# Public Key Infrastructure and Applications

Svetlin Nakov
Sofia University “St. Kliment Ohridski”
E-mail: pki@nakov.com

Nikolay Nedyalkov
Latona Development
E-mail: pki@nedyalkov.com

# Agenda

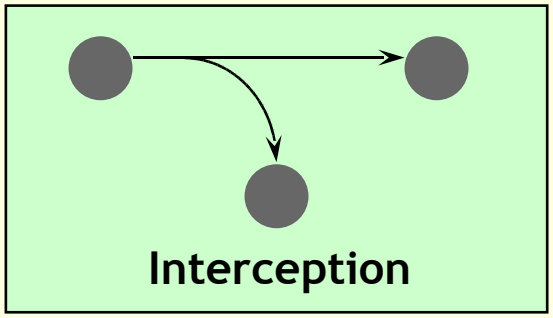
- PKI Overview
- Digital Signatures
  - What is it?
  - How does it work?
- Digital Certificates
- Public Key Infrastructure
  - PKI Components
  - Policies
- Internet Security
  - Web Security with SSL
- Smart Cards
- Email signing – S/MIME

# What's the problem?

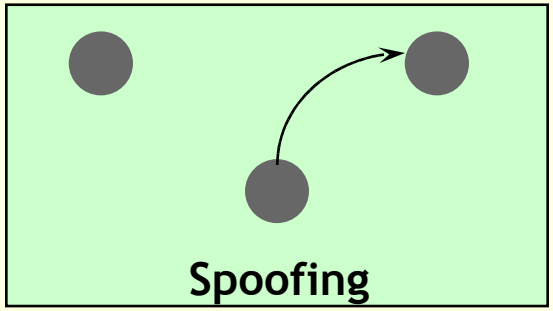
- Information over the Internet is Free, Available, Unencrypted, and Untrusted.
- Not desirable for many Applications
  - Electronic Commerce
  - Software Products
  - Financial Services
  - Corporate Data
  - Healthcare
  - Subscriptions
  - Legal Information

# Multiple Security Issues

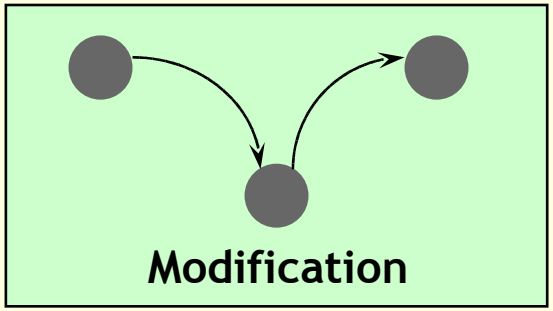
Privacy



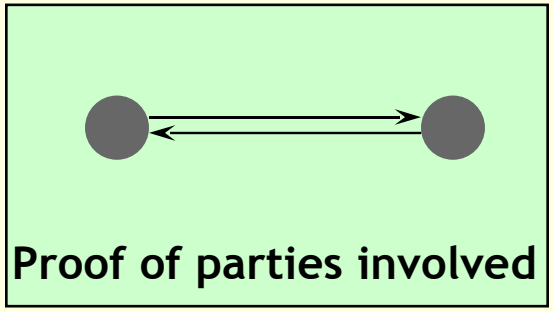

Authentication




Integrity




Non-repudiation



# Why do PKIs need Trust ?

- CAs could issue certificates without checking the owner identity.
- CAs could deliberately issues false certificates.
- Private keys could be disclosed by accident, ... or on purpose.

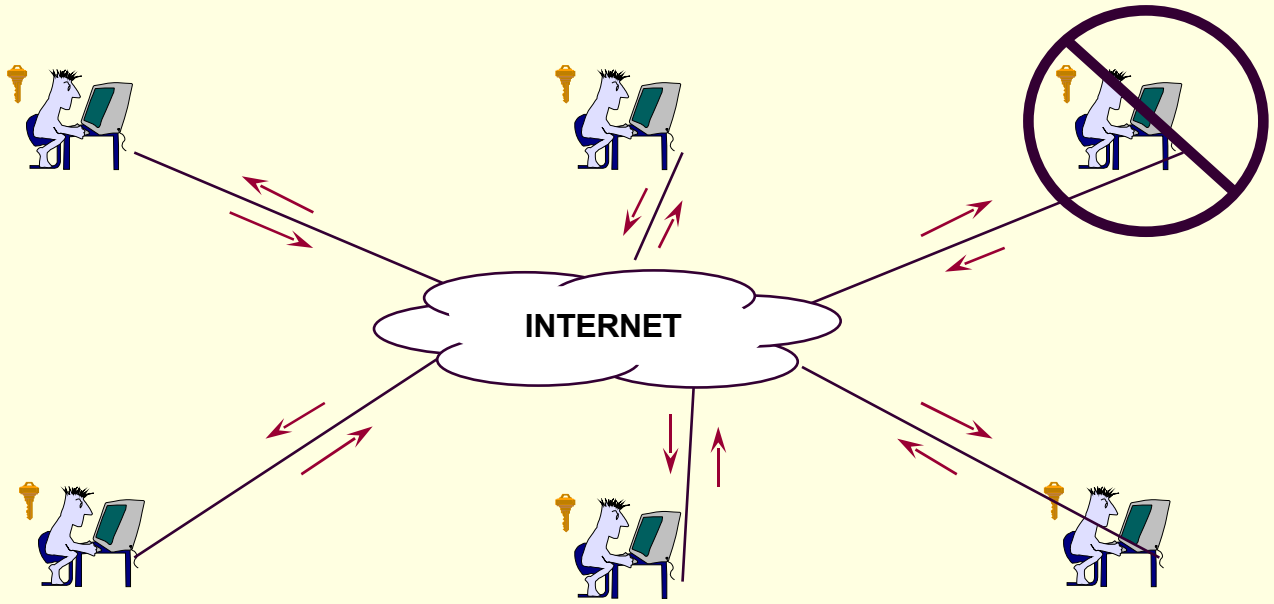
- False certificates could be inserted into your browser.
- Portals could contain false URLs.
- Knowing a principal's identity does not mean that the principal can be trusted.

# Security Algorithms

- Public Key Algorithms
  - RSA, DSA, Diffie-Hellman, Elliptic Curve
- Symmetric Algorithms
  - Triple-DES, DES, CAST, RC2, IDEA
- Hashing Algorithms
  - SHA-1, MD5, RIPEMD

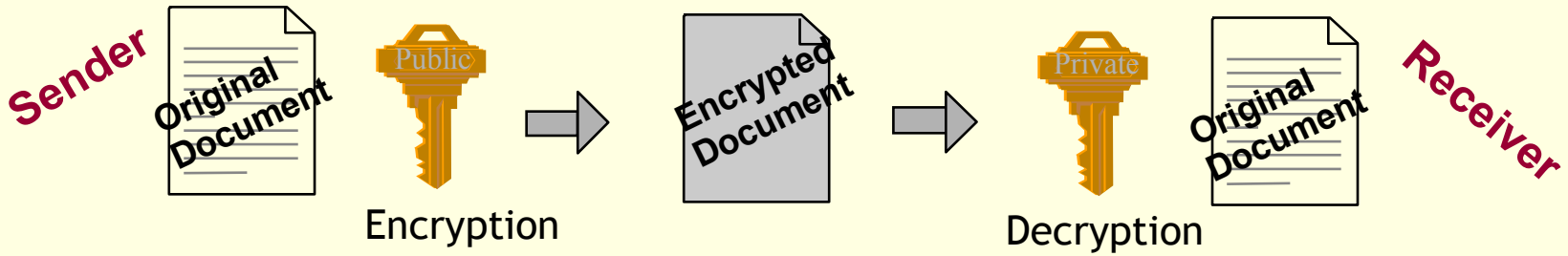
# Symmetric Key Encryption

- If any one's key is compromised, all keys need to be replaced
- Not practical or cost effective for Internet environments



# Public Key Cryptography

- Public-Key Cryptography is an encryption scheme that uses **mathematically** related, but **not identical** keys.
- Each user has a key pair (public key/private key).

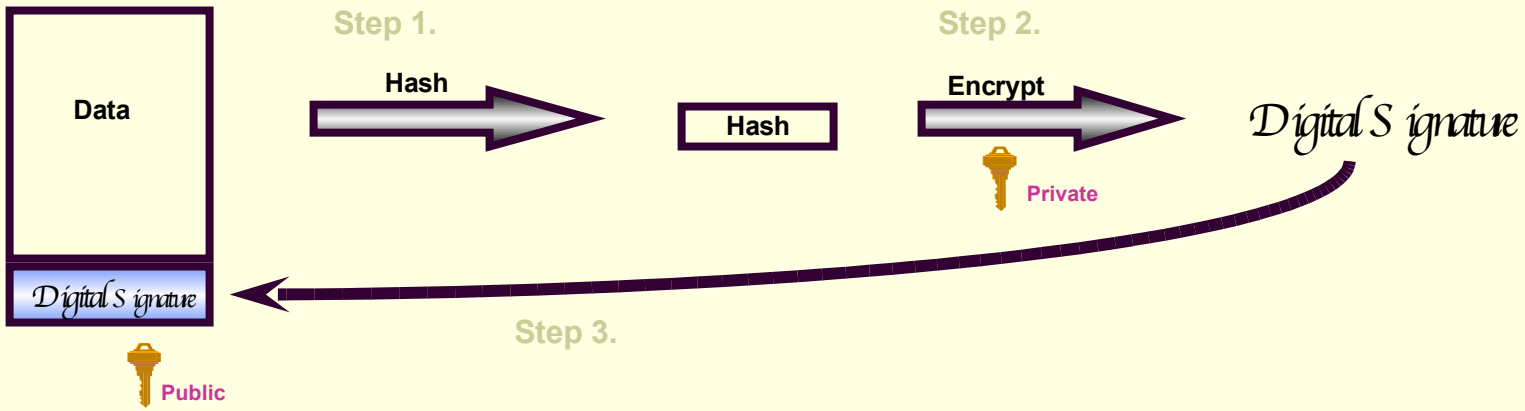



- Information encrypted with the public key can only be decrypted using the private key.

# What is a Digital Signature ?

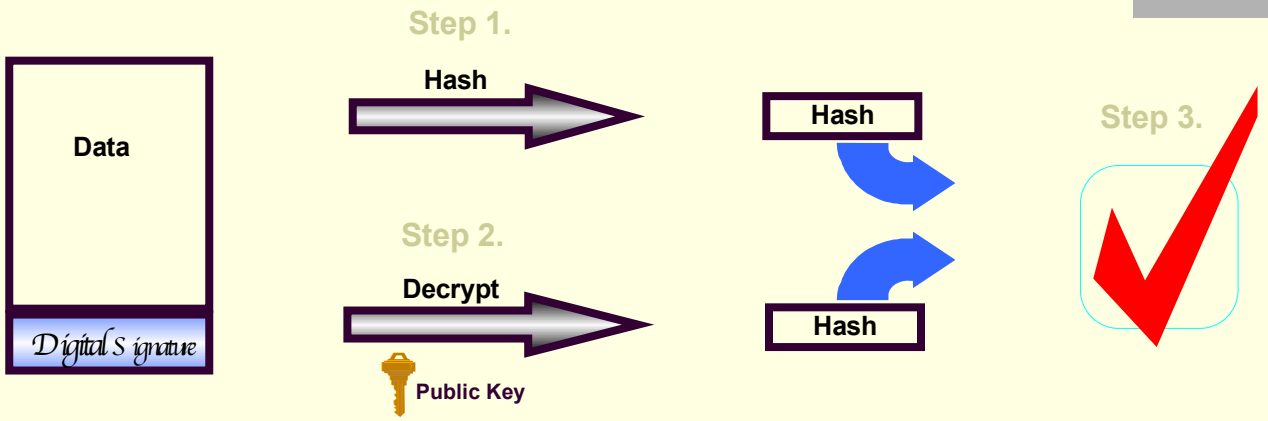
- A Digital Signature is the result of **encrypting** the Hash of the data to be exchanged.
- A Hash (or Message Digest) is the process of mathematically reducing a data stream down to a fixed length field.
- The Hash uniquely represents the original data.
- The probability of producing the same Hash with two sets of different data is <.001%.
- Signature Process is opposite to Encryption Process
  - Private Key is used to Sign (encrypt) Data
  - Public Key is used to verify (decrypt) Signature

# Digital Signature Process




- Step 1. Hash (digest) the data using one of the supported Hashing algorithms, e.g., MD2, MD5, or SHA-1.
- Step 2. Encrypt the hashed data using the sender's private key.
- Step 3. Append the signature (and a copy of the sender's public key) to the end of the data that was signed.

# Signature Verification Process




- Step 1. Hash the original data using the same hashing algorithm.
- Step 2. Decrypt the digital signature using the sender's public key. All digital signatures contain a copy of the signer's public key.
- Step 3. Compare the results of the hashing and the decryption. If the values match then the signature is verified. If the values do not match, then the data or signature was probably modified in transit.

# The Critical Questions

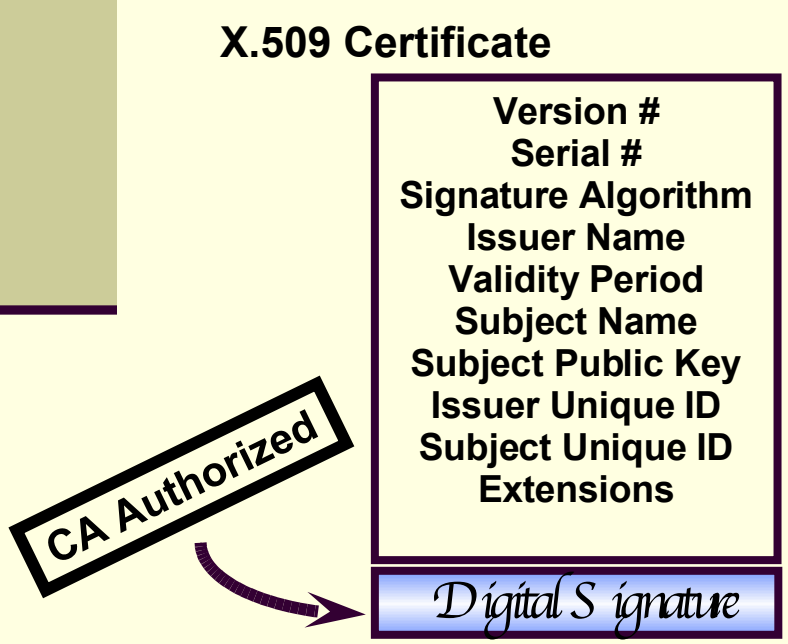
- How can the recipient know with certainty the sender's public key? (to validate a digital signature)
- How can the sender know with certainty the recipient's public key? (to send an encrypted message)

# Digital Certificates


- Before two parties exchange data using Public Key cryptography, each wants to be sure that the other party is authenticated
- Before B accepts a message with A's Digital Signature, B wants to be sure that the public key belongs to A and not to someone masquerading as A on an open network
- One way to be sure, is to use a trusted third party to authenticate that the public key belongs to A. Such a party is known as a **Certification Authority (CA)**
- Once A has provided proof of identity, the Certification Authority creates a message containing A's name and public key. This message is known as a **Digital Certificate**.

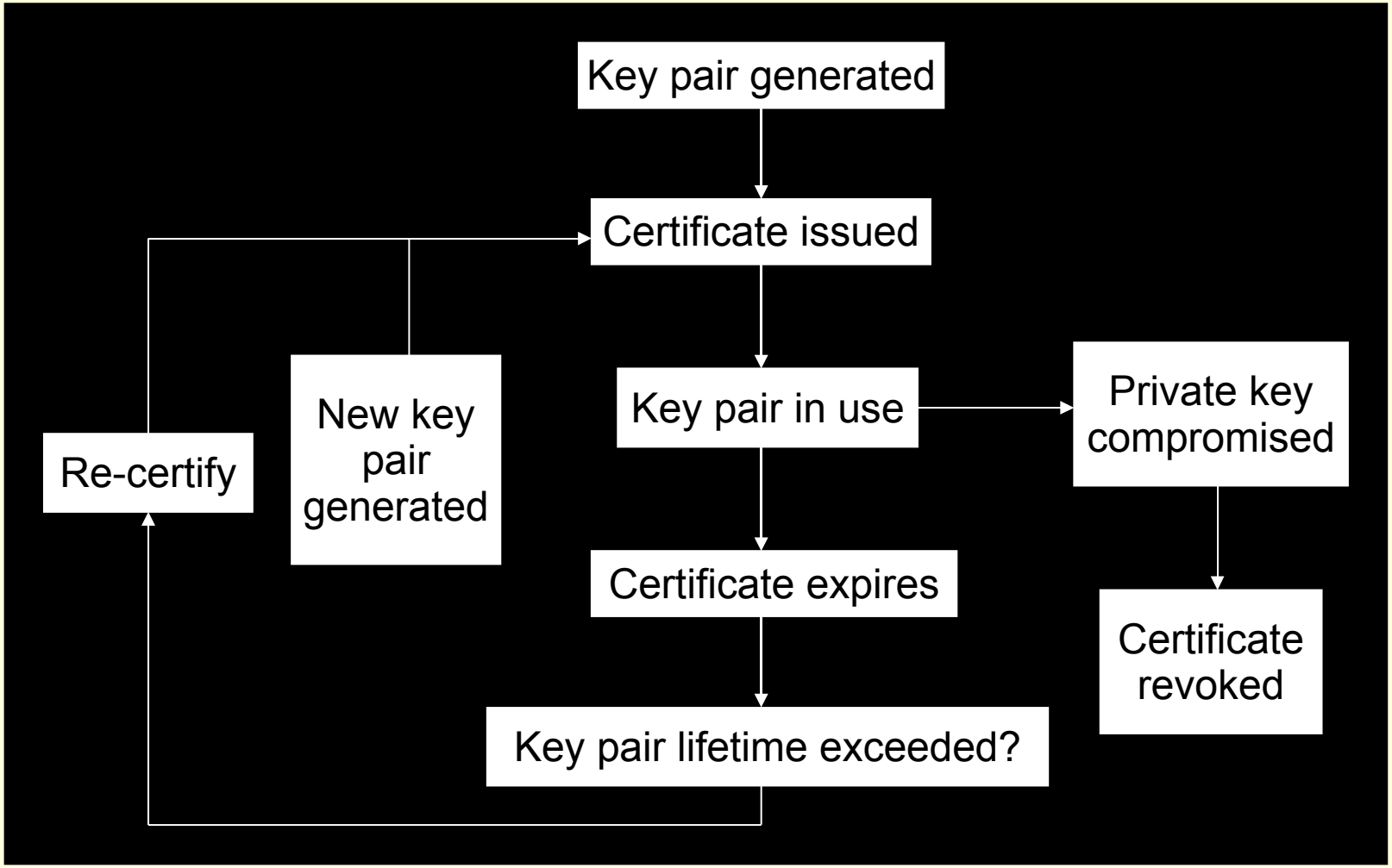
# Digital Certificates

- A Digital Certificate is simply an X.509 defined data structure with a Digital Signature. The data represents who owns the certificate, who signed the certificate, and other relevant information




- When the signature is generated by a Certification Authority (CA), the signature can be viewed as trusted.
- Since the data is signed, it can not be altered without detection.
- Extensions can be used to tailor certificates to meet the needs of end applications.

# Certificate Life Cycle



# Certificate Revocation Lists

- CA periodically publishes a data structure called a certificate revocation list (CRL).
- Described in X.509 standard.
- Each revoked certificate is identified in a CRL by its serial number.
- CRL might be distributed by posting at known Web URL or from CA's own X.500 directory entry.

# PKI Players

- Registration Authority (RA) to identity proof users
- Certification Authorities (CA) to issue certificates and CRL's
- Repositories (publicly available databases) to hold certificates and CRLs

# Certification Authority (CA)

## *Certification Authority*

- Trusted (Third) Party
- Enrolls and Validates Subscribers
- Issues and Manages Certificates
- Manages Revocation and Renewal of Certificates
- Establishes Policies & Procedures

## *What's Important*

- Operational Experience
- High Assurance Security Architecture
- Scalability
- Flexibility
- Interoperability
- Outsource vs. Inhouse
- Trustworthiness

**Certification Authority = Basis of Trust**

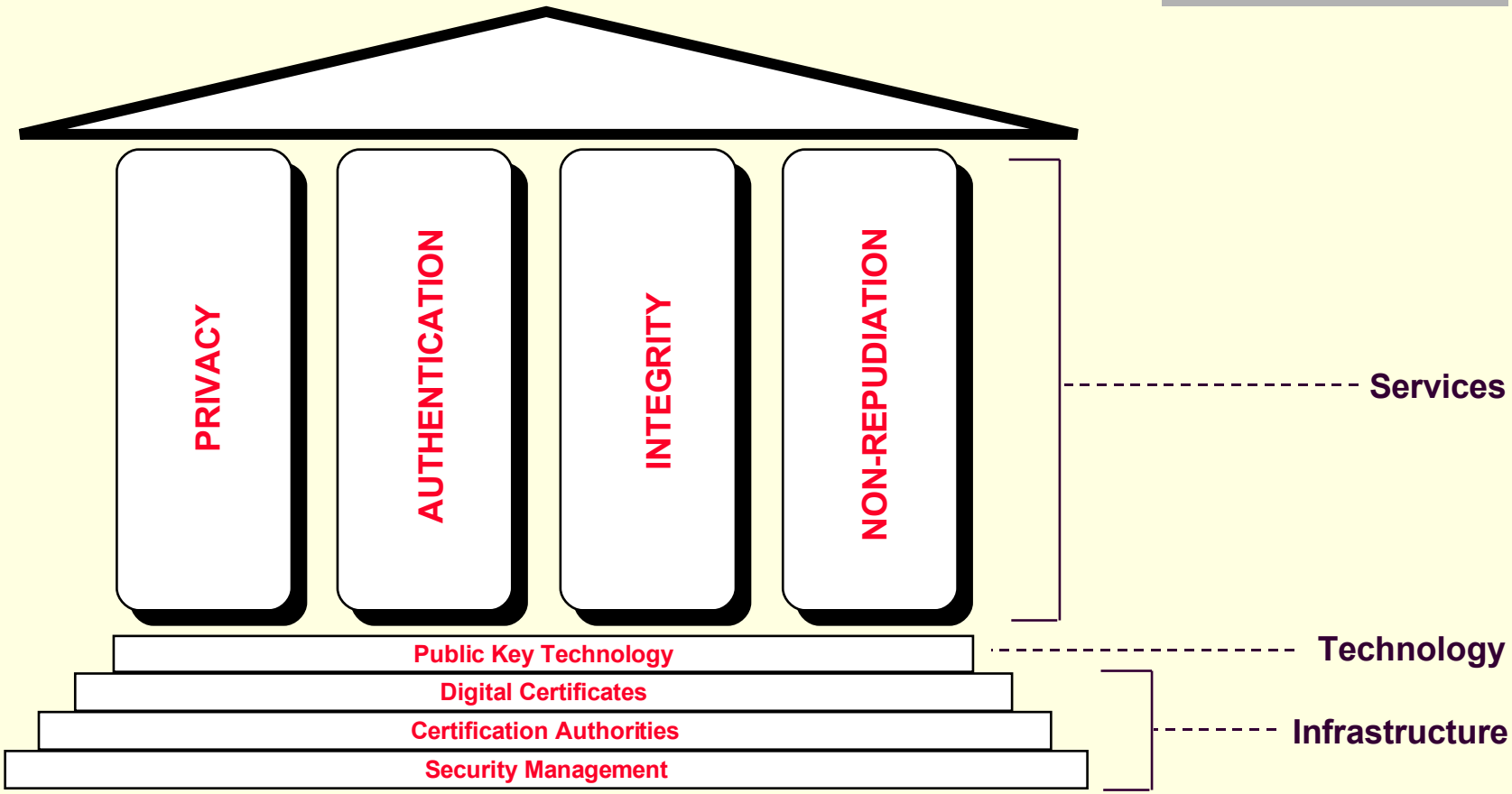
# Registration Authority (RA)

- Enrolling, de-enrolling, and approving or rejecting requested changes to the certificate attributes of subscribers.
- Validating certificate applications.
- Authorizing requests for key-pair or certificate generation and requests for the recovery of backed-up keys.
- Accepting and authorizing requests for certificate revocation or suspension.
- Physically distributing personal tokens to and recovering obsolete tokens from people authorized to hold and use them.

# Certificate Policy (CP) is …

- the basis for trust between unrelated entities
- not a formal “contract” (but implied)
- a framework that both informs and constrains a PKI implementation
- a statement of what a certificate means
- a set of rules for certificate holders
- a way of giving advice to Relying Parties

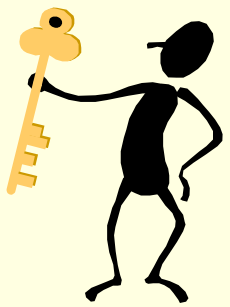
# Public Key Security




- Public Key Technology Best Suited to Solve Business Needs
- Infrastructure = Certification Authorities

# Authentication/Access Control

- Can Public Key Technology be used to perform Authentication and Access Control?



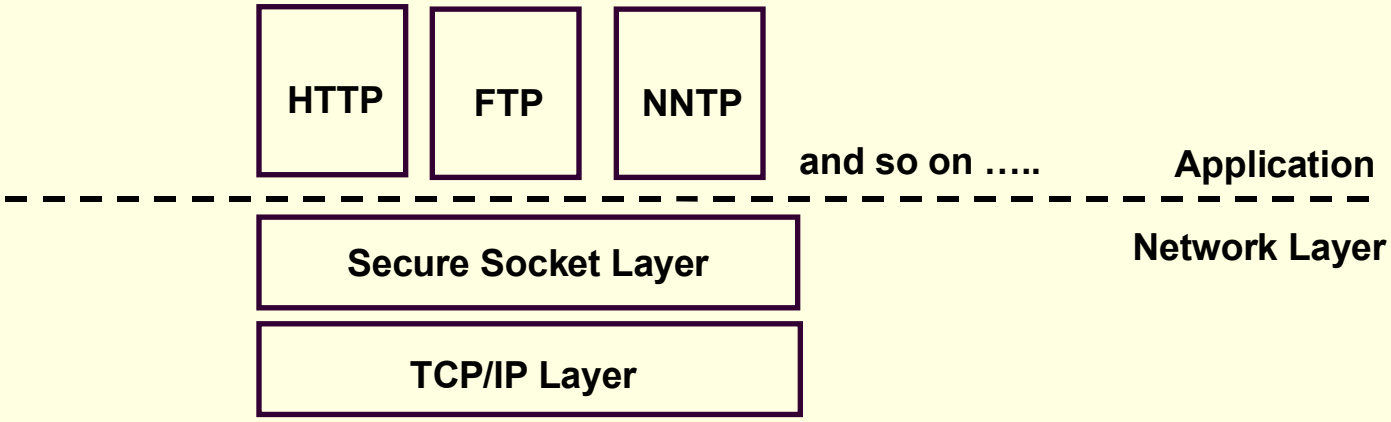
**Sure Can**

**How?**


**Using Digital Signatures and Digital Certificates**

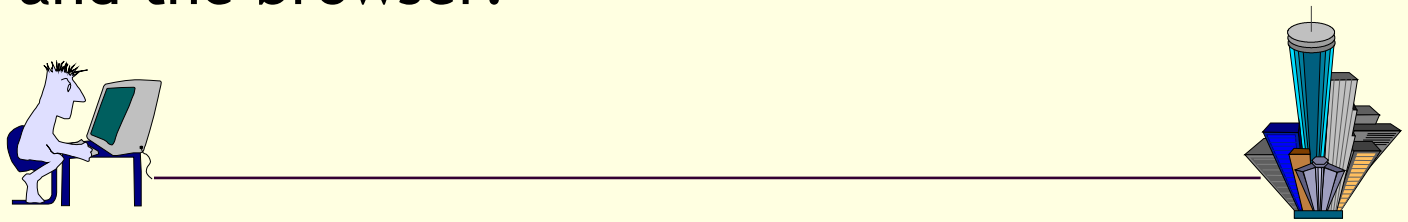
# SSL Protocol

- Secure Socket Layer (SSL) is a Network Layer protocol used to secure data on TCP/IP networks.



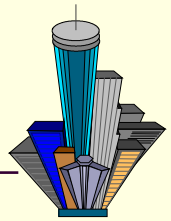
# SSL 2.0 Protocol

- SSL 2.0 provides encryption between the server and the browser.




• Browser and Server use $SessKey_B$ to encrypt all data exchanged over the Internet

# SSL 3.0 with Client Authentication



• Browser Connects to Secure Server

• Server sends copy of Server certificate ($Cert_S$) to Browser, indicating that SSL 3.0 is enabled with client authentication

$Cert_S$ - SSL 3.0

• Browser verifies signature on $Cert_S$
• Browser generates session key ($SessKey_B$)
• Browser encrypts $SessKey_B$ using $Cert_S$
• Browser asks operator to select a Browser certificate ($Cert_B$) to access server

{$SessKey_B$ } $Cert_S$ + $Cert_B$

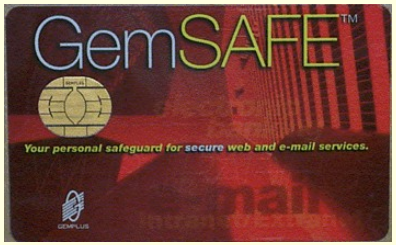
• Server verifies signature on $Cert_B$ (Server can check other information as well)
• Server decrypts $SessKey_B$ using it's private key

{Data} $SessKey_B$

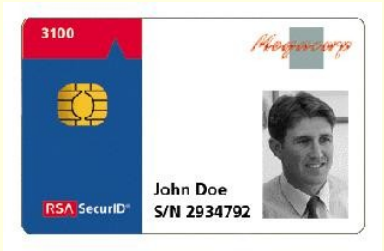
• Browser and Server use $SessKey_B$ to encrypt all data exchanged over the Internet

# Smart Cards

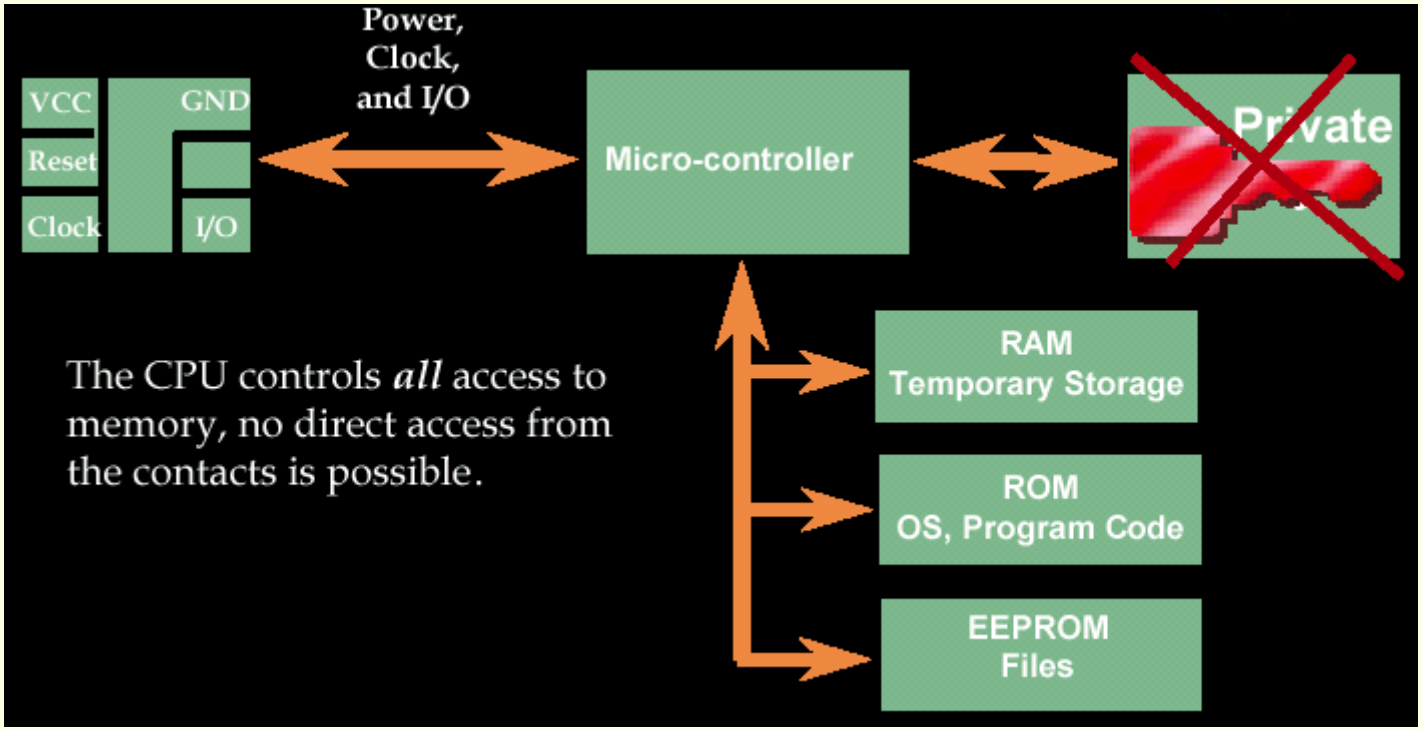
- Microprocessor with memory that can generate and store keys and certificates
- Different form factors and interface mechanisms
- Cryptographic functions using private key are processed on the card itself








# Microprocessor based smart card



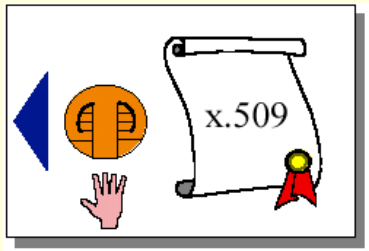
# Smart Cards and PKI

- Smart cards are «certificate wallets»
- Secure storage for:
  - Owner private key
  - Trusted root certificates
- Smart Cards are a «PC-in-your-Pocket»
  - Generation of owner's digital signature
- Smart cards provide:
  - Mobility
  - Security
  - Transparency
  - Issuer branding, loyalty

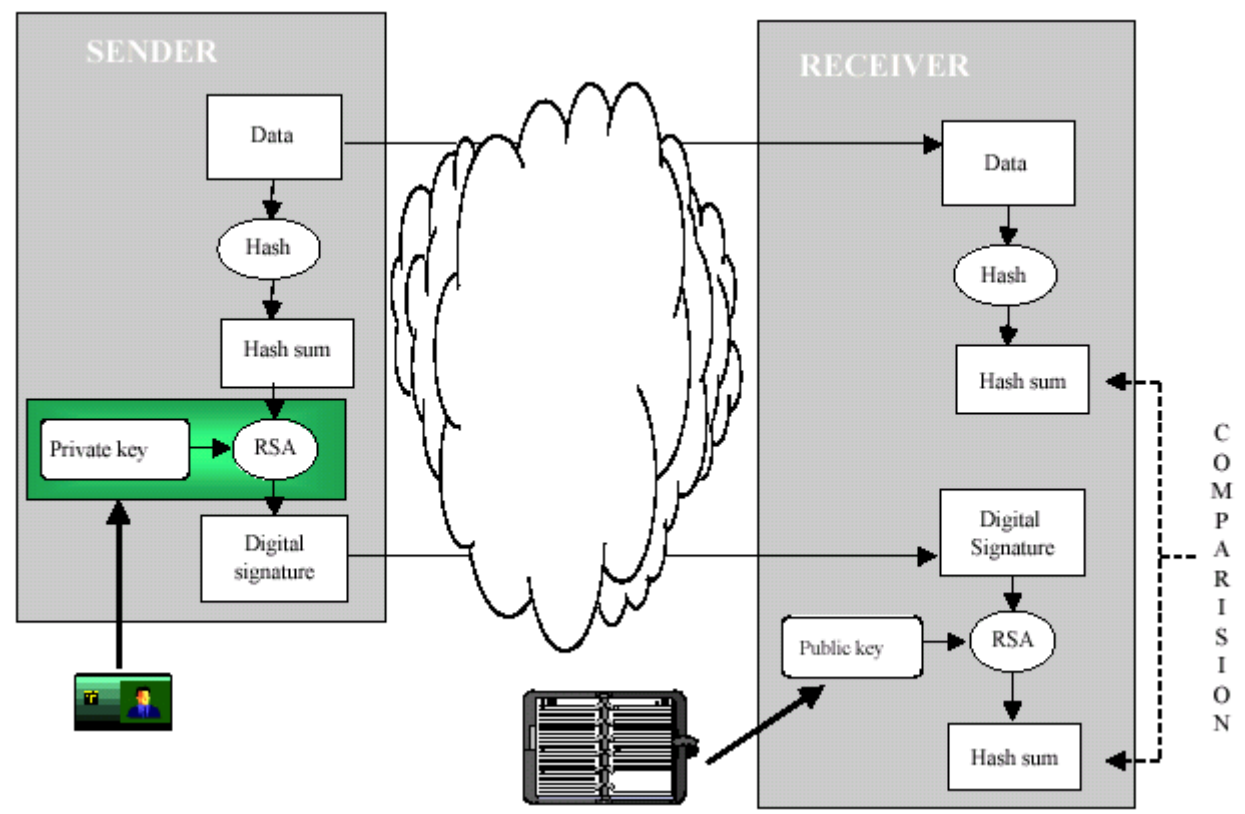
# Digital ID

- Asymmetric key-pair
  - public key
  - private key


- X.509 certificate
  - ISO standard
  - public key
  - credentials



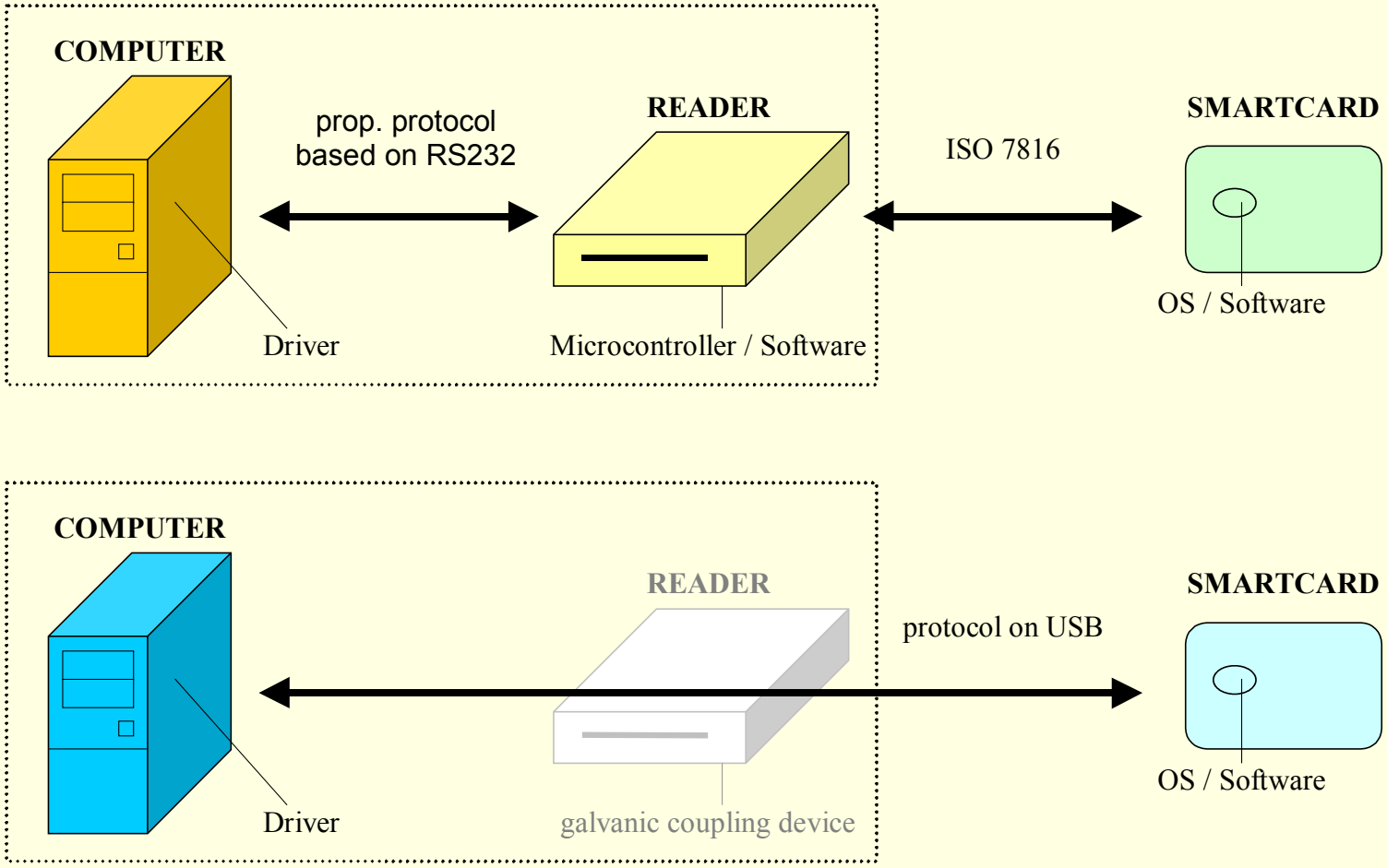
# Smart card application example: Digital Signature



# Smart card in heterogeneous environments

- Smart cards need readers and drivers
- Readers
  - desktop or embedded (keyboard, floppy slot)
  - optional display and keypad
  - PC world ready for installation
  - Mac, Unix & Linux ‘waiting’ for USB
- Drivers
  - PC/SC standard for Windows PC
  - custom developments

# ISO 7816 vs USB



# Certificate Portability: Smart Cards Holding Certificates

- Pros
  - Tamper-proof device
  - Portable
  - Visible security/theft indicator
  - Upgradeable
  - Branding, Photos,Mag-stripe
  - Biometric cards, readers coming
- Cons
  - More expensive than a pure software solution
  - infrastructure
  - Standards issues
  - Multi-application issues

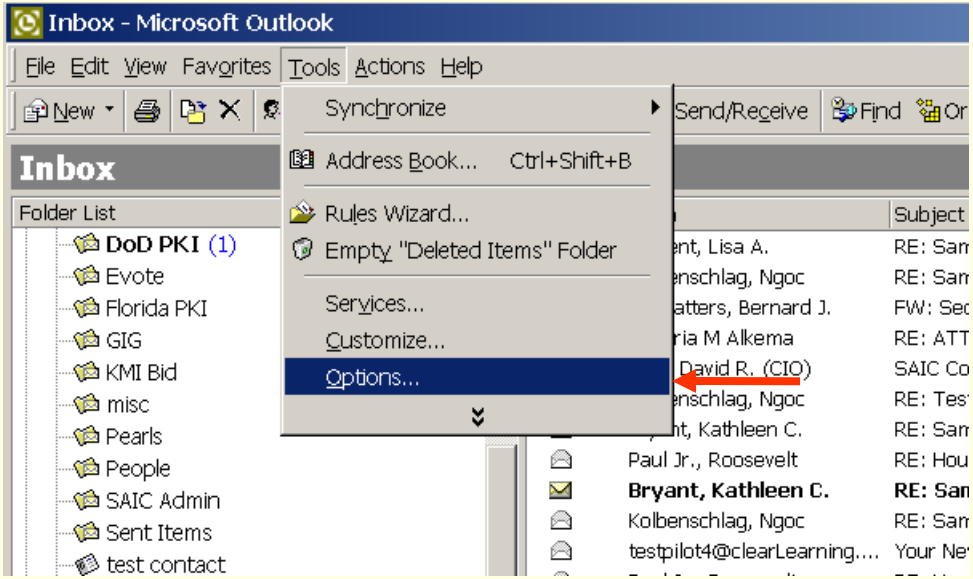
# Pay-TV, did you know it's PKI ?

- Pay-TV systems installed worldwide
  - 22 millions customers
  - pay-per-view
  - electronic purse
  - Internet
- Managed and secured with a very high proprietary secured PKI solution
  - based on a smartcard

# Signed and Encrypted Email – S/MIME

- S/MIME – Secure Multipurpose Internet Mail Extensions
- Prevent email spoofing
  - Helps preventing forged email
  - Helps preventing spam
- Protect sensitive messages & documents
- Secure business processes
  - Signed messages
  - S/MIME-based applications

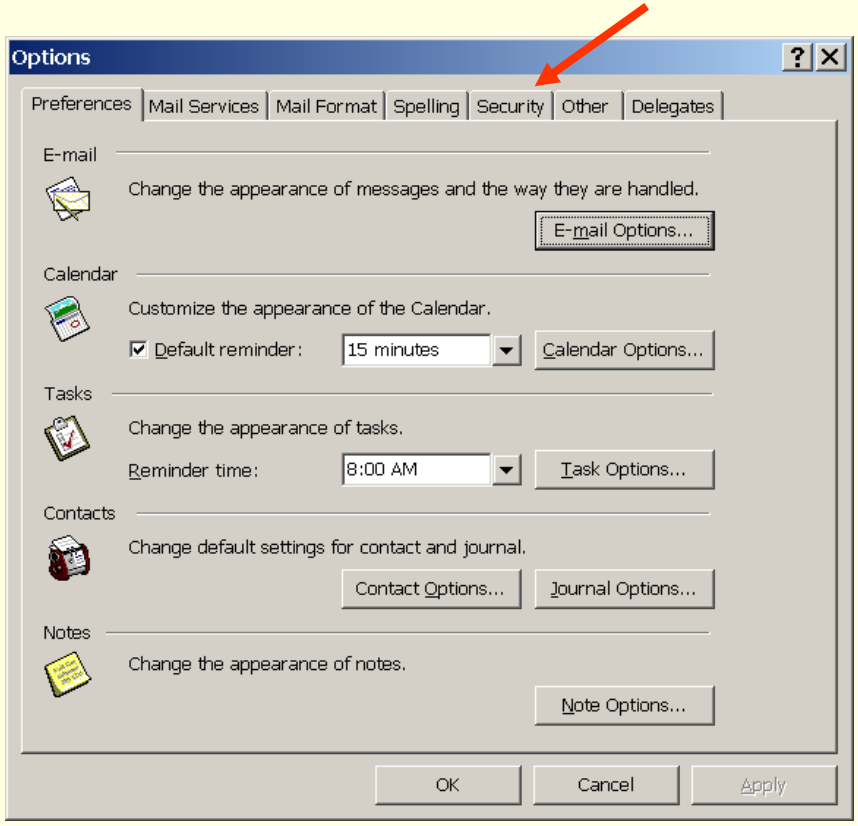
# Using PKI Certificates in Outlook (1)




1

Open **Outlook**. Select **Tools** from the main menu then choose **Options** from the drop-down menu.

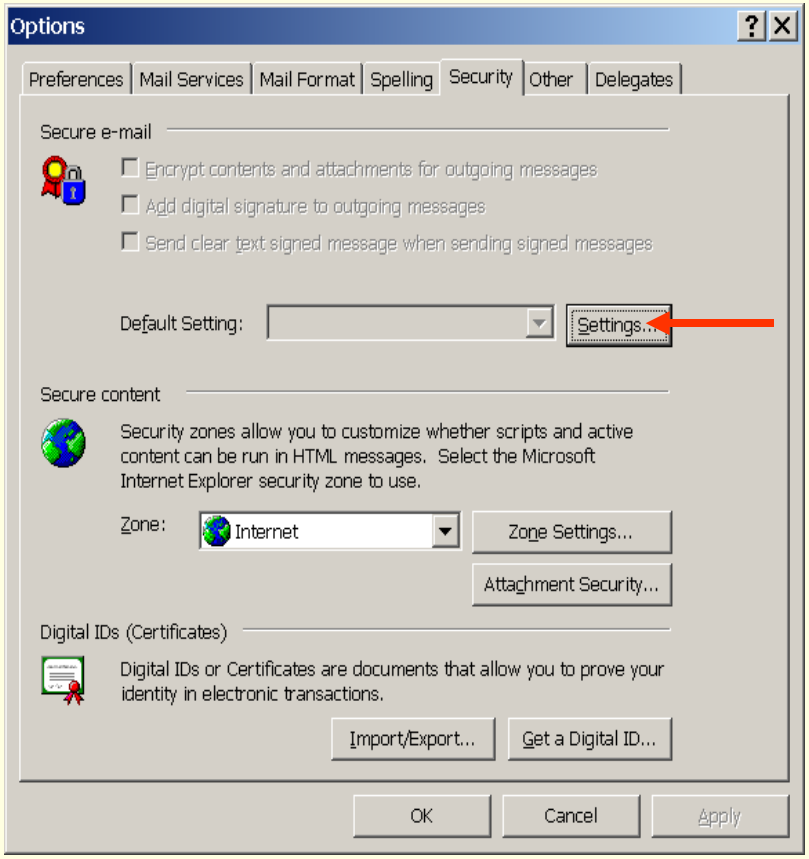
# Using PKI Certificates in Outlook (2)




2

Click on the **Security** tab.

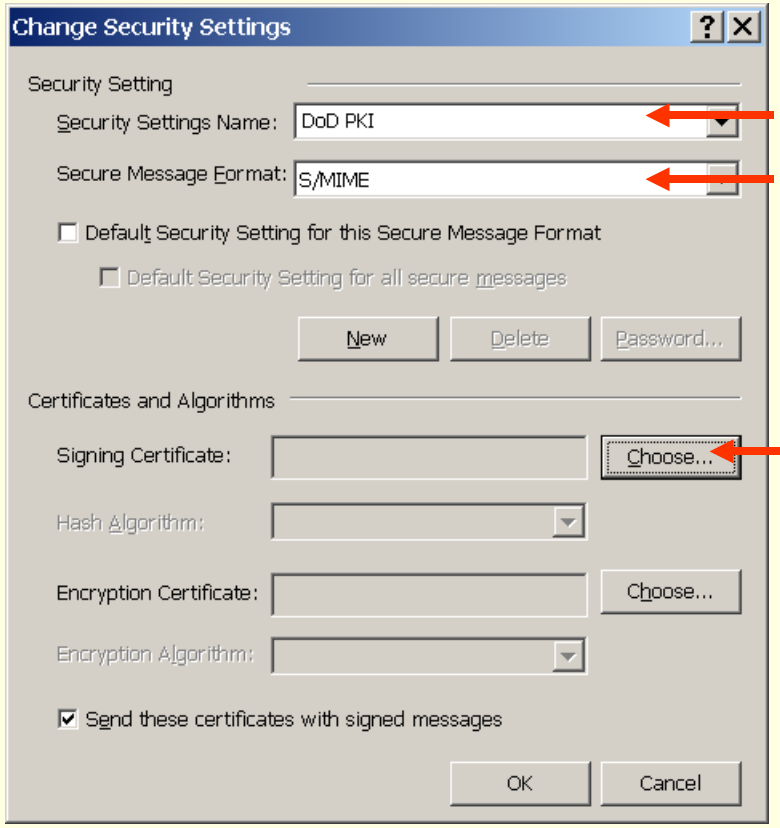
# Using PKI Certificates in Outlook (3)




3

Click the **Settings** button.

# Using PKI Certificates in Outlook (4)



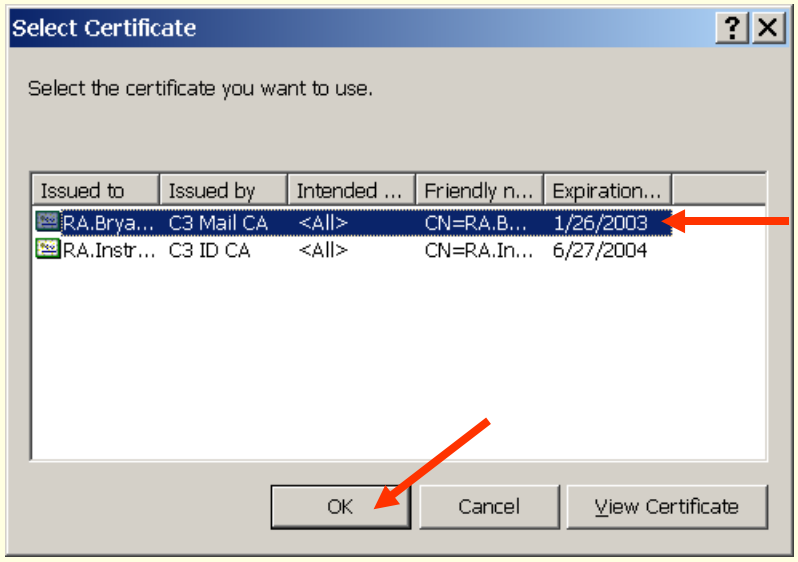

4

In the **Security Settings Name** field, enter a name for the new Security Setting .

Type **S/MIME** in the **Secure Message Format** field.

Click the **Choose** button next to the Signing Certificate field.

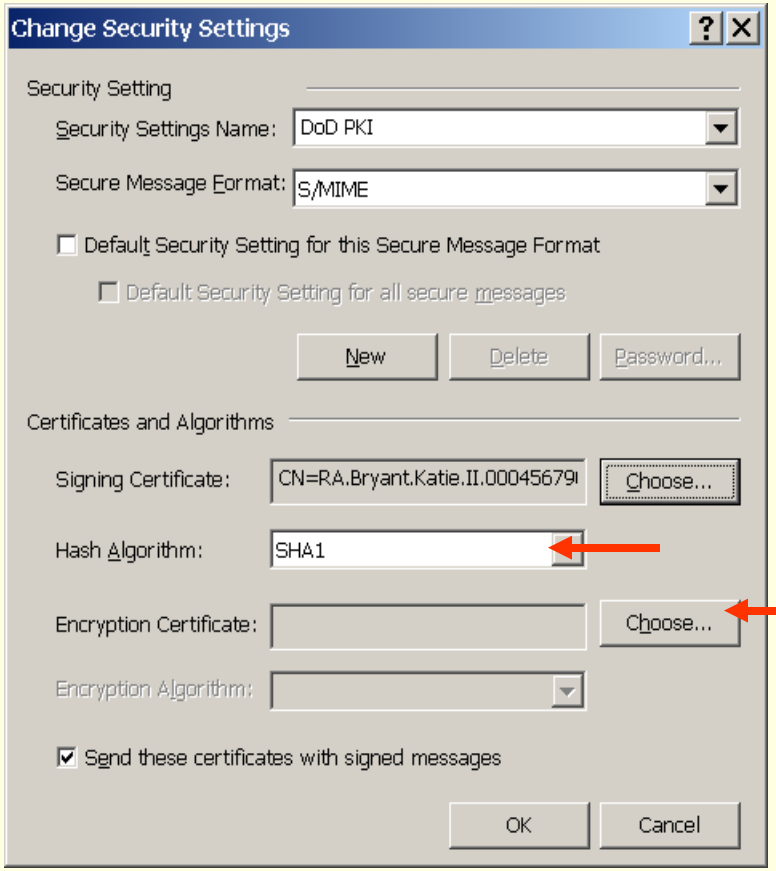
# Using PKI Certificates in Outlook (5)




5

Click on the certificate issued by **C3 Mail CA**. This is your Email Signing certificate. Click **OK**.

# Using PKI Certificates in Outlook (6)

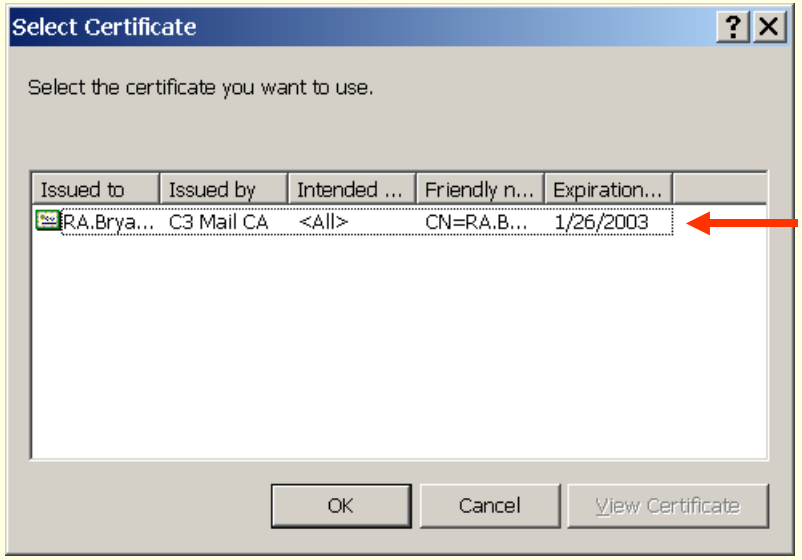



6

Choose **SHA1** from the **Hash Algorithm** drop down menu.

Click on the **Choose** button next to the **Encryption Certificate** field.

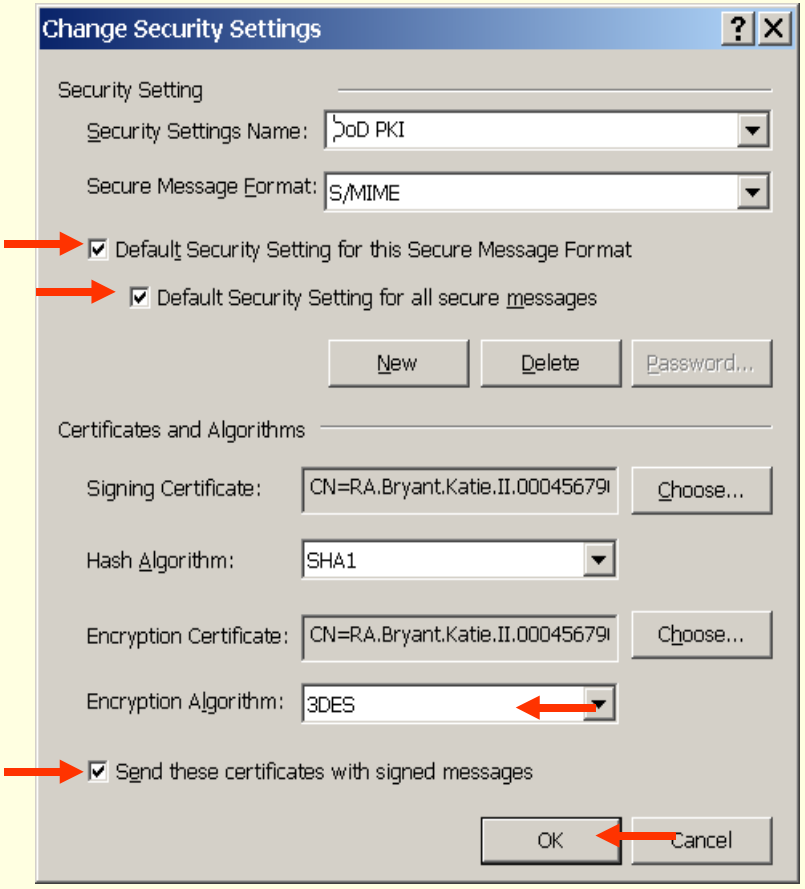
# Using PKI Certificates in Outlook (7)




7

Click on the certificate issued by **C3 Mail CA**. This is your Email Encryption certificate. Click **OK**.

# Using PKI Certificates in Outlook (8)



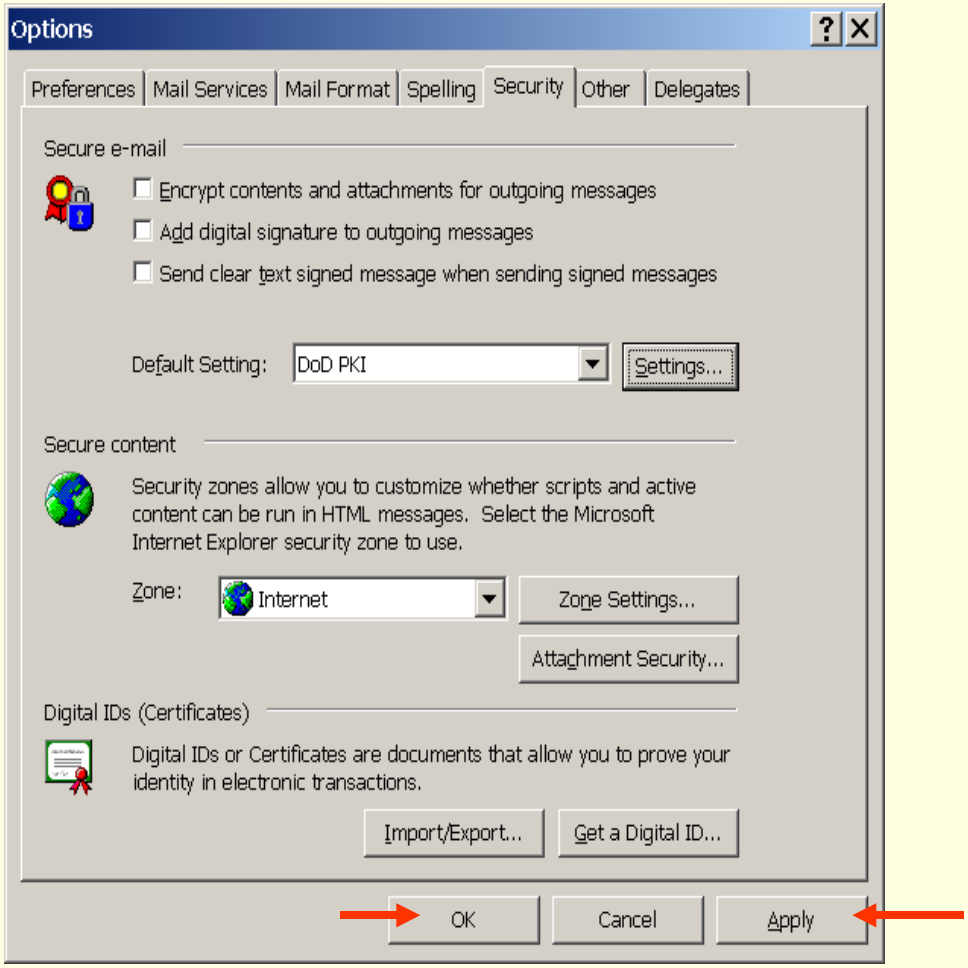

**8**

Choose **3DES** from the Encryption Certificate drop down box.

Check all 3 boxes in the Change Security Settings window.

Click **OK**.

# Using PKI Certificates in Outlook (9)




9

Click the **Apply** button then click **OK**.

# Questions?