A.A. 2010/2011
Corso di Laurea Magistrale in Matematica
Teoria dell'Informazione 1
Timing Attack ad RSA

Marco Calderini

15 maggio 2011

Sommario

In questo breve e semplice articolo descriveremo il timing attack ad RSA, implementato tramite l'algoritmo *left-to-right*, introdotto da Kocher nel 1996. Analizzeremo poi la probabilità di successo di tale attacco.

1 Introduzione

Il primo a discutere la tecnica del timing attack fu Kocher [4]. Nel 1996 Kocher presentò i suoi risultati preliminari all'"*RSA Data Security and CRYPTO conferences*".

Il timing attack è una forma di "*Side Channel Analysis*", in cui un utente malintezionato ottiene informazioni utili dall'implementazione di un crittosistema, piuttosto che dalle debolezze, del crittosistema, puramente matematiche.

Side Channel Analysis sfrutta informazioni relative ai tempi, ai consumi di energia o ad altre cose utili a recuperare informazioni sul sistema di crittografia.

Il timing attack analizza le variazioni dei tempi necessari nelle operazioni crittografiche, infatti le operazioni dipendono dall'algoritmo crittografico, quindi dalle chiavi private, e dall'imput.

2 RSA

Prima di discutere il timing attack vediamo la matematica utilizzate dal sistema RSA.

RSA è un crittosistema a chiave pubblica ideato da Rivest, Shamir e Adleman nel 1976. Nell'algoritmo RSA si generano due numeri primi p e q distinti

molto grandi. Si calcolano $N = pq$ e $\phi(N) = (p-1)(q-1)$. Dopodiché si sceglie un numero $e \in \mathbb{Z}_{\phi(N)}$ tale che $MCD(e, \phi(N)) = 1$. Scelto e si calcola $d \in \mathbb{Z}_{\phi(N)}$ per cui $de \equiv 1 mod(\phi(N))$.

A questo punto la coppia (e, N) sarà la chiave pubblica dell'utente e (d, N) la chiave privata.

Con RSA è possibile cifrare e decifrare un messaggio M con le operazioni

$$C = M^e mod(N)$$

per la cifratura e

$$C^d mod(N) = M^{ed} mod(N) = M mod(N)$$

per la decifratura. La "forza matematica" di RSA è basata sulla difficoltà computazionale della fattorizzazione di un numero molto grande.

Un altro uso del crittosistema RSA è la generazione di firme digitali, il cui scopo è quello di verificare la sorgente del messaggio.

Per apporre la firma con RSA si usa la chiave privita d e si esegue l'operazione

$$S = M^d mod(N).$$

Per verificare la validità di un messaggio il destinatario deve solamente usare la chiave pubblica del mittente ed esegue

$$S^e mod(N) = M mod(N).$$

3 L'attacco

IMPUT: $M, N, d = (d_{w-1}...d_0)_2$
OUTPUT: $S = M^d mod(N)$

1. $S \leftarrow 1$;
2. **for** $j = w - 1$ **to** 0 **do**
3. $S \leftarrow S^2 mod(N)$;
4. **if** $d_j == 1$ **then**
5. $S \leftarrow SM mod(N)$;
6. **return** S.

Figura 1: algoritmo *left-to-right*

L'operazione fondamentale usata nel crittosistema RSA è l'esponenziale modulare. Questa è usata sia per cifrare/decifrare, sia per apporre la firma ad un documento.

Gli ideatori di RSA suggerirono un algoritmo per implementare tale operazione basato su operazioni ripetitive di elevazione al quadrato e di moltiplicazione; l'algoritmo è detto *left-to-right*, figura 3.

Vediamo come lavora questo algoritmo.

In input si hanno il messaggio M, il modulo RSA N e la chiave privata d espressa in forma binaria (w è il numero di bit), l'output S è il messaggio firmato. Nei nostro conti supporremo per semplicità che $M \in \mathbb{Z}_N$.

Kocher osservò che l'algoritmo left-to-right, in base al valore dell'esponente vi era o no un'operazione in più (linea 5).

In ogni iterazione, se il bit rilevante dell'esponente è 1, vengono eseguiti entrambi sia il modulo dell'elevazione al quadrato sia il modulo della moltiplicazione (linea 3 e 5), se il bit rilevante è 0 viene eseguito solo il quadrato. Quindi l'ammontare del tempo richiesto per effettuare il ciclo for è influenzato (oltre che dal valore di M) dal valore dell'esponente d.

Il timing attack mostrato da Kocher descrive come un attaccante è in grado di dedurre il valore dei corrispondenti bit di d osservando e confrontando il tempo di esecuzione dei diversi cicli iterativi dell'algoritmo left-to-right.

Supponiamo che il nostro attaccante, Trudy, spedisca una serie di messaggi ad Alice, alla quale richiede di firmarli. Alice firma i messaggi con RSA attraverso l'algoritmo in figura 3. Trudy memorizza i tempo $T_1, ..., T_k$ necessari per firmare i messaggi $M_1, ..., M_k \in \mathbb{Z}_N$.

Ora si suppone che il nostro attaccante sia in grado di simulare i tempi per le firme su un PC identico a quello di Alice. Quindi Trudy è in grado di calcolare il tempo necessario per calcoare $M_i^2 M_i mod(N)$ per ogni $i = 1, ..., k$. Il valore di ogni singolo messaggio M_i influenza tale tempo.

Vediamo ora il timing attack nel dettaglio. Per ogni $i = 1, ..., k$ il valore T_i consiste in diversi fattori come il tempo richiesto per eseguire il quadrato, $s_{i,j}$, per ogni iterata j e il tempo per eseguire la moltiplicazione, $t_{i,j}$, che può essere anche nullo. Inoltre T_i è influenzato da un ulteriore elemento e_i che rappresenta errori di misurazione e distanza di trasmissione, cioè vi è la possibilita della presenza di una sorgente di errore.

Ora si può scrivere

$$T_i = e_i + \sum_{j=0}^{w-1} (s_{i,j} + t_{i,j}).$$

Quasi tutti i componenti della sommatoria sono influenzati dal valore dei bit dell'esponente. Per alcuni valori di j gli operandi dell'elevazione al quadrato sono completamente determinati dal valore dei bit $d_{w-1}, ..., d_{j+1}$. Gli operandi usati nel passo della moltiplicazione sono influenzati, oltre che da questi, dal bit j-esimo. Quindi $t_{i,j}$ dipende fortemente da d.

Consideriamo ora il primo ciclo iterativo, l'attaccante Trudy genera due candidati per $s_{i,w-1}$ e $t_{i,w-1}$. Per generare i candidati Trudy firmerà il messaggio M_i sia con l'esponente 0 che con l'esponente 1. Poniamo $\widetilde{T}_{i,w-1,0}$ e $\widetilde{T}_{i,w-1,1}$ i tempi richiesti per le due firme, i primi due indici indicano il messaggio e il ciclo iterativo, l'ultimo indice indica l'ipotesi fatta sul bit.

0	1
$T_1 - \widetilde{T}_{1,w-1,0}$	$T_1 - \widetilde{T}_{1,w-1,1}$
$\vdots$	$\vdots$
$T_k - \widetilde{T}_{k,w-1,0}$	$T_k - \widetilde{T}_{k,w-1,1}$

Tabella 1: Differenze di tempo

Ora ci si costruisce la tabella 1, cioè per ogni messaggio si simula il primo ciclo iterativo per ogni possibile ipotesi sul bit.

Ora calcolando la varianza delle colonne, ponendo $T_{i,0} = T_i - \widetilde{T}_{i,w-1,0}$, $T_{i,1} = T_i - \widetilde{T}_{i,w-1,1}$ ed indicando con $\overline{T}_0$ e $\overline{T}_1$ la media della prima e della seconda colonna,

$$S_0^2 = \frac{1}{k-1}\sum_{i=1}^{k}(T_{i,0} - \overline{T}_0)^2 \quad \text{e} \quad S_1^2 = \frac{1}{k-1}\sum_{i=1}^{k}(T_{i,1} - \overline{T}_1)^2,$$

dove si avrà la varianza più piccola è la colonna con l'ipotesi corretta sul bit.

Per il bit successivo d_{w-2} si procede nello stesso modo, firmando i messaggi con $d_{w-1}0$ e $d_{w-1}1$ e costruendo la tabella con le due colonne.

Analizziamo ora la correttezza di questo procedimento.

Sia ora b un determinato valore di j nell'algoritmo left-to-right e supponiamo che i valori $d_{w-1}, ..., d_{b+1}$ siano corretti.

Trudy si calcola $T_i - \widetilde{T}_{i,b,h}$, dove h è l'ipotesi fatta sul bit e

$$\widetilde{T}_{i,b,h} = \sum_{j>b}(s_{i,j} + t_{i,j}) + (s_{i,b} + \widetilde{t}_{i,b,h}),$$

$\widetilde{t}_{i,b,h}$ è il candidato per $t_{i,b}$. Si noti che se $h = 0$ risulta $\widetilde{t}_{i,b,h} = 0$ altrimenti $\widetilde{t}_{i,b,h} > 0$.

Quindi si ha

$$\begin{aligned} T_i - \widetilde{T}_{i,b,h} &= e_i + \sum_{j=0}^{w-1}(s_{i,j} + t_{i,j}) - \sum_{j>b}(s_{i,j} + t_{i,j}) - (s_{i,b} + \widetilde{t}_{i,b,h}) \\ &= e_i + \sum_{j<b}(s_{i,j} + t_{i,j}) + (t_{i,b} - \widetilde{t}_{i,b,h}). \end{aligned}$$

Nel caso in cui l'ipotesi sul bit sia corretta, allora $t_{i,b} = \widetilde{t}_{i,b,h}$, perciò

$$T_i - \widetilde{T}_{i,b,h} = e_i + \sum_{j<b}(s_{i,j} + t_{i,j}),$$

altrimenti $t_{i,b} \neq \widetilde{t}_{i,b,h}$ e quindi non ci sono cancellazioni.

Consideriamo ora le varie misurazioni di tempo come variabili aleatorie. T sarà la variabile aleatoria relativa al tempo necessario per firmare un messaggio,

s la variabile riguardante il tempo per fare l'elevazione al quadrato in $\mathbb{Z}_N$ e t la variabile relativa alla molteplicazione in $\mathbb{Z}_N$. Inoltre la variabile aleatoria $\widetilde{T}_{b,h}$ è quella che descrive quanto tempo è necessario per firmare M con solo gli $w-b$ bit più significativi dell'esponente con l'ultimo valore posto uguale ad h.

Supponendo che i tempi di esecuzione della moltiplicazione modulare siano indipendenti tra i vari cicli e anche dall'errore si ottiene, nel caso in cui d_b sia corretto,

$$\begin{aligned} Var(T - \widetilde{T}_{b,h}) &= Var\left(e + \sum_{j<b} s + \sum_{\substack{j<b \\ d_j \neq 0}} t\right) \\ &= Var(e) + bVar(s) + lVar(t), \end{aligned}$$

dove l sono i bit non nulli di d tra quelli minori di b.

Nel caso in cui d_b sia errato si ha

$$T_i - \widetilde{T}_{i,b,h} = e_i + \sum_{j<b}(s_{i,j} + t_{i,j}) + (t_{i,b} - \widetilde{t}_{i,b,h})$$

con $t_{i,b}$ o $\widetilde{t}_{i,b,h}$ uguale a zero, allora si avrà

$$Var(T - \widetilde{T}_{b,h}) = Var(e) + bVar(s) + (l+1)Var(t).$$

Perciò una colonna con un'ipotesi corretta sul bit dell'esponente avrà una varianza più bassa di $Var(t)$ rispetto all'altra.

La varianza campionaria, S^2, è una buona approssimazione della varianza effettiva, inoltre è possibile determinare una stima euristica della probabilità che si riesca adeterminare la colonna corretta.

4 Probabilità di riuscita

In questa sezione si daranno per scontati alcuni risultati base della probabilità e della statistica, comunque si rimanda al testo [3] per quanto riguarda tale teoria.

Supponiamo che l'errore sia trascurabile, cioè $Var(e) = 0$. Supponiamo inoltre che le variabili aleatorie s e t abbiano una distribuzine normale, poniamo quindi $s - \mathcal{N}(\mu_s, \sigma_s^2)$ e $t - \mathcal{N}(\mu_t, \sigma_s^t)$, cioè s ha media μ_s e varianza σ_s^2 e analogo t. Essendo la combinazione lineare di variabile aleatorie normalmente distribuite una variabile aleatoria con distribuzione normale, si ha che

$$\sum_{j<b} s + \sum_{\substack{j<b \\ d_j \neq 0}} t \quad \text{e} \quad \sum_{j<b} s + \sum_{\substack{j<b \\ d_j \neq 0}} t + t,$$

le variabili casuali relative alla colonna corrette e a quella errata, hanno distribuzione normale.

Sia $\mathcal{N}(\mu_0, \sigma_0^2)$ la dristribuzione della prima variabile con $\mu_0 = b\mu_s + l\mu_t$ e $\sigma_0^2 = b\sigma_s^2 + l\sigma_t^2$.

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

Consideriamo ora le k misurazioni del tempo delle firme e indichiamo con X e Y le variabili normali standardardizzate corrette ed errate rispettivamente, possiamo allora riscrivere la tabella 1 come mostrato sopra.

Poniamo $V_i = \sigma_0 X_i + \mu_0$ e $W_i = (\sigma_0 X_i + \mu_0) + (\sigma_t Y_i + \mu_t)$, allora

$$\begin{aligned} Pr(S_W^2 > S_V^2) &= Pr\left(\frac{1}{k-1}\sum_{i=1}^{k}(W_i - \overline{W})^2 > \frac{1}{k-1}\sum_{i=1}^{k}(V_i - \overline{V})^2\right) \\ &= Pr\left(\sum_{i=1}^{k}(W_i - \overline{W})^2 > \sum_{i=1}^{k}(V_i - \overline{V})^2\right). \end{aligned}$$

Essendo V e W normalmente distribuite, per valori di k sufficientemente grandi, $\overline{V} \approx \mu_0$ e $\overline{W} \approx \mu_0 + \mu_t$, quindi

$$\begin{aligned} Pr(S_W^2 > S_V^2) &\approx Pr\left(\sum_{i=1}^{k}(\sigma_0 X_i + \sigma_t Y_i)^2 > \sum_{i=1}^{k}(\sigma_0 X_i)^2\right) \\ &= Pr\left(\sum_{i=1}^{k}(\sigma_0^2 X_i^2 + \sigma_t^2 Y_i^2 + 2\sigma_0\sigma_t X_i Y_i) > \sum_{i=1}^{k}(\sigma_0^2 X_i^2)\right) \\ &= Pr\left(\sum_{i=1}^{k}(2\sigma_0\sigma_t X_i Y_i) + \sum_{i=1}^{k}(\sigma_t^2 Y_i^2) > 0\right) = Pr\left(2\sigma_0\sum_{i=1}^{k} X_i Y_i + \sigma_t\sum_{i=1}^{k} Y_i^2 > 0\right). \end{aligned}$$

Ora, essendo X e Y standardizzate, dall'ugualianza $Var(X) = E(X^2) - E(X)^2$, si ha che $E(X^2) = E(Y^2) = 1$, quindi $\sum_{i=1}^{k} Y_i^2 \approx \sum_{i=1}^{k} E\left(Y^2\right) = k$. Inoltre X e Y sono indipendenti quindi $E(XY) = E(X)E(Y) = 0$ e $Var(XY) = E(X^2Y^2) - E(XY)^2 = E(X^2Y^2) = E(X^2)E(Y^2) = 1$.

Per il Teorema del limite centrale segue che la variabile $\sum_{i=1}^{k} X_i Y_i$ ha una distribuzione del tipo $\mathcal{N}(0, k)$. Approssimando quindi $\sum_{I=1}^{k} Y_i^2$ con k, e posto Z variabile aleatoria con distribuzione normale standard, si ha

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

dove $\Phi(z)$ è la curva normale standard.

5 Possibili difese

Ci sono diversi tipi di difese contro il timing attack. Uno dei metodi più ovvi è quello di svolgere il calcolo della firma con algoritmi che lavorino in tempi di calcolo costanti, qualunque siano le operazione veramente necessarie. Tale

metodo può essere comunque vulnerabile agli attacchi, basta infatti controlla re anche il lavoro della CPU durante lo svolgimento dell'algoritmo, inoltre non sono possibili delle ottimizzazioni di calcolo.

Un altro metodo è l'aggiunta di ritardi, errori di misurazioni, casuli, così facendo il numero di messaggi necessari, da far firmare, per determinare l'esponente d cresce in modo esponenziale e rende vano un timing attack. Si può comunque notare che un algoritmo che introduca del rumore sul tempo di esecuzione non può essere ottimizzato nei calcoli.

Il metodo più accettato è quello di usare RSA Blinding. Cioè invece di calcolare direttamente la firma $M^d mod(N)$, si sceglie un numero random r e si calcola $S' = (r^e M)^d mod(N)$. Ora si calcola l'inverso di r, r^{-1}, e si computa $r^{-1}S' = M^d mod(N)$ che è la firma del messaggio. Quindi il tempo utile alla firma del messaggio M non è più correlato al messaggio in input, di conseguenza il timing attack non può ottenere informazioni riagurdo l'esponente d.

6 Conclusioni

Algoritmi di crittografia che si basano sull'esponente modulare come RSA e Diffie-Hellman possono essere vulnerabili ad Timing Attack. Se l'operazione di elevamento a potenza, che coinvolge la chiave segreta, può essere cronometrato da un attaccante con ragionevole accuratezza, la chiave può essere recuperato utilizzando i valori di input, accuratamente selezionati, il cui numero è proporzionale alla lunghezza della chiave.

Kocher attraverso il timing attack riuscì a determinare una chiave di 512-bit di un sistema che implementava RSA attraverso la libreria RSAREF. I sistemi che utilizzano librerie che implementano RSA attraverso il Teorema Cinese dei Resti (CRT) non sono vulnerabili al timing attack di Kocher. David Brunely e Dan Boneh [1] mostrano un attacco ad un sistema che utilizza la libreria OpenSSL, che implementa RSA con CRT per la firma digitale. Questi sono riusciti a determinare uno dei fattori del modulo RSA e quindi da questo a determinare la chiave privata.

Quindi gli attacchi di tipo Side Channel Analysis ci dicono che anche se un sistema è matematicamente forte, in pratica, la sicurezza dipende anche dalla modalità di implementazione del sistema.

La crittografia non deve essere esaminata in maniera isolata. La progettazione di un sistema sicuro dovrebbe comprendere ogni aspetto, crittografici e non.

Il Timing Attack dimostra che gli aggressori non seguono necessariamente le regole del "gioco", si presume che si attacca sempre l'anello più debole di un sistema.

Riferimenti bibliografici

[1] D.Brumley, D.Boneh, *Remote timingattacks are practical.*Proceedings of the 12th USENIX Security Symposium, pp. 114.

[2] J. F. Dhem, F. Koeune, P. A. Leroux, P. Mestre´, J. J. Quisquater and J. L. Willems, *A Practical Implementation of the Timing Attack*, Proceedings of CARDIS, Settembre 1998.

[3] P. Erto, *Probabilita' e Statistica per le scienze e l'ingegneria.* McGraw-Hill ed., 2004.

[4] P. Kocher, *Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems.* Advances in Cryptology, pp. 104–113, Febbraio 1996.

[5] R.L.Rivest,A.Shamir,L.Adleman, *A method for obtaining digital signatures and public-key cryptosystems.* Comm. ACM 21 no. 2, pp. 120-126, 1978.

[6] W. Schindler, *A timing attack against RSA with the chinese remainder theorem.* In CHES 2000, pp. 109–124, 2000.

[7] B. Schneier, *Risks of relying on cryptography*, Inside Risks 112, Comm. of the ACM , vol. 42, no. 10, Ottobre 1999.