PKI Standards

Centre for Development of Advanced Computing (C-DAC)
Bangalore

Under the Aegis of

Controller of Certifying Authorities (CCA)
Government of India

PKCS

- Why PKCS?
  - Even though vendors may agree on the basic public-key techniques, compatibility between implementations is by no means guaranteed.
  - Interoperability requires strict adherence to an agreed-upon standard format for transferred data.
  - Standards provide a basis for interoperability.

Organizations in Public Key Standard

- Major organization involved in public key cryptography standards are: ISO/IEC, ANSI, NIST, IETF, IEEE
- ISO/IEC
  - International Organization for Standardization and International Electro technical Commission:
    - Standards for application independent cryptographic techniques.
    - ISO has also been developing bank security standards under ISO technical committee TC68/SC 2 Banking and Related Financial Services.

Organizations in Public Key Standard

- ANSI
  - Developed standards for financial service under Accredited Standards Committee (ASC) ANSI X9.42:2003
- NIST
  - Developing standards for use by US federal government department.
  - These standards are released in Federal Information Processing Standards (FIPS) Publication.

Standard Organization in PKI

- IETF
  - Developing standards for use by internet community. These standards are published in RFC s
- IEEE
  - IEEE 1363-2000
    - Standard Specifications For Public Key Cryptography
  - IEEE 1363a
    - Standard Specifications For Public Key Cryptography- Amendment 1: Additional Techniques

Vendor Specific Standards

- Public Key Cryptographic Standards (PKCS)
  - Developed by RSA
- Standards for Efficient Cryptography (SEC)
  - Industry Consortium developing standards
  - SEC #1 and SEC#2
    - Elliptic curve cryptography standards

PKCS – An Overview

No	PKCS NAME	COMMENT
1	RSA Cryptographic Standards	
2, 4		Incorporated with PKCS#1
3	Diffie – Hellman Key agreement Standard	
5	Password Based Cryptography Standard	
6	Extended Certificate Syntax standard	
7	Cryptographic Message Syntax Standard	Super seeded by RFC 3369
8	Private Key Information Syntax Standard	
9	Selected Object Class and Attribute types	
10	Certification Request Syntax Standard	
11	Cryptographic Token Interface Standard	Referred as CRYPTOKI
12	Personal Information Exchange Syntax Standard	
13	Reserved for Elliptic Curve Cryptography	
14	Reserved for Pseudo random number Generation	
15	Cryptographic Token Information Syntax Standard	

PKCS #1 - RSA Encryption Standard

- Describes a method, called rsaEncryption, for encrypting data using the RSA public-key cryptosystem.
- It is intended use is in the construction of digital signatures
- Also describes the syntax for RSA public and private key

PKCS #3: Diffie-Hellman Key Agreement Standard

- Describes a method for implementing Diffie Hellman key agreement
  - Whereby two parties, without any prior arrangements, can agree upon a secret key that is known only to them
- Application of PKCS #3
  - In protocols for establishing secure connections, such as those proposed for OSI's transport and the network layers.

PKCS #5: Password-Based Encryption Standard

- Describes a method for encrypting an octet string with a secret key derived from a password.
- Applications
  - For encrypting private keys when transferring them from one computer system to another, as described in PKCS #8.
  - Can be used to encrypt arbitrary octet strings

PKCS # 6: Extended-Certificate Syntax Standard

- An extended certificate consists of an X.509 public-key certificate and a set of attributes, collectively signed by the issuer of the X.509 public-key certificate.
- Thus the attributes and the enclosed X.509 public-key certificate can be verified with a single public-key operation
- Application
  - In the cryptographic - enhancement syntax standard (PKCS #7)

PKCS #6: Extended-Certificate Syntax Standard

- When PKCS #6 was drafted, X.509 was in version 1.0 and no *extension* component was defined in the certificate.
- An X.509 v3 can contain information about a given entity in the *extensions* component.
- Since the introduction of X.509 v3, PKCS #6 has become historic, and obsolete

PKCS # 7: Cryptographic Message Syntax Standard (CMS)

- Defines syntax to digitally sign, digest, authenticate or encrypt arbitrary message content.
- Describes an encapsulation syntax for data protection.
- Allows recursion
  - One envelope can be nested inside another, or one party can sign some previously enveloped digital data
- Allows arbitrary attributes
  - Eg: signing time, can be signed along with the content of a message.
  - Attributes like countersignatures to be associated with a signature.

PKCS #8: Private key information Syntax Standard

- Security of cryptosystem is entirely dependent on the protection of private keys. Generally private keys are encrypted with a password and stored in in some storage media.
- It is important to have a standard to store private keys so that one can move private keys from one system to another.

PKCS #8: Private key Information Syntax Standard

- PKCS #8 describes a syntax for storing private key information and set of attributes and a syntax for encrypted private key information
- Password based encryption algorithm (pkcs #5) could be used to encrypt the private key information

PKCS #9: Selected Attribute Types

- Specifies two auxiliary object classes, 'pkcsEntity' and 'naturalPerson', and some new attribute types and matching rules.
  - The 'pkcsEntity' object class is a general-purpose auxiliary object class that is intended to hold attributes about PKCS-related entities.
  - It has been designed for use within directory services based on the LDAP protocol and the X.500 family of protocols.
  - The 'naturalPerson' object class is a general-purpose auxiliary object class that is intended to hold attributes about human beings
- This standard defines selected attribute type for use in PKCS #6,7,8 and 10

PKCS #10: Certification Request Syntax Standard

- PKCS #10 describes a syntax for certification requests.
  - Does not specify the forms that the certification authority returns the new certificate
  - A certification request consists of a distinguished name, a public key, and optionally a set of attributes, collectively signed by the entity requesting certification.
  - Certification requests are sent to a certification authority, who transforms the request to an X.509 public-key certificate, or a PKCS #6 extended certificate.

PKCS #11: Cryptographic Token Interface Standard

- PKCS#11 specifies an application programming interface called Cryptoki, to device which hold cryptographic information and perform cryptographic functions
- The primary goal of cryptoki was a lower level programming interface that abstracts the details of the devices, and presents to the application a common model of the cryptographic device called cryptographic token.

PKCS #12: Personal Information Exchange Syntax Standard

- PKCS #12 describes a file format for storing and transfer of private keys & personal identity information
  - Including private keys, certificates, miscellaneous secrets and extensions.
- Machines, applications, browsers etc that supports this standards will allow a user to import, export and exercise a single set of personal identity information

PKCS #15: Cryptographic Token Information Syntax Standard

- Use of cryptographic tokens (IC cards) for authentication and authorization purpose has been hampered by the lack of interoperability
  - Lacking the standard for storing a common format of digital credentials (keys, certificates) on tokens
  - Mechanism to allow multiple applications to effectively share digital credential have not yet reached maturity.

PKCS #15: Cryptographic Token Information Syntax Standard

- PKCS #15 intended to enable
  - Interoperability among components running on various platforms
  - Application to take advantage of products and components from multiple manufactures
  - Use of advances in technology without rewriting application level software
  - Consistency with existing and related standards.

PKCS – An Example

- Requirement
  - To implement a smart card authentication system based on public key cryptography technology
  - Each user has a smart card containing user's private key, public key certificate and other information
  - Users can authenticate by inserting the card into the card reader and typing the password

PKCS – An Example

- PKCS #1 can be chosen as the underlying cryptographic mechanism
- User Registration
  - User needs to register for getting the smart card
  - In registration the system will generates a public key/private key for that user
  - Using PKCS #9, the system generates object and attributes containing users information

PKCS – An Example

- Generate Certificate request
  - Using the information a certificate request can be generated according to PKCS #10
- Sending the Certificate request
  - The system can send the certificate request object to Certifying Authorities enveloped using PKCS #7
- After the identity verification, the CA signs users public key to generate a certificate for the user and sends it back to the system.

PKCS – An Example

- After receiving users certificate from CA, the system build a smart card for the user.
- Using users password (PIN), the system generate an encrypted PrivateKeyInfo object for the user according to PKCS#8 and PKCS #9
- PKCS #12 can be used to transfer users encrypted private key and other information from one system to other

PKCS – An Example

- Using the dedicated file format (PKCS #15) , users encrypted private key object EncryptedPrivateKeyInfo, certificate and other information could be stored in the smart card.
- User can take a copy of these private information on a USB. These personal information is stored in USB according to PKCS #12.
- By means of PKCS #11 API, application of these computing system can communicate with the smart card.

Federal Information Processing Standards (FIPS)

- Developed by the National Institute of Standards and Technology (NIST) for Federal computer systems .
- **Key FIPS Standards**
  - 140-2 : Standard for Security Requirements for Cryptographic Modules
  - 180-1 : Secure Hash Standard SHA-1
  - 180-2 :Updated Secure Hash Standard SHA-1 plus SHA-256, SHA-384, SHA-512
  - 186-2 :Digital Signature Standard - DSA

FIPS 140-2

– Security Requirements for Cryptographic Modules – FIPS 140-2

- This standard specifies the security requirements that are to be satisfied by a cryptographic module.
- The standard provides four increasing, qualitative levels of security.
- These levels are intended to cover the wide range of potential applications and environments in which cryptographic modules employed.

FIPS 140-2

- The security requirements cover areas related to the secure design and implementation of a cryptographic module.
  - Basic design and documentation
  - Module interfaces
  - Authorized roles and services.
  - Physical security
  - Software security
  - Operating system security
  - Key management
  - Cryptographic algorithms
  - Electromagnetic interference
  - Electromagnetic compatibility
  - Self-testing.

FIPS 180-2

- Secure Hash Standard (SHS)
- This Standard specifies four secure hash algorithms
  - SHA-1
  - SHA-256
  - SHA-384
  - SHA-512
- These algorithms differs
  - In the number of bits of security that are provided for the data being hashed
  - In terms of the size of the blocks and words of data that are used during hashing.

FIPS 186-2

- Digital Signature Standards (DSS)
  - Specifies a suite of algorithms which can be used to generate and verify digital signature
  - Prescribing three algorithms
    - Digital Signature Algorithm (DSA)
    - RSA Digital Signature Algorithm
    - Elliptic Curve Digital Signature Algorithm (ECDSA)

References

- Public-Key Cryptography Standards: PKCS – Yongge Wang
- www.rsa.com/rsalabs/node.asp?id=2124
- http://www.itl.nist.gov/fipspubs/by-num.htm

Thank You

pki@cdac.in