


PURDUE UNIVERSITY

Computer Security

CS 426

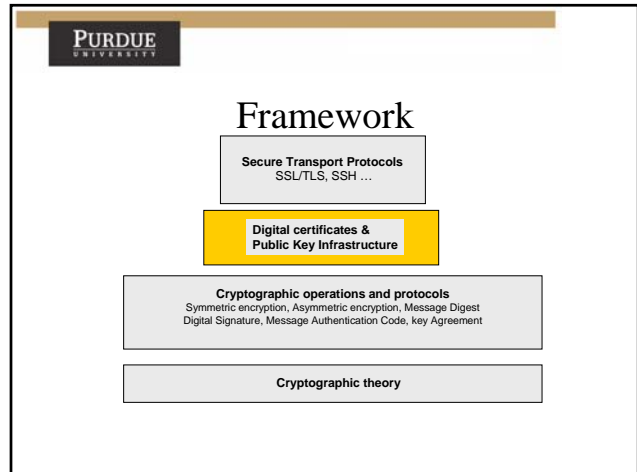
Lecture 7

Public Key Infrastructure (PKI)



Center for Education and Research
in Information Assurance and Security

Elisa Bertino
Purdue University
IN, USA
bertino@cs.purdue.edu



PURDUE UNIVERSITY

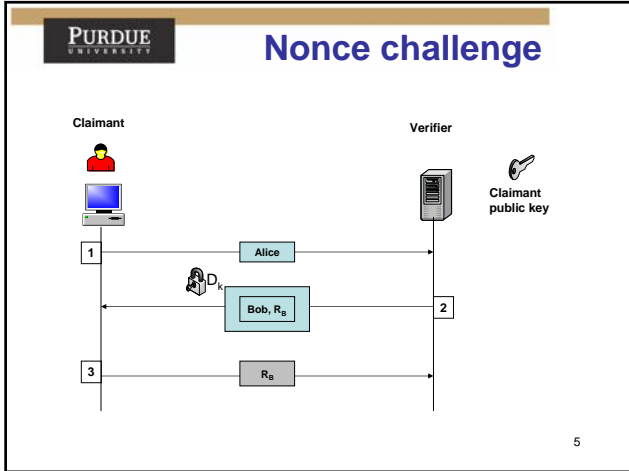
Main points of previous class

- What is entity authentication
- How to perform entity authentication by using a Challenge-Response Protocol (CRP)
- A Challenge Response Protocol (CRP) allows the claimant to prove (to the verifier) that she knows a secret, without sending the secret to the verifier
- We saw how CRP can use symmetric OR asymmetric keys
 - In the first case the claimant and the verifier MUST share a secret key
 - In the second case (use of asymmetric key), the secret is the private key of the claimant

PURDUE UNIVERSITY

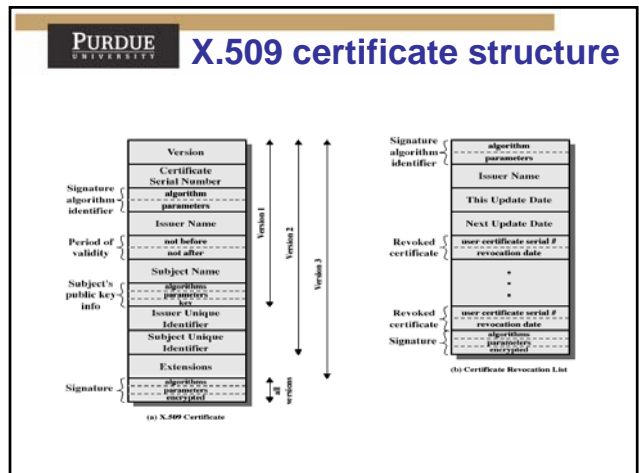
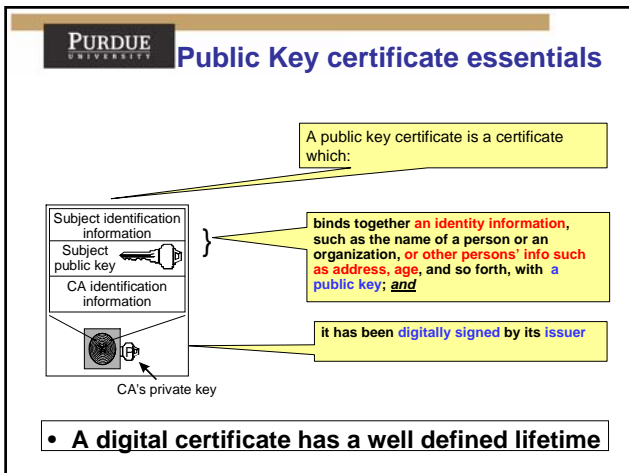
The problem to be solved

- In CRP based on asymmetric keys, the verifier encrypts the challenge using the claimants' public key:
 - How can the verifier be sure that the public key is associated with the claimant?
- In SSL we have the same problem:
 - How can the SSL client be sure that the public key contained in the digital certificate is associated with the server?



PURDUE UNIVERSITY Digital Certificate

- The verifier can be sure that the claimant public key belongs to the claimant because the public key of the claimant is contained in a digital certificate issued and signed by a trusted Certification Authority



X.509 certificate content

- A X.509 certificate is issued by a Certification Authority (CA). It contains the following info:
 - version (1, 2, or 3)
 - serial number (unique within the CA) identifying the certificate
 - signature algorithm identifier
 - issuer X.500 name (CA)
 - period of validity (from - to dates)
 - subject X.500 name (distinguished name –DN -)
 - **CN=Java Duke, OU=Java Software Division, O=Sun Microsystems Inc, C=US**
 - subject public-key info (algorithm, parameters, key)
 - issuer unique identifier (v2+)
 - subject unique identifier (v2+)
 - extension fields (v3)
 - signature (of hash of all fields in certificate)

X.500 Distinguished Name

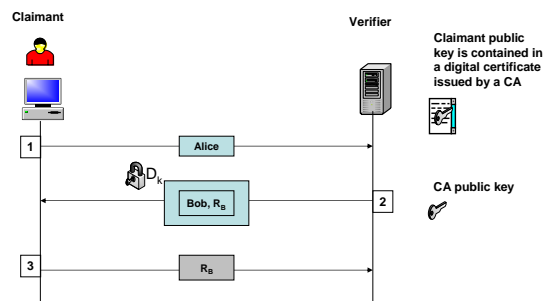
DN Field	Abbrev.	Description	Example
Common Name	CN	Name being certified	CN=Joe Average
Organization or Company	O	Name is associated with this organization	O=Snake Oil, Ltd.
Organizational Unit	OU	Name is associated with this organization unit, such as a department	OU=Research Institute
City/Locality	L	Name is located in this City	L=Snake City
State/Province	ST	Name is located in this State/Province	ST=Desert
Country	C	Name is located in this Country (ISO code)	C=XZ

CA signed Digital Certificate

- A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real:
 - Anyone can verify that what is asserted by the certificate (claims) is true → the public key of the CA must be known

11

CRP using asymmetric keys



12

Self-signed certificates

- A self-signed certificate contains:
 - a public key, information about the owner of the certificate, and the owner's signature.
 - It has an associated private key, but it does not verify the origin of the certificate through a third-party certificate authority (CA)
- When and how to use self-signed certificates?
 - It depends on security requirements
 - To achieve the highest level of authentication between critical software components, do not use self-signed certificates.
- Self-signed certificates can be used to test an SSL configuration before creating and installing a signed certificate issued by a certificate authority

Public keys and digital certificates - issues

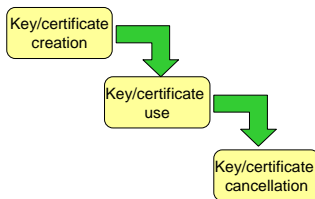
How to securely create, store, distribute, use, destroy (revoke) public keys and digital certificates?



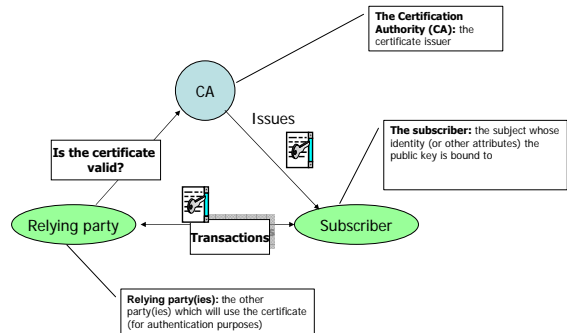
Public Key Infrastructure

Public key certificate life-cycle

A Public Key Infrastructure is a (distributed) infrastructure providing the functionalities and the services needed to support the life-time of public key certificates and their use.



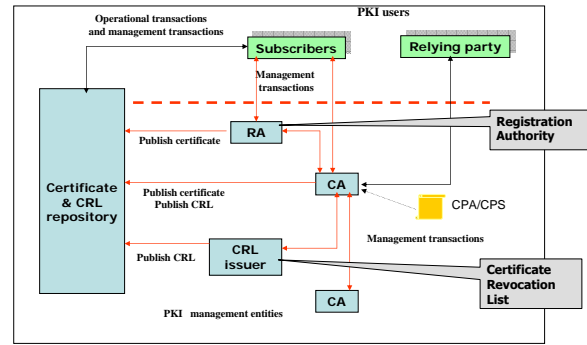
PKI – Main actors



PKI – Main actors

- The certificate issuer: the **Certification Authority (CA)**
- The subject whose identity (or other attributes) the public key is bound to: the **subject** (may be referred to also as **subscriber**)
- The other party(ies) which will use the certificate (for authentication purposes): the **relying party(ies)**
- The subjects and the relying parties are referred to as **end-entities**

PKI - A more detailed picture...



Certification Authority (CA)

- **Certification:**
 - the act of binding an identity information (as well as some other piece of information, such as a permission or a role) with a public key
 - i.e. issuance of a certificate
- **Certification Authority:** the entity responsible of the certification
- A CA operates under a **Certification Practice Statement (CPS)**:
 - A CPS describes the operational procedures of a CA
- A certificate might be issued to the end-entity in accordance with a **Certificate Policy Agreement (CPA)**:
 - A CPA is a high-level statement of requirements/restrictions associated with the intended use of the certificates issued under that policy

Certificate Repository

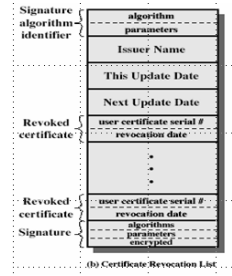
The issuance of the certificate by the CA is not enough. Relying parties need to find easily the public key associated with different subscribers:

- A **Certificate Repository (CR)** is needed.
- A CR can be implemented in several ways:
 - LDAP
 - Web server
 - DNS
 - corporate database, etc.

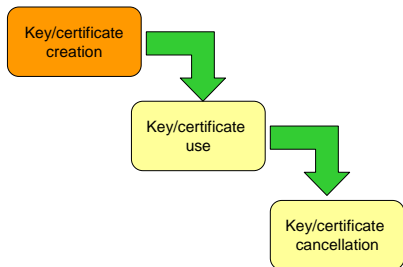
Certificate Revocation and CRL Repository

- Certificate revocation is the mechanism through which relying parties can be alerted/informed that a certificate has been revoked
- **Certificate revocation # certificate expiration**
 - The latter is the date after that the certificate is no longer valid (think of your driver license expiration date)
 - The former is triggered by an event that invalidates the binding between the public key and the subject identity:
 - A transition from a maiden name to a married name
 - The private key has been stolen
 - Revocation of a driver license
- Revoked certificates can be "published" by a CRL issuer in a CRL repository

Certificate Revocation List



Public key certificate life-cycle

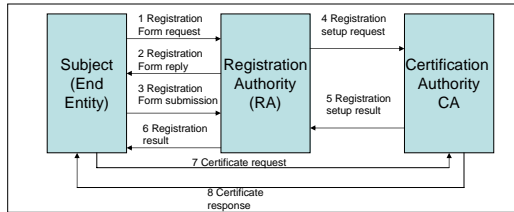


Key/Certificate Creation phase

- Also called Inizialization phase
- It encompasses:
 - Subject registration
 - Key pair generation
 - Certificate creation and Key/Certificate distribution
 - Certificate dissemination
 - Key backup (if appropriate)

Subject Registration

- It is the process in which the identity of an individual user or process is established and verified
- The “strength” of the applied procedural control depends on the CPS/CP:
 - a **Certification Practice Statement (CPS)** describes the operational procedures of a CA
 - a **Certificate Policy (CP)** states the applications which a certifying CA declares a specific public/private key fit for (e.g.: digital signature; encryption of data; verification of Web site identity; others)



Key Pair generation

- A key pair can be generated at different locations:
 - At the end-entity system (e.g. user's PC)
 - At the RA
 - At the CA
 - By a trusted third-party key generation facility
- Factors to consider:
 - Performance (e.g. generating a key pair in a mobile phone)
 - Assurance (if there is a requirement to generate the key pair according to specific cryptographic guidelines – e.g. FIPS 140-1)
 - Intended key usage (e.g.: confidentiality vs. non-repudiation)

Key Pair generation Intended key usage

- Intended key usage (e.g.: confidentiality vs. non-repudiation)
 - The underlying principle is to use distinct key pairs for different purposes/services:
 - One key pair is used for digitally signature (and the corresponding certificate is termed *verification certificate*)
 - Another key pair is used for encryption purposes (and the corresponding certificate is termed *encryption certificate*)
- The intended key usage impacts on the choice of the key generation location
 - If keys are used for non-repudiation, then they may be generated at the end-entity system → the private key is possessed only by the generating end-entity
 - However, it can be argued also that the CA is the most trusted entity in PKI, and so it can know the private key also

Key Pair generation PKI Tools at end-entity

- **Java environment:** keytool
 - used to create and manage key stores
 - functions provided:
 - Create public/private key pairs
 - Display, import, and export X.509 v1, v2, and v3 certificates stored as files
 - Create self-signed certificates
 - Issue certificate (PKCS#10) requests to be sent to CAs
 - Import certificate replies (obtained from the CAs which certificate requests were sent to)
 - Designate public key certificates as trusted

Certificate creation

- **Certificate creation:**
 - **By a CA**
 - If the public key was generated by an entity \neq CA, then the public key must be securely conveyed to the CA
 - **At the end-entity**

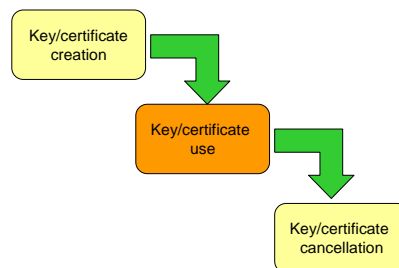
Key/Certificate distribution

- The (public) key and the certificate can be delivered:
 1. Directly to the (subject) owner
 2. To a remote repository (see Certificate and CRL repository)
 3. To both 1 and 2
- Requesting and receiving a certificate back from a CA requires the use of a secure protocol:
 - RFC2510 The Internet X.509 Public Key Infrastructure Certificate Management Protocols (CMP)
 - RFC2511 The Internet X.509 Certificate Request Message Format (CRMF)

Certificate dissemination

- How to make available the certificate to all possible relying parties?
 - Out-of-band, private dissemination (e.g. physical delivery)
 - Posting certificates in a widely known public repository
 - It allows on demand and on-line retrieval
 - In-band protocol distribution (e.g. including the certificate with a secure e-mail message – S/MIME)

Public key certificate life-cycle



Key/Certificate Use Phase

- Certificate retrieval (from a certificate repository)
 - Upon request of a relying party
- Certificate validation (upon request of a relying party)
 - Verify the integrity of the certificate (i.e. verify the issuer digital signature) → **it requires the corresponding CA public key**
 - The certificate has not expired
 - The certificate has not been revoked
- Key recovery
 - Applies to the private decryption keys, if the end-entities loose access to them
- Key update
 - The issuance of a new key pair when a certificate is near to its expiration time

Certificate Validation Certification paths

- A subscriber is initialized with a limited number of assured CA public keys
- If the public key user does not already hold an assured copy of the public key of the CA that signed the certificate, then it might need an additional certificate to obtain that public key
- A chain of multiple certificates may be needed, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs (certification path)

Certification paths and PKI Trust Models

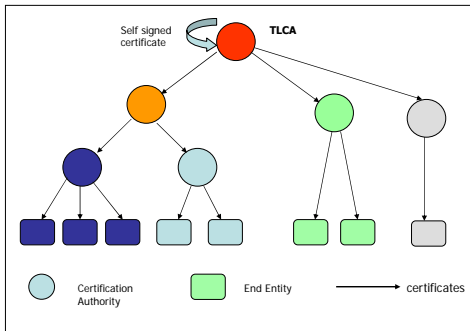
The problems to be solved:

- Which certificates an entity can trust?
- How can such trust be established?
- Under which circumstances can this trust be limited or controlled in a given environment?

PKI Trust Models

- Several trust models:
 - Hierarchy of CAs
 - Policy-based CA hierarchichies
 - Distributed trust architecture
 - Four-corner model
 - Web model
- Trust in this context has the following operational meaning:
 - *An entity can be said to trust another entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects*

Hierarchy of CAs



Hierarchy of CAs

- A Top Level CA (TLCA) – also called Root CA is established
 - The TLCA issues a self-signed certificate as the basis of trust for all the entities belonging to the hierarchy
 - The TLCA certifies zero or more CAs immediately below it
 - In turn, each of those CAs certifies zero or more CAs immediately below it
 - At the second-to-last level, the CAs can certify end-entities
- Each entity in the hierarchy (intermediate CA and end-entities) must be supplied with a copy of the TLCA's public key
- The installation of this public key is the foundation for the certificate processing for all subsequent communication in this model

CA hierarchy

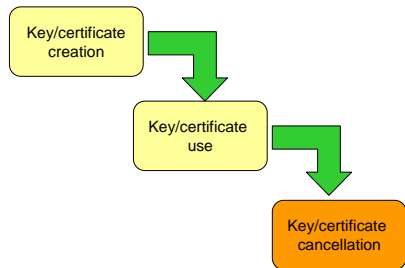
- If both users share a common CA then they are assumed to know the CA's public key
- Otherwise CA's must form a hierarchy
- Linking members of the CA hierarchy to validate other CA's
 - each CA has certificates for clients (forward) and parent (backward)
- Each client trusts parents certificates
- This enables verification of any certificate from one CA by users of all other CAs in hierarchy

Self-signed certificates

A self-signed certificate contains:

- a public key, information about the owner of the certificate, and the owner's signature.
- It has an associated private key, but it does not verify the origin of the certificate through a third-party certificate authority (CA)

Public key certificate life-cycle



Key/Certificate Cancellation

- Certificate expiration – The CA can do 3 possible actions :
 - No action - the end-entity is no longer enrolled in the PKI
 - Certificate renewal – the **same** public key is placed into a new certificate with a new validity period
 - Certificate update – a new key pair is generated and a new certificate is issued
- Certificate revocation
 - An issued certificate might be no longer valid even if it has not yet expired:
 - Suspected private key compromise
 - Changes to the status of the original requesting end-entity

Certificate Revocation

- Who and how should manage certificate revocation?
- WHO:
 - CAs are responsible for posting revocation information
 - Relying party must ascertain that a certificate is not revoked
- CA can post revocation information through Periodic publication lists - [Certificate Revocation List \(CRL\)](#)
- On-line query mechanisms – [Online Certificate Status Protocol \(OCSP\)](#)

Java & Windows
certificate-related resources

- Java:
 - Java JDK 5.0 Security
<http://java.sun.com/j2se/1.5.0/docs/guide/security/index.html>
 - Managing X.509 certificates in Java:
<http://java.sun.com/j2se/1.5.0/docs/guide/security/cert3.html>
 - Java Keytool for Windows
<http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html>
- Windows server 2003:
 - Windows Server 2003 and Windows 2000 Server, Certificate Services 2.0 <http://msdn2.microsoft.com/en-us/library/aa376539.aspx>
 - <http://technet2.microsoft.com/windowsserver/en/library/d01a80dd-479a-444b-8893-68c40d61dd9c1033.mspx?mfr=true>
 - Certificate Services is a service running on a Windows server operating system:
 - It receives requests for new digital certificates over transports such as RPC or HTTP. It checks each request against custom or site-specific policies, sets optional properties for a certificate to be issued, and issues the certificate.
 - It allows administrators to add elements to a *certificate revocation list* (CRL), and to publish signed CRLs on a regular basis.
 - Certificate services include programmable interfaces

PKI products & services

Many vendors provide PKI solutions (software products and services):

- Entrust
- Verisign
- RSA Security
- Thawte
-

Certificate Authority: Entrust

Entrust

- [Entrust Authority™ Security Manager](http://www.entrust.com/public-key-infrastructure/index.htm) (<http://www.entrust.com/public-key-infrastructure/index.htm>) is a Certification Authority (CA) system responsible for issuing and managing users' digital identities
- It provides for
 - securely storing the certificate authority (CA) private key
 - issuing certificates for users and devices
 - publishing certificate revocation lists (CRLs) that are used to verify whether a user or application's certificate is still trusted by the CA that issued it
 - maintaining an auditable database of users' private key histories for recovery purposes in the event that users lose access to their keys

Verisign

- Key management services:
 - centralized key generation
 - distribution and backup capabilities
 - archiving of key histories
 - dual key pair support, coupled with a two-step recovery process
- It does not require proprietary client software

Verisign ECA Certificates

ECA certificates (source:

<http://www.verisign.com/authentication/government-authentication/eca-certificates/index.html>):

- VeriSign is certified by the United States Department of Defense (DOD) as a provider of digital certificates for external entities (government contractors, state and local governments and individuals)
- External Certification Authority (ECA) certificates enable secure on-line transactions with government agencies. Installed in a browser or email program, ECA certificates can be used for authenticating identity for access to Web sites and applications, to digitally sign documents, and for encrypting e-mail communications.

RSA Digital Certificate Solutions

See

<http://www.rsa.com/node.aspx?id=2604>

Thawte (www.thawte.com)

- SPKI/Managed Multiple Certs:
 - enables organizations that require multiple digital certificates to be issued in a manner that saves them considerable time, effort and money
 - SSL web server certificates
 - Code signing certificates
 - Premium Server Gated Cryptography SSL certificates
- See <http://www.thawte.com/managed-multiple-certificates-spki/index.html?click=main-nav-products-spki>

How to generate a digital certificate with Thawte?

- Before you can begin the process of obtaining a Certificate, you must generate a Private Key and CSR pair off the web server.
- A CSR is basically a Public Key that you generate on your server that validates the computer-specific information about your web server and Organization when you request a Certificate from *thawte*.
- Digital ID's make use of a technology called Public Key Cryptography, which uses Public and Private Key files.
- The Public Key, also known as a Certificate Signing Request (CSR), is the key that will be sent to *thawte*.
- The Private Key is generated locally on the server, will remain on the server and should never be released into the public. *thawte* does not have access to the Private Key. The Private Key is never transmitted to *thawte*. The integrity of the Digital ID depends on the private key being controlled exclusively by the entity to which the key belongs.

Books

Carlisle Adams & Steve Lloyd
 "Understanding PKI – concepts, standards and deployment considerations - second edition
 Addison Wesley 2005 ISBN 0-672-32391-5

References

- RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile. Available at <http://www.ietf.org/rfc/rfc2459.txt>
- RFC 3280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Available at <http://www.ietf.org/rfc/rfc3280.txt>
- RFC 2560 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. Available at <http://www.ietf.org/rfc/rfc2560.txt>
- RFC 3647 Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. Available at <http://www.fqs.org/rfcs/rfc3647.html>
- Carl Ellison and Bruce Schneier Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure. In Computer Security Journal Volume XVI, Number 1, 2000. Available at <http://www.schneier.com/paper-pki.pdf>