



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Computer Security and Cryptography

CS381

来学嘉

计算机科学与工程系 电院3-423室

34205440 1356 4100825 laix@sjtu.edu.cn

2015-05



Organization



- Week 1 to week 16 (2015-03 to 2014-06)
- 东中院-3-102
- Monday 3-4节; week 9-16
- Wednesday 3-4节; week 1-16
- lecture 10 + exercise 40 + random tests 40 + other 10
- Ask questions in class – counted as points
- Turn ON your mobile phone (after lecture)
- Slides and papers:
 - <http://202.120.38.185/CS381>
 - computer-security
 - <http://202.120.38.185/references>
- TA: Geshi Huang gracehgs@mail.sjtu.edu.cn
- Send homework to the TA

Rule: do the homework on your own!



Contents



- Introduction -- What is security?
- Cryptography
 - Classical ciphers
 - Today's ciphers
 - Public-key cryptography
 - Hash functions and MAC
 - Authentication protocols
- Applications
 - Digital certificates
 - Secure email
 - Internet security, e-banking
- Computer and network security
 - Access control
 - Malware
 - Firewall
- Examples: Flame, Router, BitCoin ??



References



- W. Stallings, *Cryptography and network security - principles and practice*, Prentice Hall.
- W. Stallings, 密码学与网络安全：原理与实践（第4版），刘玉珍等译，电子工业出版社，2006
- Doug Stinson , *Cryptography Theory and Practice*, Third Edition, CRC Press, 2005
- Lidong Chen, Guang Gong, *Communication and System Security*, CRC Press, 2012.
- A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997, ISBN: 0-8493-8523-7,
<http://www.cacr.math.uwaterloo.ca/hac/index.html>
- B. Schneier, *Applied cryptography*. John Wiley & Sons, 1995, 2nd edition.
- Protocols Lounge, <http://sky.fit.qut.edu.au/~choo/lounge.html>
- Berry Schoenmakers, Cryptographic Protocols, www.win.tue.nl/~berry/2WC01
- 裴定一,徐祥, 信息安全数学基础, ISBN 978-7-115-15662-4, 人民邮电出版社,2007.



Kerberos



- **centralised** secret-key third-party **authentication** in a **distributed network**
- 1983-1991 MIT Project Athena, for authentication in campus networks
- 1988, Kerberos v4 used by :
 - MS windows / MAC OS X / RedHat Linux 4



P R O J E C T
A T H E N A

MASSACHUSETTS INSTITUTE OF TECHNOLOGY



(Greek) Kerberos- 3-headed hound which guards the gates of the Underworld.

3: authentication, clearing, auditing



Kerberos



- allows users access to services distributed through network
- without needing to trust all workstations
- rather all trust a central authentication server
- two versions in use: 4 & 5



Kerberos v4 Dialogue

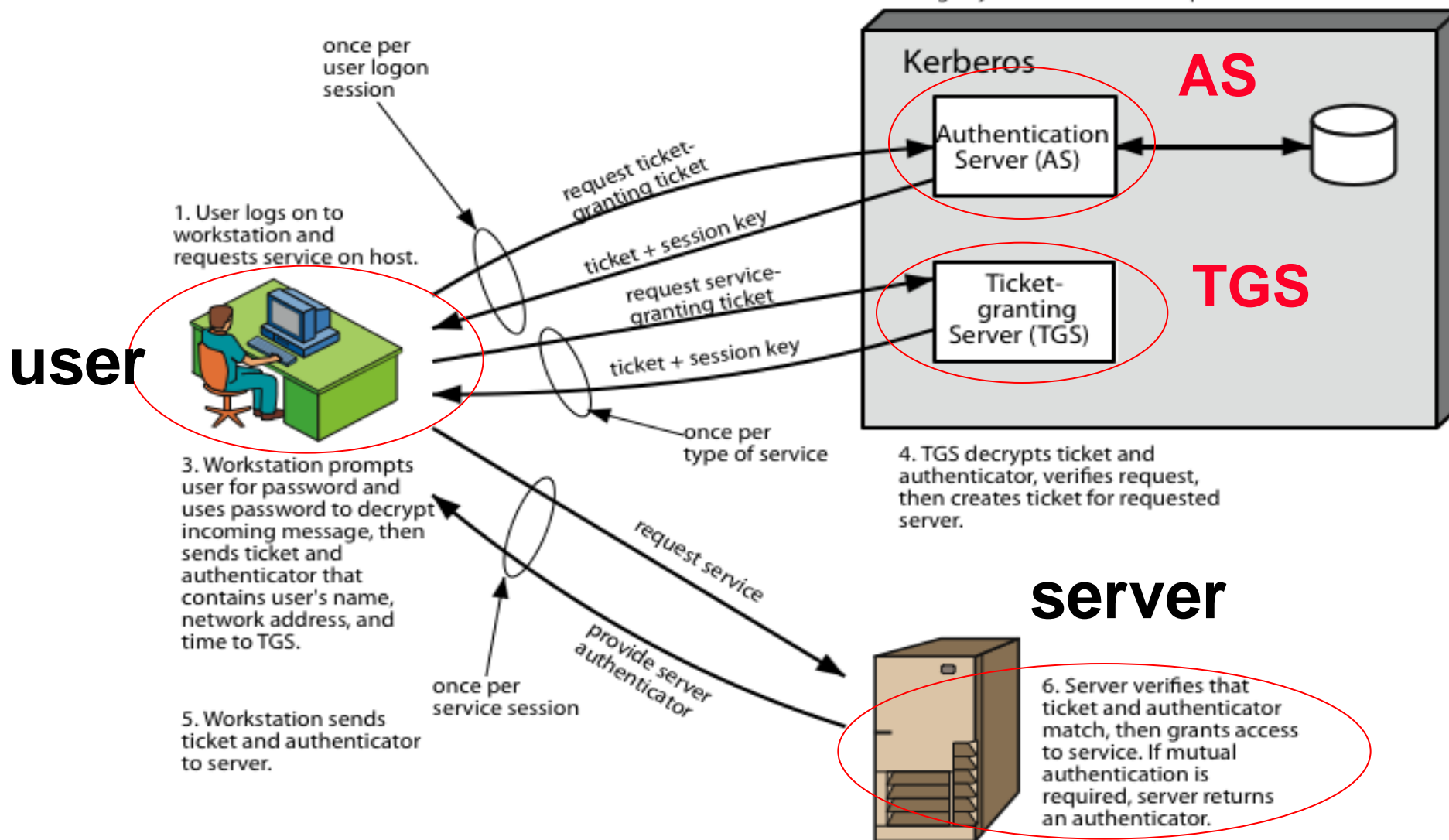


users authenticate to Authentication Server (AS)

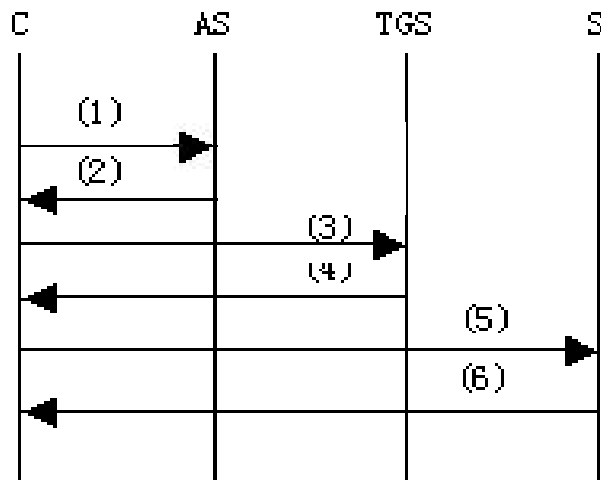
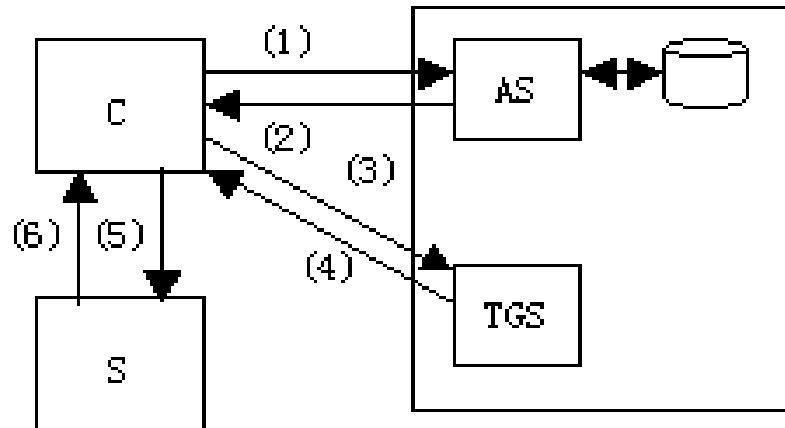
1. obtain **ticket granting ticket** from **AS**
 - once per session
2. obtain **service granting ticket** from **TGT**
 - for each distinct service required
3. client/server exchange to obtain service
 - on every service request



Kerberos v4



Kerberos v4



Kerberos v4 Dialogue



(1) C --> AS: $ID_C || ID_{tgs} || TS_1$

(2) AS --> C: $E_{K_c}[K_{c,tgs} || ID_{tgs} || TS_2 || Lifetime_2 || \text{Ticket}_{tgs}]$

$\text{Ticket}_{tgs} = E_{K_{tgs}}[K_{c,tgs} || ID_C || AD_C || ID_{tgs} || TS_2 || Lifetime_2]$

(3) C --> TGS: $ID_v || \text{Ticket}_{tgs} || \text{Authenticator}_C$

$\text{Authenticator}_C = E_{K_{c,tgs}}[ID_C || AD_C || TS_3]$

(4) TGS --> C: $E_{K_{c,tgs}}[K_{c,v} || ID_v || TS_4 || \text{Ticket}_v]$

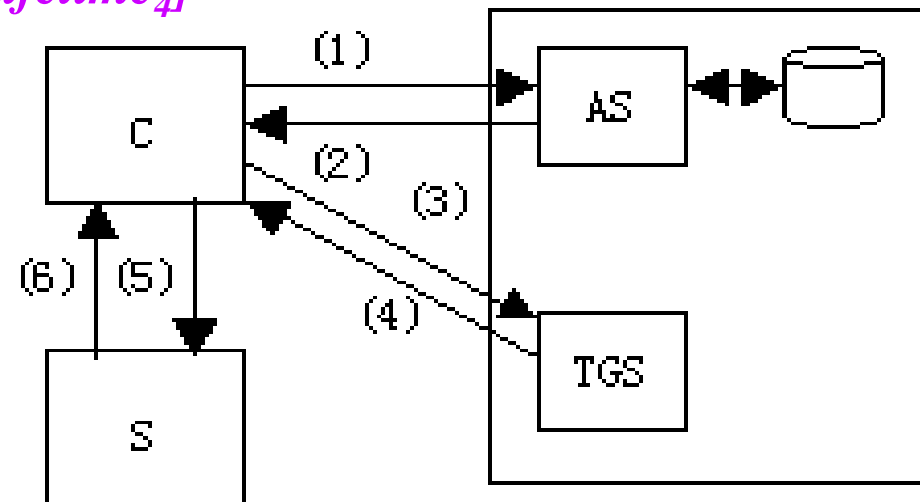
$\text{Ticket}_v = E_{K_v}[K_{c,v} || ID_C || AD_C || ID_v || TS_4 || Lifetime_4]$

(5) C --> V: $\text{Ticket}_v || \text{Authenticator}_C$

$\text{Authenticator}_C = E_{K_{c,v}}[ID_C || AD_C || TS_5]$

(6) V --> C: $E_{K_{c,v}}[TS_5 + 1]$

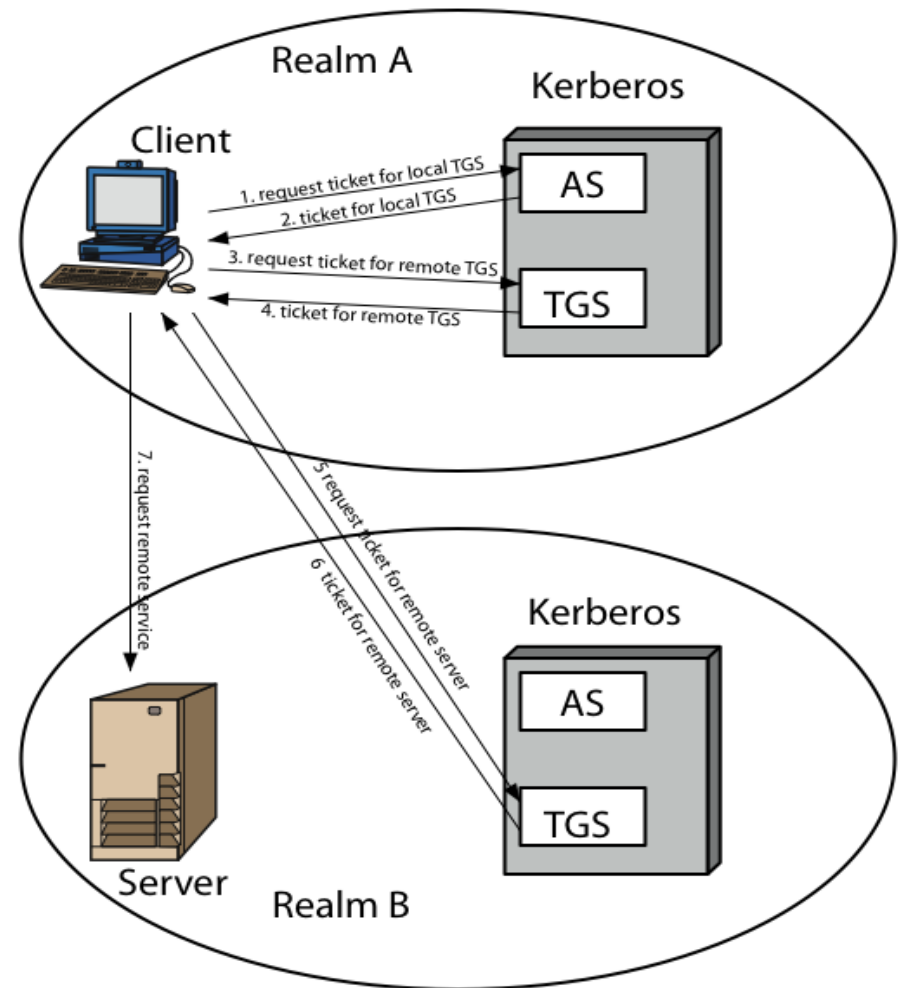
Ticket_{tgs} can be reused within lifetime



Kerberos realms



- a Kerberos **realm**:
 - a Kerberos server
 - users registered with Kerberos server
 - application servers, sharing keys with server
- typically a single administrative domain
- if have **multiple realms**, their Kerberos servers must share keys and trust





Kerberos Version 5



- developed in mid 1990's
- specified as Internet standard RFC 1510
- provides improvements over v4
 - addresses environmental shortcomings
 - encryption alg, network protocol, byte order, ticket lifetime, authentication forwarding, interrealm auth
 - and technical deficiencies
 - double encryption, non-std mode of use, session keys, password attacks

Kerberos v5



- Kerberos v5 improves v4

(1) $C \rightarrow AS: Options || ID_C || Realm_C || ID_{tgs} || Times || Nonce_1$

(2) $AS \rightarrow C: Realm_C || ID_C || \text{Ticket}_{tgs} || E_{K_C}[K_{c,tgs} || Times || Nonce_1 || Realm_{tgs} || ID_{tgs}]$

$\text{Ticket}_{tgs} = E_{K_{tgs}}[Flags || K_{c,tgs} || Realm_C || ID_C || AD_C || Times]$

(3) $C \rightarrow TGS: Options || ID_V || Times || Nonce_2 || \text{Ticket}_{tgs} || \text{Authenticator}_C$

$\text{Authenticator}_C = E_{K_{c,tgs}}[ID_C || Realm_C || TS_1]$

(4) $TGS \rightarrow C: Realm_C || ID_C || \text{Ticket}_V || E_{K_{c,tgs}}[K_{c,v} || Times || Nonce_2 || Realm_V || ID_V]$

$\text{Ticket}_V = E_{K_V}[Flags || K_{c,v} || Realm_C || ID_C || AD_C || Times]$

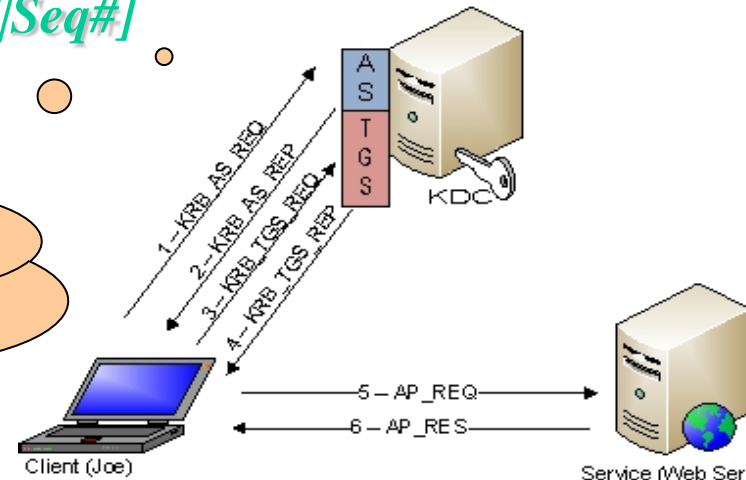
(5) $C \rightarrow V: Options || \text{Ticket}_V || \text{Authenticator}_C$

$\text{Authenticator}_C = E_{K_{c,v}}[ID_C || Realm_C || TS_2 || \text{Subkey} || \text{Seq\#}]$

(6) $V \rightarrow C: E_{K_{c,v}}[TS_2 || \text{Subkey} || \text{Seq\#}]$

Subkey and Seq are optional;
Subkey: session-key, default is $K_{c,v}$;
Seq#: the session id, to prevent replay.

SJTU





PKI



- asymmetric cipher, public-key cryptosystem:
user share some **trusted** information
- Certificates establish such trust on public-keys
- PKI (public-keys infrastructure) is the managing system of certificates



RSA public-key encryption



- Is PK_B belong to Bob? ----trust

Alice $PK_A = (n_A, e_A)$
 $SK_A = (p_A, q_A, d_A)$

Bob $PK_B = (n_B, e_B)$
 $SK_B = (p_B, q_B, d_B)$

Get PK_B ,
Compute C

$$C = E_{PK_B}[M] = (M)^{e_B} \bmod n_B$$

$$C^d = (M^e)^d = M^{k\phi(n)+1} = M^{k\phi(n)} M = M$$

$$M = E_{SK_B}[C] = (C)^{d_B} \bmod n_B$$



Trust on public-key



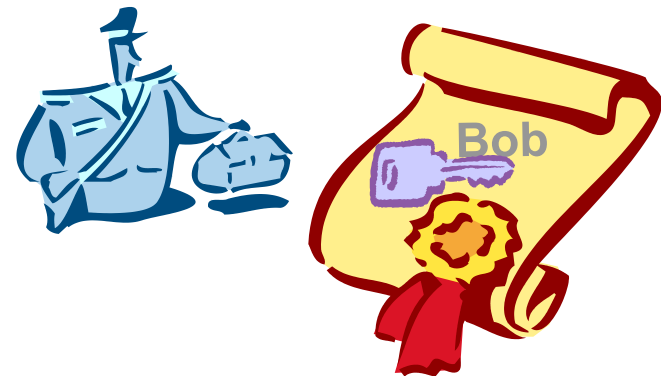
- Trust: binding between identity and public-key
- Ways to establish trust
 - Direct: by person, post, phone.
 - Indirect: by introduction (PGP)
 - Use TTP: TTP signed public-key, i.e., **certificate**.
 - PKI is a framework for managing the certificates
 - Other: Identity-based: use the identity (name, address,..) as public-key.



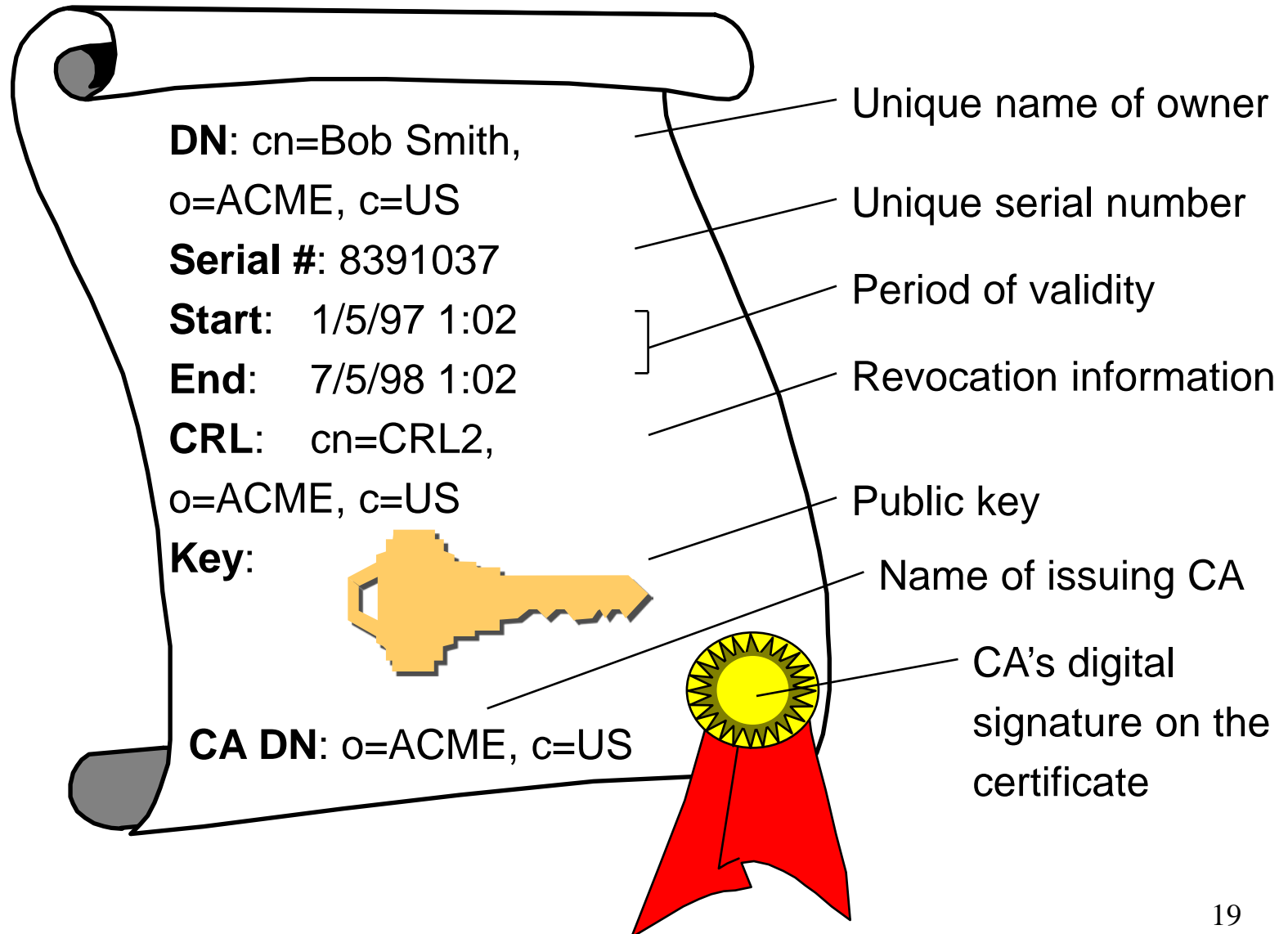
Public-key Certificates



- Certificates establish **trust** in public keys
 - through a binding of user-ID and Public-key,
 - by TTP's signature
- Certificates for:
 - encryption public keys
 - verification public keys
- Certification Authority (CA)
 - trusted third-party



What is in a certificate?



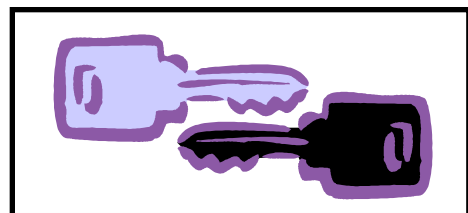


Basic PKI components

- **Registration Authority (RA):** binding the user credential with his public keys – **trust establishment**
- **Certification Authority (CA):** Issue certificates – (user's public-key, CA's signature)
- **Certificate Repository / Directory:** storing certificates – **for public accessing**

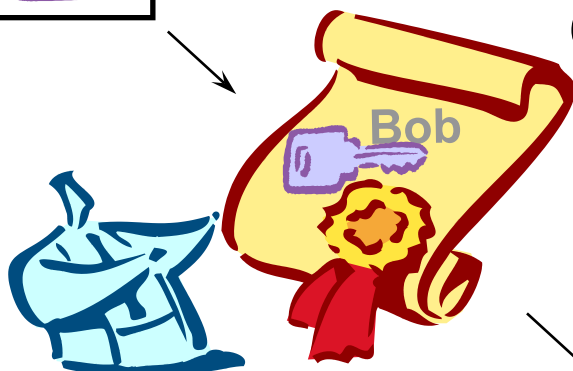


Key Lifecycle

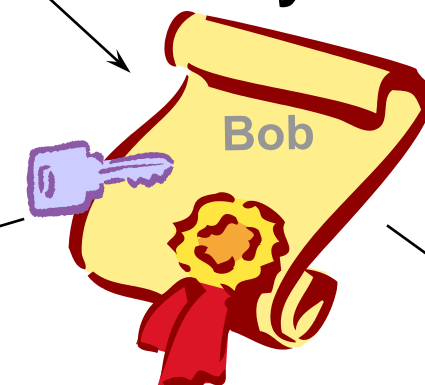


Key Generation

Certificate Issuance



Key Usage



Certificate Validation



Key Expiry



Key Update





Different lifecycle of 2 key pairs

	public	private
encryption	Encryption key destroy old keys	Decryption key keep after expiration
signature	Verification key kept after expiration	Signing key destroy old keys



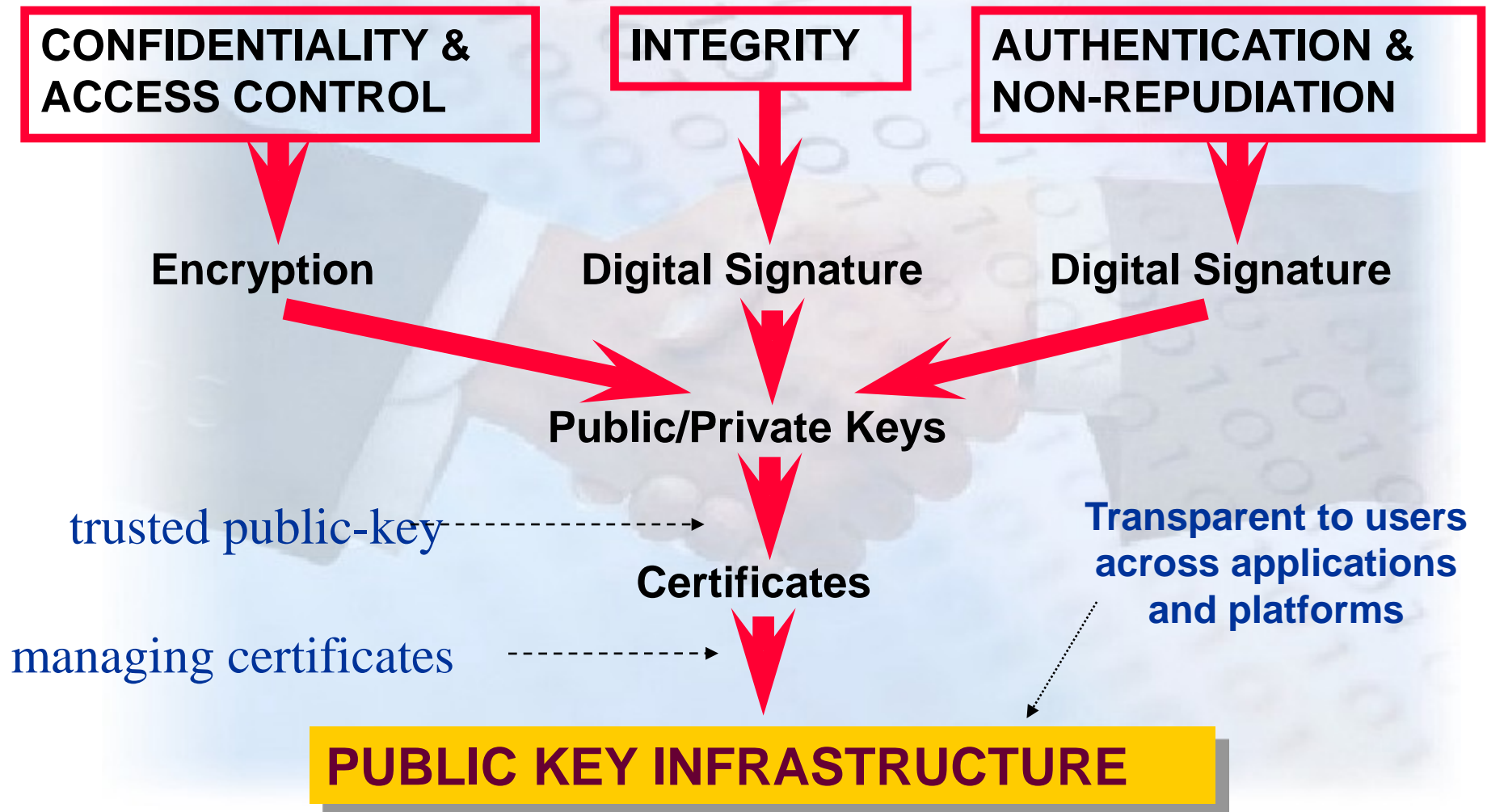
Principle of key separation



Use different key pairs for encryption and signing

- Although the same key pair can be used for both encryption and signing, it is better to use 2 pairs.
 - **Different life cycle** (private decryption key and public verification key should be kept after expiration; public encryption key and private signing key should be destroyed after expiration)
 - **To prevent oracle attacks** (use sign to decrypt, use decrypt to sign)
 - **Good practice for forward-security** (compromise of key for one application does not effect the security of other applications)
 - **Required when both non-repudiation and key recovery are needed.**

PKI (Public Key Infrastructure)





ITU-T X.509 (V4)



- ITU-T Recommendation X.509 (2000) / ISO/IEC 9594-8 (2001):
- Information Technology
 - Open Systems Interconnection
 - The Directory:
 - Public-Key and Attribute Certificate Frameworks



Standard organizations

- ISO: <http://www.iso.ch>
- IETF: <http://www.ietf.org/>
- ITU-T: <http://www.ituaj.or.jp/book/itut-rec.html>
- NIST FIPS: <http://www.itl.nist.gov/fipspubs/by-num.htm>
- 3GPP: <http://www.3gpp.org/>
- ANSI: <http://www.ansi.org/>
- CEN: <http://www.cenorm.be/>
- Common criteria: <http://csrc.nist.gov/cc/>
- ETSI: <http://www.etsi.org/main.htm>
- 全国信息安全标准技术委员会: <http://www.tc260.org.cn/>
- OID: <http://oid.elibel.tm.fr>

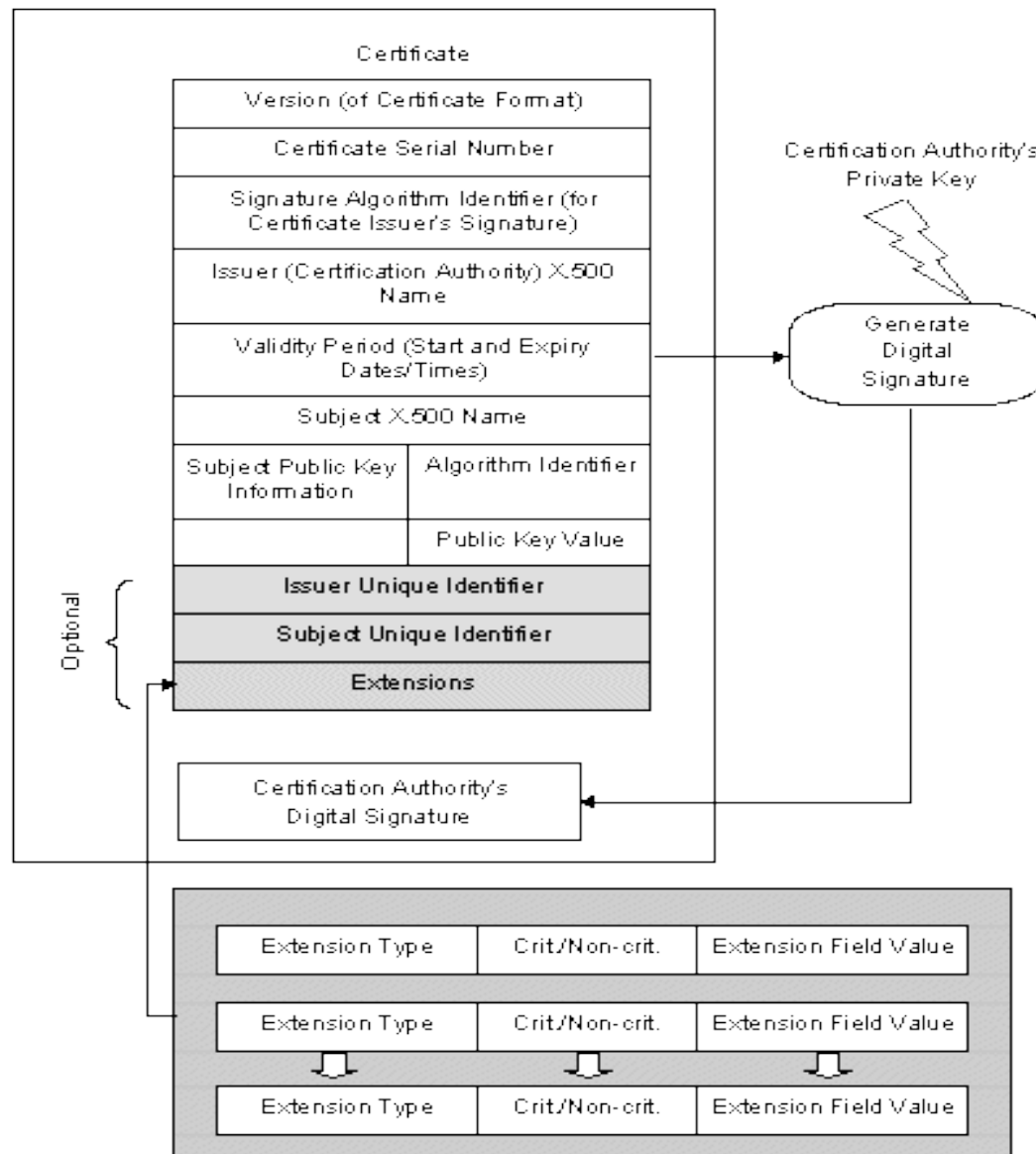


Frameworks overview



- a framework for obtaining and trusting a public key of an entity in order to
 - **encrypt information** to be decrypted by that entity,
 - or in order to **verify the digital signature** of that entity.
- The framework includes
 - the **issuance of a public-key certificate by a Certification Authority (CA)** and
 - the **validation of that certificate by the certificate user**:
 - establishing a trusted path of certificates between the certificate user and the certificate subject;
 - verifying the digital signatures on each certificate in the path;
 - validating all the certificates along that path (i.e. that they were not expired or not revoked at a given time).

X.509 Version 3 Digital Certificate





Fields in specification



- **Version:** Indicator of version 1,2 or 3.
- **Serial number:** Unique identifying number for this certificate.
- **Signature Algorithm identifier** of the digital signature algorithm of CA.
- **Issuer:** X.500 name of the issuing CA.
- **Validity:** Start and expiration dates and times of the certificate.
- **Subject:** X.500 name of the holder of the private key (subscriber).
- **Subject public-key information:** The value of the public-key for the subject together with an identifier of the algorithm with which this public-key to be used.
- **Issuer unique identifier:** An optional bit string used to make the issuing certification authority name unambiguous.
- **Subject unique identifier:** An optional bit string used to make the subject name unambiguous.
- **signature** (of hash of all fields in certificate)



Certificate specification of ASN.1 data type

- Certificate ::= SIGNED { SEQUENCE {
 version [0] Version DEFAULT v1,
 serialNumber CertificateSerialNumber,
 signature AlgorithmIdentifier,
 issuer Name,
 validity Validity,
 subject Name,
 subjectPublicKeyInfo SubjectPublicKeyInfo,
 issuerUniqueIdentifier [1] IMPLICIT UniqueIdentifier OPTIONAL,
 -- if present, version must be v2 or v3
 subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL,
 -- if present, version must be v2 or v3
 extensions [3] Extensions OPTIONAL
 -- If present, version must be v3 -- } }
- ASN.1--Abstract Syntax Notation number One



certification path

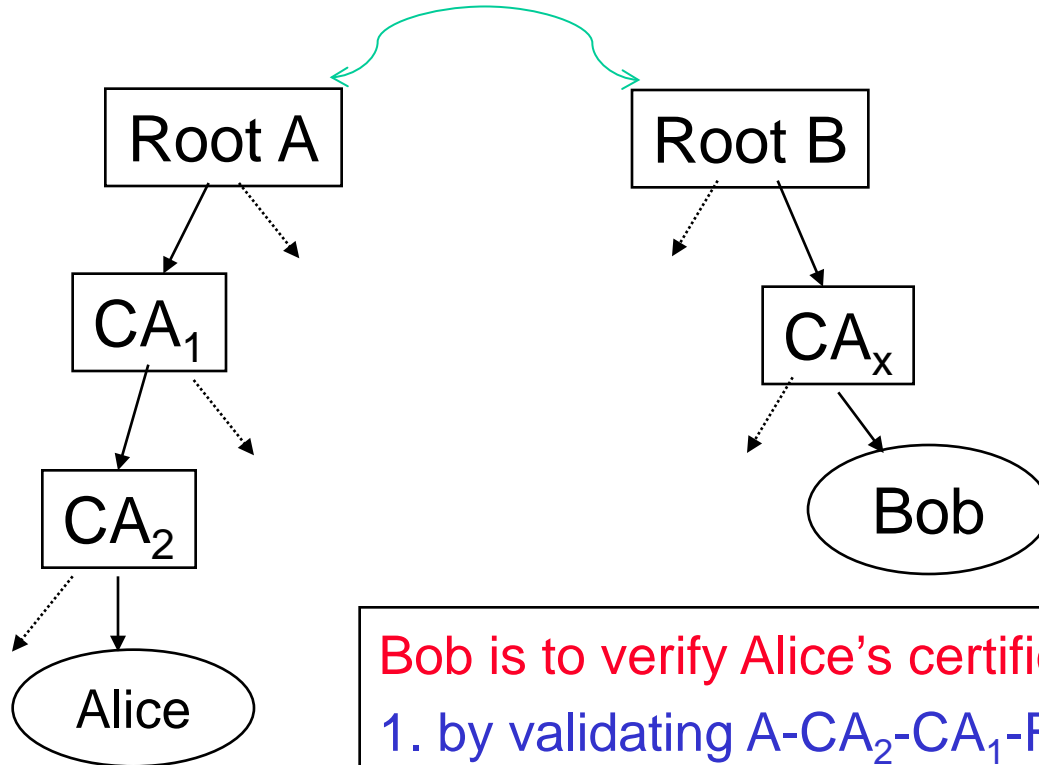


- *certification path*: A **list of certificates** needed to allow a particular **user** to trust the public key of another
 - certification path logically forms **an unbroken chain of trusted points** in the Directory Information Tree **between two users wishing to authenticate**

Certificates	::=	SEQUENCE {
userCertificate		Certificate,
certificationPath		ForwardCertificationPath OPTIONAL }
CertificationPath	::=	SEQUENCE {
userCertificate		Certificate,
theCACertificates		SEQUENCE OF CertificatePair OPTIONAL }



Certification path – 4 methods



Bob is to verify Alice's certificate

1. by validating A-CA₂-CA₁-RootA; Trust (certificate) of RootA should be in B's browser
2. By cross certification – between RootA and RootB
3. Bob can install RootA.
4. Direct trust (highest)



Certificate Revocation



- certificates have a period of validity
- may need to revoke before expiry, eg:
 1. user's private key is compromised
 2. user is no longer certified by this CA
 3. CA's certificate is compromised
- CA's maintain list of revoked certificates
 - the Certificate Revocation List (CRL)
- users should check certificates with CA's CRL

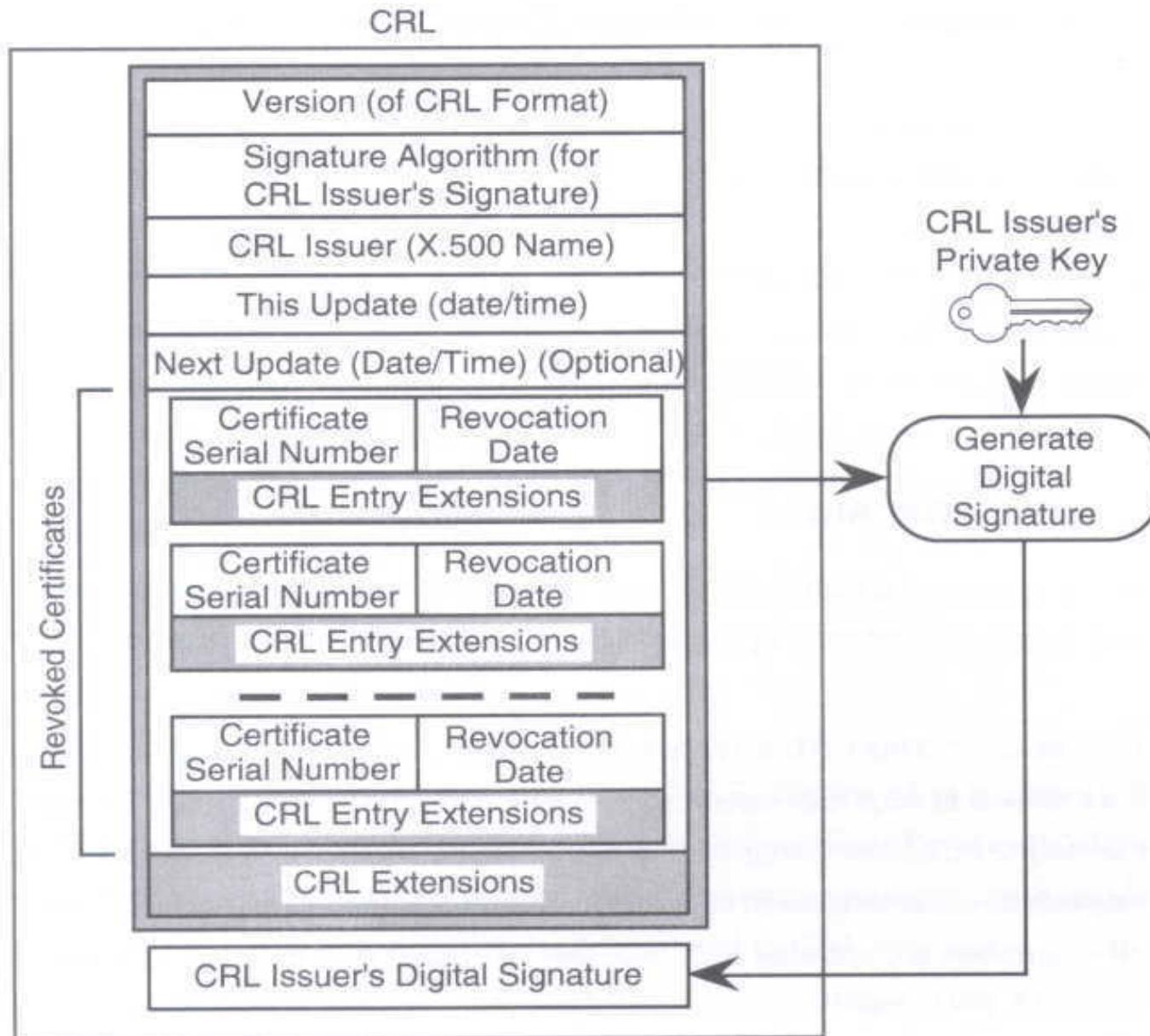


Certificate Revocation



- Revocation: Canceling a certificate before its originally scheduled validity period
 - Compromised key, change name, ...
- **Certificate Revocation Lists (CRL)**
 - A CRL is a time-stamped list of revoked certificates that has been digitally signed by a CA and made available to certificate users.
 - Each revoked certificate is identified in a CRL by its certificate serial number – generated by the issuing CA.
- Types of revocation lists:
 - Certificate revocation list (CRL)
 - Authority revocation list (ARL)
 - Delta revocation list
- **CRL checking:**
 - CRL Distribution Point: Identifies the point or points that distribute CRL's on which a revocation of this certificate would appear if this certificate were to be revoked.
 - Delta-CRL's: It is a digitally signed list of the changes that occurred since the issuance of the prior base CRL.
 - Online Status Checking: uses OCSP (Online Certificate Status Protocol)

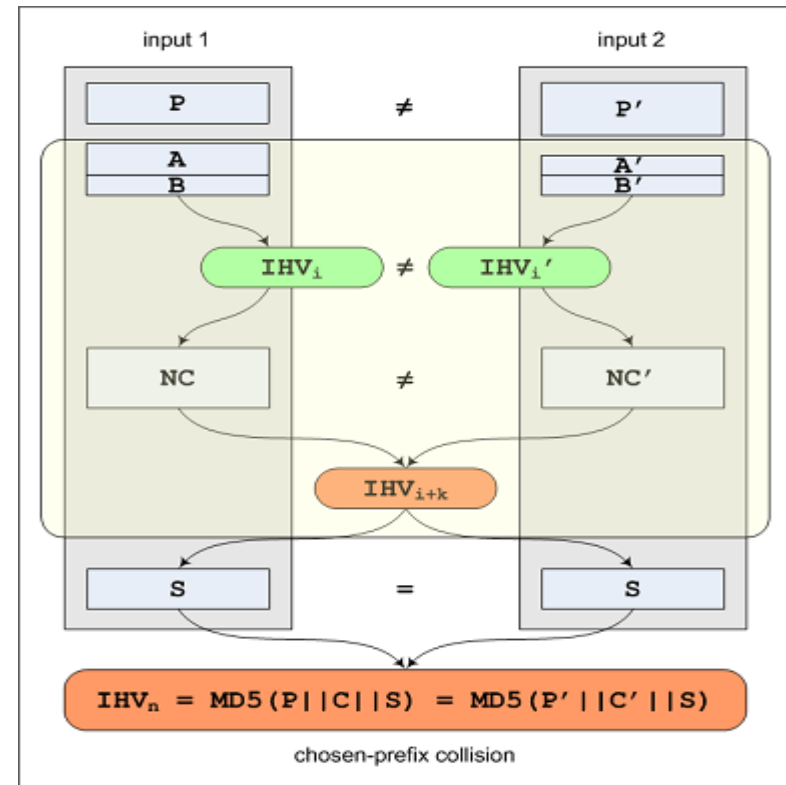
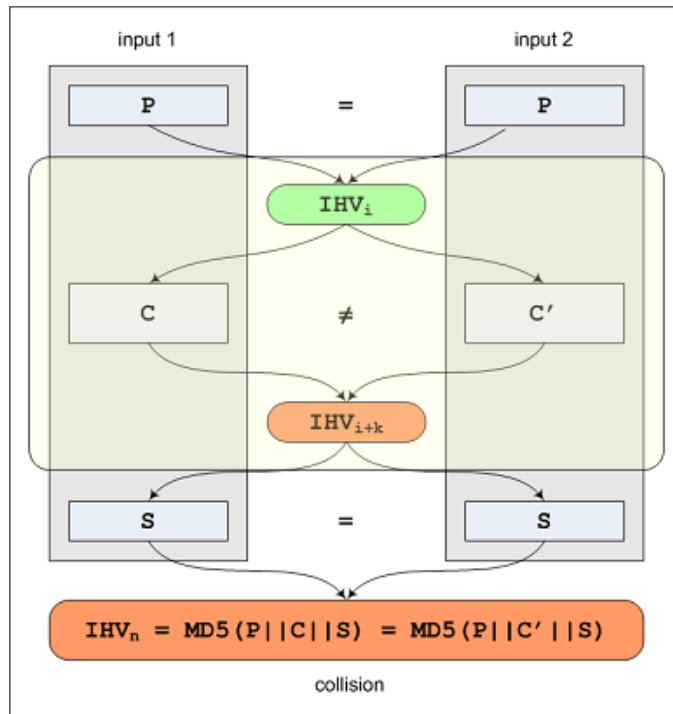
X.509 Certificate Revocation List



MD5 collision – chosen-prefix collision



- “rogue certificates” [M. Stevens,,09] <http://eprint.iacr.org/2009/111>
 - 2 certificates with different data fields (especially CA=TRUE/FALSE) and public-keys, but with same MD5 hash code.
 - free-start collision: $\text{comp.} = 2^{16}$



real certificate

[illegible]

CA=true

CA=False



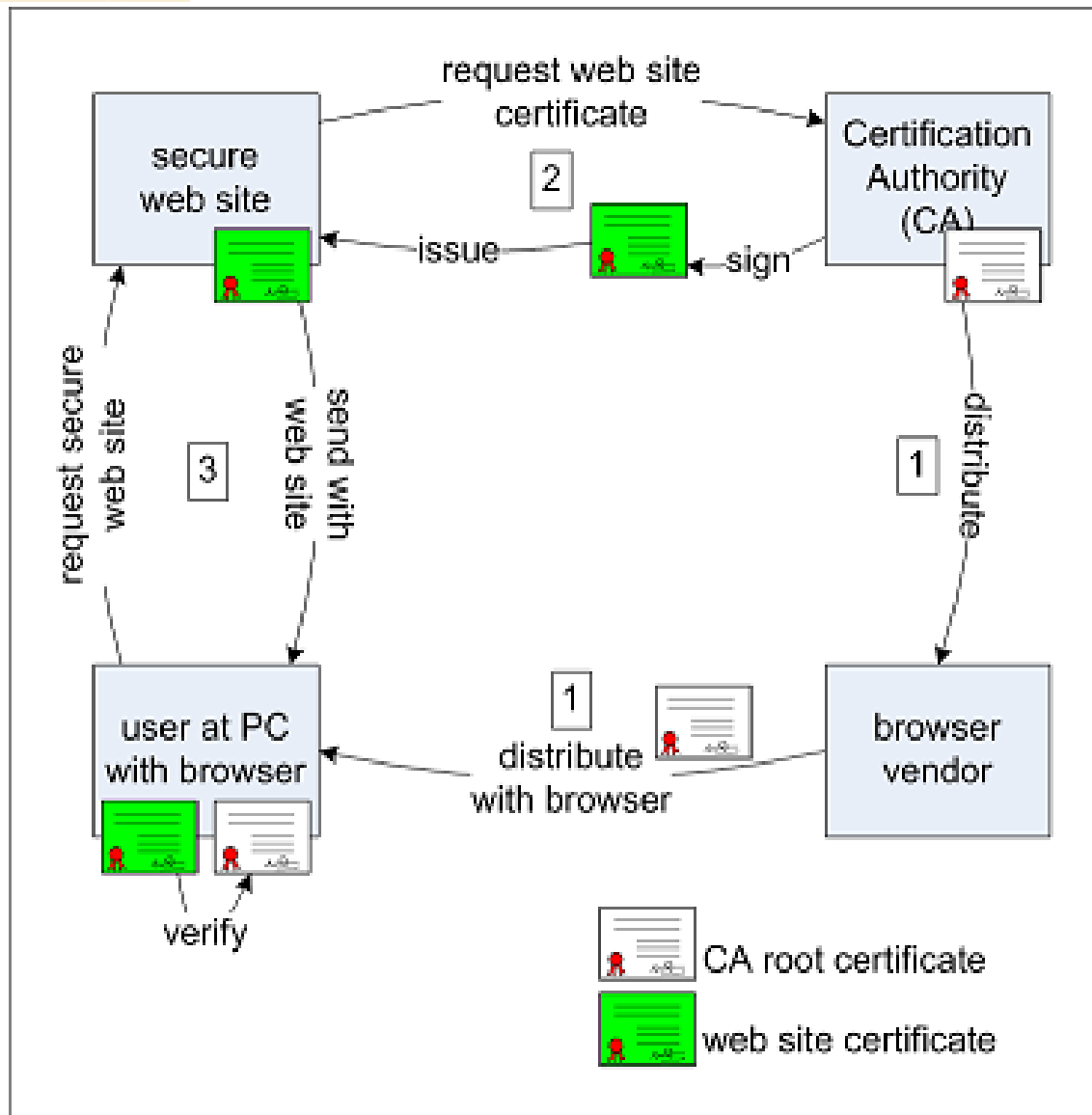
counterfeit certificate



- Flame was signed with a **fraudulent certificate** purportedly from the Microsoft Enforced Licensing Intermediate PCA certificate authority. [\[14\]](#)
- The malware authors identified a Microsoft Terminal Server Licensing Service certificate that still used the weak MD5 hashing algorithm,
- produced a counterfeit certificate that was used to sign malware to make them appear to have originated from Microsoft.
- A successful collision attack against a certificate was previously demonstrated in 2008, [\[15\]](#) but Flame implemented a **new variation of the chosen-prefix collision attack**



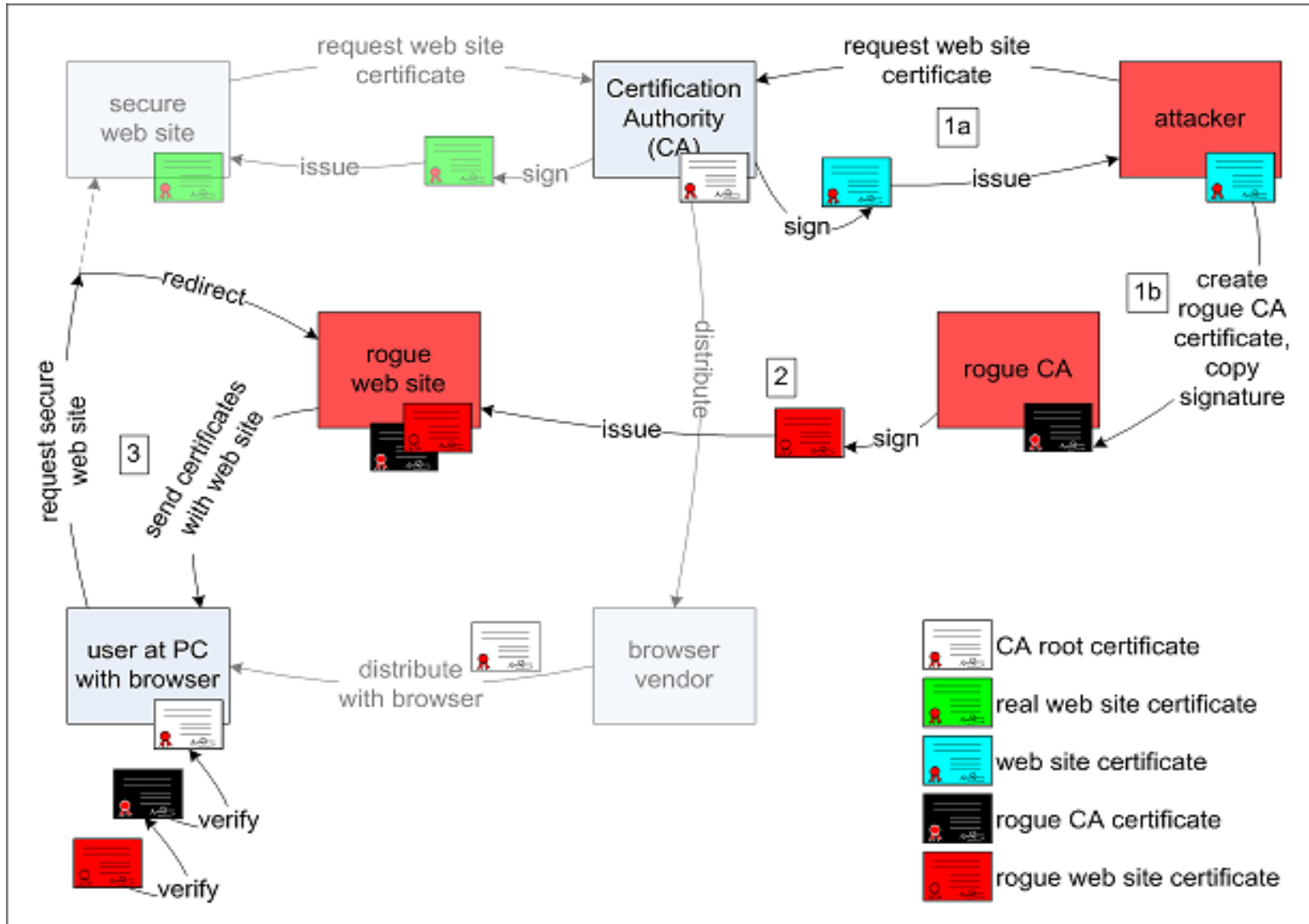
Normal use of certificates



证书验证

- **CA根证书**置于浏览器中
- **CA**签发网站证书
- 用户验证网站证书的根**CA**签名
- 普通网站证书不能用于签发下级证书

rouge certificates



伪CA证书与普通网站证书有相同的签名（hash 碰撞）

伪CA证书可签发新的伪证书（网站，软件）

伪证书可通过MS浏览器验证



问题出在哪里？



- Microsoft Terminal Server Licensing Service still used the MD5 hashing
- **MD5碰撞**--> **深入研究** --> 构造伪证书
- 用伪证书签发的软件（**Flame**）可通过**IE**验证
- **Fix: stop using MD5 (06-12)**
- "Microsoft releases Security Advisory 2718704". Microsoft. 3 June 2012.



Summary



- Kerberos
- PKI
 - Certificates
 - X.509
- Next – secure email
 - PGP
 - S/MIME