Public Key Management

- Distribution of public keys is possible by
  - Public announcement
  - Publicly available directory
  - Public-key authority
  - Public-key certificates

Public Announcement

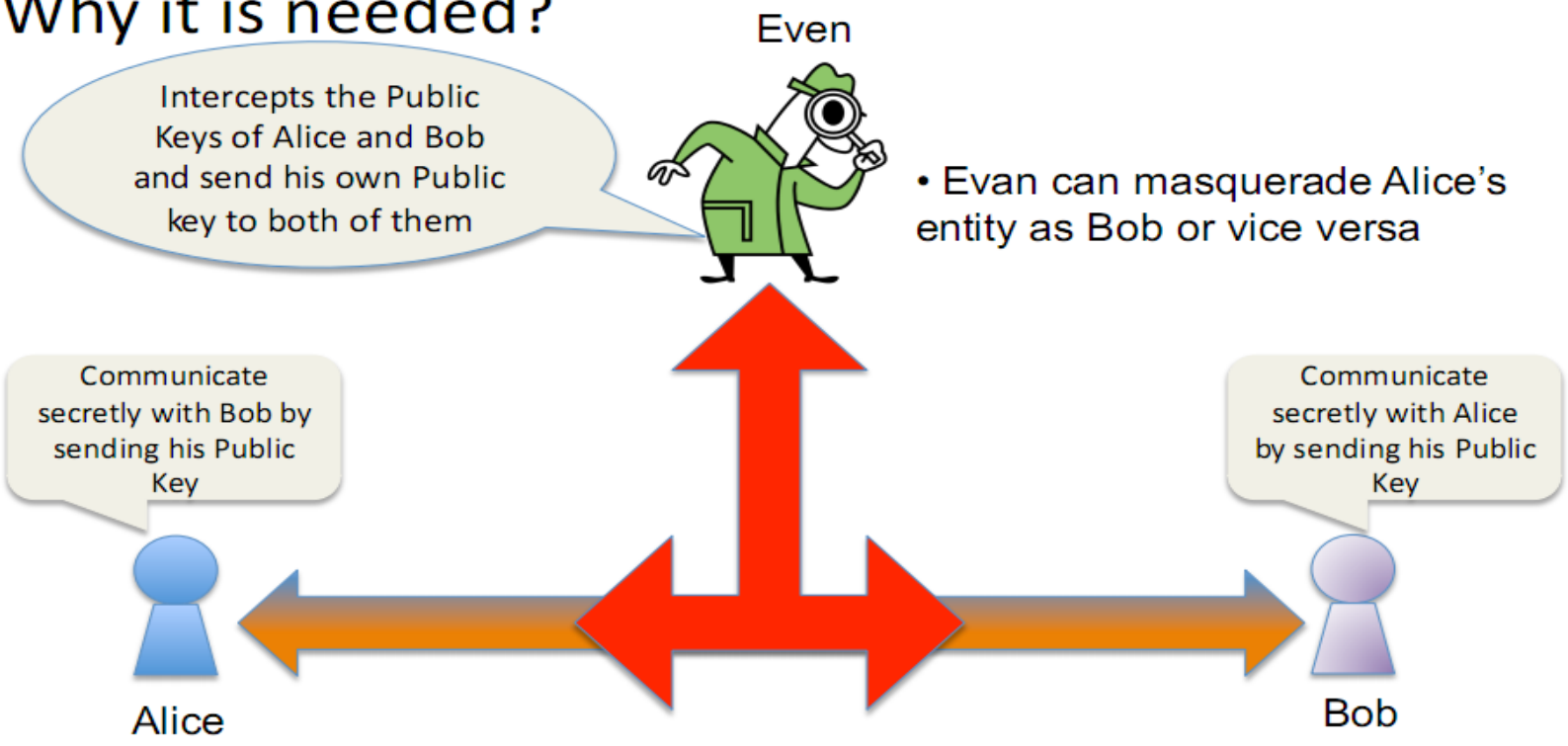
- Users distribute public keys to recipients or broadcast to community at large
  - e.g. post to news groups or email list
- Weakness: Anyone can easily forge such public announcements
  - Create a key claiming to be someone else and broadcast it
  - Masquerade as claimed user until forgery is discovered

Publicly Available Directory

- Users can register keys to a **Trusted** Public Directory
- **Requirements** for a Trusted Directory (in general):
  - contains {name, public-key} entries
  - participants register securely with directory
  - participants can replace key at any time
  - directory is periodically published
  - directory can be accessed electronically
- Still vulnerable to tampering or forgery, if channel or access to directory is vulnerable

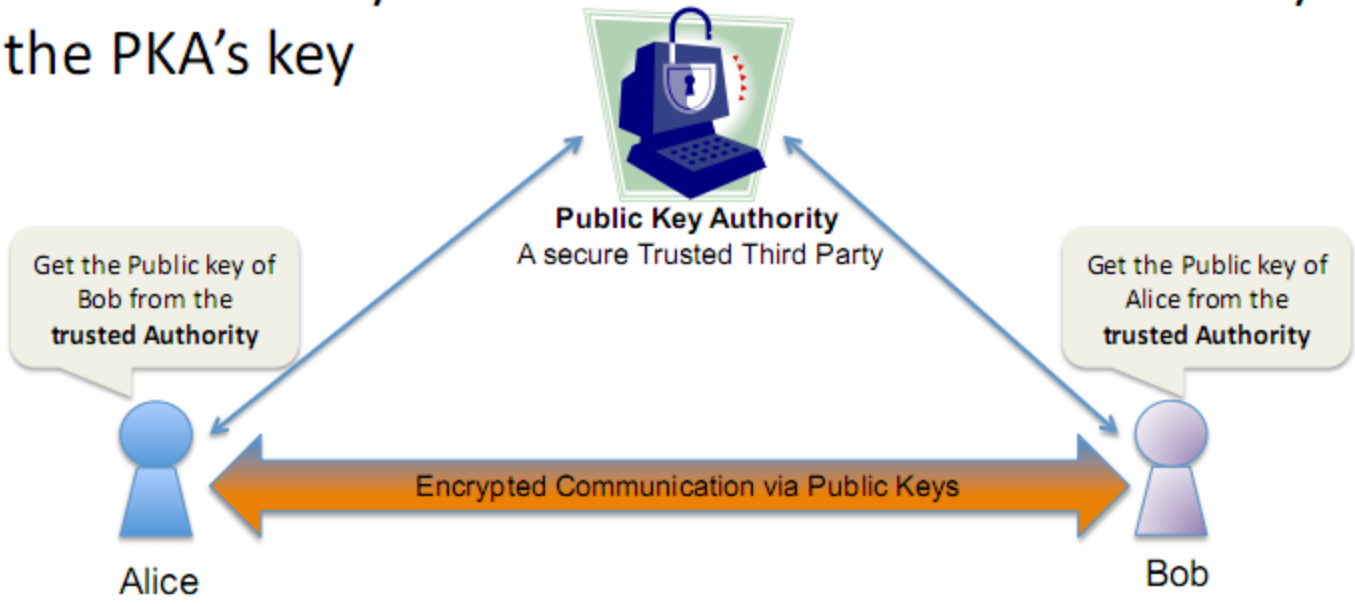
Public Key Authority (PKA)

- A well-trusted Authority is required to distribute keys from the directory
- Why it is needed?

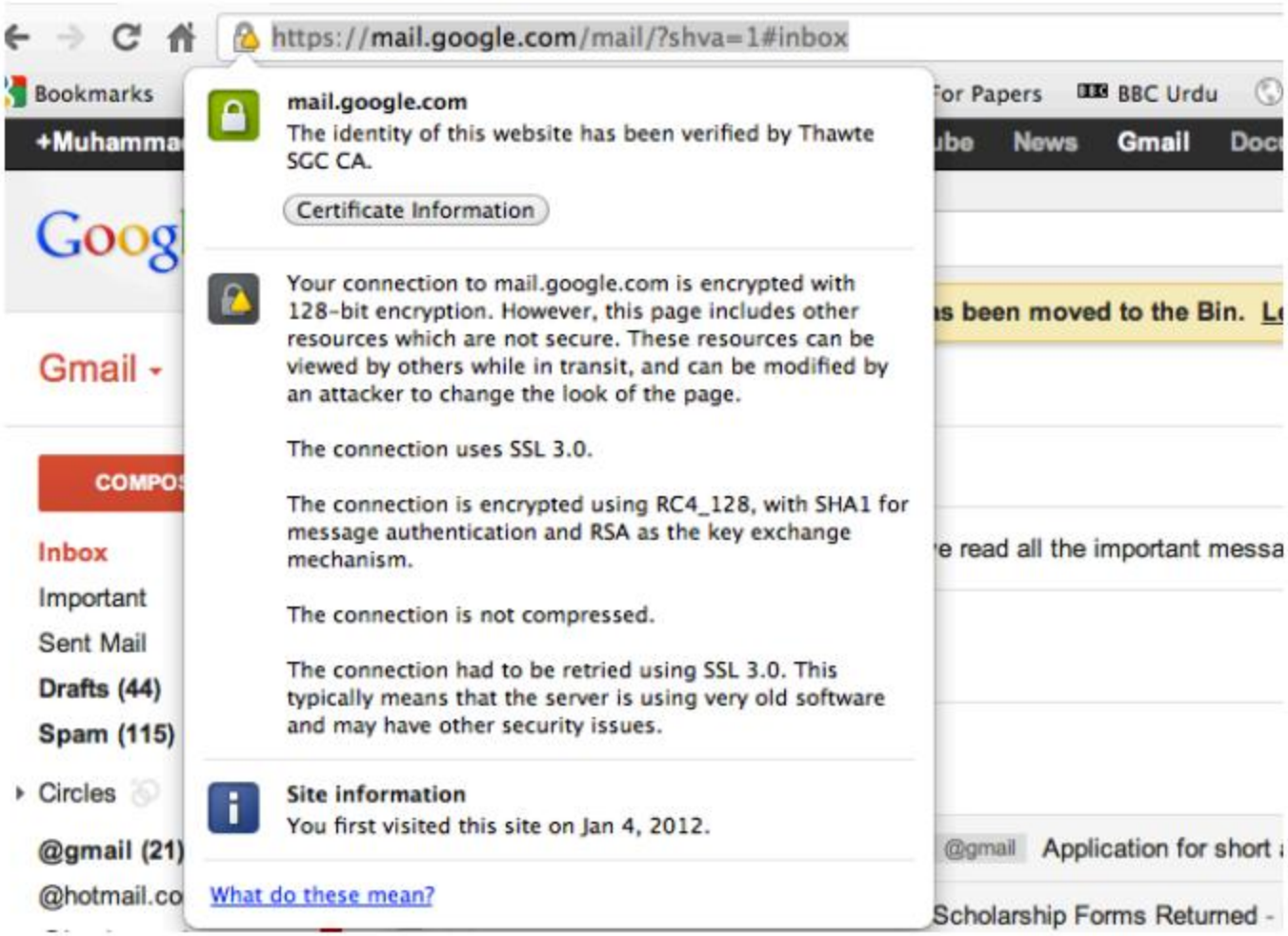


Public Key Authority (PKA)

- **Primary Role:** to digitally sign and publish the public key bound to a given user
- PKA does this with its own private key, so that trust in the user key relies on one's trust in the validity of the PKA's key

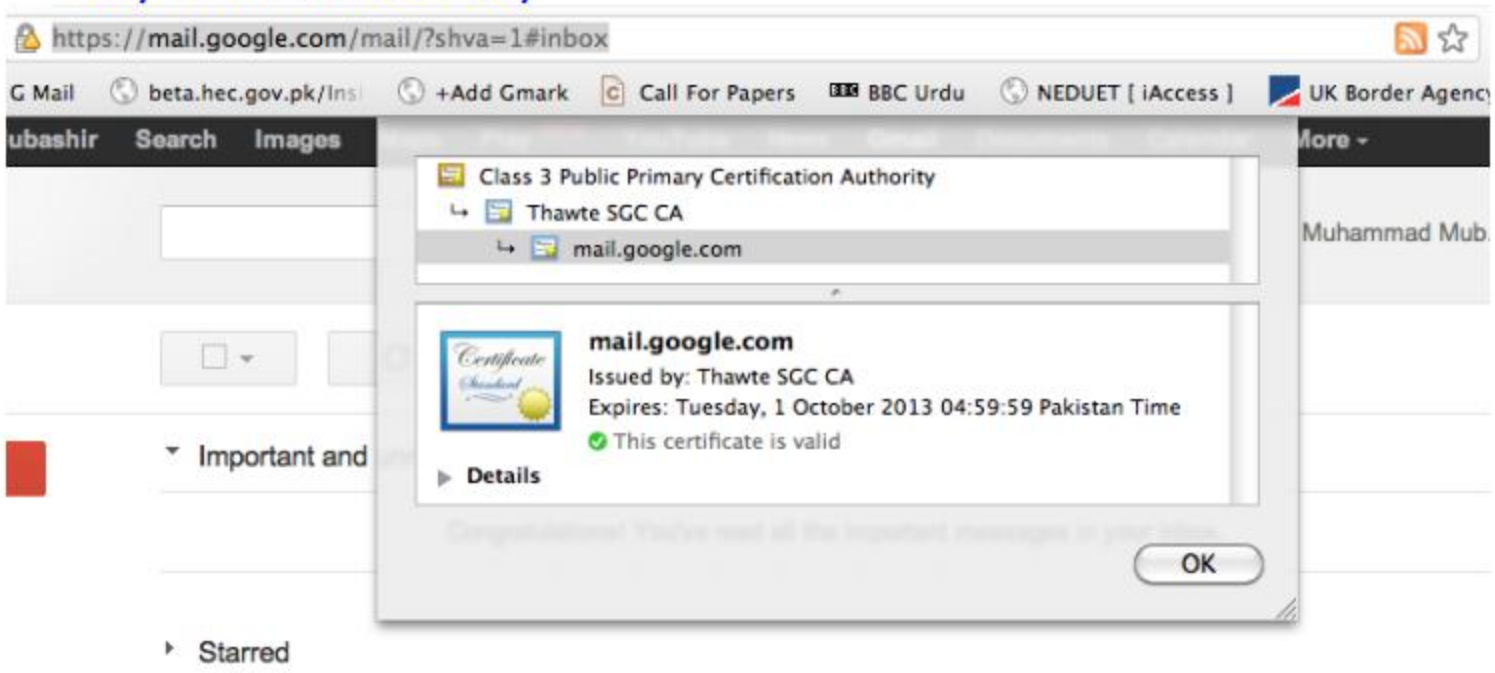


Public Key Certificates



Public Key Certificates

- In cryptography, a public key certificate (also known as a digital certificate or identity certificate) is an electronic document which uses a digital signature to bind a public key with an identity

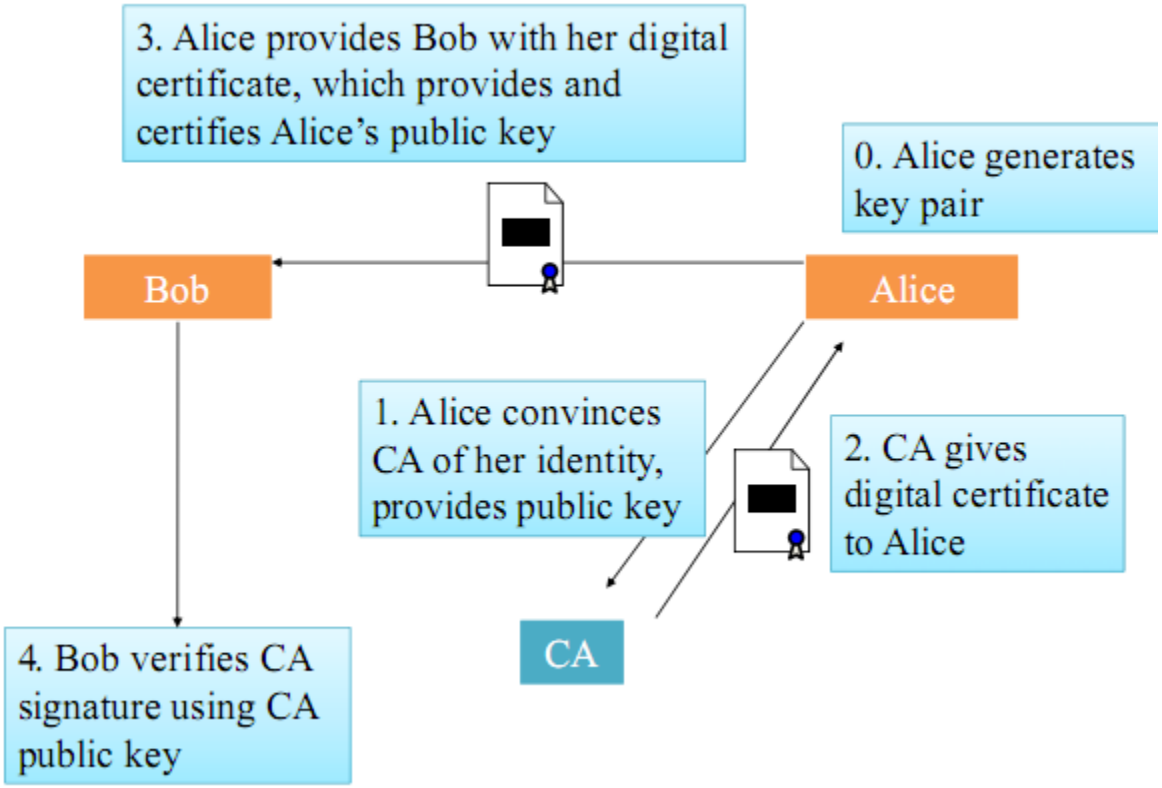


X.509 Public Key Certificate

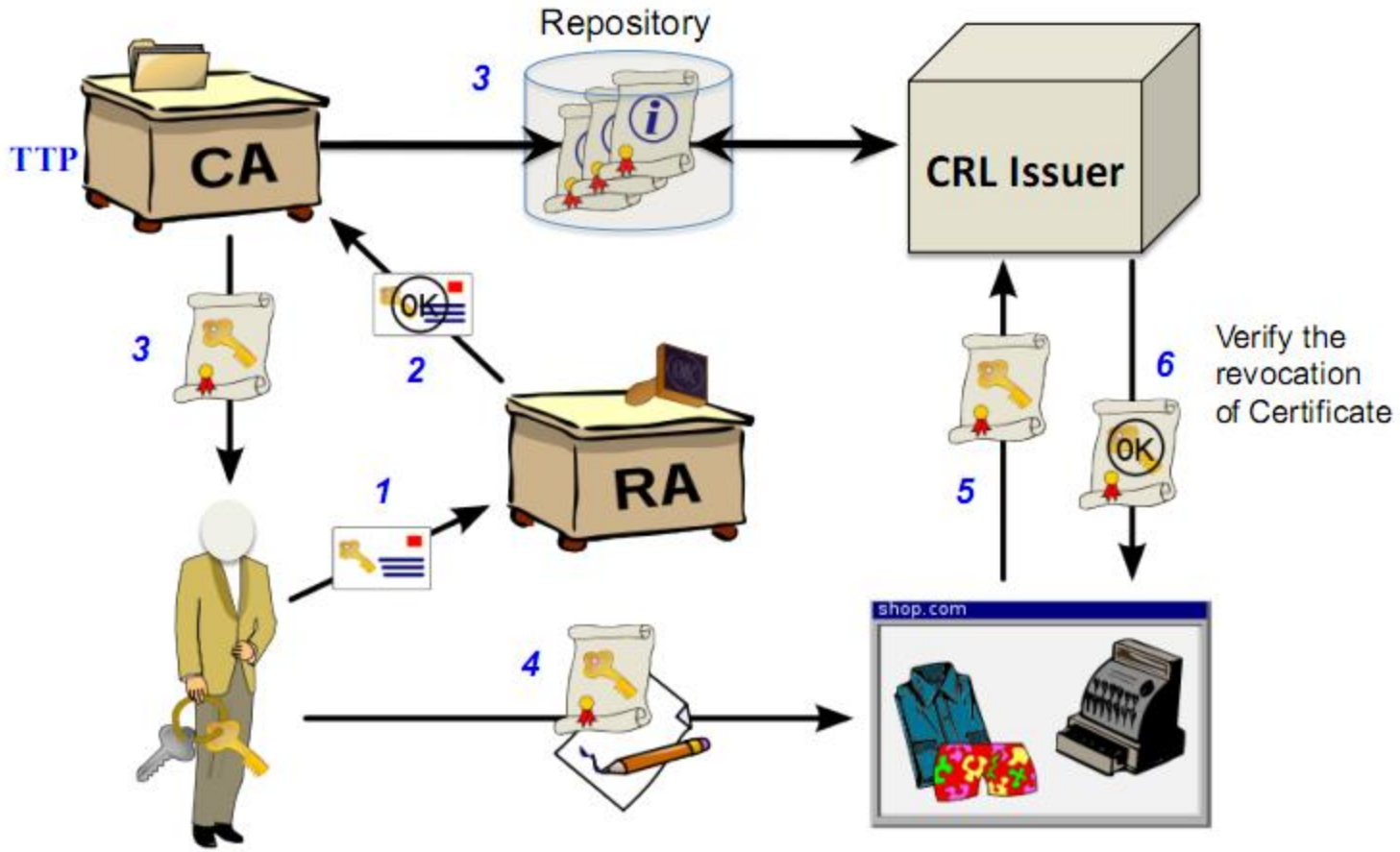
A Complete Picture

Public Key Infrastructure (PKI)

- A system for the creation, storage, and distribution of digital certificates which are used to verify that a particular Public Key belongs to a certain Entity.
- The PKI creates **digital certificates** which map public keys to entities, securely stores these certificates in a central repository, and revokes them if needed.
- A PKI consists of
  - **Certificate Authority (CA):** Both issues and verifies the digital certificates.
  - **Registration Authority** which verifies the identity of users requesting information from the CA
  - **Repository:** A central directory -- i.e. a secure location in which to store and index keys.
  - **Archive:** Store sufficient information to be used for solving future disputes of old documents



Public Key Infrastructure (PKI)



Certificate Revocation List (CRL)

- A certificate may become unreliable before the expiration date arrives, hence needed to be revoked
- Reasons:
  - user's private key is compromised
  - user is no longer certified by this CA
  - CA's certificate is compromised,... etc.
- X.509 certification revocation list (CRL) is a mechanism for preventing the use of unreliable certificates
- CRL should always keep the latest updated information
- Should be digitally signed and authenticated

X.509 CRL Format

The CRL contains the following main fields with certain extensions:

- **Version:** describes the syntax of the CRL
- **Signature:** contains the algorithm identifier for the digital signature algorithm used by the CRL issuer to sign the CRL
- **Issuer:** contains the X.500 distinguished name of the CRL issuer
- **This update:** indicates the issue date of this CRL
- **Next update:** indicates the date by which the next CRL will be issued
- **Revoked certificates:** lists the revoked certificates (contains the certificate serial number, time of revocation, and optional CRL entry extensions)

Practical issues with Certificate Revocation

- Different techniques for informing about certificate revocation

1. To publish a CRL
   - Updates are not in real-time
   - Long intervals between CRL distributions
   - Expensive to distribute
   - CRL request implosion (concurrent requests)
2. OCSP-Online certificate status protocol
   - status verification is computationally expensive (response must be digitally signed)

PKI Architectures

- Problem: Communicating users belong to different CA's.
- Solutions:
  1. CA Hierarchy
  2. Mesh Infrastructure
  3. Bridge CA's

CA Hierarchy

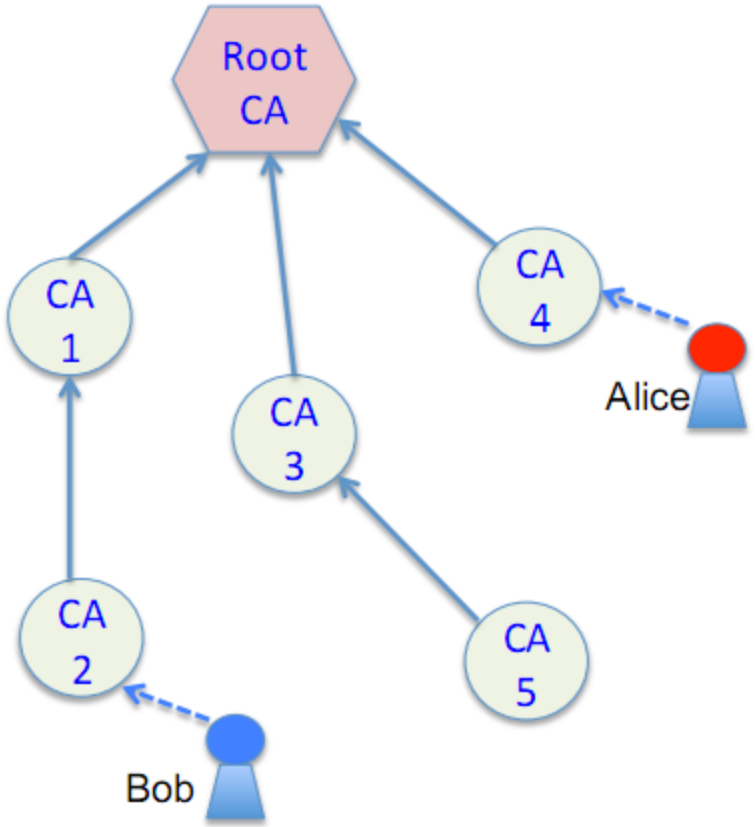
- Authorities are arranged hierarchically under a "root" CA that issues certificates to subordinate CAs.

• CAs may issue certificates to CAs below them in the hierarchy, or to users

• Every relying party knows the public key of the root CA.

• Any certificate may be verified by verifying the certification path of certificates from the root CA.

• Alice verifies Bob's certificate, issued by CA 2, then CA 2's certificate, issued by CA 1, and then CA 1's certificate issued by Root CA, whose public key she knows.



Mesh Infrastructure

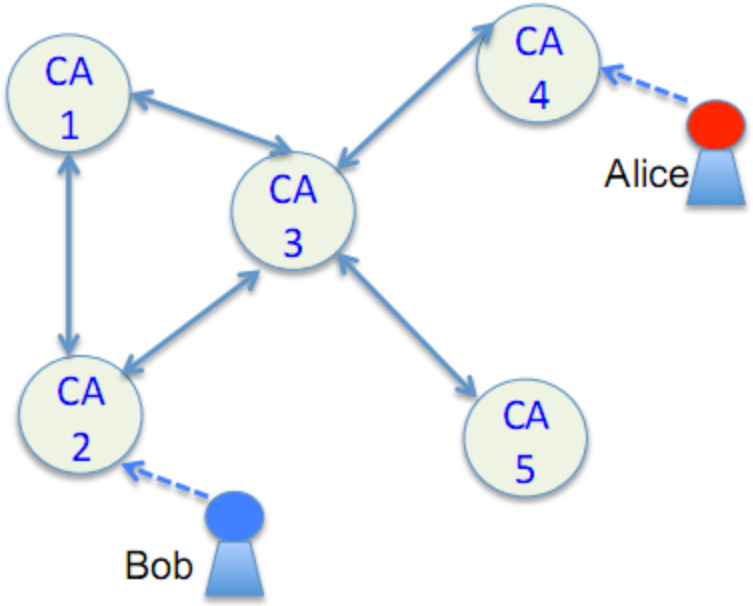
- Independent CA's cross certify each other by issuing certificates to each other

• Several certification paths are possible between Alice and Bob

• Alice knows the public key of CA 4, while Bob knows the public key of CA 2

• Any certificate may be verified by verifying the certification path of certificates from the root CA.

• Alice verifies Bob's certificate, issued by CA 2, then CA 2's certificate, issued by CA 3, and then CA 3's certificate issued by CA 4, Alice's Trusted CA, whose public key is known to her.



Bridge PKI Architecture

- Users can use the bridge of trust that exists through the Bridge CA to establish relationships with each other
- Users do not directly trust on Bridge CA
- Alice & Carol trust each other because their respective CA's trust on a common bridge CA