

# Timing Attacks ad RSA



# Agenda

- Introduzione ad RSA
- Cos'è il Timing Attacks
- La storia del Timing Attack....
- Algoritmo del Quadrato e del Multiplo
- Attacco alla esponenziazione modulare
- Timing Attacks – L'idea
- L'attacco nei dettaglio
- Probabilità di successo
- Contromisure
- RSA Blinding
- Conclusioni



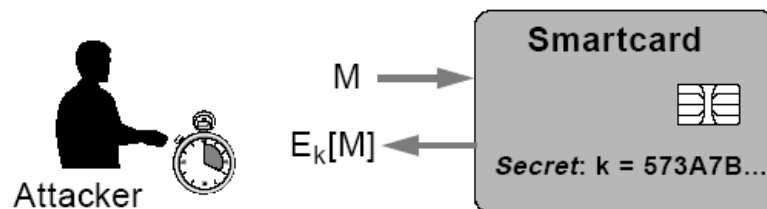
# Introduzione ad RSA (Rivest-Shamir-Adleman)

- Algoritmo di generazione chiave RSA
  - 1. Selezioniamo dapprima 2 numeri primi  $p, q$
  - 2. Sia  $n = p * q$
  - 3. Sia  $\varphi(n) = (p-1) * (q-1)$
  - 4. Scegliamo  $e$ , tale che  $\text{MCD}(e, \varphi(n)) = 1$
  - 5. Calcoliamo  $d = e^{-1} \pmod{\varphi(n)}$
  - 6. Chiave Pubblica =  $(e, n)$                       Chiave Privata =  $(d, n)$
  
- Cifratura:  $c = m^e \pmod n$   
Decifratura:  $m = c^d \pmod n$



# Cos'è il Timing Attacks

- Espone informazioni private semplicemente misurando l'ammontare di tempo richiesto durante le operazioni svolte sulla chiave privata



# La storia del Timing Attacks finora....



**Paul Kocher**, President/ Chief scientist  
Cryptography Research, Inc.

- Possibili proposte del timing attacks contro gli algoritmi di Diffie-Hellman, RSA, DSS ed altri sistemi nel 1995.
- I ricercatori non sono giunti ad alcuna conclusione usando simili attacchi contro l'algoritmo RSA con CRT.



# La storia del Timing Attacks finora....



David Brumley (sopra) / Dan Boneh (sotto), Stanford University



- "Remote Attacks are practical" – 2003
- Implementazione di attacchi contro OpenSSL in diverse situazioni.
- Dimostrazione che sono possibili i timing attacks tra computer separati anche da molteplici routers.



# Algoritmo left-to-right

- $Y = M^d \bmod N$

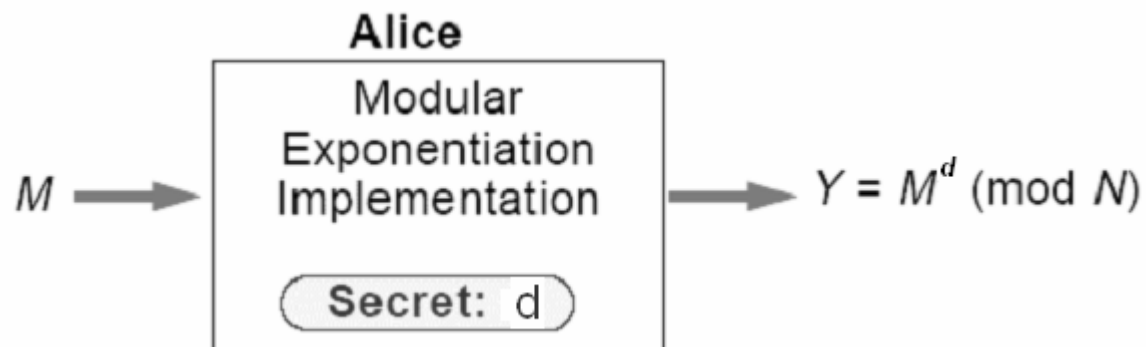
**INPUT:**  $M, N, d = (d_{n-1}d_{n-2} \dots d_1d_0)_2$

**OUTPUT:**  $S = M^d \bmod N$

```
1  $S \leftarrow 1$ 
2 for  $j = n - 1 \dots 0$  do
3      $S \leftarrow S^2 \bmod N$ 
4     if  $d_j = 1$  then
5          $S \leftarrow S \cdot M \bmod N$ 
6 return  $S$ 
```



# Attacco alla Esponenziazione Modulare



**L'attaccante colleziona molti campioni di timing data:**

For  $i = 1$  to  $K$  (  $K$  è un grande numero, potrebbe essere 1000)

{

domandiamo ad Alice di esponenziare un messaggio:  $M_i$   
misuriamo il tempo per esponenziare  $M_i$ , e chiamiamolo  $T_i$

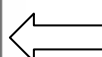
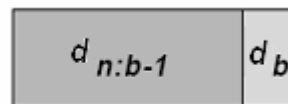
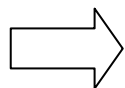
}



## Time Attack – L'idea

- L'attaccante colleziona "molte" coppie  $(M_i, T_i)$ 
  - $M_i$  rappresenta il messaggio  $i$ -esimo inviato
  - $T_i$  rappresenta il tempo impiegato per rispondere ad  $M_i$
- L'attaccante, usando lo stesso algoritmo di Alice:
  - ipotizza  $d_b=0$ , si calcola il tempo impiegato per il bit  $b$   $[T_{i,b}]$  e si calcola  $G_0 = \text{var}(T_i - T_{i,b})$
  - ipotizza  $d_b=1$ , si calcola il tempo impiegato per il bit  $b$   $[T_{i,b}]$  e si calcola  $G_1 = \text{var}(T_i - T_{i,b})$
  - se  $(G_0 < G_1)$  allora apprende che  $d_b=0$  altrimenti  $d_b=1$

(b-1) bit calcolati  
in precedenza



ipotesi del bit  
b-esimo della chiave privata



# L'attacco in dettaglio

- L'attaccante misura il tempo:

$$\text{Time to exponentiate } M_i: T_i = e + \sum_{j=1}^n t_j$$

- $t_j$  rappresenta il tempo necessario per eseguire un ciclo dell'algoritmo di esponenziazione modulare
  - $e$  rappresenta le sorgenti di errore
- Assumiamo che l'attaccante conosca  $b-1$  bits della chiave privata  $d$ , e deve indovinare il bit  $b$ -esimo:

$$T_{i,b} = \sum_{j=1}^b t_j$$



## L'attacco in dettaglio

- Ipotesi corretta del bit  $b$ :

$$\text{var}(T_i - T_{i,b}) = \text{var}\left(e + \sum_{j=1}^n t_j - \sum_{j=1}^b t_j\right) = \text{var}\left(e + \sum_{j=b+1}^n t_j\right)$$

$$= \text{var}(e) + (n-b)\text{var}(t)$$

- Ipotesi sbagliata del bit  $b$ :

$$\text{var}(T_i - T_{i,b}) = \text{var}\left(e + \sum_{j=1}^n t_j - \left(t_b + \sum_{j=1}^{b-1} t_j\right)\right) = \text{var}\left(e - t_b + \sum_{j=b}^n t_j\right)$$

$$= \text{var}(e) + (n-b)\text{var}(t) + 2\text{var}(t)$$

avendo considerato che:  $\text{var}(t) + \text{var}(-t) = 2 * \text{var}(t)$



# Probabilità di successo

$$\text{Prob}( G_0 < G_1 \mid e_b = 0 )$$

- Definiamo  $X \sim N(\mu, \sigma^2)$  come la variabile casuale  $X$  normalmente distribuita con media  $\mu$  e varianza  $\sigma^2$
- Supponiamo di avere  $k$  misure di tempo corrette:  $X_1, X_2, \dots, X_k$
- Supponiamo di avere  $k$  misure di tempo sbagliate:  $Y_1, Y_2, \dots, Y_k$
- Possiamo modellare i dati in due colonne:

$\sigma_0 X_1 + \mu_0$	$(\sigma_0 X_1 + \mu_0) + (\sigma_t Y_1 + \mu_t)$
$\sigma_0 X_2 + \mu_0$	$(\sigma_0 X_2 + \mu_0) + (\sigma_t Y_2 + \mu_t)$
$\vdots$	$\vdots$
$\sigma_0 X_k + \mu_0$	$(\sigma_0 X_k + \mu_0) + (\sigma_t Y_k + \mu_t)$



## Probabilità di successo

- Denotiamo  $V_i = \sigma_0 X_i + \mu_0$  e  $W_i = (\sigma_0 X_i + \mu_0) + (\sigma_t Y_i + \mu_t)$

$$\Pr(S_w^2 > S_v^2) = \Pr\left(\frac{1}{k-1} \sum_{i=1}^k (W_i - \bar{W})^2 > \frac{1}{k-1} \sum_{i=1}^k (V_i - \bar{V})^2\right) = \Pr\left(\sum_{i=1}^k (W_i - \bar{W})^2 > \sum_{i=1}^k (V_i - \bar{V})^2\right)$$

- Se  $k$  è grande allora  $\bar{W} \approx \mu_0 + \mu_t$  e  $\bar{V} \approx \mu_0$

$$\Pr(S_w^2 > S_v^2) \approx \Pr\left(\sum_{i=1}^k (\sigma_0 X_i + \sigma_t Y_i)^2 > \sum_{i=1}^k (\sigma_0 X_i)^2\right) =$$

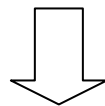
$$= \Pr\left(\sum_{i=1}^k (\sigma_0^2 X_i^2 + 2\sigma_0 \sigma_t X_i Y_i + \sigma_t^2 Y_i^2) > \sum_{i=1}^k \sigma_0^2 X_i^2\right) =$$

$$= \Pr\left(2\sigma_0 \sum_{i=1}^k X_i Y_i + \sigma_t \sum_{i=1}^k Y_i^2 > 0\right)$$

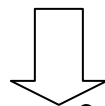


# Probabilità di successo

- Poichè  $X_i$  e  $Y_i$  sono indipendenti:  $E(X_i, Y_i) = E(X_i)E(Y_i) = 0$
- Sapendo che  $Var(X) = E(X^2) - E(X)^2$   
e  $X_i, Y_i$  normalmente distribuite  $E(X_i^2) = 1$   $E(Y_i^2) = 1$



$$E\left(\sum_{i=1}^k Y_i^2\right) = \sum_{i=1}^k E(Y_i^2) = k$$



$$Var(X_i, Y_i) = E(X_i^2 Y_i^2) - (E(X_i Y_i))^2 = E(X_i^2 Y_i^2) = E(X_i^2)E(Y_i^2) = 1$$



# Probabilità di successo

- Applicando il teorema del limite centrale la somma  $\sum_{i=1}^k X_i Y_i$  segue la distribuzione di  $N(0, k)$
- Se  $Z$  è una variazione standard, allora nel caso peggiore:

$$\Pr(S_W^2 > S_V^2) \approx \Pr(2\sigma_0(\sqrt{k}Z) + \sigma_t k > 0) = \Pr(Z > -\frac{\sigma_t}{\sigma_0} \frac{\sqrt{k}}{2}) = \Phi\left(\frac{\sigma_t}{\sigma_0} \frac{\sqrt{k}}{2}\right)$$

$$z = (G_0 - G_1) \sim N(\mu, \sigma^2)$$

$$\text{Probability of Success} = \Phi\left(\frac{\mu}{\sigma}\right)$$

← Gaussian Distribution



# Probabilità di successo

- Per essere più precisi...

$$P \left( Z > -\frac{\sqrt{j(b-c)}}{\sqrt{2(w-b)}} \right)$$

- Argomenti
  - $j$  - numero di misure accurate che l'attaccante ha a disposizione
  - $b$  - il bit corrente
  - $w$  - numero totale di iterazioni ( $\sim$ lunghezza della chiave)
  - $c$  - numero di bit ipotizzati correttamente



# Contromisure

- Esecuzione in tempi costanti
  - Non sempre pratico
  - Non si possono introdurre ottimizzazioni
- Aggiunta di ritardi casuali
  - Overhead
  - Non si possono introdurre ottimizzazioni
- Uso del Blinding



# RSA Blinding

- Generazione di un numero casuale  $r$ .
  - $r$  deve essere compreso tra 1 e  $n-1$
- Calcoliamo  $M' = M * r^e \text{ mod } N$
- Calcoliamo  $Y' = (M')^d \text{ mod } n$ 
  - Questo è il punto dove il timing attack usualmente lavora...
- Calcoliamo  $Y = Y' * r^{-1} \text{ mod } n$ 
  - $Y = Y' * r^{-1} = (M * r^e)^d * r^{-1} = (M * r) * r^{-1} = M^d$
- Dal momento che  $M'$  diventa casuale, il runtime non può essere correlato con l'input  $M$ .
  - Quindi il termine "blinding".
- Timing attack non può ottenere informazioni sulla chiave privata.



# Conclusioni

- Timing attacks pone reali rischi sui sistemi di sicurezza.
- Oggi, è indispensabile esaminare attentamente come salvaguardarsi dal Timing attack sulle molteplici implementazioni.
  - Ad ogni modo – tutto è ancora possibile!

