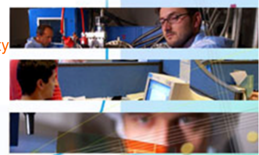
micro and nanoelectronics
microsystems
ambient intelligence
image chain
biology and health

Public Key Perturbation of Randomized RSA Implementations

A. Berzati, C. Dumas & L. Goubin

CEA-LETI Minatec & Versailles St Quentin University



leti

cea

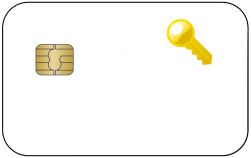
Outline

Outline

1. Introduction
   - Differential Fault Analysis
   - RSA Public Key Perturbations
2. Public Key Perturbation Against R2L Implementations
3. Application to Randomized RSA Implementations
4. Conclusion

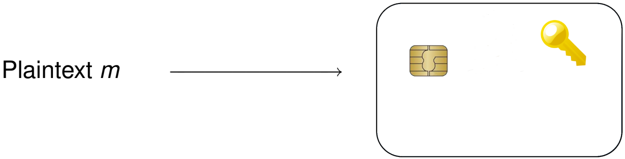
Differential Fault Analysis (DFA) [BS97]

- Perturbation of an electronic device behavior
  - Supply voltage, clock or temperature variations
  - White light, ion or laser beams



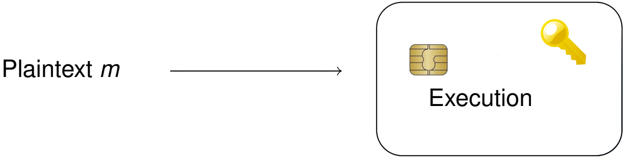
Differential Fault Analysis (DFA) [BS97]

- Perturbation of an electronic device behavior
  - Supply voltage, clock or temperature variations
  - White light, ion or laser beams



Differential Fault Analysis (DFA) [BS97]

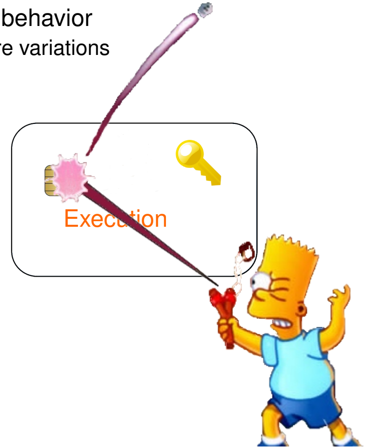
- Perturbation of an electronic device behavior
  - Supply voltage, clock or temperature variations
  - White light, ion or laser beams



Differential Fault Analysis (DFA) [BS97]

- Perturbation of an electronic device behavior
  - Supply voltage, clock or temperature variations
  - White light, ion or laser beams

Plaintext m $\longrightarrow$

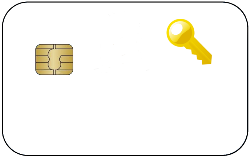


Differential Fault Analysis (DFA) [BS97]

- Perturbation of an electronic device behavior
  - Supply voltage, clock or temperature variations
  - White light, ion or laser beams

Plaintext m $\longrightarrow$

Ciphertext $\hat{C}$ $\longleftarrow$



Differential Fault Analysis (DFA) [BS97]

Analysis of the faulty output

- Identification of the perturbation
- Choice of a fault model
- Differentiation of correct and faulty outputs

$$\Delta_{\hat{C},C} = f(\varepsilon, k)$$

Differential Fault Analysis (DFA) [BS97]

Analysis of the faulty output

- Identification of the perturbation
- Choice of a fault model
- Differentiation of correct and faulty outputs

$$\Delta_{\hat{C},C} = f(\varepsilon, k)$$

Applications to implementations of cryptosystems

- Symmetric: DES [BS97], AES [DLV03], [HS04] ...
- Asymmetric: RSA [BDL97], RSA-CRT [BDL97], ...
- Stream ciphers: RC4 [HS04, BGN05], A5/1 [GKW05], Grain-128 [BCC+09], ...

Outline

RSA Signature scheme

Key Generation

- Pick large primes $\mathbf{p}$ and $\mathbf{q}$ and compute $\mathbf{N} = \mathbf{p} \cdot \mathbf{q}$
- Pick a random $\mathbf{e}$ such that $\gcd(\mathbf{e}, \varphi(N)) = 1$
- Compute $\mathbf{d} \equiv \mathbf{e}^{-1} \bmod \varphi(N)$
- The public key is $(\mathbf{e}, \mathbf{N})$
- The private key is $\mathbf{d}$

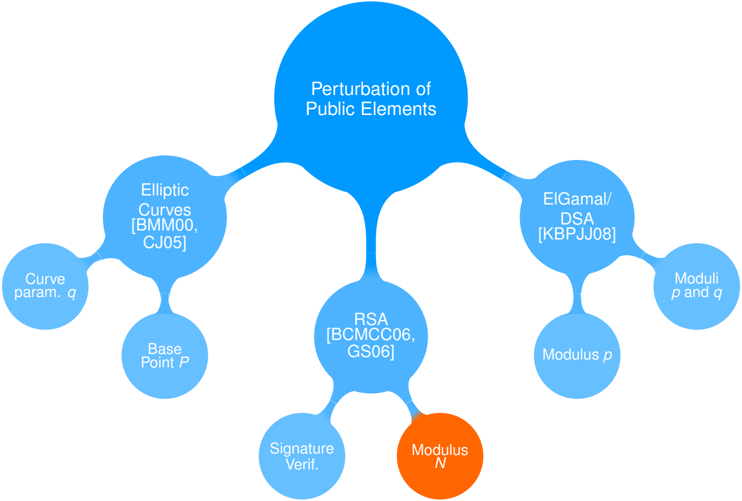
Signature

- Compute $\mathbf{S} \equiv h(m)^{\mathbf{d}} \bmod \mathbf{N}$
- Return $(\mathbf{S}, m)$

Verification

- Check that $\mathbf{S}^{\mathbf{e}} \equiv h(m) \bmod \mathbf{N}$

Previous Work



Previous Work for RSA

- *"Why one should also secure RSA Public Key Elements"*
  E. Brier *et al.*, CHES'06 [BCMCC06]
  - Provide a full private key extraction
  - The modulus is modified before the modular exponentiation:

$$\hat{S} = h(m)^{d} \bmod \hat{N}$$

  - Countermeasure: Exponent randomization

Previous Work for RSA

- *"Why one should also secure RSA Public Key Elements"*
  E. Brier *et al.*, CHES'06 [BCMCC06]
  - Provide a full private key extraction
  - The modulus is modified before the modular exponentiation:

$$\hat{S} = h(m)^{\mathbf{d}} \bmod \hat{N}$$

  - Countermeasure: Exponent randomization

Our contributions

- CHES'08 [BCG08]: Exploit faults on the modulus that occur during a *"Right-To-Left"* modular exponentiations
- CT-RSA'09 [BCDG09]: Generalization to *"Left-To-Right"* modular exponentiations

Outline

Fault Model

Fault model

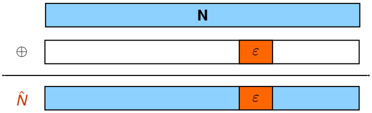
- Transient random byte modification of **N**
- Perturbation of a modular square t bits before the end of the exponentiation
- Time location of the fault known by the attacker

Fault Model

Fault model

- Transient random byte modification of **N**
- Perturbation of a modular square t bits before the end of the exponentiation
- Time location of the fault known by the attacker

- Illustration of a faulty modulus **N**



where ε is a random byte value

Example of Faulty Execution

(N, e) and **d**

(N, e)

"*Right-to-Left*" Algorithm

Input: m, d, N
Output: $A = m^d \bmod N$

```
A:=1;
B:=m;
for i from 0 upto (n − 1)
    if (d_i == 1)
        A := (A · B) mod N;
    end if
    B := B^2 mod N;
end for
return A;
```

Example of Faulty Execution

$(\mathbf{N}, \mathbf{e})$ and $\mathbf{d}$

$(\mathbf{N}, \mathbf{e})$

"Right-to-Left" Algorithm

Input: m, d, N
Output: $A = m^d \bmod N$

```
A:=1;
B:=m;
for i from 0 upto (n − 1)
    if (d_i == 1)
        A := (A · B) mod N;
    end if
    B := B^2 mod N̂;
end for
return A;
```

Faulty RSA Signature

- Fault injection:
  $A_i = A_{i-1} \cdot B_{i-1}{}^{d_i - 1} \bmod \mathbf{N}$ et
  $B_i = B_{i-1}^2 \bmod \hat{N}$

Example of Faulty Execution

($\mathbf{N}, \mathbf{e}$) and $\mathbf{d}$

($\mathbf{N}, \mathbf{e}$)

"Right-to-Left" Algorithm

Input: m, d, N
Output: $A = m^d \bmod N$

```
A:=1;
B:=m;
for i from 0 upto (n − 1)
    if (d_i == 1)
        A := (A · B) mod N̂;
    end if
    B := B^2 mod N̂;
end for
return A;
```

Faulty RSA Signature

- Fault injection:
  $A_i = A_{i-1} \cdot B_{i-1}{}^{d_i - 1} \bmod \mathbf{N}$ et
  $B_i = B_{i-1}^2 \bmod \hat{N}$
- Subsequent iteration:
  $A_{i+1} = A_i \cdot B_i{}^{d_i} \bmod \hat{N}$ et
  $B_{i+1} = B_i{}^2 \bmod \hat{N}$

Example of Faulty Execution

$(\mathbf{N}, \mathbf{e})$ and $\mathbf{d}$

$\left(m, \hat{S}\right) \longrightarrow$

$(\mathbf{N}, \mathbf{e})$

"Right-to-Left" Algorithm

Input: m, d, N
Output: $A = m^d \bmod N$

```
A:=1;
B:=m;
for i from 0 upto (n − 1)
    if (d_i == 1)
        A := (A · B) mod N̂;
    end if
    B := B^2 mod N̂;
end for
return A;
```

Faulty RSA Signature

- Fault injection:
  $A_i = A_{i-1} \cdot B_{i-1}{}^{d_{i-1}} \bmod \mathbf{N}$ et
  $B_i = B_{i-1}^2 \bmod \hat{N}$
- Subsequent iteration:
  $A_{i+1} = A_i \cdot B_i{}^{d_i} \bmod \hat{N}$ et
  $B_{i+1} = B_i{}^2 \bmod \hat{N}$
- Returned faulty signature:

$$\hat{S} = (A_i \bmod N) \cdot m^{d_{[t]}} \bmod \hat{N}$$

Outline

1 Introduction

2 Public Key Perturbation Against R2L Implementations

- Attack Model
- Fault Analysis

3 Application to Randomized RSA Implementations

4 Conclusion

Private Key Recovery from Faulty RSA Signatures

Faulty RSA signature analysis

- A part of the private key $d_{[t]}$ is isolated
- Recovery of $d_{[t]}$ and $\hat{N}$ from the pair $\left(S, \hat{S}\right)$ and the fault model
  $\Rightarrow$ *Guess-and-determine* approach
- The right pair is found with high probability

Private Key Recovery from Faulty RSA Signatures

Faulty RSA signature analysis

- A part of the private key $d_{[t]}$ is isolated
- Recovery of $d_{[t]}$ and $\hat{N}$ from the pair $\left(S, \hat{S}\right)$ and the fault model
  $\Rightarrow$ *Guess-and-determine* approach
- The right pair is found with high probability

Extraction of the private key

1. Gather sufficiently many signatures faulted at different steps
2. Repeat the previous analysis by using the knowledge of already found key parts.
3. Extract the missing bits using mathematical methods

Private Key Recovery from Faulty RSA Signatures

Faulty RSA signature analysis

- A part of the private key $d_{[t]}$ is isolated
- Recovery of $d_{[t]}$ and $\hat{N}$ from the pair $\left(S, \hat{S}\right)$ and the fault model
  $\Rightarrow$ *Guess-and-determine* approach
- The right pair is found with high probability

Extraction of the private key

1. Gather sufficiently many signatures faulted at different steps
2. Repeat the previous analysis by using the knowledge of already found key parts.
3. Extract the missing bits using mathematical methods

Key extraction on a PC for a 1024-bit RSA

- 250 faulty signatures
- A few dozen minutes for the analysis

Outline

Exponent Randomization

- Proposed by P. Kocher in 1996 [Koc96], formalized by J.S. Coron at CHES'99 [Cor99] to defeat side channel attacks
- Based on Fermat's theorem:

$$m^{\varphi(N)} \equiv 1 \bmod N$$

Exponent Randomization

- Proposed by P. Kocher in 1996 [Koc96], formalized by J.S. Coron at CHES'99 [Cor99] to defeat side channel attacks
- Based on Fermat's theorem:

$$m^{\varphi(N)} \equiv 1 \bmod N$$

RSA Exponent Randomization Algorithm

Input: $\dot{m}, N, \varphi(N), d$ and the length l
Output: $S = \dot{m}^d \bmod N$

1: *//Randomize the private exponent*
2: Pick a random $\lambda \in [\![0; 2^l - 1]\!]$;
3: $\bar{d} = d + \lambda\varphi(N)$;
4: *//Perform the exponentiation*
5: $S = \texttt{PowMod}(\dot{m}, \bar{d}, N)$;
6: **return** S;

- Typically for a 1024-bit RSA : $l = 20$ or 32 bits

Efficiency Against Public Key Perturbations

Difficulty due to Exponent Randomization

- The fault injection isolates a part of $\bar{d}$ instead of d
- Prevent from cascading the analysis

Efficiency Against Public Key Perturbations

Difficulty due to Exponent Randomization

- The fault injection isolates a part of $\bar{d}$ instead of d
- Prevent from cascading the analysis

Solution

- Randomization is not homogeneous
- Such a perturbation isolates a part of $\bar{d}$

$\Rightarrow$ $\bar{d}$ may leak information on d

Outline

Non-Homogeneity of the Masking Operation

Details of the Randomization

$$\bar{d} = d + \lambda\varphi(N)$$

Non-Homogeneity of the Masking Operation

Details of the Randomization

$$\begin{aligned}\bar{d} &= d + \lambda\varphi(N) \\ &= d + \lambda(p-1)(q-1)\end{aligned}$$

Non-Homogeneity of the Masking Operation

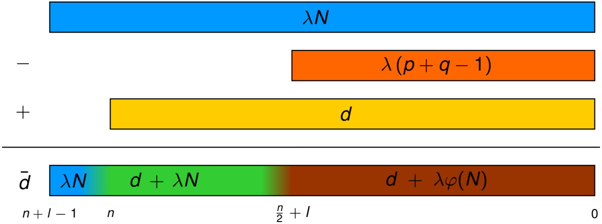
Details of the Randomization

$$\begin{aligned} \bar{d} &= d + \lambda\varphi(N) \\ &= d + \lambda(p-1)(q-1) \\ &= d + \lambda N - \lambda(p+q-1) \end{aligned}$$

Non-Homogeneity of the Masking Operation

Details of the Randomization

$$
\begin{aligned}
\bar{d} &= d + \lambda\varphi(N) \\
&= d + \lambda(p-1)(q-1) \\
&= d + \lambda N - \lambda(p+q-1)
\end{aligned}
$$



Outline

Principle of the Attack: MSB Case (1/2)

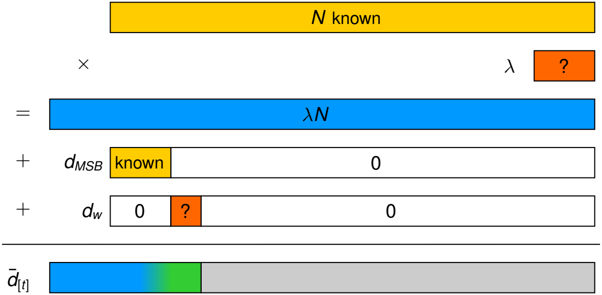
Approximation on MSB

$$\bar{d} \approx d + \lambda N$$

Principle of the Attack: MSB Case (1/2)

Approximation on MSB

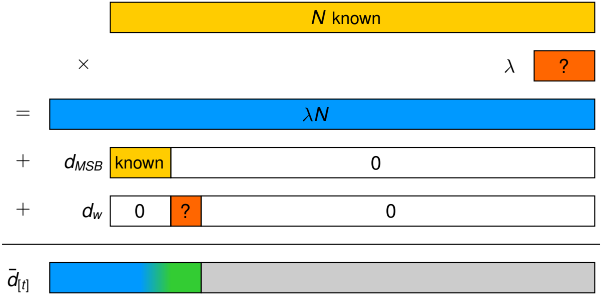
$$\bar{d} \approx d + \lambda N$$



Principle of the Attack: MSB Case (1/2)

Approximation on MSB

$$\bar{d} \approx d + \lambda N$$



$\Rightarrow$ Guessing (λ, d_w) enables to compute good candidate values for $\bar{d}_{[t]}$

Principle of the Attack: MSB Case (2/2)

Theorem

Let $\hat{S}_t$ be a faulty signature performed under an exponent randomized by λ, and S the corresponding correct signature. For all candidate pairs $(d'_w, \lambda') \in [\![0; 2^w]\!] \times [\![0; 2^l]\!]$, if $\lambda' > \lambda$, then (8) can not be satisfied.

$\Rightarrow$ If $t > l$, λ can be exactly determined by building good values for $\bar{d}_{[t]}$

Principle of the Attack: MSB Case (2/2)

Theorem

Let $\hat{S}_t$ be a faulty signature performed under an exponent randomized by λ, and S the corresponding correct signature. For all candidate pairs $(d'_w, \lambda') \in [\![0; 2^w]\!] \times [\![0; 2^l]\!]$, if $\lambda' > \lambda$, then (8) can not be satisfied.

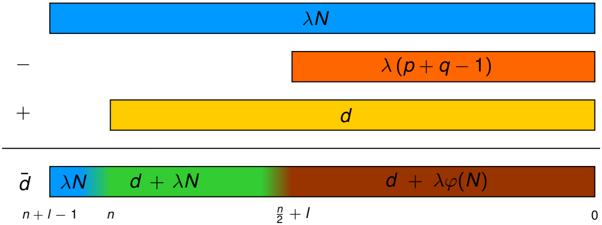
$\Rightarrow$ If $t > l$, λ can be exactly determined by building good values for $\bar{d}_{[t]}$

Faulty randomized RSA signatures analysis

1. Inject a fault on **N** during a signature
   $\Rightarrow$ A part of the blinded key $\bar{d}_{[t]}$ is isolated
2. Determine $\bar{d}_{[t]}$ and $\hat{N}$ from the pair $\left(S, \hat{S}\right)$
   $\Rightarrow$ Good candidates for $\bar{d}_{[t]}$ are built from a known part of the private key d_{MSB}, and candidate pairs for d_w and λ
   $\Rightarrow$ Candidates for $\hat{N}$ are built according to the fault model
3. Update the known part of the key d_{MSB} and repeat the analysis on signatures faulted earlier until the most significant part of **d** is determined

Principle of the Attack: LSB Case

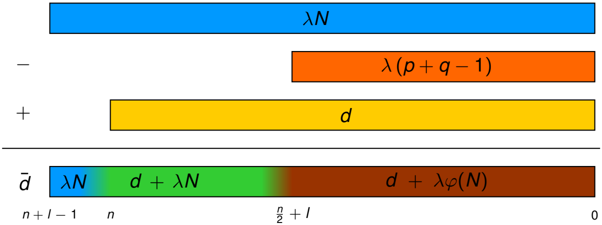
- Previous approximation not valid for LSB



$\Rightarrow$ $\bar{d}$ depends on an additional variable

Principle of the Attack: LSB Case

- Previous approximation not valid for LSB



$\Rightarrow$ $\bar{d}$ depends on an additional variable

Solution

- Get one more faulty signature to analyze (2 in practice)
- Make a variable change to boil down to the MSB case
- Solve a system to extract bits of **d**

Results

Complexity evaluation

- Estimated fault number

$$\mathcal{F} = \mathcal{O}\left(\frac{n}{w}\right) \text{ signatures}$$

- Computational complexity

$$\mathcal{C} = \mathcal{O}\left(\frac{2^{(l+w)} \cdot n^2}{w}\right) \text{ exponentiations}$$

$\Rightarrow$ Possible improvement: combine it with Coppersmith Attacks

Key extraction on a PC for a 1024-bit Randomized RSA

- $l = 20$ bits, $w = 2$ (bits of d recovered by pairs)
- 1000 faulty signatures
- Roughly 2^{40} exponentiations

Conclusion

- Physical robustness of the countermeasure
  - Randomized Exponentiation
    $\Rightarrow$ *"Doubling Attack"* [FV03]
    $\Rightarrow$ Small Public Exponent [FKJM^{+}06]
  - Blinding Operation
    $\Rightarrow$ *"Carry Analysis"* [FRVD08]

Conclusion

- Physical robustness of the countermeasure
  - Randomized Exponentiation
    $\Rightarrow$ *"Doubling Attack"* [FV03]
    $\Rightarrow$ Small Public Exponent [FKJM$^+$06]
  - Blinding Operation
    $\Rightarrow$ *"Carry Analysis"* [FRVD08]

First fault attack against randomized RSA

- Answer an open problem raised by E. Brier *et al.* at CHES 2006 [BCMCC06]
- Realistic fault model
- Reasonable performances

Conclusion

- Physical robustness of the countermeasure
  - Randomized Exponentiation
    $\Rightarrow$ *"Doubling Attack"* [FV03]
    $\Rightarrow$ Small Public Exponent [FKJM^{+}06]
  - Blinding Operation
    $\Rightarrow$ *"Carry Analysis"* [FRVD08]

First fault attack against randomized RSA

- Answer an open problem raised by E. Brier *et al.* at CHES 2006 [BCMCC06]
- Realistic fault model
- Reasonable performances

Perspectives

- Exponent blinding does not provide a strong hardware security
- What about homogeneous blinding operation ?

Thank you !


I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!
I must not spam alexandre.berzati@ceafr!

References I

A. Berzati, Cécile Canovas, G. Castagnos, B. Debraize, L. Goubin, A. Gouget, P. Paillier, and S. Salgado.
Fault Analysis of Grain-128.
In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2009)*, San Francisco (USA), 2009. IEEE Computer Society.

A. Berzati, C. Canovas, J-G. Dumas, and L. Goubin.
Fault Attacks on RSA Public Keys: Left-To-Right Implementations are also Vulnerable.
In M. Fischlin, editor, *RSA Cryptographer's Track (CT-RSA 2009)*, volume 5473 of *Lecture Notes in Computer Science*, pages 414–428, San Francisco (USA), 2009. Springer.

A. Berzati, C. Canovas, and L. Goubin.
Perturbating RSA Public Keys: an Improved Attack.
In *Cryptographic Hardware and Embedded Systems (CHES 2008)*, Lecture Notes in Computer Science, Washington DC (USA), 2008. Springer-Verlag.

E. Brier, B. Chevallier-Mames, M. Ciet, and C. Clavier.
Why One Should Also Secure RSA Public Key Elements.
In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems (CHES 2006)*, volume 4249 of *Lecture Notes in Computer Science*, pages 324–338. Springer-Verlag, 2006.

D. Boneh, R.A. DeMillo, and R.J. Lipton.
On the Importance of Checking Cryptographic Protocols for Faults.
In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer-Verlag, 1997.

E. Biham, L. Granboulan, and P. Nguyen.
Impossible Fault Analysis of RC4 and Differential Analysis of RC4.
In H. Gilbert and H. Handschuh, editors, *Fast Software Encryption (FSE 2005)*, volume 3557, pages 359–367. Springer, 2005.

References II

I. Biehl, B. Meyer, and V. Müller.
Differential Fault Attacks on Ellitic Curve Cryptosystems.
In M. Bellare, editor, *Advances in Cryptology (CRYPTO 2000)*, volume 1880 of *Lecture Notes in Computer Science*, pages 131–146. Springer-Verlag, 2000.

E. Biham and A. Shamir.
Differential Fault Analysis of Secret Key Cryptosystems.
In *Advances in Cryptology (CRYPTO 1997)*, 1997.

M. Ciet and M. Joye.
Elliptic Curve Cryptosystems in the presence of permanent and transient faults.
Designs, Codes and Cryptography, 36(1):33–43, 2005.

J-S. Coron.
Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems.
In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES 1999)*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.

P. Dusart, G. Letourneux, and O. Vivolo.
Differential Fault Analysis on AES.
In M. Yung, Y. Han, and J. Zhou, editors, *Applied Cryptography and Network Security (ANCS 2003)*, volume 2846 of *Lecture Notes in Computer Science*, pages 293–306. Springer, 2003.

P-A. Fouque, S. Kunz-Jacques, G. Martinet, F. Muller, and F. Valette.
Power Attack on Small RSA Public Exponent.
In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems (CHES 2006)*, volume 4249 of *Lecture Notes in Computer Science*, pages 339–353. Springer, 2006.

References III

P-A. Fouque, D. Réal, F. Valette, and M. Drissi.
The Carry Leakage on the Randomized Exponent Countermeasure.
In E. Oswald and P. P. Rohatgi, editors, *Cryptographic Hardware and Embedded Systems (CHES 2008)*, volume 5154 of *Lecture notes in Computer Science*, pages 198–213. Springer, 2008.

P-A. Fouque and F. Valette.
The Doubling Attack – *why Upwards Is Better than Downwards*.
In C.D. Walter, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES 2003)*, volume 2779, pages 269–280. Springer, 2003.

M. Gomulkiewicz, M. Kutilwoski, and P. Wlaz.
Synchronization Fault Analysis for Breaking A5/1.
In Sotiris E. Nikoletseas, editor, *Experimental and Efficient Algorithms (WEA 2005)*, volume 3503 of *Lecture Notes in Computer Science*, pages 415–427. Springer, 2005.

S. Gueron and J.-P. Seifert.
Is It Wise to Publish Your Public RSA Keys?
In L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC 2006)*, volume 4236 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

J. Hoch and A. Shamir.
Fault Analysis of Stream Ciphers.
In M. Joye and J-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems (CHES 2004)*, volume 3156 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 2004.

C.H. Kim, P. Bulens, C. Petit, and J.-J.Quisquater.
Fault Attaks on Public Key Elements: Application to DLP-Based Schemes.
In S.F. Mjølsnes, S. Mauw, and S.K. Katsikas, editors, *European PKI workshop Public Key Infrastructure (EuroPKI 2008)*, volume 5057 of *Lecture Notes In Computer Science*, pages 182–195. Springer, 2008.

References IV

P. Kocher.
Timing attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems.
In N. Koblitz, editor, *Advances in Cryptology (CRYPTO 1996)*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.