



Web Security

PART I: PKI

2: TLS

3: HTTPS ✓

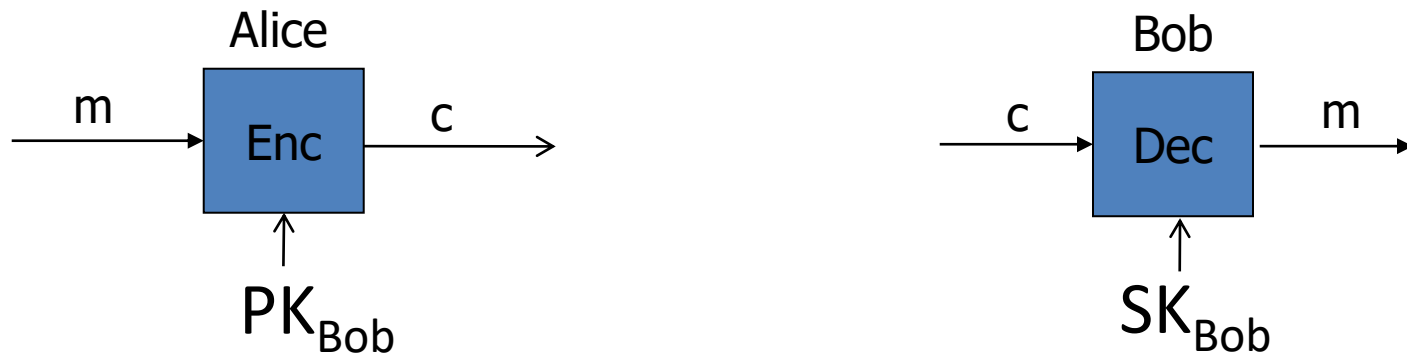
Dr. Bheemarjuna Reddy Tamma

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:

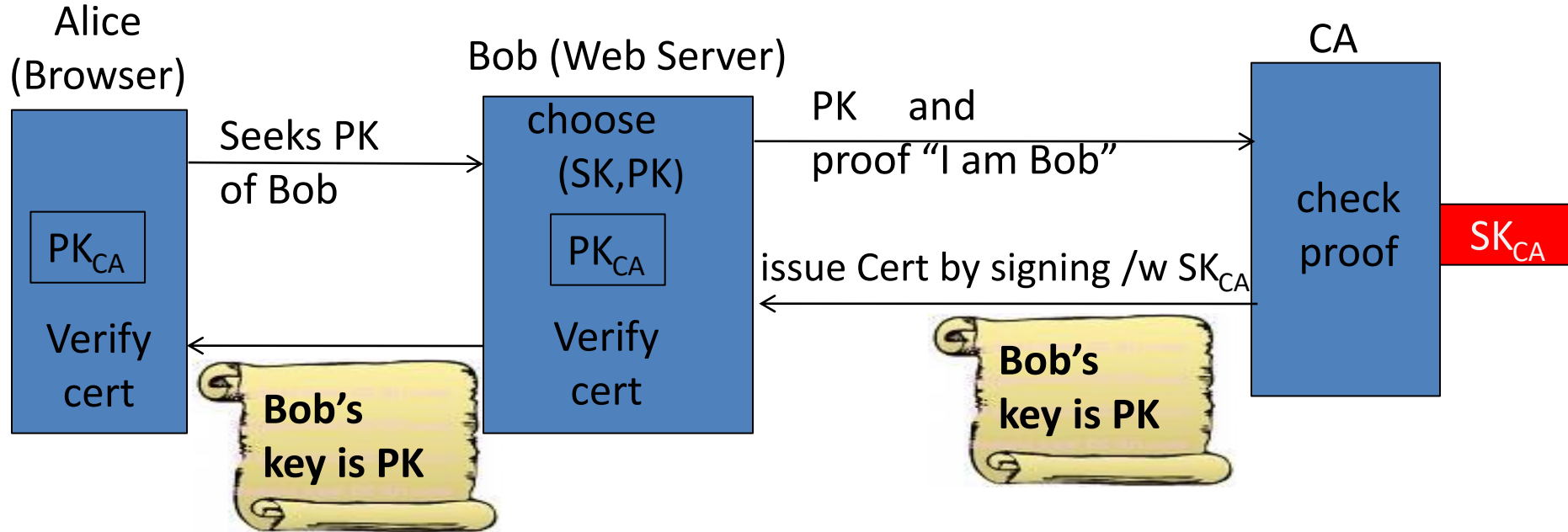


- Bob generates the key pair (SK_{Bob} , PK_{Bob})
- Using PK_{Bob} Alice encrypts messages & only Bob can decrypt them
- Any attacks possible?

Certificates: Signed Public Keys

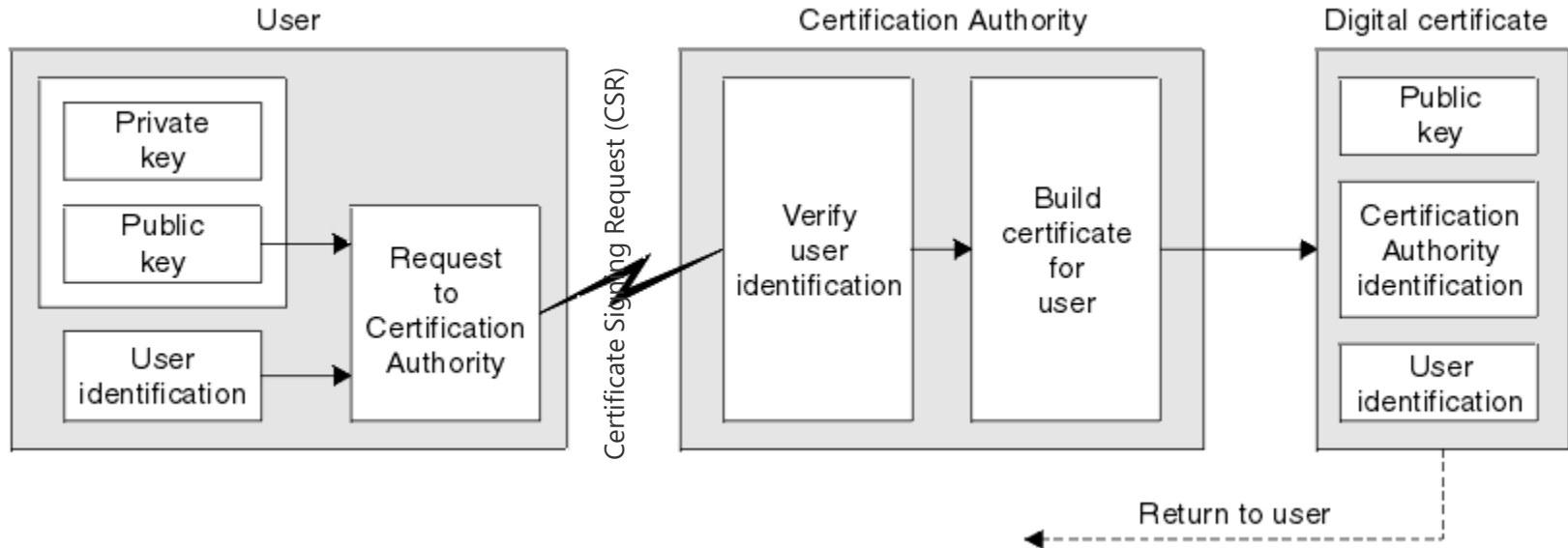
How does Alice (browser) obtain PK_{Bob} ?

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



Bob gets Cert from CA attesting to his PK for an extended period₃

Obtaining Certificate from a CA



Applications

- TLS certificates – for authenticated and encrypted browsing
- S/MIME certificates – for signing and encrypting emails
- Code signing certificates
- Document signing certificates
- Client authentication certificates
- Government-issued electronic IDs

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) or PEM
- The signatureValue field in X.509 certificate contains a digital signature computed upon the ASN.1 DER encoded tbsCertificate structure in the certificate
 - tbs: **to**bsigned

Structure of Digital Certificate

-- X.509 signed certificate

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

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

[RFC 5280](#)

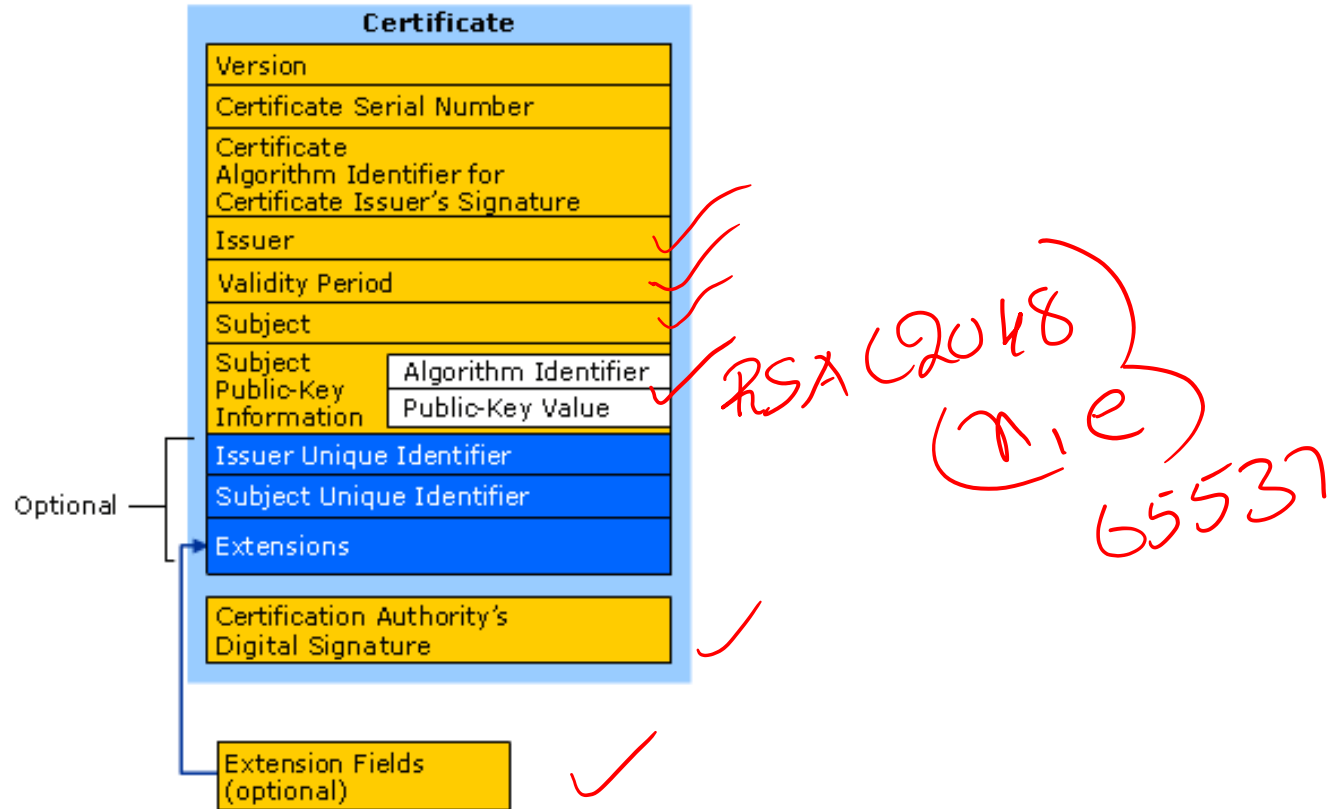
<https://docs.microsoft.com/en-us/windows/desktop/seccertenroll/about-x-509-public-key-certificates>

-- X.509 certificate to be signed

ASN-1

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

Structure of Digital Certificate



X.509 Certificates: An 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 ...	←

Equifax Secure Certificate Authority

↳ GeoTrust Global CA

↳ Google Internet Authority G2

↳ mail.google.com

 **mail.google.com**

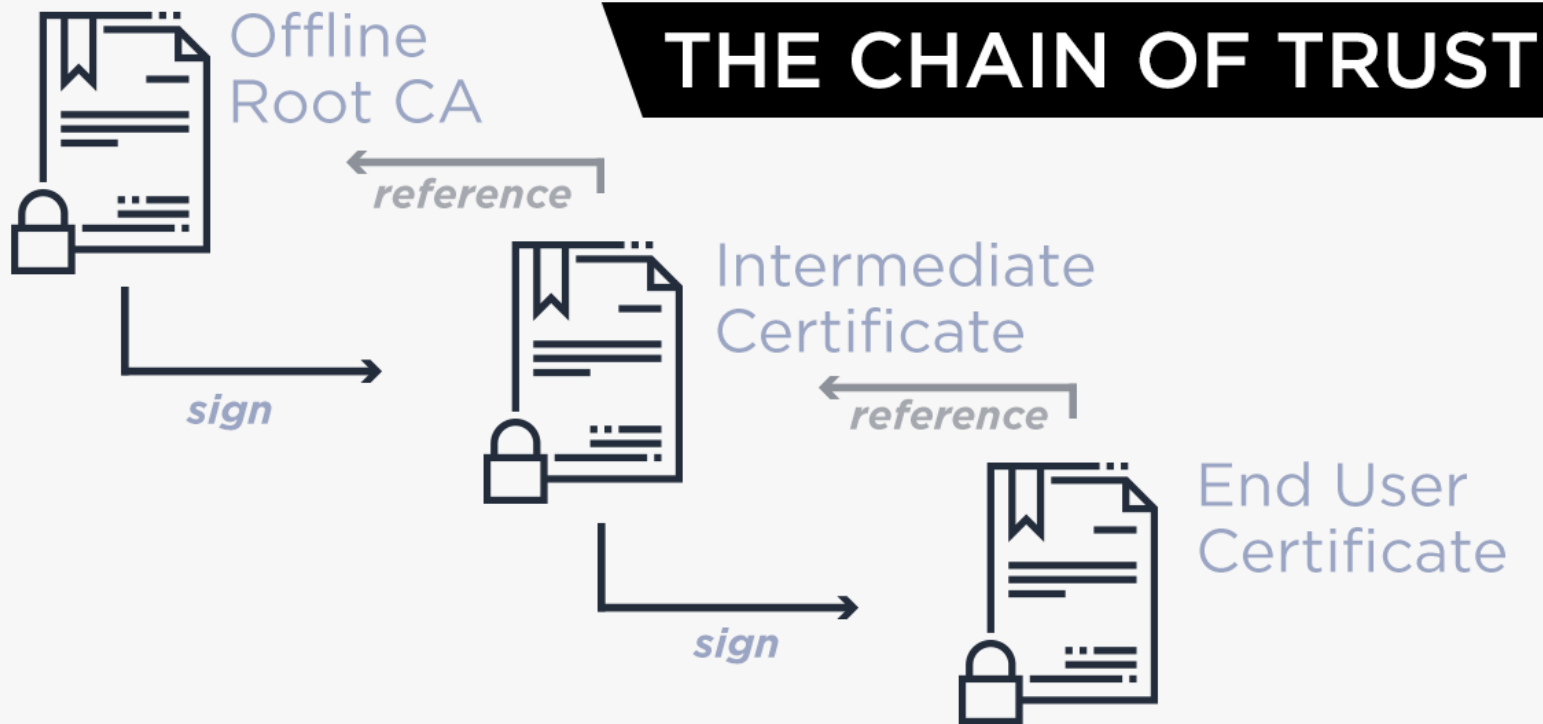
Issued by: Google Internet Authority G2

Expires: Thursday, July 31, 2014 4:59:24 AM Pacific Daylight Time

✓ This certificate is valid

▼ **Details**

Subject Name	
Country	US
State/Province	California
Locality	Mountain View
Organization	Google Inc
Common Name	mail.google.com ←
Issuer Name	
Country	US
Organization	Google Inc
Common Name	Google Internet Authority G2



- CAs are trusted signers of public keys in web PKI
- 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/>

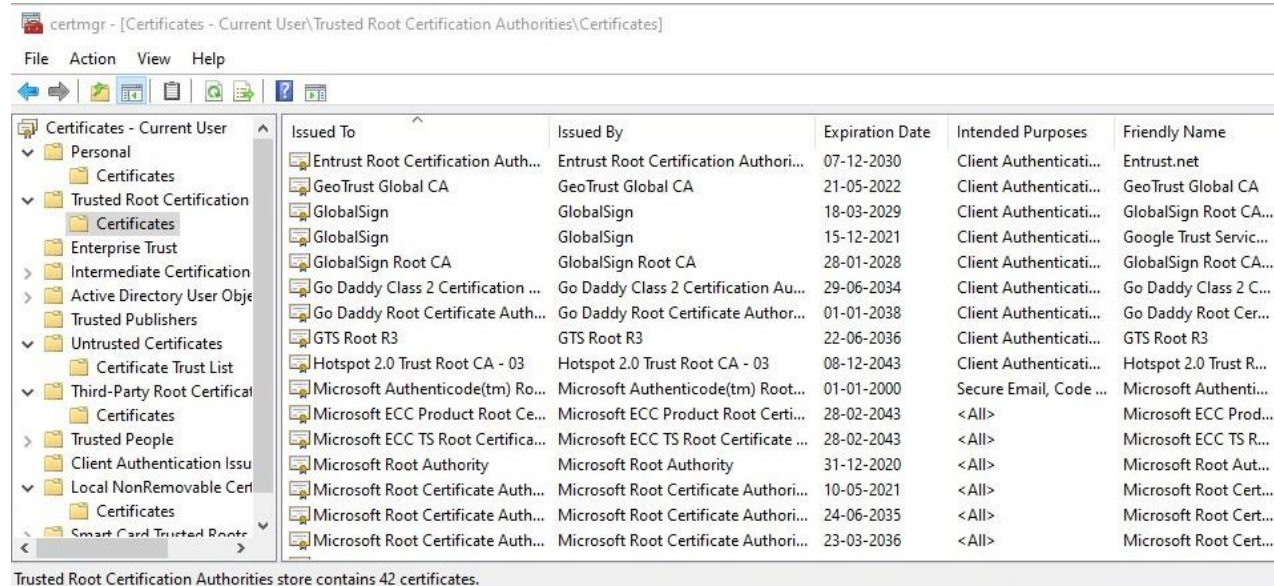
CAs and Root Stores

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

- GTS Root R3
- DigiCert
- GoDaddy

Root CAs \approx 165

Intermediate CAs \approx 3100



Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name
Entrust Root Certification Auth...	Entrust Root Certification Authori...	07-12-2030	Client Authenticati...	Entrust.net
GeoTrust Global CA	GeoTrust Global CA	21-05-2022	Client Authenticati...	GeoTrust Global CA
GlobalSign	GlobalSign	18-03-2029	Client Authenticati...	GlobalSign Root CA...
GlobalSign	GlobalSign	15-12-2021	Client Authenticati...	Google Trust Servic...
GlobalSign Root CA	GlobalSign Root CA	28-01-2028	Client Authenticati...	GlobalSign Root CA...
Go Daddy Class 2 Certification ...	Go Daddy Class 2 Certification Au...	29-06-2034	Client Authenticati...	Go Daddy Class 2 C...
Go Daddy Root Certificate Auth...	Go Daddy Root Certificate Author...	01-01-2038	Client Authenticati...	Go Daddy Root Cer...
GTS Root R3	GTS Root R3	22-06-2036	Client Authenticati...	GTS Root R3
Hotspot 2.0 Trust Root CA - 03	Hotspot 2.0 Trust Root CA - 03	08-12-2043	Client Authenticati...	Hotspot 2.0 Trust R...
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...	01-01-2000	Secure Email, Code ...	Microsoft Authenti...
Microsoft ECC Product Root Ce...	Microsoft ECC Product Root Certi...	28-02-2043	<All>	Microsoft ECC Prod...
Microsoft ECC TS Root Certifica...	Microsoft ECC TS Root Certificate ...	28-02-2043	<All>	Microsoft ECC TS R...
Microsoft Root Authority	Microsoft Root Authority	31-12-2020	<All>	Microsoft Root Aut...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	10-05-2021	<All>	Microsoft Root Cert...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	24-06-2035	<All>	Microsoft Root Cert...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	23-03-2036	<All>	Microsoft Root Cert...

Trusted Root Certification Authorities store contains 42 certificates.



<https://cabforum.org/>

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

TLS/SSL Certificate Types

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

Matching rules: “*” must occur in leftmost component, does not match “.”
example: *.a.com matches x.a.com but not y.x.a.com

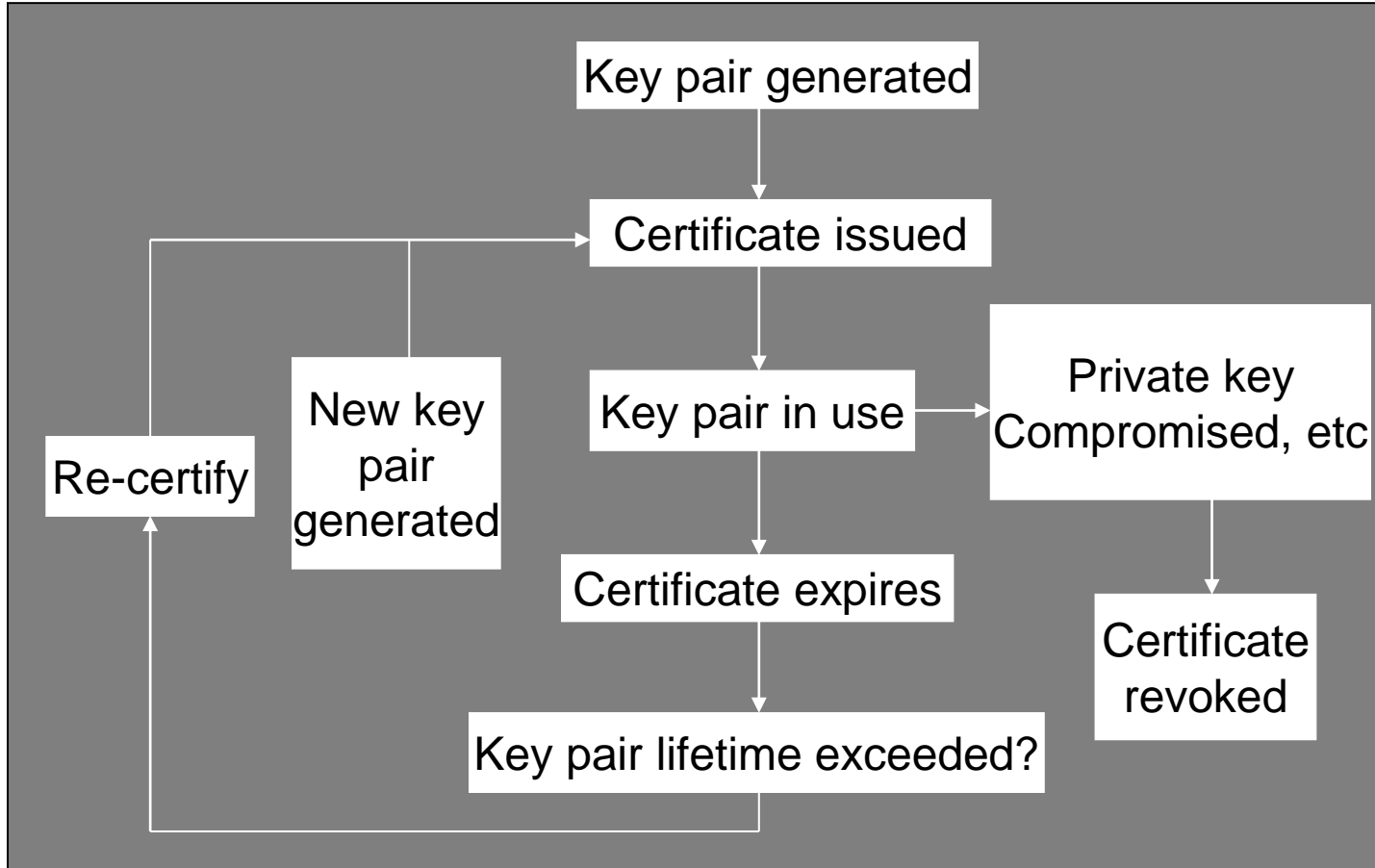
- III. Multi-Domain (SAN/UCC) cert allows 250-500 unique domains in a single cert
 - For compatibility, primary FQDN (fully qualified domain name) in CN, and the full list of FQDNs in SAN E.g., <https://www.nsa.gov/>
- IV. Multi-Domain Wildcard Certificates
 - FQDNs/wildcard domains in SAN list E.g., [X.509 - Wikipedia](#)

TLS/SSL Certificate Validation Levels

Based on thoroughness in validation process by the CA:

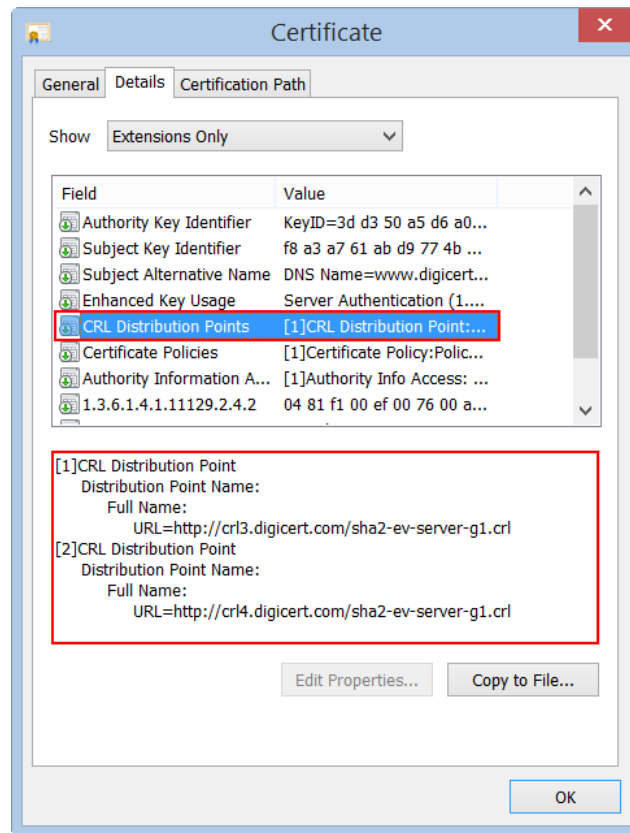
- I. DV (Domain validation) Cert
 - Prove that you own the domain
- II. OV (Organization Validation) Cert
 - A manual vetting process by CA, so expensive
 - Enterprises
- III. EV (Extended Validation Cert)
 - Full background check, very expensive
 - banks/ecommerce

Certificate Life Cycle



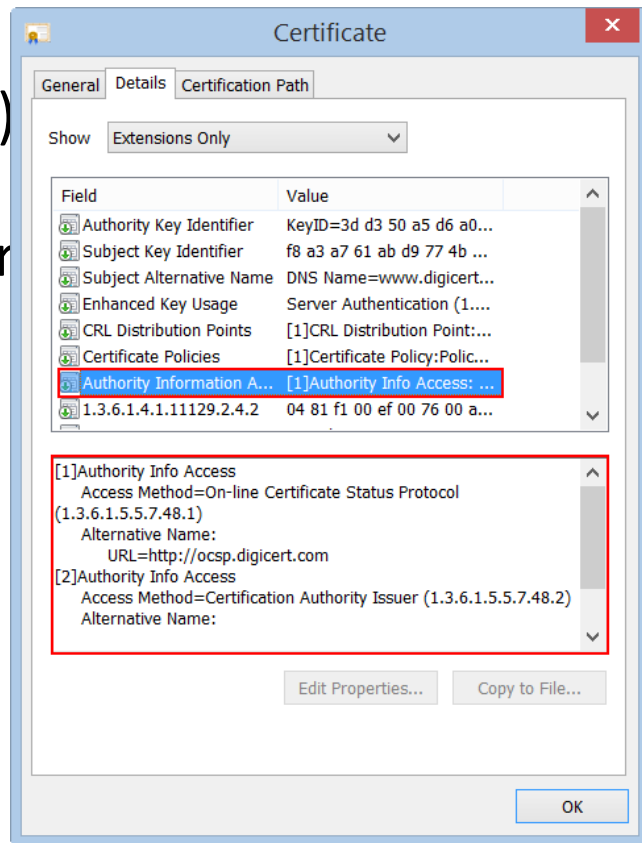
Certificate Revocation

- Two mechanisms: CRL and OCSP
- Certificate Revocation List (CRL)
 - CA periodically publishes/updates CRLs which are time-stamped & signed
 - 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 amount 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 handshake with Alice
 - Avoids delay due to OCSP TLS negotiation with CA by piggybacking in Bob's TLS pipe
 - Addresses privacy issues with CA



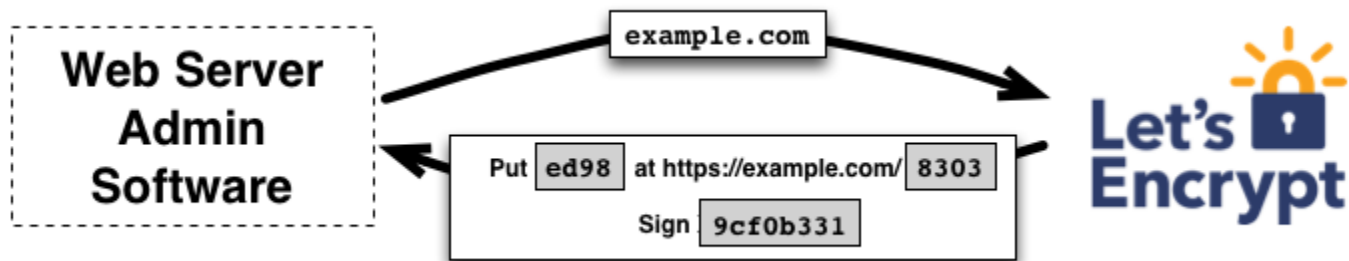
Let's Encrypt & ACME protocol

- Let's Encrypt is a free, automated, and open CA from non-profit ISRG (Internet Security Research Group)
- Certificates need to be requested, installed, and maintained frequently, which is time-consuming & boring!
- 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/>

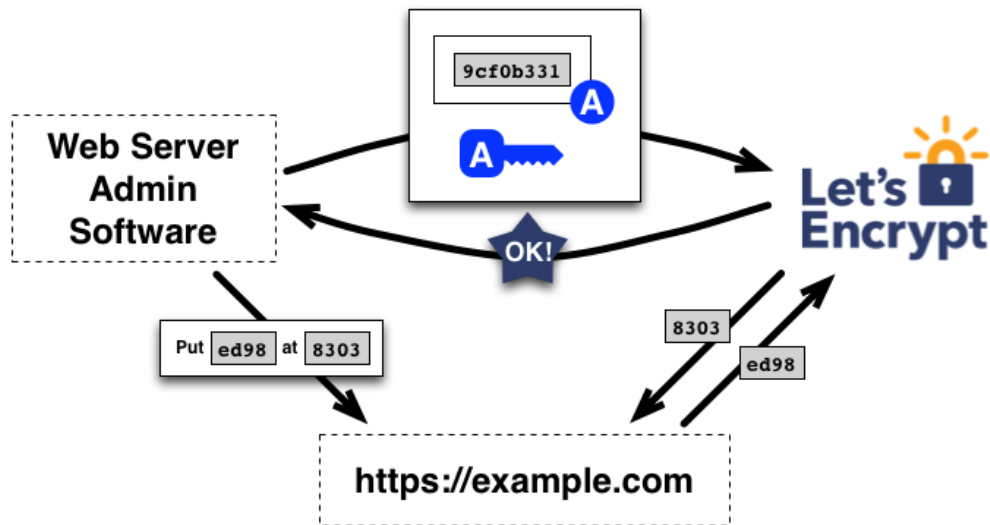
Let's Encrypt (LE)

- 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



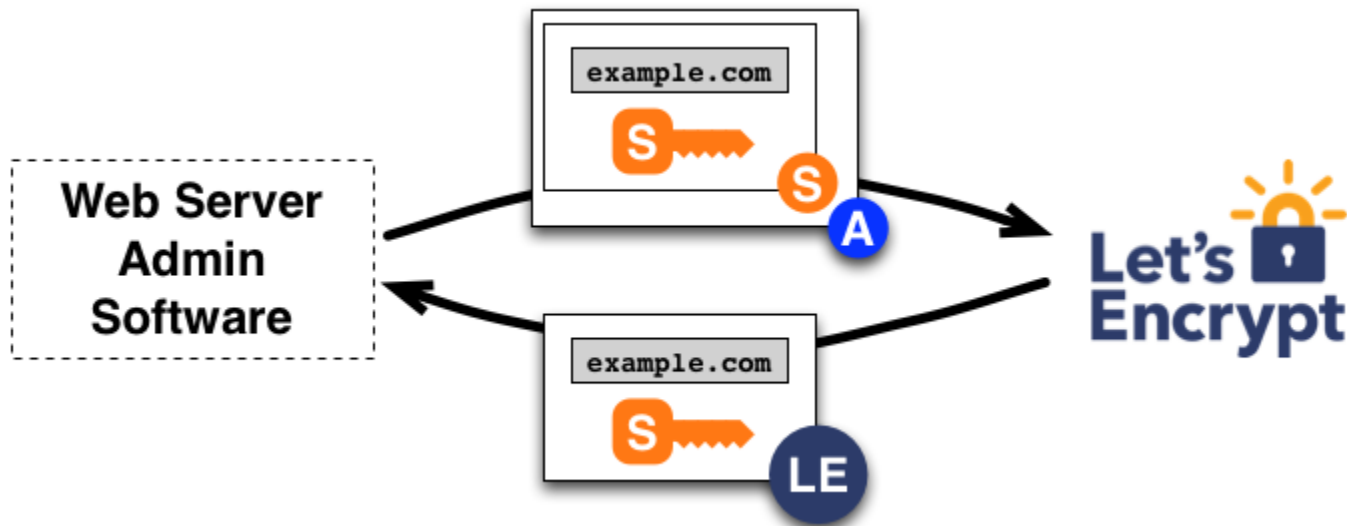
Let's Encrypt (LE): Domain Validation

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



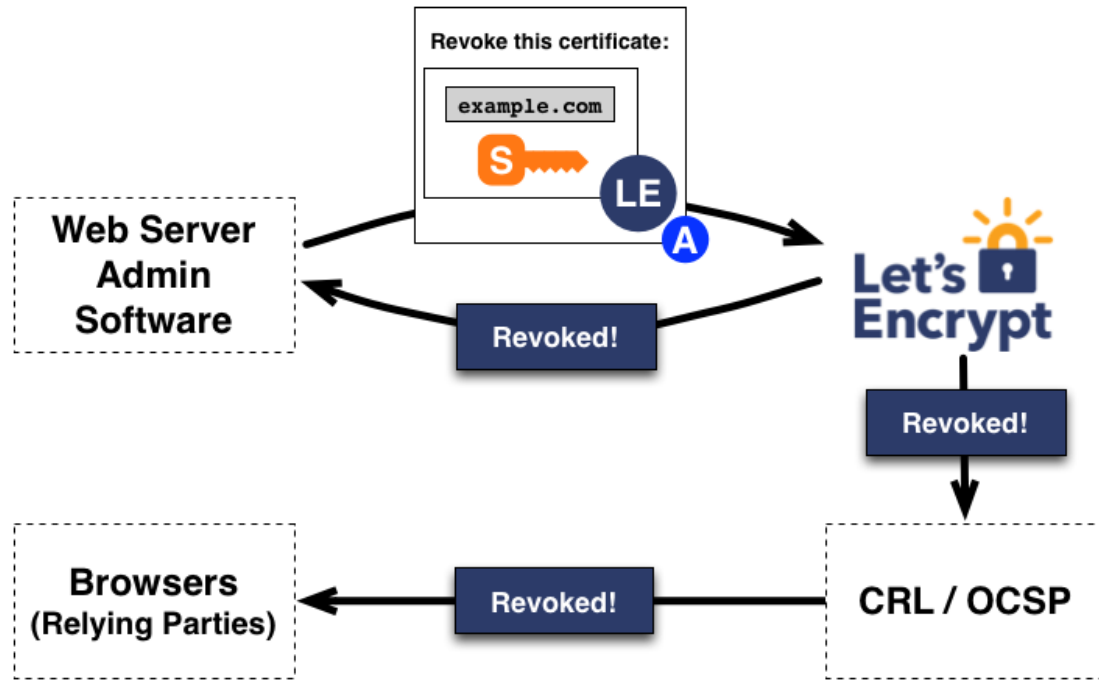
Let's Encrypt (LE): Cert Issuance

- Agent generates CSR signed /w private key (S) of webserver which is in turn signed /w private key of Agent (A)



Let's Encrypt (LE): Cert Revocation

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



Online tools to test SSL/TLS security

- <https://www.digicert.com/help/>
- <https://www.thesslstore.com/ssltools/ssl-checker.php>
 - CAs have become hacking targets too☹
 - 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

PKI Framework

Root CA of India

Licensed CAs

Root Certificate

CA Certificates

eSign

Home » CA Services Overview



Overview of Services offered by licensed CAs

Licensed CAs	Class 1-3 DSCs	eSign	SSL and Code Signing Certificates	Time Stamping
Safescript	✓	✓		✓
IDRBT	✓ Only to Banks		✓ Only to Banks	✓ Only to Banks
(n)Code Solutions	✓	✓	✓*	✓
e-Mudhra	✓	✓	✓*	✓
CDAC		✓		
Capricorn	✓	✓		✓
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		✓		
Pantesign	✓	✓		

* 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

1. Safescript
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

<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 a very key (and trusted) role
 - 3rd party attestation of identity or access
 - Public, private and open CAs
 - Let's Encrypt made them affordable to all
 - Other uses of certificates: digital signs, 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: <https://phpseclib.sourceforge.net/x509/decoder.php>
- ASN.1 parsers
 - <http://lapo.it/asn1js/#>
 - <https://phpseclib.sourceforge.net/x509/asn1parse.php>
- <https://aka.ms/RootCert>
- <https://support.apple.com/en-in/HT209143>
- <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/> & <https://letsencrypt.org/certificates/>