Introduction to Cryptography

Jean-Sébastien Coron

November 2016

Outline

- History
- Public-key cryptography
  - RSA encryption, signatures, DH key exch.
- Security models and constructions
  - Public-key encryption and signatures
- Public-key infrastructures
  - Certificates, PGP, SSL

Mono-alphabetic Cipher

- Each letter is replaced with another letter, according to a fixed substitution

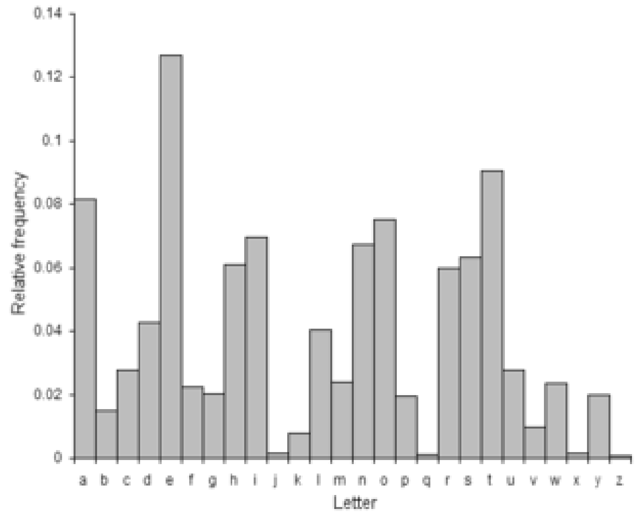
Plaintext :	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Ciphertext:	C G H U Z J T E L Y X I F O P K J W V A B D M S N Q

Then HELLO WORLD enciphers to EZIIP MPWIU

Number of possible keys is large: 26! =2^88.4 or 88 bits, but...

Frequency analysis

- Frequency of letters in English:



- Cryptanalysis of mono-alphabetic cipher:
  - The most frequent letter in the ciphertext is likely E,T or A.
  - Substitute and continue with less frequent letters.
  - WEAK

One-time pad (1917)

- Plaintext is xored with the key to produce the ciphertext

```
  011001011001
  111010010010
______________
  100011001011
```

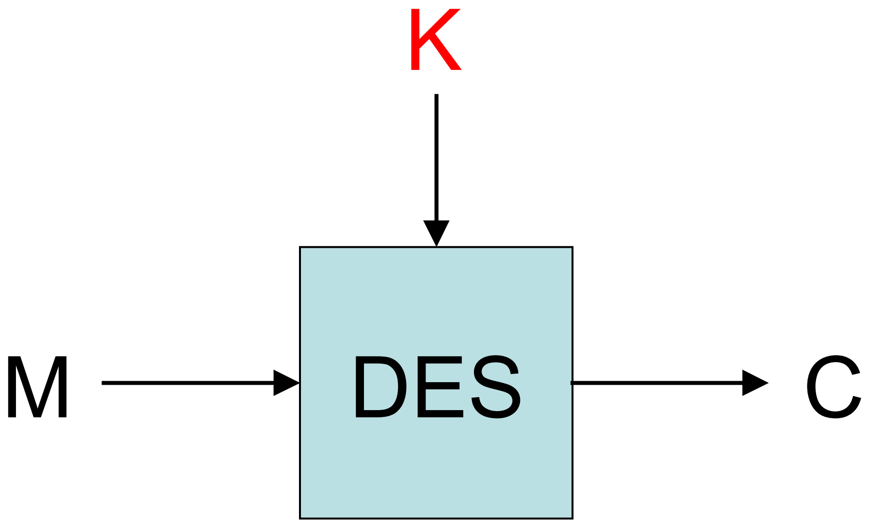
- Proved unbreakable by Shannon (1949) if key is random and as long as the plaintext.
- Issue: key as long as the plaintext.
- Used for the hotline between Washington and Moscow during the cold war.

DES (1976)

- Data Encryption Standard (DES), published as FIPS PUB 46.
- Developed by NBS (National Bureau of Standards), now NIST (National Institute of Standards and Technology), following an algorithm from IBM.
- De facto world-wide standard since 1976.
- Superseded by the AES, but remains in widespread use.

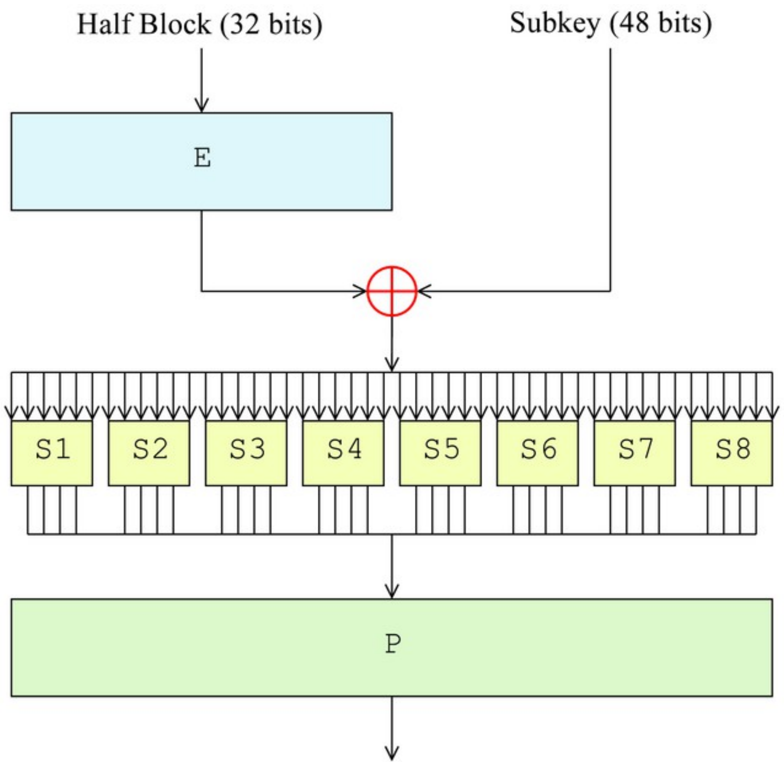
DES block-cipher

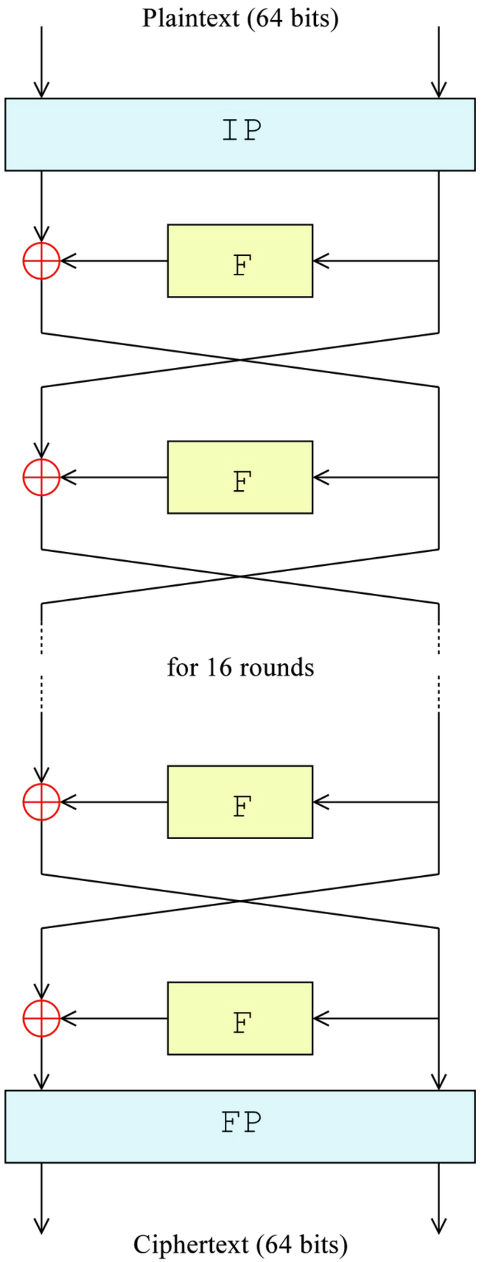
- Input length: 64 bits.
- Output length: 64 bits.
- Key length: 56 bits.



DES

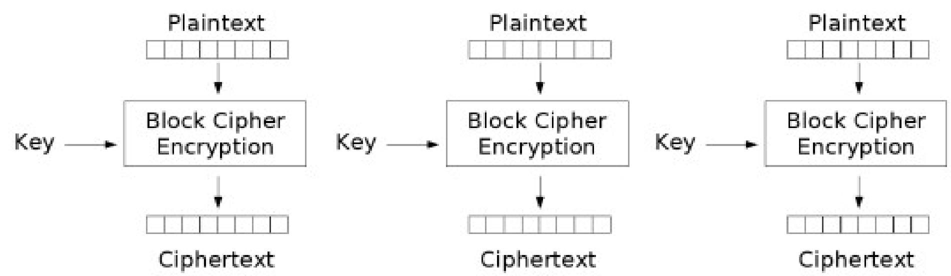
- Feistel Cipher
- F function:





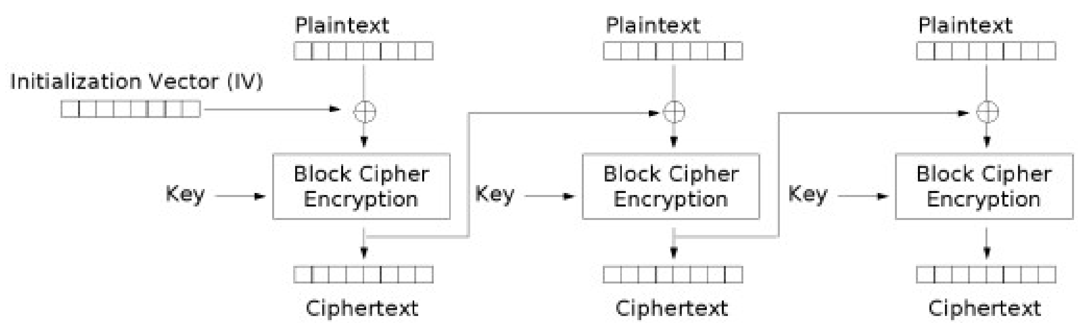
DES modes of operation

- Encrypting longer messages (>64 bits)
- FIPS-81: DES modes of operation
- ECB: WEAK



Electronic Codebook (ECB) mode encryption

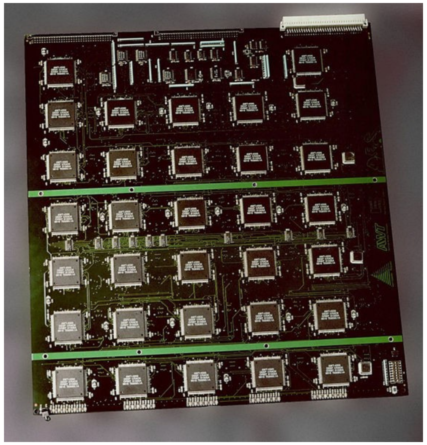
- CBC: OK



Cipher Block Chaining (CBC) mode encryption

Security of DES

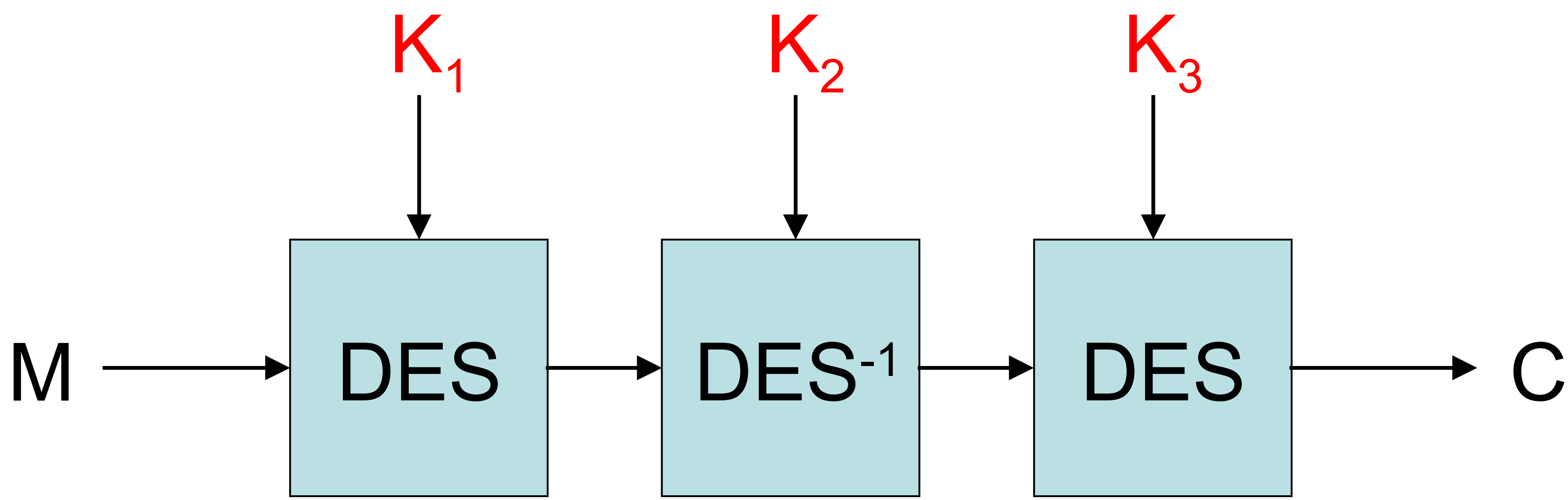
- Problem: key is too short (56 bits). Exhaustive search has become feasible
  - DES cracker from Electronic Frontier Foundation (EFF). Breaks DES in 2 days (1998).



- Other attacks
  - Differential cryptanalysis (Biham and Shamir). 2^47 chosen plaintexts
  - Linear cryptanalysis (Matsui, 1993). 2^43 known plaintexts.

TRIPLE DES

- Block cipher
  - 64-bit input and output, 168-bit key

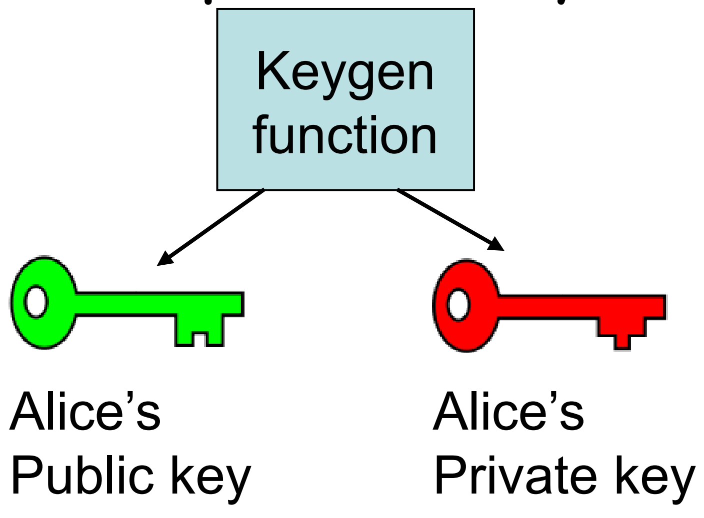


- Slowly disappearing, replaced by AES (6 times faster in software).

Public-key cryptography

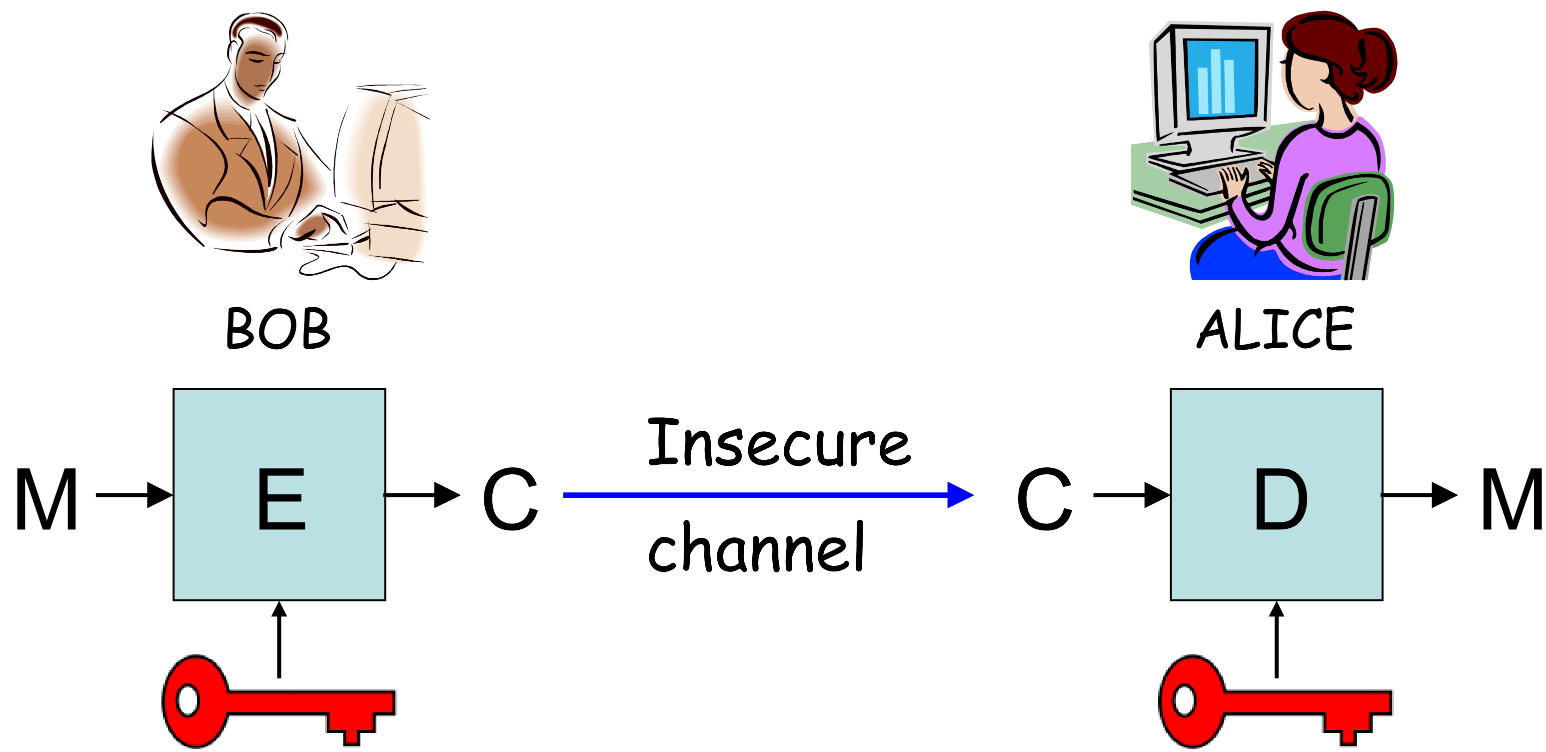
- Invented by Diffie and Hellman in 1976. Revolutionized the field.
- Each user now has two keys
  - A public key
  - A private key
- Should be hard to compute the private key from the public key.
- Enables:
  - Asymmetric encryption
  - Digital signatures
  - Key exchange
  - Identification,

  and many other protocols.



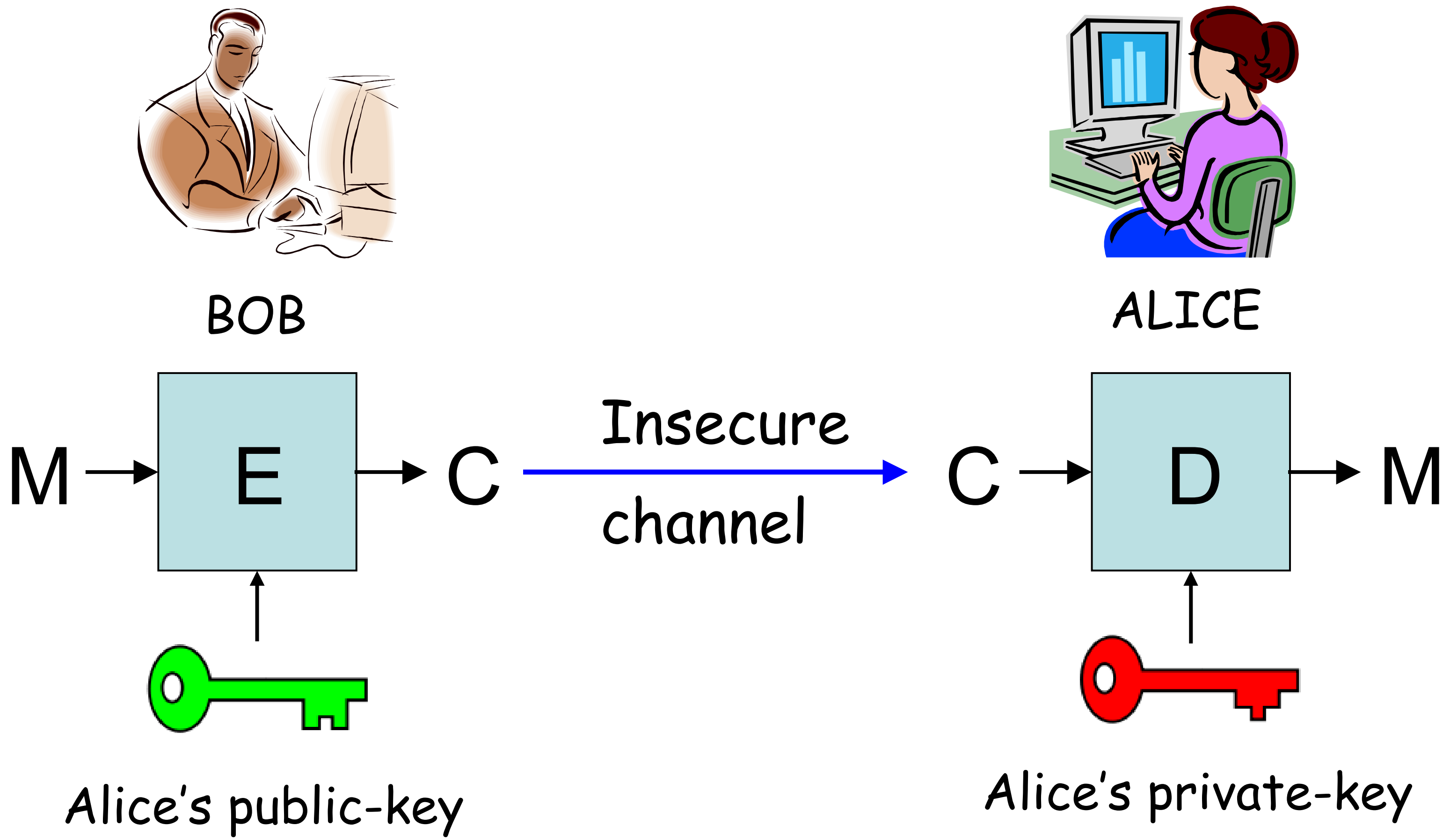
Key distribution issue

- Symmetric cryptography
  - How to initially distribute the key to establish a secure channel ?

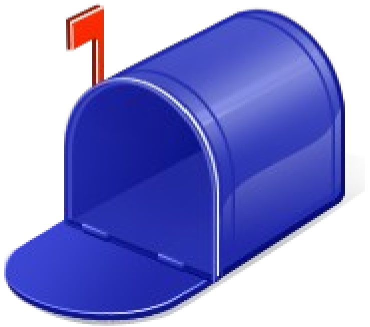


Asymmetric encryption

- Solves the key distribution issue



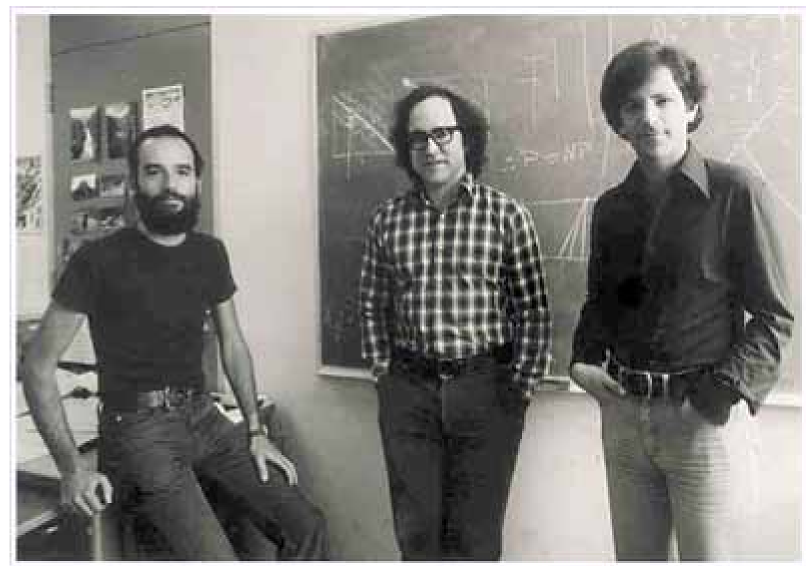
Analogy: the mailbox



- Bob wants to send a letter to Alice
  - Bob obtains Alice's adress
  - Bob puts his letter in Alice's mailbox
  - Alice opens her mailbox and read Bob's letter.
- Properties of the mailbox
  - Anybody can put a letter in the mailbox
  - Only Alice can open her mailbox

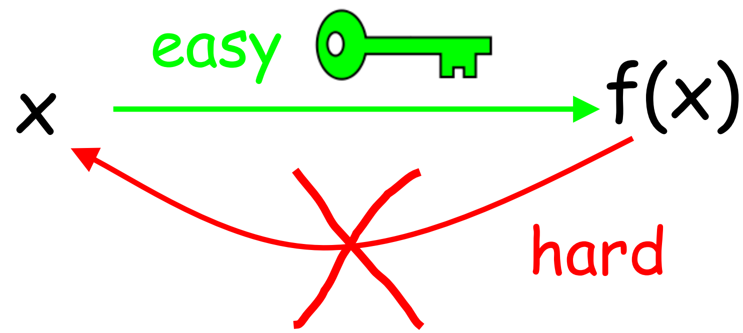
RSA (1977)

- Invented by Rivest, Shamir and Adleman
- First realization of asymmetric encryption.
- Implements a trapdoor one-way permutation.
- Still the most widely PK algorithm in use.

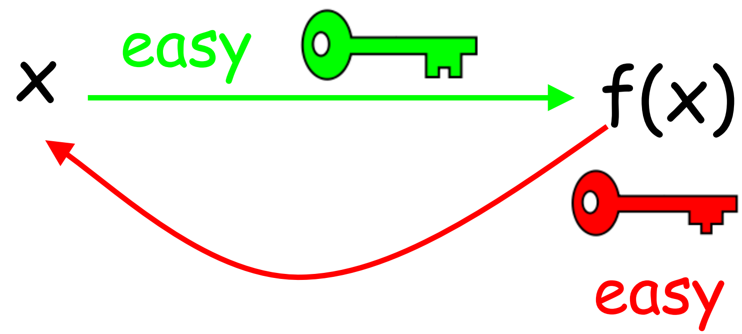


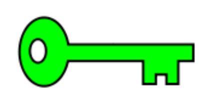
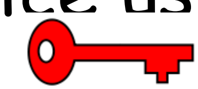
Trapdoor one-way permutation

- Trapdoor unknown:



- Trapdoor known:



- Asymmetric encryption:
  - Everybody can encrypt to Alice using 
  - Only Alice can decrypt using 

RSA

- Public key: $n=p.q$ and e
  - Primes p and q remain secret.
- Private key: d such that
  $e.d=1 \bmod (p-1)(q-1)$
- Encryption using public n,e:
  $c=m^e \bmod n$
- Decryption using private d:
  $m=c^d \bmod n$
- PKCS#1 v2.1

RSA

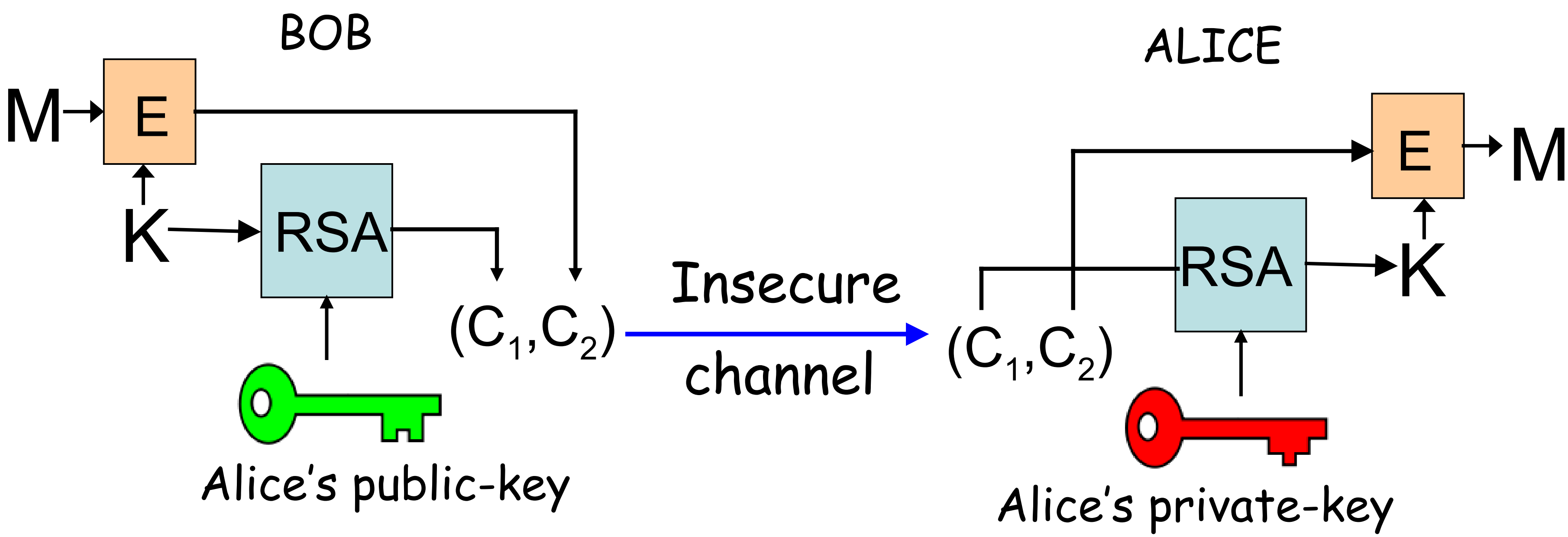
- Decryption works because $m=c^d=(m^e)^d=m^{e.d}=m$ because $e.d=1 \bmod f$
- Security is based on the hardness of factorization
  - Given $n=p.q$, no known efficient algorithm to recover p and q.
  - Factorization record: 663 bits (2005)
- Public modulus n must be large enough
  - At least 1024 bits. 2048 bits is better.

Implementation of RSA

- Required: computing with large integers
  - more than 1024 bits.
- In software
  - big integer library: GMP, NTL
- In hardware
  - Cryptoprocessor for smart-card
  - Hardware accelerator for PC.

Speed of RSA

- RSA much slower than AES and other secret key algorithms.
  - to encrypt long messages, encrypt a symmetric key K with RSA, and encrypt the long message with K.



Security of RSA

- Security of RSA is based on the hardness of factorization
  - Given $n=p.q$, no known efficient algorithm to recover p and q.
  - Factorization record: 768 bits (2009)
- Public modulus n must be large enough
  - At least 1024 bits. 2048 bits is better.
- Factoring is just one line of attack
  - not necessarily the most practical
  - more attacks to take into account…

Attacks against RSA

- Dictionary attack
  - If only two possible messages m_0 and m_1, then only two ciphertexts $c_0=m_0^e$ [n] and $c_1=m_1^e$ [n].
  - Encryption must be probabilistic (or non-static).
- Coppersmith's attack (1996)
  - Applies for RSA with small e, when some part of the message is known

Attacks against RSA

- Chosen-ciphertext attack:

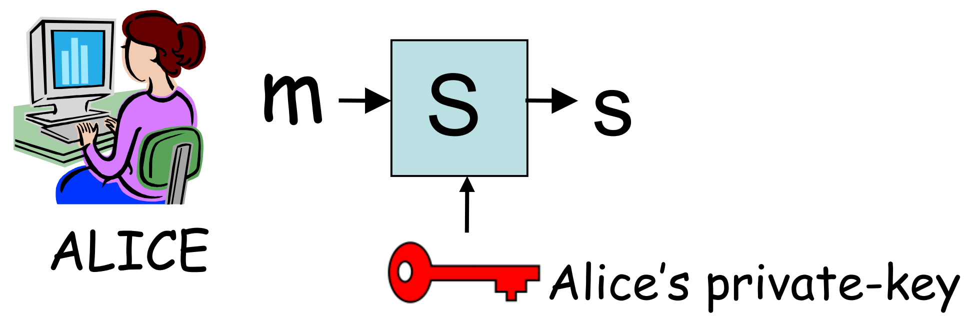
  Given ciphertext c to be decrypted
  - Generate a random r
  - Ask for the decryption of the random looking ciphertext $c' = c * (r^e) [n]$
  - One gets $m' = c'^d = c^d * (r^e)^d = c^d * r = m * r \ [n]$
  - This enables to compute $m = m'/r \ [n]$

Attacks against RSA

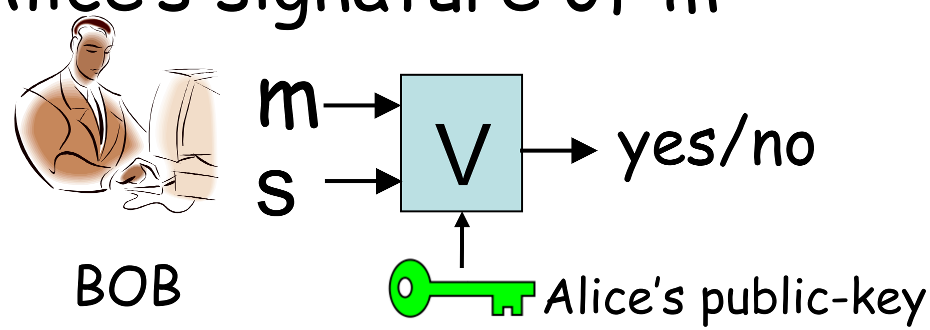
- One cannot use plain RSA encryption
  - one must add some randomness
  - one must apply some preformatting to the message
- Example: PKCS#1 v1.5
  - Encryption: $F(m)=0002 \mid r \mid 00 \mid m$, then $c=F(m)^e\,[n]$
  - Decryption: recover $F(m)$, check redundancy.
- Bleichenbacher's attack against PKCS#1 v1.5
  - Appeared in 1998. Could be use against web-servers using SSL protocol.

Digital signature

- A bit string that depends on the message m and the user's public-key
  - Only Alice can sign a message using her private-key



  - Anybody can verify Alice's signature of m given her public-key



Digital signature

- A digital signature provides:
  - Authenticity: only Alice can produce a signature of a message valid under her public-key.
  - Integrity: the signed message cannot be modified.
  - Non-repudiation: Alice cannot later claim that she did not sign the message

Signing with RSA

- Public key: $n=p.q$ and e
- Private key: d such that

  $e.d=1 \bmod (p-1)(q-1)$
- Signing using private d:

  $s=m^{d} \bmod n$
- Verifying using public n,e:

  check that $m=s^{e} \bmod n$
- ISO 9796-2, PKCS#1 v2.1

Other signature schemes

- Digital Signature Algorithm (DSA) (1993)
  - Digital Signature Standard (DSS) proposed by NIST, specified in FIPS 186.
  - Security based on the hardness of discrete log.
  - ECDSA: a variant of DSA for elliptic-curves
- Rabin signature scheme
  - Similar to RSA but with e=2
- El-Gamal signature scheme (1984)
  - Based on the discrete-log problem

Diffie-Hellman key exchange (1976)

- Public parameters: g and p

BOB

ALICE

$B = g^b$ $\xrightarrow{B}$ $A = g^a$

$\xleftarrow{A}$

$K_B = A^b = (g^a)^b = g^{a.b}$

$K_A = B^a = (g^b)^a = g^{b.a}$

$K_B = K_A$

Security of Diffie-Hellman

- Based on the hardness of the discrete-log problem:
  - Given $A=g^a \bmod p$, find a
  - No efficient algorithm for large p.
- No authentication
  - Vulnerable to the man in the middle attack
- Authenticated key exchange
  - Using a PKI. Alice and Bob can sign A and B
  - Password-authenticated key-exchange IEEE P1363.2

Lessons from the past

- Cryptography is a permanent race between construction and attacks
  - but somehow this has changed with modern cryptography and security proofs.
- Security should rely on the secrecy of the key and not of the algorithm
  - Open algorithms enables open scrutiny.

Modern cryptography

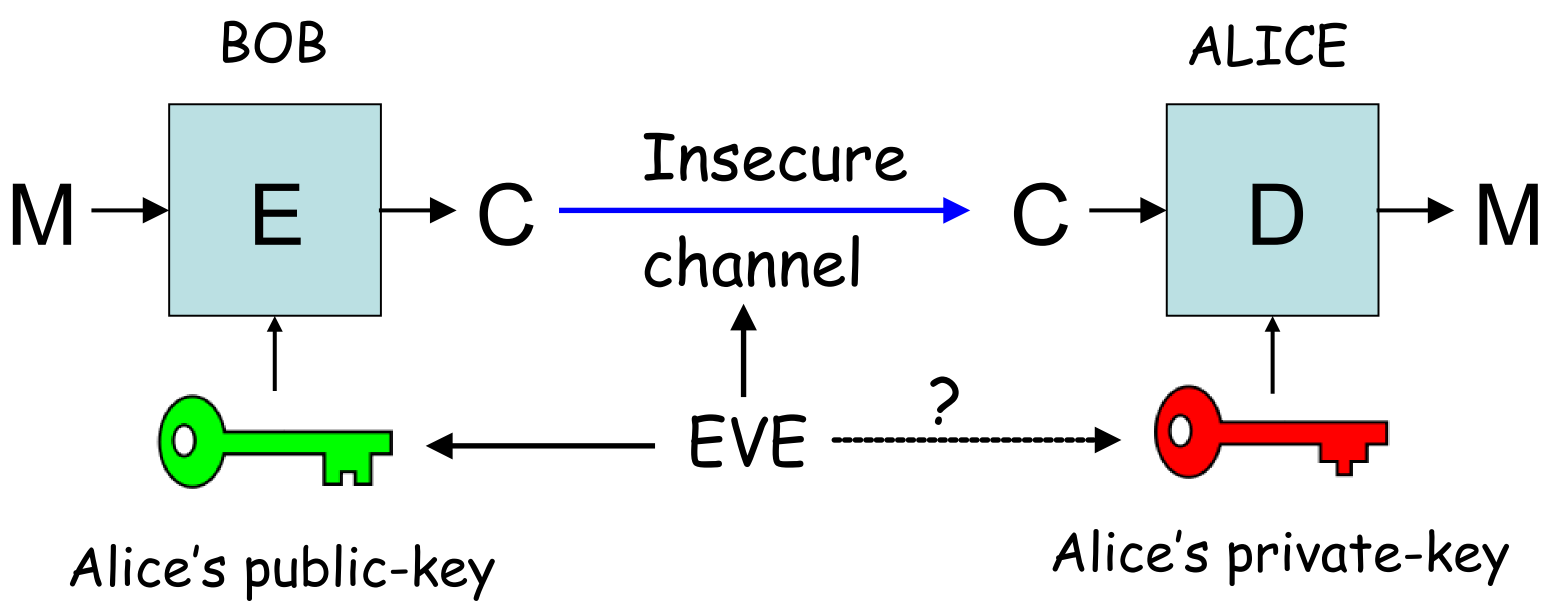
- New functionalities
  - Identity-based encryption, voting, electronic money, auction…
- Formalization of security notions
  - What is a secure encryption scheme ? a secure signature scheme ?
- Construction of schemes or protocols that provably achieve these security notions
  - Based on some hardness assumption (e.g., factoring is hard).
- Modern cryptography is about security proofs.
  - A scheme without security proof is useless.

Security models

- To be rigorous when speaking about security, one must specify
  - the attacker's *goal*:
    does he need to recover the key or only decrypt a particular ciphertext or less ?
  - the attacker's power:
    does he get only the user's public-key, or more ?

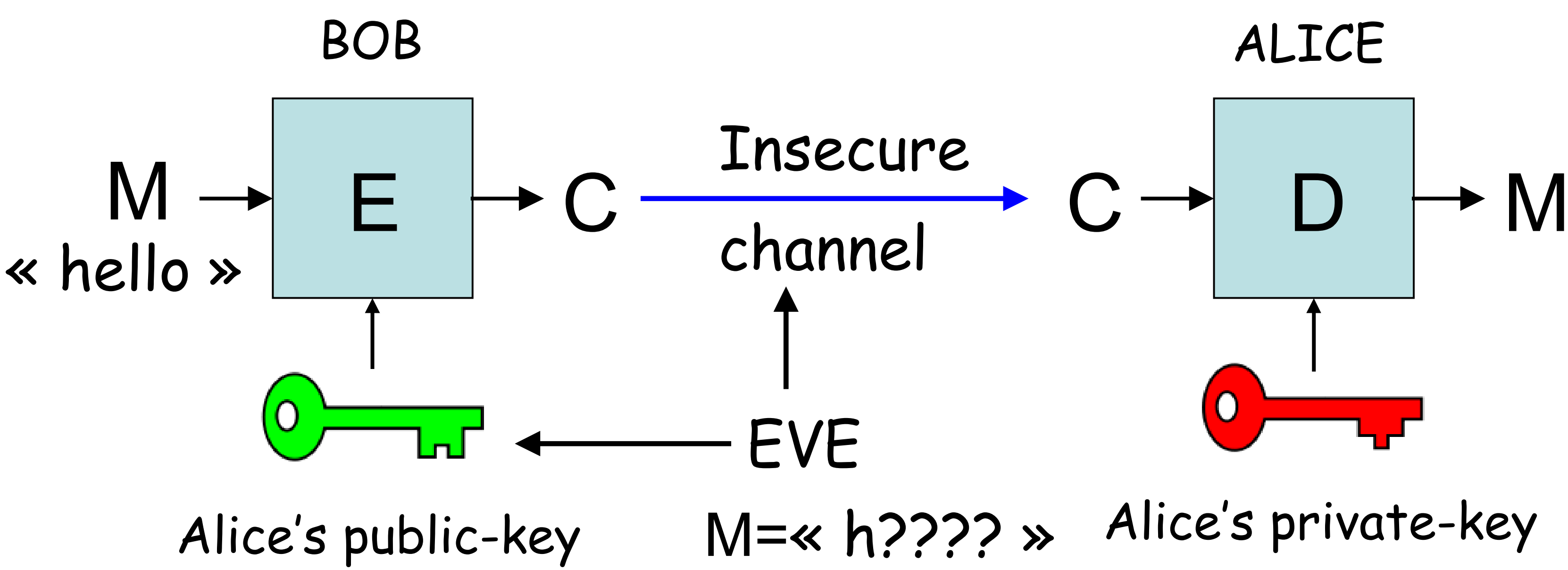
Attacker's goal

- One may think that the adversary's goal is always to recover the private key.
  - complete break
  - may be too ambitious in practice



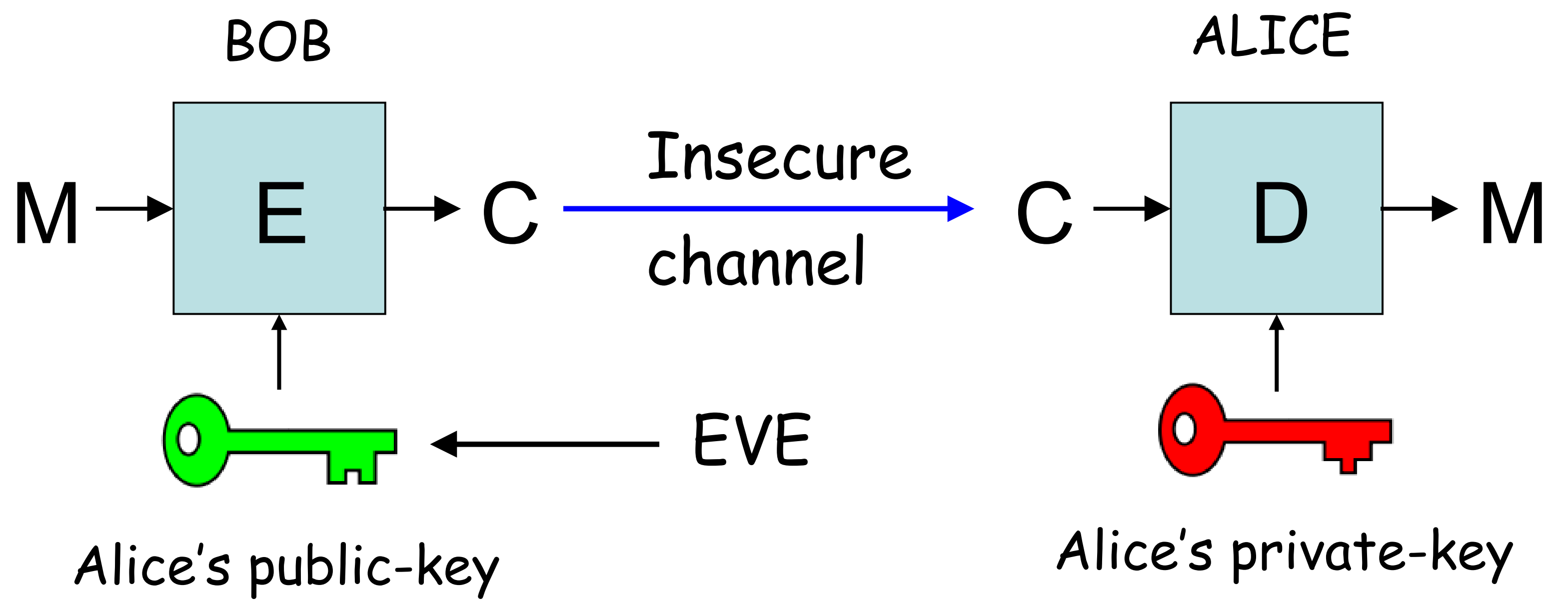
Attacker's goal

- More modest goal: being able to decrypt one ciphertext.
  - or recover some information about a plaintext (for example, the first character)



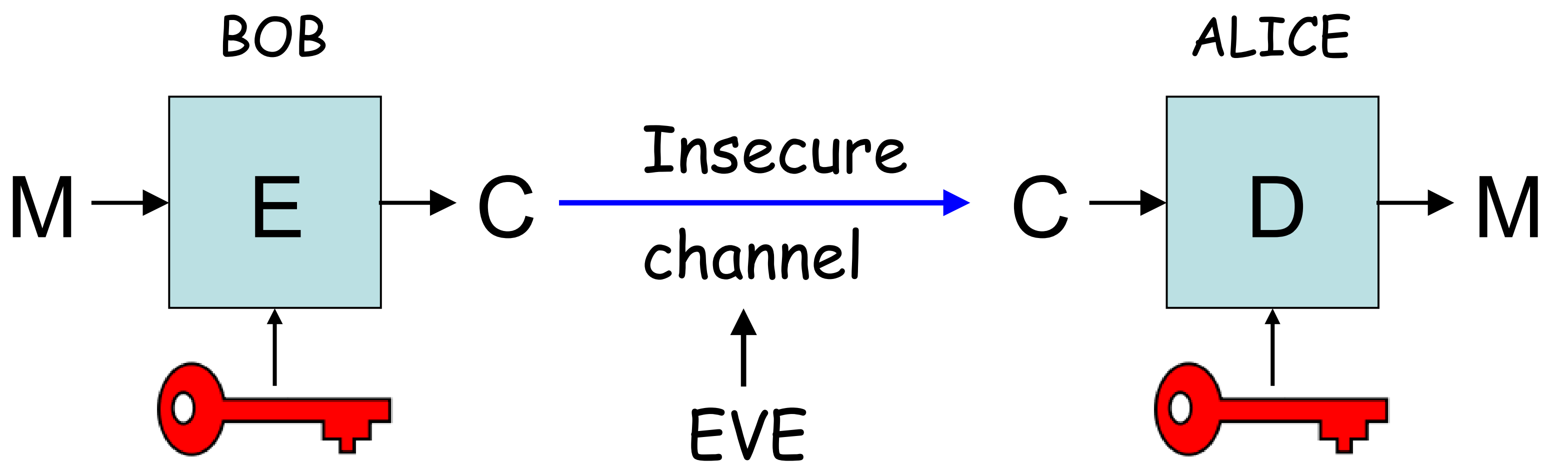
Attack models

- Specify the power of the attacker
- Public-key only
  - the attacker gets only the public-key
  - Weakest adversary



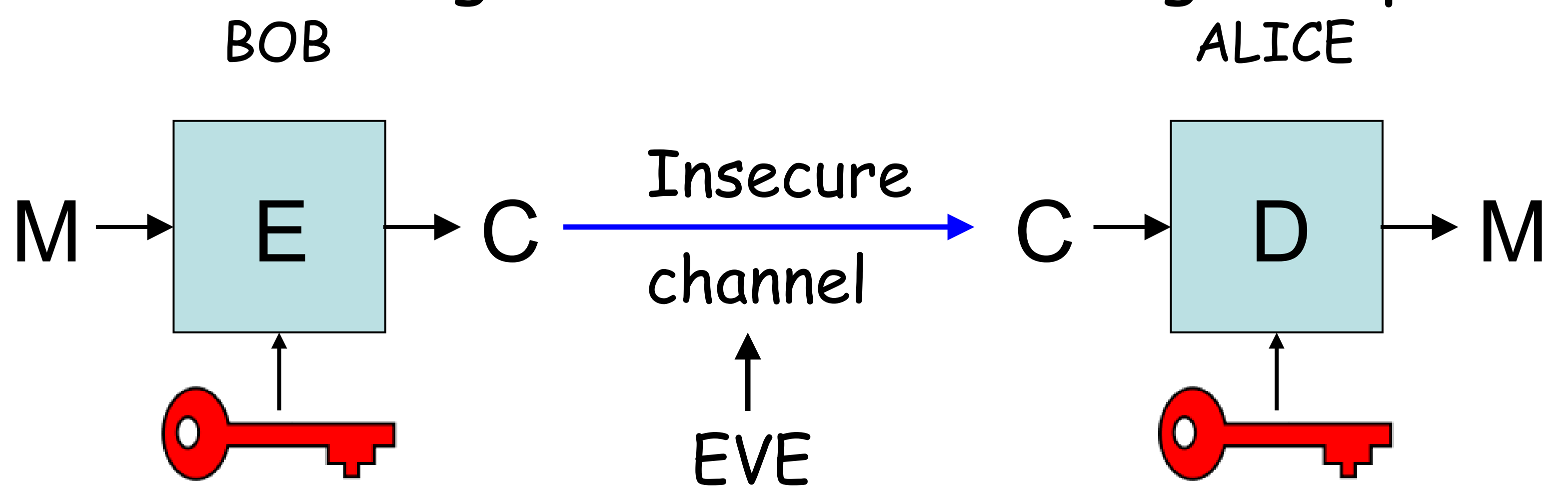
Attack models

- Ciphertext-only attack
  - the attacker gets only a set of ciphertexts
  - primitive ciphers (Ceasar's cipher, mono-alphabetic substitution cipher) were vulnerable.



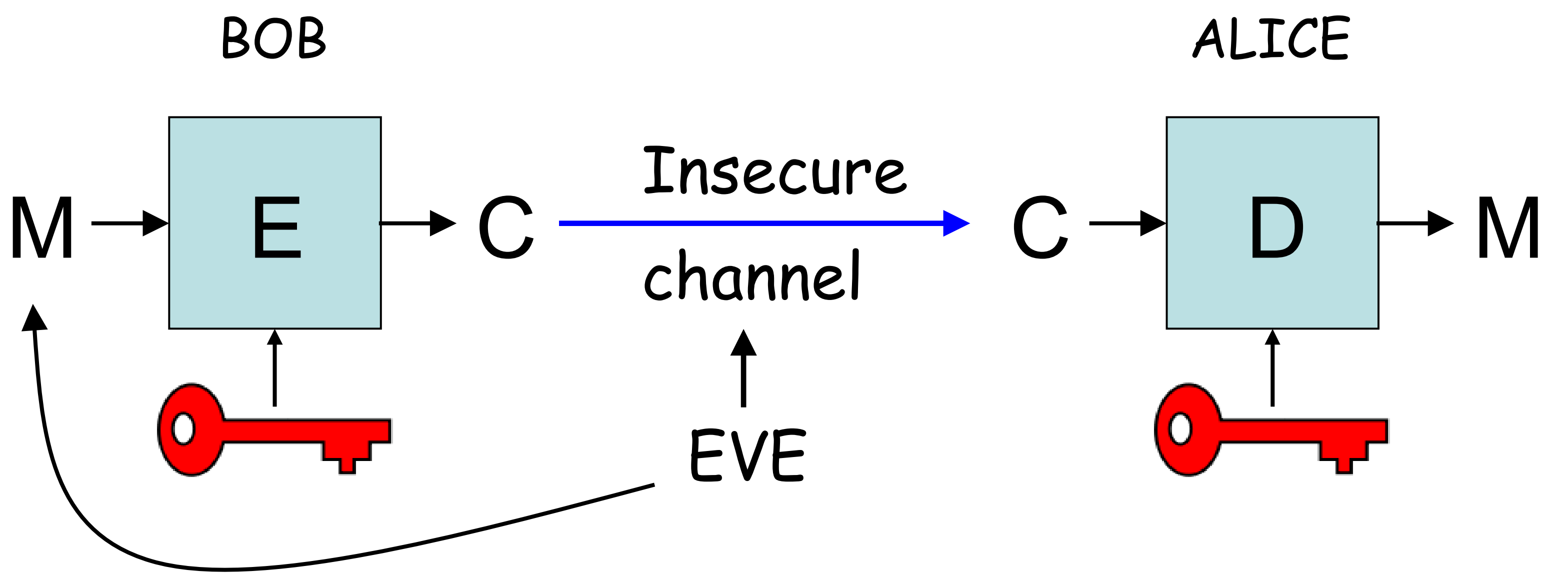
Attack models

- Known-plaintext attack
  - Attack has access to plaintext/ciphertext pairs.
  - In practice, attacker may have some hint on some plaintexts.
  - Used during WW2 to break Enigma cipher.



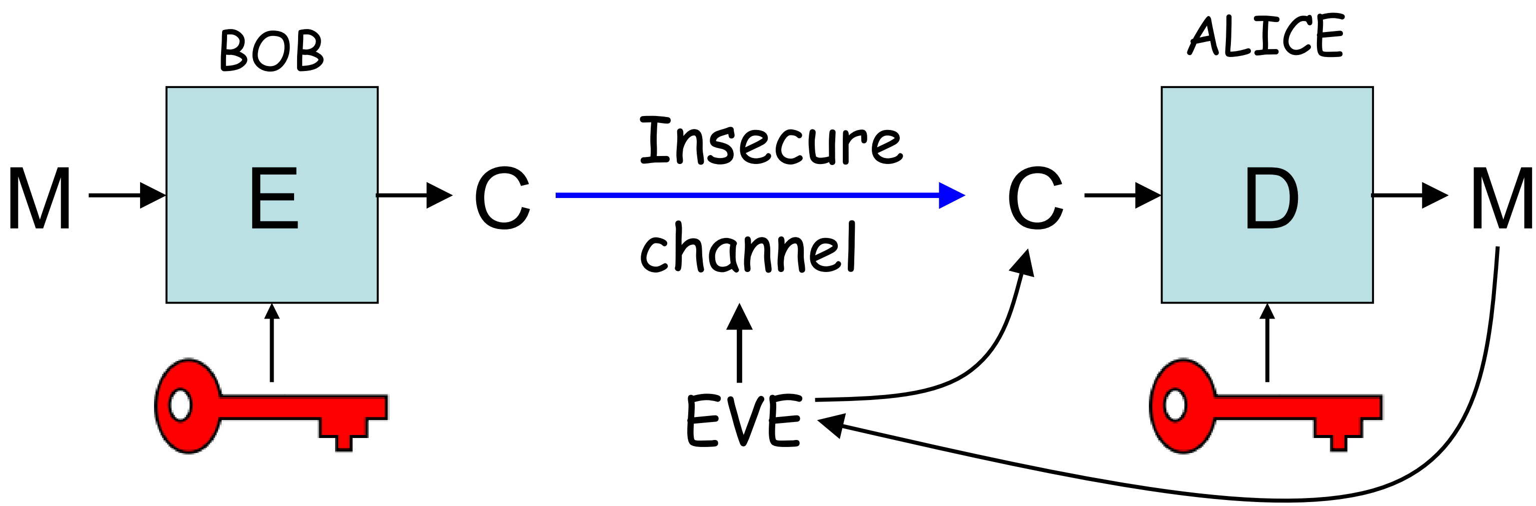
Attack models

- Chosen plaintext attack
  - Attacker can obtain encryption of plaintexts of his choice.
  - For PK encryption, equivalent to PK only attack.



Chosen-ciphertext attack

- Most powerful attack
- The attacker can obtain decryption of messages of his choice
- May be realistic in practice
  - attacker gets access to a decryption machine
  - encryption algorithm used in a more complex protocol in which users can obtain decryption of chosen ciphertexts.



Attack scenario

- One must specify
  - the attacker's *goal* (total break, partial decryption…)
  - The attack model (chosen plaintext, chosen ciphertext…)
- Strongest security model: combines
  - weakest *goal*: obtaining only one bit of information about a plaintext
  - strongest adversary: chosen ciphertext attack

Strongest security notion

- Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2)
  - Formalized in 1991 by Rackoff et Simon
  - A ciphertext should give no information about the corresponding plaintext, even under an adaptive chosen-ciphertext attack.
  - Has become standard security notion for encryption.

IND-CCA2 schemes

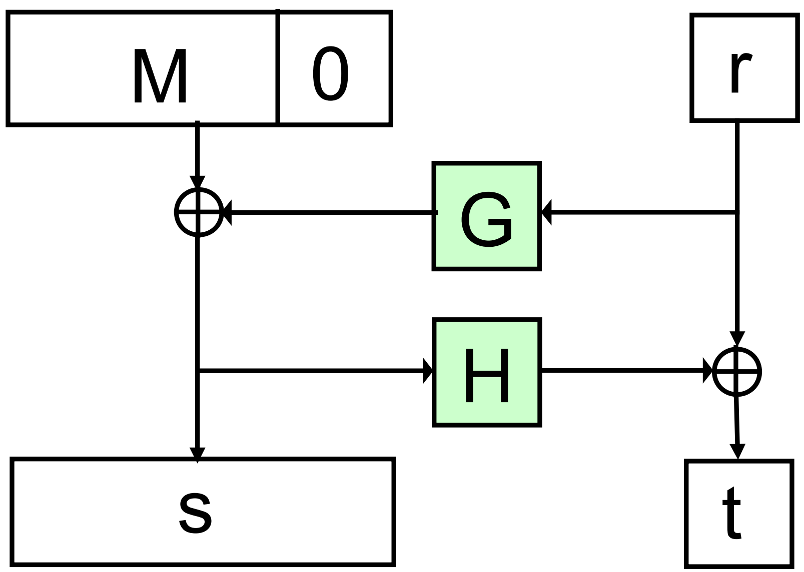
- OAEP
  - Designed by Bellare and Rogaway in 1994.
  - Appears in PKCS#1 v2.1 standard.
- Cramer-Shoup (1998)
  - Based on discrete-log.
  - Proven secure without the random oracle model.

Reminder: textbook RSA encryption

- Public key: $n=p.q$ and e
  - Primes p and q remain secret.
- Private key: d such that
  $e.d=1 \bmod (p-1)(q-1)$
- Encryption using public n,e:
  $c=m^e \bmod n$
- Decryption using private d:
  $m=c^d \bmod n$

RSA-OAEP

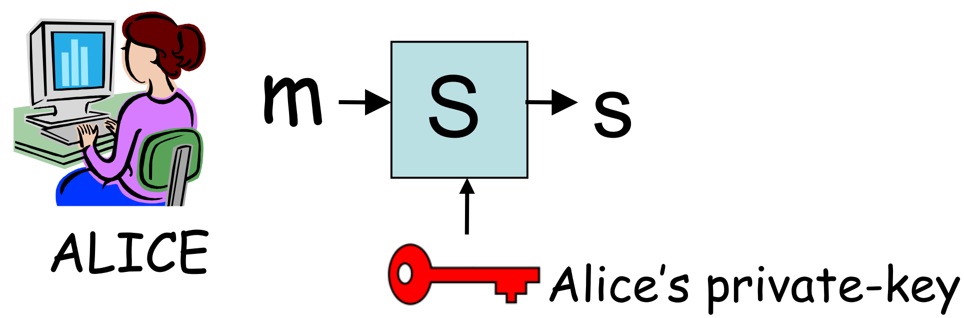
- Ciphertext is $c=(s|t)^e\ [n]$



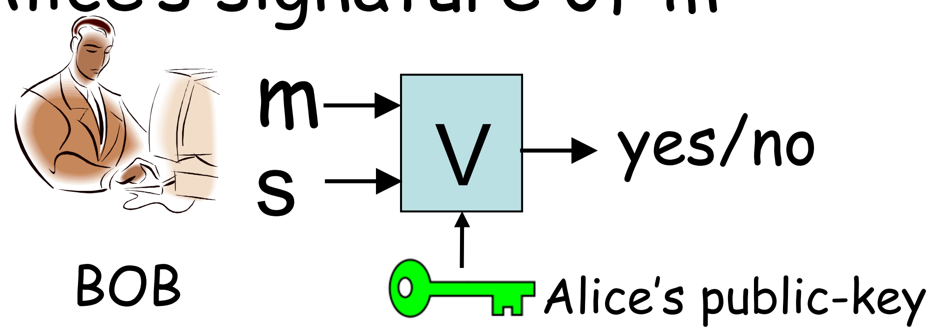
- IND-CCA2 secure in the RO model

Digital signature

- A bit string that depends on the message m and the user's public-key
  - Only Alice can sign a message using her private-key



  - Anybody can verify Alice's signature of m given her public-key



Reminder : textbook RSA signatures

- Public key: $n=p.q$ and e
- Private key: d such that

  $e.d=1 \bmod (p-1)(q-1)$
- Signing using private d:

  $s=m^d \bmod n$
- Verifying using public n,e:

  check that $m=s^e \bmod n$

Attacks against RSA signatures

- Given $s_1 = m_1^d \bmod n$ and $s_2 = m_2^d \bmod n$
  - one can compute the signature of $m_1 * m_2$ without knowing d

  $s = s_1 * s_2 = (m_1^d) * (m_2^d) \bmod n = (m_1 * m_2)^d \bmod n$
- One cannot use plain RSA signature
  - One must apply some pre-formatting to the message to cancel the mathematical structure.

RSA signature

- To prevent these attacks, one uses a hash function
  - PKCS#1 v1.5 :
    F(m)=0001 FF … FF00 ¦ c ¦ H(m)
    $s=F(m)^d \bmod n$
  - ISO 9796-2:
    F(m)=6A ¦ m[1] ¦ H(m) ¦ BC
    $s=F(m)^d \bmod n$

Attack scenario for signature schemes

- We must specify
  - the adversary's goal
  - the adversary's power
- Adversary's goal
  - Controlled forgery: the adversary produces the signature of a message of his choice
  - Existential forgery: the adversary produces the signature of a (possibly meaningless) message

Adversary's power

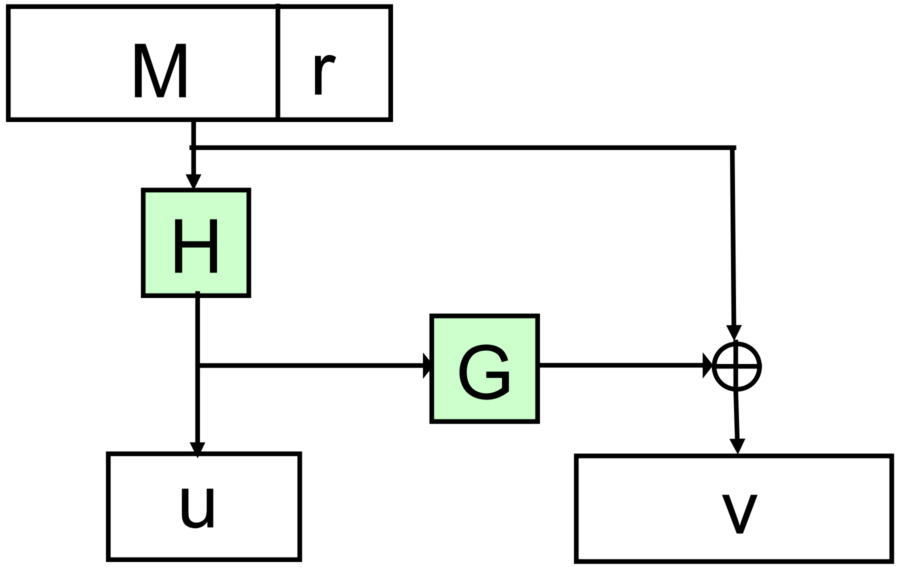
- No-message attack
  - The adversary gets only the public-key
- Known message attack
  - The adversary obtains a set of pairs message/signatures
- Chosen message attack
  - The adversary can obtain the signature of any message of his choice, adaptively.

Strongest security notion

- Combines weakest goal with strongest adversary
- Existential unforgeability under an adaptive chosen message attack
  - Defined by Goldwasser, Micali and Rivest in 1988
  - It must be infeasible for an attacker to forge the signature of a message, even if he can obtain signature of messages of his choice.

Example of secure signature schemes

- PSS
  - Designed by Bellare and Rogaway in 1996
  - IEEE P1363a standard and PKCS#1 v2.1
  - 2 variants: PSS and PSS-R that provides message recovery.



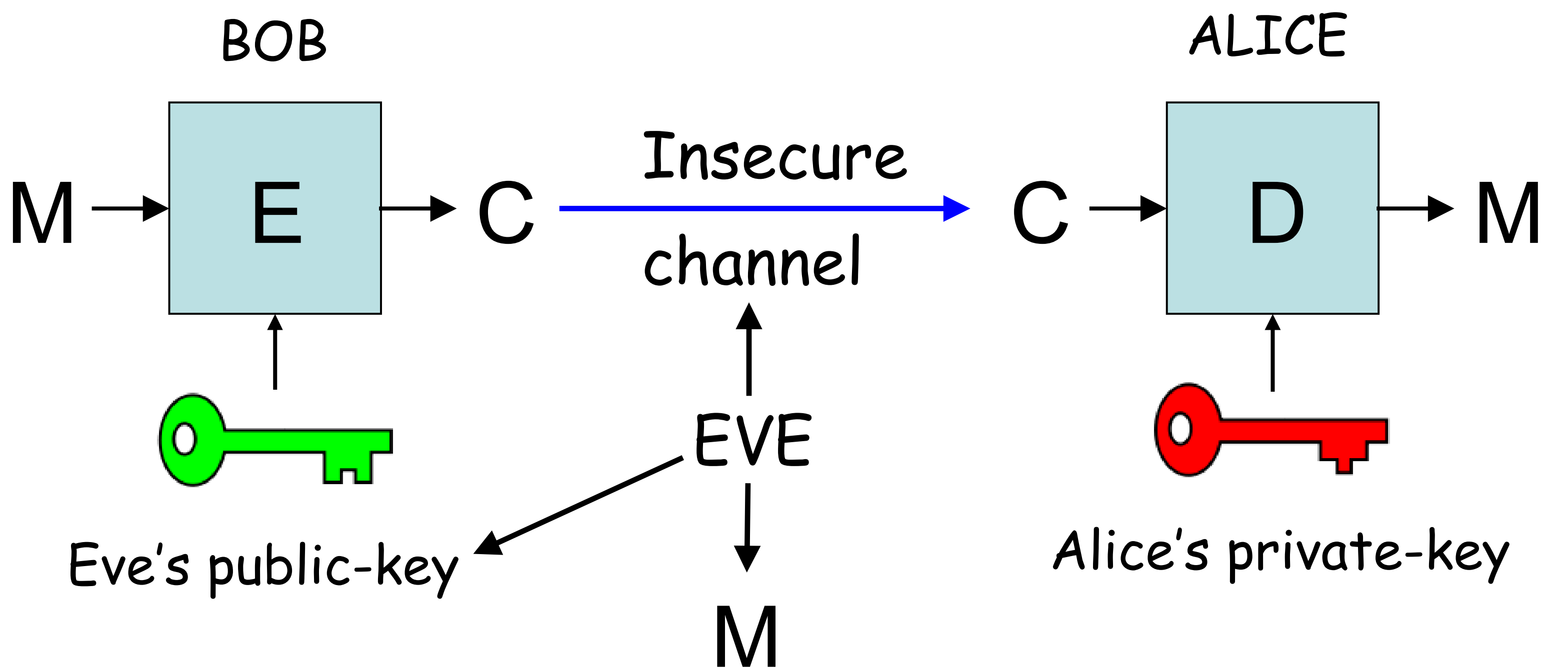
$s=(u \mid v)^d \bmod n$

Conclusion

- What is cryptography ?
  - Cryptography's aim it to construct protocols that achieve some goal despite the presence of an adversary
- Scientific approach:
  - To be rigorous, one must define what it means to be secure
  - Then one tries to construct schemes that satisfies the definition, in a provable way.

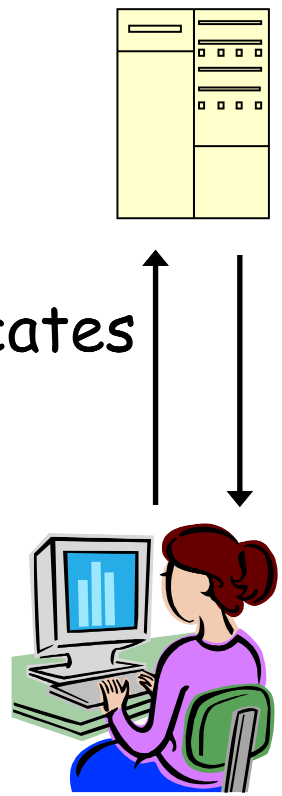
PK Authentication

- 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

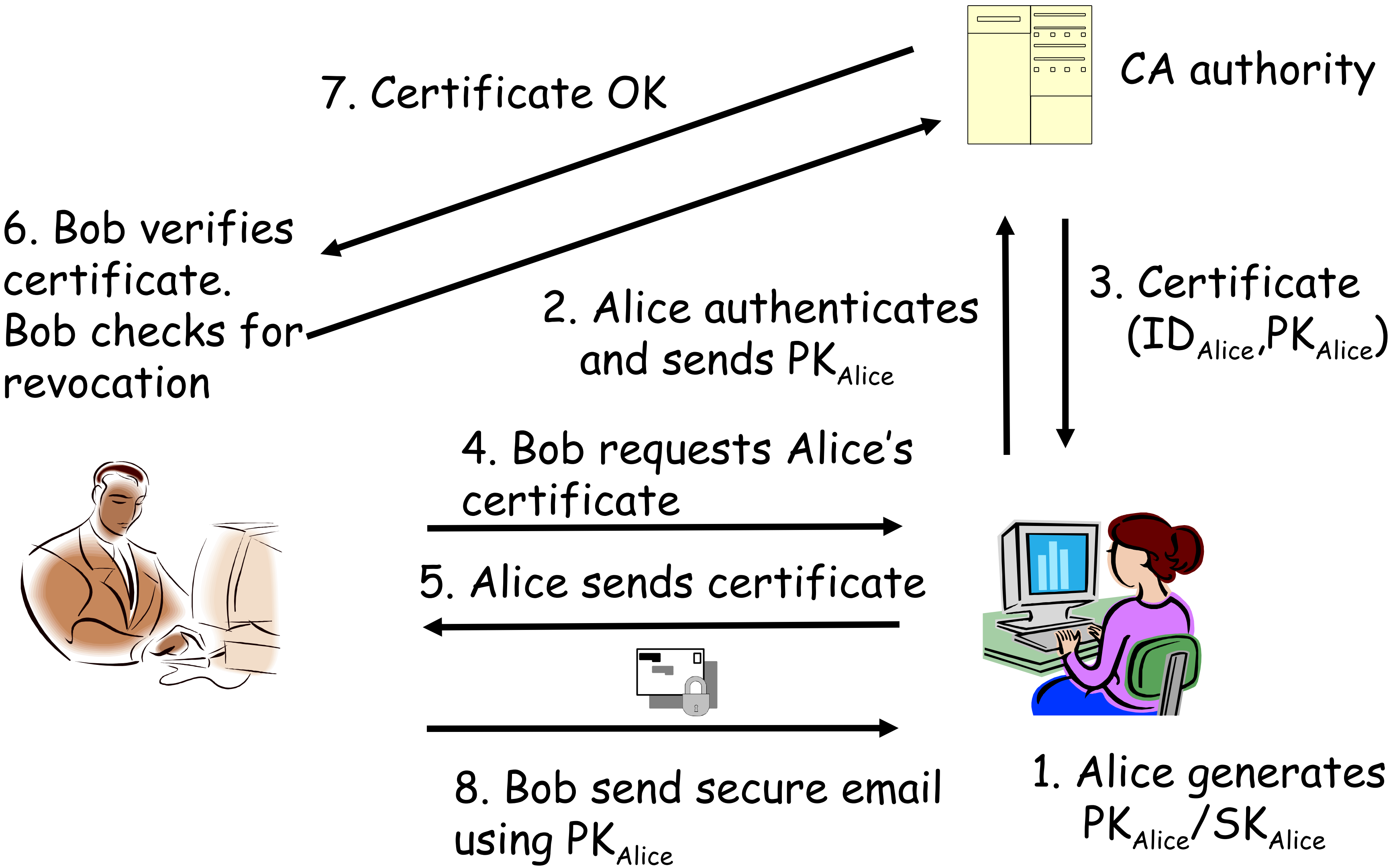
Public-key certificate

- 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

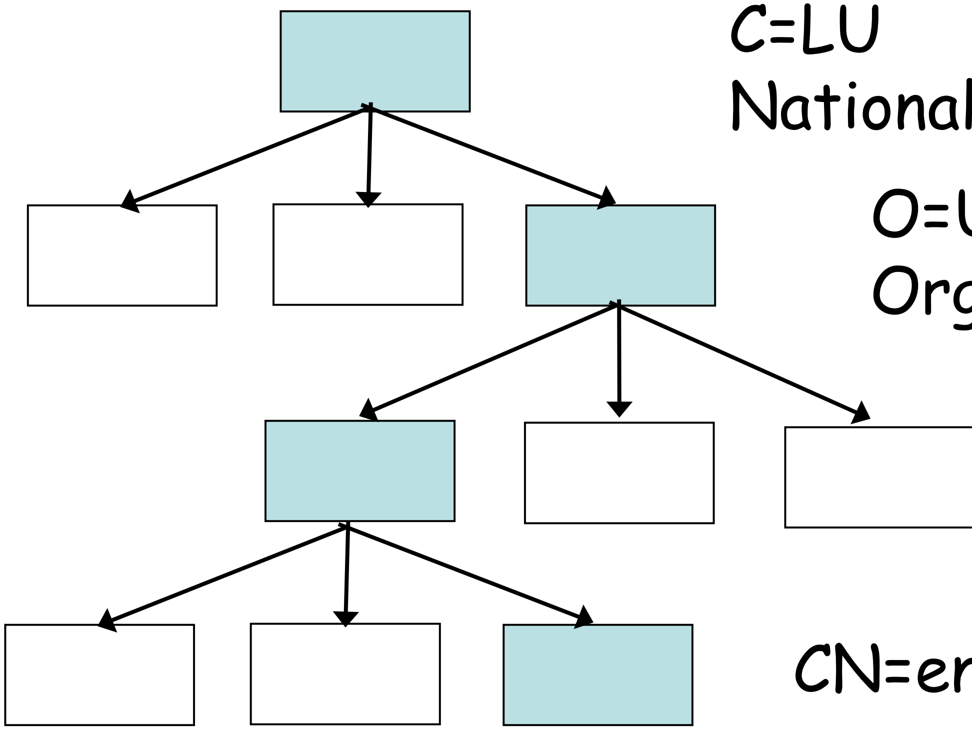
PKI encryption



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

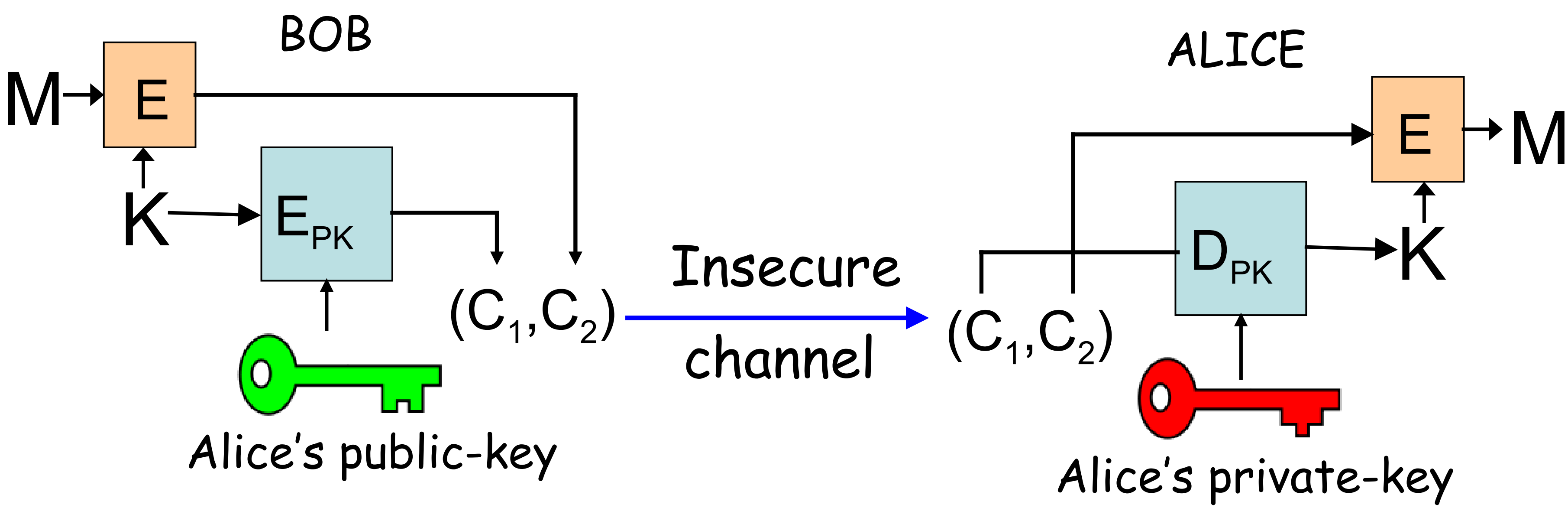
- X509
  - Most common certificate standard
  - Specifies certificate format and certificate validation path.
  - Assumes a hierarchy of CA
  - Root certificate is implicitly trusted
  - Specifies certificate revocation list (CRL) implementation

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.

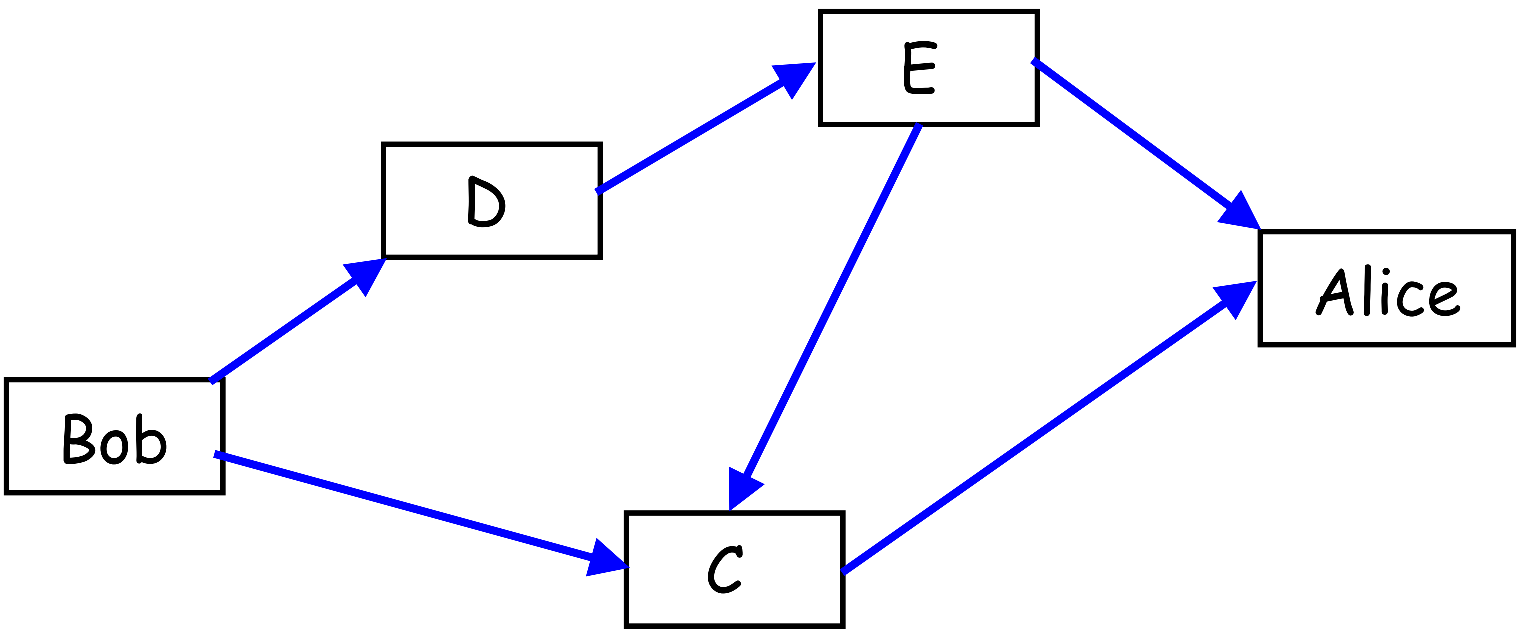


PGP

- 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)

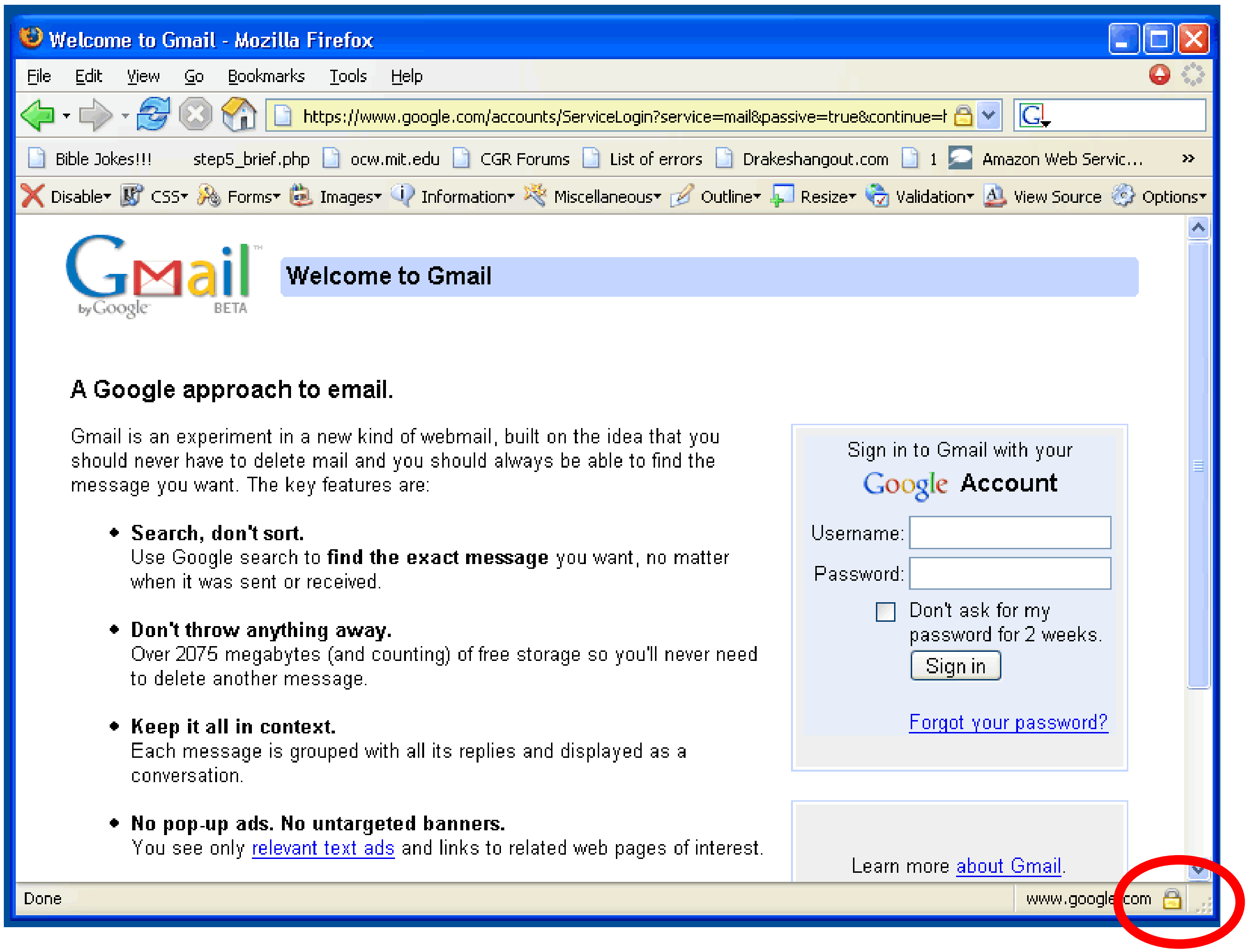
- Server authenticated by proof of possession of private key
  - Ability to decrypt client data.
  - Both sides share the same « master secret »
- Client/server finish
  - Authenticate all previously exchanged data with MACs

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

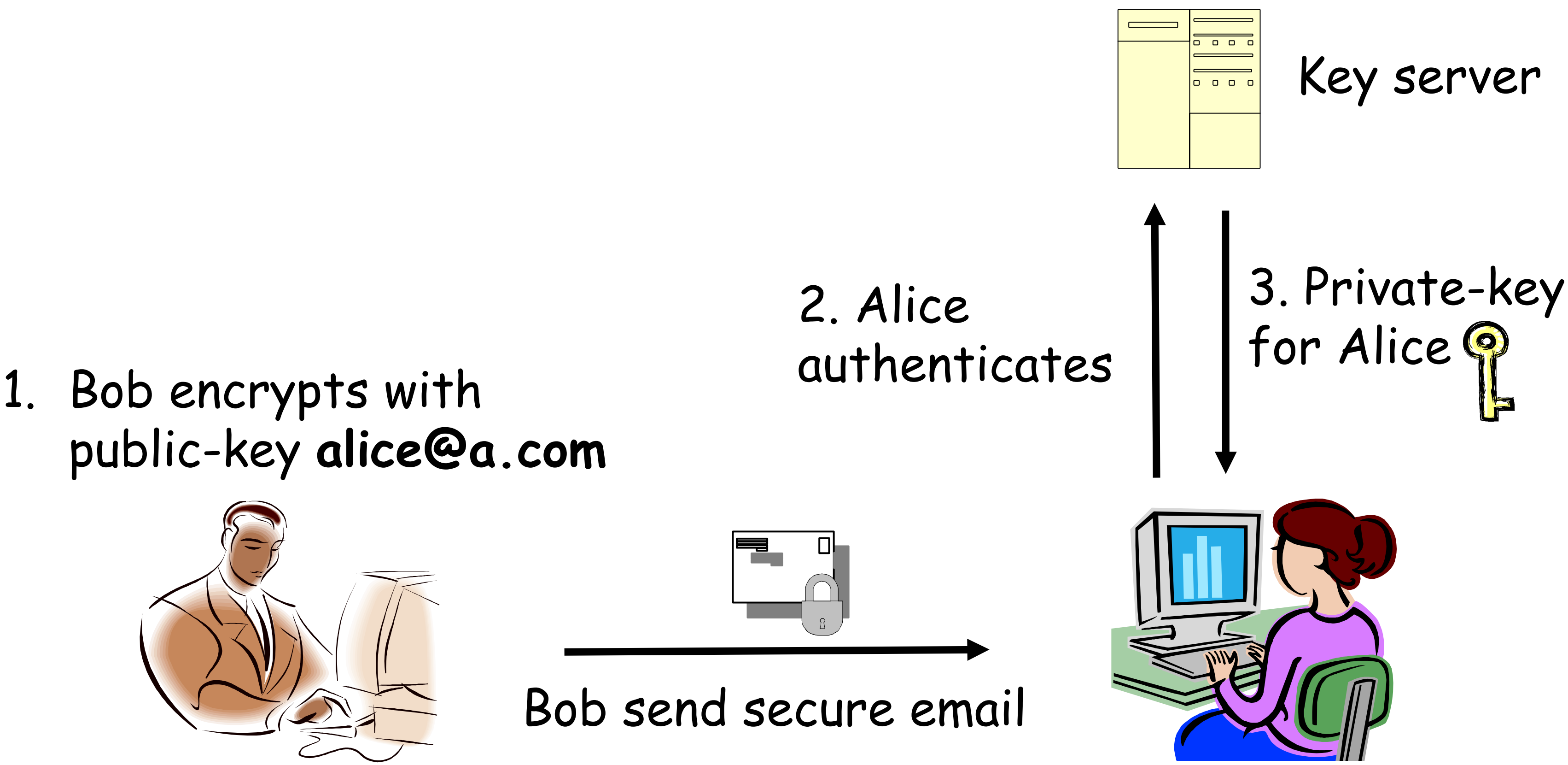
- https only protects the credit card number during transit between the user's computer and the server
  - Does not protect against an attack on the server
- Attack on the server usually easier than interception in transit.
  - Credit card number often saved in a database in merchant site
  - Attacks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

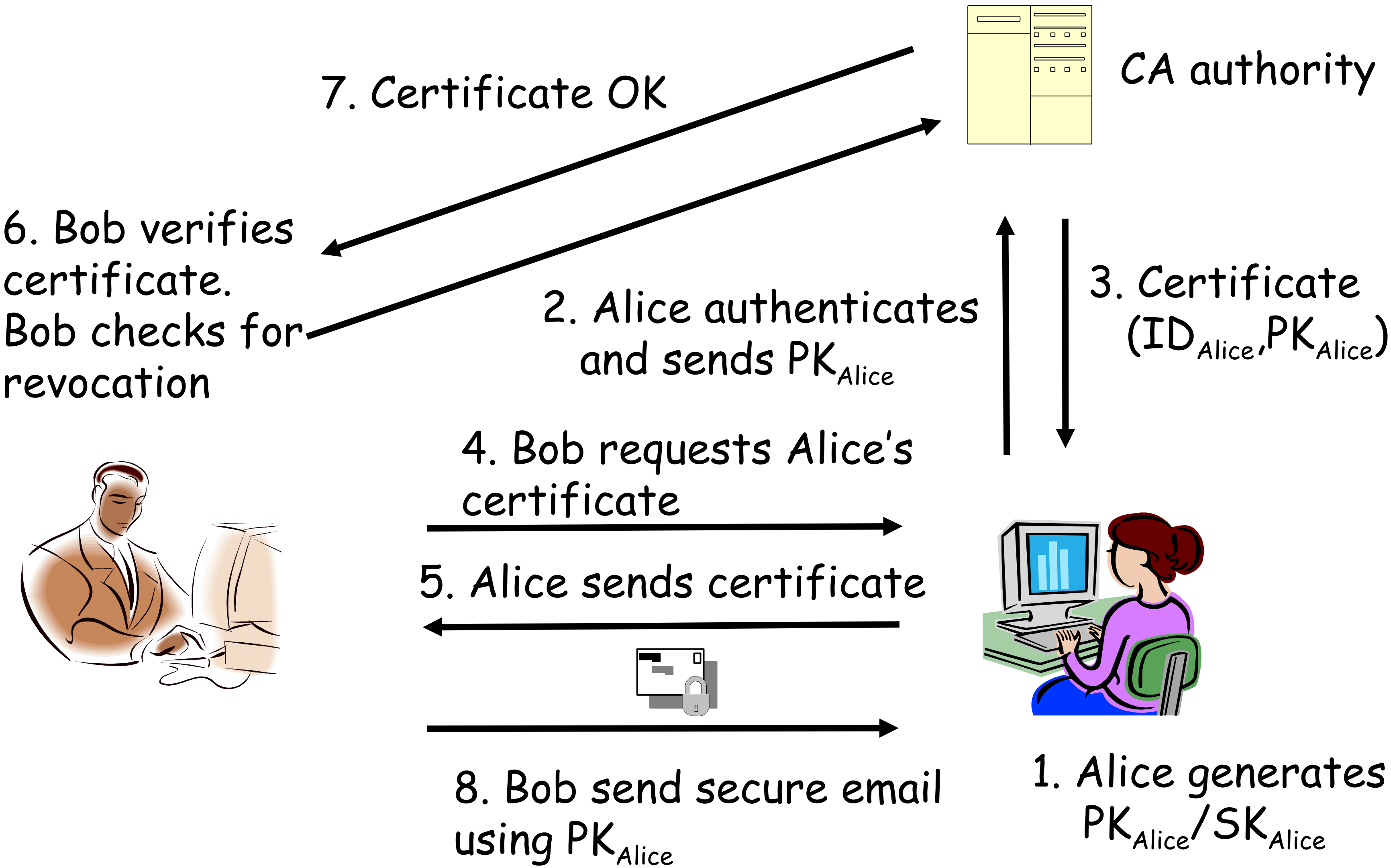
- Bob sends an email to Alice using his identity as the public-key



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



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.