Introduction to Cryptography

Part 3: real world applications

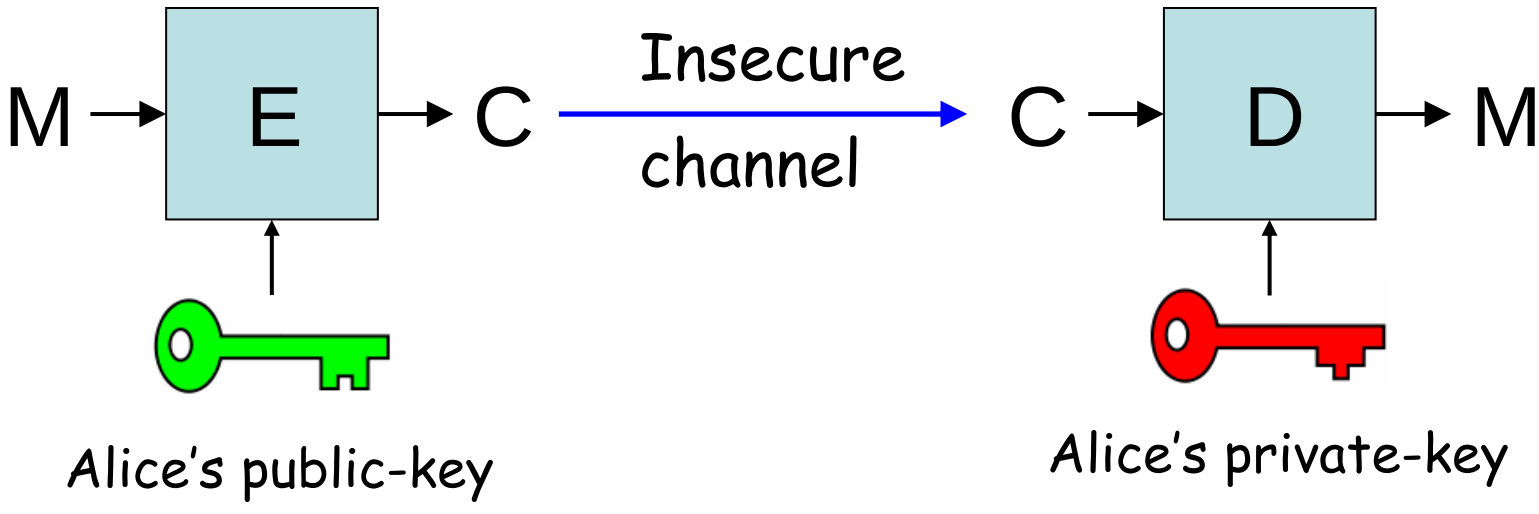
Jean-Sébastien Coron

January 2007

Public-key encryption

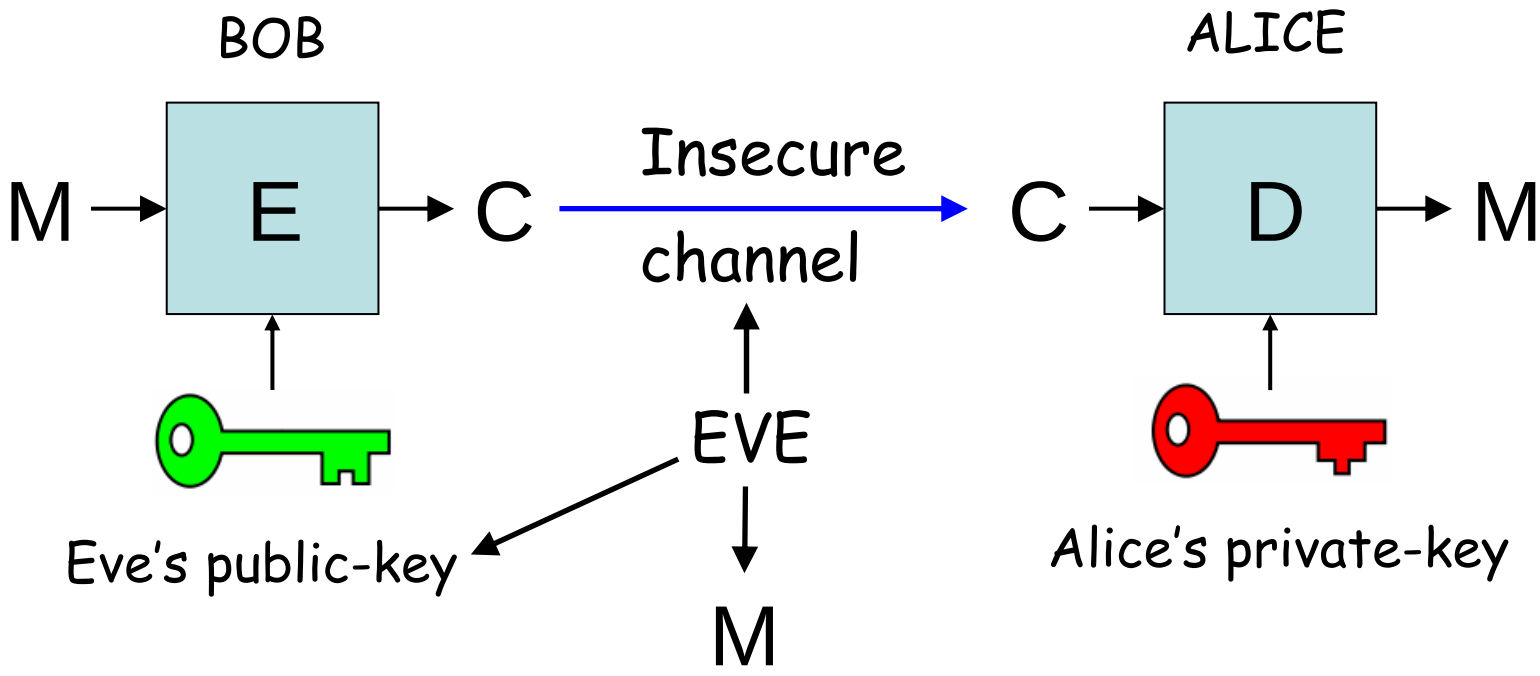
BOB

ALICE



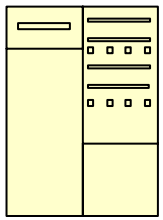
Authentication

- Public-keys need to be authenticated
  - Bob needs to be sure that the public-key belongs to Alice.
  - Otherwise, impersonation attack



Public-key Infrastructure

- A central authority binds public-keys to identities.
  - Public-key is stored in a certificate



Central authority

2) Alice authenticates and send PK_{Alice}

3) Certificate (ID_{Alice}, PK_{Alice})

1) Alice generates PK_{Alice} and SK_{Alice}

Public-key certificate

- Certificate:
  - the signature of the certificate authority binds together a public-key with an identity.
  - Bob can be sure that the public-key belongs to Alice by checking the signature using the CA public-key.
  - The CA is trusted by all participants.

Certificate Authority

- CA issues PK certificates that attest that the PK in the certificate belongs to the identity in the certificate
  - CA must verify user's identity before issuing certificate
  - If the CA's private key is compromised, security is lost.
- Largest providers of certificates
  - Verisign, Geotrust

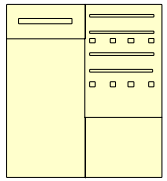
Public-key certificate

- A public-key certificate may include
  - user's public-key
  - name (person, computer, or company)
  - validity period.
  - location (URL) of a revocation center.
  - digital signature of the certificate, produced by the CA's private key.

Certificate revocation

- Certificate revocation when
  - Private-key is compromised
  - Identity/PK binding incorrect.
- A user should always check the validity of a certificate
  - CA can maintain a Certificate Revocation List (CRL)
  - Must be up to date and readily available

PKI encryption



CA authority

7. Certificate OK

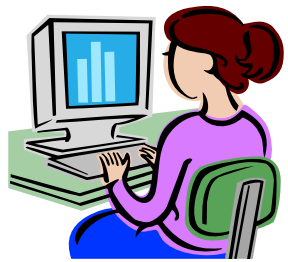
6. Bob verifies certificate. Bob checks for revocation

2. Alice authenticates and sends PK_{Alice}

3. Certificate (ID_{Alice}, PK_{Alice})

4. Bob requests Alice's certificate

5. Alice sends certificate



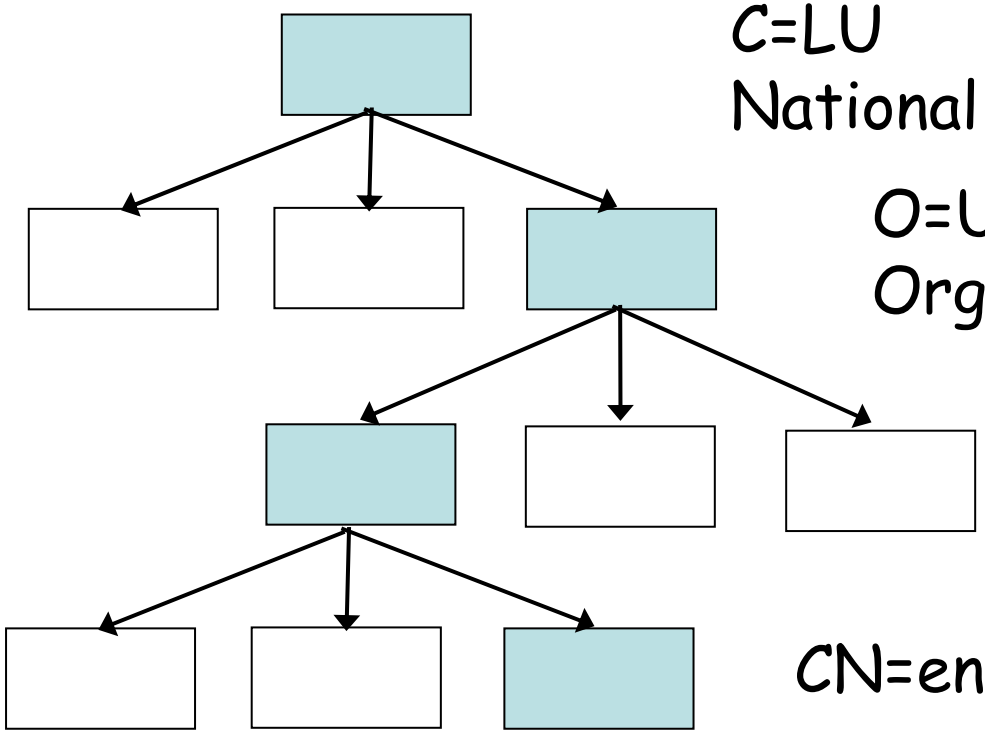
8. Bob send secure email using PK_{Alice}

1. Alice generates PK_{Alice}/SK_{Alice}

Hierarchy of certificates

- Bob may not know Alice's CA
  - The CA may be Alice's employer, and Bob may work for a different company.
- Alice's certificate can include her CA's public-key signed by a higher level CA_2
  - This CA_2 may be recognized by Bob
- This leads to a hierarchy of certificates

Certificate Hierarchy



C=LU
National CA, root certificate

O=University of Luxembourg
Organizational CA

OU=CS Department
Departmental CA

CN=end user

Certificate Standard

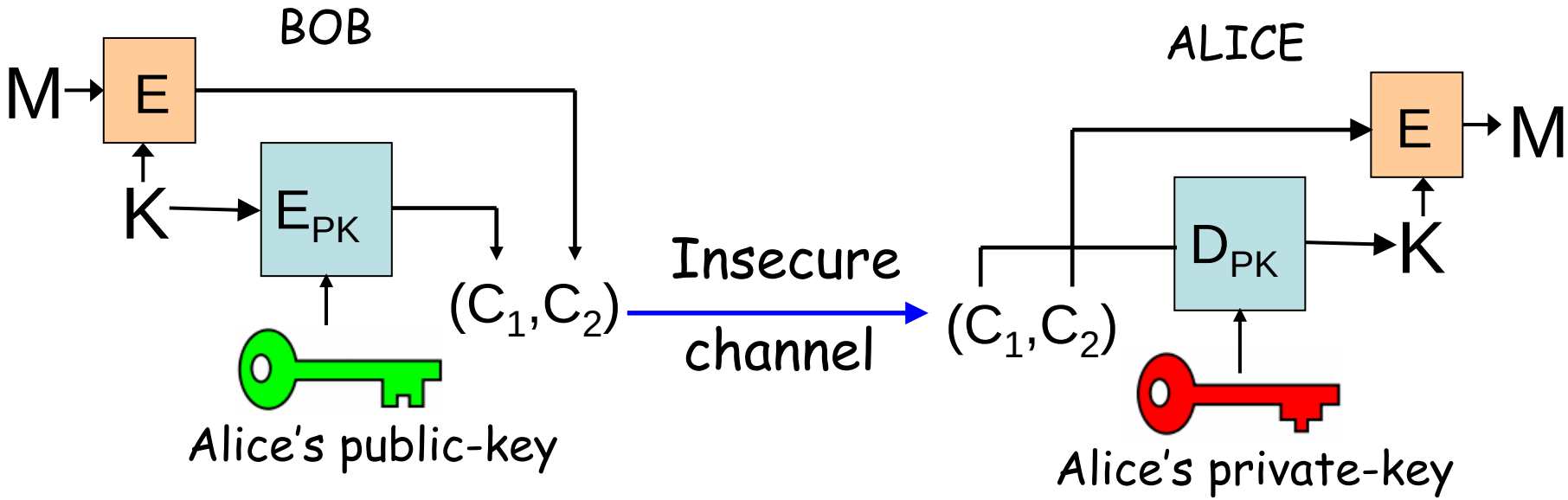
- X509
  - Most common certificate standard
  - Specifies certificate format and certificate validation path.
  - Assumes a hierarchy of CA
  - Root certificate is implicitly trusted
  - Specifies certificate revocation list (CRL) implementation

Root certificate

- Unsigned public-key certificate located at the top of a certificate chain.
  - Typically in X509 standard
  - Implicitly trusted
- Included in web browsers
  - Used for SSL/TLS connections
  - One needs to trust the browser's publisher to include correct root certificates.
  - Single point of failure
- In practice, hierarchy is flat.

PGP

- PGP (Pretty Good Privacy)
  - Software that provides email encryption and signature (and more).
  - First version by P. Zimmermann in 1991.
  - Uses PK encryption to encrypt a shared key, which is used to encrypt the message.

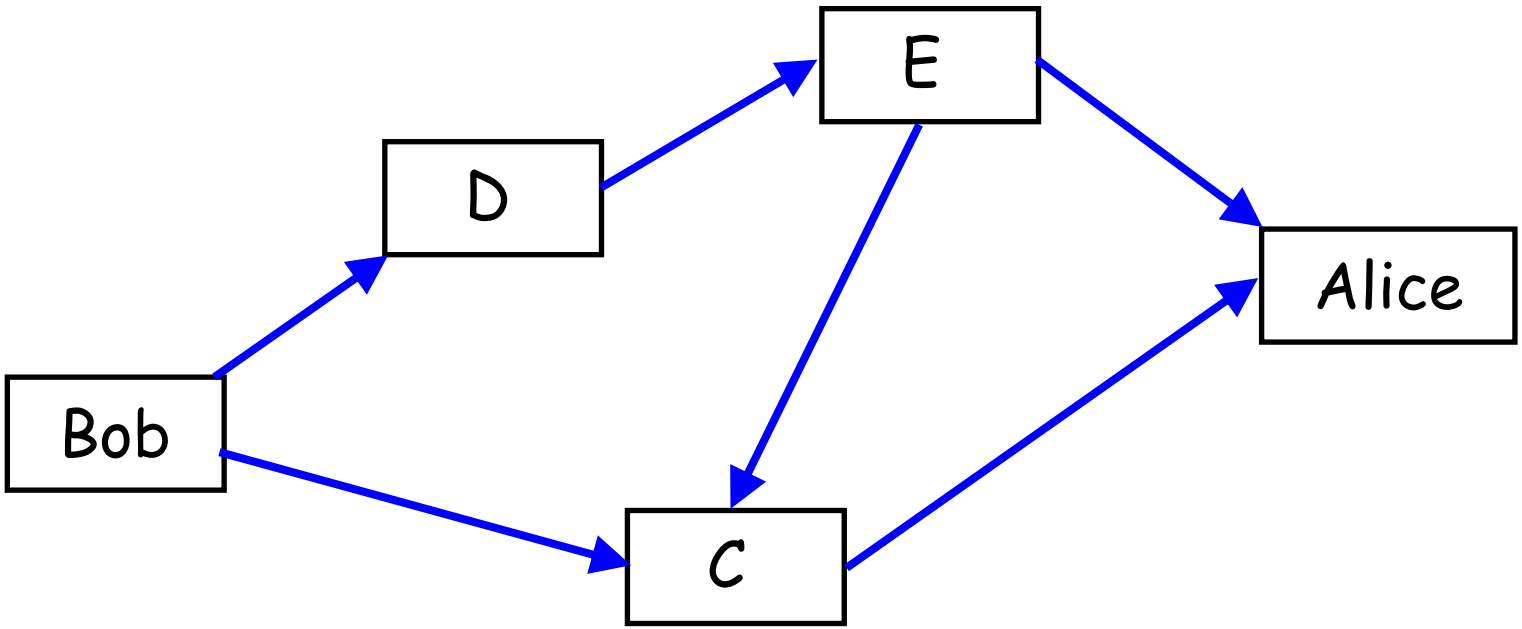


PGP

- Digital signature
  - When sending a message m, Bob can sign m with his private key.
  - Alice checks the signature with Bob's PK, so that Alice is convinced that m was sent by Bob and received unaltered.
  - RSA signature or DSA signature.
  - Used by default with encryption, but can be used for plaintext as well

PGP Web of trust

- Any party can sign the (PK,ID) of another.
- Decentralized web of trust



OpenPGP and GnuPG

- OpenPGP
  - Standard for PGP encryption since 1997.
  - Avoids patented algorithms
- GNU Privacy Guard (GnuPG)
  - developed by Free Software Foundation and freely available with source code.
  - Supports ElGamal, DSA, RSA, AES, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 and TIGER.

SSL

- Used to provide secure web-browsing.
  - SSL 3.0 similar to TLS 1.0
  - ensures confidentiality, integrity and authenticity over the Internet.
- Generally, only the server is authenticated
  - Mutual authentication requires a PKI for the client.

SSL

- Three steps
  - Negotiation for algorithms used.
  - Certificate verification and PK encryption for session key.
  - Symmetric encryption for traffic encryption.

Cipher suite negotiation

- Client sends a ClientHello message to specify supported algorithms
  - For example, RSA, AES and HMAC-SHA-1
- Server sends a ServerHello message to specify its choice of algorithm.
  - Server adapts to client capabilities.

SSL: second phase

- Server sends certificate to client.
  - Generally, X509 certificate
- Server can request client certificate for mutual authentication
  - Rarely used in practice
- Client and Server establish a « master secret »
  - by PK encryption of a random seed by the client (generally RSA)
  - or possibly by Diffie-Hellman key exchange (rarely used)

SSL second phase (2)

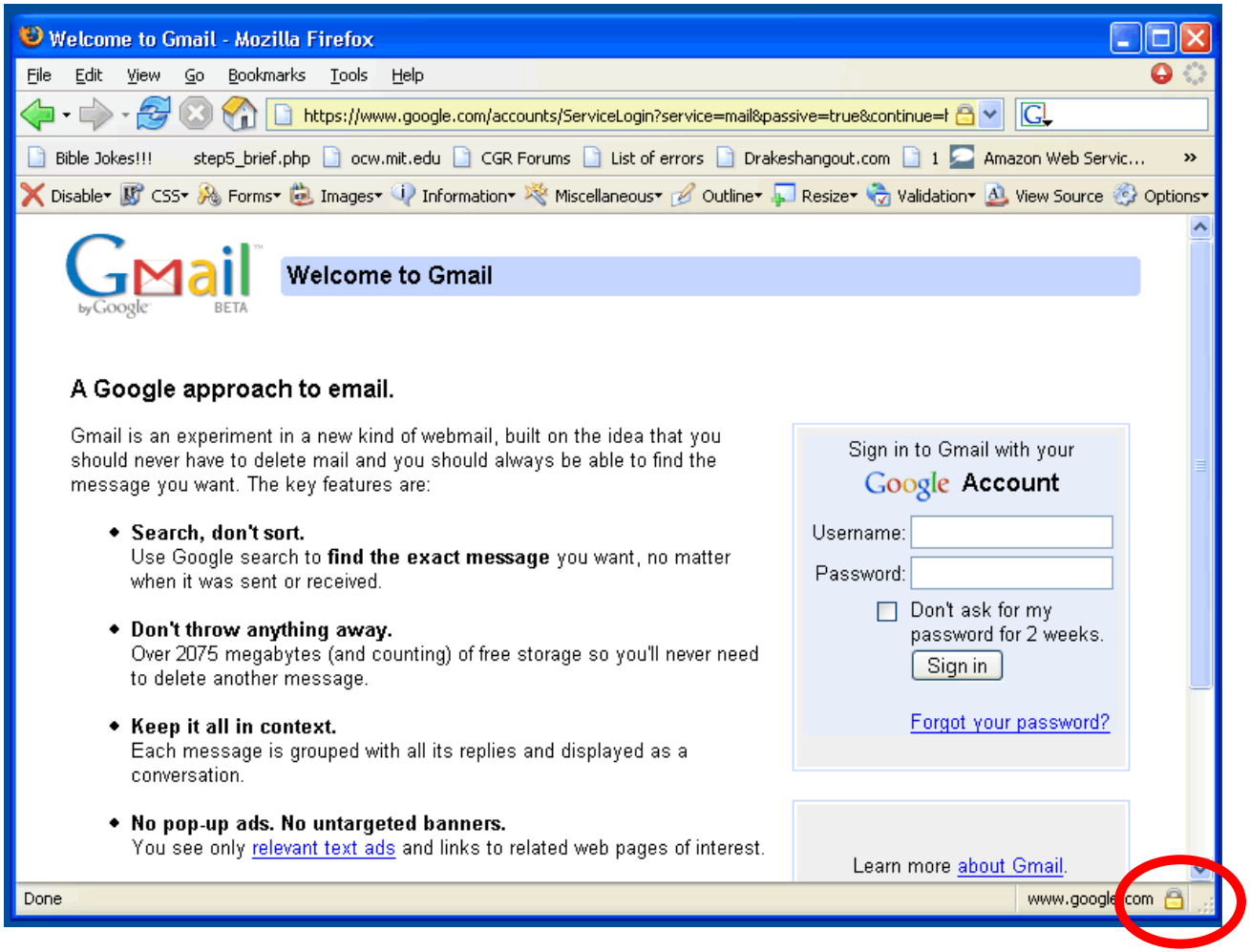
- Server authenticated by proof of possession of private key
  - Ability to decrypt client data.
  - Both sides share the same « master secret »
- Client/server finish
  - Authenticate all previously exchanged data with MACs

SSL: third phase

- Traffic encryption
  - Using symmetric cipher
  - Some early implementations of SSL used 40-bit keys because of US government restrictions on crypto export
  - Now relaxed export restrictions. Modern implementations use 128 bit keys for symmetric key.
- Integrity protection via MACs

Applications of SSL

- Mainly used to secure HTTP => HTTPS



Credit card via https

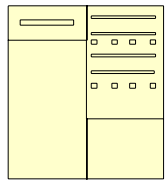
- https only protects the credit card number during transit between the user's computer and the server
  - Does not protect against an attack on the server
- Attack on the server usually easier than interception in transit.
  - Credit card number often saved in a database in merchant site
  - Attacks generally concentrate on the server and database

Identity-Based encryption

- Principle
  - Allows a party to encrypt a message using the recipient's identity as the public-key
  - The corresponding private key is provided by a central authority
- History
  - Concept invented by Shamir in 1984
  - First realization by Boneh and Franklin in 2001

IBE

- Bob sends an email to Alice using his identity as the public-key



Key server

2. Alice authenticates

3. Private-key for Alice

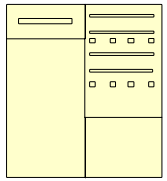
1. Bob encrypts with public-key **alice@a.com**

Bob send secure email

IBE

- Principle
  - Bob encrypts his email using Alice's email adress alice@a.com as the public key
  - Alice receives the message. She contacts the key server, authenticates, and receives her private key.
  - Alice uses her private-key to decrypt the message
  - This private-key can be used to decrypt any future message sent to Alice by Bob or any other user.

Difference with conventional PKI



CA authority

7. Certificate OK

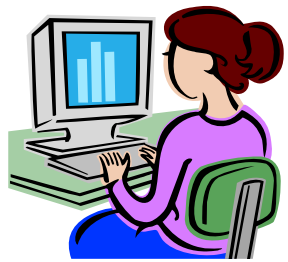
6. Bob verifies certificate. Bob checks for revocation

2. Alice authenticates and sends PK_{Alice}

3. Certificate (ID_{Alice}, PK_{Alice})

4. Bob requests Alice's certificate

5. Alice sends certificate



8. Bob send secure email using PK_{Alice}

1. Alice generates PK_{Alice}/SK_{Alice}

Adantages of IBE

- Simplification compared to PKI
  - No need to distribute PK certificates
  - Users can use their email adress as PK
  - Recipient does not have to be online to present PK certificate.
  - Sender does not have to be online to check validity of certificate
  - Bob can send an email to Alice even if Alice has not yet registered in the system

Boneh-Franklin

- First efficient IBE, proposed by Boneh and Franklin at Crypto 2001
  - Most famous IBE scheme to date.
  - Based on the bilinear pairing operation on an Elliptic-Curve.
  - Provably secure encryption scheme
  - IBCS#1 standard, published by Voltage Security.

Applications of IBE

- Email encryption
  - A company hosts the Private-Key generator (PKG) and distributes private-keys to its employees.
  - Employees can communicate securely between themselves, using their email address as their public-key
  - Nobody expect the mail recipient (and the PKG) can decipher the communications
  - Private-keys can also be distributed outside the company

Revocation of Public-keys

- Key-revocation in IBE is simple
  - Bob encrypts his email to Alice using the public-key « **alice@company.com ¦¦ current-year** »
  - Alice can only decrypt if she has obtained the private-key for the corresponding year.
  - With « **alice@company.com ¦¦ current-date** » instead, Alice must obtain a new private-key every day
  - Key revocation: the PKG simply stops issuing private-keys to Alice if Alice leaves the company. Then she can no longer read her email
- Encrypting into the future
  - With « **alice@company.com ¦¦ future-date** »

Conclusion

- Public-key Infrastructure
  - Necessary to authenticate public-keys
  - Difficult to set up and maintain
    - Certificate Revocation List
  - Used for PGP encryption and SSL/TLS.
- IBE could be an alternative
  - But central authority can decrypt everything.