Crittografia Simmetrica e Antisimmetrica - DES e RSA

Sabrina De Capitani di Vimercati

decapita@ing.unibs.it.

DEA - Università di Brescia

Crittosistemi a Chiave Simmetrica

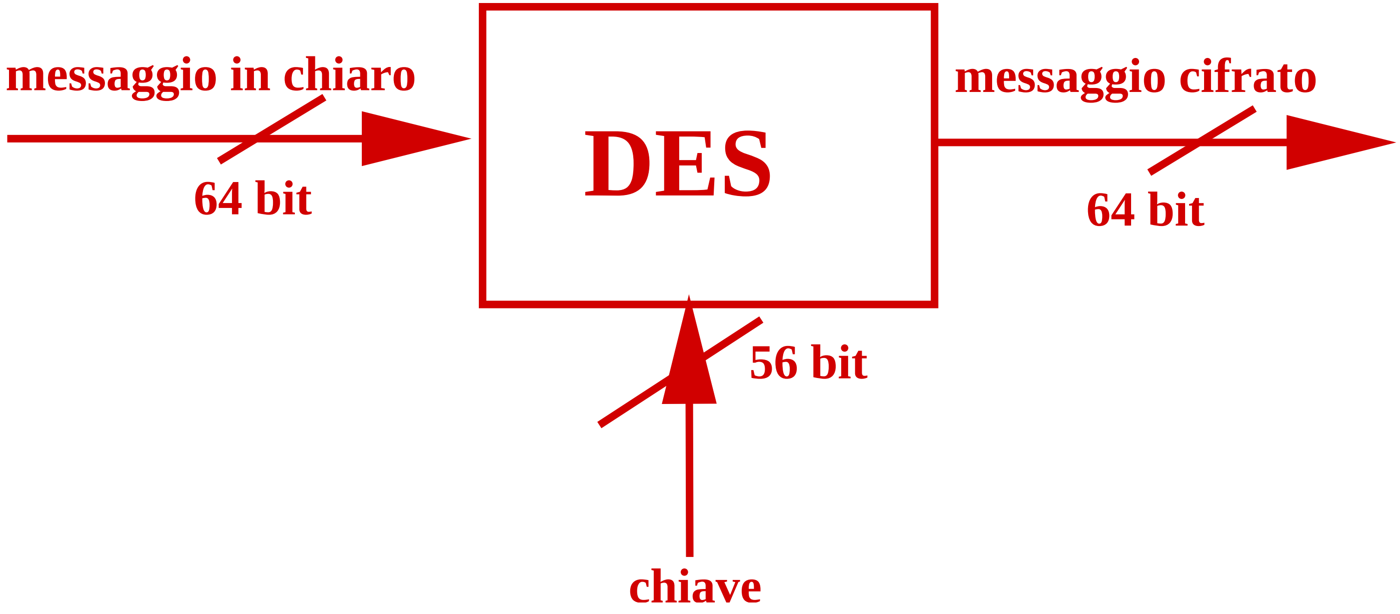
Sono anche chiamati crittosistemi a chiave segreta

- Alice e Bob conoscono la stessa chiave k
- Stream cipher: i messaggi vengono crittati carattere per carattere
- Block cipher: i messaggi sono prima divisi a blocchi e poi crittati

Data Encryption Standard (DES)

- 15 Maggio 1973, richiesta di standard della NBS oggi NIST (1974 seconda richiesta)
- Modifica di Lucifer, sviluppato da IBM (chiave da 128 a 56 bit) reso noto nel 1975
- Standard pubblicato il 15 Gennaio 1977
- Riaffermato per successivi 5 anni nel 1983, 1987, 1992
- DES challenges (giugno 1997, luglio 1998, gennaio 1999)
- Advanced Encryption Standard (AES)

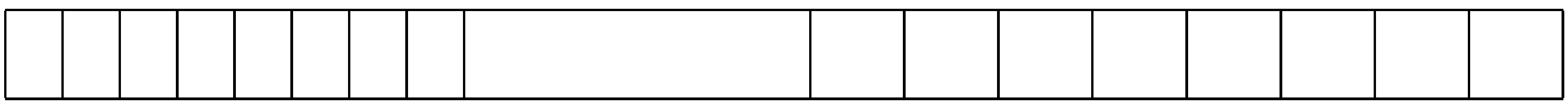
Data Encryption Standard



Lunghezza della Chiave

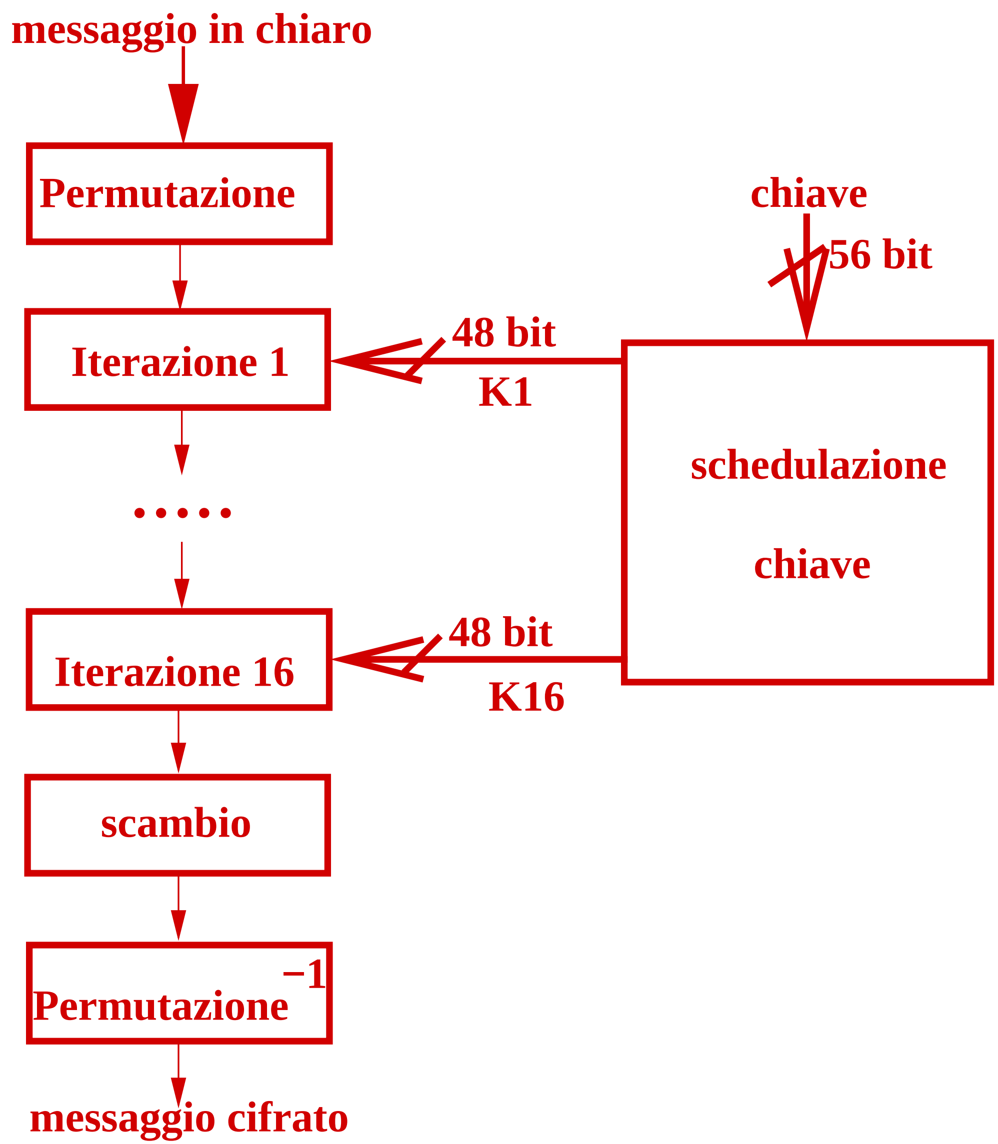
Nello standard DES la chiave è lunga 64 bit; 8 byte di cui l’ottavo bit è di parità

1 2 3 4 5 6 7 8 . . . . . . 57 58 59 60 61 62 63 64



I bit 8, 16, 24,. . ., 64 sono i bit di parità il cui valore coincide con lo xor dei precedenti 7 bit

Struttura del DES



Permutazione

La permutazione iniziale è definita dalla seguente tabella

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Ad esempio, il bit 58 viene portato nella prima posizione, il bit 50 nella seconda e così via

Permutazione Inversa

La permutazione finale è definita dalla seguente tabella

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Ad esempio, il bit 40 viene portato nella prima posizione, il bit 8 nella seconda e così via

Singola Iterazione

La parte centrale del DES consiste nella esecuzione di 16 iterazioni.

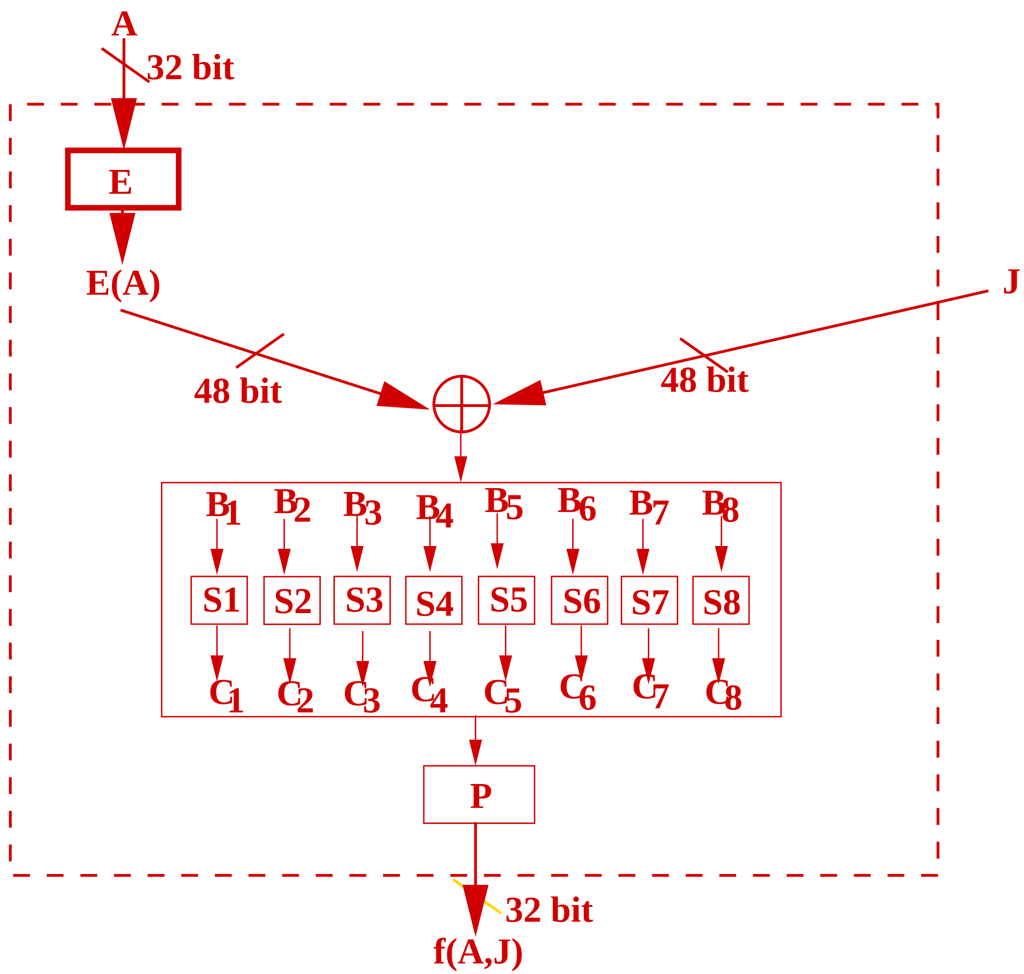
INPUT: blocco di 64 bit: L_{i-1} (parte sinistra di 32 bit) e R_{i-1} (parte destra di 32 bit)

OUTPUT: nuovo blocco di 64 bit: L_i e R_i

METODO:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

La Funzione f



Espansione E

La funzione di espansione E espande 32 bit duplicandone 16

31	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Ad esempio, il bit 32 viene portato nella prima posizione, il bit 1 nella seconda e così via

Esempio di Funzionamento delle S-box

INPUT: 101110; riga=10 (primo ed ultimo bit) colonna=0111 (secondo-quinto bit)

Box S1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	7	8	13	6	2	**11**	15	12	9	7	3	10	5	0
11	15	12	10	2	4	9	1	7	5	11	3	14	10	0	6	13

OUTPUT: 1011 (cifra decimale 11 in binario)

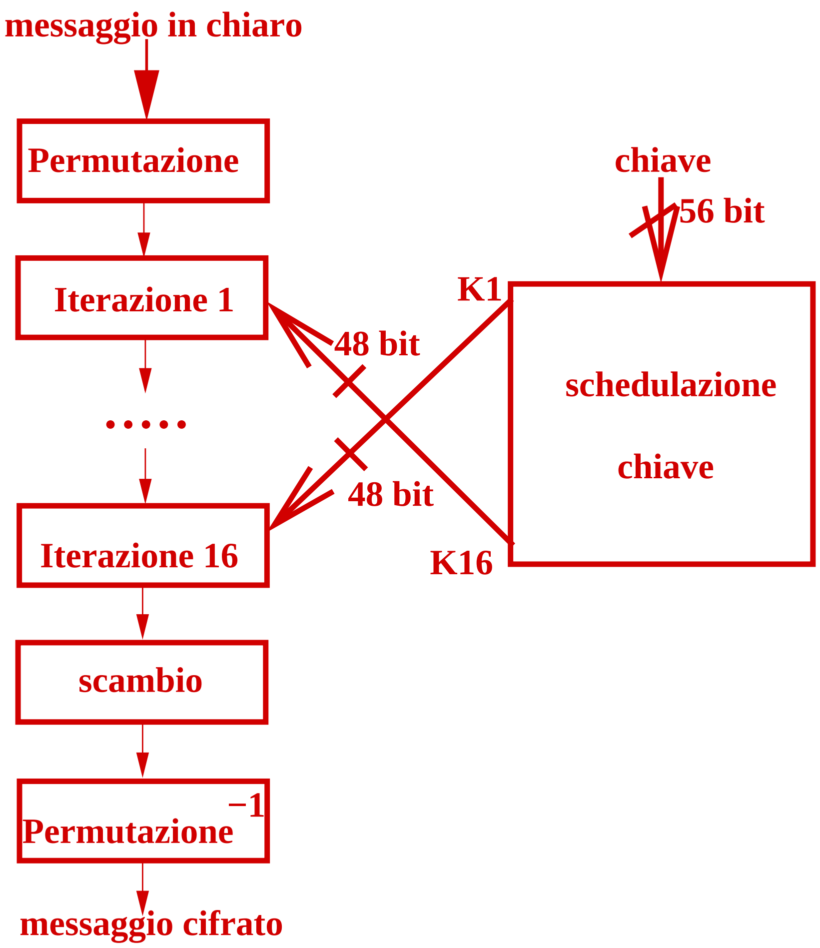
Proprietà delle S-box (1)

- Ogni riga è una permutazione degli interi $0, \ldots, 15$
- Nessuna S-box è una funzione affine o lineare dei suoi input
- Cambiando un solo bit di input ad una S-box variano almeno due bit nell'output
- Per ogni S-box S e per ogni input *x* a 6 bit: S(*x*) e S($x \oplus$001100) differiscono in almeno due bit

Proprietà delle S-box (2)

- Per ogni S-box S, input x e bit b,c: $S(x) \neq S(x \oplus 11bc00)$
- Per ogni S-box, se fissiamo un bit di input ed osserviamo i valori di un fissato bit di output, il numero degli input per i quali il bit di output vale 0 è circa uguale al numero di input per i quali tale bit vale 1

Decifratura DES



Digressione. . .

- La storia narra che IBM propose un'altra tabella per le S-box. . .
- Le S-box vennero modificate dall'NSA al momento della certificazione perché temeva che IBM avesse inserito una trap-door per controllare le comunicazioni
- IBM accettò le modifiche dopo test sulla robustezza (test eseguiti con criteri rimasti segreti!)
- . . . NSA introdusse trap-door?
- Non fu mai accertata nessuna “frode” da parte di NSA ed il DES venne accettato come standard

DES Doppio

Per aumentare lo spazio delle chiave del DES si è pensato di progettare un cifrario a blocchi dove un messaggio è cifrato due volte con due chiavi diverse

Cifratura: $\boxed{c = \mathrm{DES}_{k_2}(\mathrm{DES}_{k_1}(m))}$

Decifratura: $\boxed{m = \mathrm{DES}^{-1}_{k_1}(\mathrm{DES}^{-1}_{k_2}(c))}$

Lunghezza blocco = 64 bit
Chiave (k_1,k_2) lunga 112 bit

Sicurezza DES Doppio

Quanto è sicuro il DES doppio? DES $\equiv$ DES doppio?

Algoritmo Meet in the Middle che permette di forzare il DES doppio con uno sforzo computazionale pari a quello necessario per rompere il DES

$$\boxed{y = \mathrm{DES}_{k_2}(\mathrm{DES}_{k_1}(x))}$$

$$\boxed{A = \mathrm{DES}_{k_1}(x) = \mathrm{DES}^{-1}_{k_2}(y)}$$

Meet in the Middle (1)

Attacco known plaintext: si conosce coppia (x,y) dove x è il msg in chiaro e y il corrispondente testo cifrato
$\Rightarrow$ vogliamo determinare la coppia di chiavi (k_1,k_2)

1. cifriamo x usando tutte le 2^{56} possibili chiavi k_1
2. decifriamo y usando tutte le 2^{56} possibili chiavi k_2
3. Se esiste i,j tale che $\mathrm{DES}_{k_1,i}(x) = \mathrm{DES}^{-1}_{k_2,j}(y)$
   $\Rightarrow$ le due chiavi corrispondenti potrebbero formare la coppia cercata (in media ci sono $\frac{2^{112}}{2^{64}}$ coppie di chiavi che trasformano x in y

Meet in the Middle (2)

4. Per ogni coppia di chiavi per cui si ha che $\text{DES}_{k_1,i}(x) = \text{DES}^{-1}_{k_2,j}(y)$ si verifica se anche:

$$\text{DES}_{k_1,i}(x') = \text{DES}^{-1}_{k_2,j}(y')$$

Se la risposta è affermativa la probabilità che la coppia di chiavi corrispondente sia quella cercata è:

$$\boxed{0.99998474}$$

Triplo DES (1)

Un messaggio viene cifrato 3 volte usando 3 chiavi diverse

Applicando lo stesso attacco visto per il DES doppio si può dimostrare che:

È equivalente ad un cifrario con una chiave di 112 bit e non 168 bit

Triplo DES (2)

Nel 1992 si dimostra che il DES non è un *gruppo*

$$\boxed{\forall k_1, k_2, k_3 : \mathrm{DES}_{k_1}(\mathrm{DES}_{k_2}(m)) \neq \mathrm{DES}_{k_3}(m)}$$

Questo risultato sembra implicare che il triplo DES incrementa la sicurezza del DES

$$c = \mathrm{DES}_{k_3}(\mathrm{DES}_{k_2}^{-1}(\mathrm{DES}_{k_1}(m)))$$

- le tre chiavi sono indipendenti
- k_1 e k_2 sono indipendenti ma $k_1 = k_3$
- le tre chiavi sono identiche

Sicurezza Triplo DES

- Il triplo DES è spesso usato in pratica ed il suo grado di sicurezza è piuttosto elevato
  $\Rightarrow$ meno efficiente del DES singolo di un fattore 3
- Attualmente non sono noti attacchi crittoanalitici pratici al triplo DES

Da chi È Stato Sostituito il DES?

- Dal 1998 il DES non è più certificato come standard federale per le comunicazioni commerciali negli Stati Uniti
- Il triplo DES lo ha sostituito finché il NIST non ha scelto un nuovo cifrario (Advanced Encryption Standard (AES))
- Il primo call for algorithms risale al 12 Settembre 1997
  - deve poter essere reso di dominio pubblico, royalty-free
  - deve essere simmetrico, a blocchi $\geq$ 128 bit
  - le dimensioni delle chiavi devono essere di 128, 192 e 256 bit

RIJNDAEL

Crittosistemi a Chiave Asimmetrica

Sono anche chiamati crittosistemi a chiave pubblica definiti da Diffie-Hellman nel 1976

- I messaggi sono chiusi in uno speciale tipo di cassaforte con due lucchetti
  - con una chiave (pubblica) viene chiusa la cassaforte
  - con un altra chiave (privata) viene aperta la cassaforte

chiave pubblica $\neq$ chiave privata

Proprietà

- Solo il ricevente “originale” può leggere il messaggio
- Solo una chiave deve essere protetta
- Chiunque può crittare un messaggio usando la chiave pubblica
- Non è necessario un canale sicuro per comunicare la chiave privata agli utenti
  - Ogni utente genera la propria coppia di chiavi (public,private) e rende nota la chiave pubblica su un key server

Problemi

- La chiave privata deve essere tenuta privata
- La chiave pubblica deve realmente provenire da Bob
- Le chiavi pubbliche dovrebbero essere recuperate facilmente
- Dovrebbe essere praticamente impossibile determinare la chiave privata dalla corrispondente chiave pubblica
- La crittazione e decrittazione dei messaggi è lenta

Crittosistemi Ibridi

- I crittosistemi a chiave segreta sono più veloci di quelli a chiave pubblica
- Spesso viene usata una combinazione dei due sistemi
  - la crittografia a chiave pubblica è usata per condividere una chiave segreta s
  - i messaggi sono simmetricamente crittati tramite s

Alcune Proposte

Dopo la definizione di Diffie-Hellman, seguirono immediatamente due proposte

- una basata sul problema NP-completo dello zaino introdotta da Merkle ed Hellman
  $\Rightarrow$ è stato forzata ma esistono ancora varianti non violate
- una basata sulla difficoltà di fattorizzare grandi numeri interi (problema in NP $\cap$ co-NP) proposta da Rivest, Shamir e Adleman (RSA)
  $\Rightarrow$ ad oggi è rimasta inviolato
  (vedi “richiami algebra”)

RSA - Generazione delle Chiavi

- Alice genera due numeri primi molto grandi p e q
- Alice calcola $n = p \times q$ e $\Phi(n) = (p-1)(q-1)$
- Alice sceglie un intero $e \in \mathcal{Z}_n^*$ (quindi relativamente primo con $\Phi(n)$)
- Alice calcola l'inverso di e, cioè un intero d t.c. $d \times e \equiv 1 \mod \Phi(n)$ usando l'algoritmo di Euclide Esteso
- Alice pubblica n e e come sua chiave pubblica
- Alice conserva n e d come sua chiave privata

Cifratura e Decifratura

Cifratura: Dato un messaggio m (nota che $m < n$), il corrispondente crittogramma è:

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

Decifratura: Dato un crittogramma c (ovviamente $c < n$), il corrispondente messaggio in chiaro è:

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

Semplice Esempio

Sia p = 47 e q = 71

- $n = p \times q = 3337$ e $\Phi(n) = (p-1)(q-1) = 3220$
- sia e = 79, l'inverso di e è un numero d tale che $d \times 79 \equiv 1 \mod 3220$ $\Rightarrow d = 1019$
- chiave pubblica = (3337,79) e chiave privata = (3337,1019)
- la cifratura di m = 688 è $1570 = 688^{79} \mod 3337$
- la decifratura di c = 1570 è 688 = $1570^{1019} \mod 3337$

Attacchi Possibili ad RSA (1)

- La conoscenza di p e q (fattori di n) permette di "rompere" RSA perché con l'algoritmo esteso di Euclide è possibile calcolare d

  fattorizzare efficientemente $\Rightarrow$ forzare efficientemente RSA

  forzare efficientemente RSA $\stackrel{?}{\Rightarrow}$ fattorizzare efficientemente

  Si congettura che i due problemi siano equivalenti

- Dobbiamo necessariamente conoscere p e q per rompere RSA?

Attacchi Possibili ad RSA (2)

- Per calcolare c si può calcolare la sua radice e-esima in $\mathcal{Z}_n$
  $\Rightarrow$ problema difficile tanto quanto la fattorizzazione nel caso di n composto
- d può essere calcolato anche conoscendo $\Phi(n)$, applicando l'algoritmo di Euclide esteso su $\Phi(n)$ ed e
  $\Rightarrow$ conoscere $\Phi(n)$ vuol dire conoscere p e q

  $\Phi(n) = n - (p + q) + 1 \Rightarrow x_1 = (p + q)$
  $(p - q)^2 = (p + q)^2 - 4n \Rightarrow x_2 = (p - q)$

  Da x_1 e x_2 si ricava p e q $\Rightarrow$ fattorizzazione di n

Richiami Algebra Modulare (1)

- $\mathcal{Z}_n$ denota l'insieme degli interi minori di n
- $\mathcal{Z}_n^*$ denota l'insieme degli interi minori di n e primi con n (p.es., $\mathcal{Z}_p^* = \{1, \ldots, p-1\}$) se p primo)
- $a \equiv b \mod n$ se $\exists k : a = b + kn$
- La funzione di eulero $\Phi(n)$ indica il numeri di interi minori di n e relativamente primi con esso

$$\Phi(n) = \begin{cases} n-1 & \text{se } n \text{ è primo;} \\ n \times \Pi_{j=1}^{k}(1 - \frac{1}{p_j}) & \text{altrimenti.} \end{cases}$$

dove i p_j sono i divisori primi di n

Richiami Algebra Modulare (2)

- $\mid \mathcal{Z}_n^* \mid = \Phi(n)$
- Teorema Fermat-Eulero: Dati due numeri interi n e m primi tra loro ($m \in \mathcal{Z}_n^*$): $m^{\Phi(n)} \mod n = 1$

  L'inverso di m è: $m^{\Phi(n)-1} \mod n = 1$
- Dati due numeri interi m e p (p è primo) con $m \in \mathcal{Z}_n$: $m^{p-1} \mod n = 1$