot, token, initiate session, login user?
    - Simplifies client code
- Level of detail
  - Suggest higher level of detail than for existing profiles
    - Increases chance of interop
    - Reduce risk of confusion (?)
- Timeline