Introduction to Cryptography

Jean-Sébastien Coron

September 2014

Information Security Basics

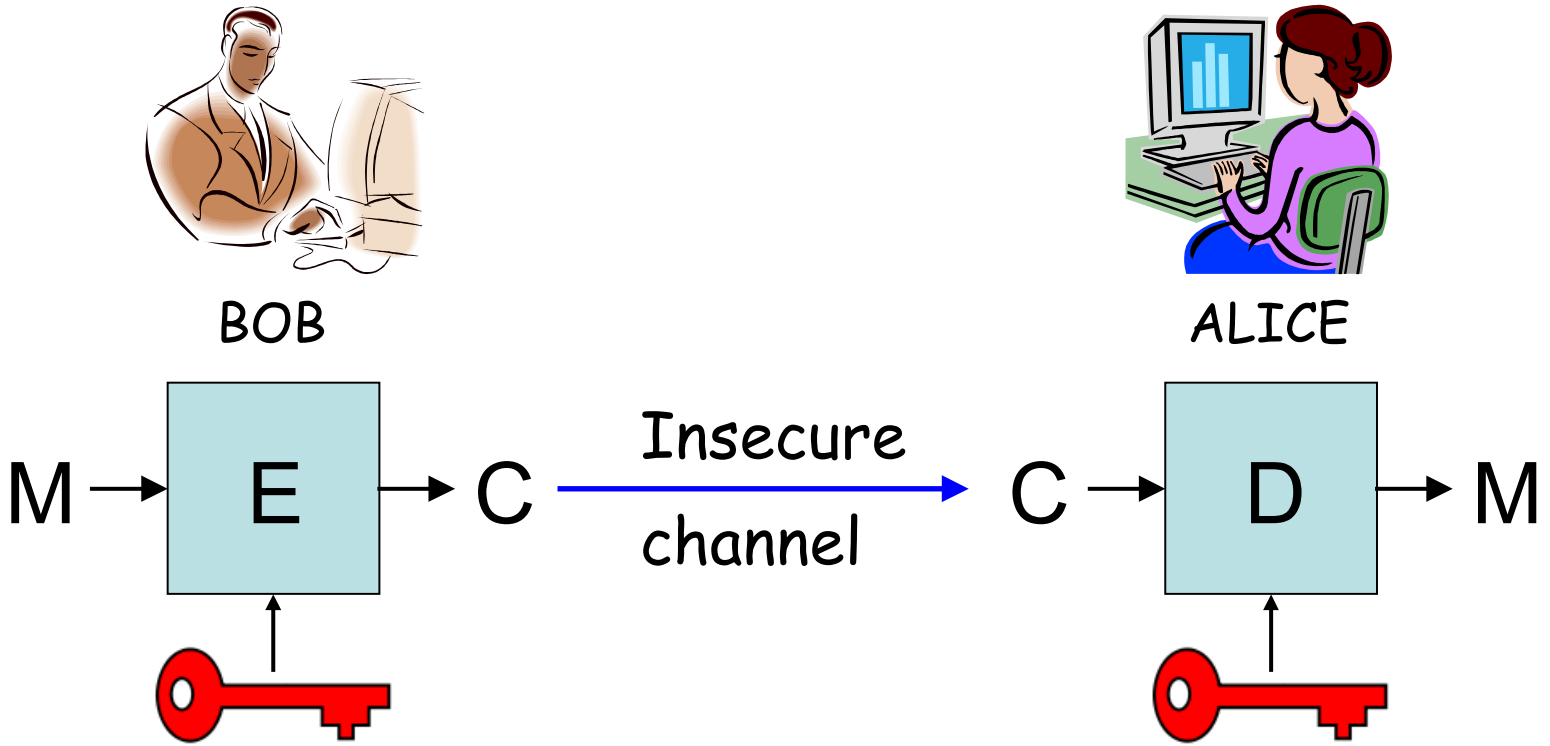
- 4 parts
- Public-key cryptography:
  - Jean-Sébastien Coron
- Symmetric-key cryptography:
  - Alex Biryukov
- Security protocols:
  - Sjouke Mauw and Hugo Jonker
- Advanced security protocols:
  - Peter Ryan

Outline

- Early symmetric-key encryption schemes
- Public-key cryptography
  - Public-key encryption
  - The RSA cryptosystem
  - Digital signatures
  - Diffie-Hellman Key-exchange

Traditional goal: encryption

- Symmetric cryptography



One-time pad (1917)

- Plaintext is xored with the key to produce the ciphertext

```
  011001011001
  111010010010
------------
  100011001011
```

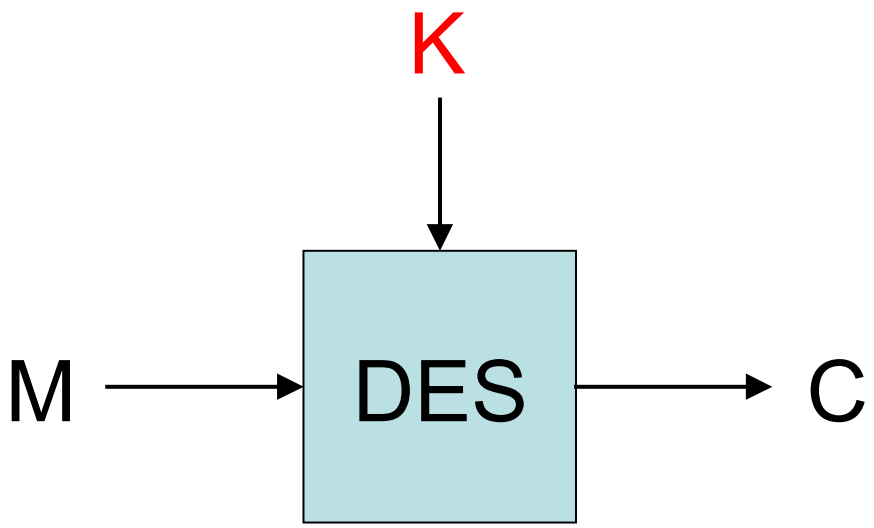
- Proved unbreakable by Shannon (1949) if key is random and as long as the plaintext.
- Issue: key as long as the plaintext.
- Used for the hotline between Washington and Moscow during the cold war.

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- Superseded by the AES, but remains in widespread use.

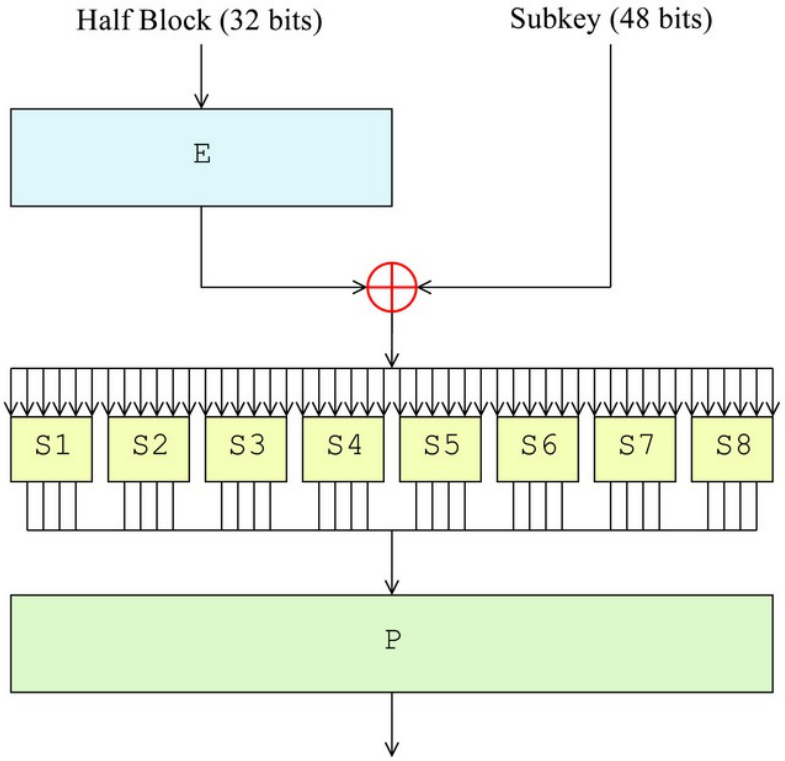
DES block-cipher

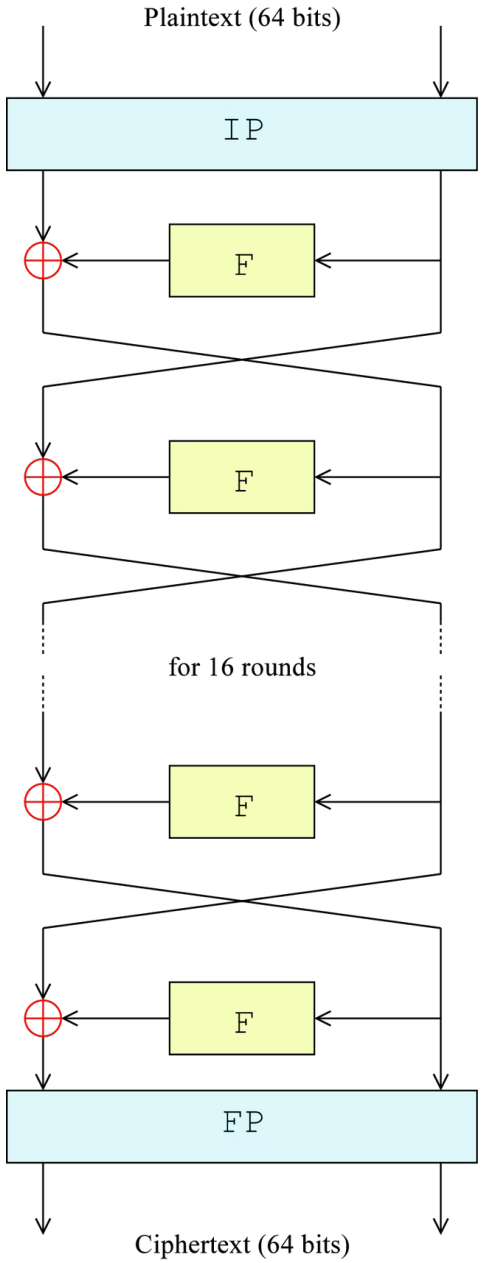
- Input length: 64 bits.
- Output length: 64 bits.
- Key length: 56 bits.



DES

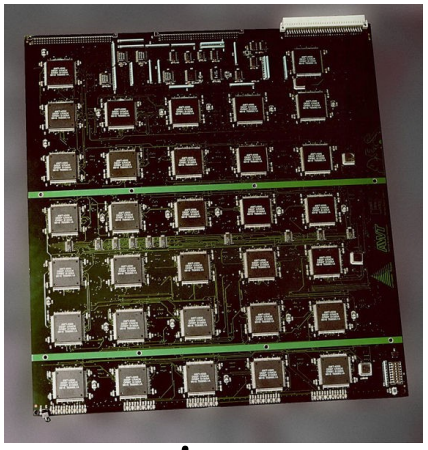
- Feistel Cipher
- F function:





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).

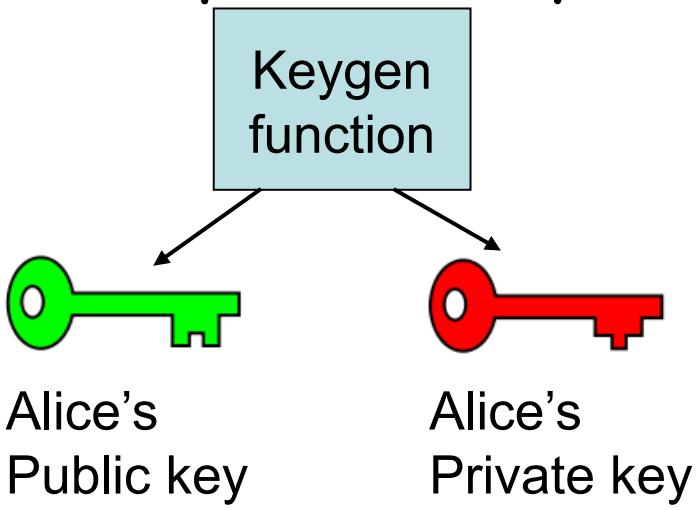


- Other attacks
  - Differential cryptanalysis (Biham and Shamir). 2^47 chosen plaintexts
  - Linear cryptanalysis (Matsui, 1993). 2^43 known plaintexts.
- New standard: AES

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.

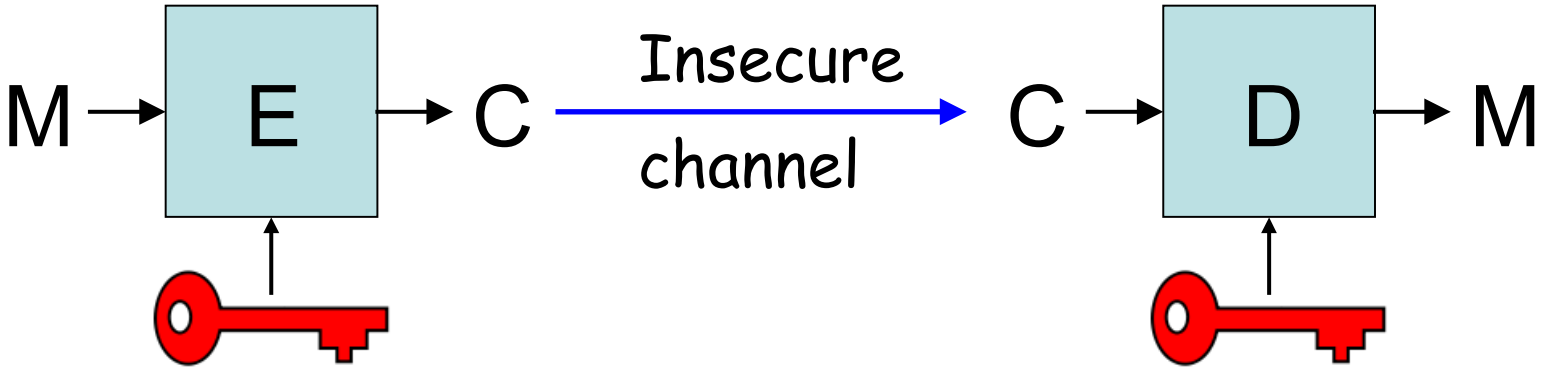


Key distribution issue

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

BOB

ALICE

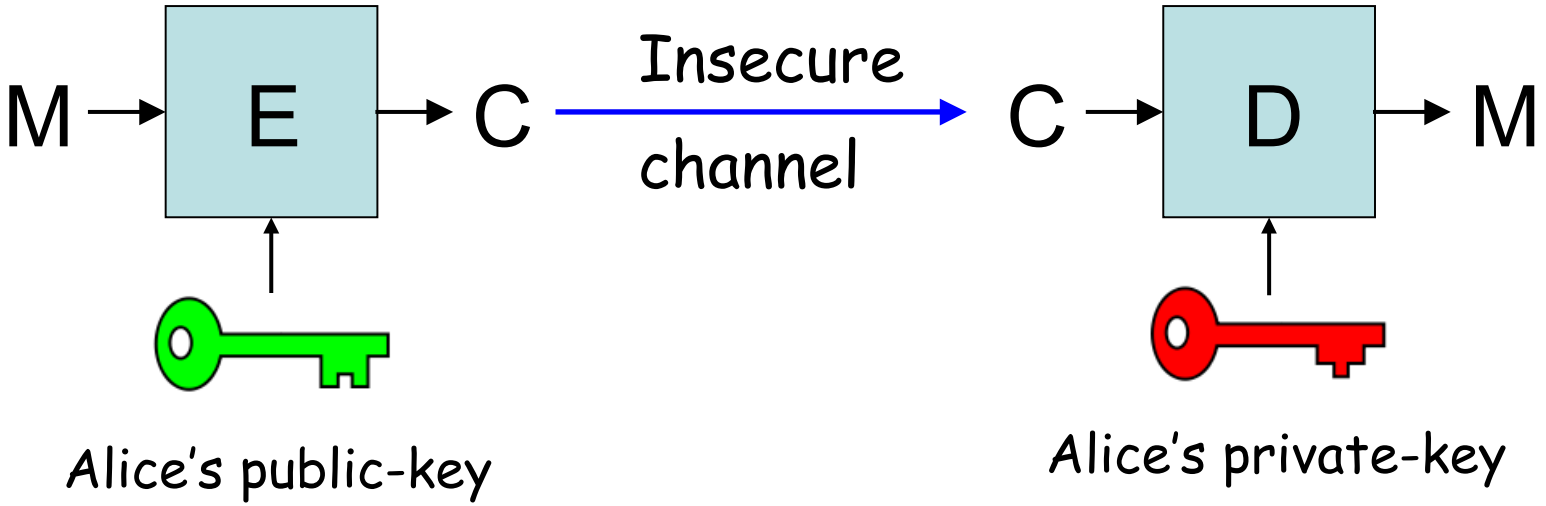


Asymmetric encryption

- Solves the key distribution issue

BOB

ALICE

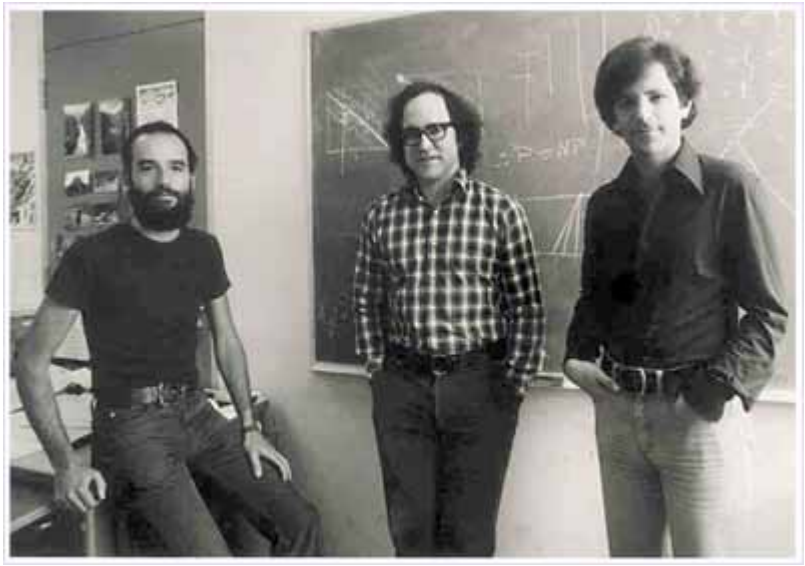


Analogy: the mailbox

- 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

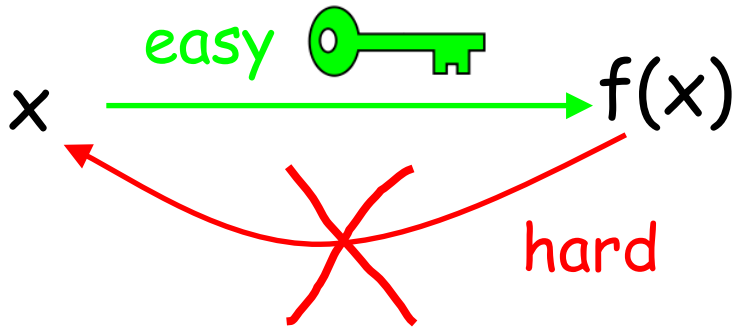
RSA (1977)

- 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.

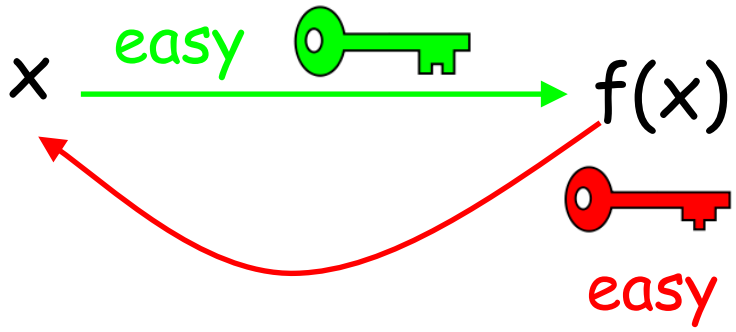


Trapdoor one-way permutation

- 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

- 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

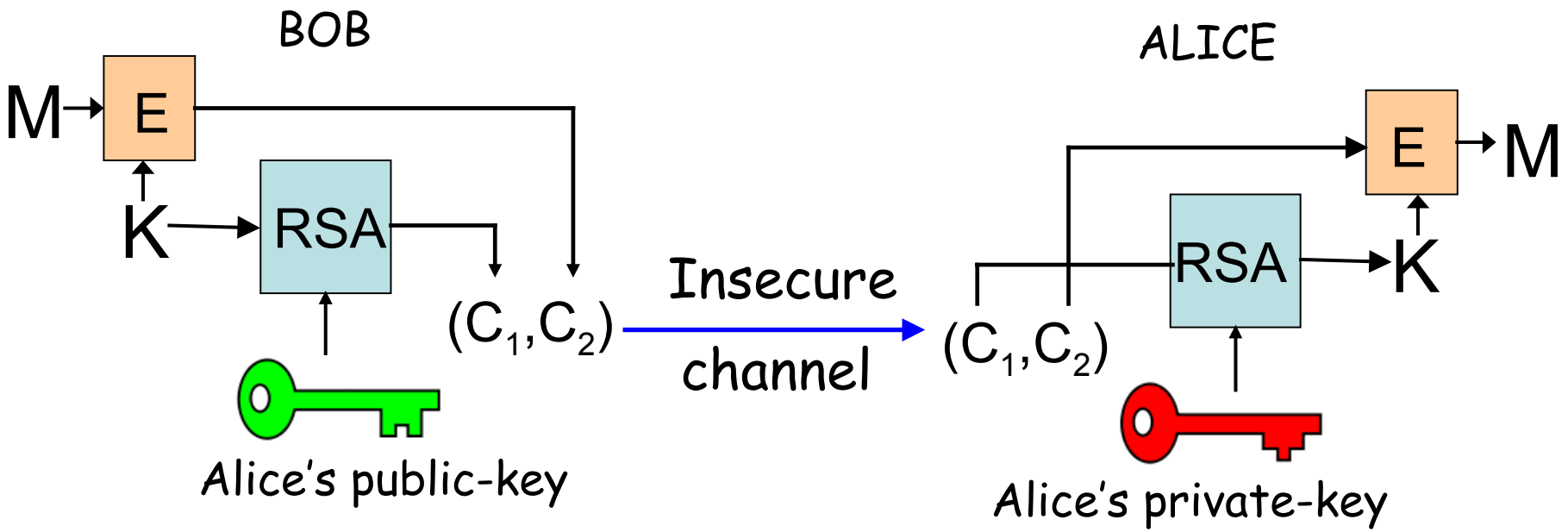
- Decryption works because $m=c^d=(m^e)^d=m^{e.d}=m$ because $e.d=1 \bmod f$
- Security is based on the hardness of factorization
  - Given $n=p.q$, no known efficient algorithm to recover p and q.
  - Factorization record: 768 bits (2010)
- Public modulus n must be large enough
  - At least 1024 bits. 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: 663 bits (2005)
- 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: $m(m)=0002 \| r \| 00 \| m$, then $c=m(m)^d [n]$
  - Decryption: recover m(m), check redundancy.
- Bleichenbacher's attack against PKCS#1 v1.5
  - Appeared in 1998. Could be use against web-servers using SSL protocol.

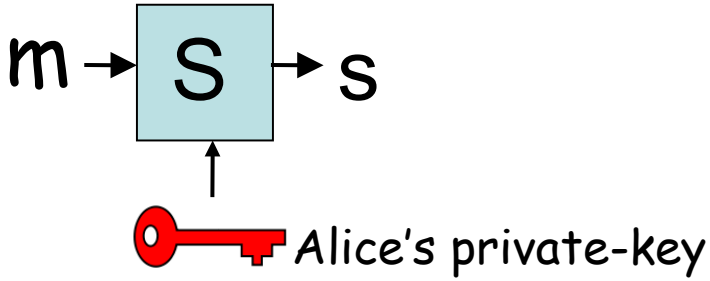
Security of RSA (and other cryptosystems)

- 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 ?

Digital signature

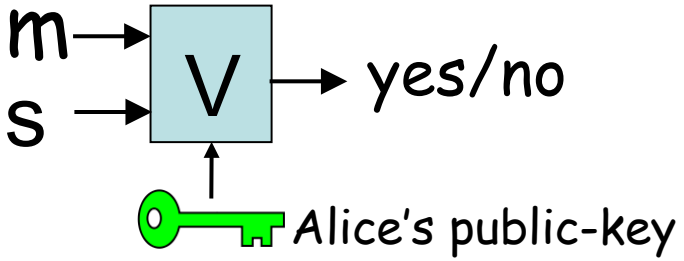
- A bit string that depends on the message m and the user's public-key
  - Only Alice can sign a message using her private-key

ALICE



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

BOB



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

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.

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

$B = g^b$

B →

← A

$A = g^a$

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

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

$K_B = K_A$

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.