A DPA Attack Against the Modular Reduction within a CRT Implementation of RSA

Bert den Boer, Kerstin Lemke, Guntram Wicke
T-Systems ISS GmbH

Presentation at CHES 2002

Contents

DPA against a CRT Implementation of RSA

RSA Cryptosystem

- Secret Primes p and q
- Public Modulus N with $N = pq$
- Public Exponent e
- Secret Exponent d with $ed \equiv 1 (\mathrm{mod}\ \mathrm{lcm}(\mathrm{p}-1, \mathrm{q}-1))$

- Decryption (RSA Decryption, RSA Signing):

$$y = x^d \bmod N$$

- Encryption

$$x = y^e \bmod N$$

DPA against a non-CRT Implementation of RSA
Square Multiply Algorithm

- 'Top-down Square Multiply' Algorithm to perform

$$c = a^b \bmod m \text{ in } \mathbb{Z}_m$$

$$b = [b_{n-1} b_{n-2} \cdots b_1 b_0]$$

```
c := 1
for k := n-1 down to 0 do {
  c := c*c mod m
  if b[k]=1 then c := c*a mod m
}
return c
```

DPA against a non-CRT Implementation of RSA
Approach for DPA Attack against the Exponent

- Key hypotheses H(j)
  - Guesses on next exponent bits or
  - Guesses on next modular operations
- Selection Functions d(x,j)
  - n-bit Hamming weight W(x) of predicted intermediate data x for each key hypothesis H(j):

$$d(x, j) = W(x, j) - E(n)$$

- Correlation between $d(x_i, j)$ and Power Consumption $P(x_i, t)$
  - Absolute maximum of correlation coefficient identifies the correct key value j

DPA against a CRT Implementation of RSA
CRT Algorithm (Garner)

- Split exponent

$$d_p = d \bmod (p-1) \qquad d_q = d \bmod (q-1)$$

- Perform 2 exponentiations:

$$v_1 = x^{d_p} \bmod p \qquad v_2 = x^{d_q} \bmod q$$

- Using $P_q = p^{-1} \bmod q$

- Calculate

```
u := (v2-v1)*Pq mod q
y := v1+u*p
return y
```

DPA Attack against a CRT Implementation of RSA
Main Idea

- The remainder r_0 of an input value x_0 modulo a secret prime q is successively attacked by DPA

$$r_0 = x_0 \bmod q$$

- The gcd of $(x_0 - r_0)$ and the public RSA modulus N = p q gives the prime q

$$q = gcd(x_0 - r_0, N)$$

DPA Attack against a CRT Implementation of RSA
General Approach

- MRED: Modular Reduction on Equidistant Data
- Use of equidistant input data x_i at each k measurement series:

$$x_i = x_0 - i \cdot (256)^k$$

- Each measurement series k compromises the k-th byte of the remainder r_0 (k=0: least significant byte of r_0)

$$F_k = r_0 \bmod (256)^k$$

$$F_k = \sum_{i=0}^{k-1} f_i \cdot (256)^i$$

DPA Attack against a CRT Implementation of RSA
Hypotheses on the Remainder

$$H_{ji} \text{ is } \left\{ (r_i \bmod (256)^{k+1}) \operatorname{div} (256^k) = (j-i) \bmod 256 \right\}.$$

H_{ji}	x_0	x_1	x_2	x_3	x_4	$\cdots$	x_i
H_{0i}	0	255	254	253	252	$\cdots$	$-i \bmod 256$
H_{1i}	1	0	255	254	253	$\cdots$	$(1-i) \bmod 256$
H_{2i}	2	1	0	255	254	$\cdots$	$(2-i) \bmod 256$
.....	...	...	...	...	...	..	....
H_{255i}	255	254	253	252	251	$\cdots$	$(255-i) \bmod 256$

DPA Attack against a CRT Implementation of RSA
Selection Function

- The selection function d(x,j) is based on 8 bit Hamming weight.

d_{ji}	x_0	x_1	x_2	x_3	x_4	$\cdots$	x_i
d_{0i}	0	8	7	7	6	$\cdots$	$W(H_{0i})$
d_{1i}	1	0	8	7	7	$\cdots$	$W(H_{1i})$
d_{2i}	1	1	0	8	7	$\cdots$	$W(H_{2i})$
....	..	..	..	..	..	..	....
d_{255i}	8	7	7	6	7	$\cdots$	$W(H_{255i})$

DPA Attack against a CRT Implementation of RSA
Successive Approximation

- Check for each measurement series k that

$$gcd(x_0 - F_k - i \cdot (256)^k, N) \stackrel{!}{=} 1.$$

- If the gcd is 1
  => Run DPA on measurement series k to compromise f_k

- else
  => The modulus N is factorized by the gcd (end criterion).

DPA Attack against a CRT Implementation of RSA

Results using simulated measurement data 1/3

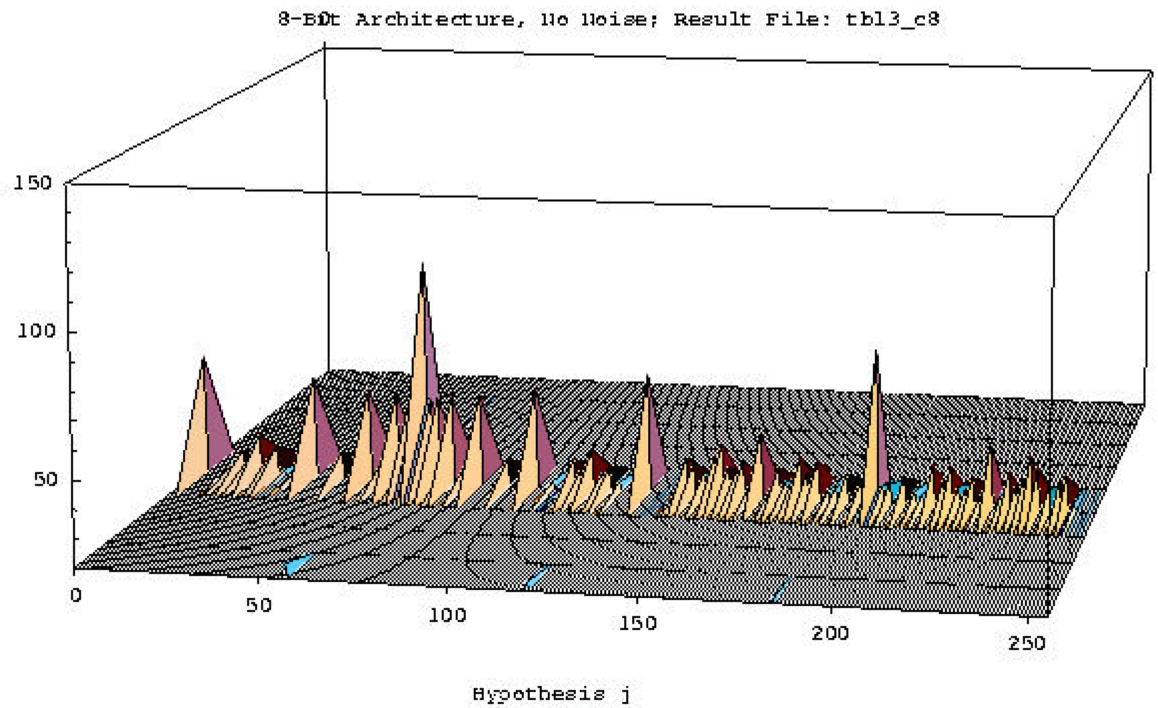

Fig. 1. Graphical representation of the absolute correlation coefficients on the base of 256 single measurements. Correlations coefficients $c(j,t) < |0.2|$ are neglected in this trace for clarity reasons.

DPA Attack against a CRT Implementation of RSA
Results using simulated measurement data 2/3

Hypothesis	Correlation Coefficient	Relative Displacement of f_0
66	+1.000000	0
194	+0.750000	+128
2	+0.625000	-64
130	+0.625000	+64
34	+0.562500	-32
98	+0.562500	+32
50	+0.531250	-16
82	+0.531250	+16
58	+0.515625	-8
74	+0.515625	+8
62	+0.507812	-4
70	+0.507812	+4
64	+0.503906	-2
68	+0.503906	+2
65	+0.501953	-1
67	+0.501953	+1

DPA Attack against a CRT Implementation of RSA
Results using simulated measurement data 3/3

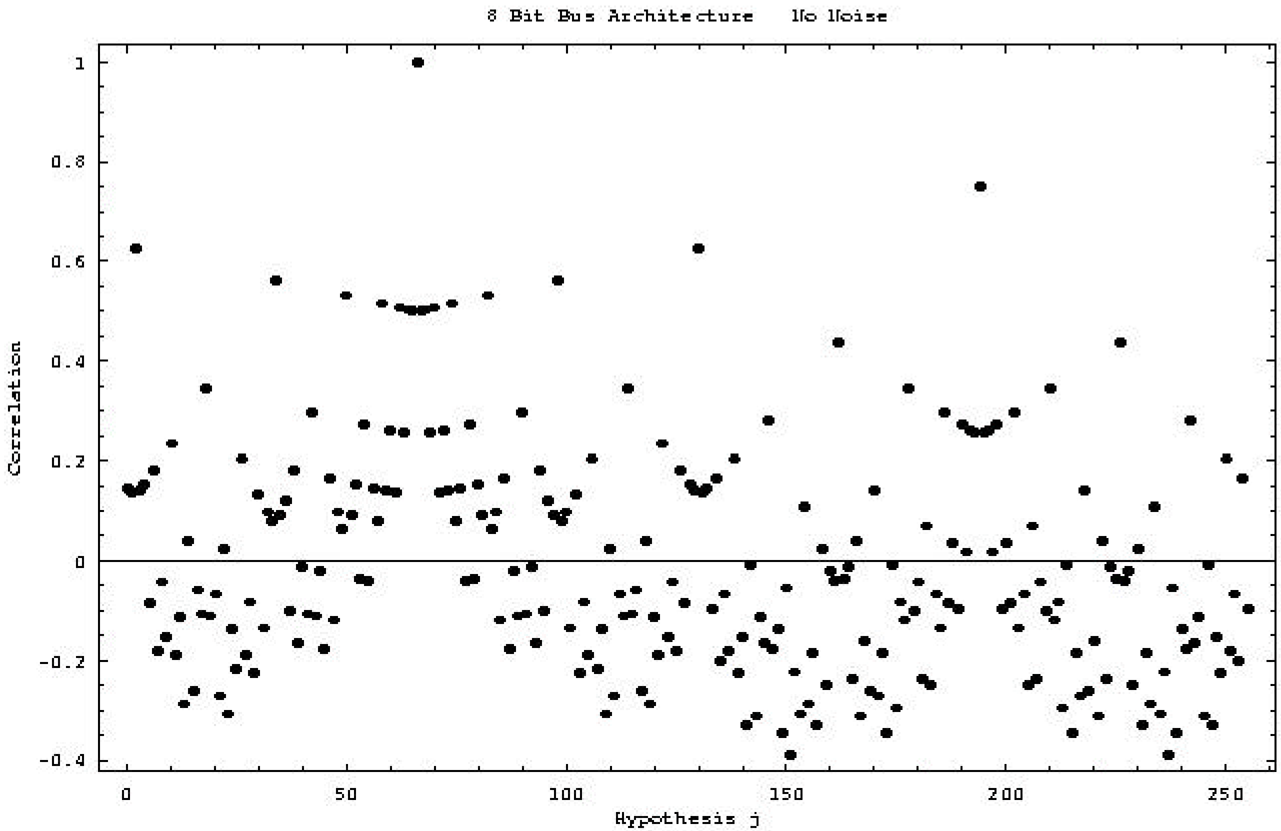

Fig. 2. Graphical representation of the correlation coefficients on the base of 256 single measurements. The smaller correlation amplitudes around $f_0 \pm 128$ of Fig. 1 turned out to be mainly of negative sign.

DPA Attack against a CRT Implementation of RSA
Attack Efforts against 1024 Bit RSA Key

Attack Tasks of MRED	
No. of Measurement Series:	60-62
No. of Single Measurements per Series:	500 - 5000
Single Measurement Data Size:	small
Overall Measurement Time:	1 day to 3 weeks
Overall Re-Synchronisation Time:	few hours to 2 days
No. of DPA calculations:	60-62
Overall DPA calculation time:	few hours to 1 day
Overall Time:	2 days to 1 month

Table 3. Summary of the Attack Efforts needed for a 1024 bit RSA key

DPA Attack against a CRT Implementation of RSA
Limitations and Countermeasures

- Basic Assumptions for MRED:
  1. High number of single measurements
  2. Variation of input data is equidistant
  3. Equidistant Variation of the input data results in equidistant variation of the remainder

- Countermeasures
  1. Usage counters / Failure Counters (RSA Decryption only)
  2. Padding Formats (RSA Signing only)
  3. Destroy

$$(x_0 - i \cdot (256)^k) \bmod q = r_0 - i \cdot (256)^k$$

e. g. by multiplicative message blinding

DPA Attack against a CRT Implementation of RSA
Conclusion

- A new DPA attack has been presented that compromises a secret prime at the modular reduction step of a CRT implementation.

- The moral is

  – to secure the reduction modulo a secret prime and to destroy the basic assumption of MRED:

$$(x_0 - i \cdot (256)^k) \bmod q = r_0 - i \cdot (256)^k$$