Practical Cryptography

Handout 7 – Public Key Distribution

Kasun de Zoysa
kasun@ucsc.cmb.ac.lk

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

Asymmetric Key / Public Key Cryptosystem

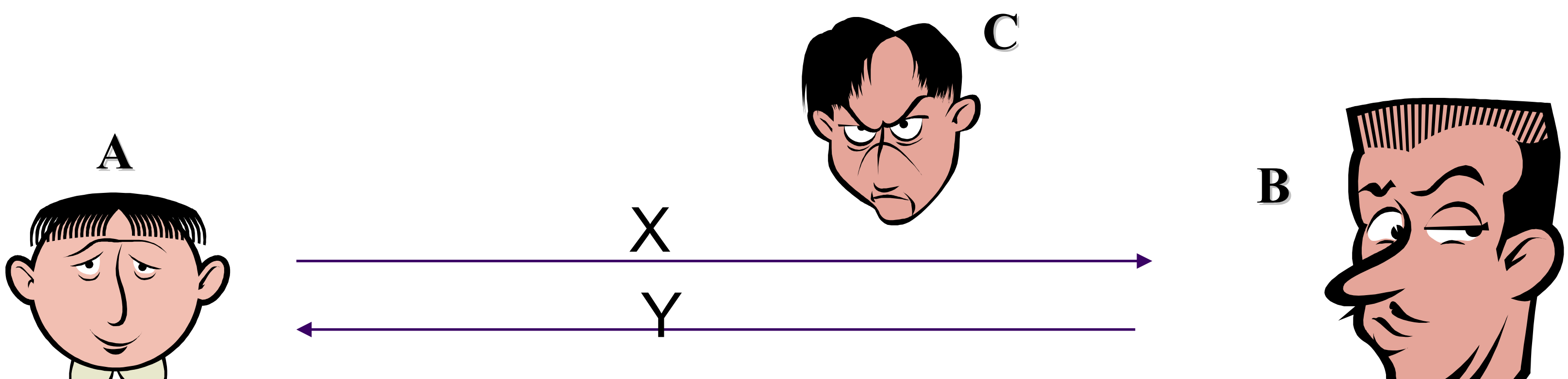
- Uses a Key Pair (Public / Private Keys) *
- Public Key shared between users
  - Strengths
    - Better Scalability than Symmetric Key Cryptosystems
    - Can provide confidentiality, authentication and nonrepudiation
    - Key Distribution Management
    - Uses one Key to encrypt, the other to decrypt
  - Weaknesses
    - Slower Algorithms than Symmetric Key System
  - Algorithms
    - RSA, Elliptic Curve Cryptosystem (ECC), Diffie-Hellman
    - DSS (Digital Signature Standard), PGP
    - ECC has higher work factor than other asymmetric algorithms

Key Distribution

- symmetric encryption schemes require both parties to share a common secret key
- issue is how to securely distribute this key without revealing it to an adversary
- many attacks are based on poor key management and distribution rather than breaking the codes
- This is, actually, the most difficult problem in developing secure systems

Diffie-Hellman Key Agreement

- **Published in 1976**
- **Based on difficulty of calculating discrete logarithm in a finite field**
- **Two parties agreed on two large numbers n and g, such that g is a prime with respect to n**



$X = g^{x} \bmod n$

$k = Y^{x} \bmod n = g^{xy} \bmod n$

$Y = g^{y} \bmod n$

$k = X^{y} \bmod n = g^{xy} \bmod n$

Possible to do man in the middle attack

Session Key / Master Key

- The idea of having a key-encryption-key (master key) to generate random and temporary session keys
- can be implemented in several ways
- Basic D-H is such an example
- public/private keys are master keys, exchanged key is a session key

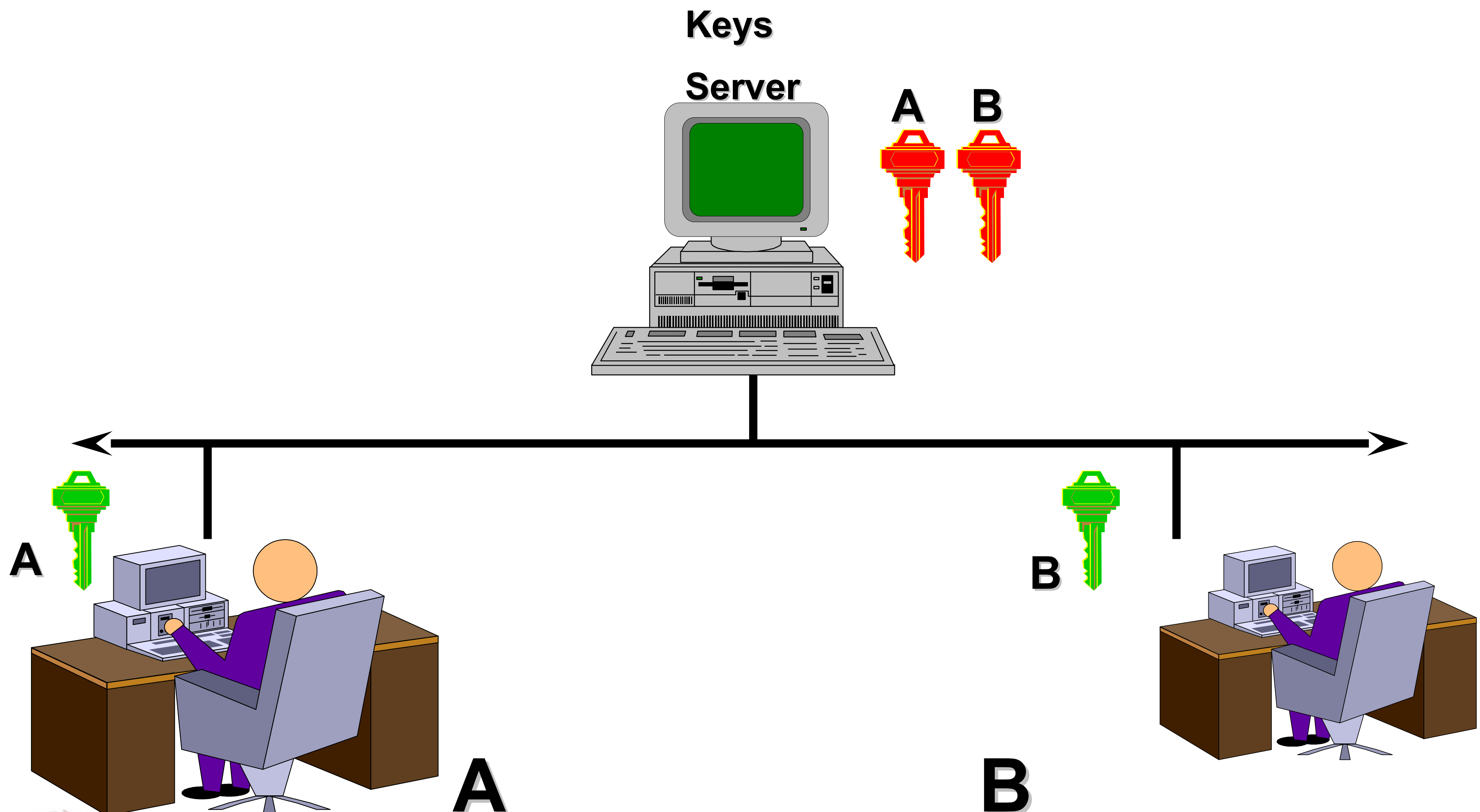
Hybrid Encryption

- Why is symmetric key encryption still used?
  - Performance
  - Also cryptographic reasons

In practice one uses hybrid encryption...

- A one-time random key is generated (“session key”)
- This is used to symmetrically encrypt the message
- The symmetric session key is encrypted through public key encryption and sent to the other party together with the (encrypted) message

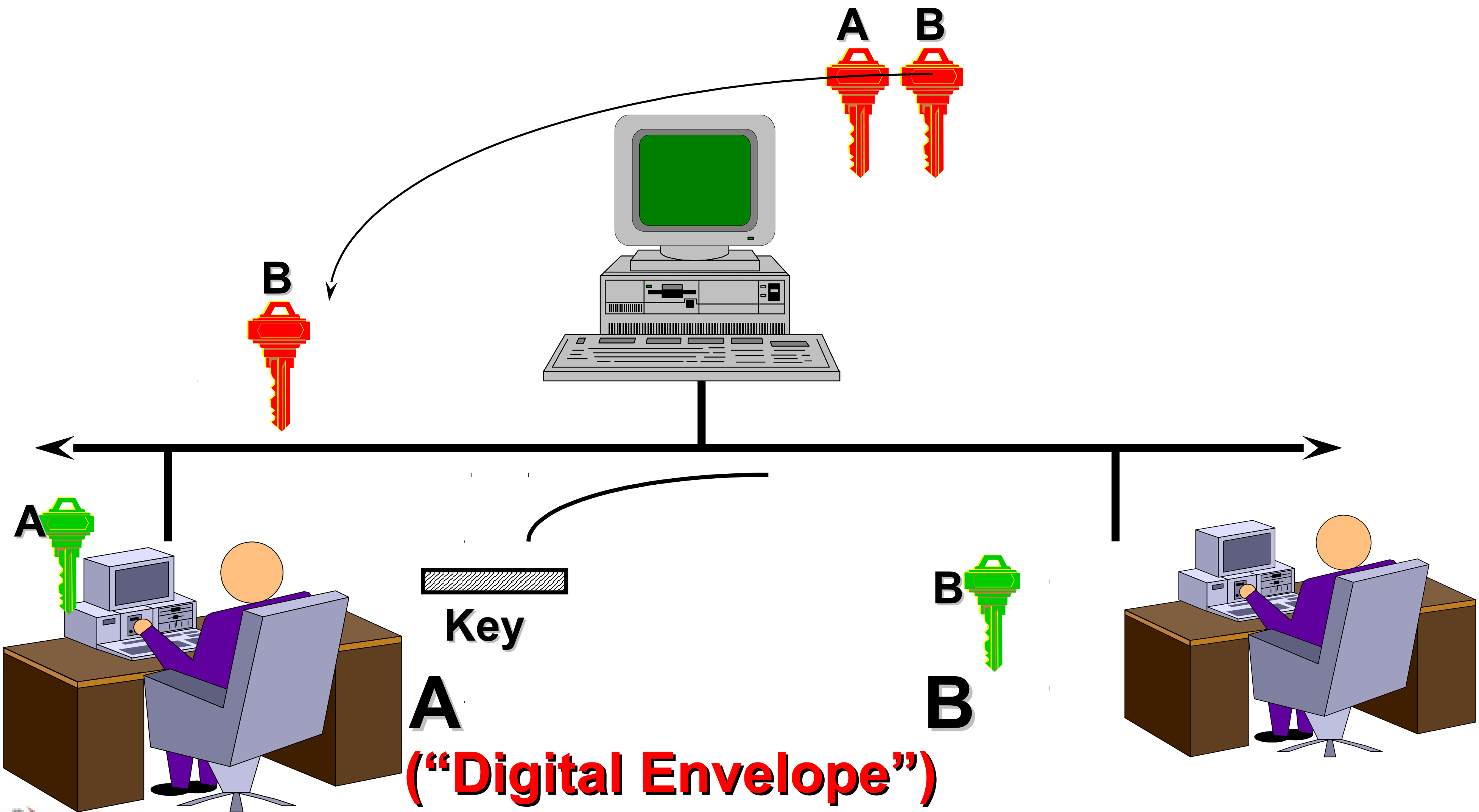
Storage and Handling Public Keys



Key Management

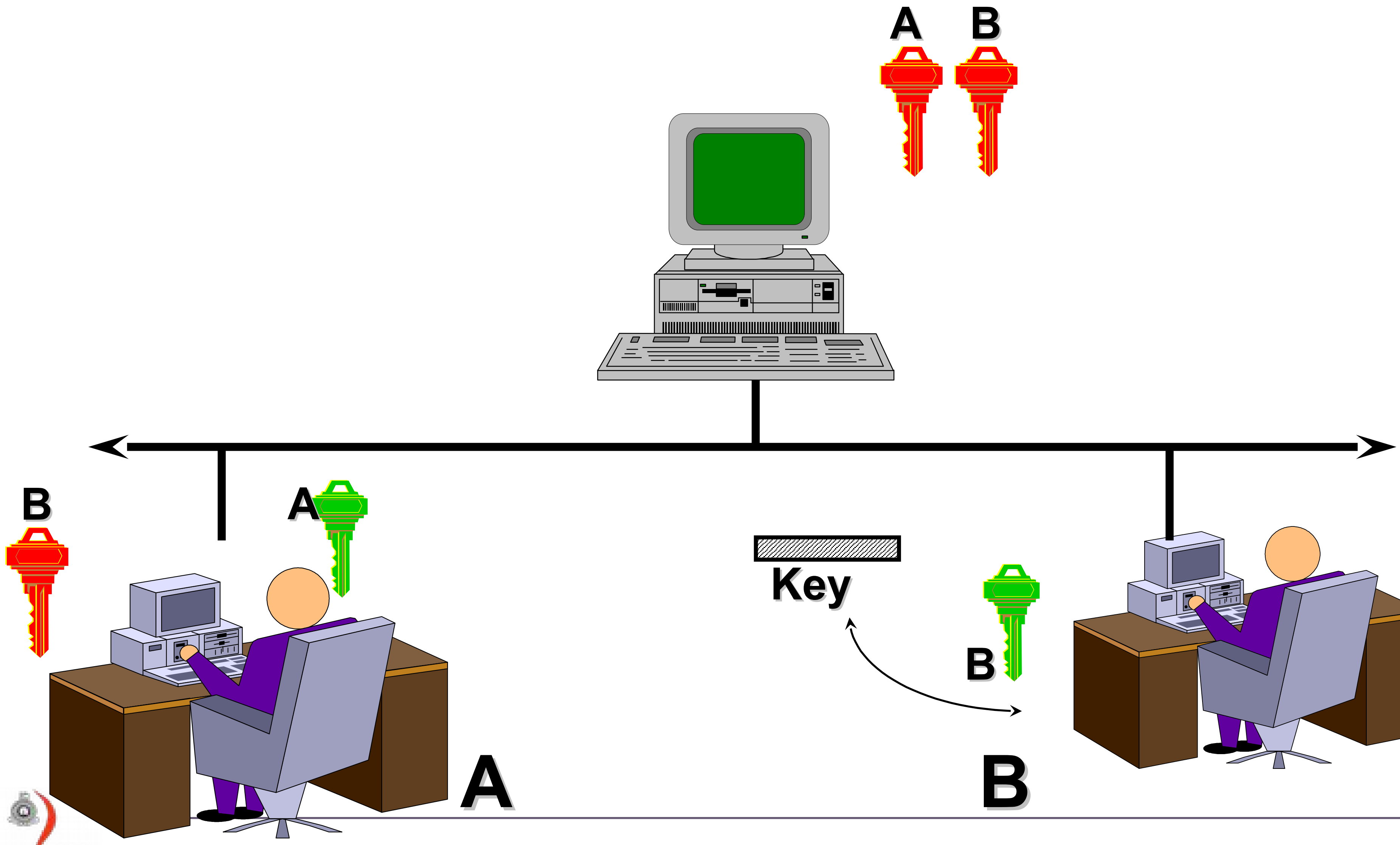
- Using a public key system, A wants to talk to B
- C is the Key Distribution Center(Key Server), has A and B's public key
- A calls C and obtains the Public key of B
- A encrypts a session key, "k", with the public key and sends the encrypted "k" to B
- A and B can then communicate

Secure Sending of secret key

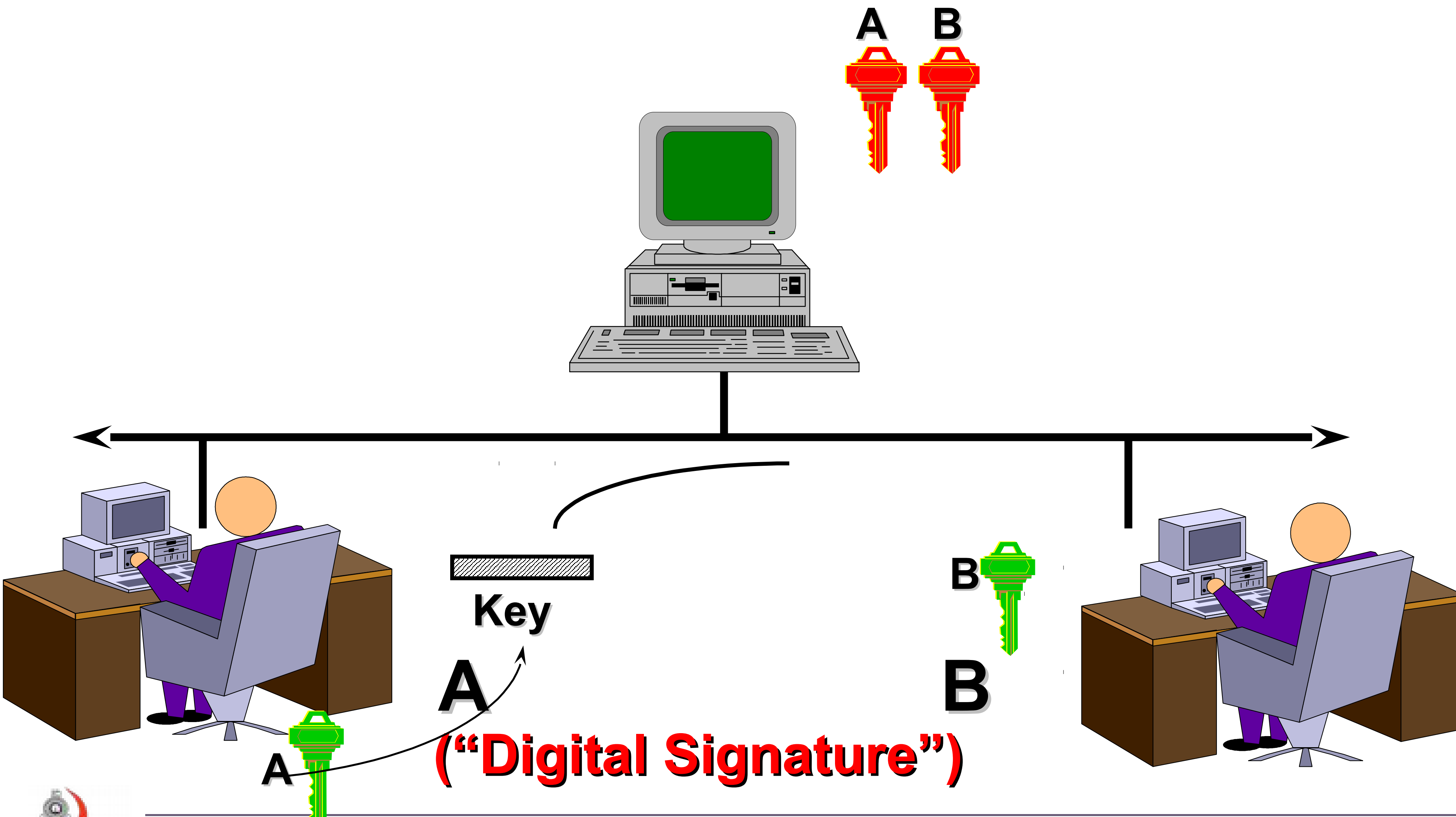


("Digital Envelope")

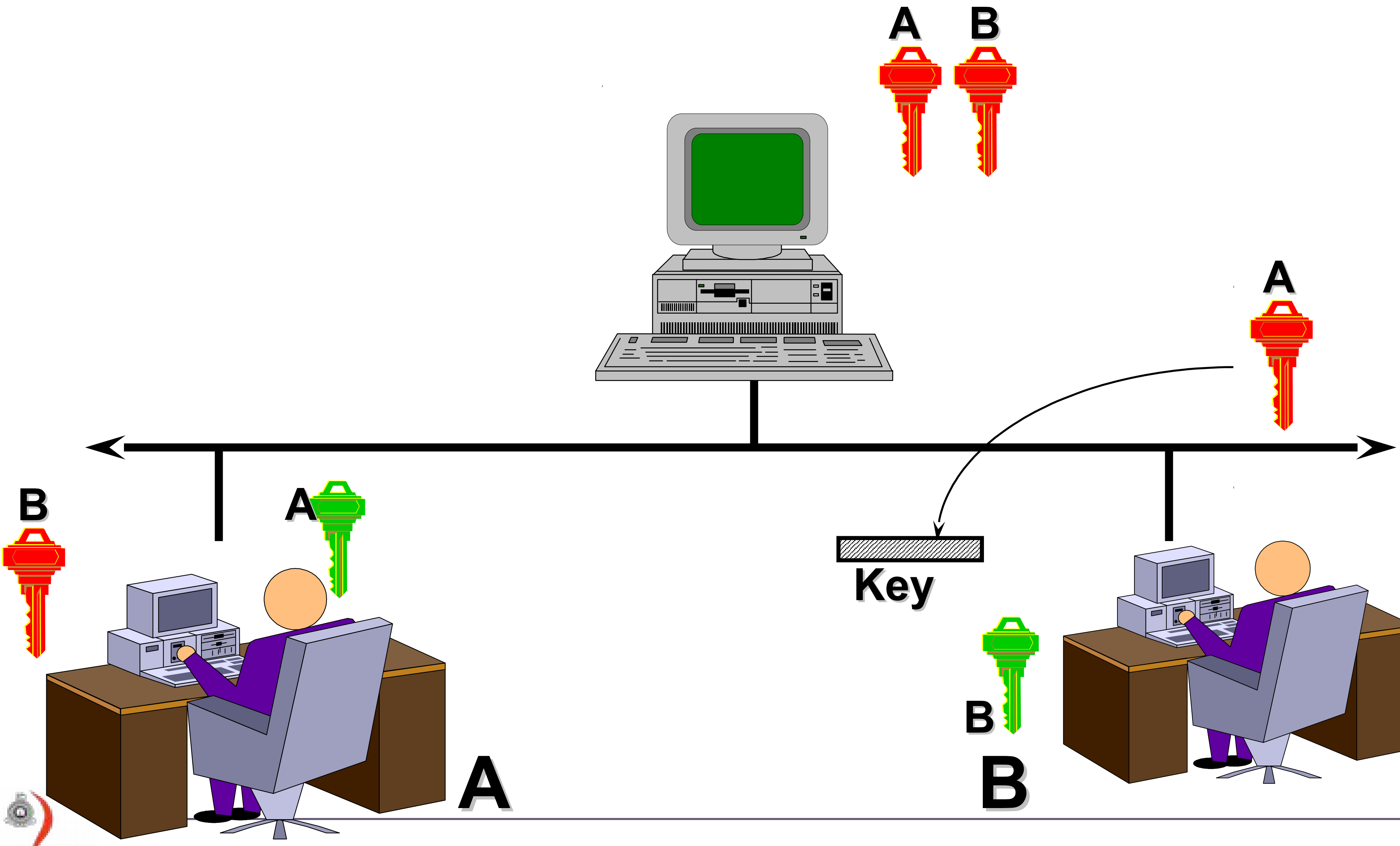
Recovery of Secret Key



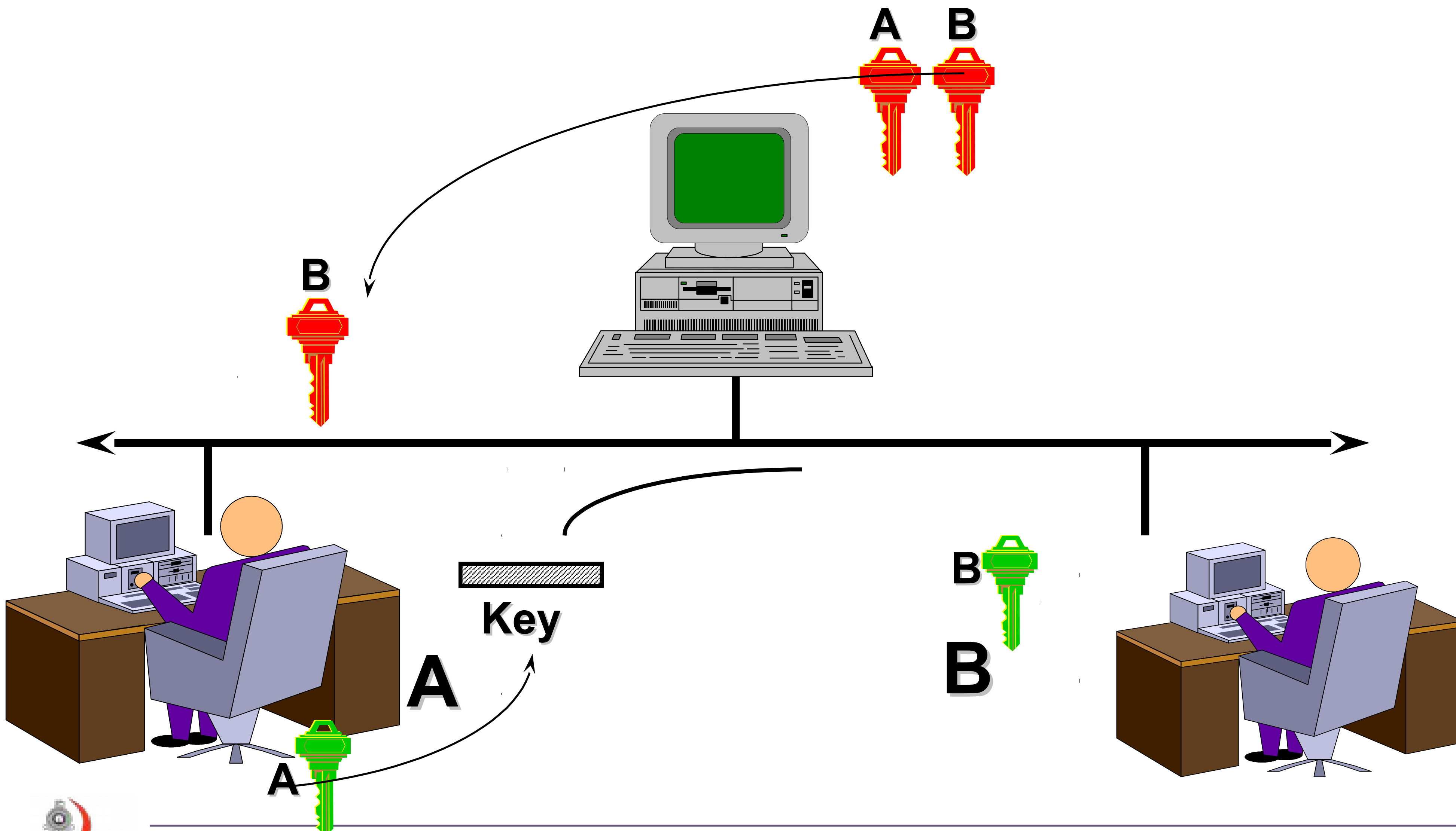
Authenticity of Sender



Verification of Signature

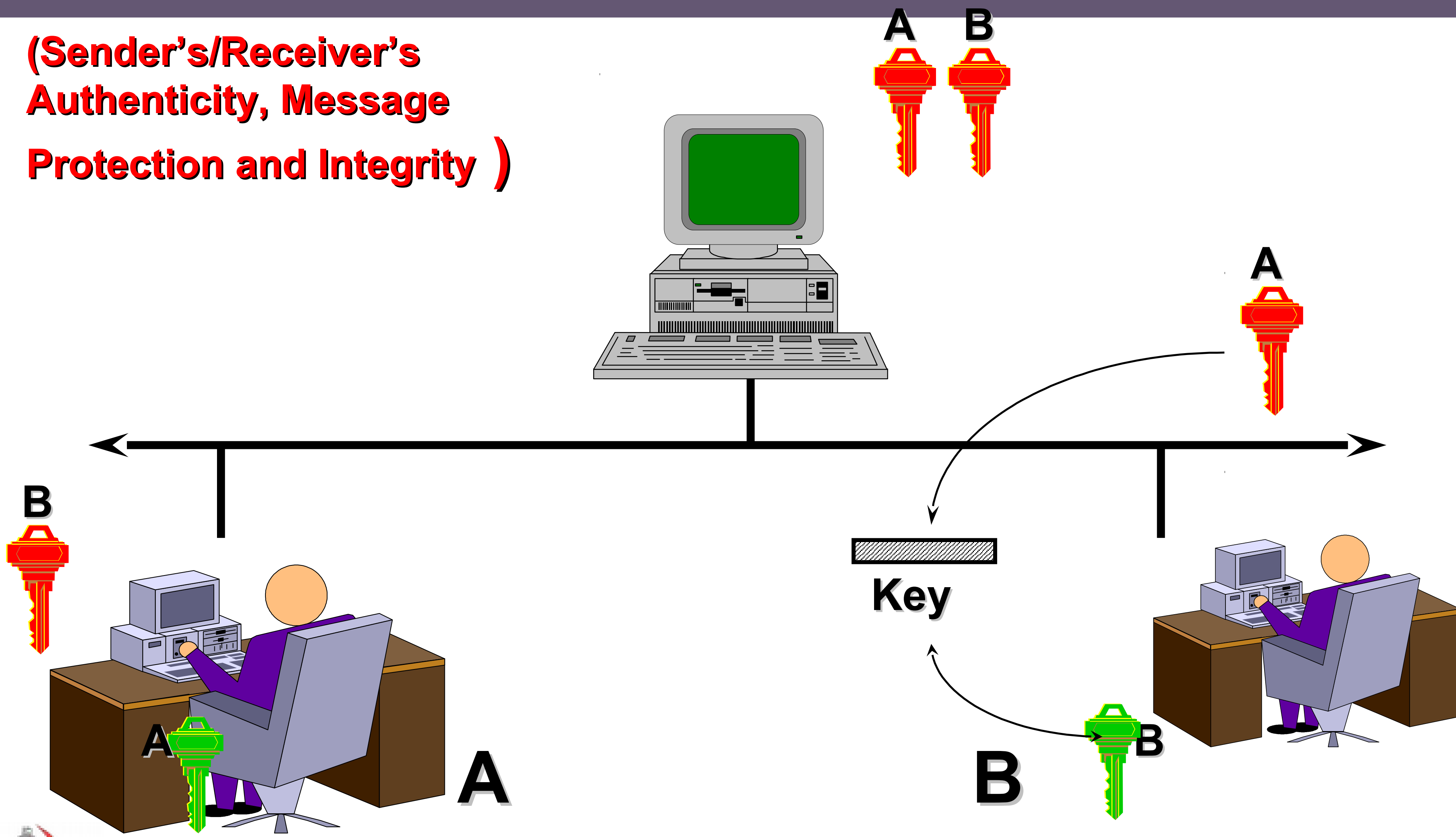


Authenticity of Sender and Receiver



Full Verification

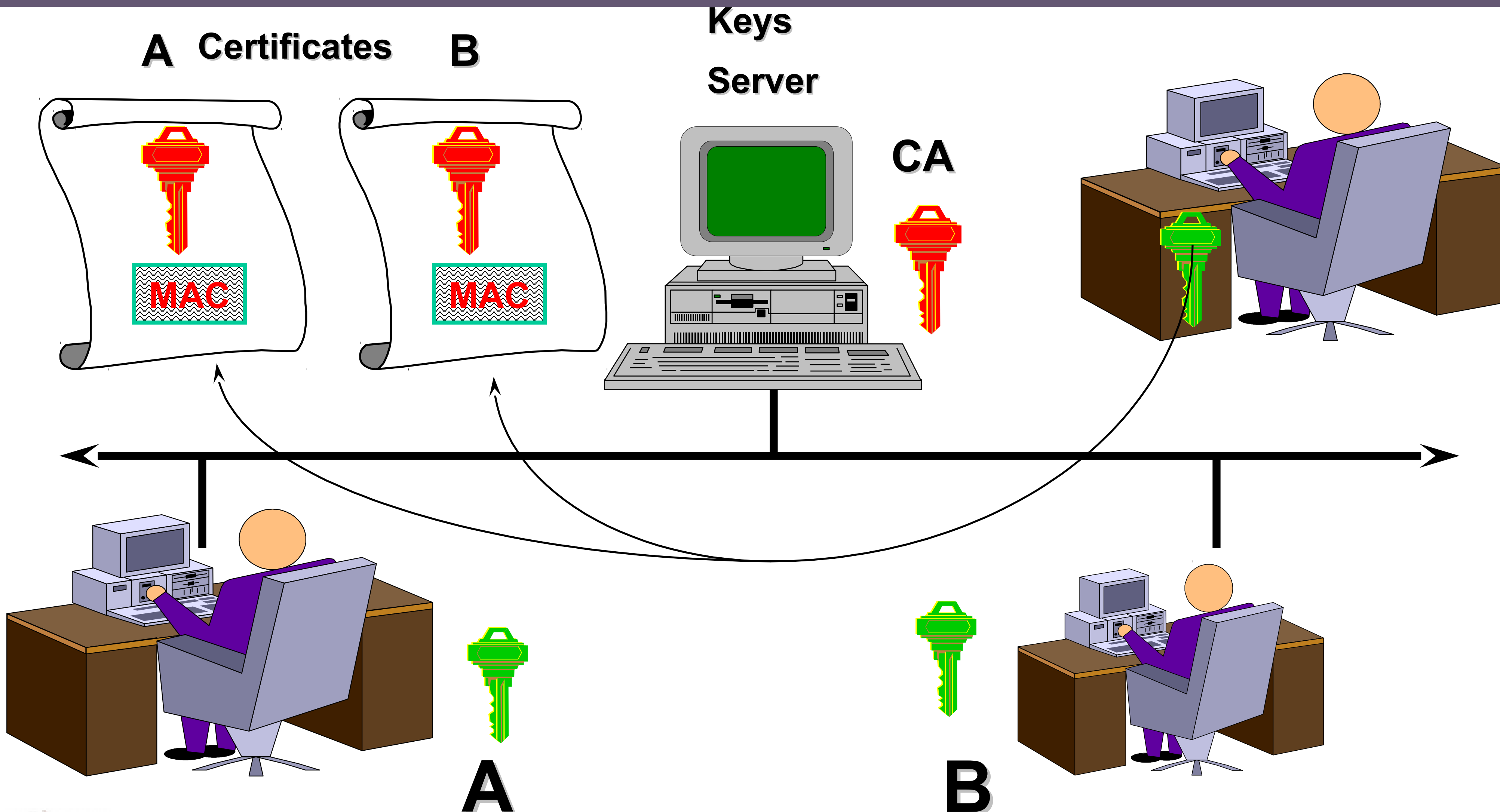
(Sender's/Receiver's Authenticity, Message Protection and Integrity)



Distribution of the Public Keys

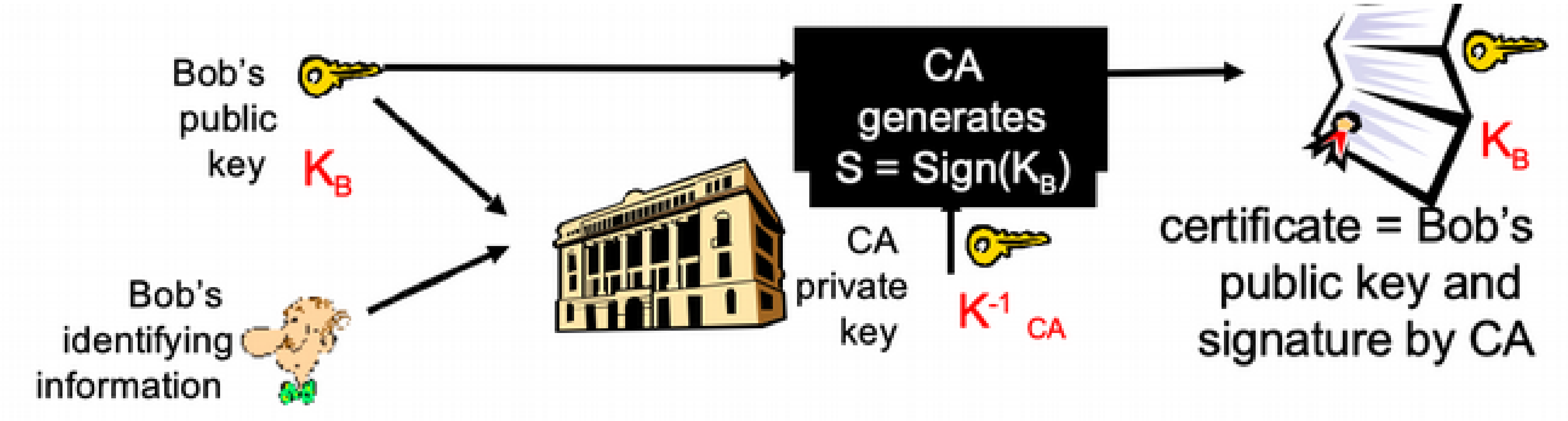
- use of Public Key Cryptography to distribute secret keys
- public/private key as a master key
- **the most important barrier against the deployment of PKC in applications is distribution of public keys**
- how can I make sure about the legitimacy of a public key?
- how can I make sure that Bob's public key really belongs to Bob, not to Charlie?

Certificate Authority



Certification Authorities

- Certification authority (CA): binds public key to particular entity,A .
- An entity A registers its public key with CA.
- A provides “proof of identity” to CA.
- CA creates certificate binding A to its public key.
- Certificate contains A’s public key AND the CA’s signature of E’s public key.

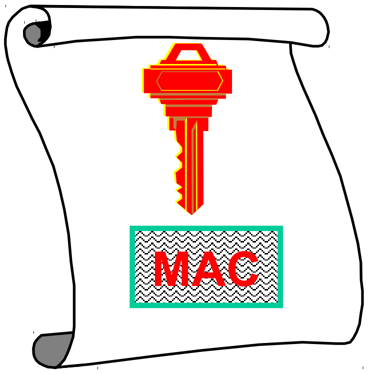


Public Key Certificates

- certificates allow public key exchange without real-time access to public-key authority
- a certificate binds identity to public key usually with other info such as period of validity, issuer's info, cryptographic details, etc
- all contents signed by a trusted Certification Authority (CA)
- can be verified by anyone who knows the **CA's public-key**
- CA must make sure about the identity of the certificate owner

Internal Structure of Certificate

- Version
- Serial Number
- Signature Algorithm
- Issuer
- Subject
- Validity
- Subject Public Key Information
- Extensions
- Signature

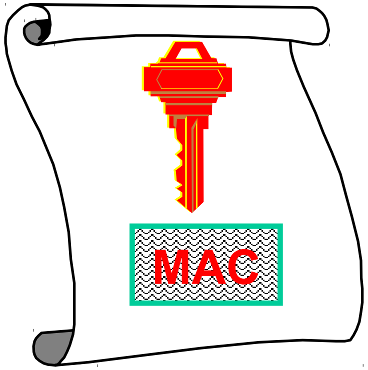


Structure of Distinguish Name

- Country Name
- State and Province Name
- Locality Name
- Organization Name
- Organization Unit Name
- Common Name
- Email Address
- URL

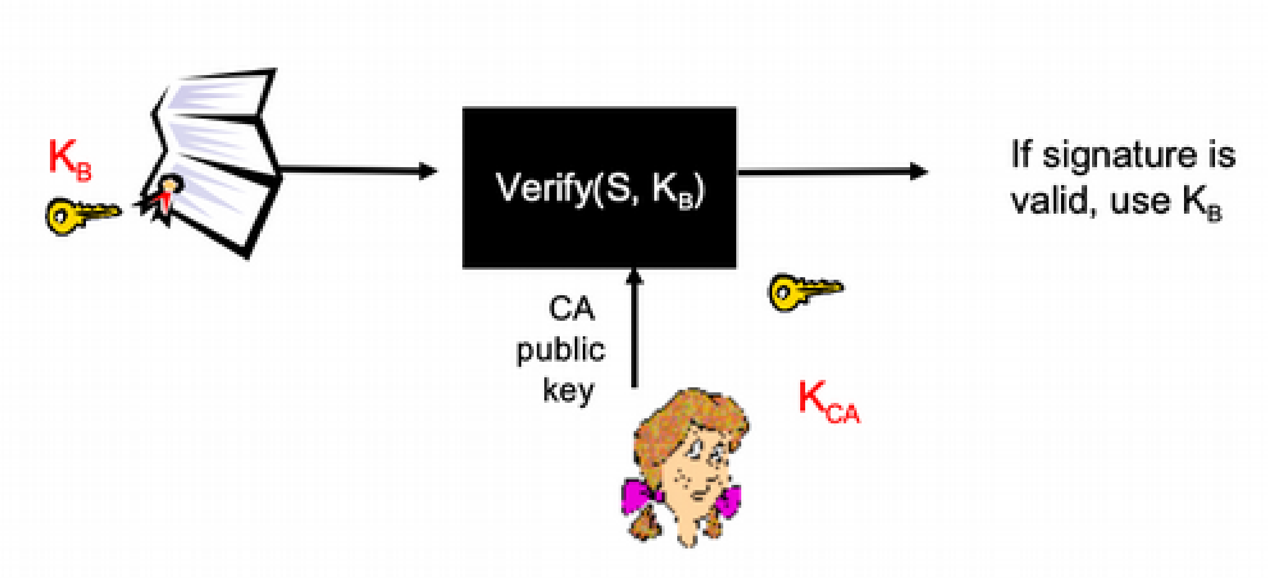
Certificate Types

- Digital Signature
- Key Encipherment
- Data Encipherment
- Key Certificate Signature
- CRL Signature
- Object Signing



Certification Authorities

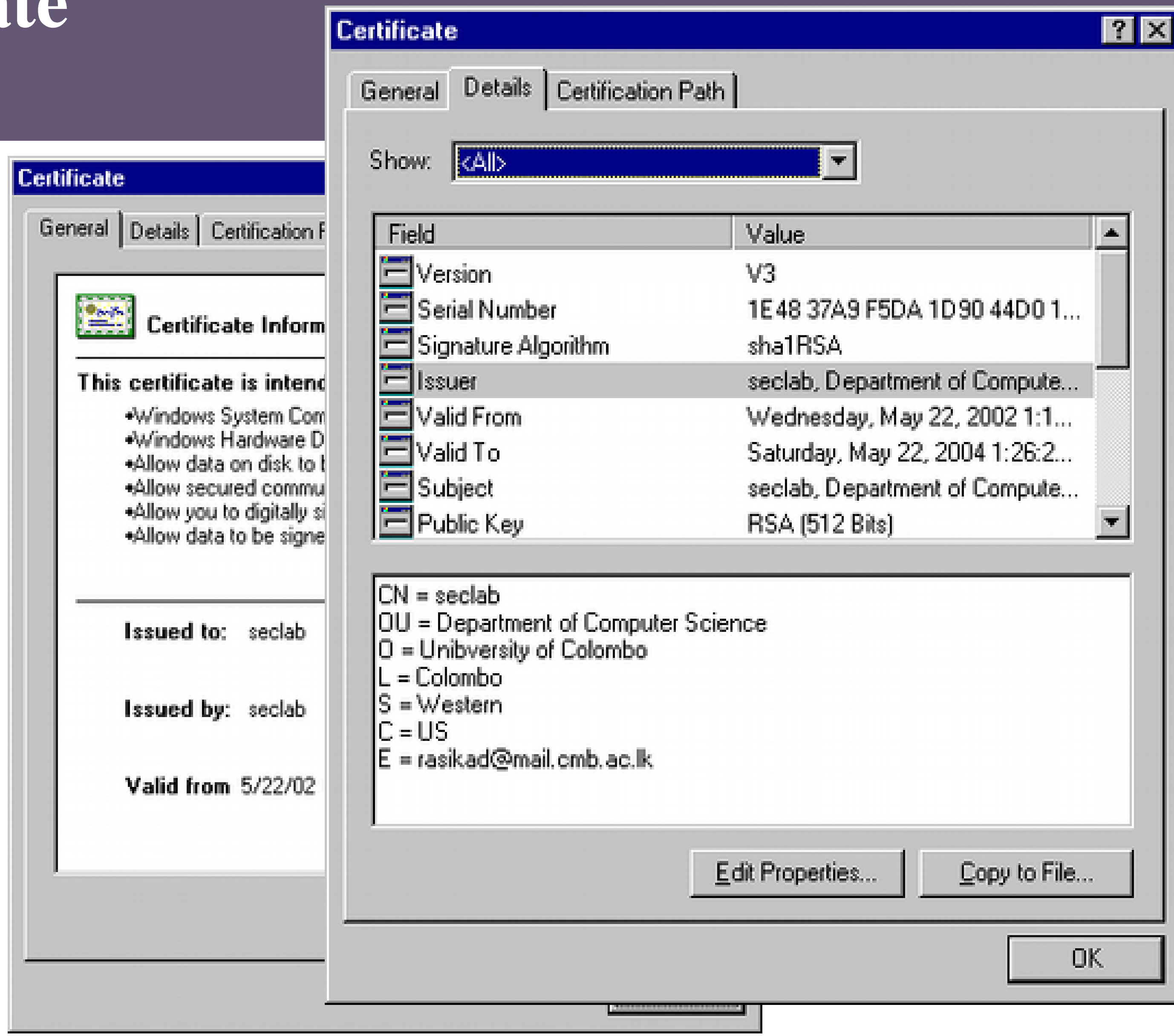
- When Alice wants Bob’s public key:
- Gets Bob’s certificate (from Bob or elsewhere).
- Use CA’s public key to verify the signature within Bob’s certificate, then accepts public key



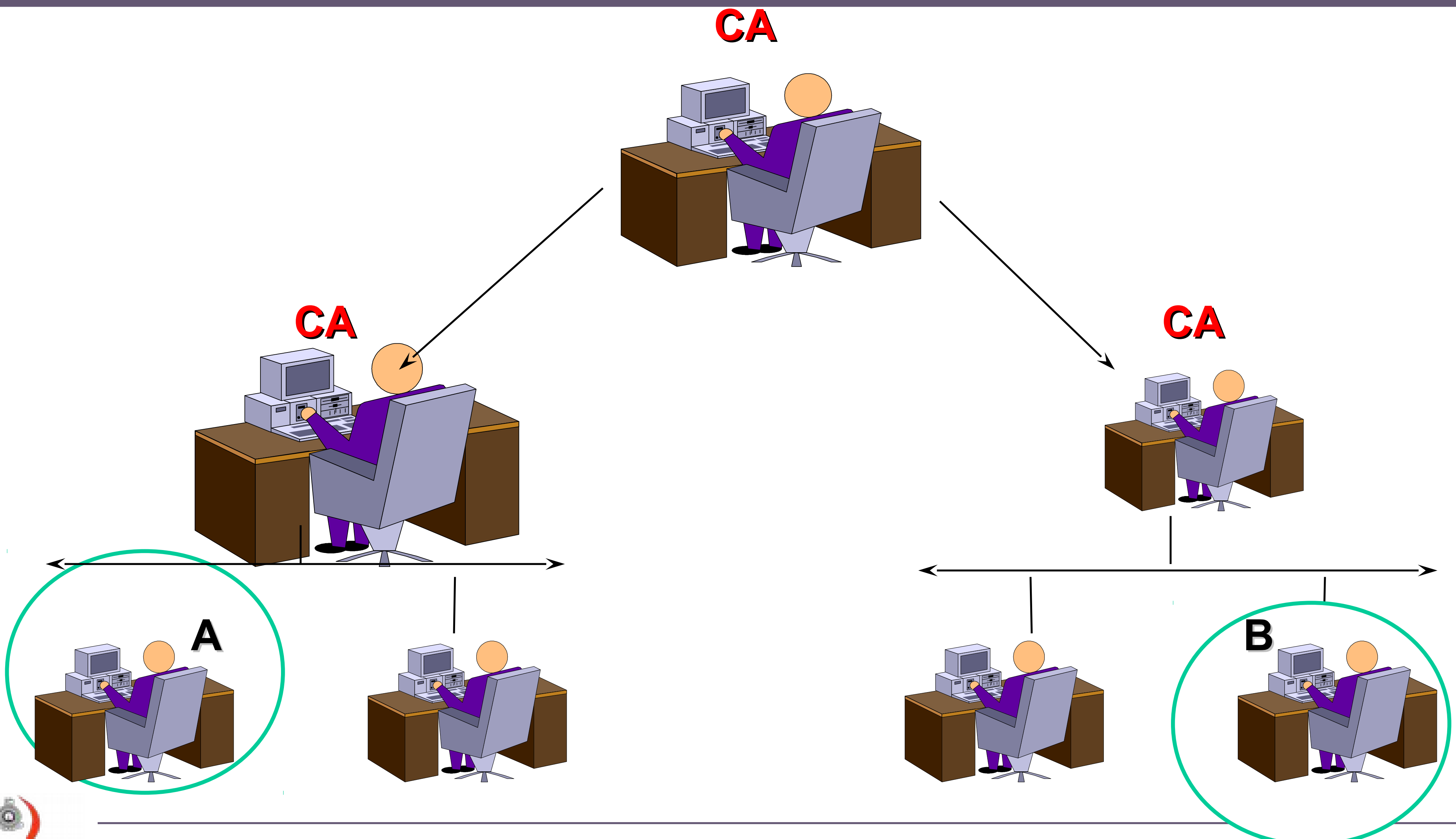
Public-Key Certificates

- Certificates are widely used in practice
- is a single CA sufficient?
- what happens if the CA’s public key is not known?
- **how to distribute CA public keys?**
- what happens if a certificate is revoked?
- How the users exchange their certificates in practical applications?

Root Certificate

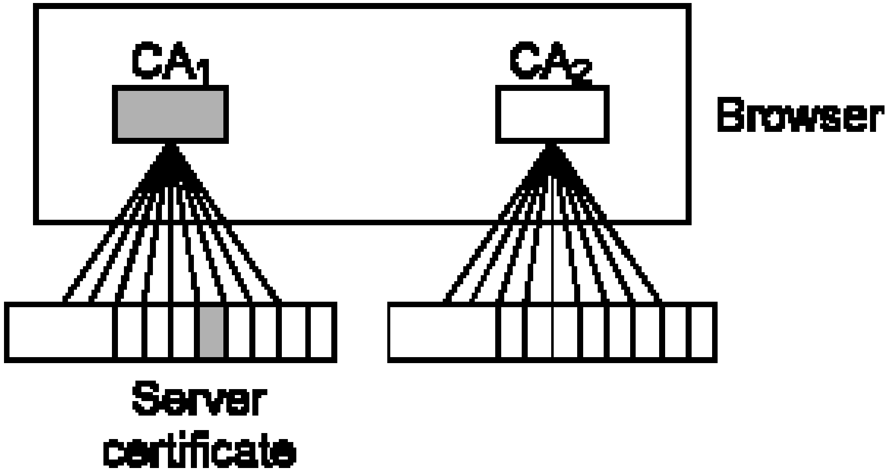


Certificate Hierarchy



CA Hierarchy in Practice

Flat or Clayton's hierarchy

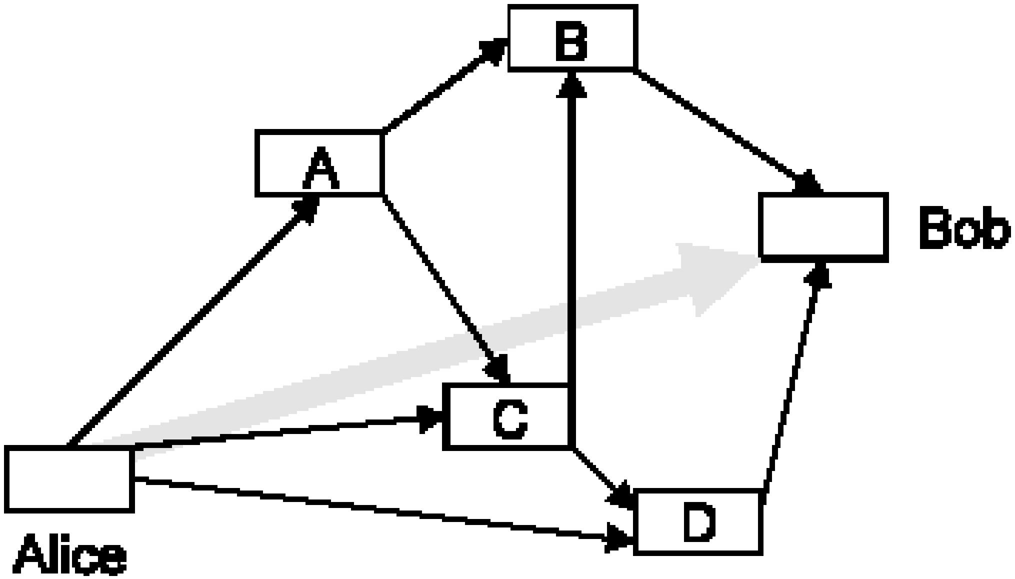


CA certificates are hard-coded into web browsers or email software

- Later software added the ability to add new CAs to the hardcoded initial set

Alternative Trust Hierarchies

PGP web of trust

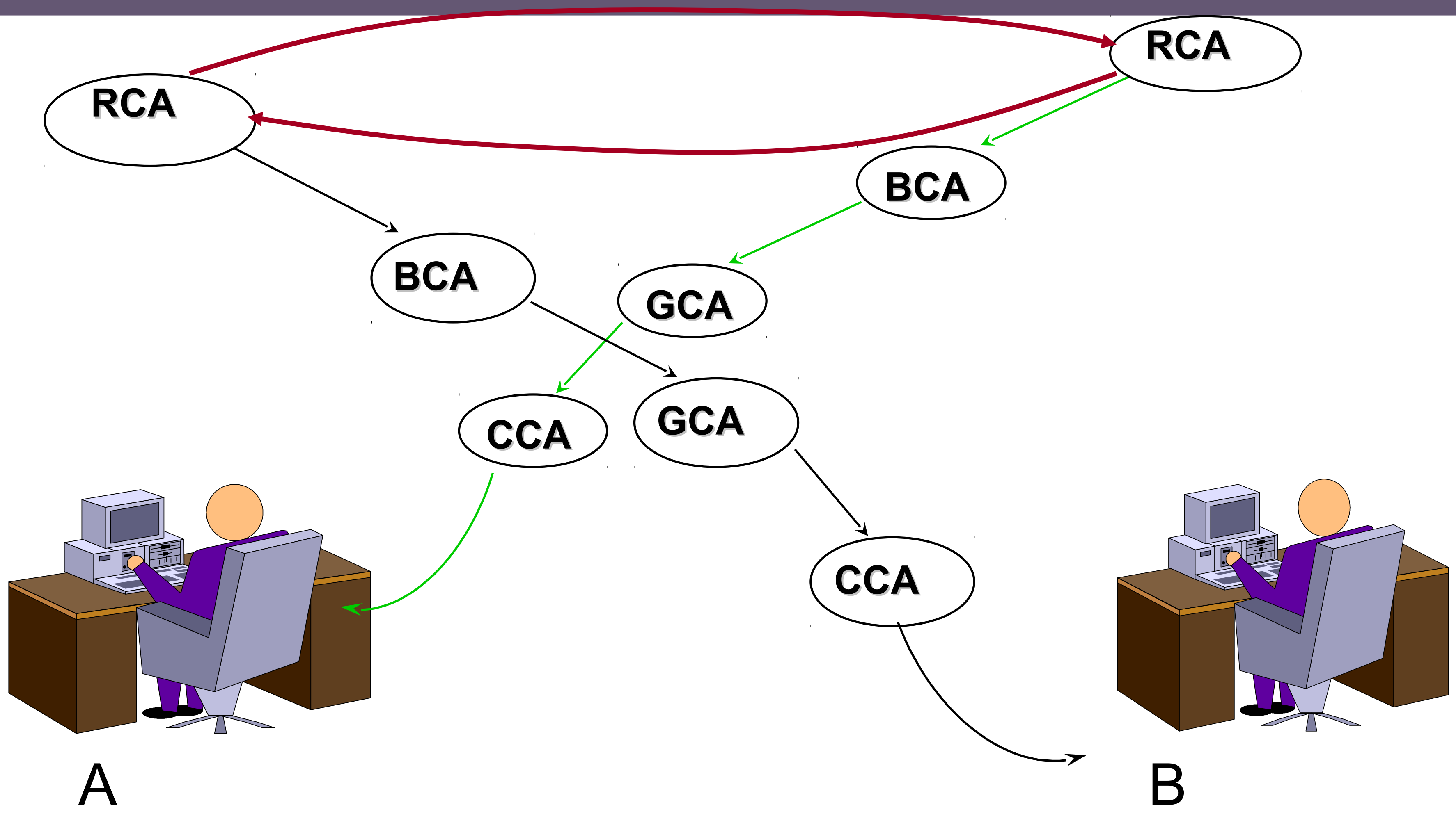


Bob knows B and D who know A and C who know Alice

$\Rightarrow$ Bob knows the key came from Alice

Web of trust more closely reflects real-life trust models

Cross Certification



Certificate Revocation

•Revocation is managed with a Certificate Revocation List (CRL), a form of anti-certificate which cancels a certificate

• Equivalent to 1970s-era credit card blacklist booklets

• Relying parties are expected to check CRLs before using a certificate

– *“This certificate is valid unless you hear somewhere that it isn’t”*

CRL Distribution Problems

• CRLs have a fixed validity period
– Valid from *issue date* to *expiry date*
• At *expiry date*, all relying parties connect to the CA to fetch the new CRL
– Massive peak loads when a CRL expires (DDOS attack)
• Issuing CRLs to provide timely revocation exacerbates the problem
– 10M clients download a 1MB CRL issued once a minute = ~150GB/s traffic
– Even per-minute CRLs aren't timely enough for high-value transactions with interest calculated by the minute

Online Status Checking

•Online Certificate Status Protocol, **OCSP**

• Inquires of the issuing CA whether a given certificate is still valid

  – Acts as a simple responder for querying CRL's

  – Still requires the use of a CRL to check validity

• OCSP acts as a selective CRL protocol

– Standard CRL process: "Send me a CRL for everything you've got"

– OCSP process: "Send me a pseudo-CRL/OCSP response for only these certs"

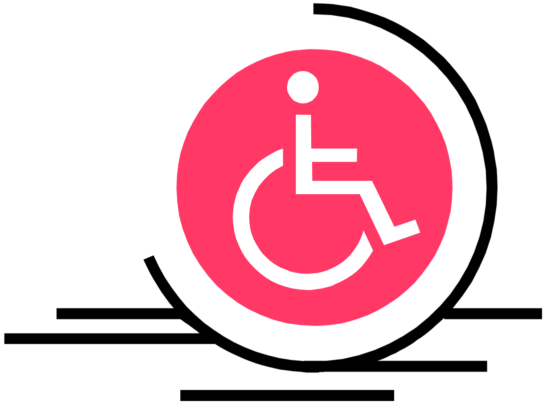
– Lightweight pseudo-CRL avoids CRL size problems

– Reply is created on the spot in response to the request

– Ephemeral pseudo-CRL avoids CRL validity period problems

Online Certificate Status Protocol (OCSP)

• Returned status values are non-orthogonal
– Status = “good”, “revoked”, or “unknown”
– “Not revoked” doesn’t necessarily mean “good”
– “Unknown” could be anything from “Certificate was never issued” to “It was issued but I can’t find a CRL for it”



OCSP Problems

•Problems are due in some extent to the CRL-based origins of OCSP

– CRL can only report a negative result

– "Not revoked" doesn't mean a cert was ever issued

– Some OCSP implementations will report "I can't find a CRL" as "Good"

– Some relying party implementations will assume "revoked" "not good", so any other status = "good"

– Much debate among implementors about OCSP semantics

Other Online Validation Protocols

- Simple Certificate Validation Protocol (SCVP)
  - Relying party submits a full chain of certificates
  - Server indicates whether the chain can be verified
  - Aimed mostly at thin clients
- Data Validation and Certification Server Protocols (DVCS)
  - Provides facilities similar to SCVP disguised as a general third-party data validation mechanism
- Integrated CA Services Protocol (ICAP)
- Real-time Certificate Status Protocol (RCSP)
- Web-based Certificate Access Protocol (WebCAP)
- Delegated Path Validation (DPV)

– Offshoot of the SCVP/DVCS debate and an OCSP alternative OCSP-X

Automatic Certificate Management Environment (ACME)

The certificate authorities in the Web PKI are trusted to verify that an applicant for a certificate legitimately represents the domain name(s) in the certificate.Today, the verification is done through a collection of ad hoc mechanisms.

ACME protocol automates process of verification and certificate issuance.

Discussion