ENEE 459-C
Computer Security

Digital signatures and certificate authorities

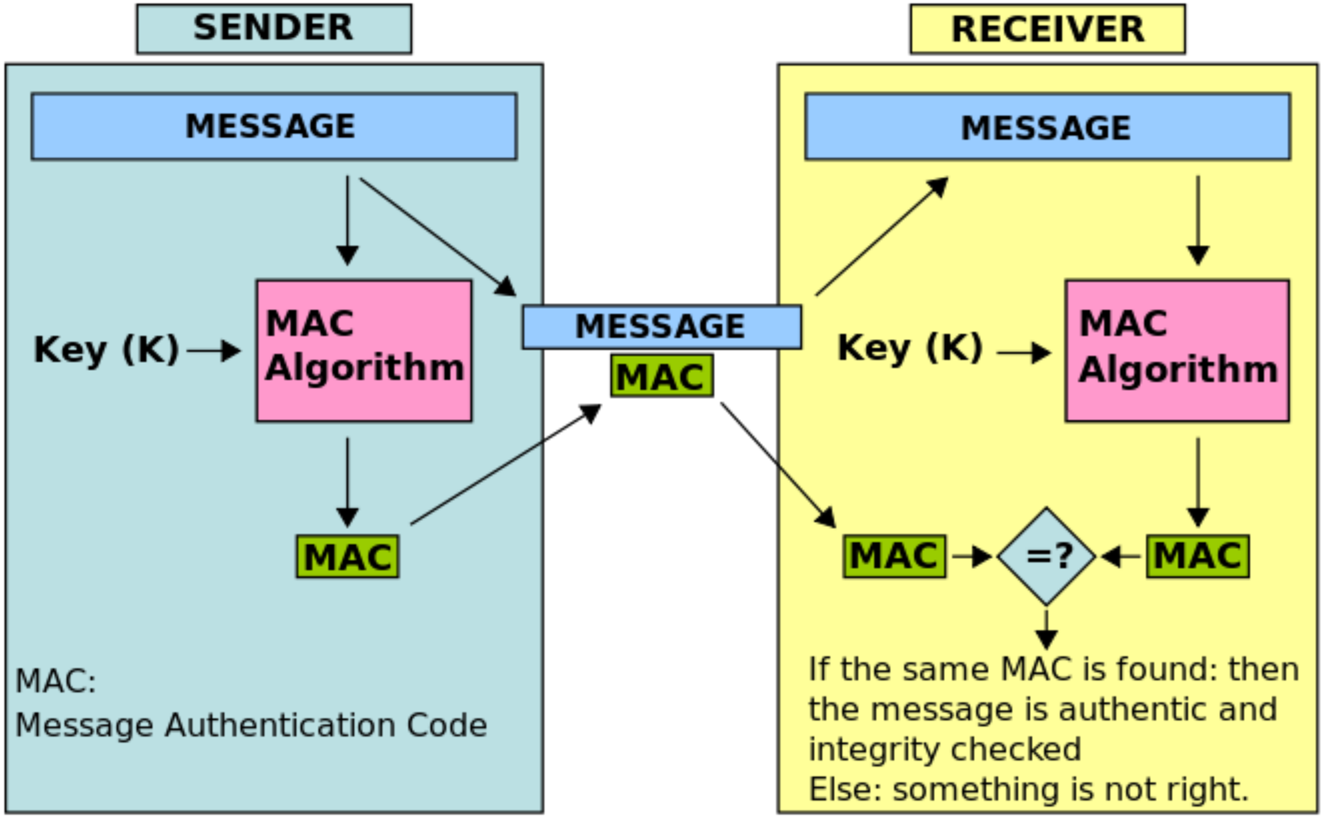
UNIVERSITY OF
MARYLAND

Signatures: The Problem

- Consider the real-life example where a buyer pays by credit card and signs a bill
- The buyer, however, later can potentially deny his signature
- Easy to fake signatures
- Can we have a service in the electronic world where it is difficult to fake a signature?

MACs for signing in the digital world!

- MAC: One party generates MAC, one party verifies integrity.
- Provides:
  - Authentication, Data integrity



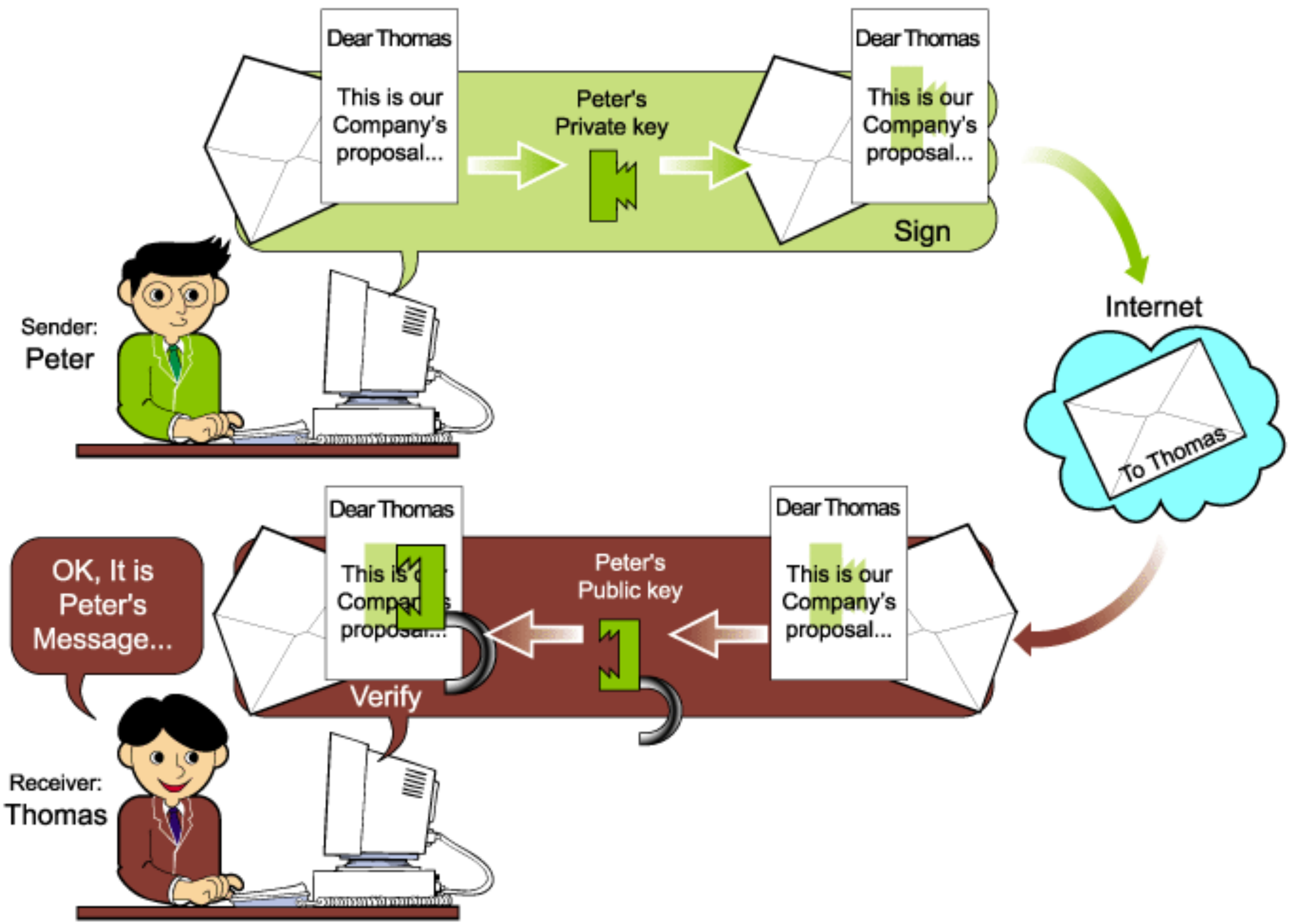
In the public key world: Digital Signatures

- What is a digital signature?
  - A data string which associates a message with some originating entity.
- Digital signatures: One party generates signature, many parties can verify.
- Algorithms:
  - a **signing algorithm**: takes a **message** and a **private key**, outputs a **signature**
  - a **verification algorithm**: takes a **public key**, a message, and a signature and it outputs ACCEPT or REJECT
- Provides:
  - Authentication, Data integrity, **Non-Repudiation**

Non-repudiation

- Nonrepudiation is the assurance that someone cannot deny something. Typically, nonrepudiation refers to the ability to ensure that a party cannot deny the authenticity of their signature on a document
- Do MACs offer non-repudiation?

Sign and verify



Security property

- Same with MACs
- Existential unforgeability
- You give the public key to the attacker
- The attacker asks for signatures S1,S2,…,Sn of messages M1, M2, …,Mn of his liking
- The attacker should not be able to output a message $M' \notin \{M1, M2, \ldots, Mn\}$ and a signature S' such that Verify(M',S')=1

RSA Signature

Key generation (as in RSA encryption):

- Select 2 large prime numbers of about the same size, p and q
- Compute $n = pq$, and $\varphi(n) = (q - 1)(p - 1)$
- Select a random integer e, $1 < e < \Phi$, s.t. $\gcd(e, \varphi(n)) = 1$
- Compute d, $1 < d < \varphi(n)$ s.t. $ed = 1 \bmod \varphi(n)$

Public key: (e, n) **used for verification**

Private key: d **used for generation**

RSA Signature algorithms

Signing message M

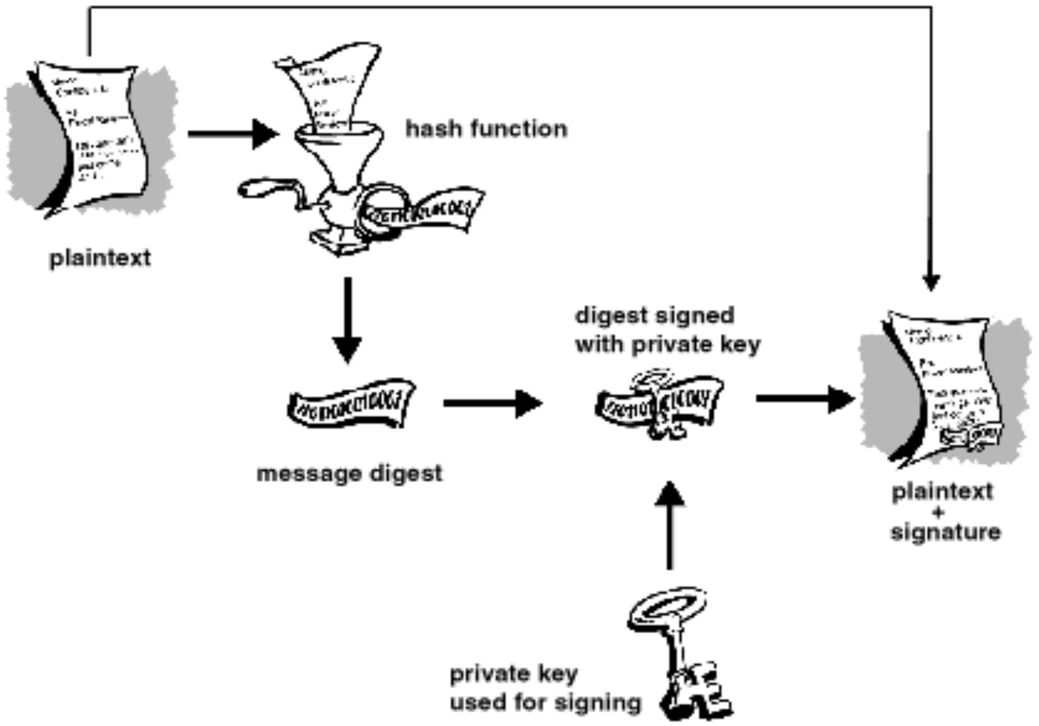
- Let h be a cryptographic hash function
- Compute sig = M^d mod n
- Send sig, M

Verifying signature S

- Use public key (e, n)
- Compute sig^e mod n = F
- If F=M output ACCEPT, else output REJECT

Digital Signatures and Hash

- Very often digital signatures are used with hash functions, hash of a message is signed, instead of the message.
- Hash function must be:
  - Preimage resistant, second-preimage resistant, Collision resistant



RSA Signatures with Hash

Signing message M

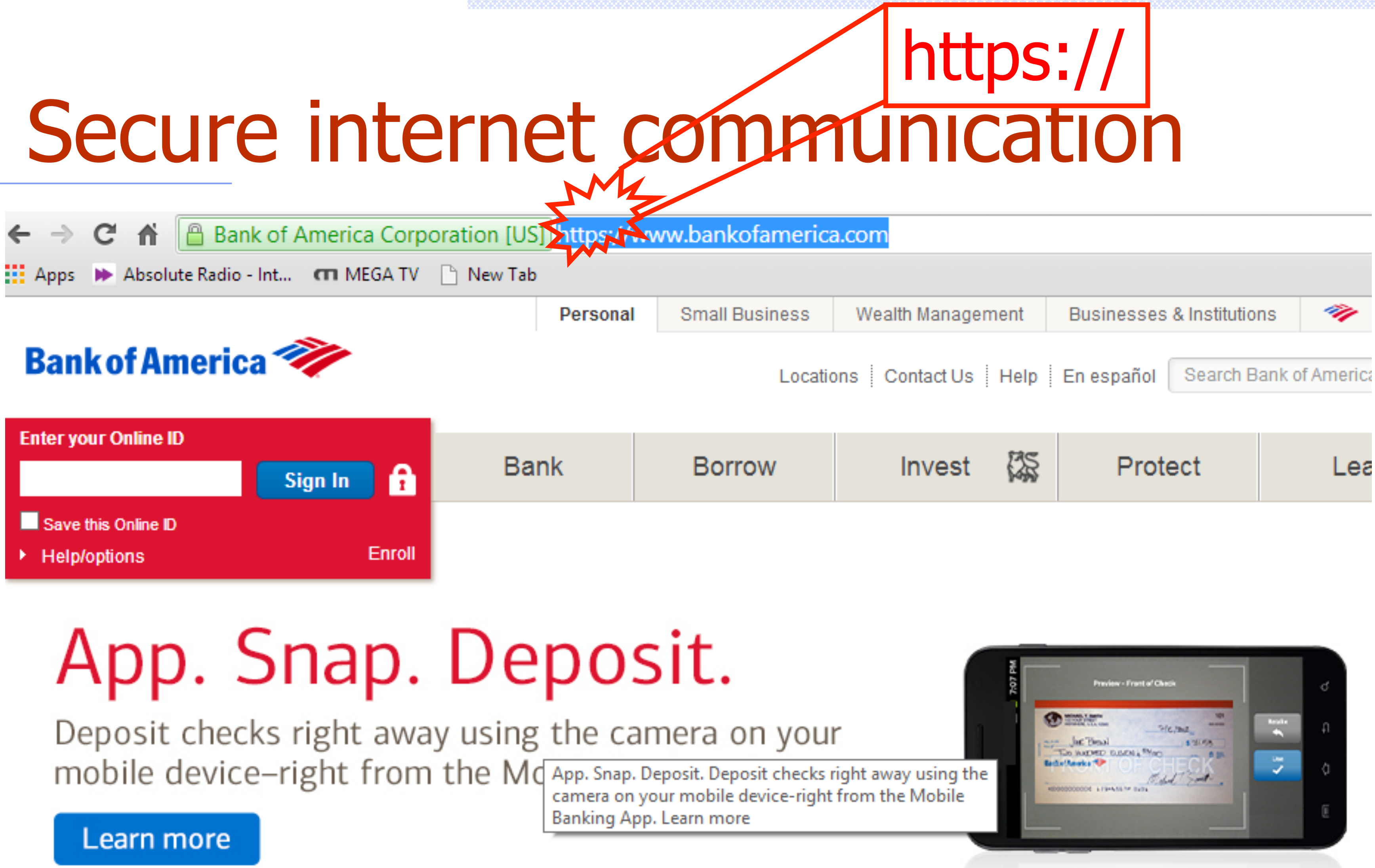
- Let h be a cryptographic hash function
- Compute sig = $h(M)^d \bmod n$
- Send sig, M

Verifying signature S

- Use public key (e, n)
- Compute $sig^e \bmod n = F$
- If F=h(M) output ACCEPT, else output REJECT

Certificates

Secure internet communication



What cryptographic keys are used to protect communication?

Public Keys and Trust

Public Key: P_A
Secret key: S_A

Public Key: P_B
Secret key: S_B

How are public keys stored?

How to obtain the public key?

How does Bob know or 'trusts' that P_A is Alice's public key?

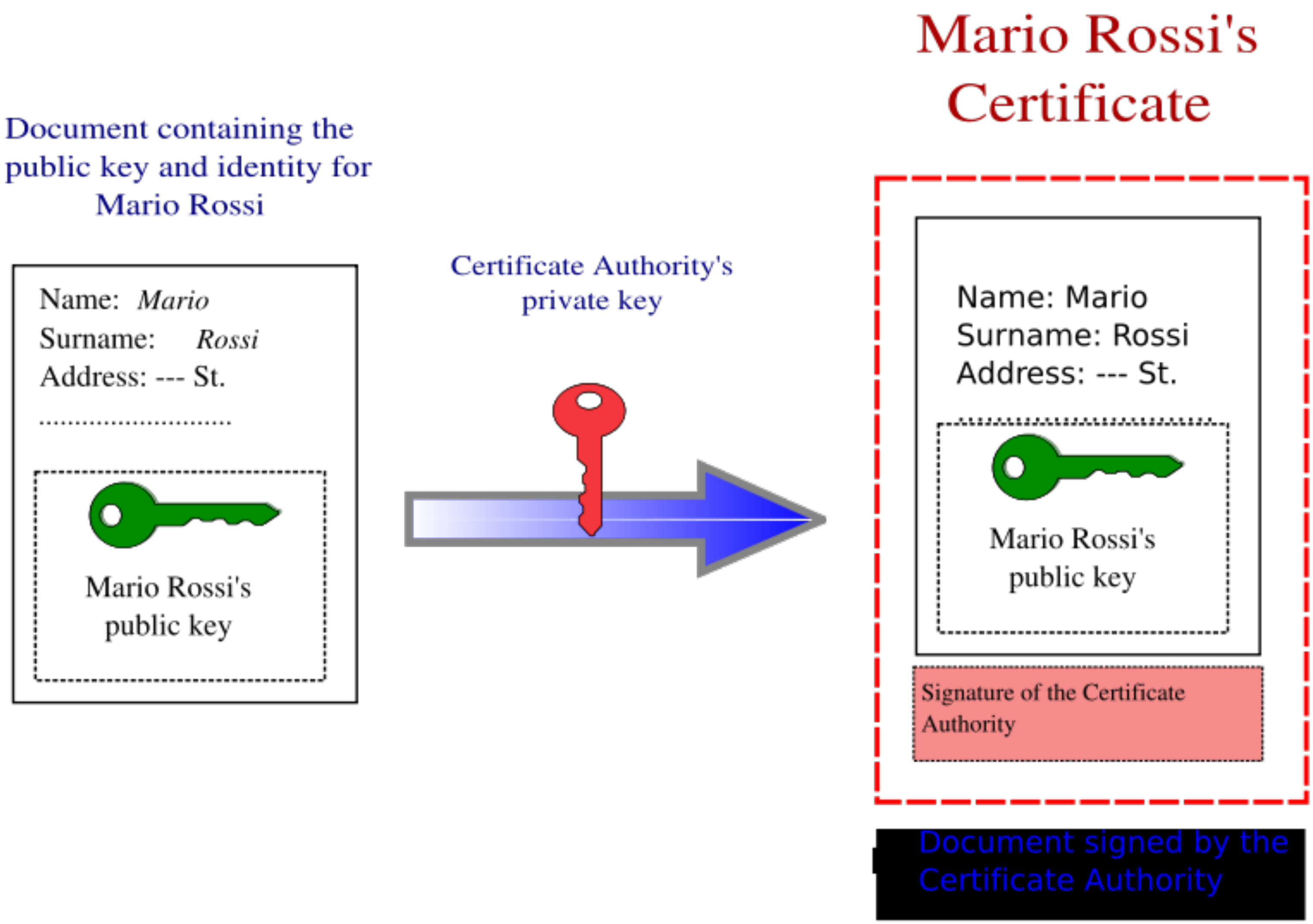
Distribution of Public Keys

- **Public announcement**: users distribute public keys to recipients or broadcast to community at large
- **Publicly available directory**: can obtain greater security by registering keys with a public directory
- Both approaches have problems, and are vulnerable to forgeries

Public-Key Certificates

- A certificate binds identity (or other information) to public key
- It is a signature on a statement “Paul’s public key is 1032xD”
- Contents digitally signed by a trusted Public-Key or Certificate Authority (CA)
  - Can be verified by anyone who knows the authority’s public-key
- For Alice to send an encrypted message to Bob, obtains a certificate of Bob’s public key

Details


Mario Rossi's
Certificate
Document containing the
public key and identity for
Mario Rossi
Name: Mario
Surname: Rossi
Address: --- St.
....
Mario Rossi's
public key
Certificate Authority's
private key
Name: Mario
Surname: Rossi
Address: --- St.
Mario Rossi's
public key
Signature of the Certificate
Authority
Document signed by the
Certificate Authority

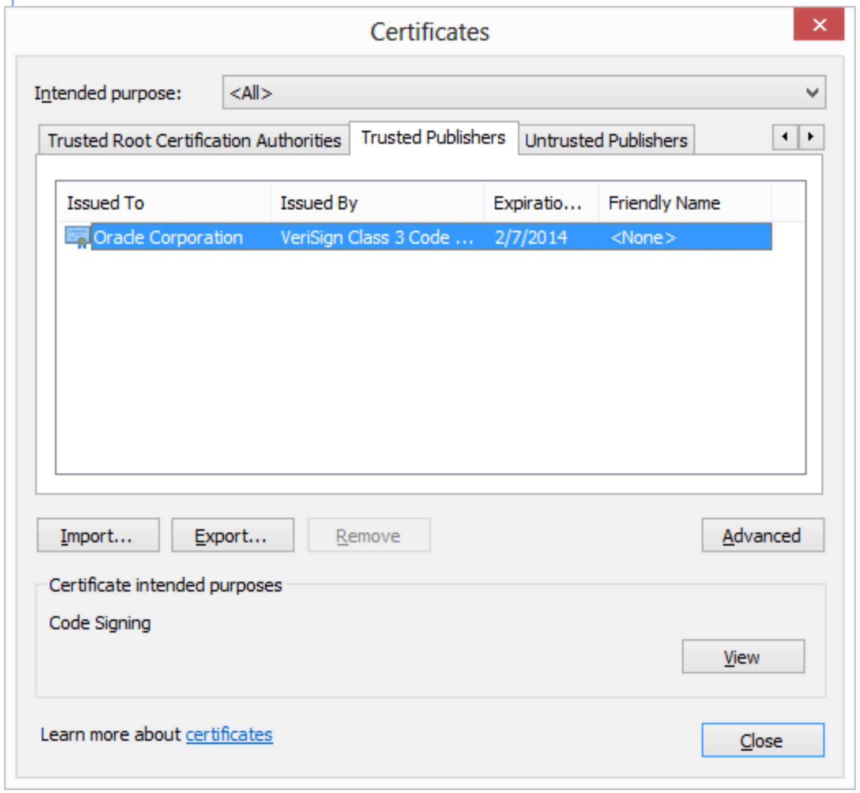
Do you trust your public key?

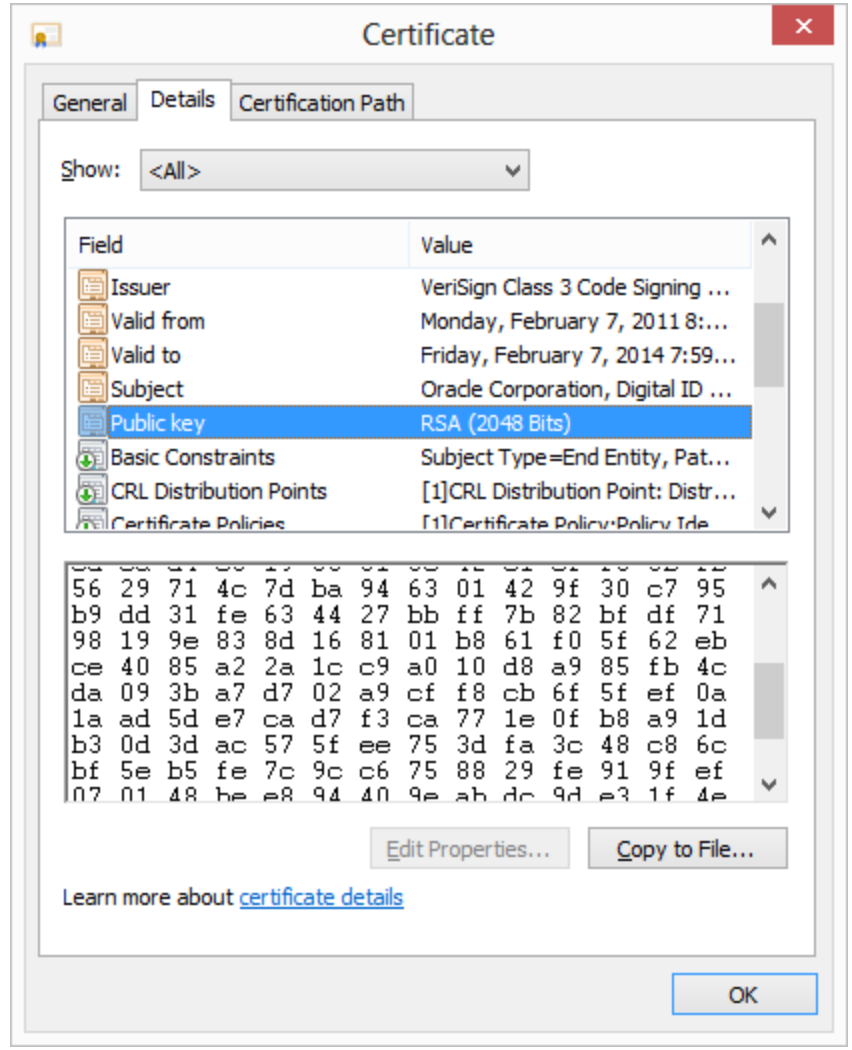
- Impostor Claims to be a True Party
  - True party has a public and private key
  - Impostor also has a public and private key
- Impostor sends impostor’s own public key to the verifier
  - Says, “This is the true party’s public key”
  - This is the critical step in the deception

X.509 Certificates

- Defines framework for authentication services:
  - Defines that public keys stored as **certificates** in a public directory.
  - Certificates are **issued and signed** by an entity called **certification authority (CA)**
- Used by numerous applications: SSL
- Example: see certificates accepted by your browser

Example: Oracle's certificate

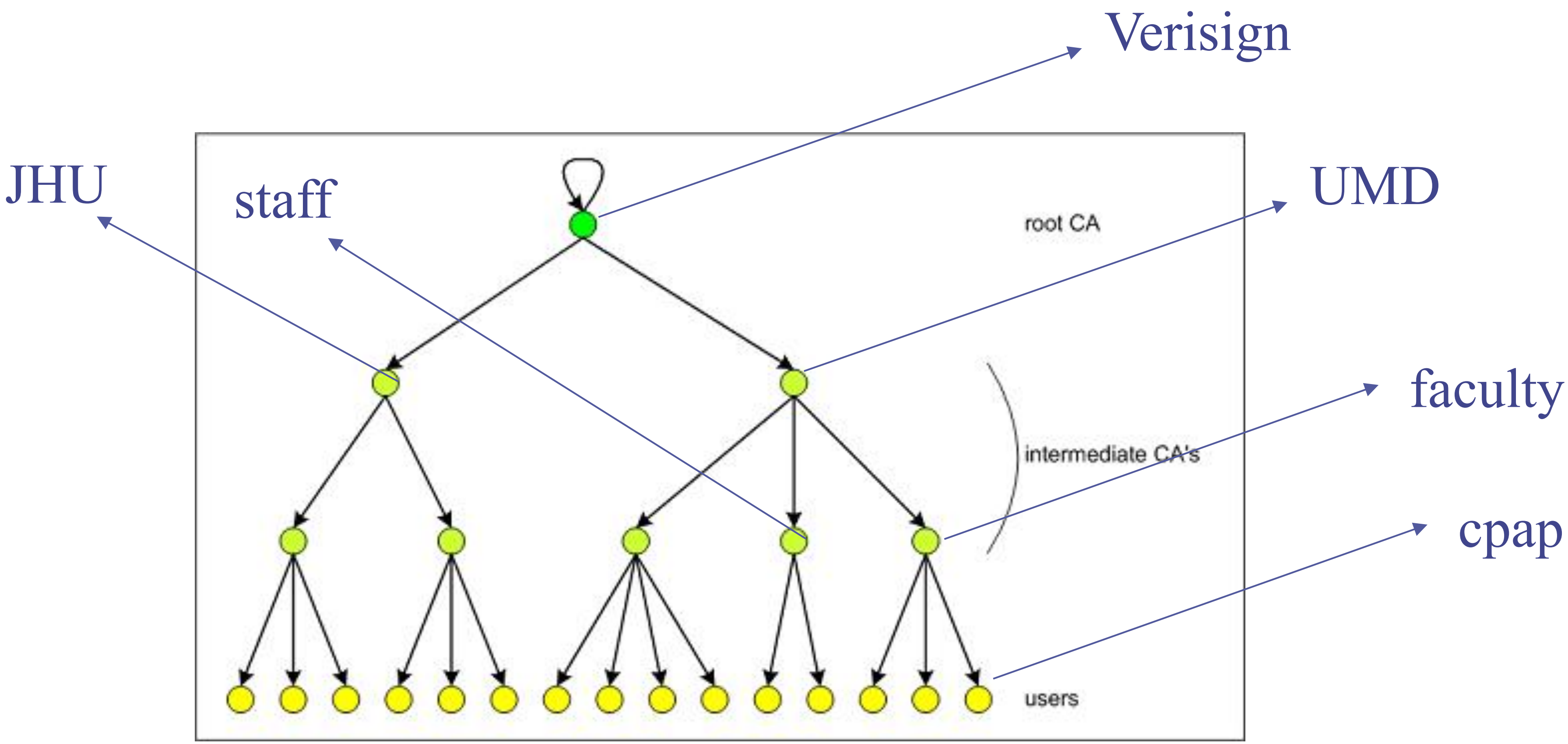




Certificate Hierarchy

- Single CA certifying every public key is impractical
- Instead, use trusted root authorities
- Root CA signs certificates for intermediate CAs, they sign certificates for lower-level CAs, etc.
  - Certificate "chain of trust"
    - $sig_{Verisign}$("UMD", PK_{UMD})
    - sig_{UMD}("faculty", $PK_{faculty}$)
    - $sig_{faculty}$("cpap", PK_{cpap})

Example



What bad things can happen if the root CA system is compromised?