

Crittosistemi asimmetrici

Diffie-Hellman

- Nel loro celebre lavoro del 1976 [1], Whitfield Diffie e Martin Hellman introdussero il paradigma della crittografia asimmetrica come soluzione per **scambio di chiavi**, **cifratura** e **autenticazione** (firma digitale).



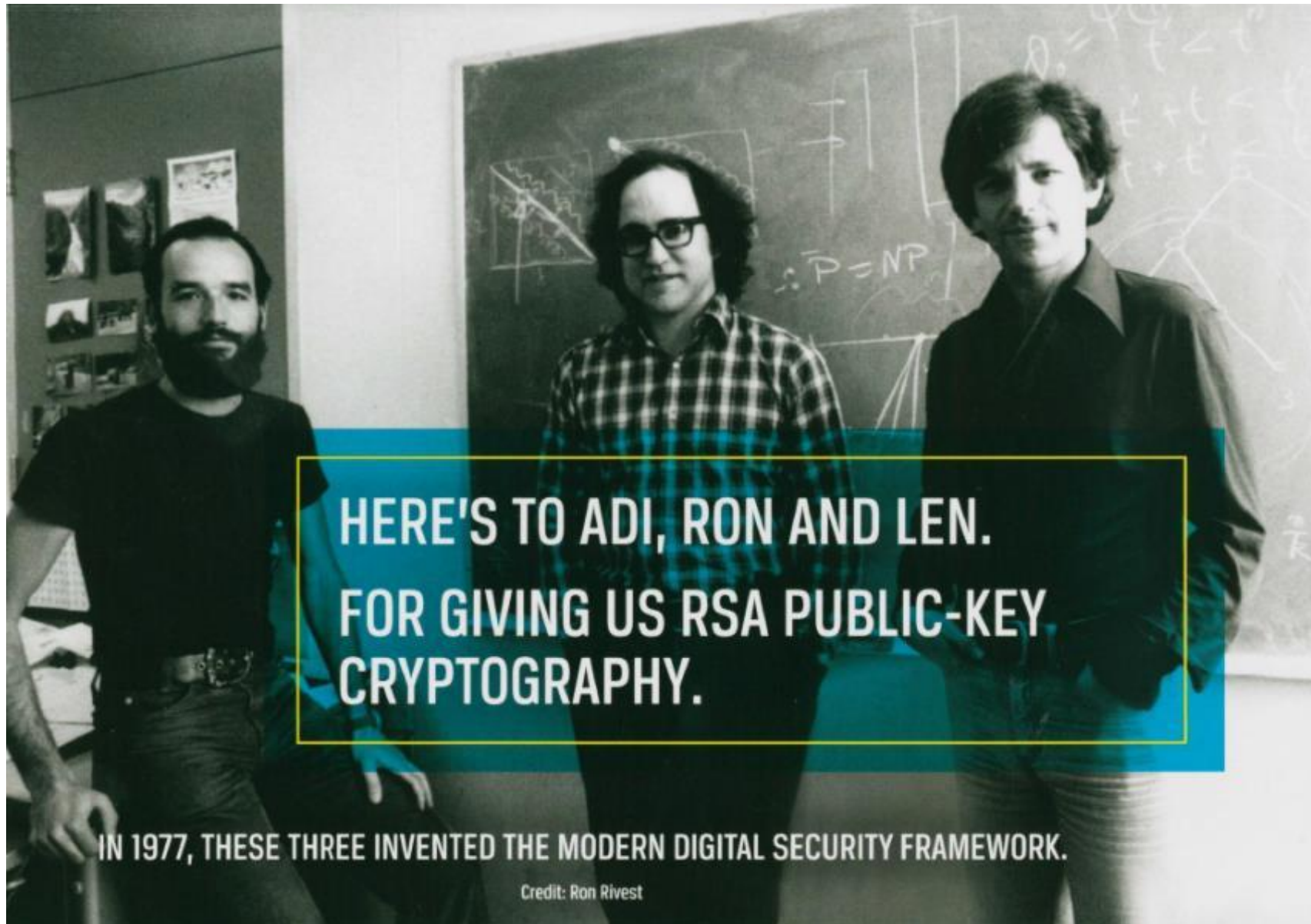
[1] W. Diffie and M. Hellman, "New directions in cryptography," in *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, November 1976.

RSA

- In termini pratici, Diffie e Hellman introdussero una procedura per lo **scambio di chiavi** basata sul logaritmo discreto
- Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman proposero un metodo per la **cifratura asimmetrica** basato sulla fattorizzazione di numeri interi



RSA



Prima di loro...



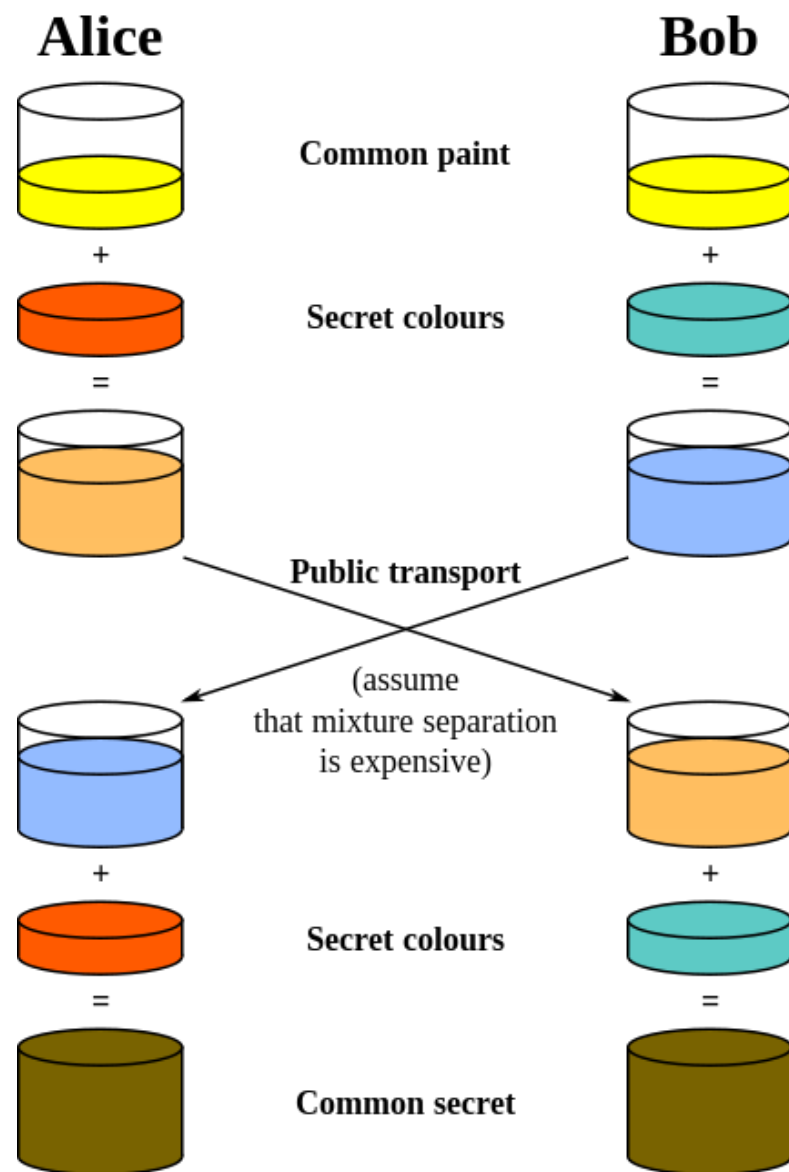
- **James Henry Ellis** nel 1970 concepì l'idea di una “cifatura non segreta”, ovvero della crittografia a chiave pubblica
- **Clifford Cocks** nel 1973 ideò lo schema che noi oggi conosciamo come RSA
- **Malcolm John Williamson** nel 1974 ideò lo schema che noi oggi conosciamo come “scambio di chiavi Diffie–Hellman”
- Tutte queste informazioni furono considerate classificate, pertanto rimasero segrete
- Solo nel 1997 il governo Britannico le ha declassificate, rendendo pubblico il contributo di Ellis, Cocks e Williamson



Condivisione di segreti su canale pubblico



Condivisione di segreti su canale pubblico

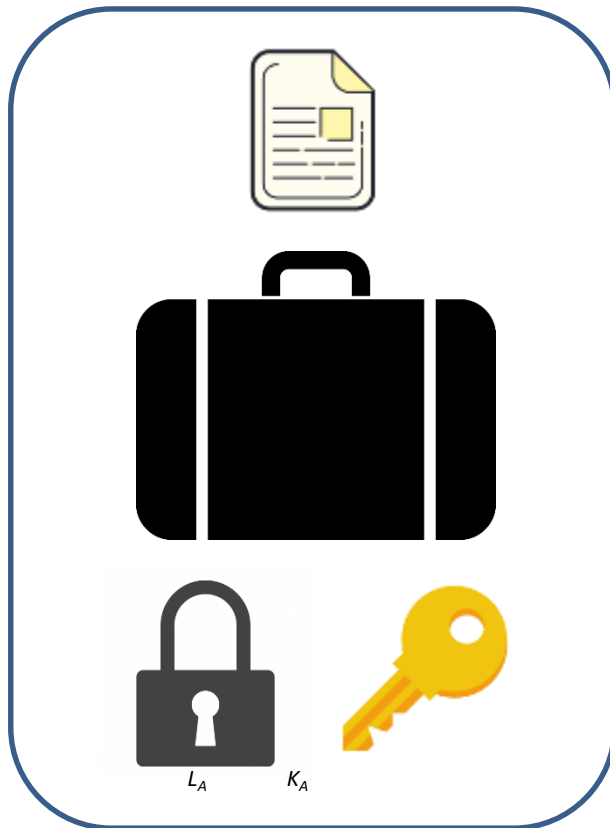


Attacco man-in-the-middle (MITM)

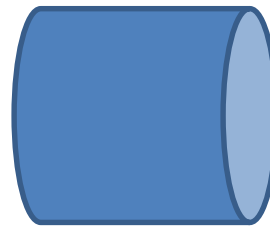
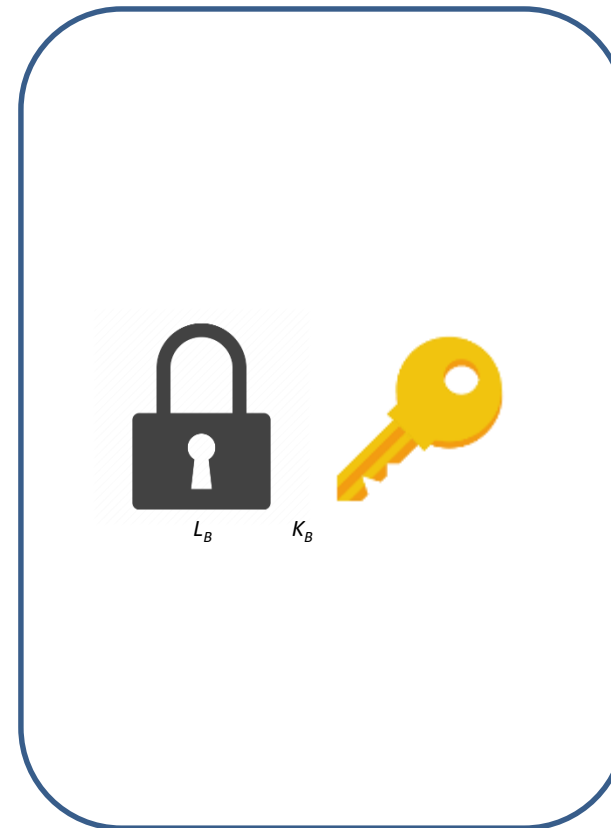


Comunicazioni segrete su canale pubblico

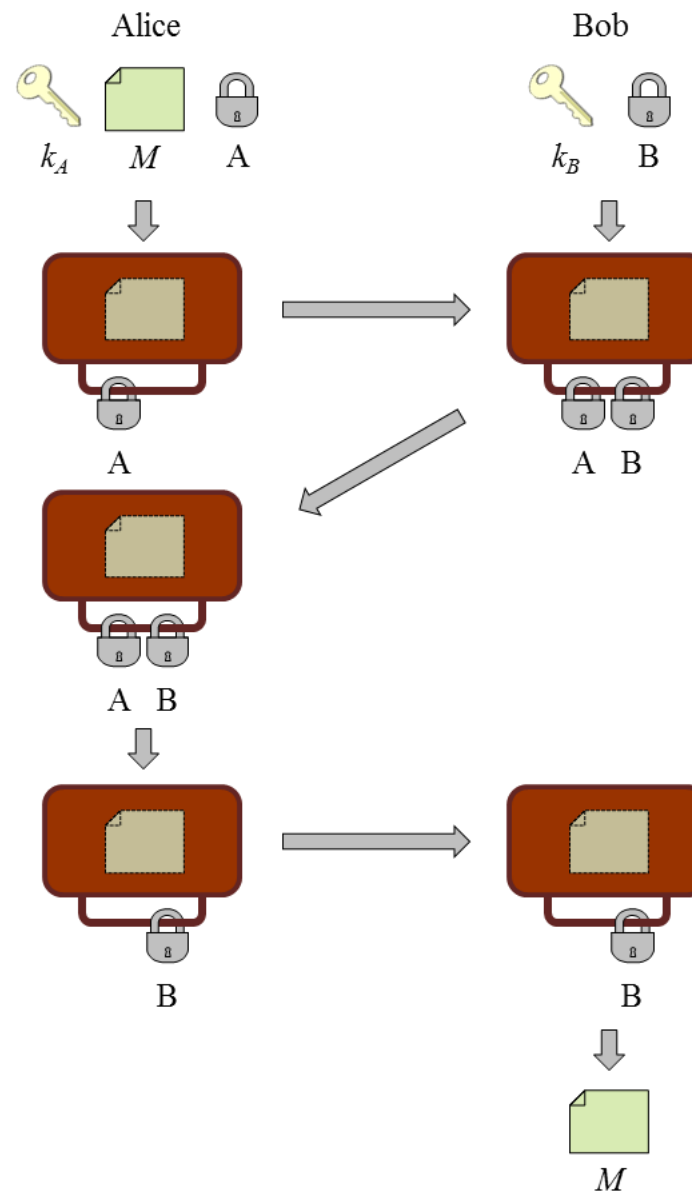
Alice



Bob



Comunicazioni segrete su canale pubblico



Implementazione (I)

Esempio 1: plaintext = 6

L_A = somma 3

L_B = sottrae 5

→ Alice cifra: $6 + 3 = 9$

Bob cifra: $9 - 5 = 4$

Alice decifra: $4 - 3 = 1$

Bob decifra: $1 + 5 = 6$

⇒ Il messaggio originale è recuperato

Implementazione (II)

Esempio 2: plaintext = 6

L_A = moltiplica per 2

L_B = sottrae 4

→ Alice cifra: $6 \times 2 = 12$

Bob cifra: $12 - 4 = 8$

Alice decifra: $8 : 2 = 4$

Bob decifra: $4 + 4 = 8$

⇒ Il messaggio originale non è recuperato

Implementazione (III)

- Normalmente, l'ordine di cifratura e decifratura deve essere speculare:

$$C_1 \rightarrow C_2 \rightarrow D_2 \rightarrow D_1$$

- Nel protocollo visto, serve invece poter invertire l'ordine di decifratura:

$$C_A \rightarrow C_B \rightarrow D_A \rightarrow D_B$$

- Ciò è possibile nell'Esempio 1 (operazioni commutative):

$$(6 + 3) - 5 = (6 - 5) + 3$$

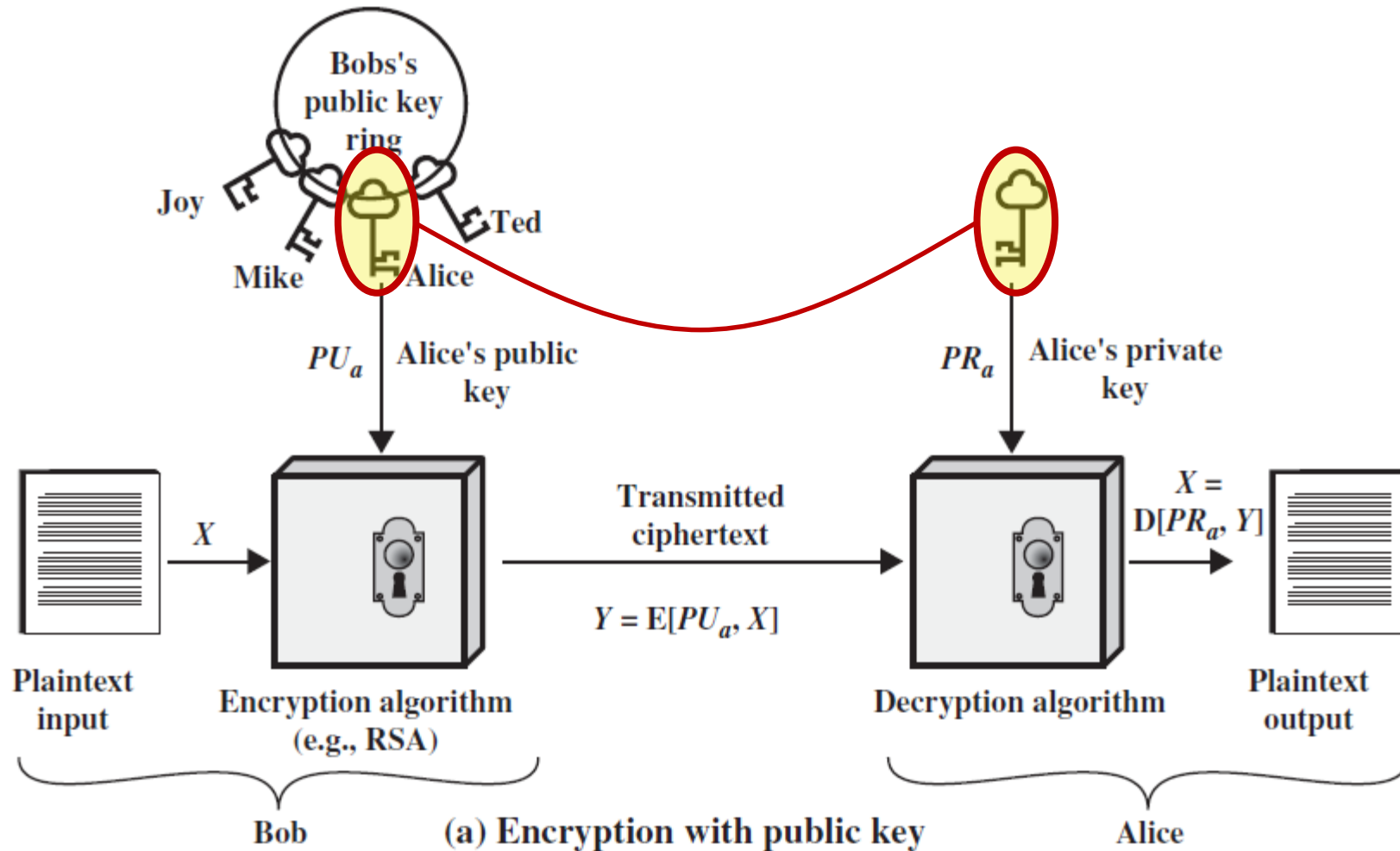
- Ciò non è possibile nell'Esempio 2 (operazioni non commutative):

$$(6 \times 2) - 4 \neq (6 - 4) \times 2$$

- Le operazioni commutative però non sono sicure!
- Nell'Esempio 1, Eve vede 9 (Alice encryption), 4 (Bob encryption) e 1 (Alice decryption) e può facilmente risalire al meccanismo che li ha generati.

Algoritmi a chiave asimmetrica

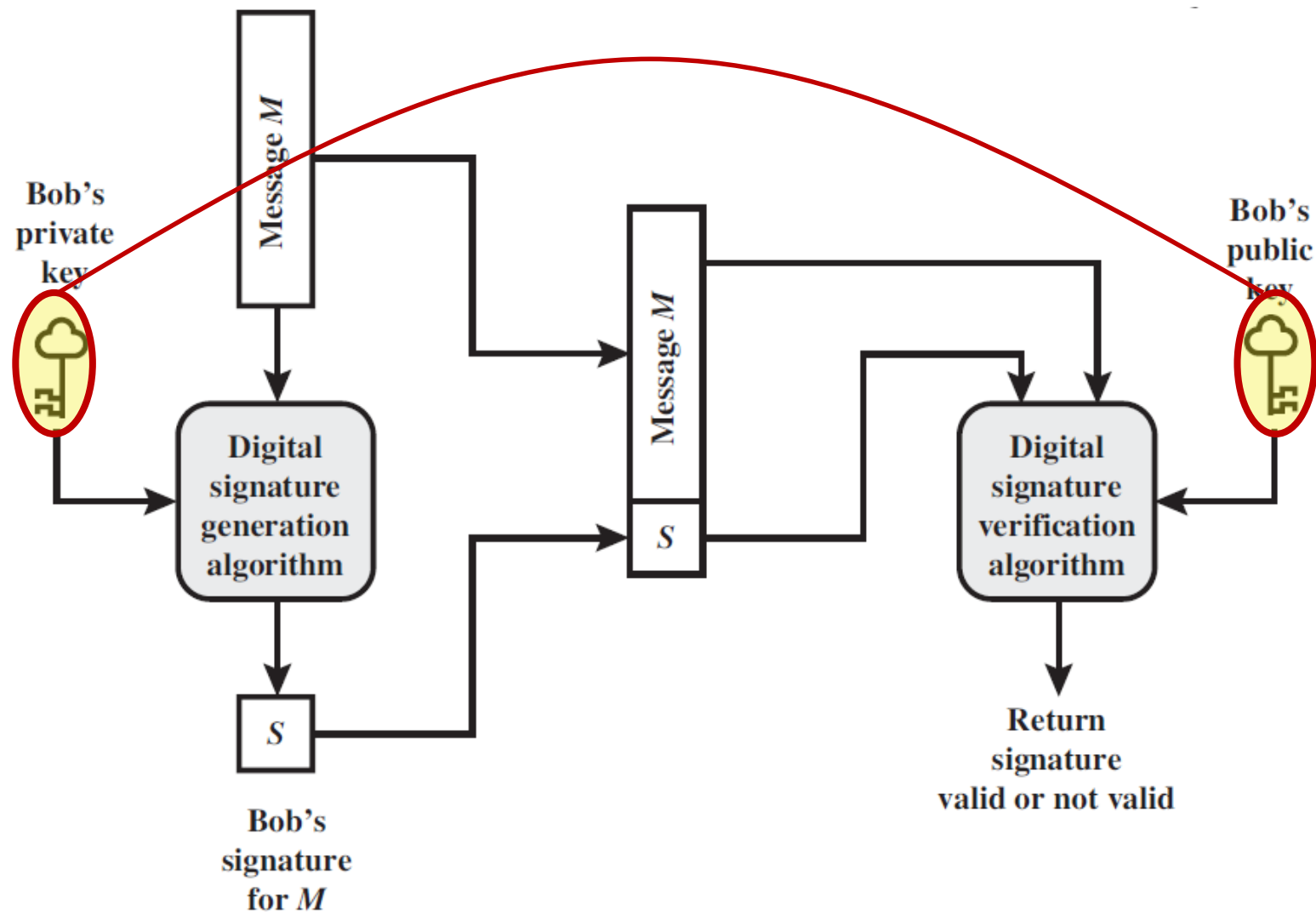
- Esempi: RSA, El Gamal, curve Ellittiche



Esempi di cifrari asimmetrici

- Diffie-Hellmann
- RSA
- Elliptic Curve Cryptography (ECC)
- El Gamal
- *post-quantum...*

Firma digitale



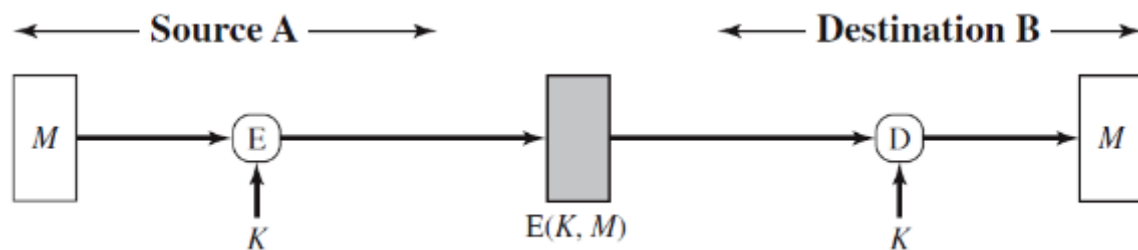
Esempi di schemi di firma digitale

- RSA
- El Gamal
- Digital Signature Algorithm (DSA)
- ECC
- *post-quantum...*

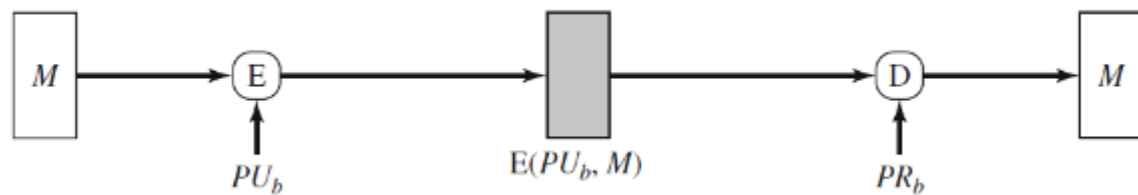
Algoritmi simmetrici e asimmetrici

- Gli algoritmi a chiave **asimmetrica**:
 - richiedono tipicamente un **onere computazionale** maggiore degli algoritmi simmetrici
 - non sono usati per la cifratura di grandi **quantità** di dati
 - si utilizzano per le **firme digitali** e l'invio di **chiavi** per il successivo utilizzo in algoritmi a chiave simmetrica

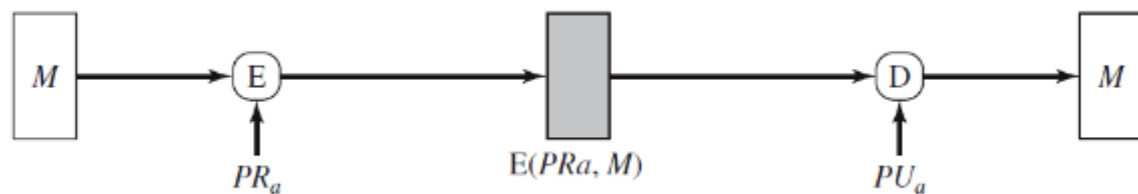
Cifratura simmetrica e asimmetrica



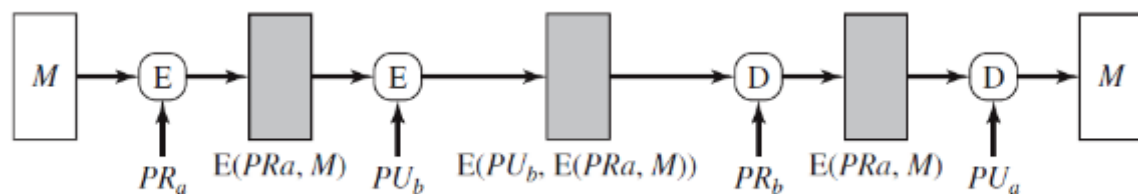
(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality



(c) Public-key encryption: authentication and signature



(d) Public-key encryption: confidentiality, authentication, and signature

Trapdoor (funzione)

- È l'elemento basilare della crittografia asimmetrica
- Chiunque può calcolare la **funzione diretta**
- Solo chi conosce un segreto può calcolare la **funzione inversa**

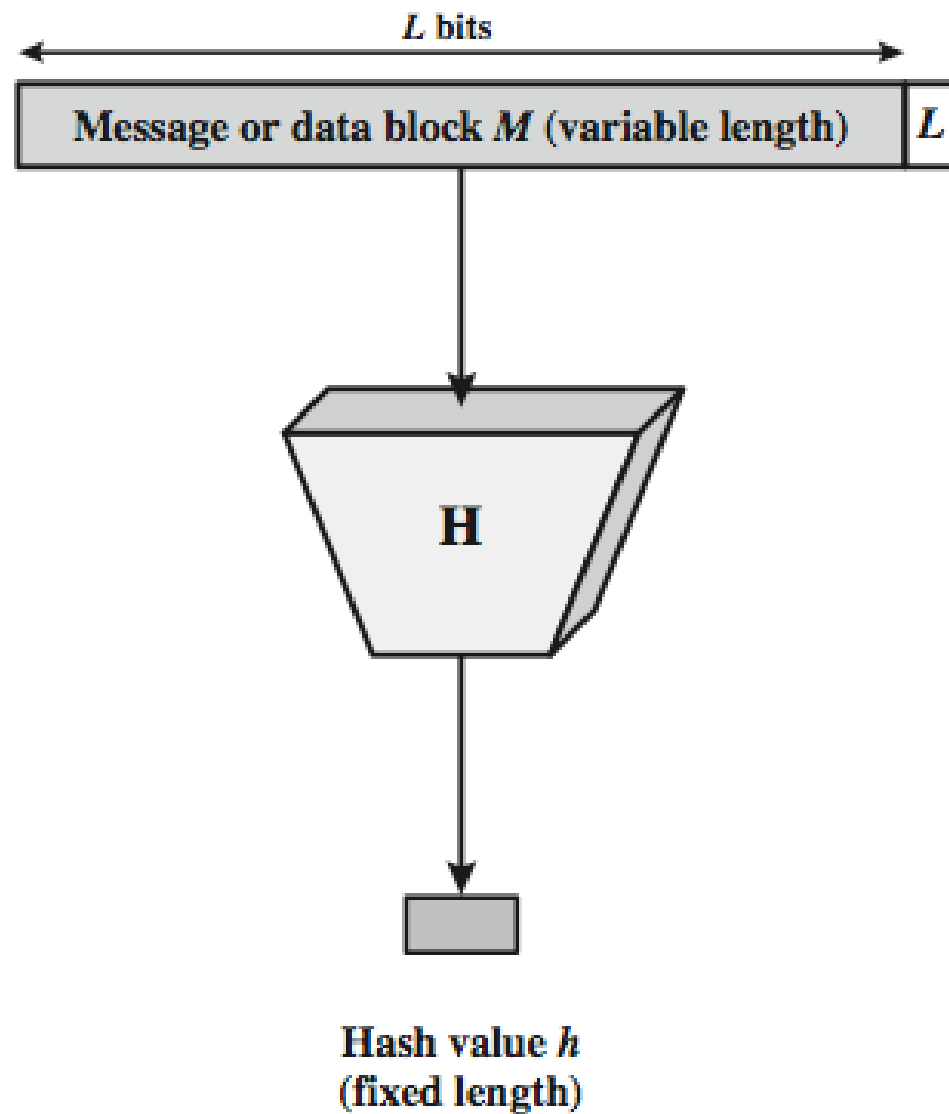


One-way (funzione)

- È una famiglia di funzioni importanti in crittografia
- Chiunque può calcolare la **funzione diretta**
- Nessun può calcolare la **funzione inversa**



Funzioni hash



Esempi di funzioni hash crittografiche

- MD4, MD5
- HAVAL-128
- RIPEMD
- SHA-1
- SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
- SHA-3 (Keccak, standardizzato dal NIST il 5 Agosto 2015)

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collisions found	Example Performance (MiB/s) ^[11]
MD5 (as reference)		128	128	512	$2^{64} - 1$	32	64	and,or,xor,rot	Yes	255
SHA-0		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Yes	-
SHA-1		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Theoretical attack (2^{51}) ^[12]	153
SHA-2	SHA-256/224	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr,rot	None	111
	SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+,and,or,xor,shr,rot	None	99

Cifrari a flusso e a blocco

- **Stream ciphers:**
 - i dati sono divisi in frammenti, composti da un singolo bit, byte o carattere, ed elaborati uno alla volta
- **Block ciphers:**
 - i bit in ingresso sono raccolti in un blocco ed elaborati tutti insieme dall'algoritmo fornendo un blocco di bit in uscita
- I cifrari **simmetrici** possono essere sia a flusso che a blocco
- I cifrari **asimmetrici** sono cifrari a blocco