

Crittossistema a chiave pubblica

chiave privata
kpriv

file pubblico	
utente	chiave pubblica
A	kpub
...	...

A nnarella

Crittografia a Chiave Pubblica 0

Cifratura

file pubblico	
utente	chiave pubblica
A	kpub
...	...

Devo cifrare il messaggio M ed inviarlo ad **A**

B giagio

Crittografia a Chiave Pubblica 1

Cifratura

file pubblico	
utente	chiave pubblica
A	kpub
...	...

C

Cifratura di M per **A**
 $C \leftarrow \text{CIFRA}(k_{\text{pub}}, M)$

B giagio

Crittografia a Chiave Pubblica 2

Decifratura

file pubblico	
utente	chiave pubblica
A	kpub
...	...

Devo decifrare il messaggio cifrato C

A nnarella

C

Crittografia a Chiave Pubblica 3

Decifratura

chiave privata
kpriv

file pubblico	
utente	chiave pubblica
A	kpub
...	...

Decifratura di C
 $M \leftarrow \text{DECIFRA}(k_{\text{priv}}, C)$

A nnarella

C

Crittografia a Chiave Pubblica 4

Crittografia a chiave pubblica ed a chiave privata

- ❑ Vantaggi della crittografia a chiave pubblica
 - Chiavi private mai trasmesse
 - Possibile la firma digitale
- ❑ Vantaggi della crittografia a chiave privata
 - Molto più veloce (ad es., DES è ≥ 100 volte più veloce di RSA, in hardware tra 1.000 e 10.000 volte)
 - Sufficiente in diverse situazioni (ad esempio, applicazioni per singolo utente)

Crittografia a Chiave Pubblica 5

Digital Envelope

file pubblico

utente	chiave pubblica
A	kpub
...	...

Cifratura di M per A
 $k \leftarrow$ genera chiave sessione
 $C_1 \leftarrow$ CIFRA (kpub, k)
 $C_2 \leftarrow$ E (k, M)

Crittografia a Chiave Pubblica 6

Digital Envelope: Decifratura

file pubblico

utente	chiave pubblica
A	kpub
...	...

chiave privata
kpriv

Decifratura di C_1, C_2
 $k \leftarrow$ DECIFRA (kpriv, C_1)
 $M \leftarrow$ D (k, C_2)

Crittografia a Chiave Pubblica 7

Digital Envelope: vantaggi

- ❑ Chiave di sessione solo per uno o pochi messaggi
- ❑ Molto più veloce della sola crittografia a chiave pubblica

Crittografia a Chiave Pubblica 8

Teoria dei Numeri

Crittosistemi a chiave pubblica più comuni basati sulla Teoria dei Numeri

"... both Gauss and lesser mathematicians may be justified in rejoicing that there is one science at any rate, and that their own, whose very remoteness from ordinary human activities keep it gentle and clean."

G. H. Hardy, *A Mathematician's Apology*, 1940

Crittografia a Chiave Pubblica 9

RSA

Proposto nel 1978 da

Rivest Shamir Adleman

Sicurezza basata sulla difficoltà di fattorizzare

Crittografia a Chiave Pubblica 10

Chiavi RSA

file pubblico

utente	chiave pubblica
A	(n,e)
...	...

chiave privata
(n,d)

$n = pq$
 p, q primi

$ed = 1 \pmod{(p-1)(q-1)}$


Crittografia a Chiave Pubblica 11

Cifratura RSA

file pubblico

utente	chiave pubblica
A	(n,e)
...	...

Devo cifrare il messaggio M ed inviarlo ad A



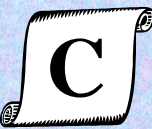
giugio

Crittografia a Chiave Pubblica 12

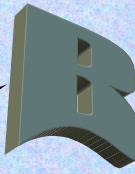
Cifratura RSA

file pubblico

utente	chiave pubblica
A	(n,e)
...	...



Cifratura di M per A
 $C \leftarrow M^e \pmod n$



giugio


Crittografia a Chiave Pubblica 13

Decifratura RSA


file pubblico

utente	chiave pubblica
A	(n,e)
...	...


Devo decifrare il messaggio cifrato C



nnarella



???



Crittografia a Chiave Pubblica 14


Decifratura RSA

file pubblico

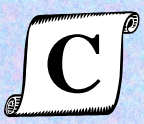
utente	chiave pubblica
A	(n,e)
...	...

chiave privata (n,d)

Decifratura di C
 $M \leftarrow C^d \pmod n$



nnarella



Crittografia a Chiave Pubblica 15

"Piccolo" esempio: Chiavi RSA


chiave privata (n=3337, d=1019)

file pubblico

utente	chiave pubblica
A	(n = 3337, e = 79)
...	...

$3337 = 47 \cdot 71$
 $p = 47, q = 71$

$ed = 79 \cdot 1019 = 1 \pmod{3220}$
 $(p-1)(q-1) = 46 \cdot 70 = 3220$



nnarella

Crittografia a Chiave Pubblica 16


Esempio: Cifratura RSA

file pubblico

utente	chiave pubblica
A	(n = 3337, e = 79)
...	...

1570

Cifratura di M = 688 per A
 $1570 \leftarrow 688^{79} \pmod{3337}$



giugio

Crittografia a Chiave Pubblica 17

Esempio: Decifratura RSA

chiave privata (n=3337, d=1019)

file pubblico	
utente	chiave pubblica
A	(n = 3337, e = 79)
...	...

Decifratura di C = 1570
 $688 \leftarrow 1570^{1019} \text{ mod } 3337$

1570

nnarella
 Crittografia a Chiave Pubblica 18

Esercizio

Svolgere "piccolo" esempio RSA

- Calcolo p, q
- Calcolo n
- Calcolo e, d
- Calcolo cifratura
- Calcolo decifratura

Crittografia a Chiave Pubblica 19

Correttezza decifratura RSA

$$\begin{aligned}
 C^d \text{ mod } n &= (M^e)^d \text{ mod } n \\
 &= M^{de} \text{ mod } n \quad \text{ed = 1 mod (p-1)(q-1)} \\
 &= M^{1+r(p-1)(q-1)} \text{ mod } n \\
 &= M \text{ mod } n \\
 &= M \quad \text{Teorema di Eulero} \\
 &\quad x \in Z_n^* \Rightarrow x^{(p-1)(q-1)} = 1 \text{ mod } n
 \end{aligned}$$

poichè $0 \leq M < n$

Prova per tutti gli x mediante il teorema del resto cinese

Crittografia a Chiave Pubblica 20

Funzione di Eulero

- $\phi(n)$ = cardinalità di $Z_n^* = \{x \mid 0 < x < n \text{ tali che } \text{gcd}(x, n) = 1\}$
- $\phi(p) = p-1$ se p primo
- $\phi(pq) = (p-1)(q-1)$ se p, q primi
- $\phi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$
 fattorizzazione $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, p_i primo, $e_i \geq 0$
- Teorema di Eulero: $x \in Z_n^* \Rightarrow x^{\phi(n)} = 1 \text{ mod } n$

Crittografia a Chiave Pubblica 21

Efficienza delle computazioni

Come effettuare le computazioni?

- Elevazione a potenza modulare
- Generazione numeri primi
- Generazione e, d $\left\{ \begin{array}{l} \text{generazione di e} \\ d \leftarrow e^{-1} \text{ mod } (p-1)(q-1) \end{array} \right.$

Crittografia a Chiave Pubblica 22

Elevazione a potenza modulare Metodo naive

Calcolo di $x^y \text{ mod } z$

```

Potenza_Modulare_naive(x, y, z)
a ← 1
for i = 1 to y do
    a ← (a · x) mod z
return a
    
```

Crittografia a Chiave Pubblica 23


Elevazione a potenza modulare Metodo naive

Calcolo di $x^y \bmod z$

```

Potenza_Modulare_naive (x, y, z)
a ← 1
for i = 1 to y do
    a ← (a · x) mod z
return a
    
```

Se y è di 512 bit, occorrono $\approx 2^{512}$ operazioni



Crittografia a Chiave Pubblica 24

Elevazione a potenza modulare Metodo left-to-right

Calcolo di $x^y \bmod z$ $y = y_0 2^0 + y_1 2^1 + \dots + y_t 2^t$

Idea: $y = y_0 + 2(y_1 + \dots + 2(y_{t-1} + 2y_t))$
 $x^y = x^{y_0} (\dots (x^{y_{t-1}} (x^{y_t})^2 \dots)^2)^2$

Esempio: x^{40}

$$40 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$$

$$40 = 0 + (2(0 + 2(0 + 2(1 + 2(0 + 2 \cdot 1))))))$$

Crittografia a Chiave Pubblica 25

Elevazione a potenza modulare Metodo left-to-right

Calcolo di $x^y \bmod z$ $y = y_0 2^0 + y_1 2^1 + \dots + y_t 2^t$

Idea: $y = y_0 + 2(y_1 + \dots + 2(y_{t-1} + 2y_t))$
 $x^y = x^{y_0} (\dots (x^{y_{t-1}} (x^{y_t})^2 \dots)^2)^2$

Esempio: x^{40}

$$40 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$$

$$40 = 0 + (2(0 + 2(0 + 2(1 + 2(0 + 2 \cdot 1))))))$$

$$x^{40} = x^0 (x^0 (x^0 (x^1 (x^0 (x^1)^2)^2)^2)^2)^2$$

Crittografia a Chiave Pubblica 26

Elevazione a potenza modulare Metodo left-to-right

Calcolo di $x^y \bmod z$ $y = y_0 2^0 + y_1 2^1 + \dots + y_t 2^t$

Idea: $y = y_0 + 2(y_1 + \dots + 2(y_{t-1} + 2y_t))$
 $x^y = x^{y_0} (\dots (x^{y_{t-1}} (x^{y_t})^2 \dots)^2)^2$

Potenza_Modulare (x, y, z) left-to-right

```

a ← 1
for i = t downto 0 do
    a ← (a · a) mod z
    if y_i = 1 then a ← (a · x) mod z
return a
    
```

Crittografia a Chiave Pubblica 27

Elevazione a potenza modulare Metodo right-to-left

Calcolo di $x^y \bmod z$ $y = y_0 2^0 + y_1 2^1 + \dots + y_t 2^t$

Idea: $x^y = (x^{2^0})^{y_0} \cdot (x^{2^1})^{y_1} \cdot \dots \cdot (x^{2^t})^{y_t}$

Esempio: x^{40}

$$40 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$$

$$x^{40} = (x^1)^0 \cdot (x^2)^0 \cdot (x^4)^0 \cdot (x^8)^1 \cdot (x^{16})^0 \cdot (x^{32})^1$$

Crittografia a Chiave Pubblica 28

Elevazione a potenza modulare Metodo right-to-left

Calcolo di $x^y \bmod z$ $y = y_0 2^0 + y_1 2^1 + \dots + y_t 2^t$

Idea: $x^y = (x^{2^0})^{y_0} \cdot (x^{2^1})^{y_1} \cdot \dots \cdot (x^{2^t})^{y_t}$

Esempio: x^{40}

$$40 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$$

$$x^{40} = (x^1)^0 \cdot (x^2)^0 \cdot (x^4)^0 \cdot (x^8)^1 \cdot (x^{16})^0 \cdot (x^{32})^1$$

x^1 x^2 x^4 x^8 x^{16} x^{32}

Crittografia a Chiave Pubblica 29

Elevazione a potenza modulare Metodo right-to-left

Calcolo di $x^y \bmod z$ $y = y_0 2^0 + y_1 2^1 + \dots + y_t 2^t$

Idea: $x^y = (x^{2^0})^{y_0} \cdot (x^{2^1})^{y_1} \cdot \dots \cdot (x^{2^t})^{y_t}$

Esempio: x^{40}

$$40 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$$

$$x^{40} = (x^1)^0 \cdot (x^2)^0 \cdot (x^4)^0 \cdot (x^8)^1 \cdot (x^{16})^0 \cdot (x^{32})^1$$

$$x^{40} = \underbrace{x^1}_{=1} \cdot \underbrace{x^2}_{=1} \cdot \underbrace{x^4}_{=1} \cdot x^8 \cdot \underbrace{x^{16}}_{=1} \cdot x^{32}$$

Crittografia a Chiave Pubblica 30

Elevazione a potenza modulare Metodo right-to-left

Potenza_Modulare (x, y, z)

```

if y = 0 then return 1
X ← x; P ← 1
if y_0 = 1 then X ← x
for i = 1 to t do
    X ← X · X mod z
    if y_i = 1 then P ← P · X mod z
return P
    
```

5⁵⁹⁶ mod 1234

i	0	1	2	3	4	5	6	7	8	9
y _i	0	0	1	0	1	0	1	0	0	1
X	5	25	625	681	1011	369	421	779	947	925
P	1	1	625	625	67	67	1059	1059	1059	1013

Crittografia a Chiave Pubblica 31

Decifratura RSA

- ❑ Pentium 90MHz, toolkit BSAFE 3.0

modulo	512 bit	1024 bit
throughput kbit/s	21,6	7,4

- ❑ RSA hardware [1993]

modulo	512 bit	970 bit
throughput kbit/s	300	185

Crittografia a Chiave Pubblica 32

Algoritmo di Euclide

Descritto negli *Elementi* di Euclide (circa 300 A. C.)

Euclide (a,b)

```

if b = 0 then return a
else return Euclide (b, a mod b)
    
```

Teorema della ricorsione del gcd
Per tutti gli interi $a \geq 0$ e $b > 0$
 $\text{gcd}(a,b) = \text{gcd}(b, a \bmod b)$

Crittografia a Chiave Pubblica 33

Algoritmo di Euclide: Esempi

Euclide (30,21) = Euclide (21,9)
= Euclide (9,3)
= Euclide (3,0) = 3

Euclide (4864,3458) = Euclide (3458,1406)
= Euclide (1406,646)
= Euclide (646,114)
= Euclide (114,76)
= Euclide (76,38)
= Euclide (38,0) = 38

Crittografia a Chiave Pubblica 34

Algoritmo di Euclide: complessità

- ❑ Assumiamo $a \geq b$
- ❑ Al massimo $\log b$ chiamate
- ❑ Per ogni chiamata $O((\log a)^2)$ operazioni su bit
- ❑ Totale: al massimo $O((\log a)^3)$ operazioni su bit
- ❑ **Euclide** (a,b) richiede al massimo $O((\log a)^2)$ operazioni su bit

Crittografia a Chiave Pubblica 35

Algoritmo di Euclide Esteso

```

Euclide-esteso (a,b)
if b = 0 then return (a, 1, 0)
(d', x', y') ← Euclide-esteso (b, a mod b)
(d, x, y) ← (d', y', x' - ⌊a/b⌋y')
return (d, x, y)
    
```

- ❑ Computa interi d, x, y tali che $d = \gcd(a,b) = ax+by$
- ❑ Stesso running time asintotico di **Euclide** (a,b)

Crittografia a Chiave Pubblica 36

Algoritmo di Euclide (iterativo)

```

Euclide_iterativo (a,b)
while b ≠ 0 do
    r ← a mod b; a ← b; b ← r;
return a
    
```

Esercizio: versione iterativa di **Euclide-esteso** (a,b)

Crittografia a Chiave Pubblica 37

Soluzione di $ax \equiv b \pmod{n}$

- ❑ Ha soluzioni se e solo se $g|b$ $g = \gcd(a,n)$
- ❑ Se $g|b$ ci sono esattamente g distinte soluzioni mod n:

$$x' \frac{b}{g} + i \frac{n}{g} \quad \text{per } i = 0, 1, \dots, g-1$$

dove $g = ax' + ny$ (da **Euclide-esteso** (a,n))

Crittografia a Chiave Pubblica 38

Soluzione di $ax \equiv 1 \pmod{n}$

- ❑ Ha soluzioni se e solo se $\gcd(a,n) = 1$
- ❑ Se $\gcd(a,n) = 1$ l'unica soluzione mod n è: x'

dove $1 = ax' + ny$ (da **Euclide-esteso** (a,n))

- ❑ Tale soluzione viene denotata con $a^{-1} \pmod{n}$

Crittografia a Chiave Pubblica 39

Generazione chiavi

1. Input L (lunghezza modulo)
2. Genera 2 primi di lunghezza L/2
3. $n \leftarrow p \cdot q$
4. Scegli a caso e
5. **If** $\gcd(e, (p-1)(q-1)) = 1$
then $d \leftarrow e^{-1} \pmod{(p-1)(q-1)}$
else goto 4.

Crittografia a Chiave Pubblica 40

Generazione chiavi (comunemente usata in pratica)

1. Input L (lunghezza modulo)
2. $e \leftarrow 3$ oppure $e \leftarrow 2^{16}+1 (= 65.537)$ solo due 1 in binario!
3. Genera 2 primi di lunghezza L/2
4. $n \leftarrow p \cdot q$
5. **If** $\gcd(e, (p-1)(q-1)) = 1$
then $d \leftarrow e^{-1} \pmod{(p-1)(q-1)}$
else goto 3.

Crittografia a Chiave Pubblica 41

Generazione di un primo 'grande'

1. Genera a caso un dispari p di grandezza appropriata
2. Testa se p è primo
3. Se p è composto, go to 1.

Crittografia a Chiave Pubblica 42

Generazione di un primo 'grande'

1. Genera a caso un dispari p di grandezza appropriata
2. Testa se p è primo
3. Se p è composto, go to 1.

Variante con sequenza di ricerca

1. Genera a caso un dispari p di grandezza appropriata
2. Testa se p è primo
3. Se p è composto, $p \leftarrow p+2$, go to 2.

Crittografia a Chiave Pubblica 43

Distribuzione dei numeri primi

- $\pi(x)$ = numero di primi in $[2,x]$
- Teorema dei numeri primi: $\lim_{x \rightarrow \infty} \frac{\pi(x)}{x/\ln x} = 1$
- Esempio: $\pi(10^{10}) = 455.052.511$
 $10^{10}/\ln 10^{10} \approx 434.294.481,9$ (4% in meno)
- Per $x \geq 17$

$$\frac{x}{\ln x} < \pi(x) < 1.25506 \frac{x}{\ln x}$$

Crittografia a Chiave Pubblica 44

Scelta di un primo di 512 bit


- Scelto un intero in $[2,2^{512}]$ la probabilità che sia primo è circa 1 su $\ln 2^{512}$ ($\ln 2^{512} \approx 354.89$)
- Numero medio di tentativi ≈ 354.89
- Se si scelgono solo dispari dimezza ≈ 177.44
- Se si scelgono dispari in $[2^{511}, 2^{512}]$ probabilità è

$$\approx \left(\frac{2^{512}}{\ln 2^{512}} - \frac{2^{511}}{\ln 2^{511}} \right) \frac{1}{2^{511}/2} \approx \frac{1}{177.79}$$

1 ... 1
510 bit scelti a caso

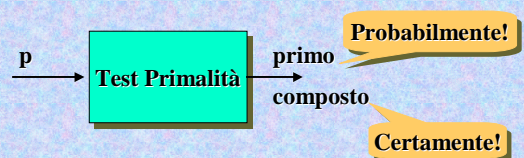
Crittografia a Chiave Pubblica 45

Test di primalità

- Il miglior algoritmo deterministico è una variante di Cohen e H. Lenstra del test di Adleman, Pomerance e Rumely [1983]
- Complessità $(\lg p)^{O(\lg \lg p)}$ 


Crittografia a Chiave Pubblica 46

Test di primalità probabilistici



Insieme di **witness** $W(p)$

- dato $a \in [0, p-1]$ è facile verificare se $a \in W(p)$
- se p è primo allora $W(p)$ è vuoto
- se p è composto allora $|W(p)| \geq p/2$

Scelgo a 

Crittografia a Chiave Pubblica 47


Diminuizione della probabilità di errore

- Probabilità di errore (n è composto ma viene dichiarato primo) di tale test $\leq 1/2$
- Se il test viene ripetuto indipendentemente t volte allora la probabilità di errore $\leq (1/2)^t$

Crittografia a Chiave Pubblica 48

Test di primalità probabilistici

- Test di Solovay-Strassen
 - Pubblicato nel 1977
 - Probabilità di errore $\leq (1/2)^t$
- Test di Miller-Rabin
 - Il più usato in pratica
 - Il più veloce
 - Probabilità di errore $\leq (1/4)^t$



Crittografia a Chiave Pubblica 49

Test di Solovay-Strassen

- $(a|n)$ simbolo di Jacobi
- **Criterio di Eulero:** Sia n un primo dispari.
 $\gcd(a,n) = 1 \Rightarrow a^{(n-1)/2} = (a|n) \pmod n$
- Sia n un numero composto dispari.
 $|\{a : \gcd(a,n) = 1 \text{ and } a^{(n-1)/2} = (a|n) \pmod n\}| \leq \phi(n)/2$
- L'insieme di *witness di Eulero*
 $W_E(n) = \{a : \gcd(a,n) > 1 \text{ oppure } a^{(n-1)/2} \neq (a|n) \pmod n\}$

Crittografia a Chiave Pubblica 50

Simbolo di Jacobi

- $(a|n) = (a|p_1)^{e_1} (a|p_2)^{e_2} \dots (a|p_k)^{e_k}$
- $(a|p) = \begin{cases} 0 & \text{se } p \text{ divide } a \\ 1 & \text{se } a \text{ è un quadrato mod } p \\ -1 & \text{se } a \text{ non è un quadrato mod } p \end{cases}$
- **Simbolo di Legendre**
 $(a|p) = a^{(p-1)/2} \pmod p$
 p primo
- Fattorizzazione dispari $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, p_i primo, $e_i \geq 0$
- Può essere computato con $O((\log n)^2)$ operazioni su bit
 - $(a|n) = 0 \Leftrightarrow \gcd(a,n) > 1$
 - $a = b \pmod n \Rightarrow (a|n) = (b|n)$
 - $(ab|n) = (a|n) (b|n)$
 - $(2|n) = (-1)^{(n^2-1)/8}$
 - $(m|n) = (n|m) (-1)^{(n-1)(m-1)/4}$

Crittografia a Chiave Pubblica 51

Simbolo di Jacobi

Può essere computato con $O((\log n)^2)$ operazioni su bit

- $a = b \pmod n \Rightarrow (a|n) = (b|n)$
- $(ab|n) = (a|n) (b|n)$
- $(2|n) = (-1)^{(n^2-1)/8}$
- $(m|n) = (n|m) (-1)^{(n-1)(m-1)/4}$

Esempio: $(158|235) = (2|235) (79|235)$
 $= (-1) (235|79) (-1)^{(78)(234)/4}$
 $= (77|79)$
 $= (79|77) (-1)^{(76)(78)/4}$
 $= (2|77)$
 $= -1$

Crittografia a Chiave Pubblica 52

Test di Miller-Rabin

- Insieme di *witness di Miller-Rabin*
 $W_{MR}(n) = \{a : a^r \neq 1 \pmod n \text{ and } a^{2^j r} \neq -1 \pmod n \text{ per ogni } j \in [0, s-1]\}$
 $n-1 = 2^s r$ con r dispari
- Se n è un primo dispari
 $W_{MR}(n)$ è vuoto
- Se n è un numero composto dispari ($n \neq 9$)
 $|W_{MR}(n)| \geq (3/4) \cdot \phi(n)$

Crittografia a Chiave Pubblica 53

Witness di Miller-Rabin

n=91

91 = 7 · 13

91-1 = 2¹ · 45

Witness di Miller-Rabin

$$W_{MR}(91) = \{a : a^{45} \neq 1 \pmod{91} \text{ and } a^{2 \cdot 45} \neq -1 \pmod{91} \text{ per ogni } j \in [0,0]\}$$

Se n è un numero composto dispari

$$|W_{MR}(n)| \geq (3/4) \cdot \phi(n)$$

Strong liars: elementi in $\{1,2,\dots,90\} / W_{MR}(91)$
 {1,9,10,12,16,17,22,29,38,53,62,69,74,75,79,81,82,90}
 Sono 18 = $\phi(91)/4$ elementi

Crittografia a Chiave Pubblica 54

Witness di Miller-Rabin

n=105

105 = 3 · 5 · 7

105-1 = 2³ · 13

Witness di Miller-Rabin

$$W_{MR}(105) = \{a : a^{13} \neq 1 \pmod{105} \text{ and } a^{2 \cdot 13} \neq -1 \pmod{105} \text{ per ogni } j \in [0,2]\}$$

Strong liars: elementi in $\{1,2,\dots,104\} / W_{MR}(105)$
 {1,104}

Sono solo 2 < 12 = $\phi(105)/4$ elementi

In genere strong liars molto meno di $\phi(n)/4$

Crittografia a Chiave Pubblica 55

Fattorizzazione

Dato n calcolare l'unica fattorizzazione

$$n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$$

con p_i primo ed $e_i \geq 0$

Splitting di n: calcolare due interi $a, b > 1$
 (fattori non-triviali) tali che $n = a \cdot b$

Crittografia a Chiave Pubblica 56

Fattorizzazione: un semplice algoritmo

Calcolo di un fattore primo:

Per tutti i primi p in $[2, \sqrt{n}]$
 Se $p|n$ allora p è fattore di n

Complessità caso peggiore $\Theta(\sqrt{n})$
 (accade se $n = pq$)

Se n ha 512 bit allora $\sqrt{n} \approx 2^{256}$

Crittografia a Chiave Pubblica 57

Fattorizzazione: complessità algoritmi

Complessità di tempo sub-esponenziale in media

$$L_q[a,c] = O(e^{(c+o(1))(\ln q)^a (\ln \ln q)^{1-a}})$$

con $c > 0$ ed $0 < a < 1$

- Algoritmo basato su curve ellittiche: $L_n[1/2, 1]$
- Quadratic sieve:** $L_n[1/2, 1]$
- General Number field sieve:** $L_n[1/3, 1.923]$

il più veloce

Crittografia a Chiave Pubblica 58

Quadratic sieve in pratica

- Quadratic sieve** è migliore dell'algoritmo basato su curve ellittiche
- Implementazioni del **Quadratic sieve**

anno	numero digit	mips per anno
1984	71	0.01
1988	106	140
1993	120	825
1994	129	5000

1 mips per anno
 $\approx 3 \cdot 10^{13}$ istruzioni

425 bit

RSA-129
 1600 computer per 8 mesi factoring by e-mail

Crittografia a Chiave Pubblica 59

General Number field sieve in pratica

- General Number field sieve è migliore del Quadratic sieve solo per interi di almeno 110-120 cifre decimali
- RSA-130, fattorizzato nel 1996 con 500 mips per anno
- RSA-140, fattorizzato il 2 febbraio 1999, elapsed time 9 settimane
- RSA-155, fattorizzato il 22 agosto 1999, con 8000 mips per anno, elapsed time 7.4 mesi

160	175-400 MHz SGI and Sun workstations
8	250 MHz SGI Origin 2000 processors
120	300-450 MHz Pentium II PCs
4	500 MHz Digital/Compaq boxes

512 bit!

Crittografia a Chiave Pubblica 60

Calcolo di fattori "piccoli"

Algoritmi per calcolare un fattore p di n:

- Algoritmo Rho di Pollard: tempo medio $O(\sqrt{p})$
- Algoritmo basato su curve ellittiche: tempo medio $L_p[1/2, \sqrt{2}] = O(e^{(\sqrt{2}+o(1))(\ln n)^{1/2}(\ln \ln n)^{1/2}})$

I numeri più difficili da fattorizzare sono del tipo $n = pq$ con p,q primi della stessa lunghezza

Crittografia a Chiave Pubblica 61

Una strategia generale per fattorizzare

Cerca fattori "piccoli"

- Prova divisione per alcuni piccoli primi (2,3,5,7,...)
- Prova algoritmo Rho di Pollard
- Prova algoritmo basato su curve ellittiche

Cerca fattori "grandi"

- Prova Quadratic sieve oppure General Number field sieve

Crittografia a Chiave Pubblica 62

Che modulo scegliere?

- Ad oggi, i numeri più difficili da fattorizzare sono del tipo $n = p \cdot q$ con p,q primi della stessa lunghezza
- ... e di almeno (per essere tranquilli!)
 - 768 bit (= 230 digit) per uso personale
 - 1024 bit per le aziende
 - 2048 per chiavi "importanti" ad esempio **Autorità di Certificazione**

Crittografia a Chiave Pubblica 63

Intrattabilità della fattorizzazione

La sicurezza di molte tecniche crittografiche si basa sulla intrattabilità della fattorizzazione:

- crittosistema RSA, Rabin
- firme digitali RSA
- ...


Crittografia a Chiave Pubblica 64

Sicurezza di RSA


Se  potesse fattorizzare...

- Fattorizza n
- Computa $(p-1)(q-1)$
- Computa $d \leftarrow e^{-1} \text{ mod } (p-1)(q-1)$


Crittografia a Chiave Pubblica 65




Sicurezza di RSA

Se  potesse computare $\varphi(n)=(p-1)(q-1)$ potrebbe calcolare $d \leftarrow e^{-1} \bmod (p-1)(q-1)$

Crittografia a Chiave Pubblica 66




Sicurezza di RSA

Se  potesse computare $\varphi(n)=(p-1)(q-1)$, potrebbe calcolare $d \leftarrow e^{-1} \bmod (p-1)(q-1)$


$$\begin{cases} n = pq \\ \varphi(n) = (p-1)(q-1) \end{cases}$$

sostituendo $p = n/q$
 $p^2 - (n-\varphi(n)+1)p + n = 0$
 Due soluzioni: p, q

Crittografia a Chiave Pubblica 67



Sicurezza di RSA

Se  potesse computare $\varphi(n) = (p-1)(q-1)$, potrebbe calcolare $d \leftarrow e^{-1} \bmod (p-1)(q-1)$


$$\begin{cases} n = pq \\ \varphi(n) = (p-1)(q-1) \end{cases}$$

sostituendo $p = n/q$
 $p^2 - (n-\varphi(n)+1)p + n = 0$


$$\begin{cases} 84.773.093 = pq & p^2 - 18.426 p + 84.773.093 = 0 \\ 84.754.668 = (p-1)(q-1) \end{cases}$$

radici: 9539 e 8887


Crittografia a Chiave Pubblica 68




Sicurezza di RSA

Se  potesse computare $d \dots$


Crittografia a Chiave Pubblica 69




Sicurezza di RSA

Se  potesse computare $d \dots$ ma questo è computazionalmente equivalente a fattorizzare!

Crittografia a Chiave Pubblica 70



Sicurezza di RSA

Se  potesse computare $d \dots$ ma questo è computazionalmente equivalente a fattorizzare!

Un algoritmo che computa d (con input n, e) può essere usato come oracolo in un algoritmo **Las Vegas** che fattorizza n con probabilità $\geq 1/2$


Crittografia a Chiave Pubblica 71

Piccoli messaggi

❑ Posso cifrare un solo digit 0,1,...,9 con RSA ?

$$C \leftarrow M^e \bmod n$$

Esempio: $125 \leftarrow 5^3 \bmod 6.012.707$



Crittografia a Chiave Pubblica 72

Piccoli messaggi


❑ Posso cifrare un solo digit 0,1,...,9 con RSA ?

$$C \leftarrow M^e \bmod n$$

Esempio: $125 \leftarrow 5^3 \bmod 6.012.707$

❑ Se $M < n^{1/e}$ allora

- uso di Digital Envelope
- *saltin*g del messaggio M



Crittografia a Chiave Pubblica 73

Proprietà moltiplicativa di RSA

Proprietà di omomorfismo

$$\left. \begin{array}{l} C_1 = M_1^e \bmod n \\ C_2 = M_2^e \bmod n \end{array} \right\} (M_1 M_2)^e = M_1^e M_2^e = C_1 C_2 \bmod n$$

Crittografia a Chiave Pubblica 74

Proprietà moltiplicativa di RSA

$$\left. \begin{array}{l} C_1 = M_1^e \bmod n \\ C_2 = M_2^e \bmod n \end{array} \right\} (M_1 M_2)^e = M_1^e M_2^e = C_1 C_2 \bmod n$$

Chosen-ciphertext attack adattivo
 Obiettivo: decifrare C (= $M^e \bmod n$)

Decifrazione (d,n)

Scelgo x a caso

$$C' \leftarrow C \cdot x^e \bmod n$$

$$M' \leftarrow (C')^d \bmod n$$

$$M \leftarrow M' \cdot x^{-1} \bmod n$$

Crittografia a Chiave Pubblica 75

Attacchi ad RSA

- ❑ Piccolo esponente pubblico (ad es., $e = 3$)
 - Stesso messaggio inviato a più utenti
- ❑ Attacchi ad implementazioni:
 - Timing Attack
 - Random Faults
 - Attacco di Bleichenbacher [1998] su PKCS 1 (vecchia versione): *uso del Web browser come oracolo!*

02 Random 00 M

16 bit

Crittografia a Chiave Pubblica 76

Informazioni parziali per RSA

Dato C $C = M^e \bmod n$

- potrebbero esserci informazioni parziali sul messaggio M "facili" da ottenere (senza dover decifrare C !)
- e ... "difficili" ?

hard bit : ultimi $c \cdot \log \log n$ bit

Crittografia a Chiave Pubblica 77

Un bit "facile" per RSA

$C = M^e \text{ mod } n$

E' facile calcolare il simbolo di Jacobi del testo in chiaro

$$(C|n) = (M^e|n) = (M|n)^e = (M|n)$$

Crittografia a Chiave Pubblica 78

Un bit "difficile" per RSA

$$\text{half}_{n,e}(C) = \begin{cases} 0 & \text{se } M < n/2 \\ 1 & \text{se } M > n/2 \end{cases}$$

Crittografia a Chiave Pubblica 79

Un bit "difficile" per RSA

$$\text{half}_{n,e}(C) = \begin{cases} 0 & \text{se } M < n/2 \\ 1 & \text{se } M > n/2 \end{cases}$$

Calcolare $\text{half}_{n,e}(\cdot)$ è computazionalmente equivalente ad invertire RSA

The diagram shows a flow: $C \rightarrow$ [Calcola $\text{half}_{n,e}(\cdot)$] $\rightarrow \text{half}_{n,e}(C) \rightarrow$ [Decifra $\text{half}_{n,e}$] \rightarrow [Calcola $\text{half}_{n,e}(\cdot)$] $\rightarrow M$. A red arrow points from the first box to the second.

Crittografia a Chiave Pubblica 80

Un bit "difficile" per RSA

$$\text{half}_{n,e}(C) = \begin{cases} 0 & \text{se } M < n/2 \\ 1 & \text{se } M > n/2 \end{cases}$$

The diagram shows a horizontal bar representing the interval $M \in [0, n]$. A vertical line at $n/2$ divides it into two halves. The left half is labeled '0' and the right half is labeled '1'.

Crittografia a Chiave Pubblica 81

Un bit "difficile" per RSA

$$\text{half}_{n,e}(C) = \begin{cases} 0 & \text{se } M < n/2 \\ 1 & \text{se } M > n/2 \end{cases}$$

$C' \leftarrow C \cdot (2^e \text{ mod } n) \quad C' = (2 \cdot M)^e \text{ mod } n$

The diagram shows a horizontal bar representing the interval $M \in [0, n]$. Vertical lines at $n/4, n/2, 3n/4$ divide it into four quarters. The bits are 0, 1, 0, 1 from left to right.

Crittografia a Chiave Pubblica 82


Un bit "difficile" per RSA

$$\text{half}_{n,e}(C) = \begin{cases} 0 & \text{se } M < n/2 \\ 1 & \text{se } M > n/2 \end{cases}$$

$C'' \leftarrow C \cdot (4^e \text{ mod } n) \quad C'' = (4 \cdot M)^e \text{ mod } n$

The diagram shows a horizontal bar representing the interval $M \in [0, n]$. Vertical lines at $n/8, n/4, 3n/8, n/2, 5n/8, 3n/4, 7n/8$ divide it into eight eighths. The bits are 0, 1, 0, 1, 0, 1, 0, 1 from left to right.


Crittografia a Chiave Pubblica 83



Un bit "difficile" per RSA

- ❑ Ricerca binaria
- ❑ Dopo $\log n$ chiamate ad $\text{half}_{n,e}(\cdot)$ l'intervallo consta di un solo valore

Crittografia a Chiave Pubblica 84



Predicato **parità**

$C = M^e \text{ mod } n$


$\text{parità}_{n,e}(C)$ = bit meno significativo di M

I predicati $\text{parità}_{n,e}(\cdot)$ e $\text{half}_{n,e}(\cdot)$ sono computazionalmente equivalenti

$\text{parità}_{n,e}(C) = \text{half}_{n,e}(y)$

$y = C \cdot (2^{-1})^e \text{ mod } n$
 $= (M \cdot 2^{-1})^e \text{ mod } n$

Crittografia a Chiave Pubblica 85




Crittografia probabilistica

Testo in chiaro $M = M_1 M_2 M_3 \dots$

Testo cifrato $C = C_1 C_2 C_3 \dots$

$M_i = \text{predicato_difficile}(C_i)$


Crittografia a Chiave Pubblica 86




Alcuni Crittosistemi a chiave pubblica

- ❑ RSA [1977] (fattorizzazione)
- ❑ Merkle-Hellman [1978] (zaino 0-1) **rotto!**
- Molte varianti... rotte! Resiste Chor-Rivest [1988]
- ❑ Rabin [1979] (fattorizzazione)
- ❑ McEliece [1978] (decodifica codici lineari)
- ❑ El-Gamal [1984] (logaritmo discreto)
- ❑ Uso di curve ellittiche [1985]
- curve iperellittiche [1989]
- automi cellulari [1985]



Crittografia a Chiave Pubblica 87



Funzioni one-way

"Facili" da calcolare e 

"difficili" da invertire

Crittografia a Chiave Pubblica 88



Funzioni one-way trapdoor

"Facili" da calcolare e "difficili" da invertire




... a meno che si conosca una "trapdoor"



Crittografia a Chiave Pubblica 89

Alcuni brevetti software (USA)

numero	oggetto	inventore	scadenza
3.962.539	DES	Ehrsam ed al.	1993
4.200.770	Diffie-Hellman	Hellman, Diffie, Merkle	1997
4.218.582	Critt. chiave pubblica	Hellman, Merkle	1997
4.424.414	Pohling-Hellman	Hellman, Pohling	3 gen 2001
4.405.829	RSA	Rivest, Shamir, Adleman	20 sett 2000
4.748.668	Shamir-Fiat	Shamir, Fiat	2005
4.850.017	Vettori di controllo	Matyas, Meyer, Bracht	2006
5.140.634	Guillou-Quisquater	Guillou-Quisquater	2009
5.231.668	DSA	Kravitz	2010
5.214.703	IDEA	Massey-Lai	2010
5.276.737	Fair Cryptosystems	Micali	2011

Crittografia a Chiave Pubblica 90

Export dagli USA

International Traffic in Arms Regulations (ITAR) [1989]

- ❑ United States Munitions List
- ❑ Strong encryption
 - RSA \geq 512 bit
 - cifrari a blocco (DES, IDEA, RC6,...) \geq 40 bit
- ❑ eccetto se limitato a
 - controllo accessi (ad es., ATM)
 - autenticazione dati (ad es., MAC, firme)
- ❑ Permesso dal *Department of Commerce*,
ad es. *Cybercash*, cifratura RSA 768 bit per transazioni finanziarie

non deve essere facilmente utilizzabile per cifrare!

Crittografia a Chiave Pubblica 91