**Module 6 Cryptography & PKI**

**6.4 PKI**

**Public & Private Keys**

* Encrypt document with recipient’s public key. Only their private key needs to be kept secret & only it can decrypt message
* Sender’s private key used to sign message

**PKI Components**

* Solves issues with key management
* Set of roles, policies & procedures needed to manage public-key (asymmetric) encryption
* Process of creating, managing, distributing, storing, using & revoke keys & digital certificates
* Public Key Infrastructure X.509 (PKIX) is working group formed by the IETF to develop standards & models PKI

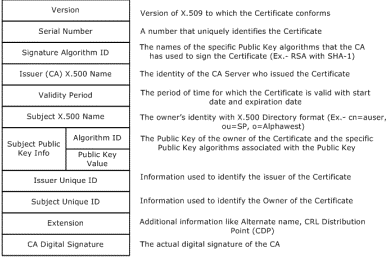
**PKI Components – Digital Certificate**

* Digitally signed block of data used to prove the ownership of public key issued by a Certificate Authority
* Includes

1. Information about the key
2. Information about the identity of its owner (the subject)
3. Digital signature of an entity that has verified the certificate’s contents (the issuer)

* X.509 v3 standard defines the certificate formats & field for public keys

**Digital Certificate Components**



**X.509 Certificate Types**

* Root Certificates

1. For root authorities
2. Usually self-signed by that authority & often kept offline

* Domain Validation (DV) – includes only domain name
* Organisational Validation (OV)

1. Organisations vetted against official government sources
2. Common for public-facing websites

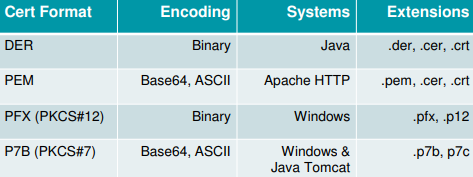
* Extended Validation (EV)

1. Highest level of trust
2. Requires comprehensive validation of business

* Wildcard Certificates – allows subdomains for single registered domain (\*.example.com)
* Subject Alternate Name (SAN) – special X.509 that allows additional items (IP addresses, domain names etc.)
* Code Signing Certificates – used to sign computer code
* Machine/Computer Certificates – X.509 certificates assigned to specific computer
* Email Certificates – securing email (S/MIME)
* User Certificates – for individual users

**Certificate Formats**

* DER
* PEM
* PFX
* CER
* P12
* P7B



**PKI Components – Certificate Authority (CA)**

* Trusted entities
* Internal AKA self-signed
* External/3rd party (Symantec, GoDaddy etc.)
* Duties

1. Issues certificates
2. Verifies holder of digital certificate
3. Ensures that holders of certificates are who they claim to be

**PKI Components – Registration Authorities (RA)**

* Offloads work from the CA
* Validate user’s/end-point’s identities
* Accept registrations
* Distribute keys
* DOES NOT issue certificates

**PKI Components – Certificate-Signing Request (CSR)**

* Request from applicant to CA to apply for digital certificate
* Includes

1. Applicant’s public key
2. Fully qualified domain name
3. Legally incorporated name of company
4. Address

**PKI Components – Certificate Revocation**

* Process of invalidating a certificate before it’s expiration date often due to private key loss/compromise
* 3 levels – valid, suspended & revoked
* Certificate Revocation List (CRL)

1. Method for distributing certificate revocation information
2. Certificate compared against CRL
3. CRL must be updated & maintained

**PKI Certificate Revocation – OCSP (Online Certificate Status Protocol)**

* Checks certificate status in real-time
* OCSP Stapling

1. Reduces load on CA
2. Allows web server to “staple” time-stamped OCSP response as part of TLS handshake with client
3. Web server now responsible for handling OCSP requests instead of CA

**PKI Components – Certificate Trust Models**

* Single CA

1. Simplest no redundancy
2. Self-signed certificate

* Hierarchical Model

1. Root CA – top of hierarchy, may be offline
2. Intermediate CA – subordinate CAs provide redundancy & load balancing

* Certificate Chaining
* Web of trust

1. Cross-certification model
2. Peer-to-peer trust relationship with other CAs

* Bridge CA – cross-certification model using central point of trust

**Key Escrow**

* Trusted 3rd party maintains keys
* Addresses possibility that a cryptographic key may be lost. If key lost, then data is lost
* Key Recovery Agent – entity that has the ability to recover a key, key components or plain-text messages as needed

**Pinning**

* Hashes of public keys for popular web servers included with apps (Eg. Web browsers)
* Mitigates use of fraudulent certificates
* HTTP Public Key Pinning (HPKP) uses public key pins which are hashed values of the public key communicated to browser client from server in HTTP header