Esercizi Sicurezza (DES-RSA)

Esercitazione QoS e Sicurezza nelle reti

Ing. Amilcare - Francesco Santamaria

[Anno]

Sommario

[2 Esercizio - 1 2](#_Toc452974728)

[3 Applicazione DES 6](#_Toc452974729)

# Esercizio - 1

Date la seguente configurazione iniziale p = 11 e q = 13 trovare l’esponente della chiave pubblica, e l’esponente della chiave private al fine di trasferire il seguente messaggio M = 123; In particolare selezionare l’esponente pubblico tra il seguente set <5,3,7>.

Il primo STEP consiste nel trovare il fattore n e f(n)

Sappiamo quindi che il valore di e deve rispettare il seguente vincolo

Inoltre esso deve essere un numero primo non divisore di f(n); fattorizziamo quindi f(n)

Dal set iniziale l’unica soluzione possibile è selezionare e = 7

Il prossimo step è calcolare

Il valore d è l’esponente della chiave privata, di conseguenza bisogna utilizzare l’algoritmo Euclideo Esteso. Vediamo nel dettaglio come esse funzioni.

Si parte realizzando la seguente equazione

Ponendo “a=e” e “b=f(n)” si possono eseguire i passi seguenti

Il primo set di operazioni rappresenta l’inizializzazione dell’algoritmo, in particolare

Step (I)

|  |  |
| --- | --- |
| a | e |
| b | f(n) |
| x | 0 |
| y | 1 |
| u | 1 |
| v | 0 |

Step(II)

A questo punto possiamo passare al loop che terminerà solo quando a=0.

Step (II sub 1)

|  |  |
| --- | --- |
| q |  |
| r |  |
| m |  |
| n |  |

Prepariamo i valori per la prossima iterazione

Step (II sub 2)

|  |  |
| --- | --- |
| b | a |
| a | r |
| x | u |
| y | v |
| u | m |
| v | n |

Quando il valore a = 0 la condizione di uscita si verifica e si può passare allo step III; altrimenti si ripete lo step II, ritornando i valori b,x,y

Step III

Se b != 1 abbiamo un errore : Non è possibile ricavare l’esponente con la configurazione e,f(n) scelta

Altrimenti

Ritornando all’esercizio andiamo ad effettuare i calcoli:

g,x,y = egcd(a,b) = egcd(7,120)

Step I

|  |  |
| --- | --- |
| a | 7 |
| b | 120 |
| x | 0 |
| y | 1 |
| u | 1 |
| v | 0 |

Step (II sub 1)

|  |  |
| --- | --- |
| q | = |
| r |  |
| m |  |
| n |  |

Step (II sub 2)

|  |  |  |
| --- | --- | --- |
| b | a | 7 |
| a | r | 1 |
| x | u | 1 |
| y | v | 0 |
| u | m | -17 |
| v | n | 1 |

Step (II sub 1)

|  |  |
| --- | --- |
| q | = |
| r |  |
| m |  |
| n |  |

Step (II sub 2)

|  |  |  |
| --- | --- | --- |
| b | a | 1 |
| a | r | 0 |
| x | u | -17 |
| y | v | 1 |
| u | m | 120 |
| v | n | -7 |

g = 1, x = -17, y = 1

Il valore di d sarà semplicemente ricavato effettuando l’operazioe modulo

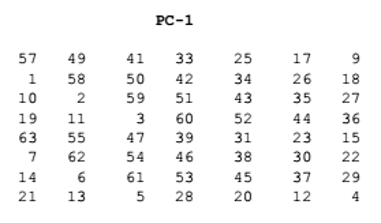
# Applicazione DES

Dato il seguente Messaggio **K = 0x89182E4260A62464** Calcolare i valori di C0 e D0 e C8

Per Ottenere tali valori dobbiamo applicare la permutazione **PC-1** alla chiave. Il primo passo è portare la chiave in binario

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | 1-8 | 9-16 | 17-24 | 25-32 | 33-40 | 41-48 | 49-56 | 57-64 |
|  | 1000 1001 | 0001 1000 | 0010 1110 | 0100 0010 | 0110 0000 | 1010 0110 | 0010 0100 | 0110 0100 |

Applichiamo la permutazione PC-1



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K(Permutata) | 1-7 | 8-14 | 15-21 | 22-28 | 29-35 | 36-42 | 43-49 | 50-56 |
|  | 0010000 | 1100110 | 0011110 | 1000000 | 0010110 | 0111001 | 0000000 | 1110010 |

Dalla chiave permutata possiamo ricavare le sotto-chiavi C0 e D0

**C0 = 0010000 1100110 0011110 1000000** (28 bit)

**D0 = 0010110 0111001 0000000 1110010** (28 bit)

Calcolare C8 vuol dire calcolare la sottochiave all’ 8-th iterazione considerando la seguente tabella delle iterazioni.

 Utilizzando tale tabella per arrivare all’ottava iterazione dobbiamo effettuare 14 shift iterativi a sinistra, questo comporta spostare i bit di 14 posizioni a sinistra. Sapendo che il primo bit spostato di una posizione a sinistra indica bit in posizione 28 => otteniamo la seguente sotto-chiave C8.

**C8 = 00011110 000000 0010000110011**