5: Key Distribution Protocols

IT5306 - Principles of Information Security

Level III - Semester 5

List of sub topics

5.1. Diffie-Hellman Algorithm

5.2. Key Exchange with Public Key Cryptography

5.3. Concept of Digital Certificate

5.4. Certificate Authorities and its roles

5.5. Public Key Infrastructures (PKI)

5.6. Certificate Revocations

5.7 Automatic Certificate Management Environment (ACME)

Hybrid Encryption

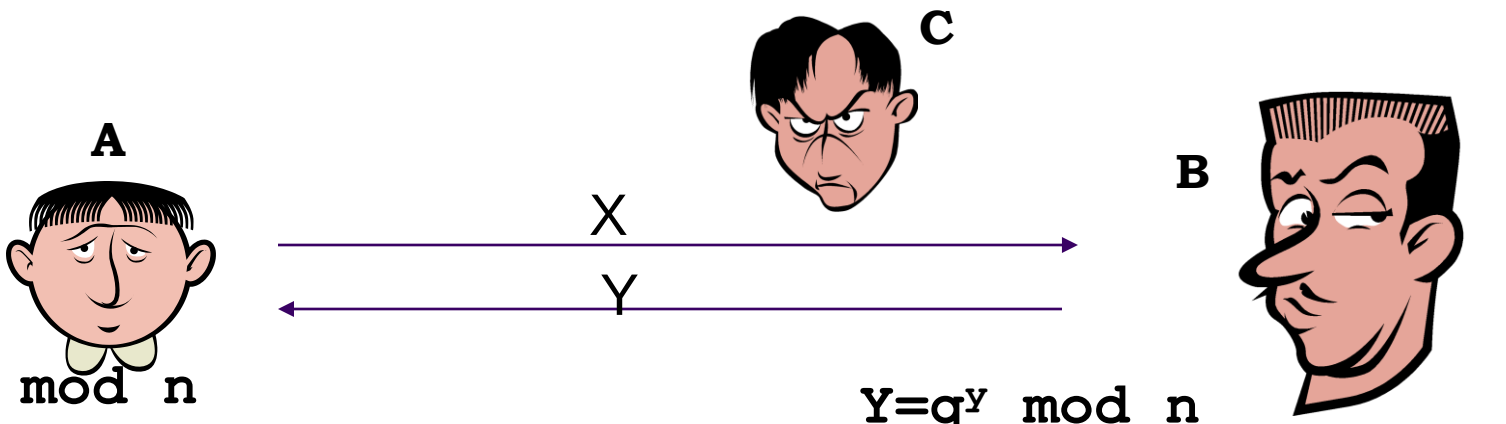
- Why is symmetric key encryption still used?
  - Performance
  - Also cryptographic reasons

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

Diffie-Hellman Key Agreement

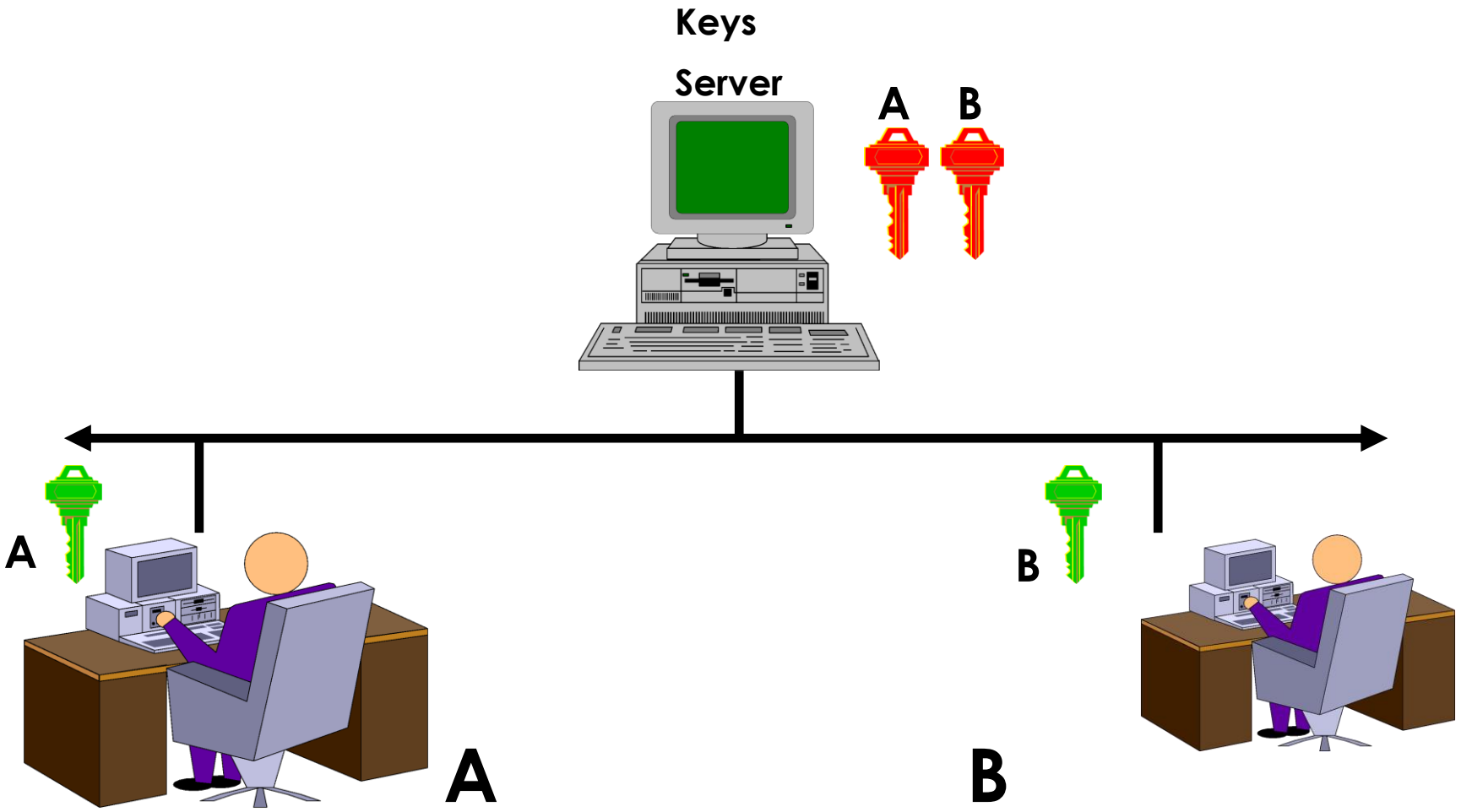
- **Published in 1976**
- **Based on difficulty of calculating discrete logarithm in a finite field**
- **Two parties agreed on two large numbers n and g, such that g is a prime with respect to n**



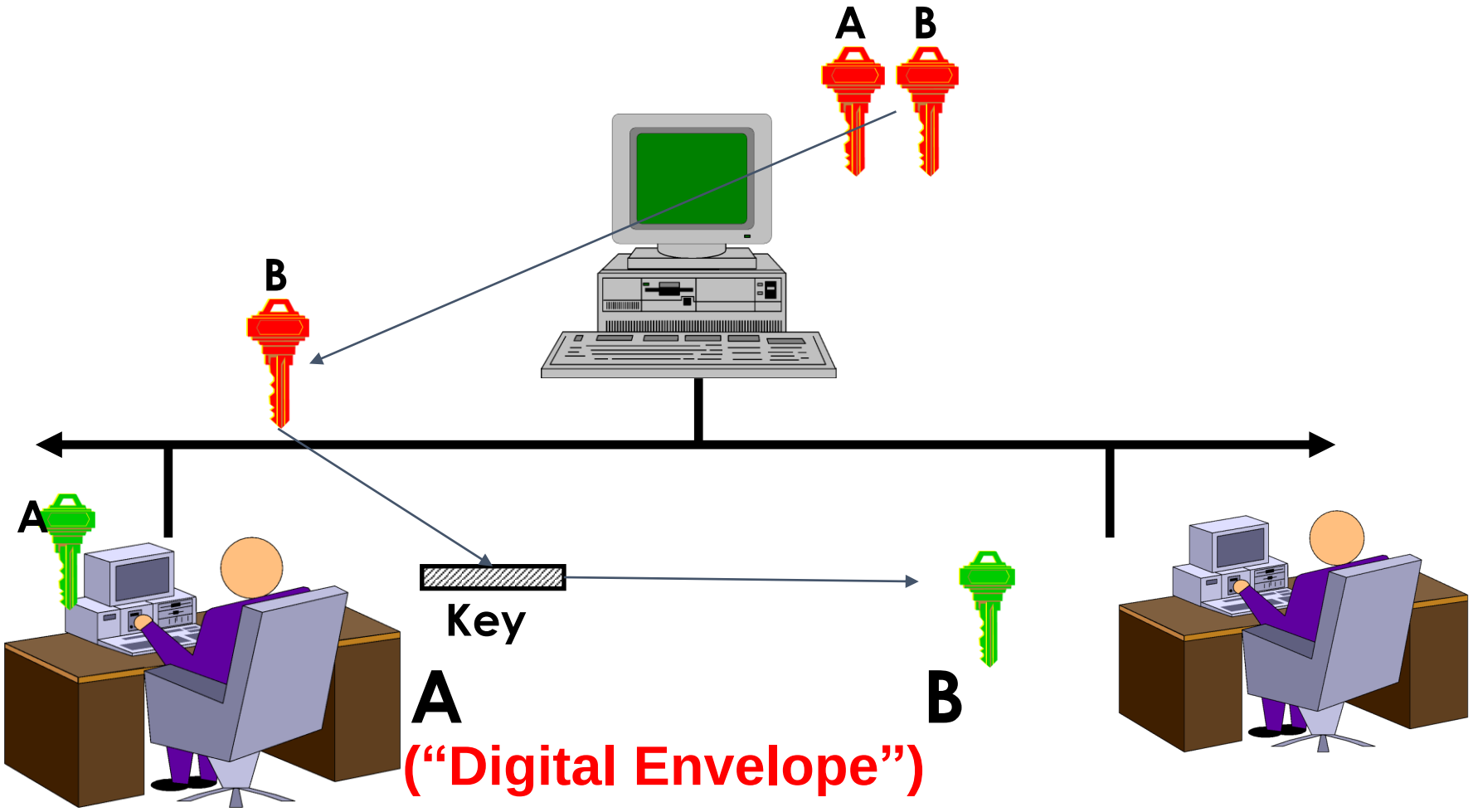
Key Management

- 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 B, and the calling protocol contacts C
- C encrypts a session key, “k”, with the public keys and sends the encrypted “k” to A and B
- A and B can then communicate

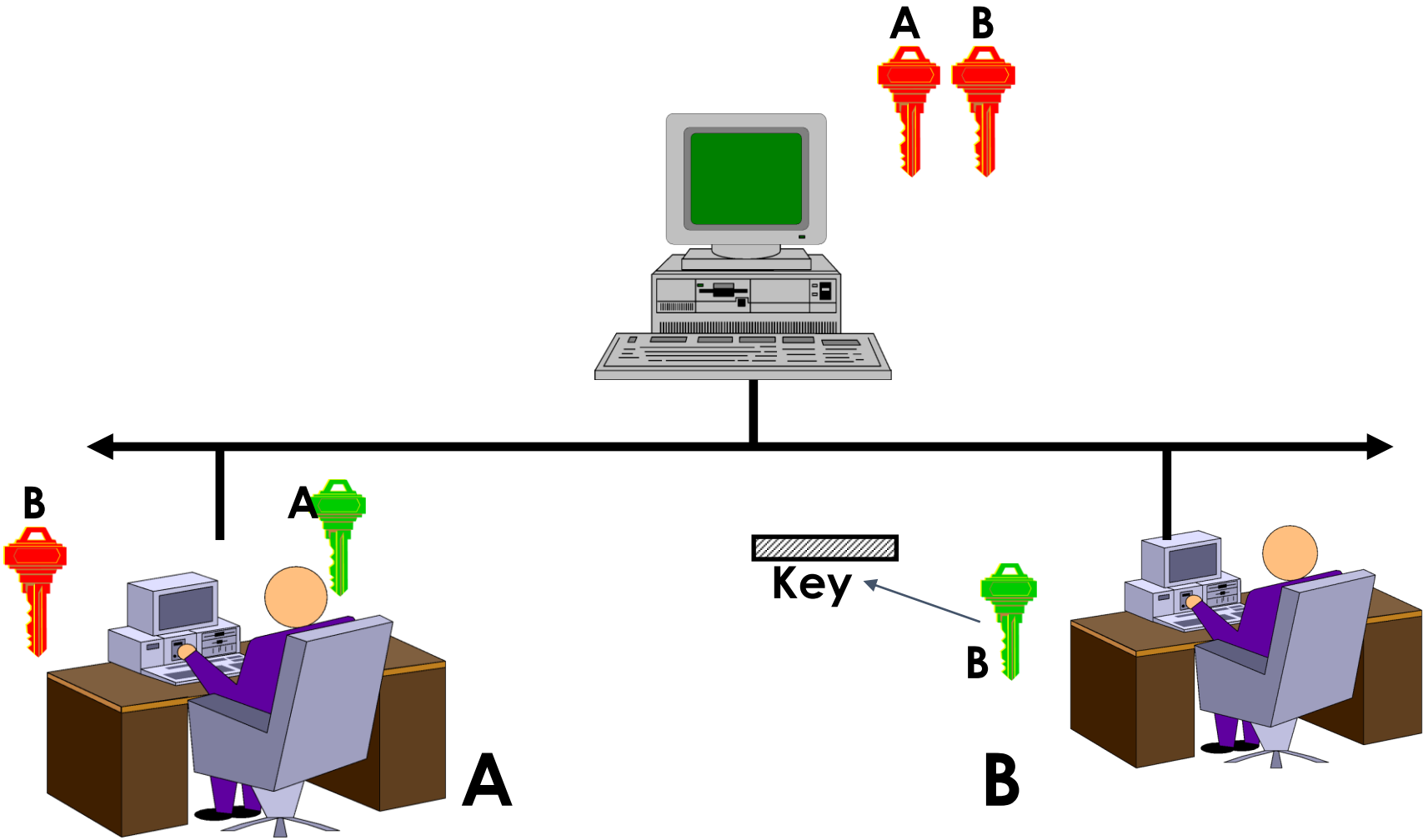
Storage and Handling Public Keys



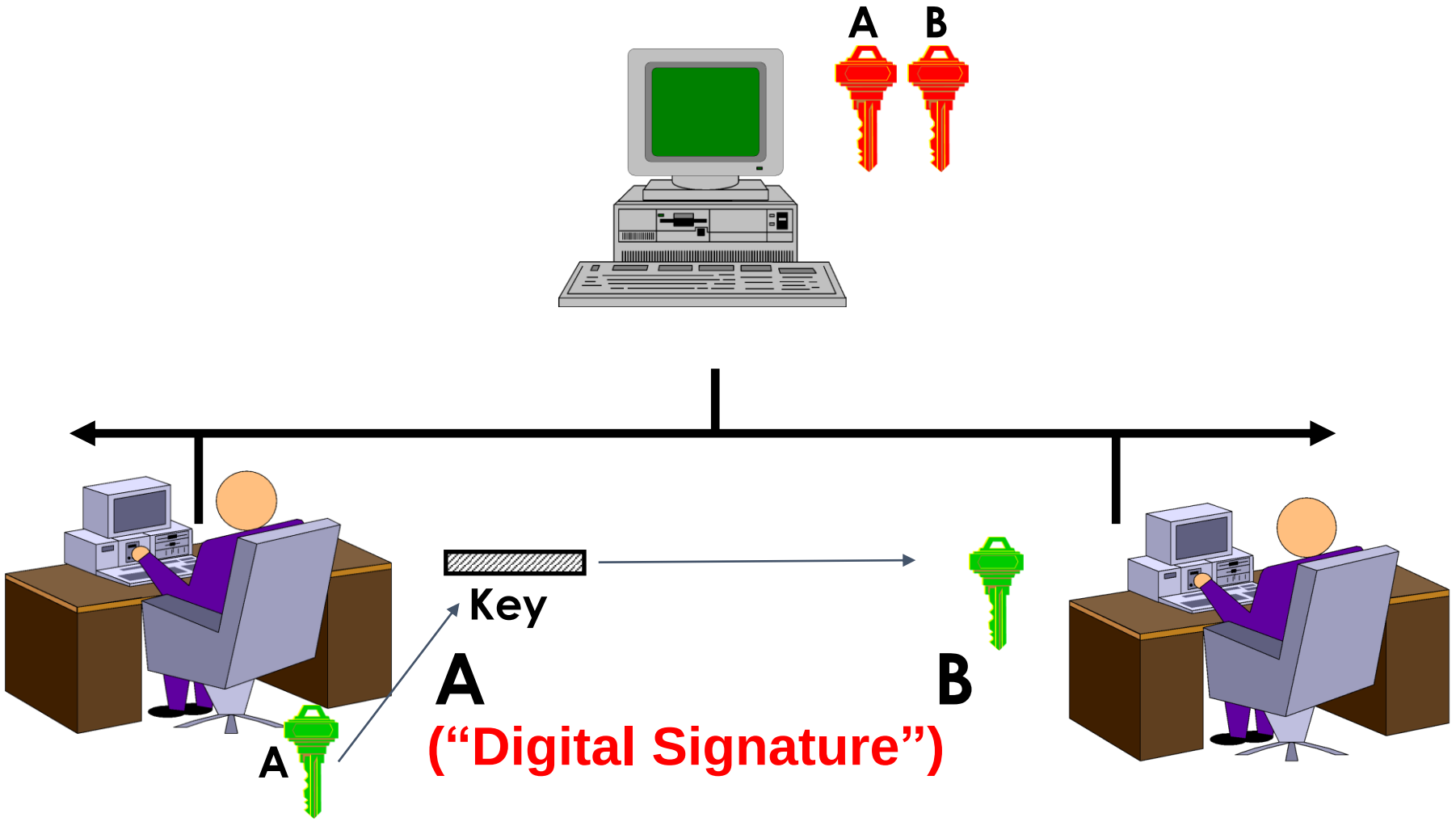
Secure Sending of secret key



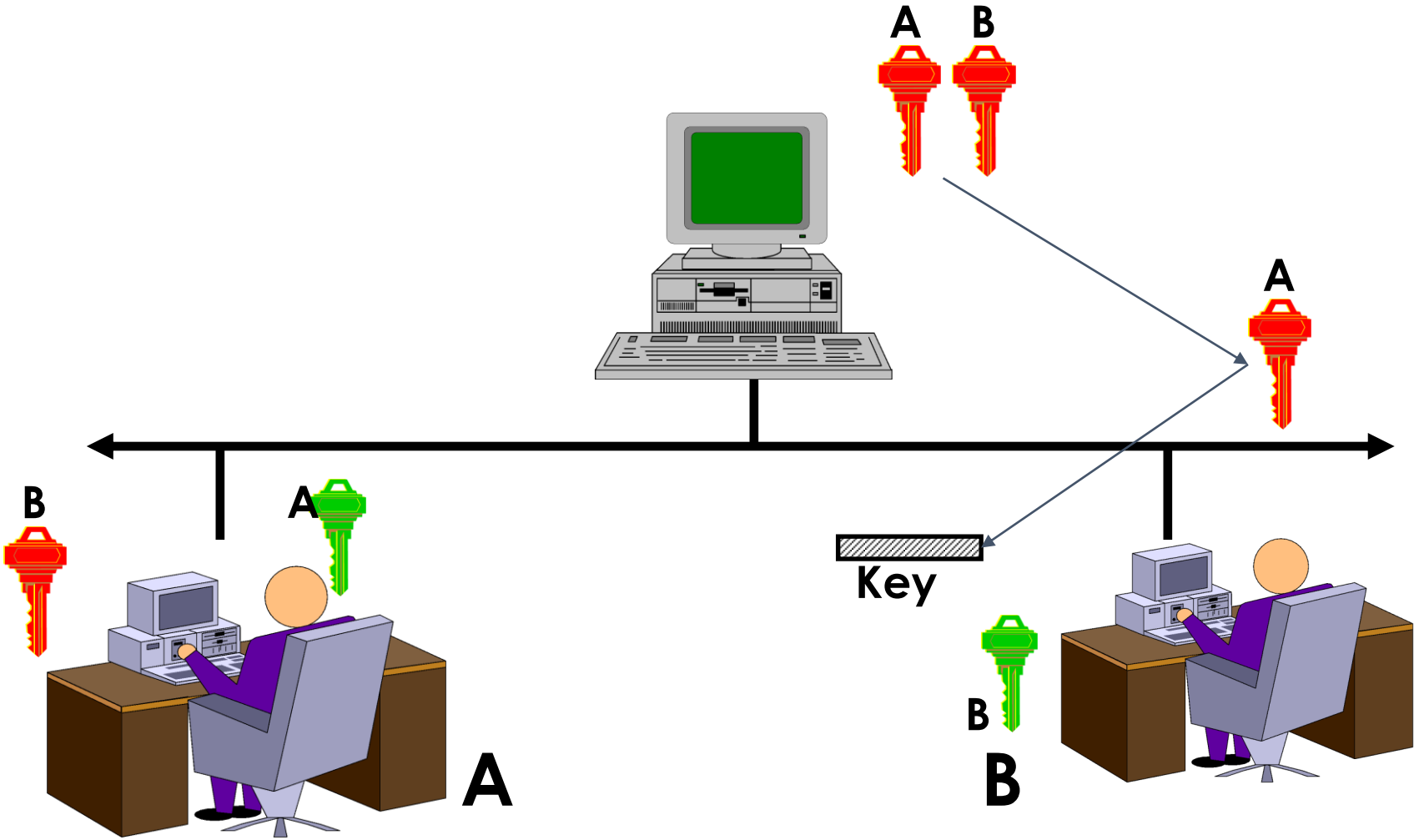
Recovery of Secret Key



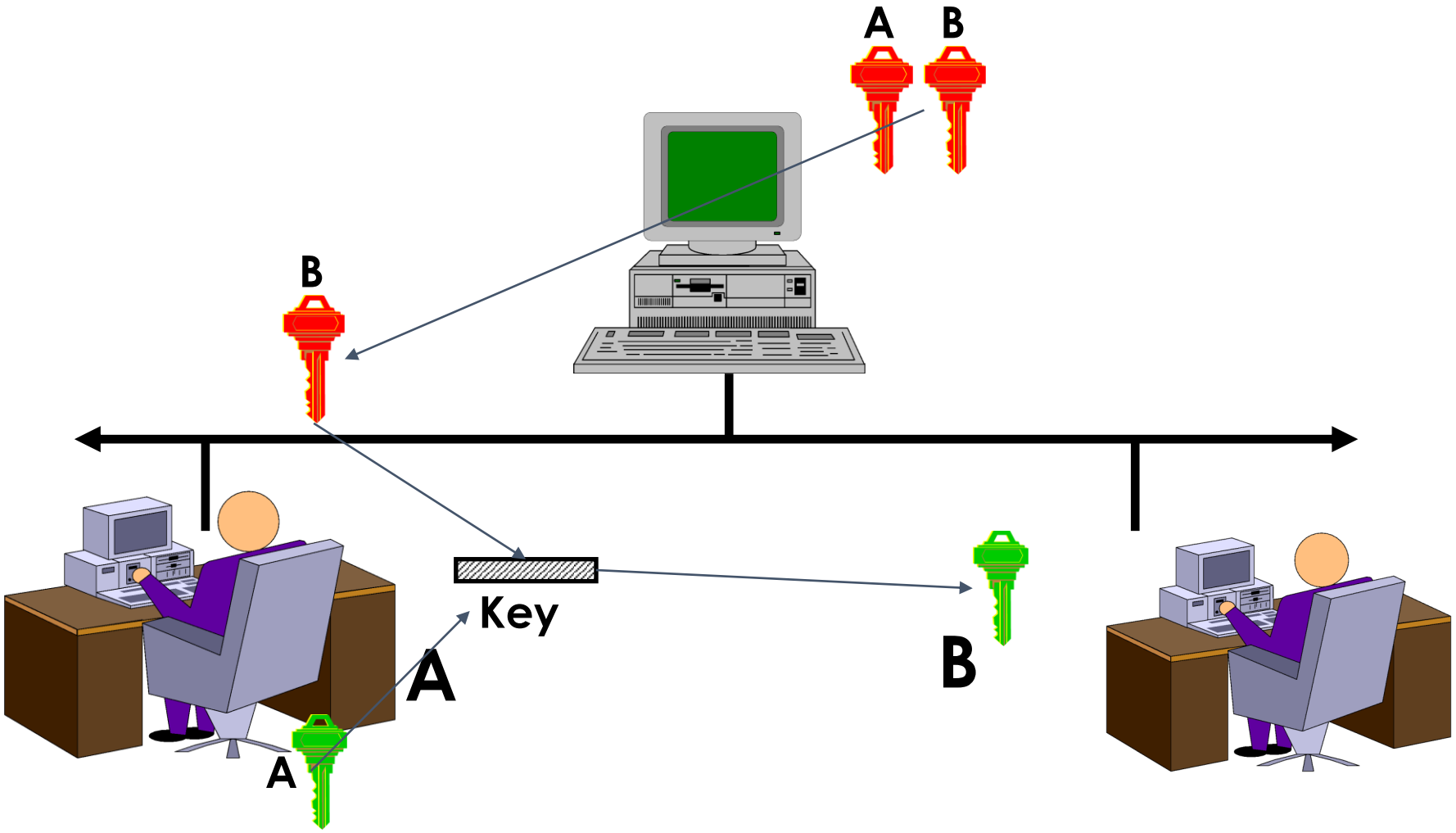
Authenticity of Sender



Verification of Signature

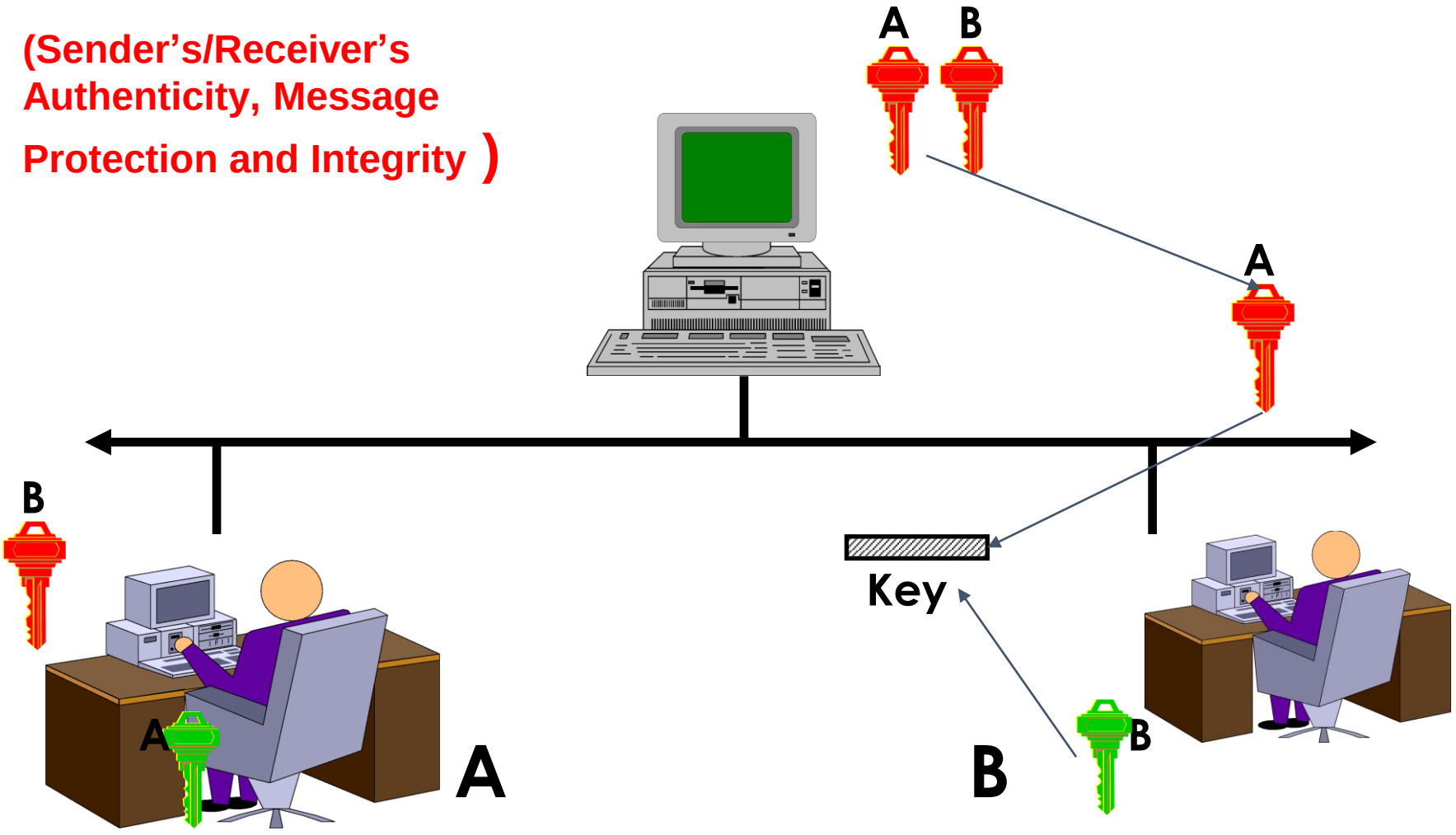


Authenticity of Sender and Receiver

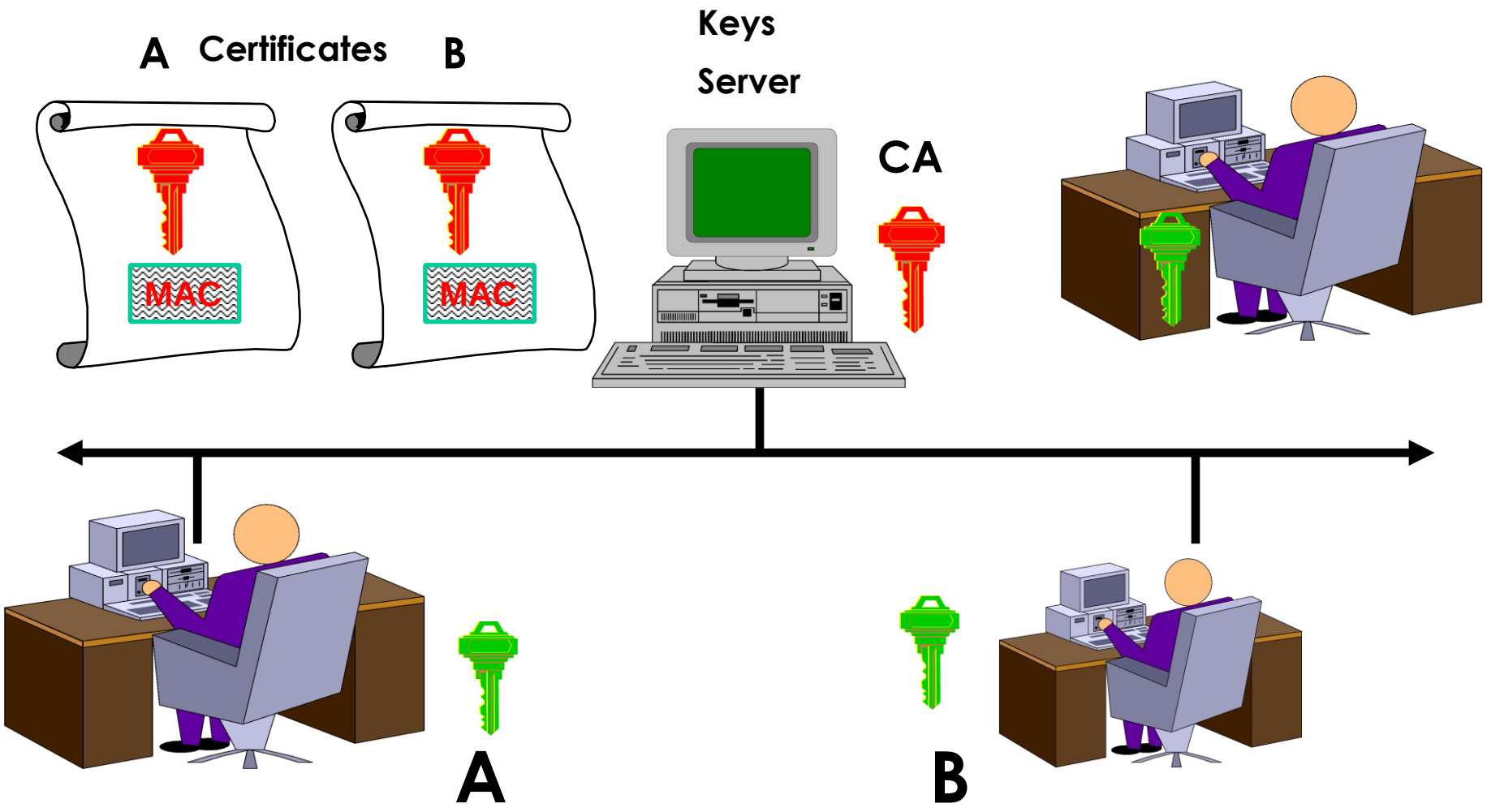


Full Verification

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



Certificate Authority



Problems with Public Key Cryptosystems

- Where and how can the public key for an entity be found?
- How to determine if a public key really belongs to whom it is supposed to?
- What is the legitimate use for a given public key? (e.g., signing checks, signing contracts)
- When is a public key no longer valid?
- How can you protect the corresponding private keys?
- What happens if a private key is lost?

Solution:
Use **Public Key Certificates** and **Public Key Infrastructure (PKI)**

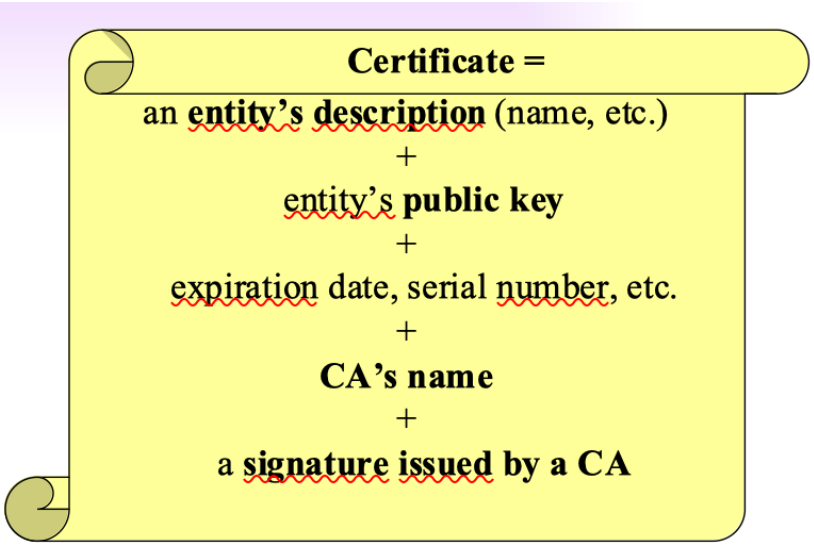
Public Key Certificates

- Binds a public key to a subject (e.g., person, employee, officer, company, institution, authority)
- Signed by an authority that vouches for the integrity of the binding and the identity of the subject
- Includes a serial number to help detect unauthorized or forged certificates
- Specifies when certificate will expire
- Indicates under what policies the certificate was granted and what policies it is used
- Provides other useful information regarding the subject or use of the certificate

Public Key Certificate

A public key certificate binds an entity with its public key. It's just a digitally signed piece of data.
The certificate is issued and signed by a trusted Certificate Authority (CA)

Digital signature:
CA signature = certificate hash, encrypted with CA's private key



Certificate Authorities (CAs)

- Abstraction of traditional sources of authority (e.g., manager, official, bureaucrat)
- Responsible for issuing certificates and renewing expired public key certificates
- Sets policies for public key certificate issuing and their use
- Revokes public key certificates for cause
- Establishes operational policies and procedures for issuing, renewing, and revoking certificates

Key Functions of CA

Generating key pairs − The CA may generate a key pair independently or jointly with the client.
Issuing digital certificates − the CA issues a certificate after client provides the credentials to confirm his identity. The CA then signs the certificate to prevent modification of the details contained in the certificate.

Publishing Certificates − The CA need to publish certificates so that users can find them. There are two ways of achieving this. One is to publish certificates in the equivalent of an electronic telephone directory. The other is to send your certificate out to those people you think might need it by one means or another.

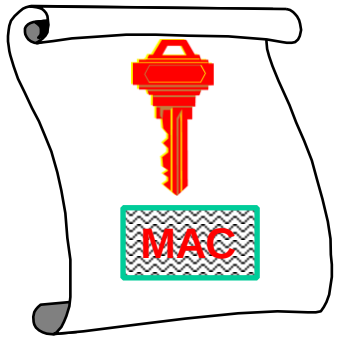
Key Functions of CA

Verifying Certificates − The CA makes its public key available in environment to assist verification of his signature on clients' digital certificate.

Revocation of Certificates − At times, CA revokes the certificate issued due to some reason such as compromise of private key by user or loss of trust in the client. After revocation, CA maintains the list of all revoked certificate that is available to the environment.

Internal Structure of Certificate

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

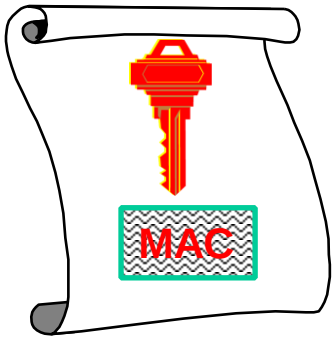


Structure of Distinguish Name

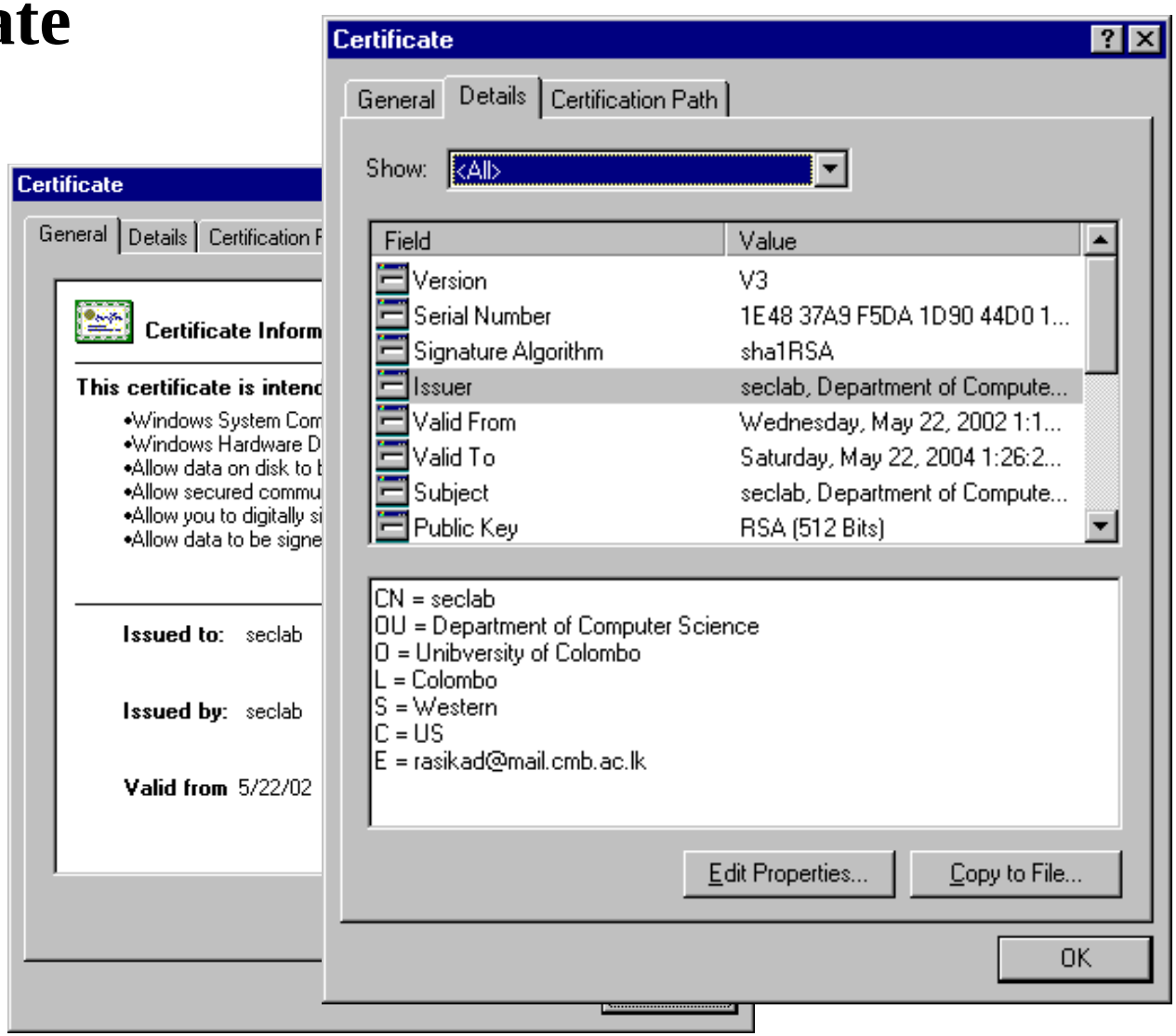
- Country Name
- State and Province Name
- Locality Name
- Organization Name
- Organization Unit Name
- Common Name
- Email Address
- URL

Certificate Types

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



Root Certificate



Public key infrastructure (PKI)

Certificates need some infrastructure in place to allow users to verify a given certificate.

- This can be done centrally or via a distributed system.
- So how are certificates, and their certificate chains, verified and disseminated?

(1) Trusted Third Party (TTP)
(2) Certificate Authority (CA)
(3) Simple Public Key Infrastructure (SPKI)

Public key infrastructure (PKI)

- Public key infrastructure (PKI) - provides the foundation necessary for secure e-business through the use of cryptographic keys and certificates
  - Enables secure electronic transactions
  - Enables the exchange of sensitive information

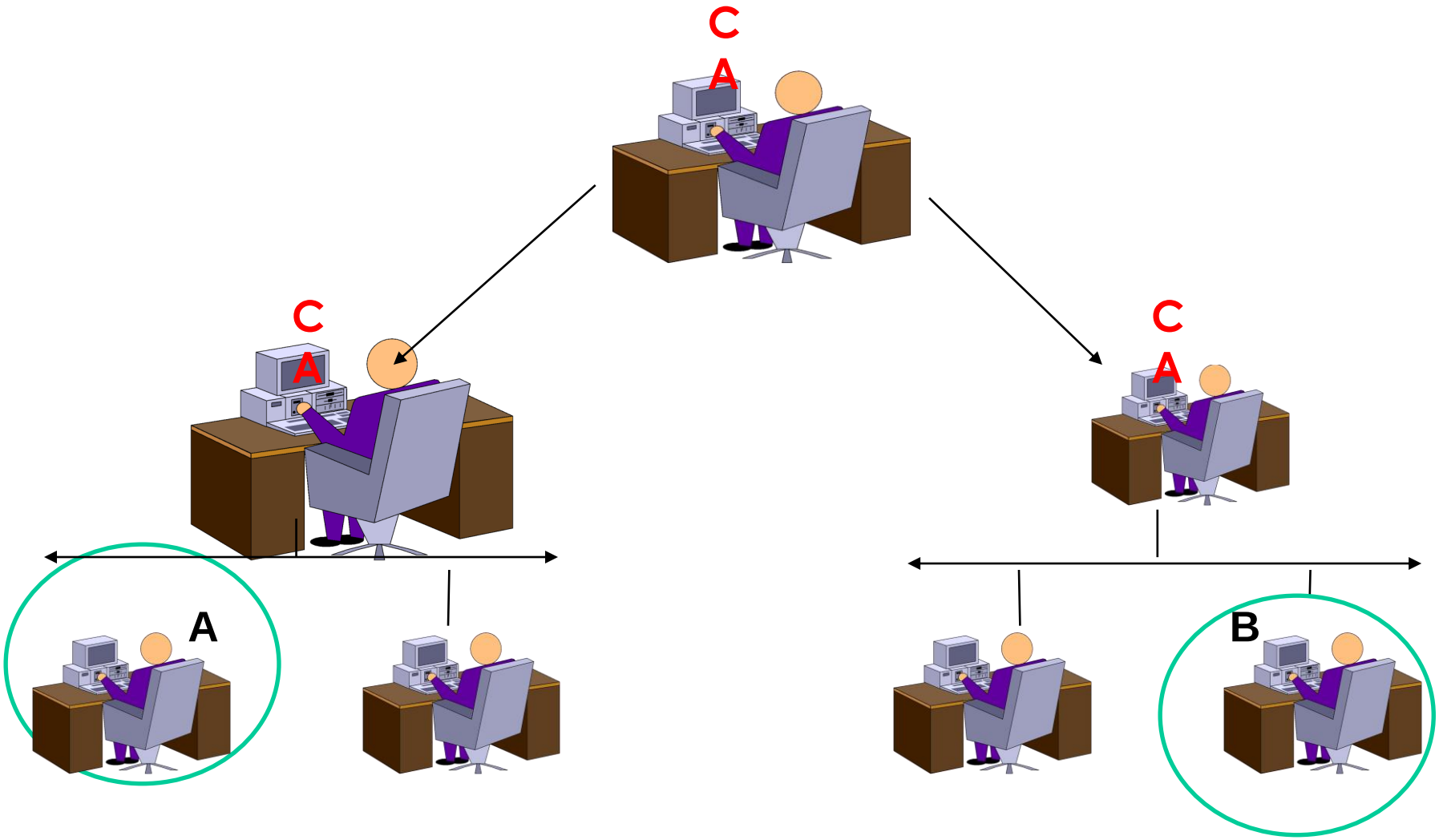
Public key infrastructure (PKI)

How to distribute public keys ?
Public Key Server (PKS), key exchange protocols

Public Key Infrastructure (PKI):
PKI = N x (Entities with private keys) + public key exchange system

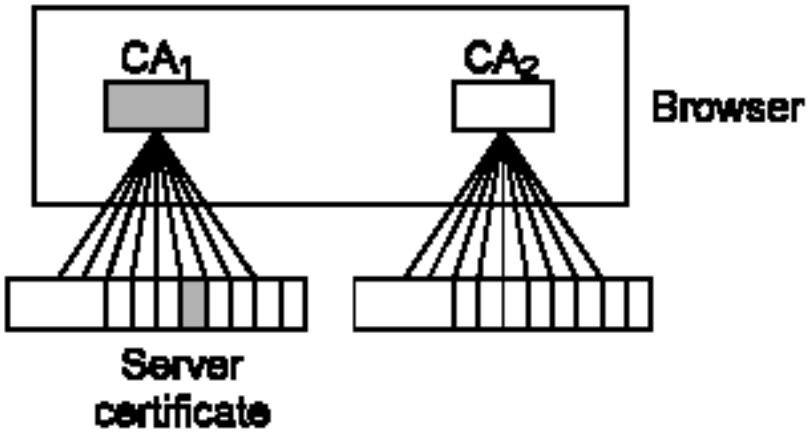
Public Key algorithms are slow
Need to use both Public & Secret Key Cryptography
Public Key Protocols work in 3 phases
Authentication via Public Key Cryptography (challenge)
Exchange of a session Secret Key, encrypted with Public Key Crypto
Session encrypted with Symmetric Cryptography

Certificate Hierarchy



CA Hierarchy in Practice

Flat or Clayton's hierarchy

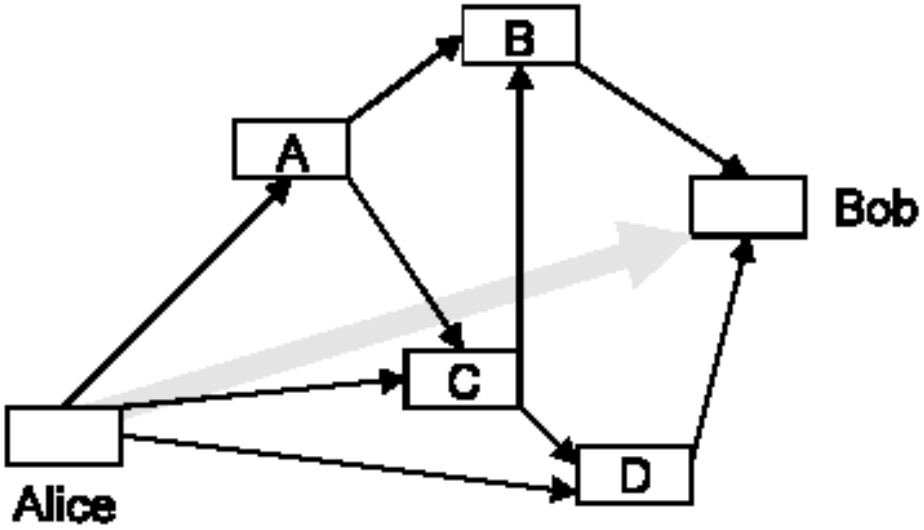


CA certificates are hard-coded into web browsers or email software

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

Alternative Trust Hierarchies

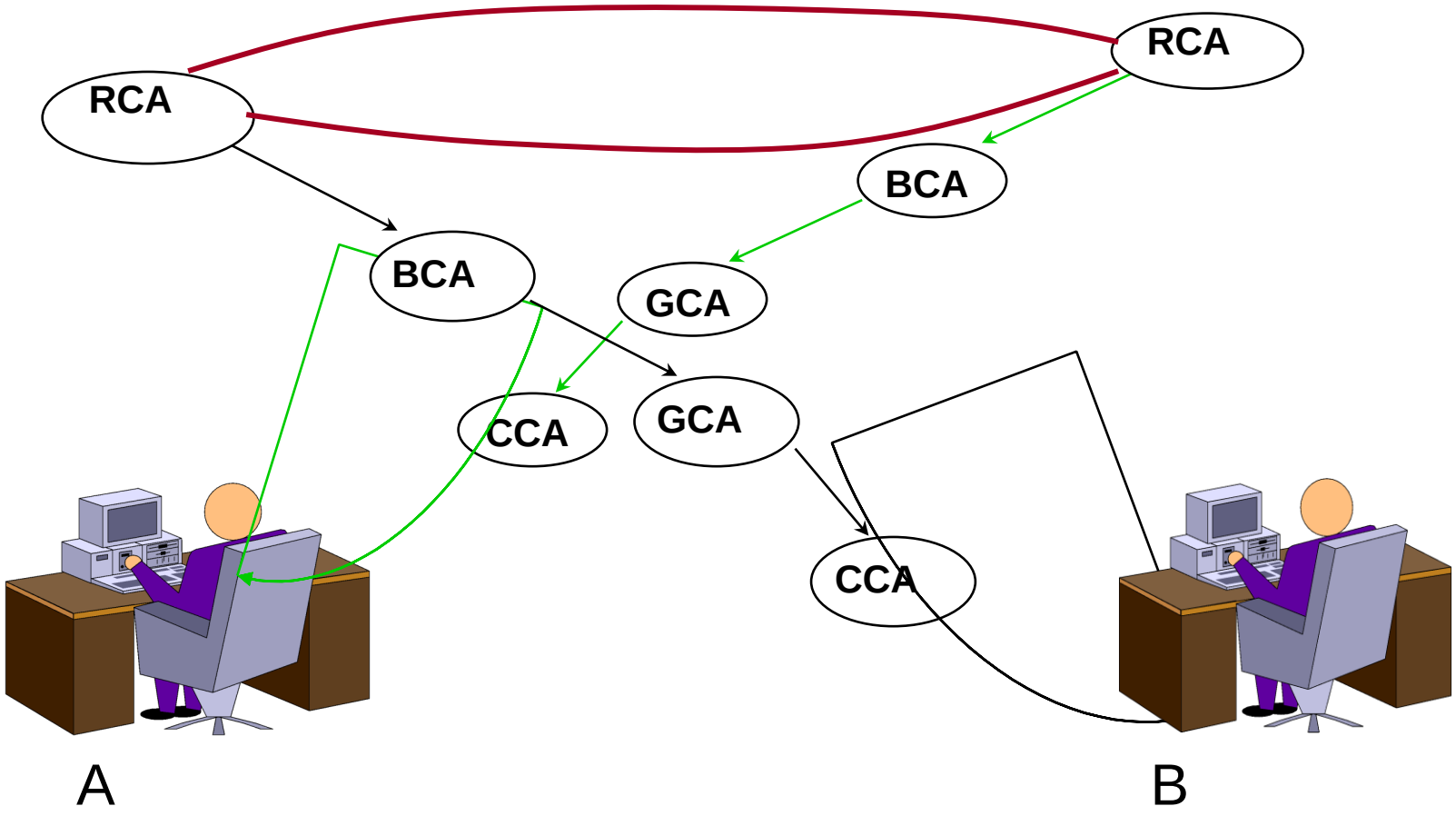
PGP web of trust



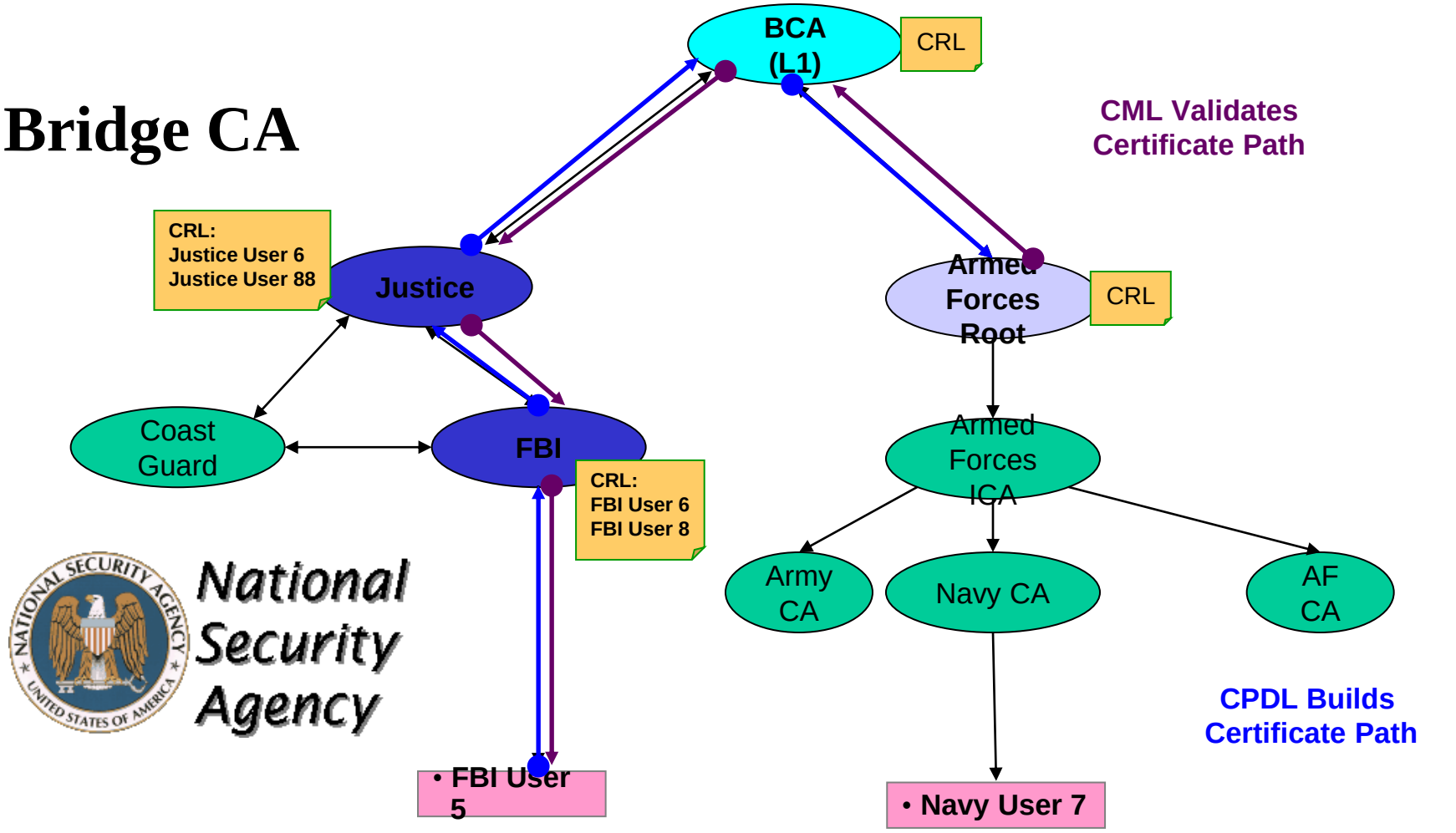
Bob knows B and D who know A and C who know Alice
$\Rightarrow$ Bob knows the key came from Alice

Web of trust more closely reflects real-life trust models

Cross Certification



Bridge CA



National Security Agency

Entrust User Signs and Transmits Encrypted Message to SPYRYUS User

Original Message (Decrypted, Sig Verified)

SPYRUS User Verifies Entrust User Signature Cert, Verifies Signature, Decrypts and Displays Message

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"

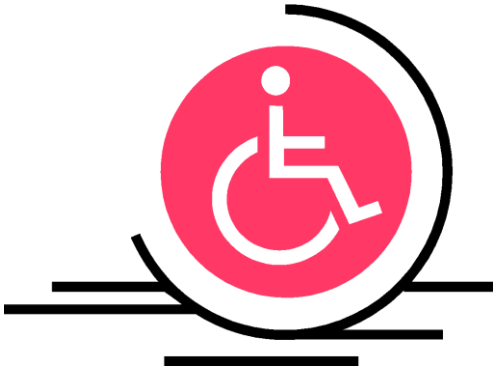
– Lightweight pseudo-CRL avoids CRL size problems

– 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)

Certificates in PKI using X.509 (PKIX) are used for a number of purposes, the most significant of which is the authentication of domain names.

Thus, 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, thi verification is done through a collection of ad hoc mechanisms.

ACME protocol automates process of verification and certificate issuance.

Automatic Certificate Management Environment (ACME)

The ACME (Automated Certificate Management Environment) protocol was originally developed by the Internet Security Research Group for its public CA, L**et's Encrypt**.

ACME is what facilitates Let's Encrypt's entire business model, allowing it to issue 90-day domain validated SSL certificates that can be renewed and replaced without website owners ever having to lift a finger.

The ACME protocol functions by installing a certificate management agent on a given web server.

Thank You