Web Security

PART I: PKI

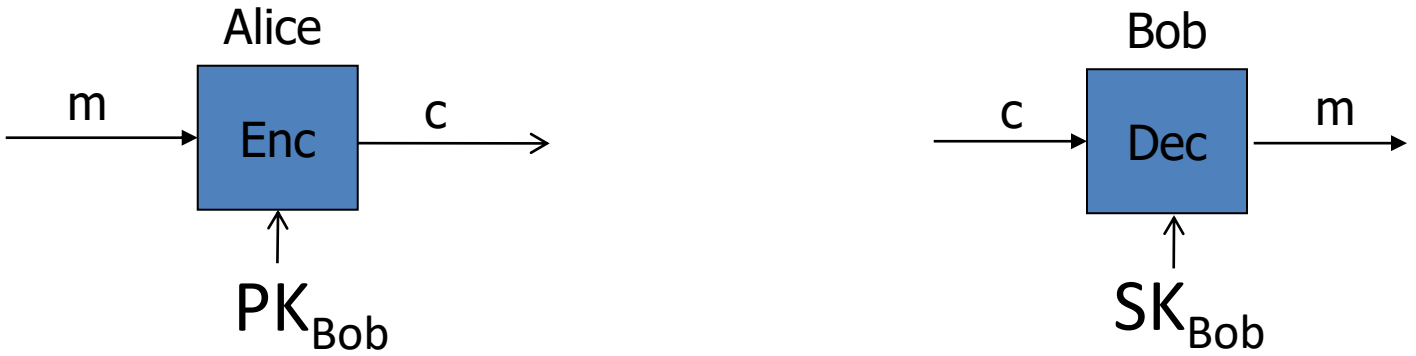
Dr. Bheemarjuna Reddy Tamma

IIT HYDERABAD

Note: This is a revised version of slide deck of Prof. Dan Boneh (Stanford) with material from various Internet sources

Secure Communication

Public-key encryption of messages:

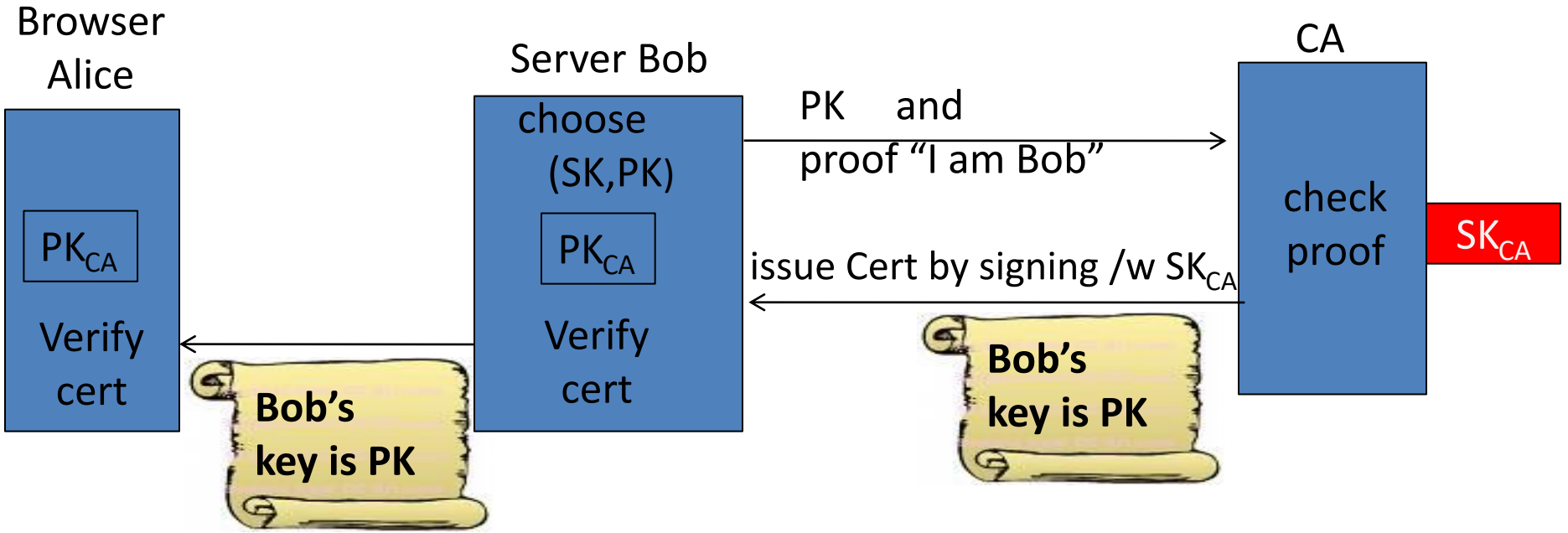


- Bob generates the key pair (SK_{Bob} , PK_{Bob})
- Alice: Using PK_{Bob} encrypts messages and only Bob can decrypt them
- Any attacks possible?

Certificates

How does Alice (browser) obtain PK_{Bob} ?

Alice & Bob rely on a trusted 3rd party in web PKI: CA



Bob gets Cert from CA attesting to his PK for an extended period

Digital Certificates

- A certificate is a signed data structure that binds a public key to an entity.
- Digital Certificates (also called X.509 certificates), as well as many other things in the X.509 standard, are described using Abstract Syntax Notation One (ASN.1)
- ASN.1 is a standard used to exchange information between systems independently of their encoding techniques
- Digital certificates are defined using ASN.1 and encoded using Distinguished Encoding Rules (DER)
- The signatureValue field in X.509 certificate contains a digital signature computed upon the ASN.1 DER encoded tbsCertificate structure in the certificate
  - tbs: tobesigned

Structure of X.509 Certificate

-- X.509 signed certificate

```
SignedContent ::= SEQUENCE
{
  tbsCertificate          CertificateToBeSigned,
  signatureAlgorithm      AlgoIdentifier,
  signatureValue          BITSTRING
}
```

https://cipherious.wordpress.com/2013/05/13/constructing-an-x-509-certificate-using-asn-1/

RFC 5280

https://docs.microsoft.com/en-us/windows/desktop/seccertenroll/about-x-509-public-key-certificates

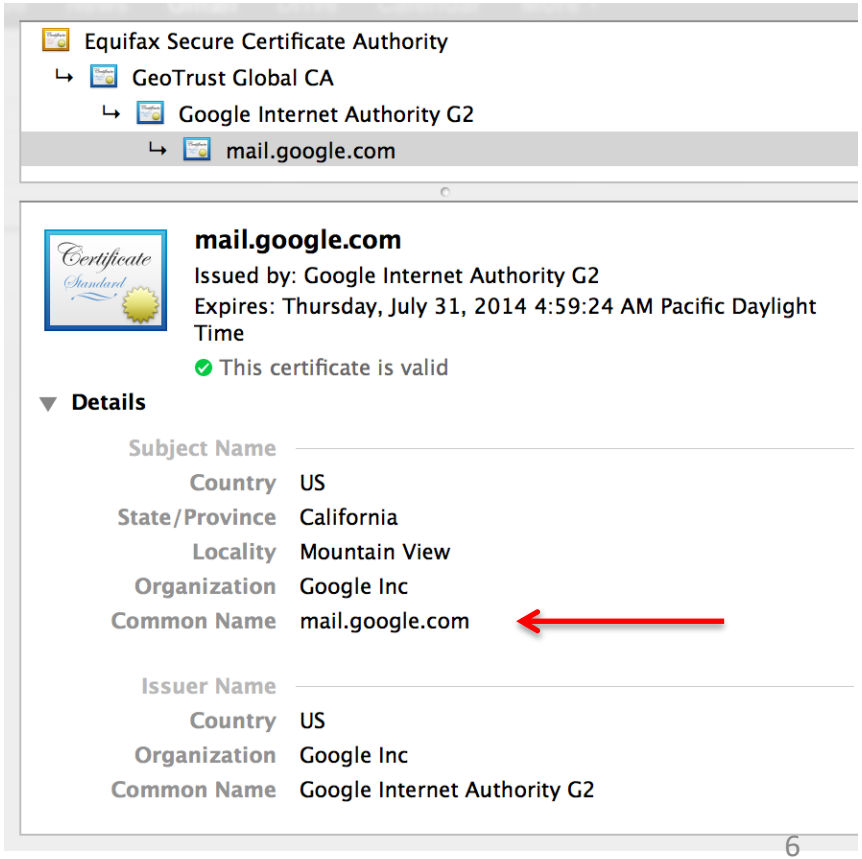
-- X.509 certificate to be signed

```
CertificateToBeSigned ::= SEQUENCE
{
  version                 [0] CertificateVersion DEFAULT v1,
  serialNumber            CertificateSerialNumber,
  signature               AlgorithmIdentifier,
  issuer                  Name
  validity                Validity,
  subject                 Name
  subjectPublicKeyInfo    SubjectPublicKeyInfo,
  issuerUniqueIdentifier  [1] IMPLICIT  UniqueIdentifier
OPTIONAL,
  subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier
OPTIONAL,
  extensions              [3] Extensions OPTIONAL
}
```

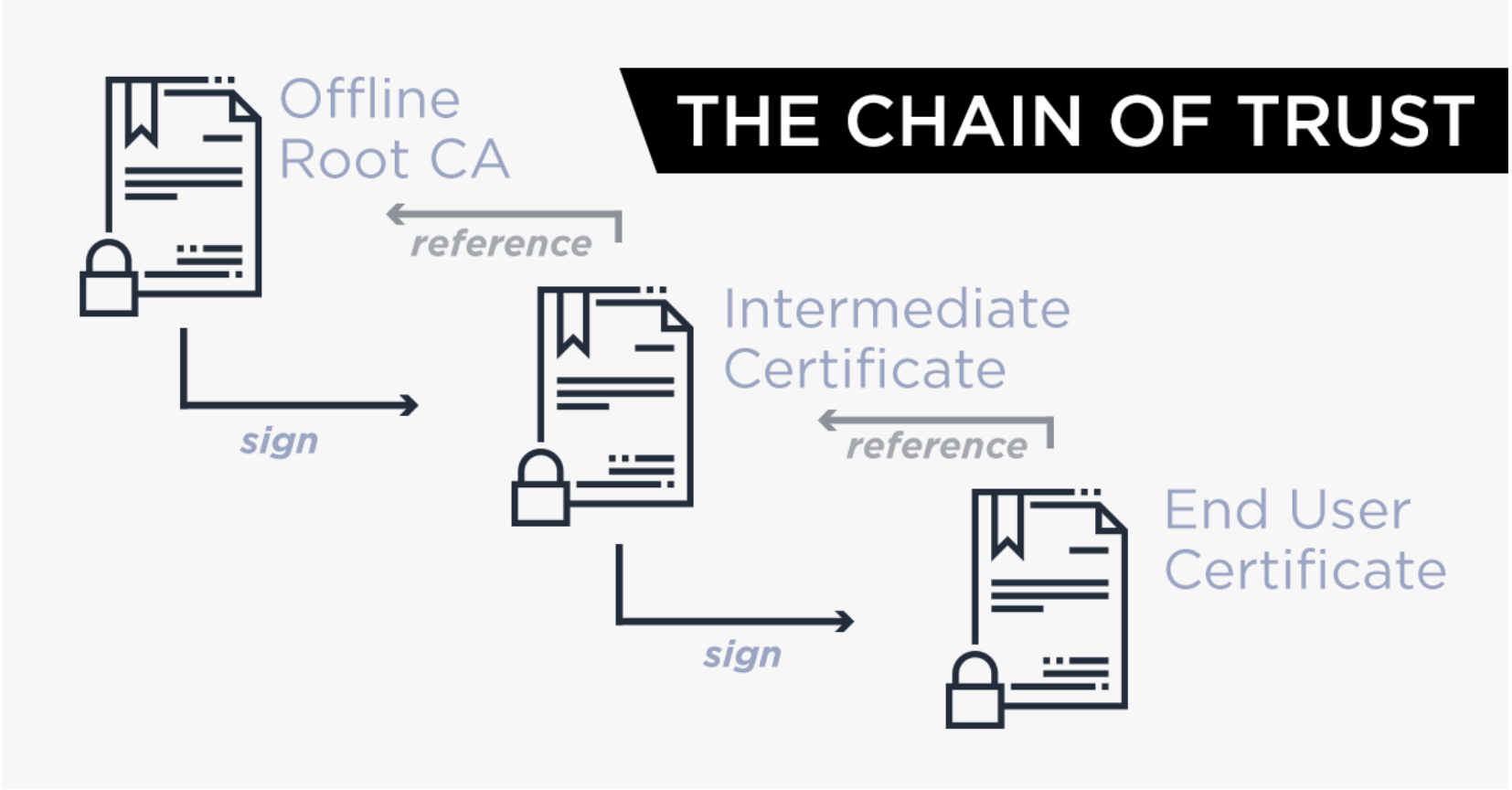
Certificates: example

Important fields:

Serial Number	5814744488373890497
Version	3
Signature Algorithm	SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)
Parameters	none
Not Valid Before	Wednesday, July 31, 2013 4:59:24 AM Pacific Daylight Time
Not Valid After	Thursday, July 31, 2014 4:59:24 AM Pacific Daylight Time
Public Key Info	
Algorithm	Elliptic Curve Public Key (1.2.840.10045.2.1)
Parameters	Elliptic Curve secp256r1 (1.2.840.10045.3.1.7)
Public Key	65 bytes : 04 71 6C DD E0 0A C9 76 …
Key Size	256 bits
Key Usage	Encrypt, Verify, Derive
Signature	256 bytes : 8A 38 FE D6 F5 E7 F6 59 …



Refer: https://www.sslsupportdesk.com/details-digital-certificate-mean/



- CAs are trusted signers of public keys
- Many browsers/OSes have root stores /w pre-installed certs of root and intermediate CAs
- Google operates several root and intermedia CAs: https://pki.goog/

chrome://settings/security

https://www.exoscale.com/syslog/securing-web-properties-through-https/

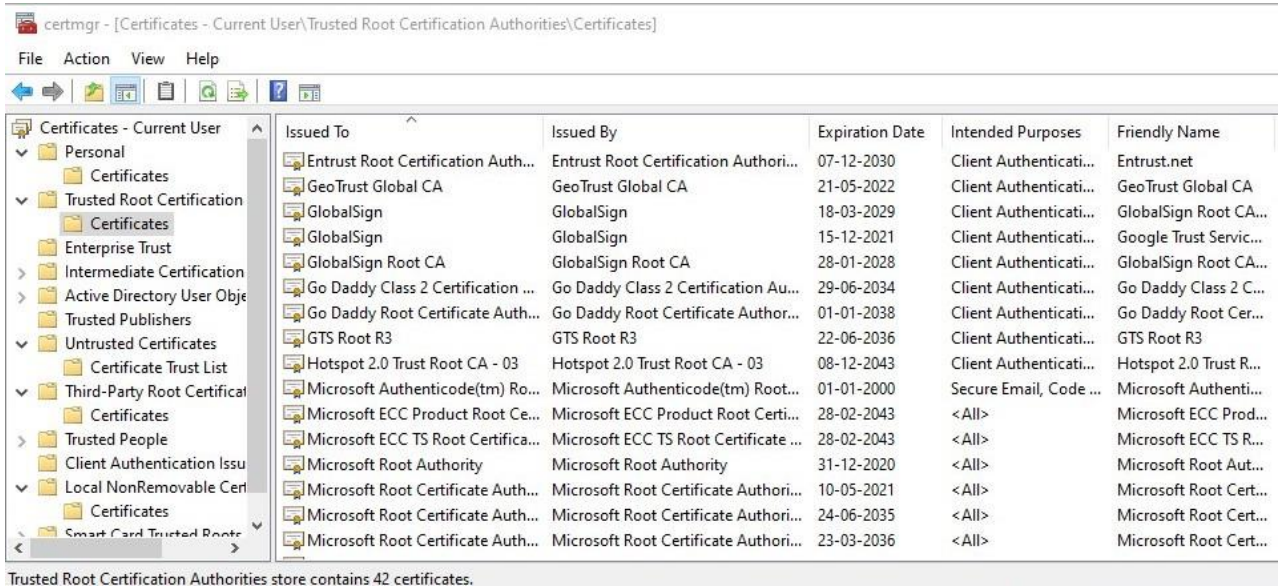
CAs and Root Stores

Browsers/Oses are pre-loaded with certs of several root and intermediate CAs

- GTS Root R3
- DigiCert
- GoDaddy

Root CAs ≈ 150

Intermediate CAs ≈ 3100



https://ccadb-public.secure.force.com/mozilla/IncludedCACertificateReport

TLS/SSL Certificates on the web

Subject's CommonName (CN) or SubjectAlternativeName (SAN) can be:

I. Single domain cert (explicit name) e.g. cse.iith.ac.in, iith.ac.in or

II. Wildcard cert, e.g. *.iith.ac.in or cse*.iith.ac.in or

III. Multidomain (SAN/UCC) cert allows 500 unique domains in a single cert

Matching rules:

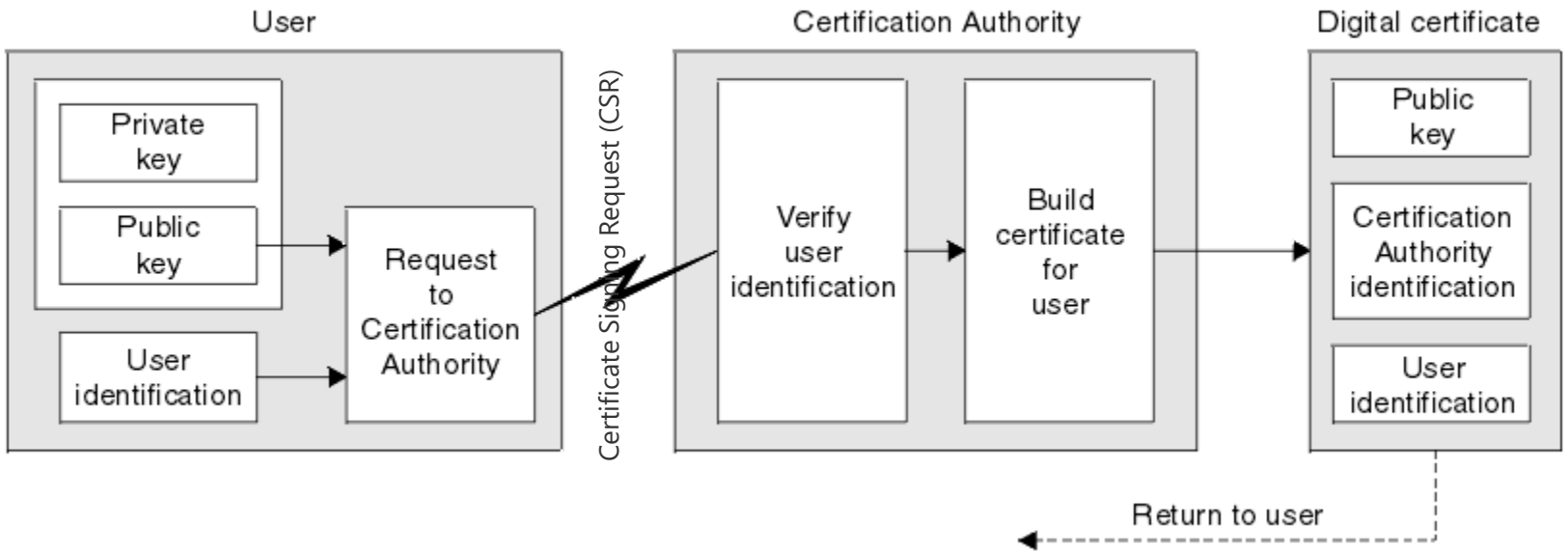
"*" must occur in leftmost component, does not match "."

example: *.a.com matches x.a.com but not y.x.a.com

(as in RFC 2818: "HTTPS over TLS")

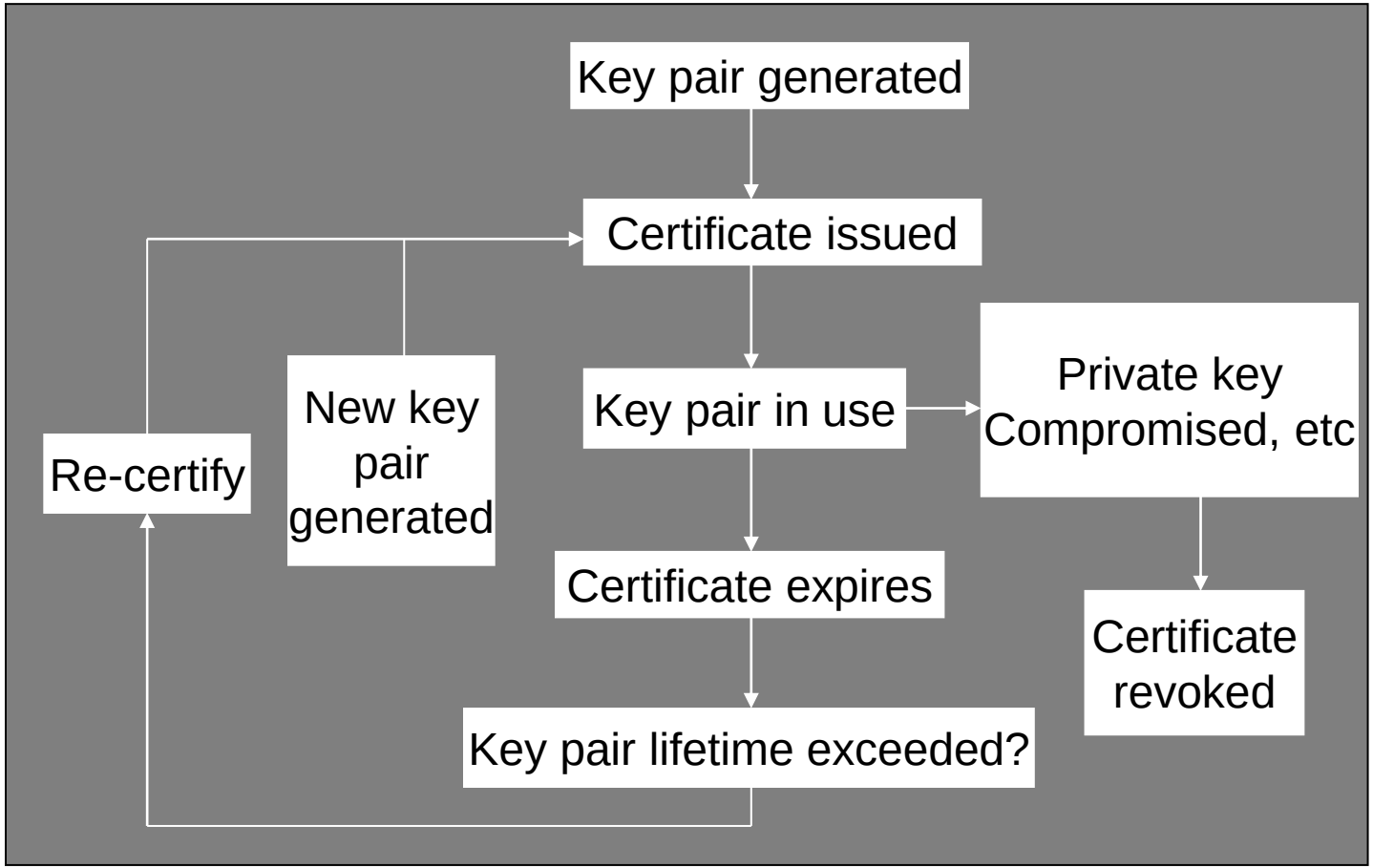
- For compatibility, primary FQDN (fully qualified domain name) in CN, and the full list of FQDNs in SAN

Obtaining Certificate from a CA



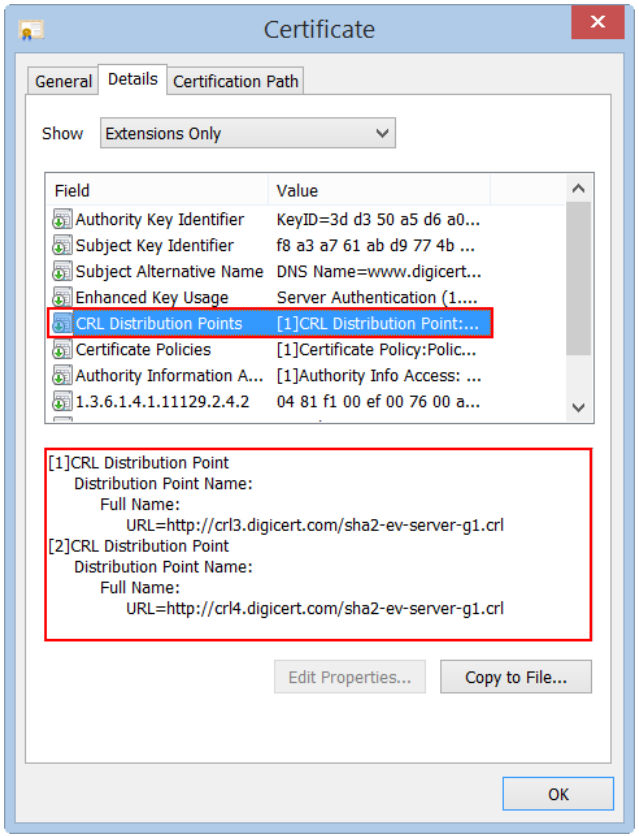
https://sites.google.com/site/ddmwsst/digital-certificates

Certificate Life Cycle



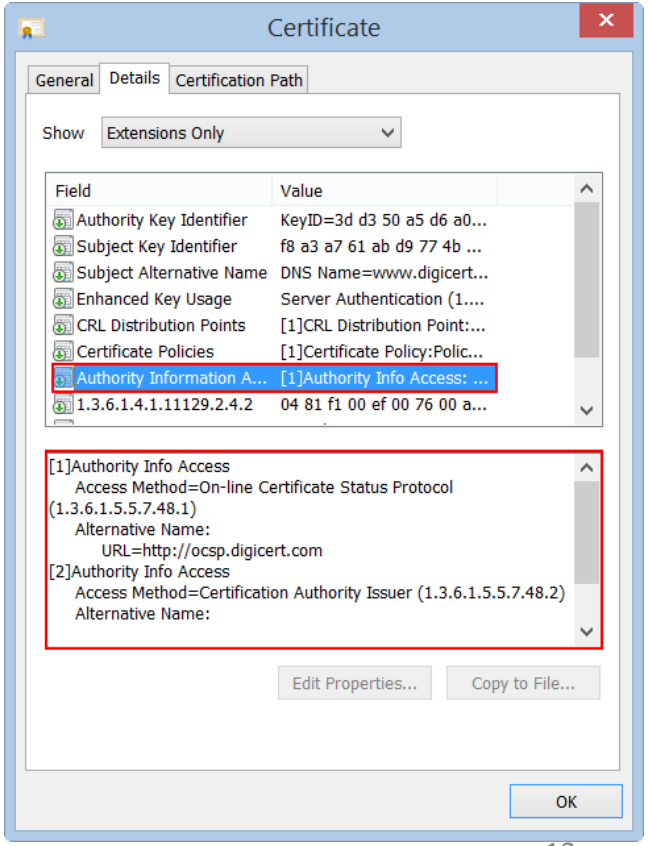
Certificate Revocation

- Two mechanisms: CRL and OCSP
- Certificate Revocation List (CRL)
  - CA periodically publishes/updates CRLs
  - Each revoked certificate is identified in a CRL by its serial number
  - CRL might be distributed by posting at known URL or from CA's own X.500 directory entry, specified in the certificate
  - Browsers have to download a large number of cert revocation info from the updated CRLs
  - What if CRL server is down?



Certificate Revocation

- OCSP (Online Certificate Status Protocol)
  - No need of downloading & searching
- Browser (Alice) queries CA's OCSP server about status of webserver's (Bob) cert before trusting it
- OCSP Stapling
  - Bob queries OCSP servers & caches it
  - Bob includes recent OCSP status when performing TLS/SSL handshake with Alice



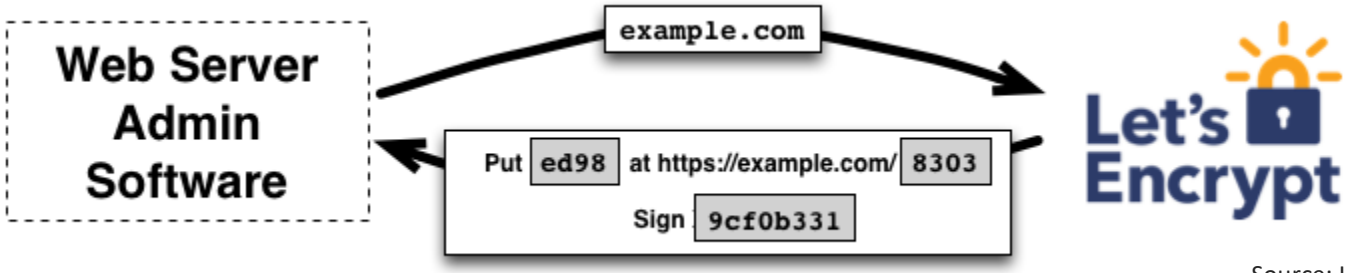
Let’s Encrypt & ACME protocol

- Let’s Encrypt is a free, automated, and open CA from nonprofit ISRG
- Certificates need to be requested, installed, and maintained, which is time-consuming
- So, Let’s Encrypt uses an open protocol to automate the deployment of certificates: ACME
- Automated Certificate Management Environment (ACME) implements automated interactions between CA and web servers, removing all the burden of getting and maintaining certificates
- Many tools based on ACME
  - Certbot: client to fetch, install, renew certificates
  - Caddy: Open source, webserver

https://www.exoscale.com/syslog/securing-web-properties-through-https/
https://letsencrypt.org/how-it-works/

Let’s Encrypt (LE)

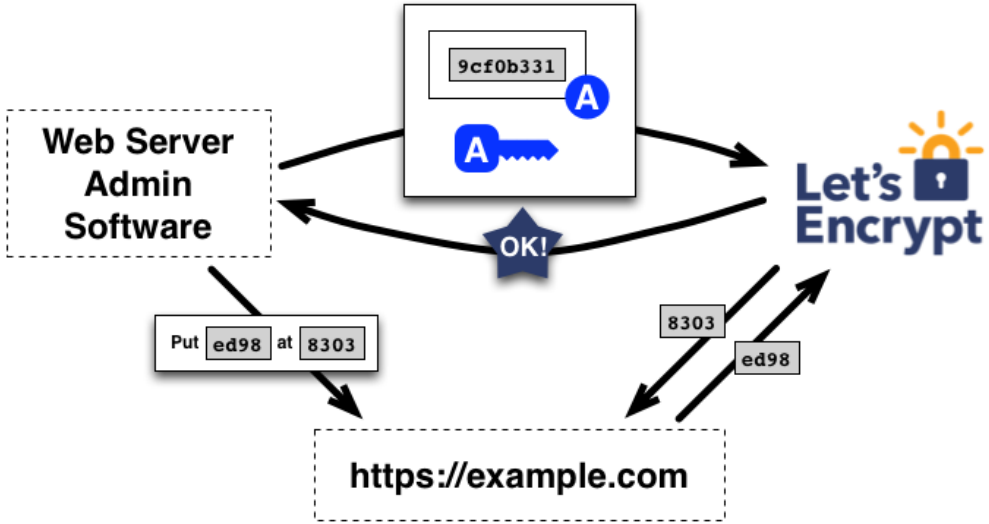
- ACME protocol automates requests, installation & maintenance of TLS/SSL certificates
- Agent s/w on webserver has to solve challenges posed by LE CA to prove ownership of the domain name
  - HTTP-01 challenge: a specified file in a specified location on a webserver accessible on port 80 and sign a token
  - TLS-SNI-01 challenge: a special temporary certificate on a webserver accessible on port 443
  - DNS-01 challenge: set up a specified DNS record



Source: Let’s Encrypt

Let’s Encrypt (LE): Domain Validation

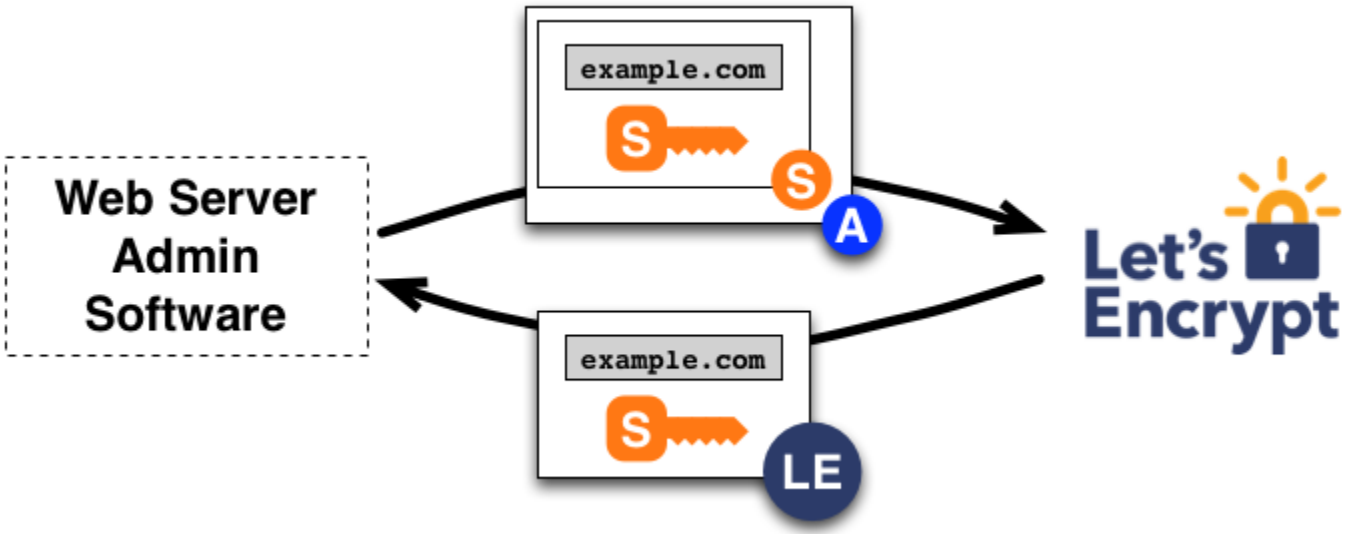
- Agent identified by the public key is authorized to do certificate management for example.com.
  - The key pair the agent used an “authorized key pair” for example.com



Source: Let’s Encrypt

Let’s Encrypt (LE): Cert Issuance

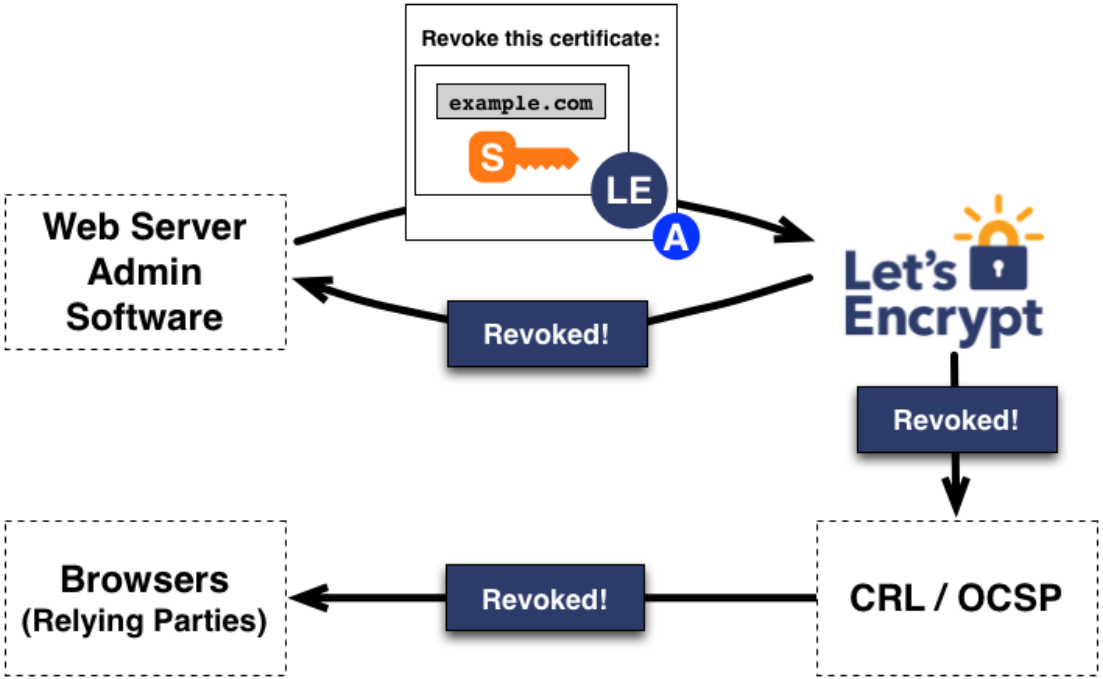
- Agent generates CSR signed /w private key of webserver which is in turn signed /w private key of Agent



Source: Let’s Encrypt

Let’s Encrypt (LE): Cert Revocation

- Agent generates cert revocation request signed /w private key of Agent



Source: Let’s Encrypt

Online tools to test SSL/TLS security

- https://www.digicert.com/help/
- https://www.thesslstore.com/ssltools/ssl-checker.php
  - Cert checker (Example)
  - Cas have become hacking targets
    - 2011: Comodo & Diginotar issued fraudulent certs for Hotmail, Gmail, Skype, Yahoo Mail, Firefox
    - 2013:TurkTrust issued cert for gmail
    - 2014: Indian NIC issued certs for Google and Yahoo!
  - Google stopped trusting Symantec, GeoTrust, Thawte, VeriSign, Equifax, and RapidSSL issued certs, prior to June 2016, by removing them from **Trusted Root Certification Authorities store on Chrome 66 browser**
    - Needed to replace /w DigiCert issued certificates
    - More details at Google's security blog

CCA: Root CA of India

Controller of Certifying Authorities
Ministry of Eletronics & Information Technology
Government of India

Home About CCA Services FAQ

PKI Framework | Root CA of India | Licensed CAs | Root Certificate | CA Certificates | eSign

Home » CA Services Overview

1. Safescrypt
2. IDRBT
3. (n)Code Solutions
4. e-Mudhra
5. CDAC
6. Capricorn
7. NSDL e-Gov
8. Vsign (Verasys)
9. Indian Air Force
10. CSC
11. RISL (RajComp)
12. Indian Army
13. IDSign
14. CDSL Ventures
15. Panta Sign

CA Services Overview

Overview of Services offered by licensed CAs

Licensed CAs	Class 1 -3 DSCs	eSign	SSL and Code Signing Certificates	Time Stamping
Safescrypt	✓	✓		✓
IDRBT	✓ Only to Banks		✓ Only to Banks	✓ Only to Banks
(n)Code Solutions	✓	✓	✓*	✓
e-Mudhra	✓	✓	✓*	✓
CDAC		✓		
Capricom	✓	✓		✓
NSDL e-Gov		✓		
Vsign (Verasys)	✓	✓		
Indian Air Force	✓ Only to IAF			✓ Only to IAF
CSC		✓		
RISL (RajComp)	✓	✓	✓	✓
Indian Army	✓ Only to Army		✓ Only to Army	✓ Only to Army
IDSign	✓	✓		✓
CDSL Ventures		✓		
Pantasign	✓	✓		

* The Root CA Certificate of India is listed only in Microsoft products (Including IE)

Closed CAs

Licensed CAs	Class 1 -3 DSCs	SSL and Code Signing Certificates
MTNL	NA	NA
iCERT	NA	NA
TCS	NA	NA
NIC	NA	NA

http://cca.gov.in/CAServicesOverview.html

Summary

- Public key crypto is a powerful tool
  - Underlies https, ssh, virtually all software updates, etc
  - But does n't solve the key distribution problem
- Certificate authorities (CA) occupy key (and trusted) role
  - 3rd party attestation of identity or access
  - Public, private and open CAs
    - Let's Encrpt made them affordable to all
  - Other uses of certificates: eSign, code signing, timestamping, etc
  - Ongoing efforts to police CAs

References

- Public Key Infrastructure| Microsoft Docs
- X.509 std: RFC 5280 and ACME https://tools.ietf.org/html/rfc8555
- X.509 debugger: http://phpseclib.sourceforge.net/x509/decoder.php
- ASN.1 parsers
  - http://lapo.it/asn1js/#
  - http://phpseclib.sourceforge.net/x509/asn1parse.php
- https://aka.ms/RootCert
- https://support.apple.com/en-us/HT208125
- https://pki.goog/
- Basics of Digital Certificates and Certificate Authority - Web Service Security Tutorial
- https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html
- https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html
- https://www.thesslstore.com/blog/difference-sha-1-sha-2-sha-256-hash-algorithms/
- https://cipherious.wordpress.com/2013/05/13/constructing-an-x-509-certificate-using-asn-1/
- https://www.sslsupportdesk.com/category/ssl-library/
- https://letsencrypt.org/how-it-works/