Managing Enterprise Cybersecurity
MIS 4596

Unit #11

Agenda

- Public Key Infrastructure
- Digital Certificate
- Public key Certificates
- Roles in PKI: Certificate Authority (CA)
- Roles in PKI: Registration Authority (RA)
- PKI Steps
- Chain of Trust
- Root Programs
- Certificate Revocation List (CRL)
- PKI Roles / Workflows…

Public Key Infrastructure (PKI)

Public key cryptography enables entities previously unknown to each other to verify the identity of each other, validate the information being transferred, and securely communicate on an insecure public network

- **Public key infrastructure**
  - Enables online activities requiring more trust and proof of identity than simple passwords
  - Provides a hierarchy of trust relationships that:
    - Enable knowing a public key really belongs to the person/system you want to communicate with
    - Are necessary for hybrid cryptography
    - Facilitate secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email

Public Key Infrastructure (PKI)

Is a system for creating, storing, distributing, validating, revoking and managing **digital certificates** used to verify the identity the owner of a public key contained within the certificate

- Assumes
  - Receiver's and Sender's identities can be positively ensured through digital certificates
  - Asymmetric algorithm will automatically carry out the process of key exchange

- Contains components that
  - Identify users
  - Creates and distributes certificates
  - Maintains and revokes certificates
  - Distributes and maintains encryption keys
  - Enables information technologies to communicate and work together to achieve confidentiality, authentication, integrity, and non-repudiation
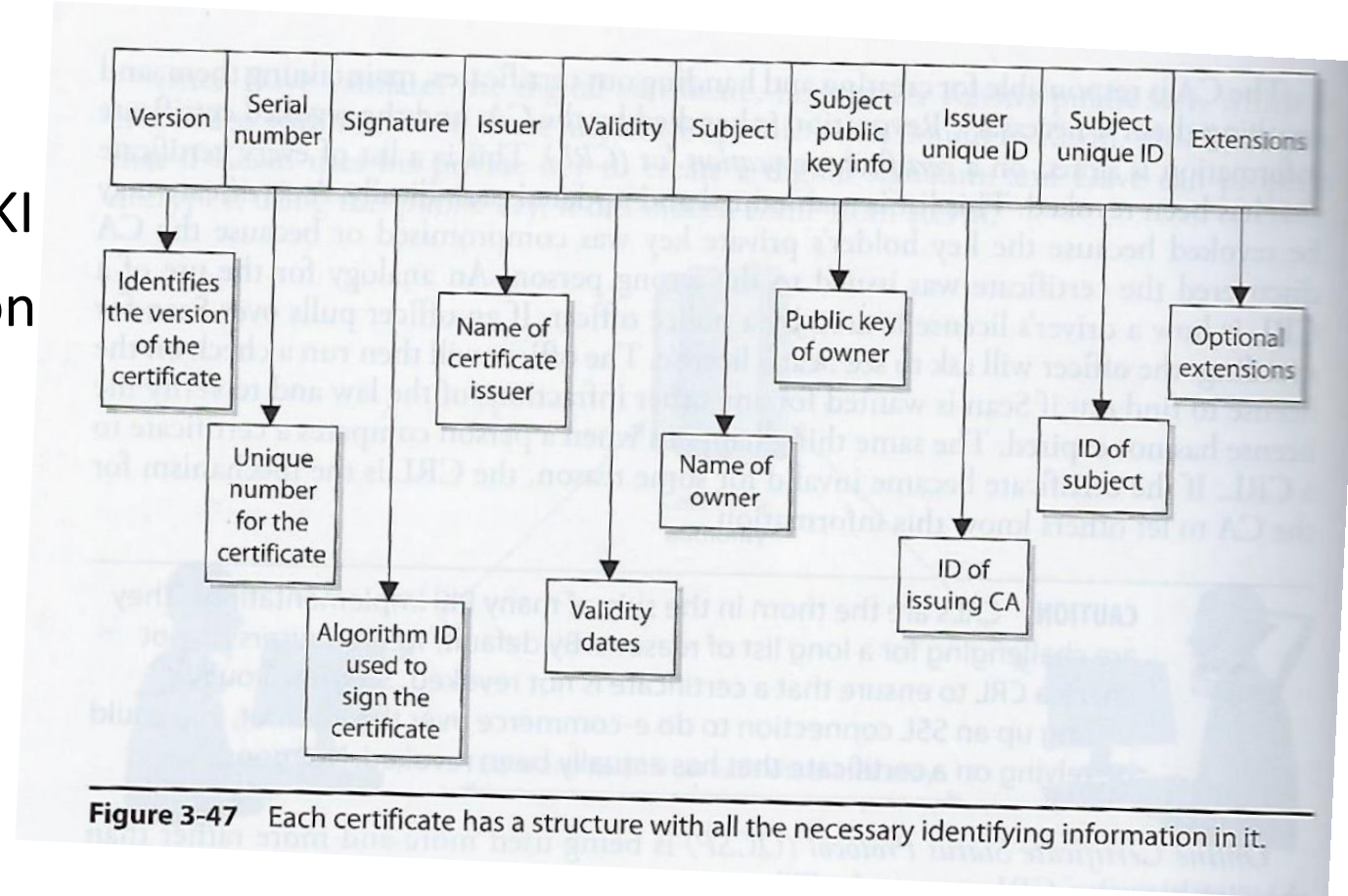
Public Key Infrastructure (PKI)

Consists of:

- **Public key certificates ("digital certificates")** are electronic documents used to prove the ownership of public keys
- *Roles*
  - *Certificate Authorities* (CA) store, issue and sign the digital certificates
  - *Registration Authorities* (RA) verify identities of entities requesting their digital certificates be stored at the CA
- *Technologies*
  - *Central directory* provides a secure location in which keys are stored and indexed
  - *Certificate management system*
    - Creates and delivers new certificates to be issued
    - Searches, retrieves and accesses to stored certificates
- *Certificate policy* states procedures for allowing outsiders to analyze the PKI's trustworthiness

Digital Certificate

One of the most important pieces of a PKI

- Associates a public key with information for uniquely identifying its owner

Version	Serial number	Signature	Issuer	Validity	Subject	Subject public key info	Issuer unique ID	Subject unique ID	Extensions
Identifies the version of the certificate	Unique number for the certificate	Algorithm ID used to sign the certificate	Name of certificate issuer	Validity dates	Name of owner	Public key of owner	ID of issuing CA	ID of subject	Optional extensions

Figure 3-47 Each certificate has a structure with all the necessary identifying information in it.

- X.509 standard defines the format of public key certificates used in many Internet cryptographic protocols for HTTPS for servers & clients, secure email, code signing, digital signatures…

Public Key Certificate

Electronic documents used to prove ownership of a public key

Figure 3-47   Each certificate has a structure with all the necessary identifying information in it.

A certificate includes the following common fields:
- Information about the certificate
  - **Serial Number:** Used to uniquely identify the certificate
  - **Issuer:** Entity that verified the information and signed the certificate
  - **Signature Algorithm**: The algorithm used to sign the public key certificate
  - **Signature**: A signature of the certificate body by the issuer's private key

- Information about the public key
  - **Not Before**: Earliest time and date on which the certificate is valid.
  - **Not After**: Time and date past which the certificate is no longer valid
  - **Key Usage**: Valid cryptographic uses of the certificate's public key, e.g. digital signature validation, key encipherment, and certificate signing
  - **Extended Key Usage**: Applications the certificate may be used for, e.g. TLS server authentication, email protection, code signing, or electronic signature

- Information about the identity of its owner (called the subject)
  - **Subject:** Entity a certificate belongs to, e.g. individual, machine, or organization
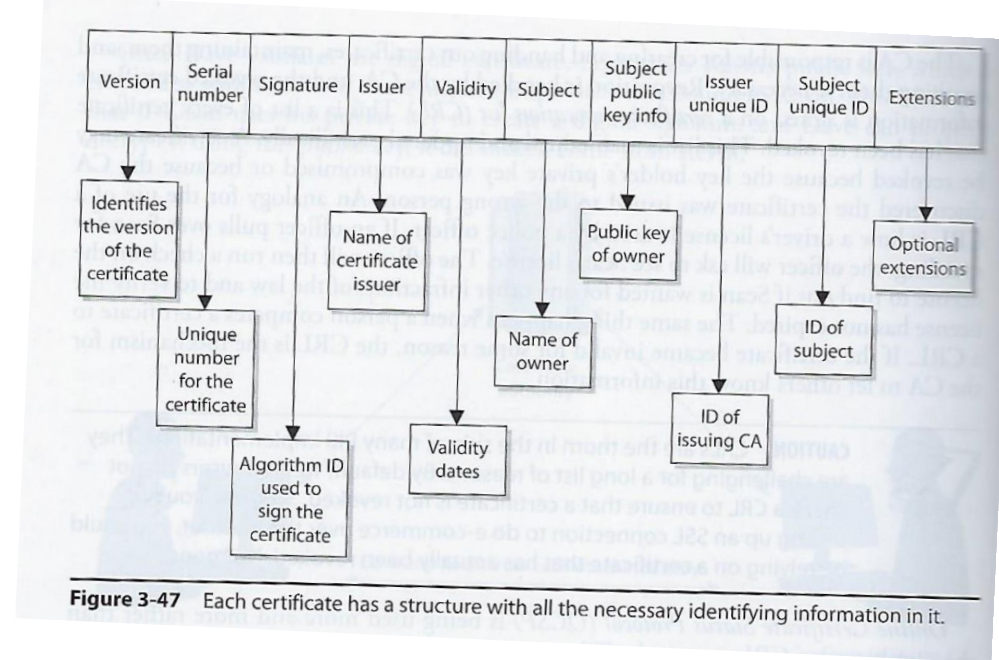
Certificate

https://community.mis.temple.edu/mis4596sec002fall20%

About community.mis.temple.edu ✕

🔒 Connection is secure ›

🔑 Permissions for this site

Cookies (2 cookies in use) ›

📹 Tracking prevention for this site (Balanced) ⬤

Trackers (0 blocked) ›

---

https://community.mis.temple.edu/mis4596sec002fall20%

← **Connection is secure**

This site has a valid certificate issued to Temple University-Of The Commonwealth System of Higher Education [US], issued by a trusted authority.

This means information (such as passwords or credit cards) will be securely sent to this site and cannot be intercepted.

Learn more

Certificate Viewer: community.mis.temple.edu ✕

General   Details

Issued To

Common Name (CN)   community.mis.temple.edu
Organization (O)   Temple University-Of The Commonwealth System of Higher Education
Organizational Unit (OU) <Not Part Of Certificate>

Issued By

Common Name (CN)   GlobalSign Extended Validation CA - SHA256 - G3
Organization (O)   GlobalSign nv-sa
Organizational Unit (OU) <Not Part Of Certificate>

Validity Period

Issued On   Tuesday, December 13, 2022 at 2:06:02 PM
Expires On   Sunday, January 14, 2024 at 2:06:01 PM

Fingerprints

SHA-256 Fingerprint   88 2E 09 05 FC 2A 99 99 CF 7C 13 6B E9 DA 0C 21 77 D2 97 EA 06 16 0E 09 2F B6 DD F6 BC 5C 89 CE

SHA-1 Fingerprint   DF CA A5 83 11 D5 23 BE 76 94 29 06 02 7E 73 36 E5 1A 61 63

Certificate Viewer: community.mis.temple.edu ✕

General   **Details**

Certificate Hierarchy

▼ GlobalSign
　▼ GlobalSign Extended Validation CA - SHA256 - G3
　　community.mis.temple.edu

Certificate Fields

▼ community.mis.temple.edu
　▼ Certificate
　　Version
　　Serial Number
　　Certificate Signature Algorithm
　　Issuer
　▼ Validity
　　Not Before

Field Value

Export...

Certificate Viewer: community.mis.temple.edu

General | **Details**

Certificate Hierarchy

▾ GlobalSign
   ▾ GlobalSign Extended Validation CA - SHA256 - G3
      community.mis.temple.edu

Certificate Fields

▾ community.mis.temple.edu
   ▾ Certificate
      Version
      Serial Number
      Certificate Signature Algorithm
      Issuer
     ▾ Validity
        Not Before

Field Value

CN = GlobalSign Extended Validation CA - SHA256 - G3
O = GlobalSign nv-sa
C = BE

Export...

Certificate Viewer: community.mis.temple.edu

General | **Details**

Certificate Hierarchy

▾ GlobalSign
   ▾ GlobalSign Extended Validation CA - SHA256 - G3
      community.mis.temple.edu

Certificate Fields

Issuer
   ▾ Validity
      Not Before
      Not After
   Subject
   ▾ Subject Public Key Info
      Subject Public Key Algorithm
      Subject's Public Key

Field Value

CN = community.mis.temple.edu
O = Temple University-Of The Commonwealth System of Higher Education
STREET = 1801 N Broad St
L = Philadelphia
ST = Pennsylvania
C = US
jurisdictionStateOrProvinceName = Pennsylvania
jurisdictionCountryName = US
serialNumber = 354000
businessCategory = Private Organization

Export...

Certificate Viewer: community.mis.temple.edu

General **Details**

Certificate Hierarchy

- GlobalSign
  - GlobalSign Extended Validation CA - SHA256 - G3
    - community.mis.temple.edu

Certificate Fields

- Issuer
- Validity
  - Not Before
  - Not After
- Subject
- Subject Public Key Info
  - Subject Public Key Algorithm
  - Subject's Public Key

Field Value

```
Modulus (2048 bits):
  98 8D A5 16 84 27 9F 43 91 3D 5F A4 D3 25 01 43
66 82 82 17 27 DE BE 42 5F 73 C7 30 27 F4 62 92
5B 6D 72 D5 78 71 D8 F0 15 CA A1 9D E8 77 72 A1
97 72 7F 9E 31 DC 9F 8C F7 29 92 4D 3A EC 35 38
CA 31 07 07 2F 4A 03 BF 04 C3 D4 A3 7E 5F FC FD
F6 67 49 ED D2 CC 51 ED 52 72 57 98 DF 31 8A 1C
86 B7 68 AE 47 03 B5 36 CC 70 92 38 C1 D7 CB 39
7D F3 DA 6E F2 17 25 4E ED 43 84 C7 21 4A 7C 1B
C3 0C 16 66 84 3C A2 E2 46 7C A5 8A BF 1B 96 CD
B8 DF B7 C4 88 CB F2 83 C2 05 07 A8 C8 B2 C2 08
F8 AD 4C 22 78 DF 95 8F 2A 53 CC 4C E7 E8 6B B0
2E 13 7D 64 03 21 B9 62 D0 82 F8 EF 89 70 99 E7
1F E9 B9 18 D4 C2 7E E2 61 DC 65 48 5B 52 97 09
31 B5 58 AF 18 81 E8 B9 D0 EB 95 24 BA 90 AD 76
77 78 9F D2 AC C3 1A D4 D8 5A 6F 35 18 2B 12 01
AA F0 6E 44 9E 56 B9 57 28 1B 44 97 21 ED 2D F3

Public Exponent (17 bits):
  01 00 01
```

Export...

---

cryptii

https://cryptii.com

VIEW Text

```
98 8D A5 16 84 27 9F 43 91 3D
5F A4 D3 25 01 43
66 82 82 17 27 DE BE 42 5F 73
C7 30 27 F4 62 92
5B 6D 72 D5 78 71 D8 F0 15 CA
A1 9D E8 77 72 A1
97 72 7F 9E 31 DC 9F 8C F7 29
92 4D 3A EC 35 38
CA 31 07 07 2F 4A 03 BF 04 C3
D4 A3 7E 5F FC FD
F6 67 49 ED D2 CC 51 ED 52 72
57 98 DF 31 8A 1C
86 B7 68 AE 47 03 B5 36 CC 70
92 38 C1 D7 CB 39
7D F3 DA 6E F2 17 25 4E ED 43
84 C7 21 4A 7C 1B
C3 0C 16 66 84 3C A2 E2 46 7C
A5 8A BF 1B 96 CD
B8 DF B7 C4 88 CB F2 83 C2 05
07 A8 C8 B2 C2 08
F8 AD 4C 22 78 DF 95 8F 2A 53
CC 4C E7 E8 6B B0
2E 13 7D 64 03 21 B9 62 D0 82
F8 EF 89 70 99 E7
1F E9 B9 18 D4 C2 7E E2 61 DC
65 48 5B 52 97 09
31 B5 58 AF 18 81 E8 B9 D0 EB
95 24 BA 90 AD 76
77 78 9F D2 AC C3 1A D4 D8 5A
6F 35 18 2B 12 01
AA F0 6E 44 9E 56 B9 57 28 1B
44 97 21 ED 2D F3
```

ENCODE DECODE **Numeral system**

READ
- Binary (2)
- Octal (8)
- Decimal (10)
- Hexadecimal (16)
- Roman numerals

CONVERT TO
- Binary (2)
- Octal (8)
- Decimal (10)
- Hexadecimal (16)
- Roman numerals

→ Encoded 918 chars

VIEW Text

```
152 141 165 22 132 39 159 67
145 61 95 164 211 37 1 67
102 130 130 23 39 222 190 66 95
115 199 48 39 244 98 146
91 109 114 213 120 113 216 240
21 202 161 157 232 119 114 161
151 114 127 158 49 220 159 140
247 41 146 77 58 236 53 56
202 49 7 7 47 74 3 191 4 195
212 163 126 95 252 253
246 103 73 237 210 204 81 237
82 114 87 152 223 49 138 28
134 183 104 174 71 3 181 54 204
112 146 56 193 215 203 57
125 243 218 110 242 23 37 78
237 67 132 199 33 74 124 27
195 12 22 102 132 60 162 226 70
124 165 138 191 27 150 205
184 223 183 196 136 203 242 131
194 5 7 168 200 178 194 8
248 173 76 34 120 223 149 143
42 83 204 76 231 232 107 176
46 19 125 100 3 33 185 98 208
130 248 239 137 112 153 231
31 233 185 24 212 194 126 226
97 220 101 72 91 82 151 9
49 181 88 175 24 129 232 185
208 235 149 36 186 144 173 118
119 120 159 210 172 195 26 212
216 90 111 53 24 43 18 1
170 240 110 68 158 86 185 87 40
27 68 151 33 237 45 243
```

https://cryptii.com/

Types of Certificates: Different cryptographic protocols ("applications")

X.509 is a standard of the International Telecommunications Union which defines the format of public key certificates used in many Internet cryptographic protocols, including:

1. **Transport Layer Security** (TLS/SSL) HTTPS protocol for securely browsing the web
   Certificate's subject is typically a computer or other device, but may also identify organizations or individuals
   - **Server certificate**
     - A server is required to present a certificate as part of the initial connection setup. A client connecting to that server will validate the certificate by checking that
       1. The certificate's subject matches the hostname (i.e. domain name) to which the client is trying to connect
       2. The certificate is signed by a trusted certificate authority
   - **Client certificate** (less common than server certificates)
     - Used to authenticate the client connecting to a TLS service (e.g. for access control)
     - Most client certificates contain an email address or personal name rather than a hostname

2. **Email encryption certificate**
   - A certificate's subject is typically a person or organization
   - For secure email, senders use an email certificate to discover which public key to use for any given recipient

3. **Code signing certificate**
   - A code signing certificate is used to validate signatures on programs to ensure they were not tampered with during delivery

4. **Qualified digital certificate**
   - A "Qualified digital certificate" identifies an individual for electronic signature purposes

Certificate Key Usage

Certificate Viewer: community.mis.temple.edu

General **Details**

Certificate Hierarchy

- GlobalSign
    - GlobalSign Extended Validation CA - SHA256 - G3
        - community.mis.temple.edu

Certificate Fields

- Subject's Public Key
- Extensions
    - Certificate Key Usage
    - Authority Information Access
    - Certificate Policies
    - Certificate Basic Constraints
    - CRL Distribution Points
    - Certificate Subject Alternative Name

Field Value

```
Critical
Signing
Key Encipherment
```

Export...

Roles in PKI - Certificate Authority (CA)

Serves as a trusted third party responsible for verifying identities and signing digital certificates of identity ("digital signature") which are exchanged between two parties introducing themselves to each other

Each person wanting to participate in a PKI requires a digital certificate

- Digital certificate is a credential containing the public key for that individual along with other identifying information

A CA is a trusted organization (or server) responsible for:

- Issuing (creating and handing) out digital certificates
- Maintaining digital certificates
- Revoking digital certificates

Use of PKI and exchanging digital certificates is intended to block Man-in-the-Middle attacks where 2 users are not working in PKI environment do not truly know the identity of the owners of public keys

Roles in PKI - Certificate Authority (CA)

Each person wanting to participate in a PKI requires a digital certificate

- Digital certificate is a credential containing the public key for that individual, computer or organization along with other identifying information

When a CA signs the certificate, it binds the individual's, computer's or organization' identity to the public key

- The CA takes liability for the authenticity of the identity
    - *Making a CA the "trusted 3rd party" that allows people who have never met to use their public keys to authenticate each other and communicate in a secure way*

Certificate Revocation Information

CA's are also responsible for maintaining up-to-date revocation information about certificates they have issued, indicating when certificates of identity are no longer valid

Certificate authority

Dave and Kevin trust each other indirectly.

Kevin trusts the CA.

Dave trusts the CA.
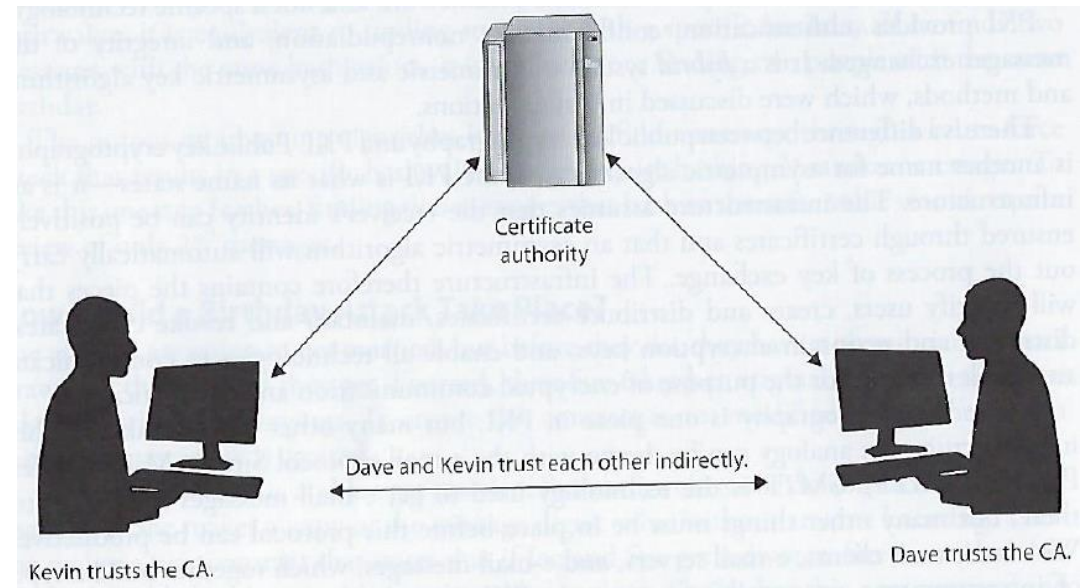
Roles in PKI – Certificate Authority (CA)

New Certificate Requests

A CA processes requests from people or organizations requesting certificates (called "subscribers")

1. Verifies the subscriber's information
2. Potentially signs an end-entity certificate based on the subscriber's information

Identity Information and
Public Key of Mario Rossi

Name:	*Mario Rossi*
Organization:	*Wikimedia*
Address:	*via ....*
Country:	*United States*

Public Key
of
Mario Rossi

Certificate Authority
verifies the identity of Mario Rossi
and encrypts with its Private Key

Certificate of Mario Rossi

Name:	*Mario Rossi*
Organization:	*Wikimedia*
Address:	*via ....*
Country:	*United States*
Validity:	*1997.07.01 - 2047.06.30*

Public Key
of
Mario Rossi

Digital Signature
of the Certificate Authority
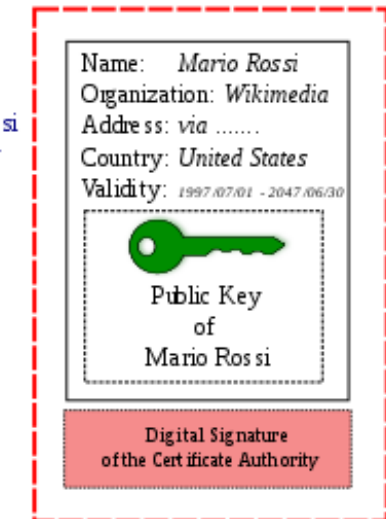
Digitally Signed by
Certificate Authority

Registration Authority (RA)

When a user needs a new certificate, the user makes a request to the RA

RA serves as a broker between the user and the CA, and performs certain certification registration tasks

- Performs the certificate life-cycle management functions
- Establishes and confirms the identity of the individual
  - The RA verifies all the necessary identification information before allowing a request to go to the CA
- Initiates the certification process with the CA for the end user

RA cannot issue certificates

PKI Steps

Suppose: John needs to obtain a digital certificate to participate in PKI

1. John requests a digital certificate from a RA

2. The RA requests John's identification information
   - E.g. driver's license, address, phone number, email, …

3. RA receives John's information, verifies it, and sends his certificate request to CA

4. CA creates a certificate with John's public key and embedded identity information
   - Private/Public key pair is generated on John's machine or by the CA (depends on system configuration)
     - Usually – user generates this pair and sends his public key in as part of registration process
     - If CA creates key pair, John's private key needs to be sent to him via secure means

Now John is registered and is able to participate in PKI
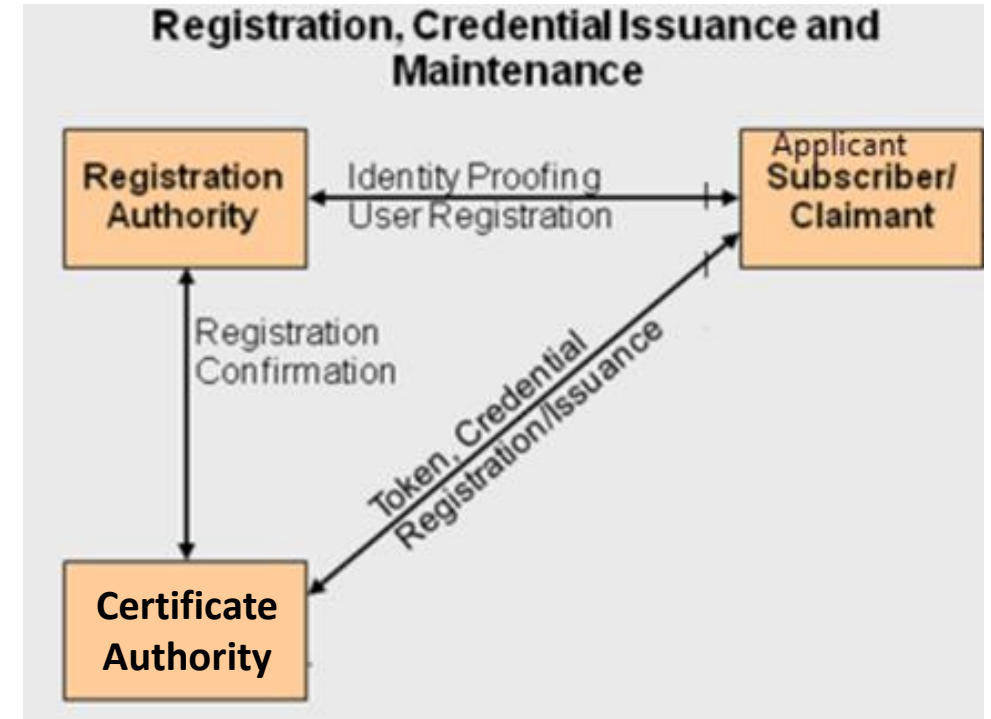
Token, Credential = Public Key

PKI Steps

John and Diane decide to communicate securely using PKI…

1. John requests Diane's public key form a public directory

2. The directory (a.k.a. repository) sends Diane's digital certificate

3. John verifies the digital certificate…
    *   extracts her public key, uses the public key to encrypt a session key that will be used to encrypt their messages
    *   John sends the encrypted session key to Diane
    *   John also sends his certificate, containing his public key to Diane

4. Diane browser receives John's certificate, **looks to see if it trusts the CA** that digitally signed the certificate
    *   Diane's browser trusts this CA
    *   After verifying the certificate, both John and Diane can communicate using encryption

Directory

CA

John requests Diane's public key.

Diane's public key is sent.

Diane validates John's public key.

Session key is encrypted with Diane's public key.

John

Diane

Types of certificates: Chain of trust

A PKI is often set up with multiple levels of CAs, for practical reasons:
- There is a top-level CA, called the root, which issues certificates on the keys of lower-level CAs, which in turn certify the user keys
- The system of identity validation still behaves in the same way, but now Diane has to check two certificates to verify John's key

- **Root certificate**
  - Self-signed certificate used to sign other certificates

- **Intermediate certificate**
  - A certificate used to sign other certificates.
  - Must be signed by either a root certificate or another intermediate certificate

- **End-entity ("leaf") certificate**
  - Cannot be used to sign other certificates
  - Include:
    - TLS/SSL server and client certificates
    - Email certificates
    - Code signing certificates
    - Qualified certificates

End-entity Certificate

| Owner's name |
| Owner's public key |
| Issuer's (CA's) name |
| Issuer's signature |

reference

sign

Intermediate Certificate

| Owner's (CA's) name |
| Owner's public key |
| Issuer's (root CA's) name |
| Issuer's signature |

reference

sign

| Root CA's name |
| Root CA's public key |
| Root CA's signature |

self-sign

Root Certificate

Recall... MIS Community's certificate...

Types of certificates: Chain of trust

End-entity Certificate

Owner's name
Owner's public key
Issuer's (CA's) name
Issuer's signature

reference

Intermediate Certificate

Owner's (CA's) name
Owner's public key
Issuer's (root CA's) name
Issuer's signature

sign

reference

Root CA's name
Root CA's public key
Root CA's signature

sign

self-sign

Root Certificate

Types of certificates: Chain of trust

Certificate Viewer: community.mis.temple.edu   ✕

General   **Details**

Certificate Hierarchy

▼ GlobalSign
  ▼ GlobalSign Extended Validation CA - SHA256 - G3
      community.mis.temple.edu

Certificate Fields

▼ community.mis.temple.edu
  ▼ Certificate
      Version
      Serial Number
      Certificate Signature Algorithm
      Issuer
    ▼ Validity
        Not Before

Field Value

CN = GlobalSign Extended Validation CA - SHA256 - G3
O = GlobalSign nv-sa
C = BE

Export...

Certificate Viewer: community.mis.temple.edu   ✕

General   **Details**

Certificate Hierarchy

▼ GlobalSign
  ▼ GlobalSign Extended Validation CA - SHA256 - G3
      community.mis.temple.edu

Certificate Fields

▼ GlobalSign Extended Validation CA - SHA256 - G3
  ▼ Certificate
      Version
      Serial Number
      Certificate Signature Algorithm
      Issuer
    ▼ Validity
        Not Before

Field Value

CN = GlobalSign
O = GlobalSign
OU = GlobalSign Root CA - R3

Export...

Certificate Viewer: community.mis.temple.edu   ✕

General   **Details**

Certificate Hierarchy

▼ GlobalSign
  ▼ GlobalSign Extended Validation CA - SHA256 - G3
      community.mis.temple.edu

Certificate Fields

▼ GlobalSign
  ▼ Certificate
      Version
      Serial Number
      Certificate Signature Algorithm
      Issuer
    ▼ Validity
        Not Before

Field Value

CN = GlobalSign
O = GlobalSign
OU = GlobalSign Root CA - R3

Export...

CN = Common Name
O = Organization
OU =Organizational Unit
C = Country

Types of certificates: Chain of trust

End-entity Certificate
| Owner's name |
| Owner's public key |
| Issuer's (CA's) name |
| Issuer's signature |

reference

Intermediate Certificate
| Owner's (CA's) name |
| Owner's public key |
| Issuer's (root CA's) name |
| Issuer's signature |

sign   *reference*

Root Certificate
| Root CA's name |
| Root CA's public key |
| Root CA's signature |

sign   *self-sign*

To perform its role effectively, a CA needs to have one or more broadly trusted <u>root certificates </u>or intermediate certificates and the corresponding private keys
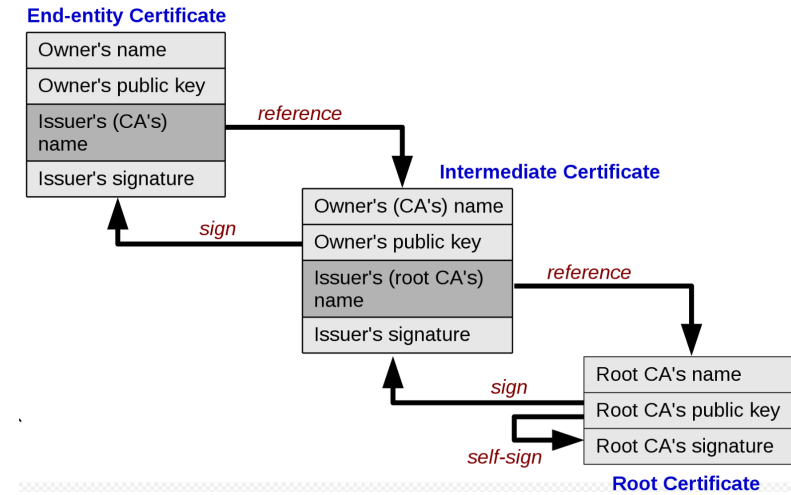
A CA may achieve broad trust by:
Having its root certificates included in popular software
Obtaining a cross-signature from another CA delegating trust

Or a CA may be trusted within a relatively small community, like a business
In which its root certificates are distributed by other mechanisms like Windows Group Policy

Types of certificates: Chain of trust
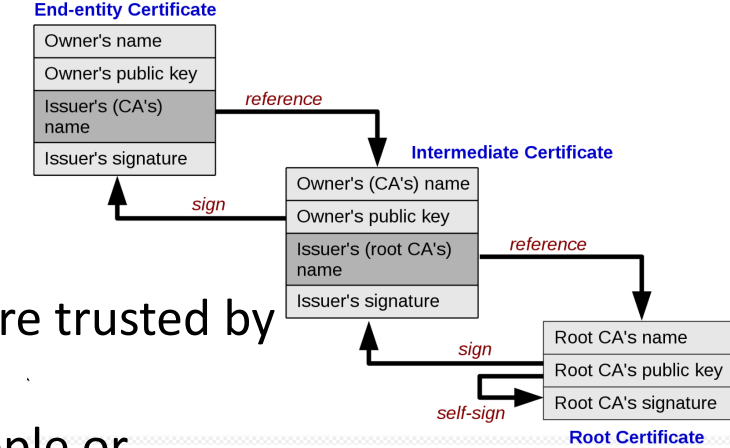
Root programs:

- Some major software products contain a list of certificate authorities that are trusted by default
- This makes it easier for end-users to validate certificates, and easier for people or organizations that request certificates to know which certificate authorities can issue a certificate that will be broadly trusted
- This is particularly important in HTTPS, where a web site operator generally wants to get a certificate that is trusted by nearly all potential visitors to their web site

The most influential root programs are:
- Microsoft Root Program
- Apple Root Program
- Mozilla Root Program
- Oracle Java root program
- Adobe Approved Trust List and EUTL root programs (used for document signing)

End-entity Certificate

| Owner's name |
| Owner's public key |
| Issuer's (CA's) name |
| Issuer's signature |

reference

Intermediate Certificate

| Owner's (CA's) name |
| Owner's public key |
| Issuer's (root CA's) name |
| Issuer's signature |

sign

reference

| Root CA's name |
| Root CA's public key |
| Root CA's signature |

sign

self-sign

Root Certificate

Types of certificates: Chain of trust

End-entity Certificate

| Owner's name |
| Owner's public key |
| Issuer's (CA's) name |
| Issuer's signature |

reference

Intermediate Certificate

| Owner's (CA's) name |
| Owner's public key |
| Issuer's (root CA's) name |
| Issuer's signature |

sign   *reference*

Root Certificate

| Root CA's name |
| Root CA's public key |
| Root CA's signature |

sign   *self-sign*

Root programs:

Browsers generally use the operating system's facilities to decide which certificate authorities are trusted:

- Google Chrome on Windows trusts certificate authorities included in Microsoft Root Program
- Google Chrome on macOS or iOS trusts certificate authorities in Apple Root Program
- Edge and Safari use their respective operating system trust stores as well, but each is only available on a single OS.

- Firefox, in contrast, uses the Mozilla Root Program trust store on all platforms

Microsoft Windows Root Program's Trust Stores

1. Run **mmc.exe**

2. Select **File -> Add/Remove Snap-in**

3. Select **Certificates**, click **Add**

4. Select **Computer Account,** click **next,** click **Finish**

5. Expand the **Certificates** node -> **Trusted Root Certificate Authorities Store**

Microsoft Windows Root Program's Trust Stores

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	S
AAA Certificate Services	AAA Certificate Services	12/31/2028	Client Authenticatio...	Sectigo (AAA)	
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/12/2025	Client Authenticatio...	DigiCert Baltimore R...	
Certum Trusted Network CA	Certum Trusted Network CA	12/31/2029	Client Authenticatio...	Certum Trusted Net...	
Class 3 Public Primary Certificati...	Class 3 Public Primary Certification ...	8/1/2028	Client Authenticatio...	VeriSign Class 3 Pub...	
COMODO RSA Certification Aut...	COMODO RSA Certification Autho...	1/18/2038	Client Authenticatio...	Sectigo (formerly Co...	
Copyright (c) 1997 Microsoft Corp.	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping	Microsoft Timestam...	
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/9/2031	<All>	<None>	
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/9/2031	Client Authenticatio...	DigiCert	
DigiCert Global Root CA	DigiCert Global Root CA	11/9/2031	Client Authenticatio...	DigiCert	
DigiCert Global Root G2	DigiCert Global Root G2	1/15/2038	Client Authenticatio...	DigiCert Global Roo...	
DigiCert Global Root G3	DigiCert Global Root G3	1/15/2038	Client Authenticatio...	DigiCert Global Roo...	
DigiCert High Assurance EV Roo...	DigiCert High Assurance EV Root CA	11/9/2031	Client Authenticatio...	DigiCert	
DigiCert Trusted Root G4	DigiCert Trusted Root G4	1/15/2038	Client Authenticatio...	DigiCert Trusted Ro...	
DST Root CA X3	DST Root CA X3	9/30/2021	Client Authenticatio...	DST Root CA X3	
Entrust Root Certification Autho...	Entrust Root Certification Authority	11/27/2026	Client Authenticatio...	Entrust	
Entrust Root Certification Autho...	Entrust Root Certification Authorit...	12/7/2030	Client Authenticatio...	Entrust.net	
Entrust.net Certification Authorit...	Entrust.net Certification Authority (...	7/24/2029	Client Authenticatio...	Entrust (2048)	
GlobalSign	GlobalSign	3/18/2029	<All>	<None>	
GlobalSign	GlobalSign	3/18/2029	Client Authenticatio...	GlobalSign Root CA ...	
GlobalSign Root CA	GlobalSign Root CA	1/28/2028	<All>	<None>	
GlobalSign Root CA	GlobalSign Root CA	1/28/2028	Client Authenticatio...	GlobalSign Root CA ...	
Go Daddy Class 2 Certification A...	Go Daddy Class 2 Certification Aut...	6/29/2034	Client Authenticatio...	Go Daddy Class 2 C...	
Go Daddy Root Certificate Auth...	Go Daddy Root Certificate Authori...	12/31/2037	<All>	<None>	
Go Daddy Root Certificate Auth...	Go Daddy Root Certificate Authori...	12/31/2037	Client Authenticatio...	Go Daddy Root Cert...	
ISRG Root X1	ISRG Root X1	6/4/2035	Client Authenticatio...	ISRG Root X1	
Logitech Inc	DigiCert Trusted G4 Code Signing ...	4/10/2025	Code Signing	<None>	
Microsoft Authenticode(tm) Roo...	Microsoft Authenticode(tm) Root ...	12/31/1999	Secure Email, Code ...	Microsoft Authentic...	
Microsoft ECC Product Root Cert...	Microsoft ECC Product Root Certifi...	2/27/2043	<All>	Microsoft ECC Prod...	
Microsoft ECC TS Root Certificat...	Microsoft ECC TS Root Certificate ...	2/27/2043	<All>	Microsoft ECC TS Ro...	
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	<All>	Microsoft Root Aut...	
Microsoft Root Certificate Autho...	Microsoft Root Certificate Authority	5/9/2021	<All>	Microsoft Root Certi...	
Microsoft Root Certificate Autho...	Microsoft Root Certificate Authorit...	6/23/2035	<All>	Microsoft Root Certi...	
Microsoft Root Certificate Autho...	Microsoft Root Certificate Authorit...	3/22/2036	<All>	Microsoft Root Certi...	
Microsoft Time Stamp Root Cert...	Microsoft Time Stamp Root Certifi...	10/22/2039	<All>	Microsoft Time Sta...	
NO LIABILITY ACCEPTED, (c)97 Ve...	NO LIABILITY ACCEPTED, (c)97 VeriS...	1/7/2004	Time Stamping	VeriSign Time Stam...	
QuoVadis Root CA 2 G3	QuoVadis Root CA 2 G3	1/12/2042	Client Authenticatio...	QuoVadis Root CA 2...	
QuoVadis Root Certification Aut...	QuoVadis Root Certification Autho...	3/17/2021	Client Authenticatio...	QuoVadis Root Certi...	
SecureTrust CA	SecureTrust CA	12/31/2029	Client Authenticatio...	Trustwave	
Security Communication RootCA1	Security Communication RootCA1	9/29/2023	Client Authenticatio...	SECOM Trust Syste...	
Security Communication RootCA2	Security Communication RootCA2	5/29/2029	Client Authenticatio...	SECOM Trust Syste...	
SSL.com Root Certification Auth...	SSL.com Root Certification Authori...	2/12/2041	Client Authenticatio...	SSL.com Root Certifi...	
Starfield Class 2 Certification Aut...	Starfield Class 2 Certification Auth...	6/29/2034	Client Authenticatio...	Starfield Class 2 Cert...	
Starfield Root Certificate Authori...	Starfield Root Certificate Authority...	12/31/2037	Client Authenticatio...	Starfield Root Certifi...	
Symantec Enterprise Mobile Ro...	Symantec Enterprise Mobile Root f...	3/14/2032	Code Signing	<None>	
Thawte Timestamping CA	Thawte Timestamping CA	12/31/2020	Time Stamping	Thawte Timestampi...	
USERTrust ECC Certification Auth...	USERTrust ECC Certification Author...	1/18/2038	Client Authenticatio...	Sectigo ECC	
USERTrust RSA Certification Auth...	USERTrust RSA Certification Author...	1/18/2038	Client Authenticatio...	Sectigo	
VeriSign Class 3 Public Primary C...	VeriSign Class 3 Public Primary Cert...	7/16/2036	Client Authenticatio...	VeriSign	
VeriSign Universal Root Certifica...	VeriSign Universal Root Certificatio...	12/1/2037	<All>	<None>	

MIS 5214

Mac OS X

The root store is in the Keychain.app

1. Search Finder (Spotlight) for "keychain"
2. Double-click Keychain Access app
3. Select "System Roots" in the left-hand pane

Certificate Revocation List (CRL) – in principal

CRL is the mechanism for the CA to let others know that a certificate has become invalid for some reason
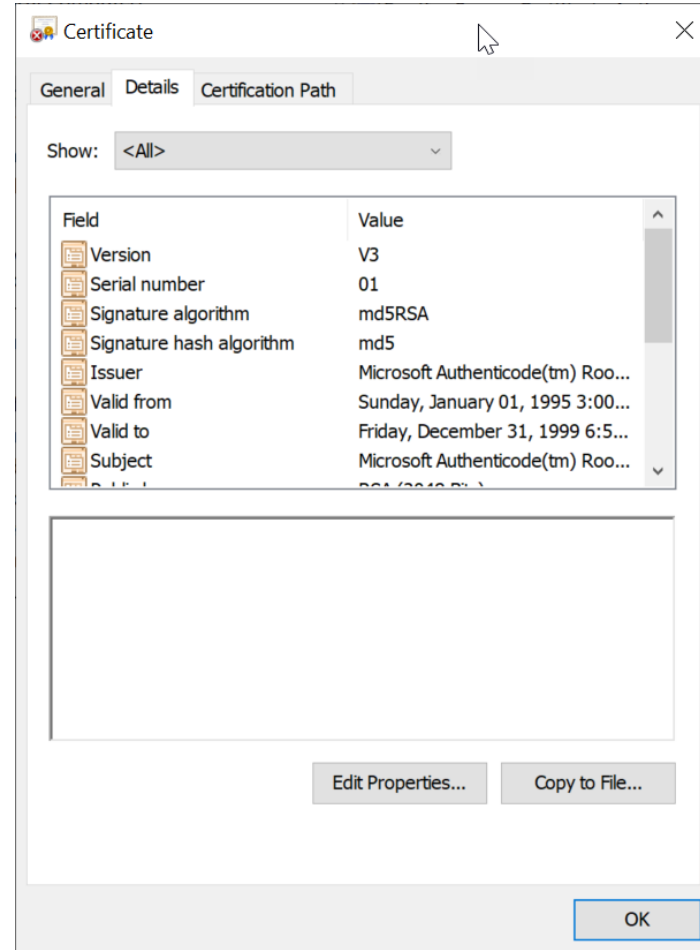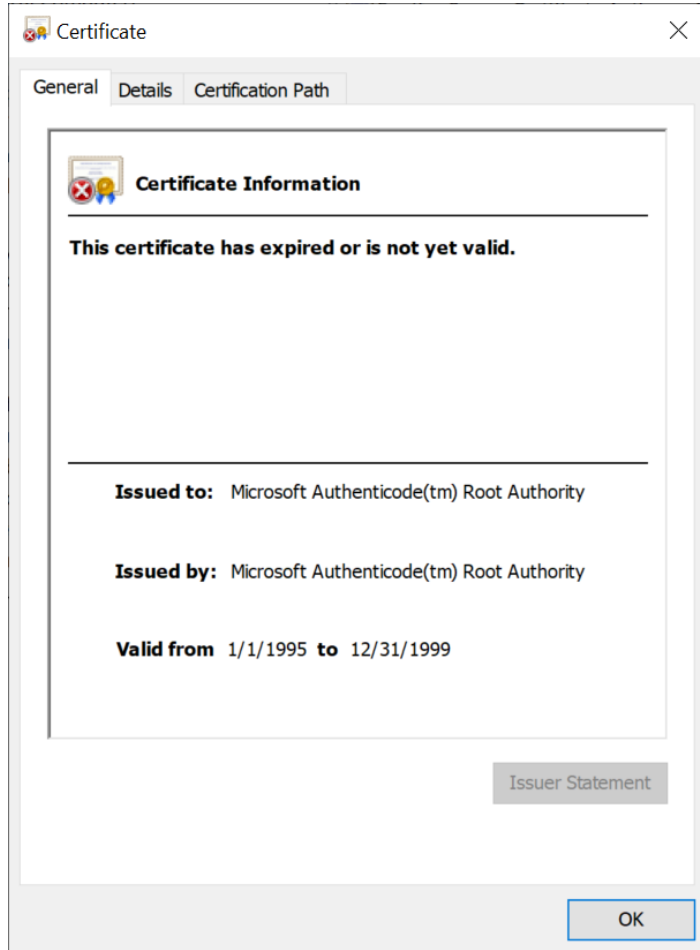
A certificate may be revoked because
- The key holder's private key was compromised
- CA discovered the Certificate was issued to the wrong person
- The certificate expired
- The certificate became invalid for other reasons...

The CA handles revocation by putting the revoked certificate's information on a ***certificate revocation list*** (CRL)
- The CRL is a list of every certificate that has been revoked
- The CRL is maintained and updated

Microsoft Windows Root Program's Trust Stores

Examples of Browsers Rejecting Revoked Certificates

- **Safari:** Generic `This Connection is Not Private` message. If you click the **Show Details** button and then the **view the certificate** link, you can confirm that the certificate is, in fact, revoked.

- **Chrome:** `NET::ERR_CERT_REVOKED`

Your connection is not private

Attackers might be trying to steal your information from **revoked-ecc-dv.ssl.com** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_REVOKED

SSL.com Root Certification Authority RSA
↳ SSL.com RSA SSL subCA
↳ revoked-rsa-dv.ssl.com

revoked-rsa-dv.ssl.com
Issued by: SSL.com RSA SSL subCA
Expires: Sunday, June 27, 2021 at 9:13:48 AM Central Daylight Time
⊗ "revoked-rsa-dv.ssl.com" certificate is revoked

▸ **Details**

?   OK

- **Edge:** `ERROR_INTERNET_SEC_CERT_REVOKED` (visible after clicking **Details** link on **This site is not secure** message).

This site is not secure

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

▢ Go to your Start page

Details

This website's security certificate has been revoked, so you can't go there at this time.

Error Code:
ERROR_INTERNET_SEC_CERT_REVOKED
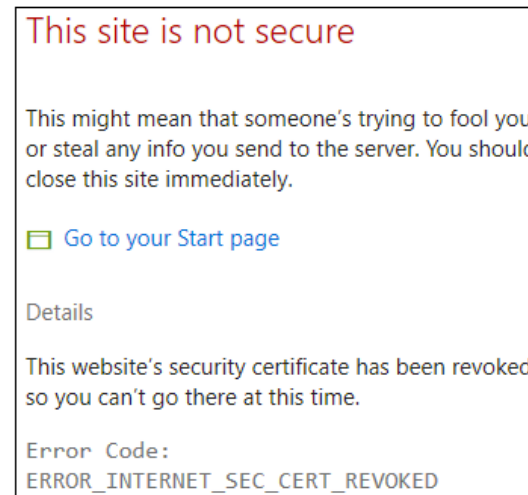
Certificate Revocation List (CRL) – in practice

CRLs are problematic in many PKI implementations for many reasons
- Either user's browser must check a central CRL to find out if a certificate has been revoked
- …or the CA must continually push out CRL values to clients to ensure they have an updated CRL

By default, web browsers do not check a CRL to ensure that a certificate is not revoked
- So when you are setting up a SSL connection to do e-Commerce over the Internet, you may be relying on a revoked certificate and not know it

Online Certificate Status Protocol (OCSP) is increasingly being used…
- If OCSP is implemented, it works automatically
- OCSP does real-time certificate validation
  - Checks the CRL maintained by the CA
  - Notifies user if certificate is valid, invalid, or unknown
- Publicly trusted CAs (e.g. SSL.com) maintain HTTP servers called OCSP responders
  - OCSP responders sign their responses with the CA's private signing key so browsers can verify that the received revocation status was generated by the actual CA

Example of "invalid" certificate: https://www.iad.gov/nietp/caerequirements.cfm

PKI Roles and Workflows
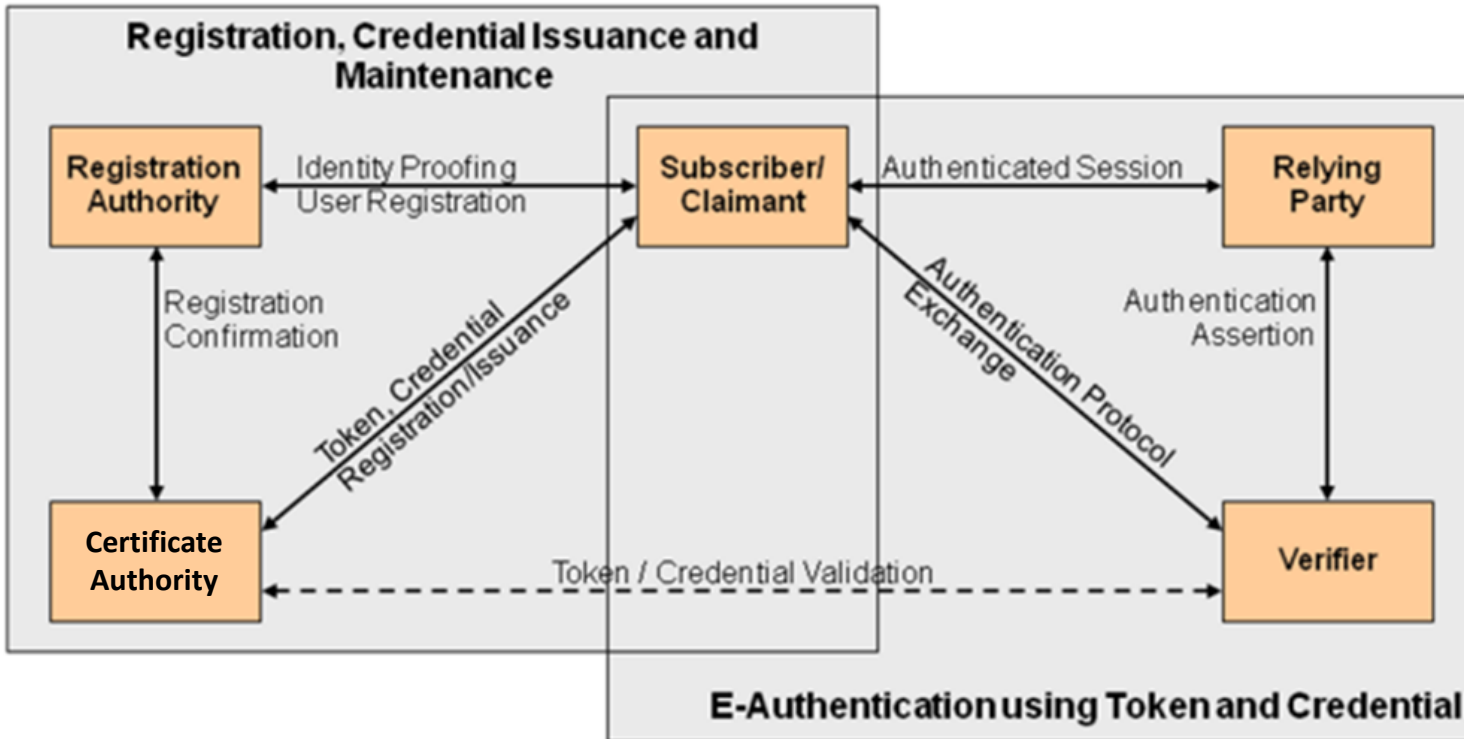
Token, Credential = Public Key

Basic Online Certificate Status Protocol (OCSP)

1. Alice and Bob have public key certificates issued by Carol, the certificate authority (CA)
2. Alice wishes to perform a transaction with Bob and sends him her public key certificate
3. Bob, concerned that Alice's public key may have been compromised, creates an 'OCSP request' that contains Alice's certificate serial number and sends it to Carol
4. Carol's OCSP responder reads the certificate serial number from Bob's request. The OCSP responder uses the certificate serial number to look up the revocation status of Alice's certificate. The OCSP responder looks in a CA database that Carol maintains. In this scenario, Carol's CA database is the only trusted location where a compromise to Alice's certificate would be recorded
5. Carol's OCSP responder confirms that Alice's certificate is still OK, and returns a signed, successful 'OCSP response' to Bob
6. Bob cryptographically verifies Carol's signed response. Bob has stored Carol's public key sometime before this transaction. Bob uses Carol's public key to verify Carol's response
7. Bob completes the transaction with Alice

Online Certificate Status Protocol (OCSP) – In Practice

Contacting a responder and waiting for a response for every certificate encountered by a browser encounters adds perceptible network overhead, especially in pages containing third-party content stored in remote content-distribution servers

- *Amazon calculated that a delay of one second can cost them about $1.6 billion yearly*

This motivated browsers and other client software to implement OCSP checking in soft-fail mode

If an OCSP server cannot be reached or times out while giving its response, browsers consider the certificate valid and proceed with the HTTPS connection anyway

Man-in-the-middle (MITM) attackers can exploit this behavior by blocking all connections to OCSP responders, and then can use a stolen certificate and key pair for a malicious site, regardless of the certificate's revocation status

OCSP Stapling Solution

Servers include (or **staple**) the cached OCSP response in their HTTPS responses alongside the SSL certificate

- This enables browsers before the secure connection is established to verify the CAs signature on the OCSP response and be assured that the certificate has not been revoked
- OCSP stapling enables servers to retrieve cached OCSP responses in non-real-time and remove performance overhead imposed by CRLs and OCSP
- OCSP stapling does not completely solve OCSP's soft-fail security issue, since stapling is implemented in the server and browsers cannot know if a server actually supports Stapling or not

- OCSP Must-Staple (extension of SSL Certificates: [RFC 7633](#))
  - Mandates OCSP stapling for the certificate
  - If a browser encounters a certificate with this extension that is used without OCSP Stapling, then it will be rejected
  - Enabling OCSP stapling on servers improves security and performance for your web site at the same time

https://www.ssl.com/article/page-load-optimization-ocsp-stapling/

Agenda

- ✓ Public Key Infrastructure
- ✓ Digital Certificate
- ✓ Public key Certificates
- ✓ Roles in PKI: Certificate Authority (CA)
- ✓ Roles in PKI: Registration Authority (RA)
- ✓ PKI Steps
- ✓ Chain of Trust
- ✓ Root Programs
- ✓ Certificate Revocation List (CRL)
- ✓ PKI Roles / Workflows…