PKCS

- Public-key cryptography standards (PKCS)
- Owned by RSA and motivated to promote RSA
- Created in early 1990’s
- Numbered from PKCS1 to PKCS15
- Some along the way have
  - lost interest
  - folded into other PKCS
  - taken over by other standards bodies
- Continue to evolve

© Ravi Sandhu 2000-2004
PKCS 1

- RSA Cryptography Standard
  - Version 2.0 onwards (1998)
- RSA Encryption Standard
  - Version 1.5 (1993)

PKCS 1

- Specifies how to use the RSA algorithm securely for encryption and signature
- Why do we need this?
  - Padding for encryption
  - Different schemes for signature
PKCS 1

- Chosen ciphertext attack based on multiplicative property of RSA
  - Attacker wishes to decrypt $c$
  - Choose $r$, compute $c' = c.r^e \mod n$
  - Get victim to decrypt $c'$ giving $c^d.r \mod n$
  - $c^d.r.r^{-1} \mod n = c^d \mod n$

- Padding destroys multiplicative property

PKCS 1

- Version 1.5, 1993
  - Encryption padding was found defective in 1998 by Bleichenbacher
  - Possible to generate valid ciphertext without knowing corresponding plaintext with reasonable probability of success (chosen ciphertext)
PKCS 1

- Version 2.0, 1998
  - Uses Optimal asymmetric encryption protocol (OAEP) by Bellare-Rogoway 1994
    - provably secure in the random oracle model
    - Informally, if hash functions are truly random, then an adversary who can recover such a message must be able to break RSA
    - plaintext-awareness: to construct a valid OAEP encoded message, an adversary must know the original plaintext
  - PKCS 1 version 1.5 padding continues to be allowed for backward compatibility
  - Accommodation for multi-prime RSA
    - Speed up private key operations

PKCS 1

- Cryptographic primitives
- Cryptographic scheme
  - Encryption scheme
  - Signature scheme
    - Signature with appendix: supported
    - Signature with message recovery: not supported
- Encoding and decoding
  - Converting an integer message into an octet string for use in encryption or signature scheme and vice versa
PKCS 1

- **Cryptographic primitives**
  - Encrypt: $\text{RSAEP}((n,e),m)$
  - Decrypt: $\text{RSADP}((n,d),c)$
  - Sign: $\text{RSASP1}((n,d),m)$
  - Verify: $\text{RSAVP1}((n,e),s)$
- **Basically exponentiation with differently named inputs**

PKCS 1

- **Encryption scheme**
  - Combines encryption primitive with an encryption encoding method
  - message → encoded message → integer message representative → encrypted message
- **Decryption scheme**
  - Combines decryption primitive with a decryption decoding method
  - encrypted message → integer message representative → encoded message → message
- **Original version 1.5 scheme and new version 2.0 scheme**
PKCS 1

- **Signature scheme**
  - Combines signature primitive with a signature encoding method
  - message → encoded message → integer message representative → signature

- **Decryption scheme**
  - Combines verification primitive with a verification decoding method
  - signature → integer message representative → encoded message → message

- **Original version 1.5 scheme**
  - Signature with appendix

PKCS 1

- **The future**
- **Probabilistic signature scheme (PSS)**
  - Provably secure in random oracle model
  - Natural extension to message recovery
PKCS 5

- **Password-Based Cryptography Standard**
  - Version 1.5, 1993
  - Version 2.0, 1999
- Oriented towards protection of private keys
- Does not specify a standard for password format

PKCS 5

- **Password-based key derivation function**
  - Key = PBKDF(passwd, salt, iteration count)
- salt allows same password to give many keys
  - May actually have same password
  - Separate dictionary attack for every salt
- Iteration count controls complexity of dictionary attack
PKCS 5

- Version 1.5 PBKDF1
  - Key size limited to 160 bits
  - Only MD5 and SHA as underlying hash functions
  - Assumes key will be used for CBC
  - 8-byte salt
  - No security proof

PKCS 5

- Version 2.0 adds PBKDF2
  - Arbitrary length key
  - Any underlying hash function, most likely with HMAC
  - Salt not fixed at 8 bytes
  - Provable security in random oracle model
PKCS 5

- **Encryption schemes**
  - PBES1
    - PBKDF1 with DES or RC2 in CBC
  - PBES2
    - PBKDF2 with some underlying encryption scheme
- **MAC scheme**
  - PBMAC1
    - PBKDF2 with some underlying MAC scheme

PKCS 10

- Certification Request Syntax Standard
- Specifies format of unsigned certificate requested to be signed
- Does not specify format of returned signed certificate
PKCS 10

- Version 1.0, 1993
  - In widespread use
- Version 1.5, 1998
- Version 1.7, 2000
  - Minor changes such as references to PKCS 6 replaced by references to X.509v3

PKCS 10

- CertificationRequestInfo
  - version
  - subjectName
  - subjectPublicKeyInfo
  - attributes
PKCS 10

- CertificationRequest
  - certificationRequestInfo
  - signatureAlgorithm
  - signature
- Signed with private key corresponding to public key in request
  - very RSA specific
  - IETF RFC 2511 defines a different format: certificate request message format

PKCS 8

- Private-Key Information Syntax Standard
  - Version 1.2, 1993
PKCS 8

- `PrivateKeyInfo`
  - `version`
  - `privateKeyAlgorithm`
  - `privateKey`
  - `attributes`

PKCS 8

- `encryptedPrivateKeyInfo`
  - `encryptionAlgorithm`
  - `encryptedData`
    - `privateKeyInfo` BER-encoded and encrypted
  - Usually encrypted using PKCS 5
PKCS 12

- Personal Information Exchange Syntax Standard
  - Version 1, 1999
- Builds on PKCS 8
- Further evolution PKCS 15

PKCS 12

- 6 types of information
  - PKCS 8 shrouded key
  - Private key
  - Certificates
    - X.509v3
    - SDSI
  - CRLs
    - X.509
  - Secret
    - Whatever
  - Recursive composition of these
PKCS 12

📍 Each of these can be
  ➢ Plaintext
  ➢ Enveloped
    • Encrypted using a secret key which is encrypted using a public key
  ➢ Encrypted
    • Secret key encrypted
    • Usually password derived
      – Use PKCS 5 and a password formatting standard which is part of PKCS 12

PKCS 12

📍 The entire stuff is then either
  ➢ Signed
    • And accompanied with signing certificate
  ➢ MAC’ed
    • PKCS 5 based and accompanied with salt and iteration count

📍 Notice: opposite of usual sequence
  ➢ Encrypt and then authenticate, versus
  ➢ Authenticate and then encrypt
PKCS
DISCONTINUED OR DISINTERESTED

- PKCS 2
  - discontinued, incorporated into PKCS 1
- PKCS 3
  - Diffie-Hellman Key Agreement, 1993
- PKCS 4
  - discontinued, incorporated into PKCS 1

PKCS
TAKEN OVER BY OTHERS

- PKCS 6
  - Extended Certificate Syntax Standard
  - Taken over by X.509v3
- PKCS 7
  - Cryptographic Message Syntax Standard
  - Taken over by IETF PKIX CMS
PKCS 9

- PKCS 9
  - Selected Attribute Types
  - For use in PKCS 6, 7, 8, 10

PKCS 11

- PKCS 11
  - Cryptographic Token Interface Standard
  - API used by Netscape (pre 6.0)
  - Microsoft CSP (Cryptographic Service Provider) is a competitor
PKCS IN DEVELOPMENT

- PKCS 13 (new, in development)
  - Elliptic Curve Cryptography Standard
  - There are IEEE standards, so not clear why
- PKCS 14 (new, in development)
  - Pseudorandom Number Generation Standard
- PKCS 15 (new, in development)
  - Cryptographic Token Information Format Standard
  - Crypto API neutral

PKCS 11 vs PKCS 15

- Crypto Application (Browser, email client etc)
- Standard Crypto API (PKCS 11, CSP, etc)
- Cryptographic Token Information Format Standard (PKCS 15)