Updated Recommendations for Blinded Exponentiations vs. Single Trace Analysis

COSADE Workshop - Paris, 7 March 2013.

Christophe Clavier
XLIM-CNRS
Limoges University, France

Benoit Feix
UL Security Lab, UK
XLIM, Limoges University, France
Work done when author was with Inside Secure

Agenda

Exponentiation and side-channels

Chosen message scenario

Relaxed side-channel leakage models

Countermeasures

Conclusion

Exponentiation and side-channel

Some previous publications …

- 1996 – Kocher et al.: simple side-channel analysis (SSCA)
- 1999 – Messerges : differential side-channel analysis (DSCA)
- 2001 – Walter: Big-Mac Attack
- 2005 – Yen et al.: chosen messages on protected exponentiations
- 2010 – Courrège et al.: SSCA study on blinded exponentiation

- Not an exhaustive list …

Notations

- $x = (x_{l\text{-}1}, \ldots, x_0)_b$ x decomposition in base b (t-bit words)
- LIM(x,y): Long Integer Multiplication $x \times y$
- BarrettRed(a,n): Barrett modular reduction a mod n
- ModMul(x,y,n) = BarrettRed(LIM(x,y),n)

Exponentiation

Algorithm 2.1 Long Integer Multiplication

Require: $x = (x_{\ell-1}x_{\ell-2}\dots x_1x_0)_b, y = (y_{\ell-1}y_{\ell-2}\dots y_1y_0)_b$
Ensure: multiplication result $\mathsf{LIM}(x,y) = x \cdot y$

for $i = 0$ **to** $2\ell - 1$ **do**
  $w_i \leftarrow 0$
for $i = 0$ **to** $\ell - 1$ **do**
  $c \leftarrow 0$
  for $j = 0$ **to** $\ell - 1$ **do**
    $(uv)_b \leftarrow w_{i+j} + x_j \cdot y_i + c$
    $w_{i+j} \leftarrow v$ and $c \leftarrow u$
  $w_{i+\ell} \leftarrow c$
return w

Algorithm 2.2 Exponentiation

Require: integers m and n with $m < n$, k-bit exponent $d = (d_{k-1}d_{k-2}\dots d_1d_0)_2$
Ensure: $\mathsf{Exp}(m,d,n) = m^d \bmod n$

$R_0 \leftarrow 1$; $R_1 \leftarrow m$
for $i = k - 1$ **down to** 0 **do**
  $R_0 \leftarrow \mathsf{ModMul}(R_0, R_0, n)$
  if $d_i = 1$ **then** $R_0 \leftarrow \mathsf{ModMul}(R_0, R_1, n)$
return R_0

Blinded Exponentiation

Algorithm 2.3 Blinded exponentiation

Require: integers m and n with $m < n$, $\ell \cdot t$-bit exponent $d = (d_{\ell \cdot t-1} d_{\ell \cdot t-2} \dots d_1 d_0)_2$, a security parameter λ

Ensure: $\mathsf{Exp}(m,d,n) = m^d \bmod n$

$r_1 \leftarrow random(1, 2^\lambda - 1)$

$r_2 \leftarrow random(1, 2^\lambda - 1)$

$R_0 \leftarrow 1 + r_1 \cdot n \bmod r_2 \cdot n$

$R_1 \leftarrow m + r_1 \cdot n \bmod r_2 \cdot n$

$i \leftarrow \ell \cdot t - 1$; $\alpha \leftarrow 0$

while $i \geqslant 0$ **do**

  $R_i \leftarrow \mathsf{ModMul}(R_0, R_\alpha, n)$

  $\alpha \leftarrow \alpha \oplus d_i$;

  $i \leftarrow i - 1 + \alpha$

return R_0

- Loop operation : atomicity principle from Chevallier-Mames et al.
- Additive message blinding
- Exponent message blinding

$d^* = d + r.\varphi(n)$ (r : λ-bit random)

→ not useful here as our analysis focuses on a single trace

Side Channel Leakage on Multiplier
First leakage model

[A_0] A null word $x_i = 0$ in some operand x (a so-called *tag*) provokes a particularly visible leakage during LIM(x,y).

For atomic left-to-right exponentiation, a tag on the message m can leak on every LIM(a,m) which reveals the secret exponent d.

Study done by Courrège et al. on random messages

→ leakage probability were given depending on multiplier base bit size t,

→ showed bias in $u = r_1 \bmod r_2$ in additive message blinding

$m^* \leftarrow m + u.n$ when r_1 and r_2 are chosen both randomly.

Agenda

Exponentiation and side-channels

Chosen message scenario

Relaxed side-channel leakage models

Countermeasures

Conclusion

Chosen Message Scenario

- It is possible to choose m such that some particular word m^*_i is tagged whenever u takes some specific value $u^{(i)}$.
- It is even possible to simultaneously target l different random values $u^{(i)}$

  m^*_0 is tagged for $u^{(0)}$

  m^*_1 is tagged for $u^{(1)}$

  …

  m^*_{l-1} is tagged for $u^{(l-1)}$

- This increases the probability for a blinded message m* to be tagged.

Chosen Message Scenario

- How to target simultaneously many random values $u^{(i)}$ on message m^*

Algorithm 3.1 Chosen message construction

Require: a ℓ-word modulus n and a set $\left(u^{(0)}, \ldots, u^{(\ell-1)}\right)$ of targeted randoms

Ensure: a message m whose randomization is tagged for any specified target

$m \leftarrow 0$
for $i = 0$ **to** $\ell - 1$ **do**
  $s^{(i)} \leftarrow u^{(i)}\, n$
  $\mu \leftarrow -\left\lfloor \frac{\overline{s^{(i)}_i} + \underline{m_i}}{b^i} \right\rfloor \bmod b$
  $m \leftarrow m + \mu\, b^i$
return m

$$\begin{aligned} \overline{x_i} &= x \bmod b^{i+1} = (x_i \ldots x_1 x_0)_b \\ \underline{x_i} &= x \bmod b^i = (x_{i-1} \ldots x_1 x_0)_b \quad \text{with} \quad \underline{x_0} = 0 \end{aligned}$$

Chosen Message Scenario

- $\mathrm{Tag}^{(i)}(m^*)$ occurs either if $u=u^{(i)}$ or by pure chance on a t-bit word
- $\mathrm{Proba}(\mathrm{tag}^{(i)}(m^*))$ $= \mathrm{Proba}(u=u^{(i)}) + 2^{-t}$

  $= 2^{-\lambda} + 2^{-t}$

  $\approx \max(2^{-\lambda}, 2^{-t})$
- m^* is tagged whenever it is tagged on any of its words m^*_i.
- $\mathrm{Proba}(\mathrm{tag}(m^*))$ $\approx l.\max(2^{-\lambda}, 2^{-t})$
- If random bit-length is lower than base length we gain factor $2^{t-\lambda}$
- Optimal blinding requires $\lambda = t$.
- If r_1 and r_2 are uniformly distributed, then smaller u values are more probable and one should preferably choose $u^{(i)}=i$
- Gain a factor 21 for the tag probability for $\lambda = 32$, $t = 64$, (1024 bits).

Simulation results

- Simulation results of the chosen message attack for a 1024-bit RSA modulus with biased randomization.

		Tag probability		Number of traces		Gain ω	
		Simu	Theory	Simu	Theory	Simu	Theory
$\lambda = 8$ (10^6 runs)	$t = 16$	$6.50\,10^{-1}$	$6.51\,10^{-1}$	1.54	1.54	2.60	2.60
	$t = 32$	$4.28\,10^{-1}$	$4.28\,10^{-1}$	2.33	2, 33	3.43	3.43
	$t = 64$	$2.63\,10^{-1}$	$2.62\,10^{-1}$	3.80	3.81	4.21	4.20
$\lambda = 16$ (10^7 runs)	$t = 16$	$8.30\,10^{-3}$	$8.30\,10^{-3}$	121	121	8.50	8.50
	$t = 32$	$4.49\,10^{-3}$	$4.48\,10^{-3}$	223	223	9.19	9.18
	$t = 64$	$2.42\,10^{-3}$	$2.41\,10^{-3}$	414	415	9.89	9.86
$\lambda = 24$ (10^8 runs)	$t = 16$	—	—	—	—	—	—
	$t = 32$	$2.77\,10^{-5}$	$2.81\,10^{-5}$	36062	35590	14.5	14.7
	$t = 64$	$1.48\,10^{-5}$	$1.47\,10^{-5}$	67476	68049	15.5	15.4
$\lambda = 32$ (10^9 runs)	$t = 16$	—	—	—	—	—	—
	$t = 32$	—	—	—	—	—	—
	$t = 64$	$8.3\,10^{-8}$	$7.78\,10^{-8}$	$12.0\,10^6$	$12.8\,10^6$	22.3	20.9

Instead of $8.7\ 10^{-19}$ in random message scenario. ($1.15\ 10^{18}$ traces)

Agenda

Exponentiation and side-channels

Chosen message scenario

Relaxed side-channel leakage models

Countermeasures

Conclusion

Relaxed side-channel leakage models

- Previous leakage model was:
- [A_0] : side-channel tag originates when a whole *t*-bit word equals zero in the operand *m*.

- We consider two less restrictive but realistic leakage models
- [A_1] : side-channel tag originates from the fact that at least τ consecutive bits in a *t*-bit word of *m* are set to zero, with $\tau < t$.
- [A_2] : side-channel tag originates from the fact that the Hamming weight *h* of the *t*-bit word is lower than a value ν, with $h \leq \nu < t$.

Relaxed side-channel leakage models

- Probability for an operand *m* to be tagged is:

  $$\text{Proba}(\text{tag}(m)) = 1 - (1-p)^l \approx l.p$$

  where *p* is the probability that a word is tagged.

- Model [A_1] (consecutive zeros)
  - Exhaust n_τ the number of existing *t*-bit words with their longest consecutive zero sequence being of length τ .
  - $p_1(t, \tau) = n_\tau 2^{-t}$ the probability for a *t*-bit word to have its longest sequence of consecutive zero bit to be exactly τ .
  - $p = \sum_{i=\tau}^{t} p_1(t, i)$

Relaxed side-channel leakage models [A_1]
Examples

τ	t-bit word number	$p_1(t,\tau)$	$P(tag^{(i)}_{A_1}(x))$	$P(tag_{A_1}(m_{512}))$	$P(tag_{A_1}(m_{1024}))$	$P(tag_{A_1}(m_{2048}))$
0	1	$2.33\,10^{-10}$	1	1	1	1
8	111246728	$2.59\,10^{-02}$	$5.02\,10^{-02}$	$5.61\,10^{-01}$	$8.08\,10^{-01}$	$9.63\,10^{-01}$
16	311296	$7.25\,10^{-05}$	$1.37\,10^{-04}$	$2.20\,10^{-03}$	$4.39\,10^{-03}$	$8.75\,10^{-03}$
24	704	$1.64\,10^{-07}$	$2.98\,10^{-07}$	$4.77\,10^{-06}$	$9.54\,10^{-06}$	$1.91\,10^{-05}$
32	1	$2.33\,10^{-10}$	$2.33\,10^{-10}$	$3.73\,10^{-09}$	$7.45\,10^{-09}$	$1.49\,10^{-08}$

Table 3. [A_1] Leakage probability examples for some τ values when $t = 32$

- Probability a 1024-bit integer is tagged reduced from $7{,}45.10^{-9}$ to $4{,}39.10^{-3}$ from model [A_0] to model [A_1] with $\tau = 16$.
- Then 1480 messages are required instead of $8{,}73.10^{8}$ for attack success probability at 0.999.

Relaxed side-channel leakage model [A_2]

- Model [A_2] (small Hamming weight)

  - $p_2(t, \mu) = \binom{\mu}{t} . 2^{-t}$ the probability that a t-bit word has its Hamming weight equal to μ.
  - $p = \sum_{\mu=0}^{\nu} p_2(t, \mu)$

Relaxed side-channel leakage models [A_2]

ν	t-bit word number	$p_2(t,\nu)$	$P(tag^{(i)}_{A_2}(x))$	$P(tag_{A_2}(m_{512}))$	$P(tag_{A_2}(m_{1024}))$	$P(tag_{A_2}(m_{2048}))$
0	1	$2.33\,10^{-10}$	$2.33\,10^{-10}$	$3.73\,10^{-09}$	$7.45\,10^{-09}$	$1.49\,10^{-08}$
4	35960	$8.37\,10^{-06}$	$9.65\,10^{-06}$	$1.54\,10^{-04}$	$3.09\,10^{-04}$	$6.17\,10^{-04}$
8	10518300	$2.45\,10^{-03}$	$3.50\,10^{-03}$	$5.46\,10^{-02}$	$1.06\,10^{-01}$	$2.01\,10^{-01}$
16	601080390	$1.40\,10^{-01}$	$5.70\,10^{-01}$	1	1	1
24	10518300	$2.45\,10^{-03}$	$9.99\,10^{-01}$	1	1	1
32	1	$2.33\,10^{-10}$	1	1	1	1

Table 5. [A_2] Leakage probability for some ν values when $t = 32$

- Probability a 1024-bit integer is tagged reduced from $7.45\ 10^{-9}$ to $3.09\ 10^{-4}$ from model [A_0] to model [A_2] with $\nu = 4$.
- Then $2.1\ 10^{4}$ messages are required instead of $8.73\ 10^{8}$ for attack success probability at 0.999.

Comparison example

τ, ν	t-bit word number	p	$P(tag_{A_i}(m_{512}))$	$P(tag_{A_i}(m_{1024}))$	$P(tag_{A_i}(m_{2048}))$
$[A_2]$ $\nu = 4$	$8.37\ 10^{-06}$	$9.65\ 10^{-06}$	$1.54\ 10^{-04}$	$3.09\ 10^{-04}$	$6.17\ 10^{-04}$
$[A_1]$ $\tau = 16$	$7.25\ 10^{-05}$	$1.37\ 10^{-04}$	$2.20\ 10^{-03}$	$4.39\ 10^{-03}$	$8.75\ 10^{-03}$
$[A_0]$	$2.33\ 10^{-10}$	$2.33\ 10^{-10}$	$3.73\ 10^{-09}$	$7.45\ 10^{-09}$	$1.49\ 10^{-08}$

Table 6. Leakage probability examples for t=32

τ, ν	m_{512}	m_{1024}	m_{2048}
$[A_2]$ $\nu = 4$	$4.22\ 10^4$	$2.11\ 10^4$	$1.06\ 10^3$
$[A_1]$ $\tau = 16$	$3\ 10^3$	$1.5\ 10^3$	750
$[A_0]$	$1.75\ 10^9$	$8.73\ 10^8$	$4.37\ 10^8$

Table 7. Number of messages/executions needed for leakage probability at $0,999$, for t=32

Agenda

Exponentiation and side-channels

Chosen message scenario

Relaxed side-channel leakage models

Countermeasures

Conclusion

Countermeasures

- Evaluate precisely the leakage characteristics of the hardware multiplier
  - Determine τ and ν for both leakage models [A_1] and [A_2] and leakage probabilities
- Practical results on an IC will also depends on
  - The efficiency of the hardware countermeasures present in the device
  - Signal processing capabilities
- Prefer right-to-left to left-to-right algorithms for the implementation
- And\or apply new randomization on message after each modular multiplication

Agenda

Exponentiation and side-channels

Chosen message scenario

Relaxed side-channel leakage models

Countermeasures

Conclusion

Conclusion

- We have given a chosen message attack improvement which justifies to choose $\lambda = t$ on blinded exponentiations.
- We evaluated attack efficiency in two relaxed but realistic leakage models.
- It justifies the need for a precise leakage characterization of hardware multipliers.

Thanks for your attention …