

Introduction to Cryptography

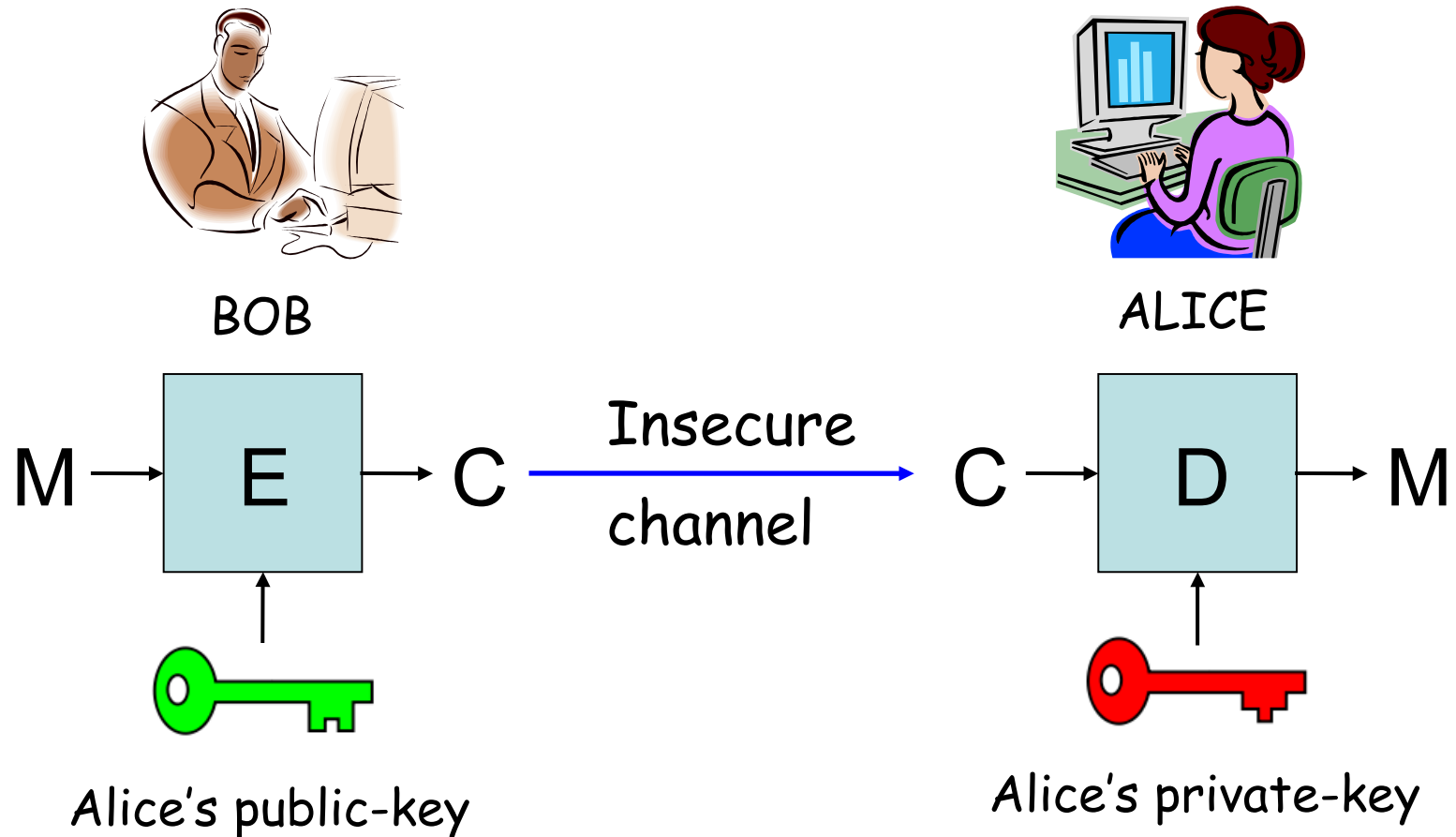
Public-key cryptography :
encryption and signatures

Jean-Sébastien Coron

September 2014

Public-key encryption

- Solves the key distribution issue



RSA

- Invented by Rivest, Shamir and Adleman in 1977.
- Still the most widely used PK algorithm.
- Public key: $n=p.q$ and e
 - Primes p and q remain secret.
- Private key: d such that
$$e.d=1 \pmod{(p-1)(q-1)}$$

RSA

- Encryption using public n, e :

$$c = m^e \bmod n$$

- Decryption using private d :

$$m = c^d \bmod n$$

- Decryption works because:

$$m = c^d = (m^e)^d = m^{e \cdot d} = m$$

because:

$$e \cdot d = 1 \bmod \phi$$

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

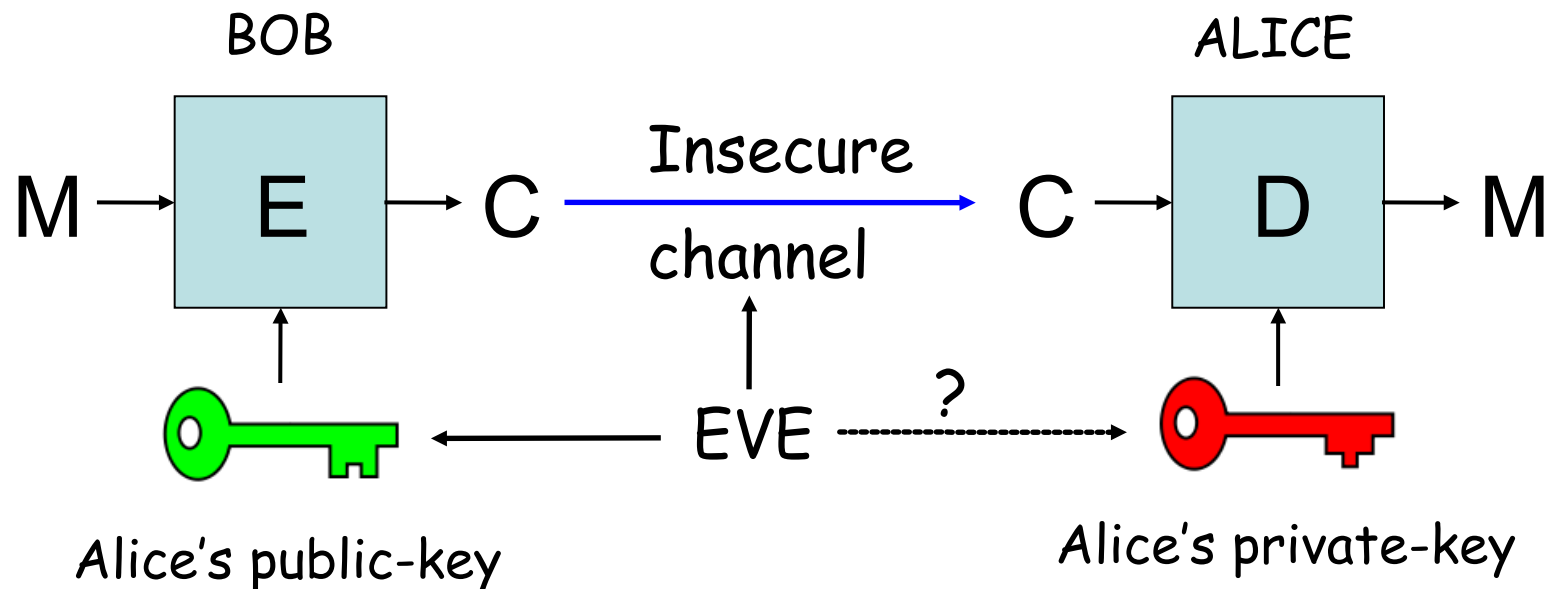
- One cannot use plain RSA encryption
 - one must add some randomness
 - one must apply some preformatting to the message
- Example: PKCS#1 v1.5
 - Encryption: $m(m) = 0002 \parallel r \parallel 00 \parallel m$, then $c = m(m)^d [n]$
 - Decryption: recover $m(m)$, check redundancy.
- Bleichenbacher's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

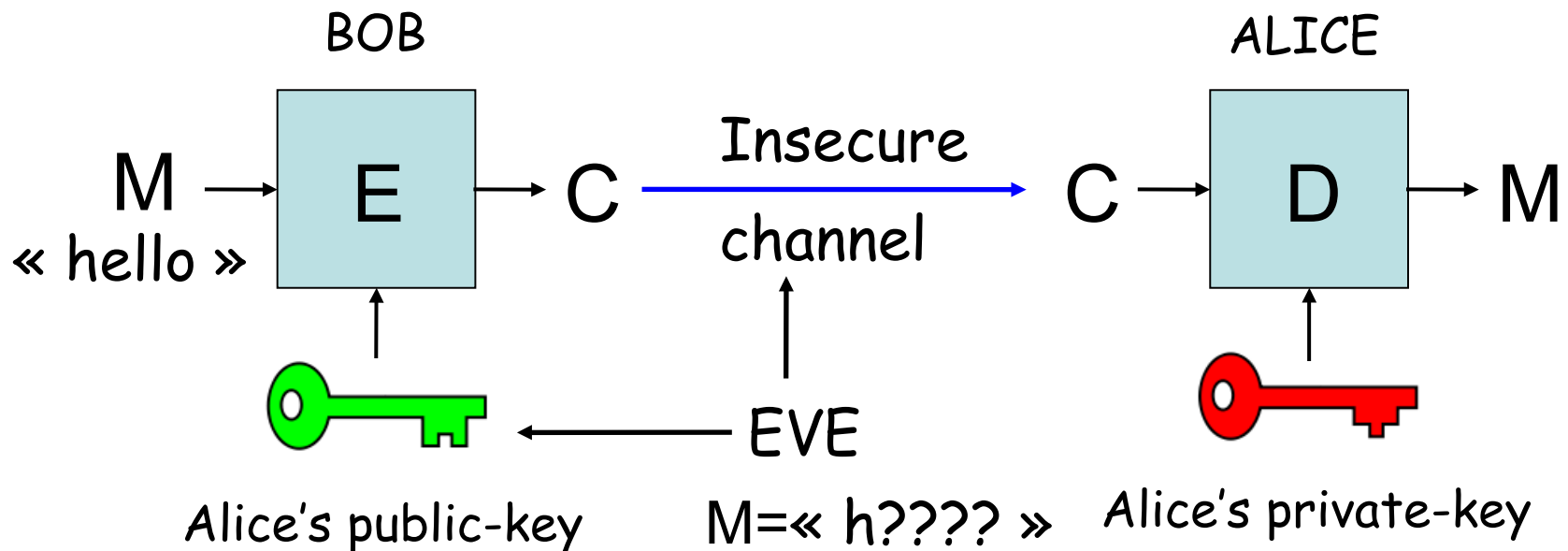
Attacker's goal

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



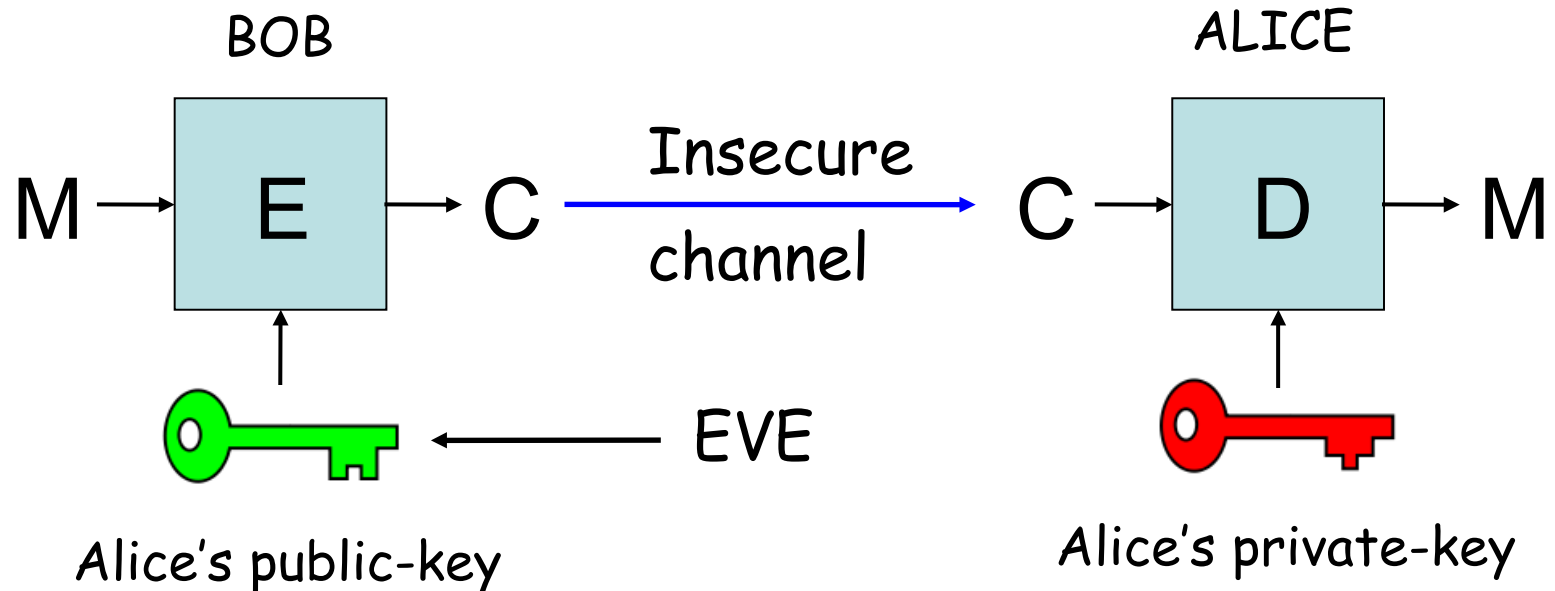
Attacker's goal

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



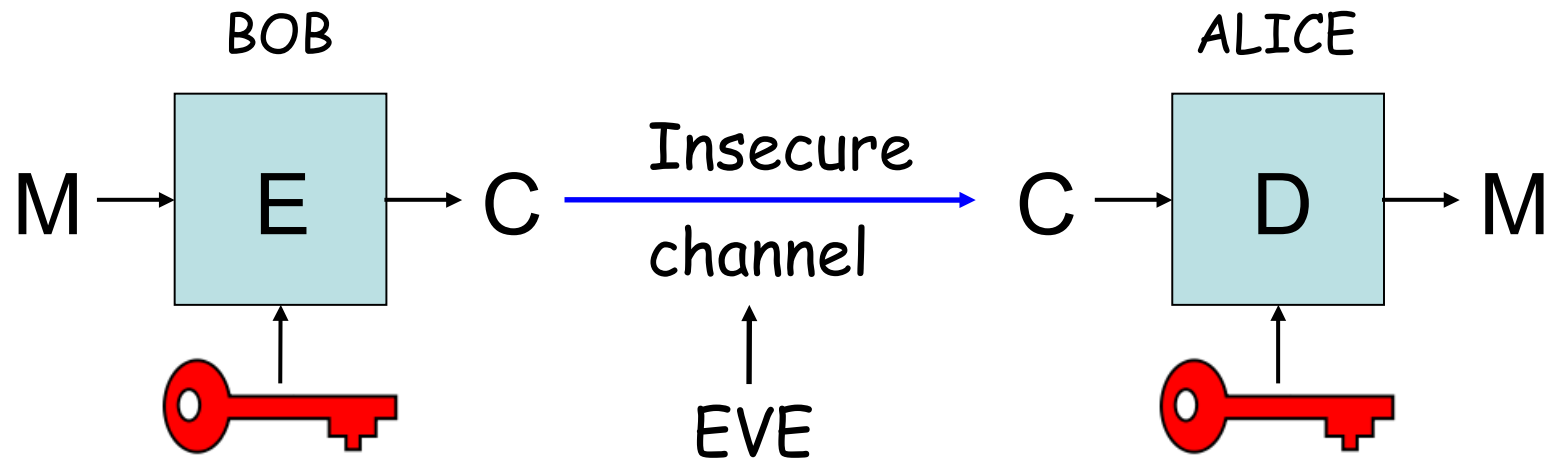
Attack models

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



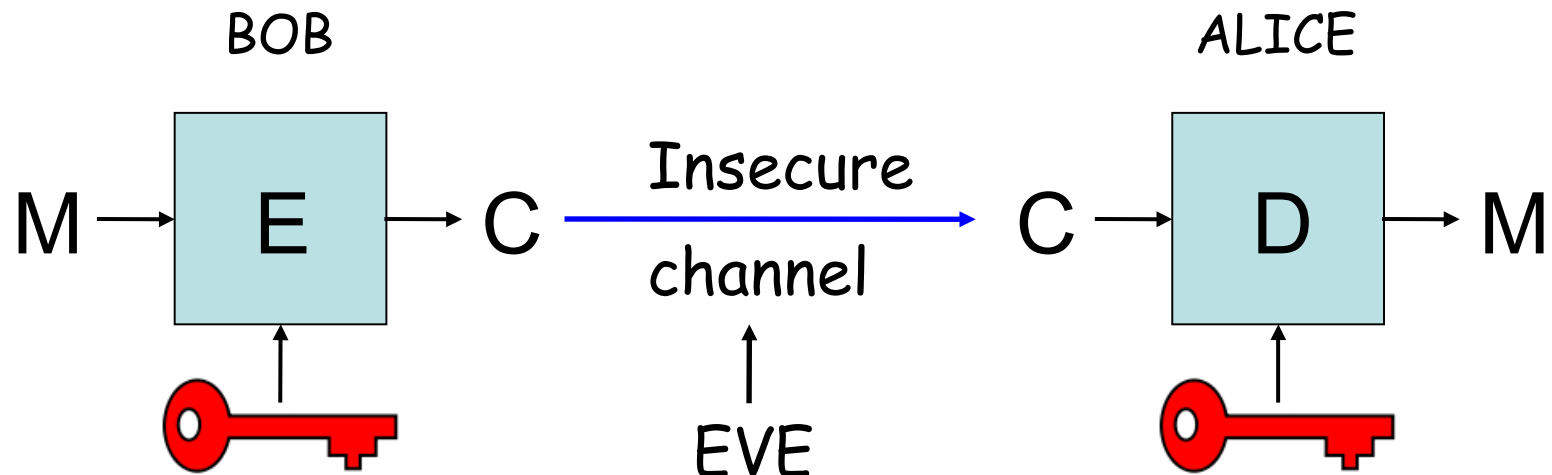
Attack models

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



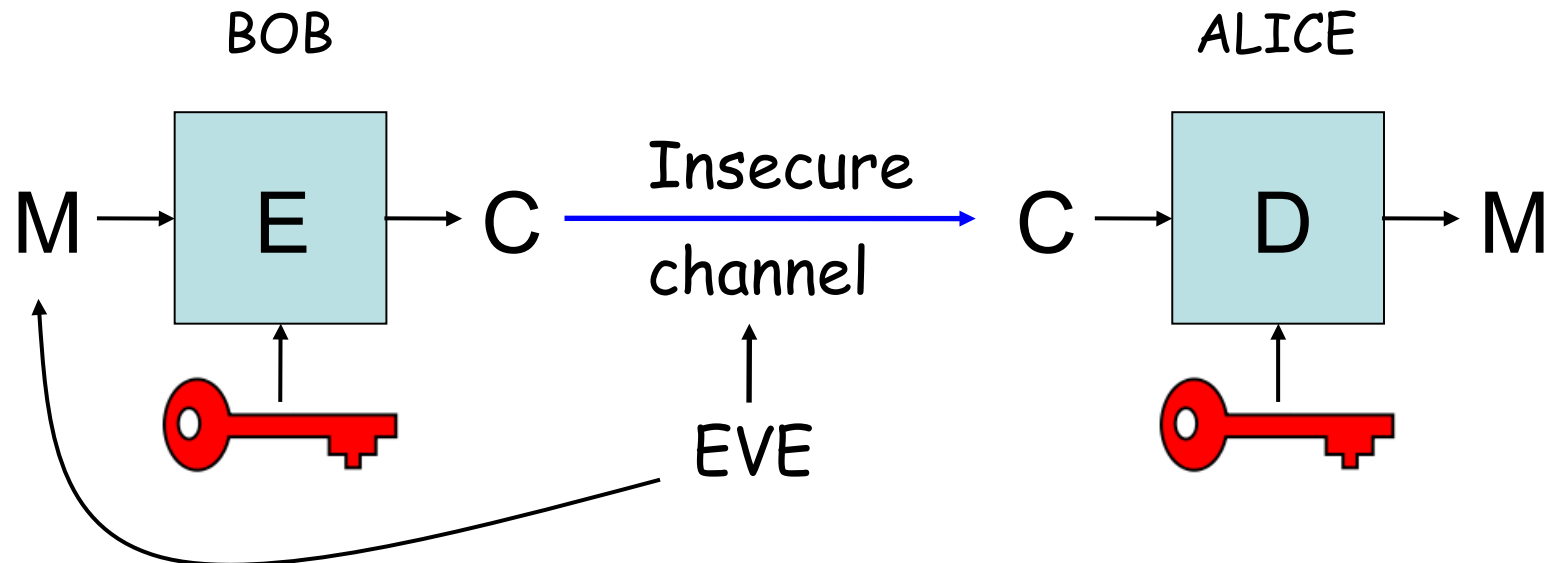
Attack models

- 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.



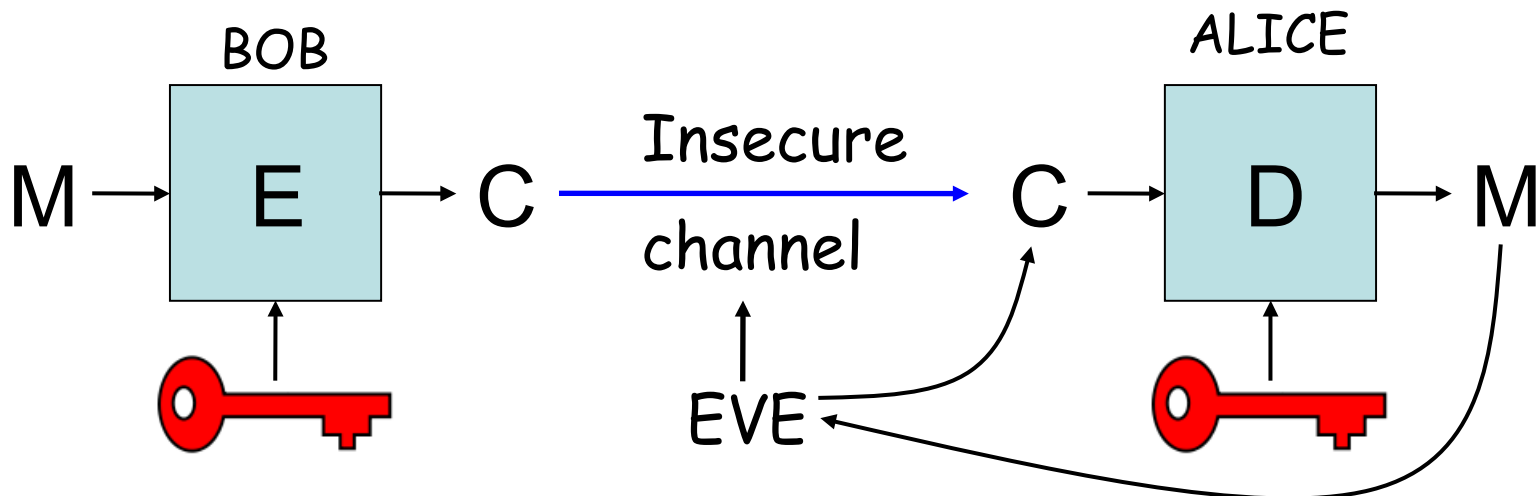
Attack models

- 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

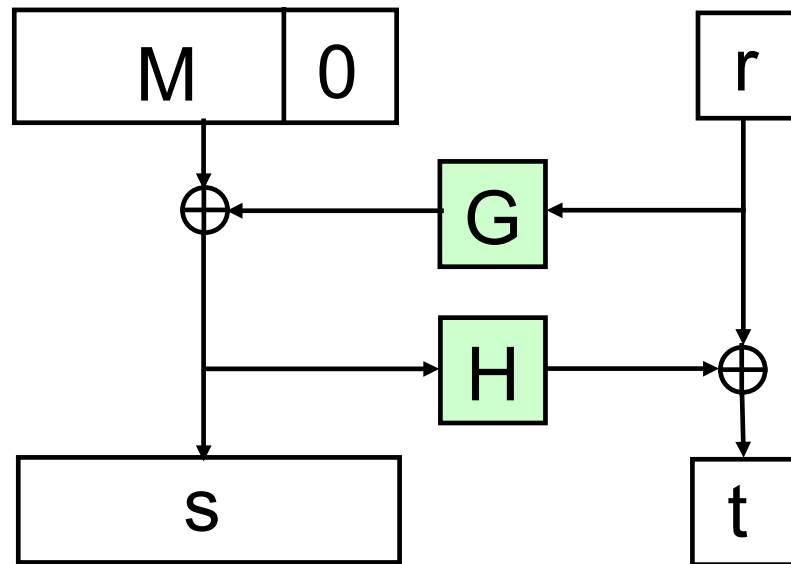
- Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2)
 - Formalized in 1991 by Rackoff et Simon
 - A ciphertext should give no information about the corresponding plaintext, even under an adaptive chosen-ciphertext attack.
 - Has become standard security notion for encryption.

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.

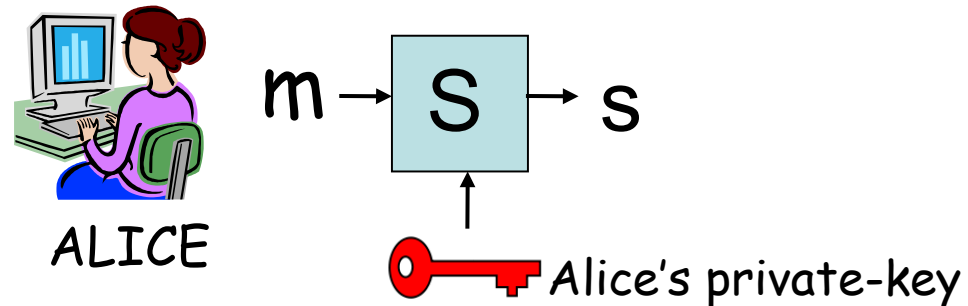
OAEP

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

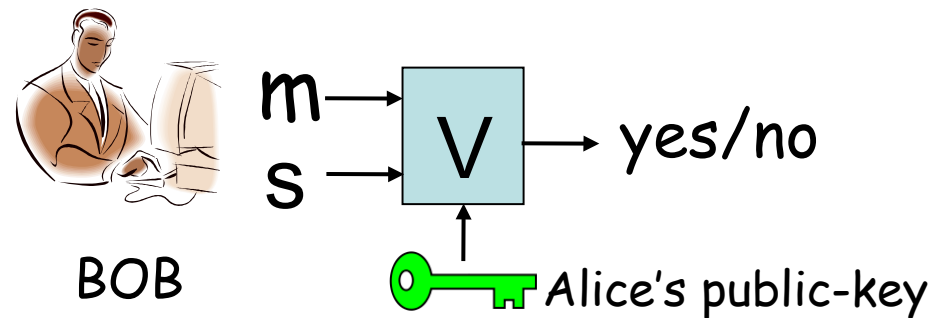


Digital signature

- 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
$$e.d=1 \pmod{(p-1)(q-1)}$$
- Signing using private d :
$$s=m^d \pmod n$$
- Verifying using public n,e :
check that $m=s^e \pmod n$
- 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.

RSA signature

- To prevent these attacks, one uses a hash function
 - PKCS#1 v1.5 :
 $m(m) = 0001 \text{ FF } \dots \text{ FF00} \mid c \mid H(m)$
 - ISO 9796-2:
 $m(m) = 6A \mid m[1] \mid H(m) \mid BC$

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.

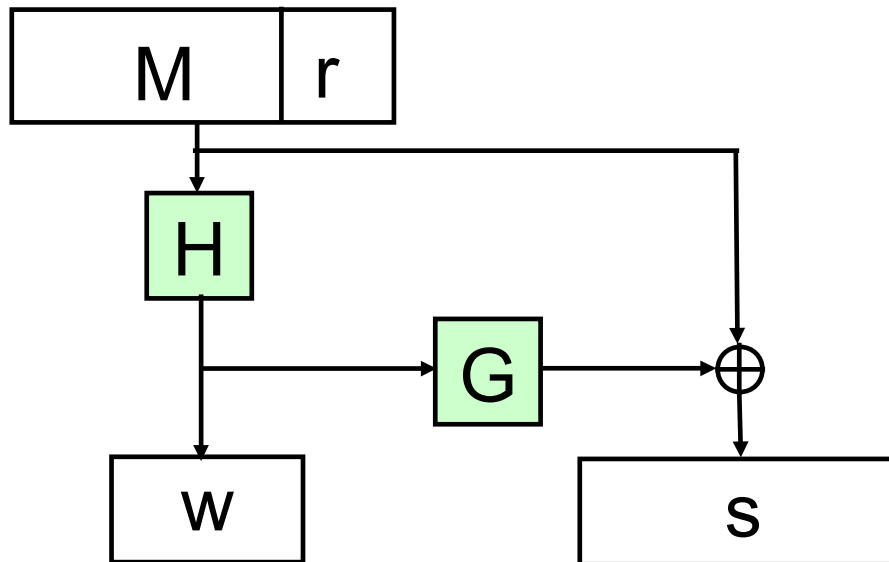
Strongest security notion

- 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.



$$s = (w \parallel s)^d \pmod{n}$$

Conclusion

- What is cryptography ?
 - Cryptography's aim is 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 satisfy the definition, in a provable way.