FD6 – Introduction to IP Security

Protecting the Data in the Network

- **Ensure confidentiality of data**
  - Solution: Symmetric encryption
    - RC2, RC4, DES, 3DES, AES, CAST, IDEA, Blowfish
- **Protect the encryption keys**
  - Solution: Asymmetric encryption
    - RSA, DSA, Diffie-Hellman, Elliptic Curve
- **Ensure data integrity and non-repudiation**
  - Solution: Digital signatures
    - MD5, SHA
- **Manage identities and encryption keys**
  - Solution: Digital certificates
  - Solution: Public Key Infrastructure

Cryptography

- From the Greek word "KRYPTOS", Cryptography is the study of ways to convert information from a readable form and put it into an unreadable form.

Message → h7TWz~rD9yuI?wQss+

Plain Text **Ciphertext**

Steganography (by contrast)

- The study of ways to hide information within other information.

Message →

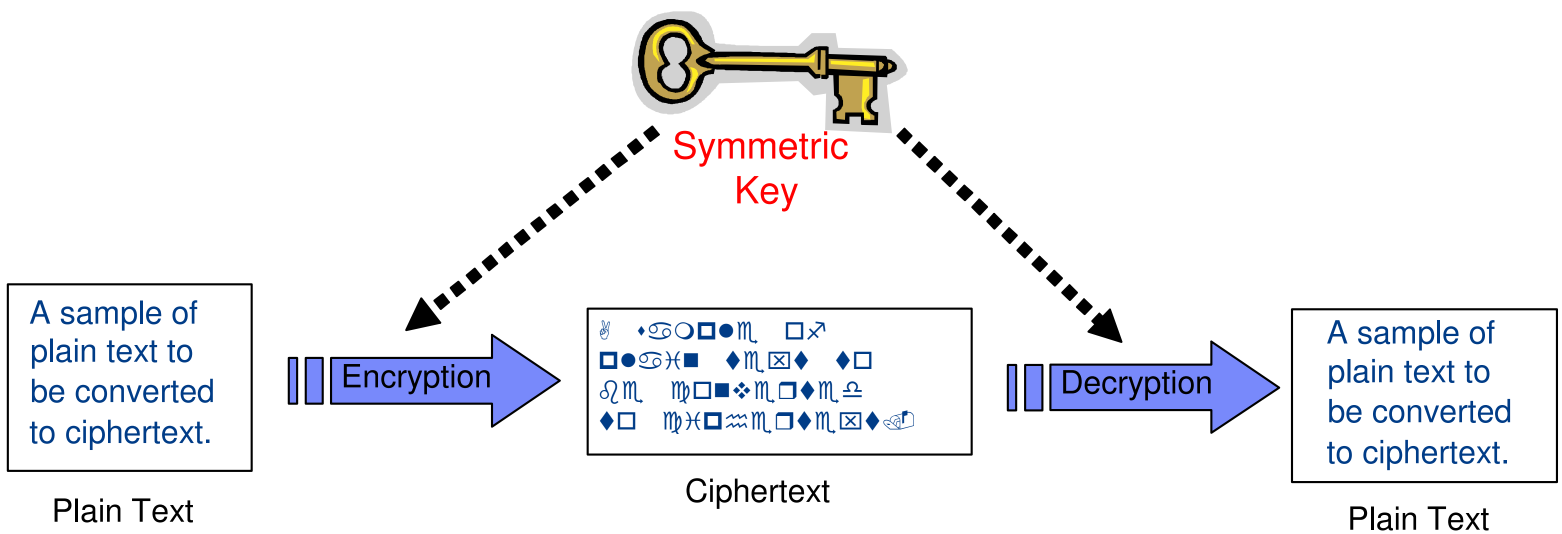
Encryption Techniques Through the Ages

- Simple Substitution (Monoalphabetic)
- Multiple Substitution
- Multiple Ciphers in a Message
- Mechanical Ciphers
- Enigma
- Computers - DES etc.
- Quantum Cryptography

- One Time Pad is best, but impractical

Symmetric Encryption

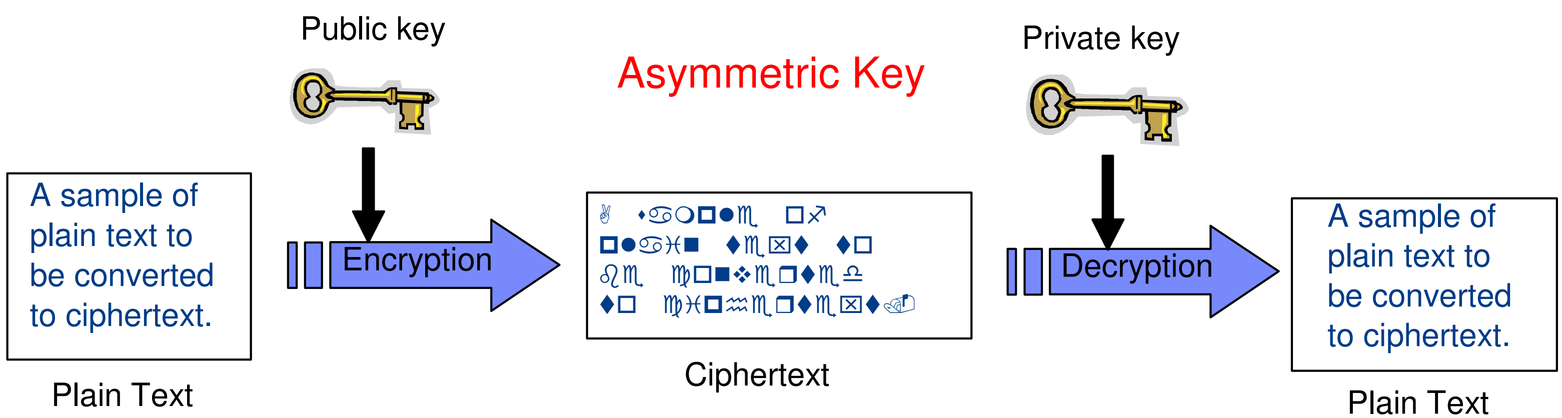
- Symmetric Key - a secret key that is used both to encrypt and to decrypt messages
- Known only by sender and receiver
- Relatively fast and light on CPU power
- Until recently, the ONLY form of encryption
- Challenge : exchanging keys with many people



- Commonly used algorithms are Triple DES and AES
- DES is discouraged - not secure enough

Asymmetric Encryption

- Asymmetric key - public/private key pairs
- Message encrypted with partner's public key and decrypted with your private key
- Slower and more CPU-intensive than symmetric key cryptography
- The private key cannot be derived from the public key
- Either key can undo what the other does
- Used to exchange symmetric keys, and optionally for authentication



- Commonly used algorithms are RSA and Diffie-Hellman

Basic Principles of Cryptography

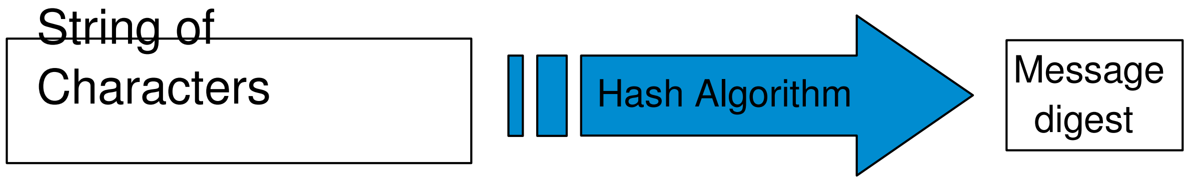
1. Publish the algorithm but protect the keys
   - Remember the voting machine fiasco?
2. Ensure that the algorithm exhibits no patterns
   - Only a brute force attack can break the key
3. Make the key as long as practicable
   - Difficulty of cracking goes up exponentially as key length increases

Security Issues... so far

- Confidentiality - how do I keep anyone from reading my message? YES!
- Integrity - how do I know if anyone has tampered with my message? NOT YET
- Authentication - how do I know that my communication partner is who it claims to be? GETTING THERE
- Non-repudiation - how do I know that the identity of the message sender is authentic? NOT YET

Data Integrity

- How do I know if anyone has looked at my message and changed it?
- Answer : Use a hash algorithm
  - Like a weapons-grade check digit
- A hash algorithm transforms a message into a short string of a fixed length. This string is called a *message digest*.
- Any tampering with the message will result in a different message digest.
- Creating two messages with the same message digest is *extremely* difficult.



But.... the hacker knows the algorithm too!

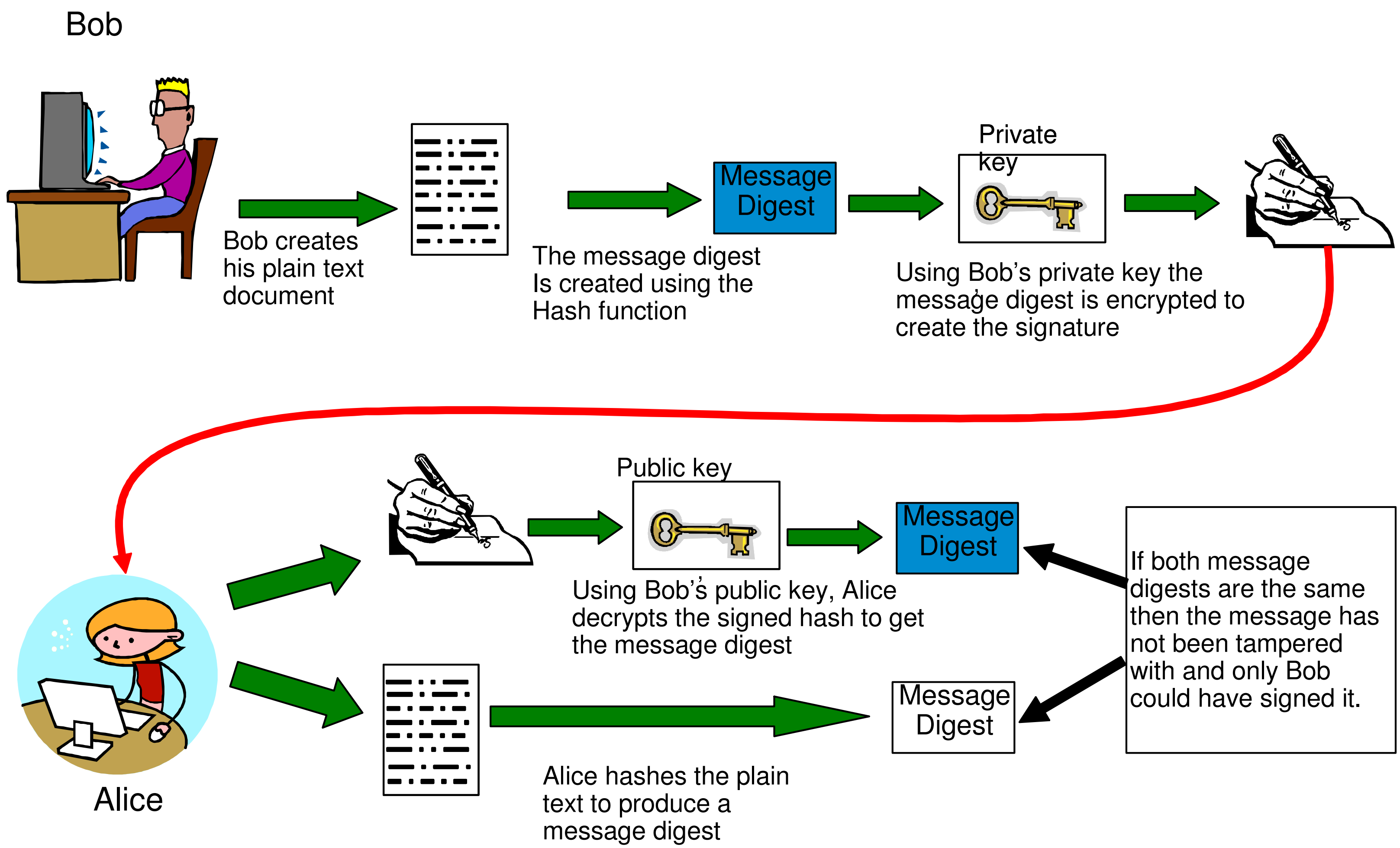
Cryptographic Hash Algorithms

- Answer: Encrypt the message digest!
  - With a shared symmetric key (during communication)
    - In practice, do the digest over the message plus the key
  - With your own private key (long term)
    - This is known as a ***digital signature.***
- Cryptographic Hash algorithms include:
  - MD4, MD5 (no longer considered secure)
  - SHA-1, SHA-128, SHA-256, SHA-512
    - Use the later ones, SHA-1 is suspect

Data integrity!

Non-repudiation!

Digital Signature



Impersonation / Authentication, revisited

- How do I REALLY know who the party on the other end is?
- By the use of digital signatures
  - Provide the ability to authenticate who sent the message
  - Incorporate the use of Asymmetric keys and cryptographic hash functions
  - The signature is encrypted with the sender's private key
  - If the sender's public key can decrypt the signature then the sender must be authentic.
- Final problem: How to ensure that this public key belongs to this sender?

Digital Certificates

- Digital Certificates address the authentication problem.
- A Digital Certificate can be thought of as an electronic identity card that establishes your credentials (authenticates you) when communicating securely.
- A Digital Certificate contains:
  - Your name
  - A serial number
  - Start and expiry dates
  - Your public key
  - And a digital signature, to verify that this public key belongs to this entity.

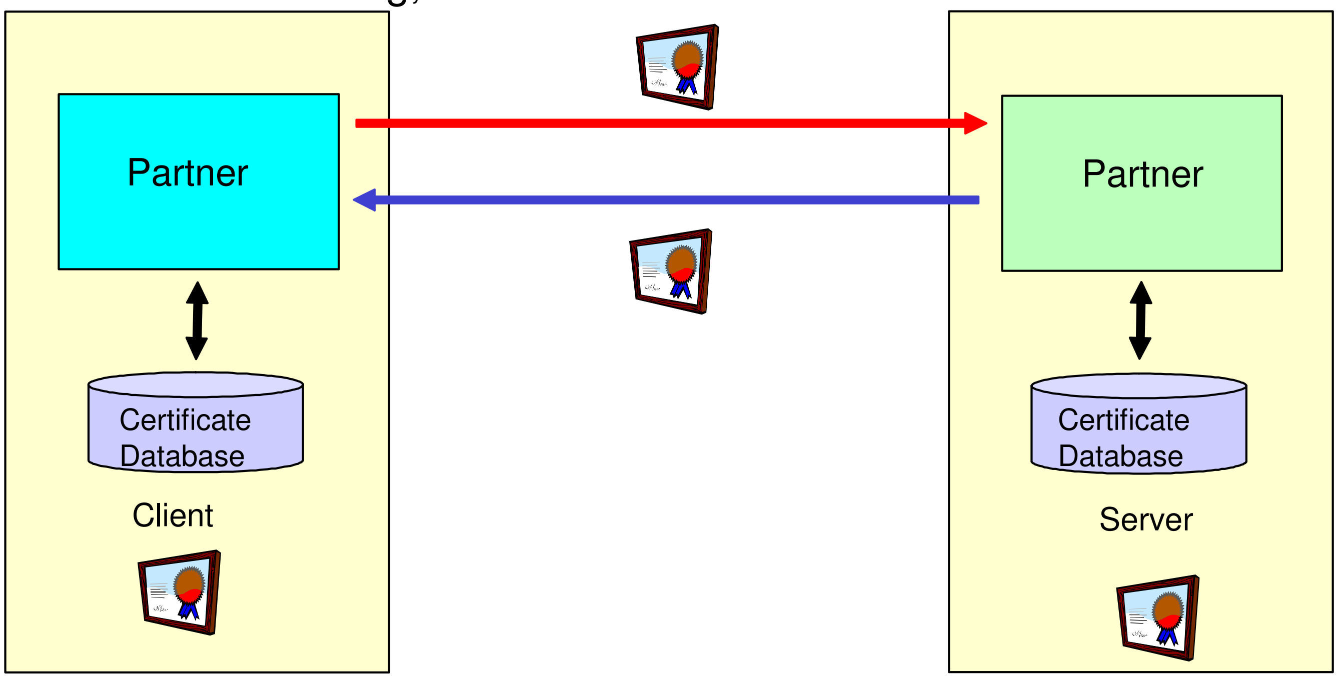
But whose signature?

Where do certificates come from?

- Ultimately, the only way to be sure of authenticity is a ***physical*** exchange of digital certificates.
  - Any attempt at exchanging them over a network is doomed unless it is done securely.
  - But to do it securely you need a digital certificate to start with....

Self Signed Certificate (the poor man's method)

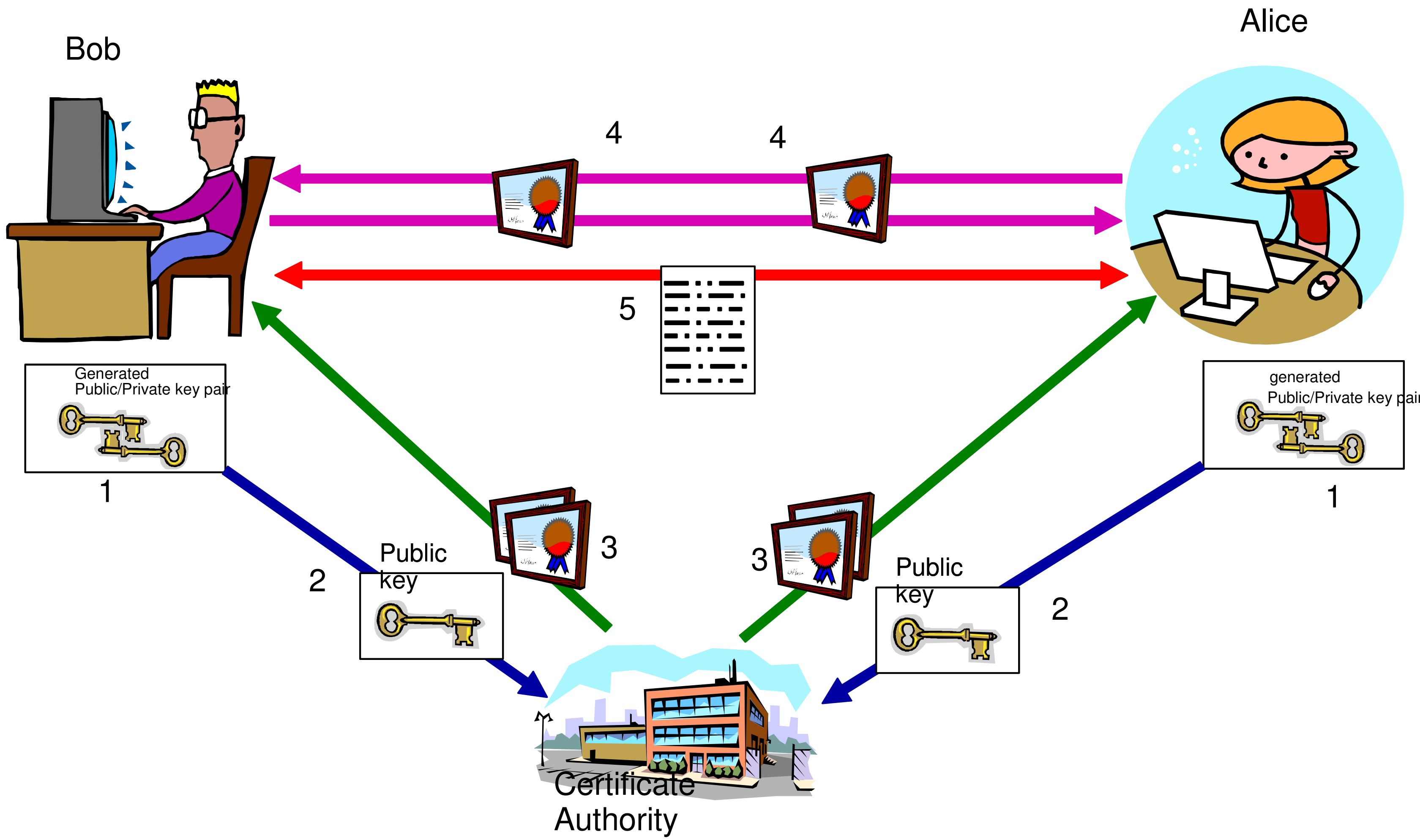
- Minimum requirement :
  - At least one end must have a copy of the other's certificate.
  - Often, both ends must have each other's certificates.
  - They must be exchanged securely (physically, or over a controlled Intranet connection)
  - Suitable for testing, but **NOT** for Internet communication



CA Certificates

- A CA can be thought of as a certificate distributor.
- To obtain your personal certificate, you must send your information to a CA:
  - Create an Asymmetric public/private key pair.
  - Send your identity with your public key to the CA.
  - Wait for them to check you out and (maybe) pay them a large sum of money.
- The CA sends you back your certificate, signed by them.
- Now, if your communication partner trusts the CA then you are authenticated.

Using CA Certificates

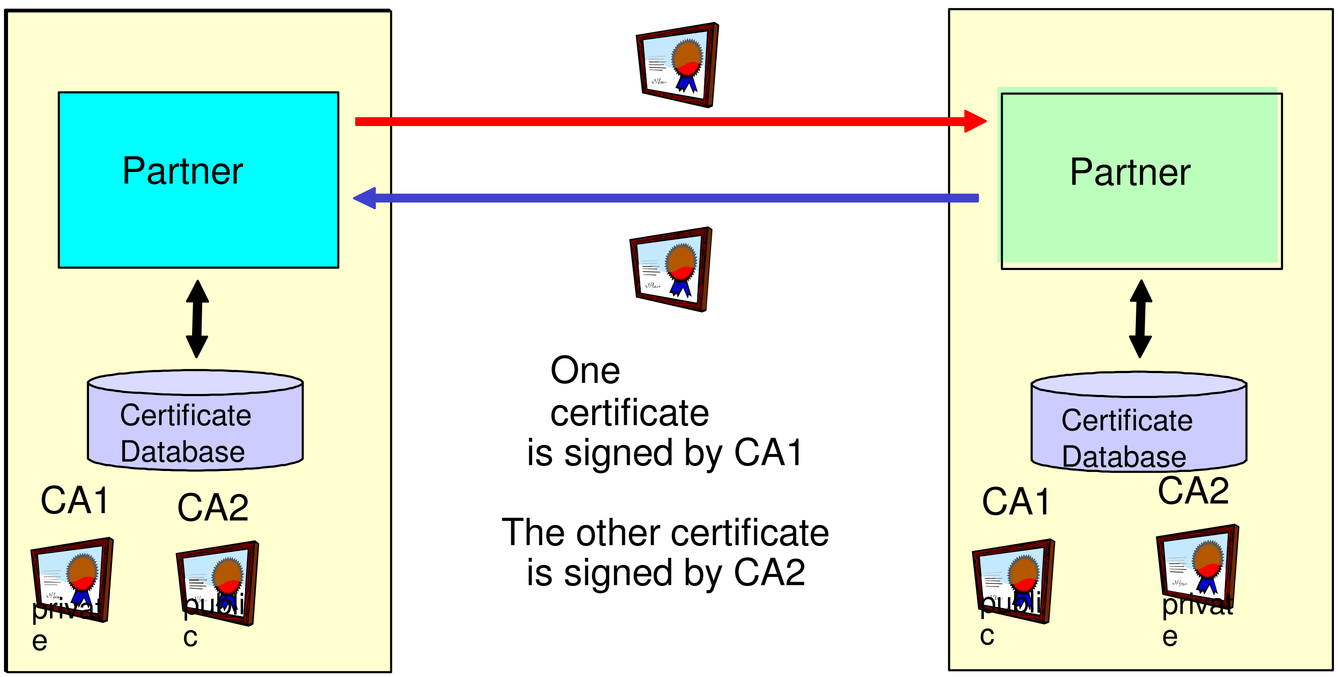


The Final Step

- If you trust the small group of CAs then you can authenticate any partner down the chain of trust.
- You need the CA's self-signed certificate, which contains their all-important public key.
- How can you be sure that the CA certificate is authentic?
- Ultimately, physical receipt of the certificate is necessary.
  - They are delivered with the machine!
  - With the operating system (z/OS - RACF)
  - With the application
    - Web browser (Internet Explorer, Firefox)
    - TN3270 client (PComm)
    - Web server (IHS, WAS...)
- You can verify the message digest of the CA (root) certificate on the CA's web site.

Certificate Management - CA Certificates

Before any communication can take place, at least one end must already have a copy of the CA certificate that signed the other end's personal certificate.



Certificate Management

- Certificates are used to authenticate the partner and (with the embedded public key) to encrypt and exchange symmetric keys.
- Three types of certificate may be found in a certificate database:
  1. CA certificates (for verifying partners' personal certificates)
  2. Personal certificates (to identify you, signed by your preferred CA)
  3. Self-signed certificates (to identify you, but not so secure)
- Certificates have a life span
  - If a certificate has expired, it should no longer be trusted.
  - If on the Internet, it MUST not be trusted.
- Certificates can be revoked
  - If the private key has been compromised, for example
  - The issuer places the revoked certificate on an LDAP server (the CRL - certificate revocation list)
  - The partner attempting authentication SHOULD check the CRL
  - Most often, this is not done.