Lezione del 20 Maggio 2016 - Sicurezza e MD5 hashing

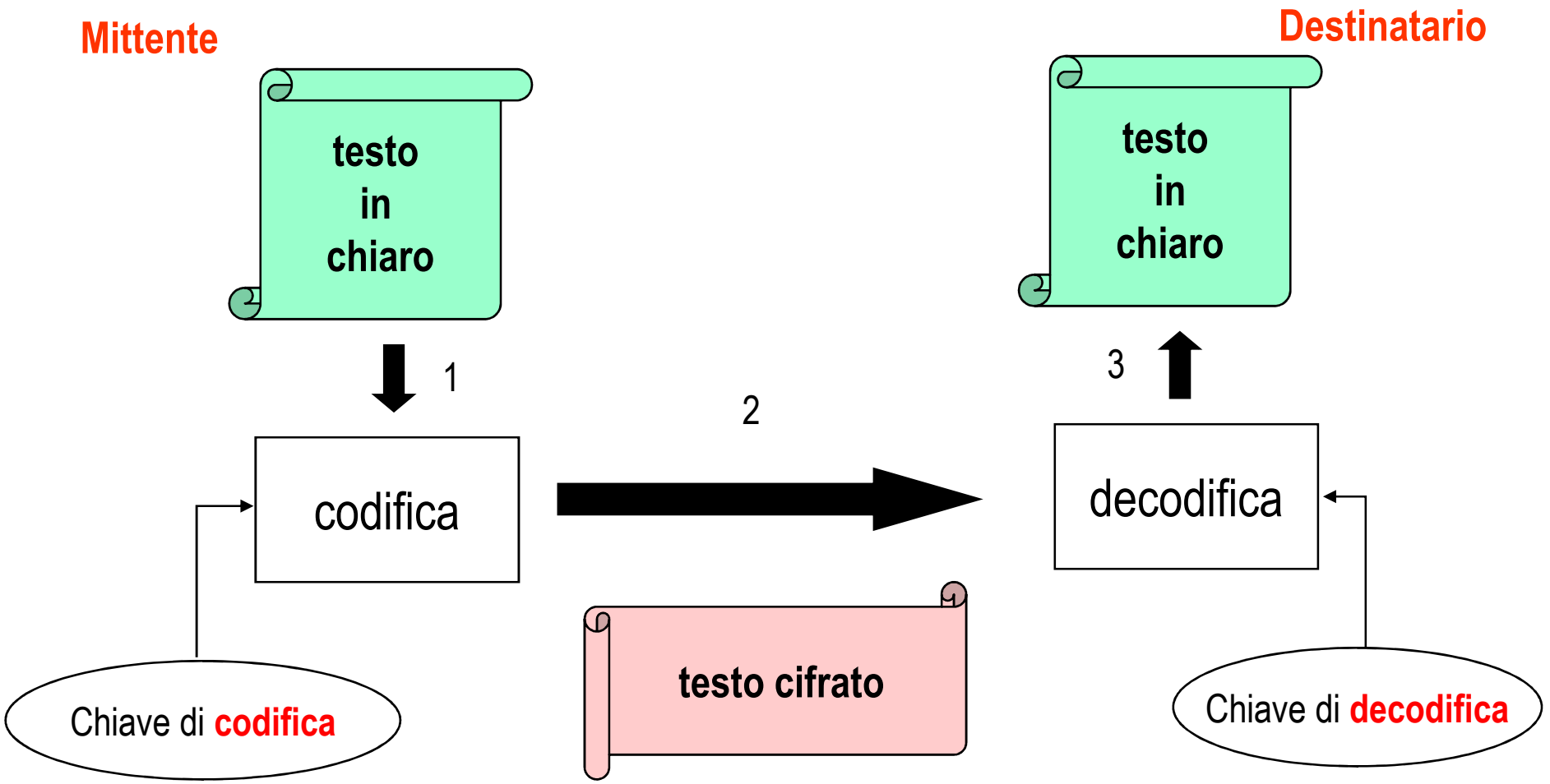
Crittografia

- Scienza antichissima: codificare e decodificare informazione
- Tracce risalenti all'epoca di Sparta
- Antica: crittografia simmetrica
- Moderna: crittografia asimmetrica (1977)

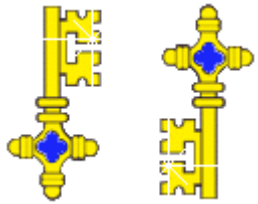
Crittografia

- Codificare: testo in chiaro → testo codificato
- Decodificare: testo codificato → testo in chiaro
- Ambedue basate su: **algoritmo** e **chiave**
  - *Es: "Shiftare" di k una stringa*
- Sicurezza data da:
  1. segretezza della chiave
  2. robustezza dell'algoritmo

Codifica e decodifica



Crittografia simmetrica



- Medesima chiave per codifica e decodifica
- Segretezza, autenticazione, integrità dalla segretezza della chiave

\+ Di solito (*DES*) usano chiavi di 64-128 bit (17-34 cifre decimali) e sono molto veloci

\- Distribuire chiave a coppie di utenti

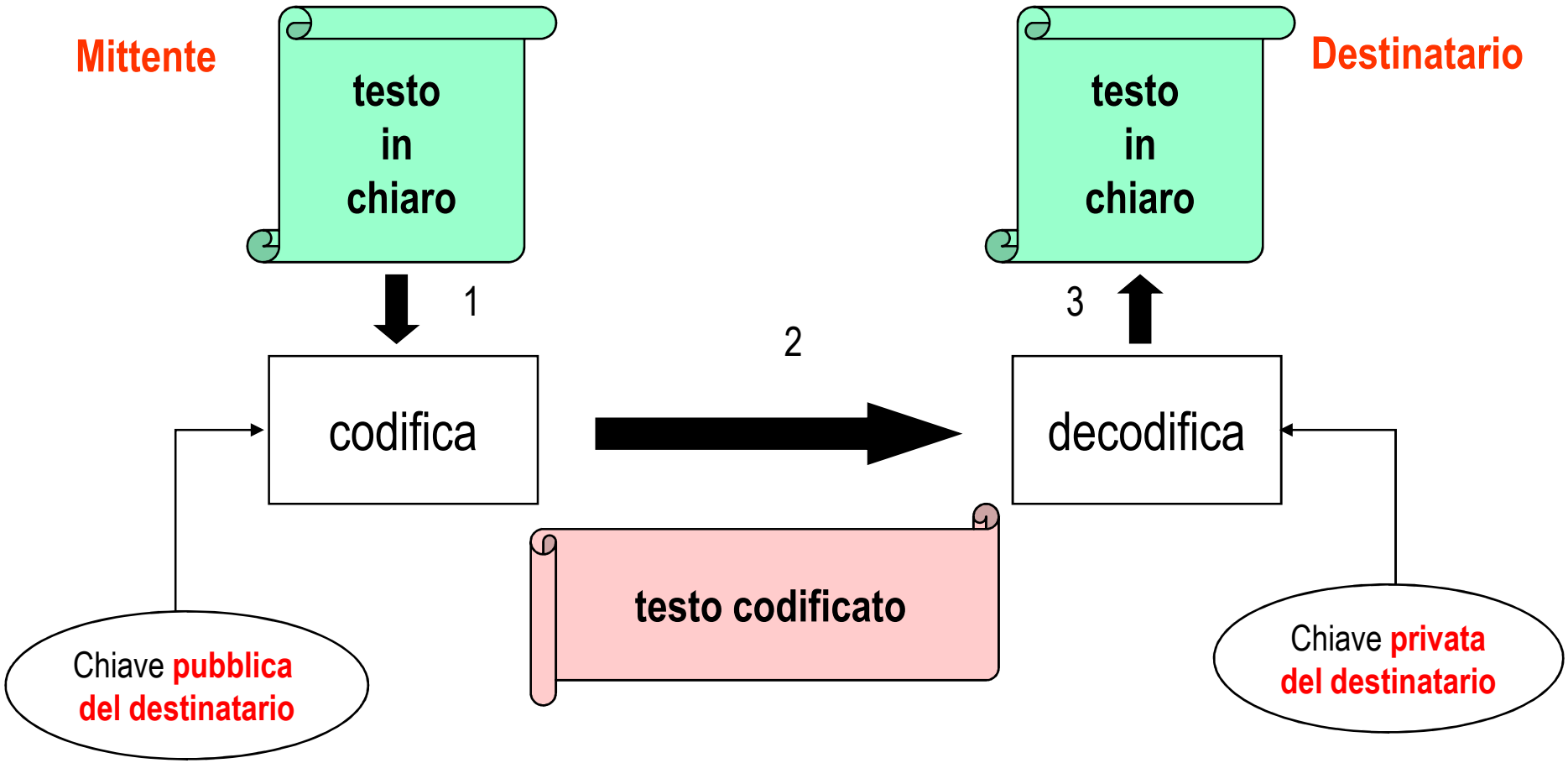
Crittografia asimmetrica

- Una chiave per codifica, un'altra per decodifica
- Ogni utente ha una coppia di chiavi
  - **chiave privata**: segreto da custodire
  - **chiave pubblica**: informazione da diffondere
- Entrambe usabili per codificare o decodificare

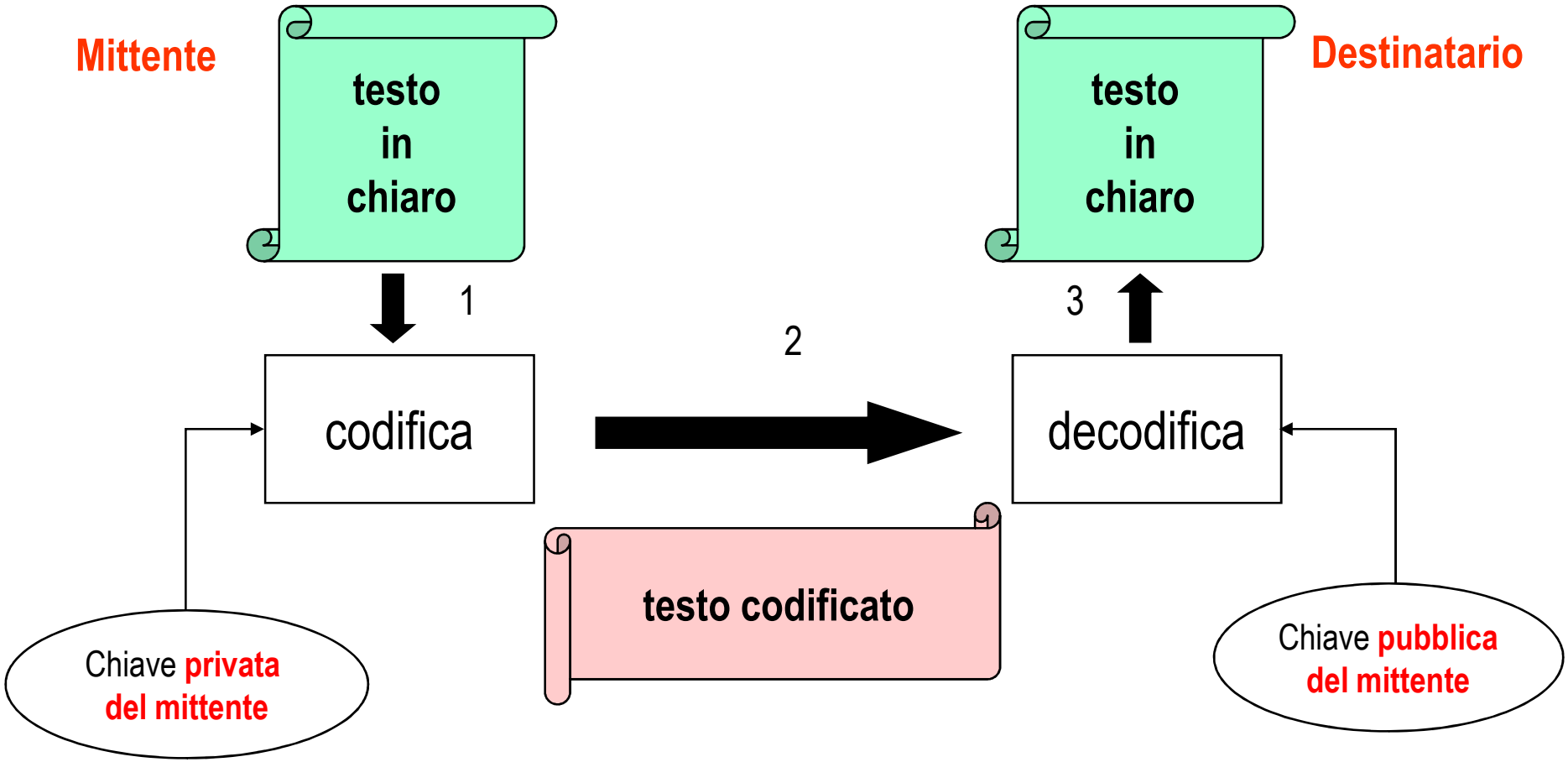
- Di solito (*RSA*) usano chiavi di 1024-2048 bit (circa 160-320 cifre decimali) e sono lenti

+ Segretezza, autenticazione, integrità…

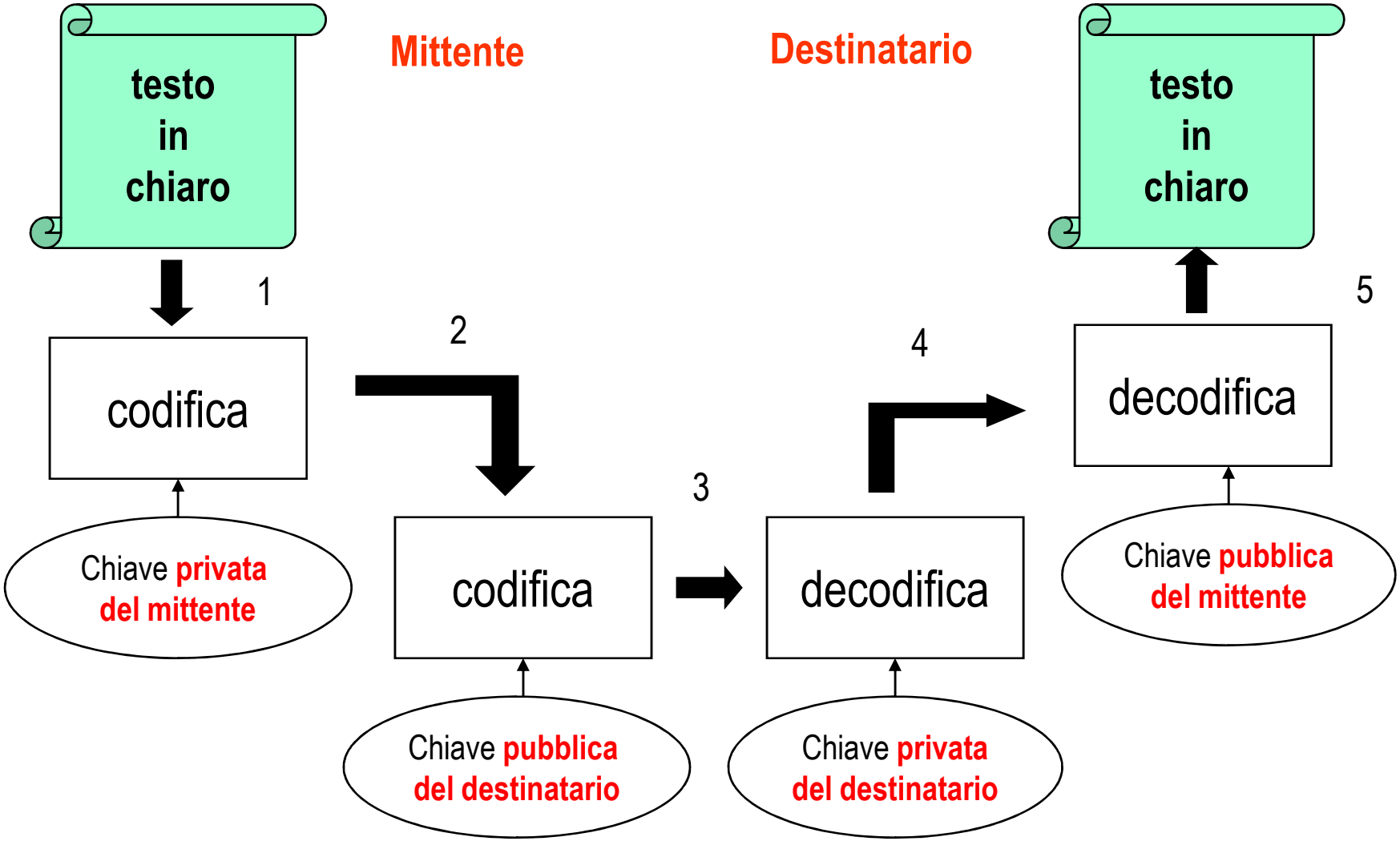
Segretezza


Mittente
testo in chiaro
1
codifica
2
testo codificato
Chiave pubblica del destinatario
Destinatario
testo in chiaro
3
decodifica
Chiave privata del destinatario

Autenticazione e integrità



Le tre insieme



Attacchi crittografici (crittoanalisi)

1. **Cyphertext only**: noto solo il testo codificato
2. **Known plaintext**: testo in chiaro noto
3. **Chosen plaintext**: testo in chiaro scelto
4. **Brute-force**: attacco alla chiave

Crittografia perfetta

Def. Nessun testo codificato rilascia informazione alcuna né sulla chiave usata per la codifica, né sul testo in chiaro, il quale può essere recuperato se e solo se la chiave è disponibile

- Ideale: nessun tipo di crittoanalisi possibile
- Probabilità nulla di ricavare informazioni supplementari da un testo codificato
- Crittografia in pratica quasi mai perfetta

Funzioni hash irreversibili (digest)

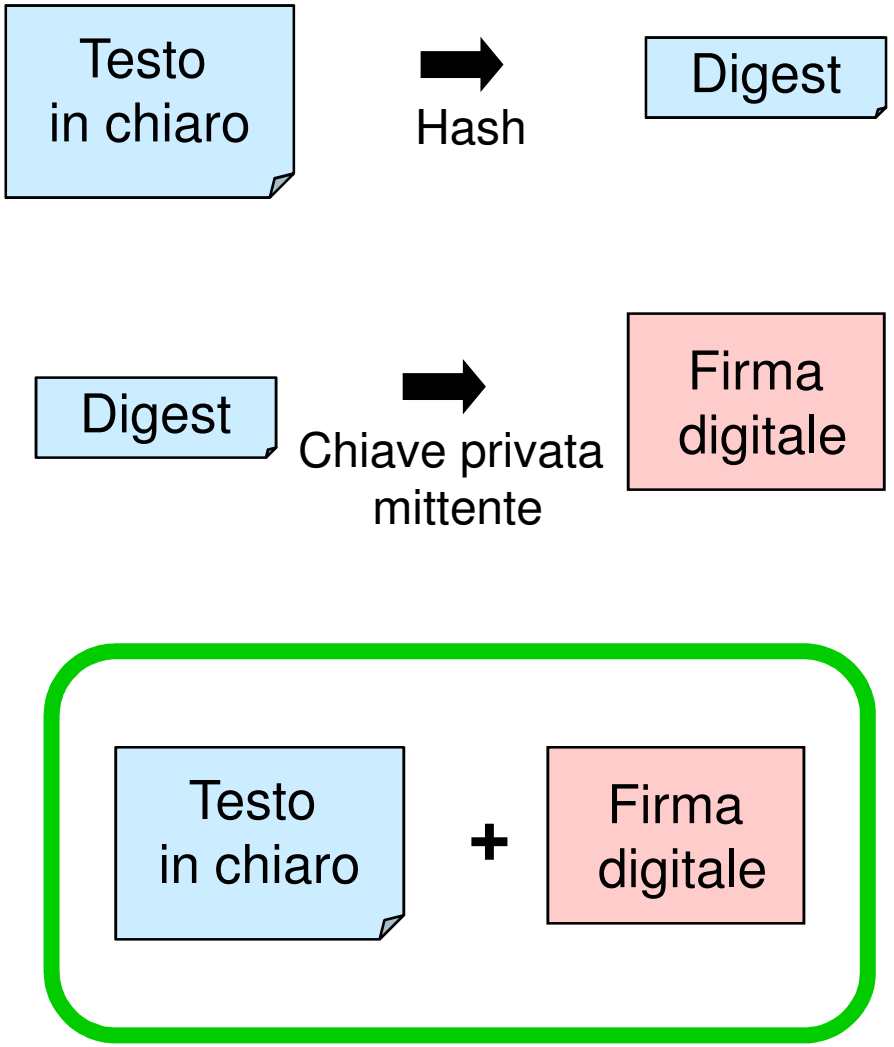
- h : X → Y è **hash** se
  1. h can be applied to a block of data at any size
  2. h produces a fixed length output
  3. Dato x ∈ X è computazionalmente facile (tempo polinomiale nella dim. dell'input) calcolare h(x)
- …è **irreversibile** se
  1. For any given block x, it is computationally infeasible to find x such that H(x) = h(x)
  2. For any given block x, it is computationally infeasible to find y≠x with H(y) = H(x).
  3. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y)
- Integrità di un testo

Firma digitale

- Basata su crittografia asimmetrica
- Ottiene solo autenticazione e integrità
- Firmare non è esattamente codificare
- Verificare una firma non è esattamente decodificare

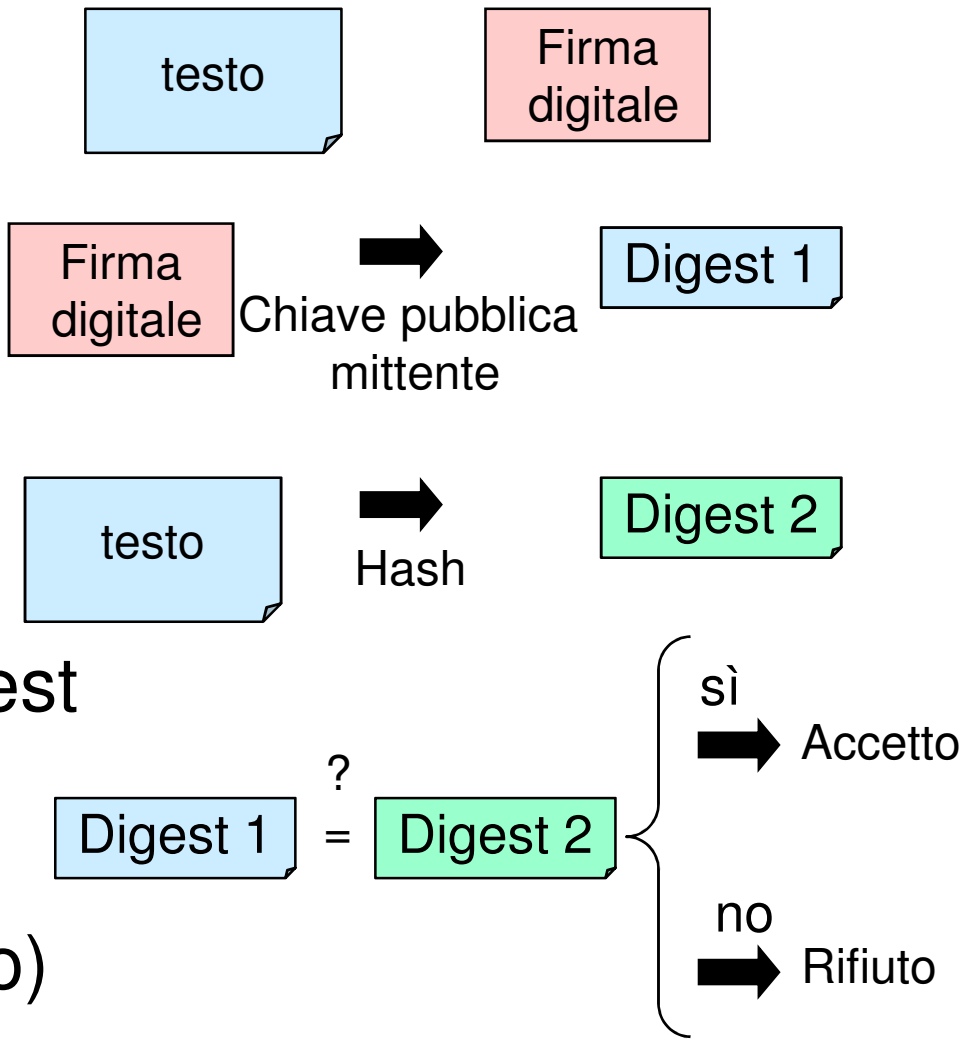
Creazione della firma

1. Calcolare il **DIGEST** del testo
2. Codificare il digest con la chiave privata del mittente (si ottiene la firma digitale vera e propria)
3. Creare coppia testo+firma e spedirla



Verifica della firma

1. Separare il testo dalla firma
2. Decodificare la firma con la chiave pubblica del mittente
3. Calcolare il digest del testo
4. Verificare che i due digest coincidano
   - sì: accetto (testo OK)
   - no: rifiuto (testo alterato)



Garanzie

La firma digitale garantisce che:

- **Autenticità: Il messaggio arrivi proprio da chi dice di essere il mittente**
- **Integrità: Il messaggio non abbia subito modifiche o manomissioni**

Autorità di certificazione

- Chi garantisce che la chiave pubblica di Bob, che otteniamo da un registro pubblico, sia stata rilasciata proprio a Bob?
- Una terza parte fidata: l'autorità di certificazione (**CA**), che certifica il legame utente/chiave pubblica mediante apposito certificato digitale

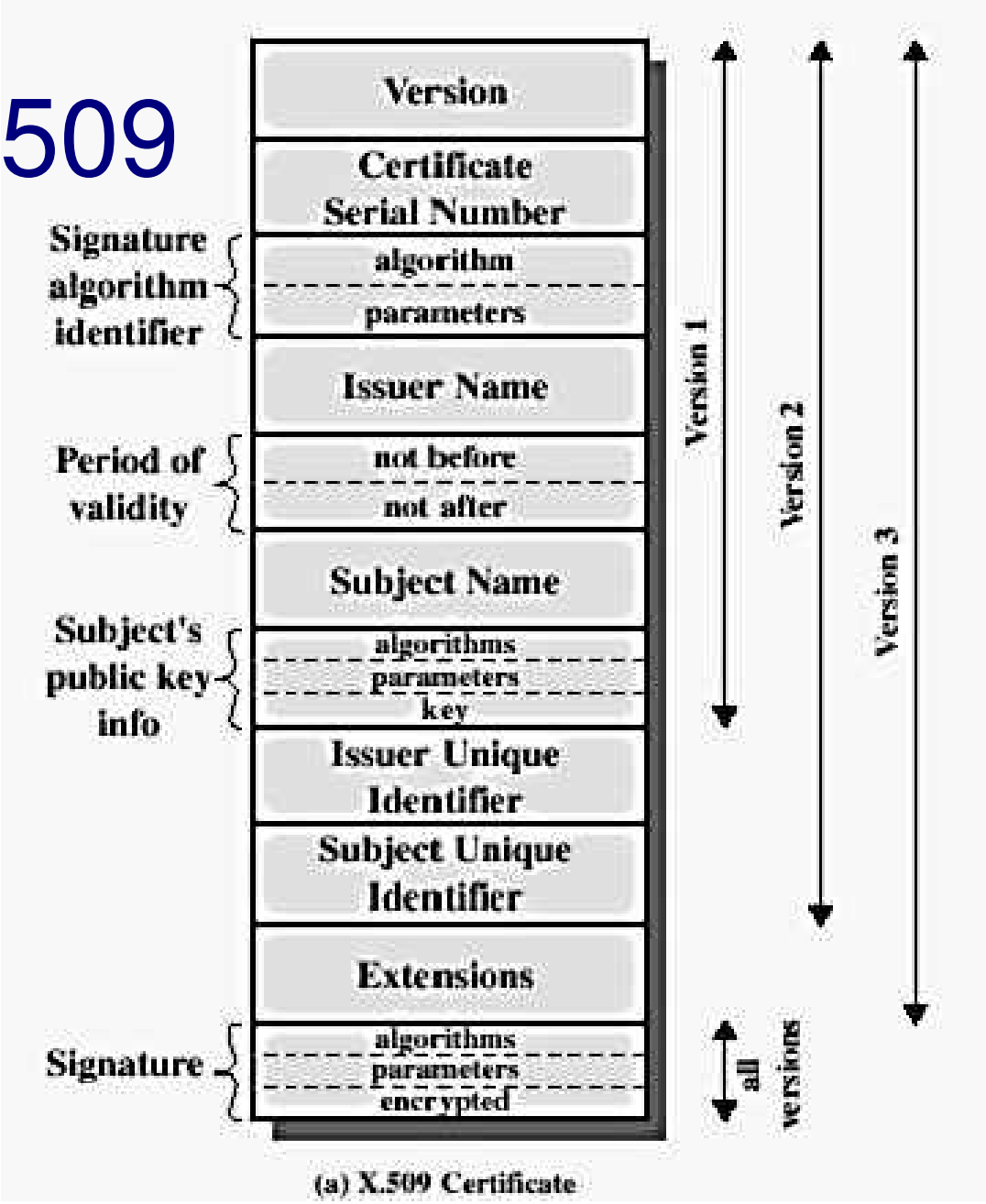
Certificato reale

- Cartaceo
  - *Carta d'identità, etc.*
- Emesso da un'autorità riconosciuta
- **Associa l'identità** di una persona (nome, cognome, data di nascita, …) al suo **aspetto fisico** (foto)

Certificato digitale

- Elettronico
- Associa **l'identità** di una persona ad una **chiave pubblica**
- Emesso da una CA riconosciuta
- Firmato con la chiave privata della CA
- Formato tipico: *X.509*
  - Raccomandato dall'ITU (International Telecommunication Union)

Certificato X.509 - struttura



(a) X.509 Certificate

MD5

MD5 HASHING

1) SUDDIVIDO LA PAROLA IN BLOCCHI DI 512 bit (16 parole da 32 bit) DOPO AVER APPLICATO IL PADDING

1.a) SI ACCODA UN BIT PARI AD 1 ALLA FINE DEL MSG; SI AGGIUNGONO TANTI BIT POSTI A 0 FINO A RAGGIUNGERE UN MULTIPLO DI 512, A CUI SI SOTTRAGGONO 64 BIT, RIEMPITI CON LUNGHEZZA DEL MSG ORIGINALE, MODULO 2^{64}

2) OGNI BLOCCO MD5 LAVORA CON 128 BITS DI STATO, DIVISI IN 4 PAROLE DA 32 BIT (A, B, C, D), INIZIALIZZATE CON DELLE COSTANTI NOTE. I 512 BIT DI OGNI BLOCCO SI USANO PER CAMBIARE LO STATO DEI 128 BIT.

OGNI BLOCCO DA 512 È ANALIZZATO IN 4 PARTI ANALOGHE. OGNI PARTE (ROUND) È DIVISA IN 16 OPERAZIONI UGUALI,

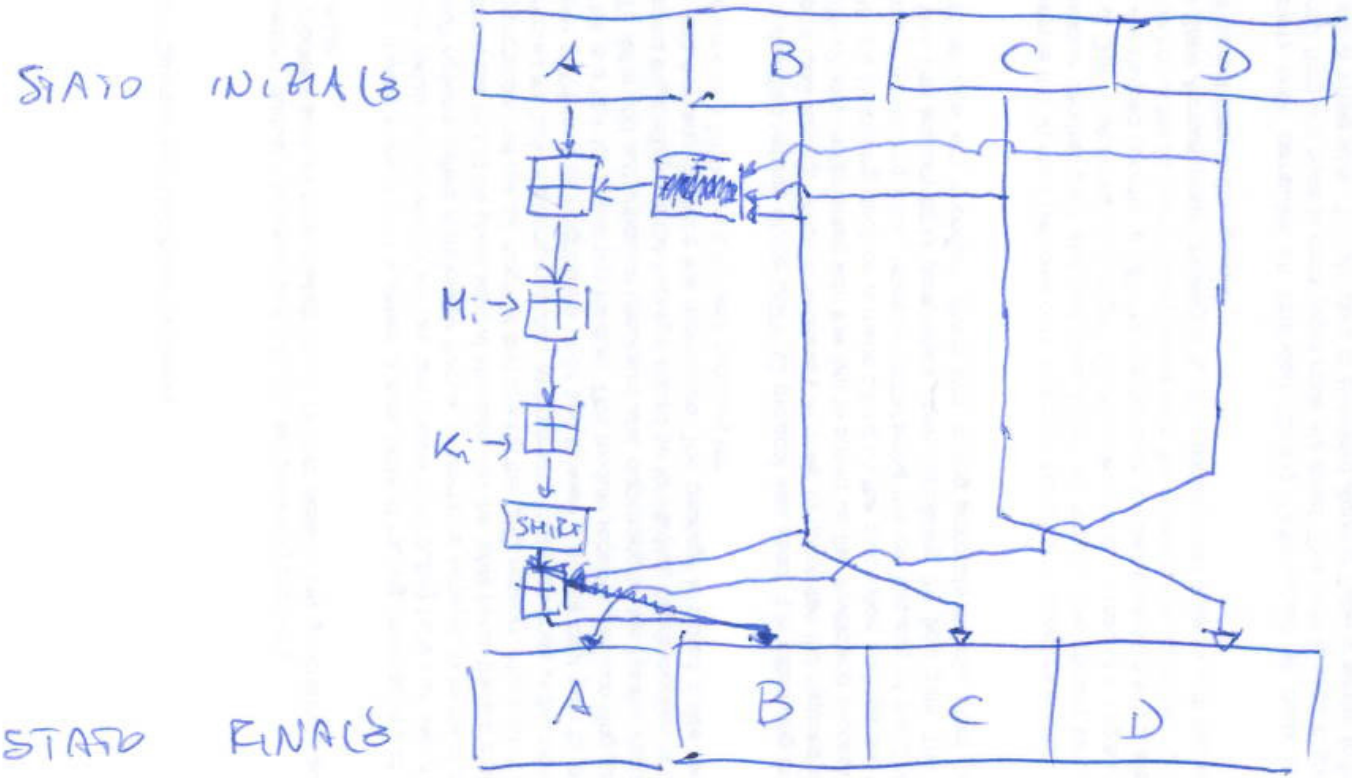
I 512 BIT DI OGNI BLOCCO SI USANO PER MANAN LO STATO DEI 128 BIT.

OGNI BLOCCO DA 512 È ANALIZZATO IN 4 PARTI ANALOGHE.

OGNI PARTE (ROUND) È DIVISA IN 16 OPERAZIONI UGUALI,

BASATE SU: A) UNA FUNZIONE NON LINEARE, B) ADDIZIONE MODULARE E

C) ROTAZIONE SINISTRA

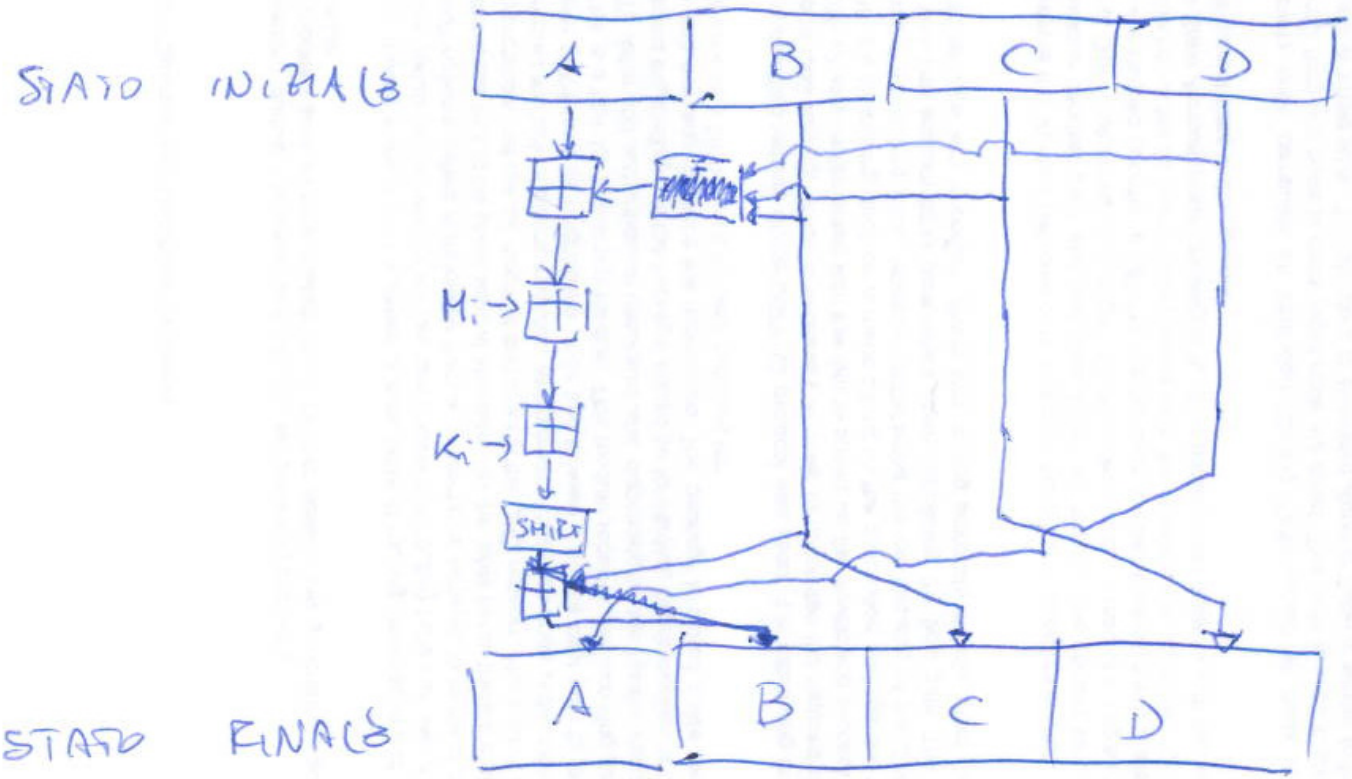


M_i = blocco di 32 bit del msg iniziale

K_i = blocco di 32 bit costante, diverso ∀ oper

SHIFT = spostamento a sinistra di s posizioni con s variabile ∀ op

⊞ = somma modu



M_i = blocco di 32 bit del msg iniziale

K_i = blocco di 32 bit costante, diverso x operaz.

SHIFT = spostamento a sinistra di s posizioni con s variabile x operaz.

$\boxplus$ = somma modulo 2^{32}

3) AD OGNUNO DI 4 ROUND SI ASSOCIANO DIVERSE FUNZIONI

$F1(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$ $\quad F3(B,C,D) = B \oplus C \oplus D$

$\oplus$ XOR
$\wedge$ AND
$\vee$ OR

ESEMPIO MD5 SEMPLIFICATO:

(32) ⟶ (4)

MAPPING 128 ⟶ 16 $\Rightarrow \|A\| = \|B\| = \|C\| = \|D\| = 4$

STATO INIZIALE ≡ A - [illegible] B - [illegible] C - [illegible] D - [illegible]

fattore 4 di suddivisione

512 ⟶ 128 bit ⟶ 64 bit INVECE DI 4 ROUND DA 16

(SOLO 2 ROUND) (16 bit) (SOMMA MODULO 8) IMMAGINIATO DI AVERE 2 ROUND DA 4

F_1 = FROUND 1 = $F_1(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$

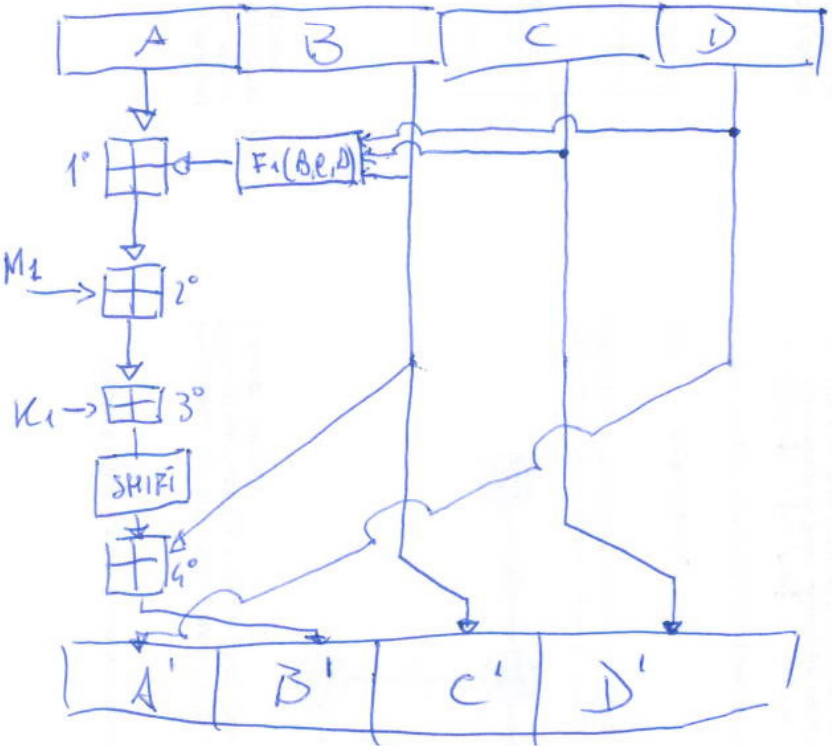
F_2 = FROUND 2 = $F_2(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$

RISULTATO CHIAVE HASH A 16 bit

M: { [1001 [illegible]
[1110
[1011
[1100
[0100
[- -

⊞ SOMMA MODULO 8

ROUND 1)



K_1 = [1010 [illegible]]

K_2 = [0101 [illegible]]

S = { 2, 3 }

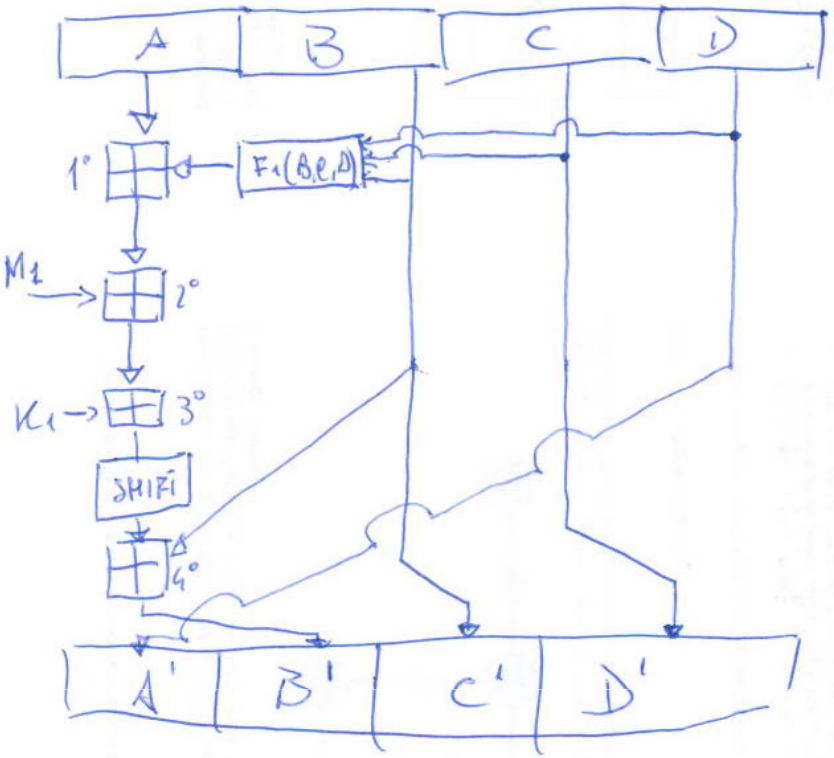
A = 0000

B = 0001

C = 0010

D = 0100

$F_1 = (B \wedge C) \vee (\neg B \wedge D)$



$K_1 = [1010 \ldots]$

$K_2 = [0101 \ldots]$

$S = \{2, 3\}$

$A = 0000$

$B = 0001$

$C = 0010$

$D = 0100$

$F_1 = (B \wedge C) \vee (\neg B \wedge D)$

$B\ 0001 \quad \neg B\ 1110$

$C\ 0010 \quad D\ 0100$

$0000 \quad 0100$

$\Rightarrow F_1(B, C, D) = [0100]$

0100

$1° \Rightarrow \quad 0000 = 0$

$0100 = 4$

$0100 : 4 \ \text{MOD}\ 4 \Rightarrow 0$

$2° \quad 0000 +$

1001

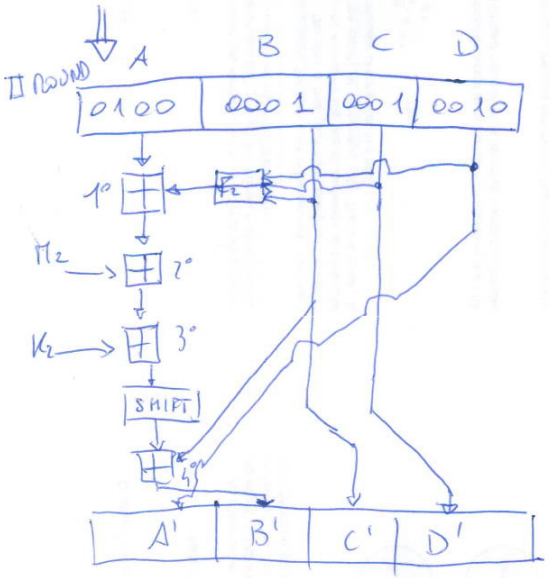
$1001 \ \text{MOD}\ 4 \Rightarrow 1$

0001

3° 0001 + 1010 = 1011 MOD 4 ⇒ 3 0011

SHIFT(2) ⇒ 1100

4° 1100 + 0001 = 1101 MOD 4 ⇒ 0001 ≡ B'



$F_2(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$

B ∧ D: 0001 + 0010 = 0000

C ∧ ¬D: 0001 + 1101 = 0001

0001

$F_2(B,C,D) = 0001$

1° 0100 + 0001 = 0101 MOD 4 ⇒ 0001

2° 0001 + 1110 = 1111 ⇒ 0011

3° 0011 + 0101 = 1000 ⇒ 0000 SHIFT 0000 ⇒ 4° 0000 + 0001 = 0001 ≡ B'

| 0010 0001 0001 0001 | MD5 KEY 16 bit