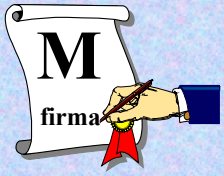
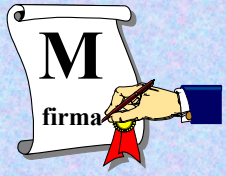
Firma Digitale



Equivalente alla firma convenzionale

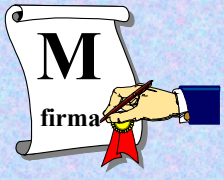
Firma Digitale



Equivalente alla firma convenzionale

Soluzione naive:
incollare firma digitalizzata

Firma Digitale



Equivalente alla firma convenzionale

Soluzione naive:
incollare firma digitalizzata

Desiderata per la Firma Digitale

La firma digitale deve poter essere facilmente prodotta dal legittimo firmatario

Nessun utente deve poter riprodurre la firma di altri

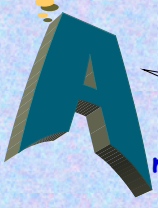
Chiunque può facilmente verificare una firma

Firma digitale

chiave privata
kpriv

file pubblico

utente	chiave pubblica
A	kpub
…	…



Devo firmare M

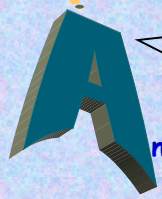
nnarella

Firma digitale

chiave privata
kpriv

file pubblico

utente	chiave pubblica
A	kpub
…	…



Firma di M
F ← FIRMA (M,kpriv)

nnarella

Verifica firma digitale

file pubblico

utente	chiave pubblica
A	kpub
…	…

Devo verificare se F è una firma di A per M

erificatore

Verifica firma digitale

file pubblico

utente	chiave pubblica
A	kpub
…	…

Verifica firma di M
vera se **VERIFICA** (F,M,kpub) = SI
falsa altrimenti

erificatore

Firme digitali che vedremo

- RSA
- Digital Signature Standard (DSS)

RSA

Proposto nel 1978 da


Rivest


Shamir


Adleman

Sicurezza basata sulla difficoltà di fattorizzare

Chiavi RSA

chiave privata (n,d)

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

n = pq
p,q primi

ed = 1 mod (p-1)(q-1)

nnarella

Firma RSA

chiave privata (n,d)

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Devo firmare M

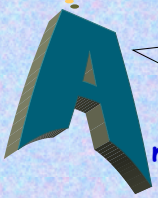
nnarella

Firma RSA

chiave privata (n,d)

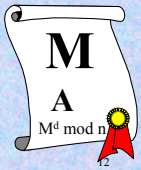
file pubblico

utente	chiave pubblica
A	(n,e)
…	…



Annarella

Firma di M
$Firma_{(n,d)}(M) = M^d \bmod n$



Verifica firma RSA

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Verificatore

Devo verificare se F è una firma di A per M

Verifica firma RSA

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Verificatore

Verifica firma di M
vera se $M = F^e \bmod n$
falsa altrimenti

"Piccolo" esempio: Chiavi RSA

chiave privata (n=3337, d=1019)

file pubblico

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

Annarella

3337 = 47 ·71
p = 47, q = 71

ed = 79 ·1019 = 1 mod 3220
(p-1)(q-1) = 46 ·70 = 3220

"Piccolo" esempio: Firma RSA

chiave privata (n=3337, d=1019)

file pubblico

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

Annarella

Devo firmare M=1570

"Piccolo" esempio: Firma RSA

chiave privata (n=3337, d=1019)

file pubblico

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

Annarella

Firma di 1570
$= 1570^{1019} \bmod 3337$
$= 668$

"Piccolo" esempio: Verifica firma

file pubblico

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

Verifica firma di 1570
$1570 = 668^{79} \bmod 3337$

V erificatore

Correttezza verifica firma RSA

$F^e \bmod n = (M^d)^e \bmod n$
$= M^{ed} \bmod n$
$= M \bmod n$
$= M$

$ed = 1 \bmod (p-1)(q-1)$
$x \in Z_n^* \Rightarrow x^{(p-1)(q-1)} = 1 \bmod n$

poichè $0 \le M < n$

Prova per tutti gli x mediante il teorema del resto cinese

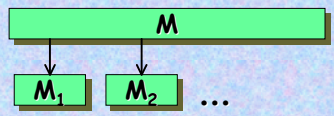
Esercizio

Svolgere "piccolo" esempio firma RSA
- Fissato e = 3
- Calcolo p, q
- Calcolo n
- Calcolo d
- Calcolo firma
- Verifica firma

Firma digitale di messaggi grandi

Se M>n, come si firma?

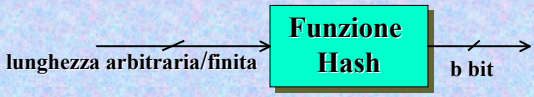
Prima soluzione



$Firma(M) \leftarrow (Firma(M_1), Firma(M_2), \ldots)$

Problemi
- Efficienza
- Permutazione/composizione delle firme → nuova firma

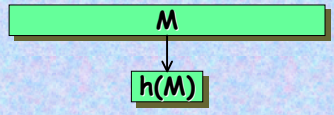
Funzioni Hash



- Il valore hash h(M) è una rappresentazione non ambigua e non falsificabile del messaggio M
- Proprietà:
  - comprime
  - facile da computare
  - **Sicurezza forte**: computazionalmente difficile trovare 2 diversi messaggi con lo stesso valore hash
  - **One-way**: dato y è computazionalmente difficile trovare M tale che y = h(M)

Firma digitale di messaggi grandi

Soluzione



$Firma(M) \leftarrow (Firma(h(M))$

Vantaggi
- Efficienza
- Integrità

Sicurezza firma RSA

Voglio falsificare la firma di M da parte di A

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Devo calcolare $M^d \bmod n$

Equivalente a "rompere" crittosistema RSA

Sicurezza firma RSA

Voglio generare messaggi e firme da parte di A

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

1. Scelgo F a caso
2. $M \leftarrow F^e \bmod n$

Firma RSA con hash

chiave privata (n,d)

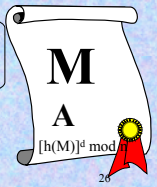
file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Firma di M

$Firma_{(n,d)}(M) = [h(M)]^d \bmod n$

nnarella



Verifica firma RSA con hash

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Devo verificare se F è una firma di A per M

erificatore

Verifica firma RSA

file pubblico

utente	chiave pubblica
A	(n,e)
…	…

Verifica firma di M
vera se $h(M) = F^e \bmod n$
falsa altrimenti

erificatore

Sicurezza firma RSA

Voglio generare messaggi e firme da parte di A

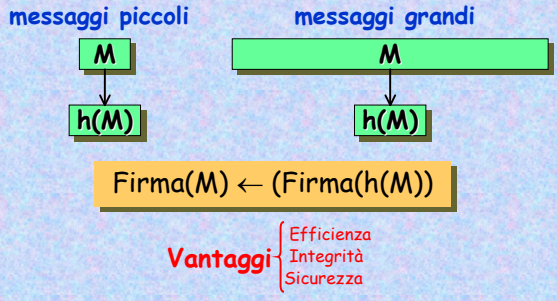
file pubblico

utente	chiave pubblica
A	(n,e)
…	…

1. Scelgo F a caso
2. $z \leftarrow F^e \bmod n$
3. $M \leftarrow h^{-1}(z)$

Come faccio ad invertire h?
$M \leftarrow h^{-1}(z)$

Firma digitale con hash

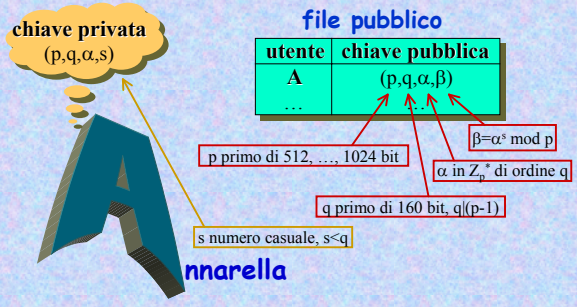


Digital Signature Standard (DSS)

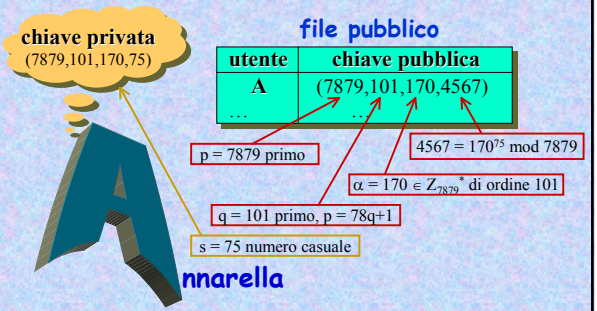
- Proposto nell'agosto del 1991 dal National Institute of Standard and Technology (NIST)
  - Digital Signature Algorithm (DSA)
  - Digital Signature Standard (DSS)
- Standard rivisto nel 1993, in risposta alle critiche
- Modifica ingegnosa dello schema di El Gamal
- Firme DSS sempre di 320 bit (buone per smart card)
- Sicurezza basata sulla difficoltà del logaritmo discreto

Chiavi DSA

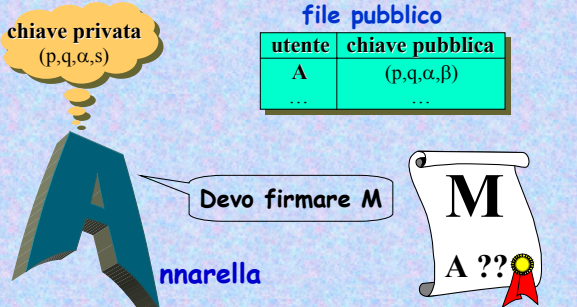
Chiavi DSA



Chiavi DSA ("piccolo" esempio)



Firma DSA



Firma DSA

chiave privata
(p,q,α,s)

file pubblico

utente	chiave pubblica
A	(p,q,α,β)
…	…

Firma di M
r ← numero casuale in [1,q-1]
γ ← (α^r mod p) mod q
δ ← (SHA(M)+sγ)r^{-1} mod q
$firma_{(p,q,\alpha,s)}(M,r) = (\gamma,\delta)$

A nnarella

M
A
(γ,δ)

Verifica firma DSA

M
A (γ,δ)

file pubblico

utente	chiave pubblica
A	(p,q,α,β)
…	…

Devo verificare se (γ,δ) è una firma di A per M

Verificatore

Verifica firma DSA

M
A (γ,δ)

file pubblico

utente	chiave pubblica
A	(p,q,α,β)
…	…

Verifica firma di M
e' ← SHA(M)δ^{-1} mod q
e'' ← γδ^{-1} mod q
vera se γ = ($\alpha^{e'}\beta^{e''}$ mod p) mod q
falsa altrimenti

Verificatore

Efficienza firma DSA

Firma_DSA(M,p,q,α,s)
r ← numero casuale in [1,q-1]
γ ← (α^r mod p) mod q
δ ← (SHA(M)+sγ)r^{-1} mod q
output $firma_{(p,q,\alpha,s)}(M,r) = (\gamma,\delta)$

- Lunghezza firma = 320 bit
- Computazioni off-line: r, sγ, r^{-1} mod q
- Computazioni on-line: SHA(M), +, ·

Verifica firma DSA

Verifica_firma_DSA(M,γ,δ,p,q,α,β)
e' ← SHA(M)δ^{-1} mod q
e'' ← γδ^{-1} mod q

$ver_{(p,q,\alpha,\beta)}(M,\gamma,\delta)$ = { vera se γ = ($\alpha^{e'}\beta^{e''}$ mod p) mod q ; falsa altrimenti }

Output $ver_{(p,q,\alpha,\beta)}(M,\gamma,\delta)$

Correttezza verifica firma DSA

($\alpha^{e'}\beta^{e''}$ mod p) mod q — e' = SHA(M)δ^{-1} mod q; e'' = γδ^{-1} mod q; β = α^s mod p

= ($\alpha^{SHA(M)\delta^{-1} \bmod q}$ $\alpha^{s\gamma\delta^{-1} \bmod q}$ mod p) mod q

α è di ordine q

= ($\alpha^{SHA(M)\delta^{-1}+s\gamma\delta^{-1}}$ mod p) mod q

$\delta^{-1}(SHA(M)+s\gamma) = r \bmod q$

= (α^r mod p) mod q

= γ

Generazione di p e q

Scegli p

Scegli q di 160 bit tale che q|(p-1)

Generazione di p e q

Scegli p

Scegli q di 160 bit tale che q|(p-1)

Generazione di p e q

- Scegli un primo q di 160 bit
- Scegli un primo p di 512/1024 bit tale che q|(p-1)
  - Scegli X di 512 bit (oppure ... 1024 bit)
  - $p \leftarrow X - ((X \bmod 2q)-1)$ 2q|(p-1)
  - se p è primo e $p \geq 2^{511}$ esci altrimenti riprova

Scelta di un elemento di ordine q

- Ordine di $\alpha \in Z_n^*$ = il più piccolo intero positivo r tale che $\alpha^r = 1 \bmod n$
- p,q primi tali che q|(p-1)

Scegli_ordineq (p,q)

1. $g \leftarrow$ elemento scelto a caso in Z_p^*
2. $\alpha \leftarrow g^{(p-1)/q} \bmod p$
3. **if** $\alpha \neq 1$ **then return** α **else** go to 1.

Correttezza di Scegli_ordineq

Scegli_ordineq (p,q)

1. $g \leftarrow$ elemento scelto a caso in Z_p^*
2. $\alpha \leftarrow g^{(p-1)/q} \bmod p$
3. **if** $\alpha \neq 1$ **then return** α **else** go to 1.

- $\alpha^q \equiv (g^{(p-1)/q})^q \equiv g^{p-1} \equiv 1 \bmod p$
- q è il più piccolo intero tale che $\alpha^q \equiv 1 \bmod n$
- α è di ordine q

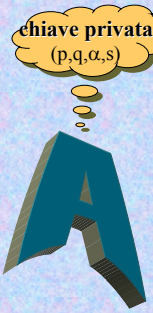
dal Teorema di Lagrange l'ordine di α divide q

Probabilità successo singola iterazione

- Se g è un generatore allora $g^{(p-1)/q} \neq 1 \bmod p$
- Probabilità successo $\geq$ Probabilità che g è generatore $> 1/(6\ln\ln(p-1))$
- Numero medio di iterazioni $< 6\ln\ln(p-1)$

512 bit	$6 \cdot \ln\ln(2^{512}) \approx 35{,}23$
1024 bit	$6 \cdot \ln\ln(2^{1024}) \approx 39{,}38$
2048 bit	$6 \cdot \ln\ln(2^{2048}) \approx 43{,}54$

Chiavi globali ed individuali



file pubblico

utente	chiave pubblica
A	(p,q,α,β)
…	…

- Sicurezza basata sul valore privato s
- I valori p,q,α possono essere gli stessi per un gruppo di utenti
- Un'autorità sceglie p,q,α
- Il singolo utente sceglie solo s,β

Chiavi globali ed individuali

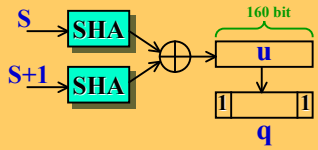
file pubblico

utente	chiave pubblica
A	(p,q,α,β)
…	…

- Sicurezza basata sul valore privato s
- I valori p,q,α possono essere gli stessi per un gruppo di utenti
- Un'autorità sceglie p,q,α
- Il singolo utente sceglie solo s,β

Generazione di q

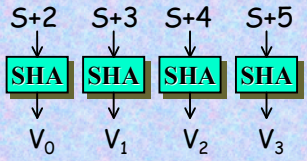
- Scegli a caso S di ≥160 bit



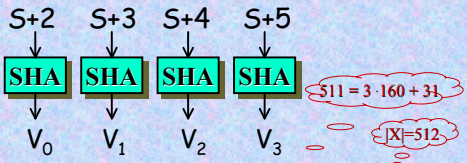
- Ripeti con un nuovo S finchè q è primo

S è un testimone della validità di q

Generazione di p (512 bit)



Generazione di p (512 bit)

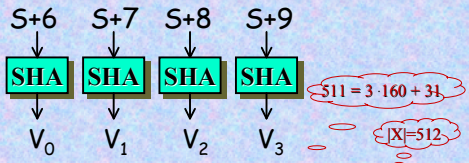


$X \leftarrow V_0+V_1 2^{160}+ V_2 \cdot 2^{2\cdot160}+(V_3 \bmod 2^{31}) \cdot 2^{3\cdot160}+2^{511}$

$p \leftarrow X - ((X \bmod 2q)-1)$ — $2q(p-1)$

se p è primo e $p \geq 2^{511}$ esci altrimenti …

Generazione di p (512 bit)



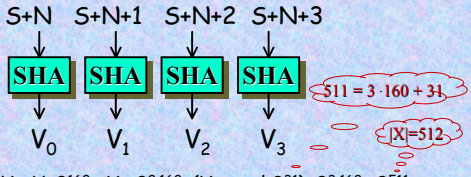
$X \leftarrow V_0+V_1 2^{160}+ V_2 \cdot 2^{2\cdot160}+(V_3 \bmod 2^{31}) \cdot 2^{3\cdot160}+2^{511}$

$p \leftarrow X - ((X \bmod 2q)-1)$ — $2q(p-1)$

se p è primo e $p \geq 2^{511}$ esci altrimenti …

Generazione di p (512 bit)

N←2,6,10,…, (per ≤4096 volte)



$X \leftarrow V_0+V_1 \cdot 2^{160}+ V_2 \cdot 2^{2 \cdot 160}+(V_3 \bmod 2^{31}) \cdot 2^{3 \cdot 160}+2^{511}$

$p \leftarrow X - ((X \bmod 2q)-1)$ — 2q|(p-1)

se p è primo e $p \geq 2^{511}$ esci — S,(N-2)/4 sono testimoni

altrimenti …

Generazione di p e q

```
Seleziona_pq(L)
(1) Computa interi n e b tali che L-1=160n+b
(2) repeat
(3)     repeat
(4)         S ← sequenza casuale di almeno 160 bit
(5)         g ← |S|
(6)         U ← SHA(S)⊕ SHA((S+1) mod 2^g)
(7)         Forma q da U ponendo il MSB ed il LSB ad 1
(8)     until q primo
(9)     C ← 0
(10)    N ← 2
(11)    repeat
(12)        for k=0 to n do V_k ← SHA(S+N+k) mod 2^g
(13)        W ← V_0+V_1·2^160+...+V_n-1·2^160(n-1)+(V_n mod 2^b)·2^160n
(14)        X ← W+2^L-1
(15)        p ← X-((X mod 2q)-1)
(16)    until (p primo) or (p<2^L-1)
(17)    if p<2^L-1
(18)        then C ← C+1
(19)           N ← N+n+1
(20)           if C<4096 then goto step (12)
(21)                     else Help ← falso
(22)        else Help ← vero
(23) until Help
(24) return p,q,S,C
```

Confronto tempi firme RSA e DSA

	DSS	RSA	DSS con p,q,a comuni
precomputazioni	14 sec		4 sec
firma	0.3 sec	15 sec	0.3 sec
verifica	16 sec	1.5 sec	10 sec
	1-5 sec Off Cards		1-3 sec Off Cards

- Implementazioni su smart card [1993]
- Computazioni Off Cards su 80386 a 33MHz

Prestazioni

Pentium II 400
OpenSSL

	bit chiave	firme/s	verifiche/s
RSA	512	342	3287
DSA	512	331	273
RSA	1024	62	1078
DSA	1024	112	94
RSA	2048	10	320
DSA	2048	34	27