

micro and nanoelectronics
microsystems
ambient intelligence
image chain
biology and health



Fault Attacks on Public Keys

Cécile Canovas and Alexandre Berzati

CEA-LETI Minatec et Université de Versailles

5 Juin 2009



Outline

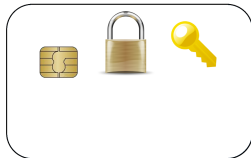
Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	oo ooo	ooo ooo	oo oooo	

- 1 Introduction
- 2 IFP-based algorithms
- 3 DLP-based algorithms
- 4 ECDLP-based algorithms
- 5 Conclusion

Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○○	

■ Signature

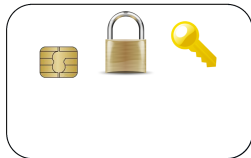


Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ Signature

hash message m



Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ Signature

hash message m



Asymmetric cryptography

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

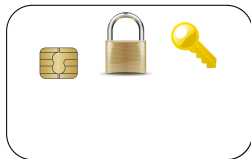
○○
○○○○

■ Signature

hash message m



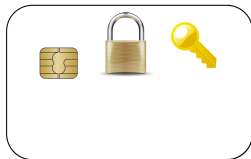
signature S



Fault Attacks on Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○○	

- Differential Fault Analysis (*DFA*)



Fault Attacks on Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ Differential Fault Analysis (*DFA*)

hash message m

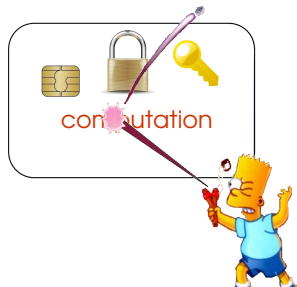


Fault Attacks on Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○○ ○○○○	

■ Differential Fault Analysis (DFA)

hash message m



Fault Attacks on Asymmetric cryptography

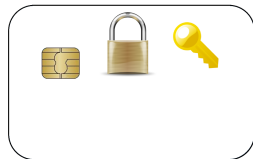
Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ Differential Fault Analysis (*DFA*)

hash message m



signature \hat{S}



Fault Attacks on Asymmetric cryptography

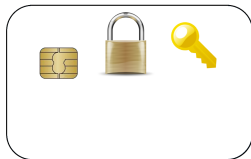
Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ Differential Fault Analysis (DFA)

hash message m



signature \hat{s}

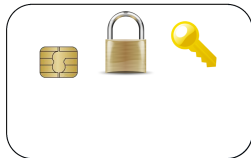


The key is recovered from the difference between s and \hat{s}

Fault Attacks on Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○○	

■ "Structure" Fault Attacks



Fault Attacks on Asymmetric cryptography

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ "Structure" Fault Attacks

hash message m



Fault Attacks on Asymmetric cryptography

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

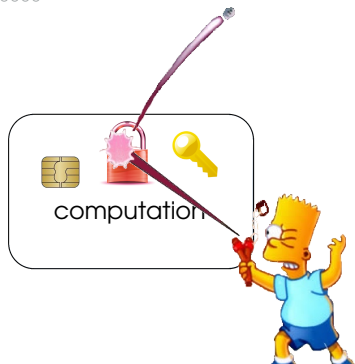
○○
○○○

○○○
○○○

○○
○○○○

■ "Structure" Fault Attacks

hash message m



Fault Attacks on Asymmetric cryptography

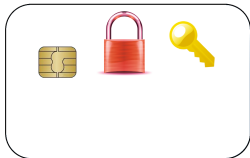
Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ "Structure" Fault Attacks

hash message m



signature \hat{S}



Fault Attacks on Asymmetric cryptography

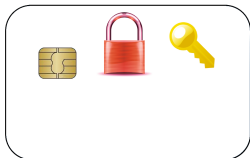
Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ "Structure" Fault Attacks

hash message m



signature \hat{S}



The key is recovered from \hat{S} because of the weak algebraic structure

Outline

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	●○ ○○○	○○○ ○○○	○○ ○○○	

- 1 Introduction
- 2 IFP-based algorithms
 - RSA Signature Scheme
 - Fault Attacks
- 3 DLP-based algorithms
- 4 ECDLP-based algorithms
- 5 Conclusion

RSA Signature Scheme

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○● ○○○	○○○ ○○○	○○○ ○○○○	

■ Key generation

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

RSA Signature Scheme

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○● ○○○	○○○ ○○○	○○○ ○○○○	

■ Key generation

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

■ Signature

- Return $S \equiv h(m)^d \pmod{N}$

RSA Signature Scheme

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○● ○○○	○○○ ○○○	○○○ ○○○○	

■ Key generation

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

■ Signature

- Return $S \equiv h(m)^d \pmod{N}$

■ Signature verification

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

Outline

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ●○○	○○○ ○○○	○○ ○○○	

- 1 Introduction
- 2 IFP-based algorithms
 - RSA Signature Scheme
 - Fault Attacks
- 3 DLP-based algorithms
- 4 ECDLP-based algorithms
- 5 Conclusion

Why One Should Also Secure RSA Public Key Elements (BCMCC06)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○●○	○○○ ○○○	○○ ○○○○	

■ Fault Model

- The attacker performs a perturbation campaign by collecting faulty signatures computed under unknown faulty moduli

Why One Should Also Secure RSA Public Key Elements (BCMCC06)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○●○	○○○ ○○○	○○ ○○○	

■ Fault Model

- The attacker performs a perturbation campaign by collecting faulty signatures computed under unknown faulty moduli

■ Fault Analysis

- From some faulty signatures, the attacker recovers small residues of d by solving small D.L.
- The whole d is recovered with the Chinese Remainder Theorem

■ Variant

- Use of a constrained fault model and moduli dictionary

Fault Attacks on RSA Public Keys (BCDG09)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○●	○○○ ○○○	○○ ○○○	

■ Fault Model

- A byte of the modulus is corrupted during the exponentiation
- The faulty modulus has to be prime or *smooth*
- A dictionary of prime faulty moduli has to be computed

Fault Attacks on RSA Public Keys (BCDG09)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○●	○○○ ○○○	○○ ○○○	

■ Fault Model

- A byte of the modulus is corrupted during the exponentiation
- The faulty modulus has to be prime or *smooth*
- A dictionary of prime faulty moduli has to be computed

■ Fault Analysis

- The faulty signature is:

$$\hat{S} = A^{2^w} \cdot h(m)^{d_w} \bmod \hat{N} \quad (1)$$

where A denotes an intermediate value before the perturbation and d_w a partial value of d

- The values (d_w, \hat{N}) are guessed and determined
- Computation of square roots
- The whole d is gradually recovered

Outline

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	●○○ ○○○	○○ ○○○	

- 1 Introduction
- 2 IFP-based algorithms
- 3 DLP-based algorithms
 - ElGamal Signature Scheme
 - DSA Signature Scheme
- 4 ECDLP-based algorithms
- 5 Conclusion

ElGamal Signature Scheme

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○● ○○○	○○ ○○○○	

■ Key generation

- Pick a random prime p , g a generator of $\mathbb{Z}/p\mathbb{Z}^*$ and a random x s.t.

$$y = g^x \bmod p \quad (2)$$

- The public key is (y, g, p)
- The private key is x

ElGamal Signature Scheme

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	●○○ ○○○	○○ ○○○○	

■ Key generation

- Pick a random prime p , g a generator of $\mathbb{Z}/p\mathbb{Z}^*$ and a random x s.t.

$$y = g^x \bmod p \quad (2)$$

- The public key is (y, g, p)
- The private key is x

■ Signature

- Pick a random k s.t. $\gcd(k, p-1) = 1$
- Compute $u \equiv g^k \bmod p$ and $v \equiv \frac{h(m) - xu}{k} \bmod (p-1)$
- Return the couple (u, v)

ElGamal Signature Scheme

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○●
○○○

○○
○○○○

■ Key generation

- Pick a random prime p , g a generator of $\mathbb{Z}/p\mathbb{Z}^*$ and a random x s.t.

$$y = g^x \bmod p \quad (2)$$

- The public key is (y, g, p)
- The private key is x

■ Signature

- Pick a random k s.t. $\gcd(k, p-1) = 1$
- Compute $u \equiv g^k \bmod p$ and $v \equiv \frac{h(m) - xu}{k} \bmod (p-1)$
- Return the couple (u, v)

■ Signature verification

- Check that $y^u \cdot u^v \equiv g^{h(m)} \bmod p$

Fault Attack (Reference (KBPJJ08))

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○● ○○○	○○ ○○○○	

■ Fault Model

- The attacker can generate random faults on \mathbf{p}
- He knows (or can guess) the resulting faulty modulus \hat{p}
- If $\gcd(\mathbf{k}, \hat{p} - 1) = 1$, we have:

$$\hat{u} \equiv \mathbf{g}^{\mathbf{k}} \bmod \hat{p} \quad \text{and} \quad \hat{v} \equiv \frac{h(m) - \mathbf{x}\hat{u}}{\mathbf{k}} \bmod (\hat{p} - 1) \quad (3)$$

Fault Attack (Reference (KBPJJ08))

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○● ○○○	○○ ○○○○	

■ Fault Model

- The attacker can generate random faults on \mathbf{p}
- He knows (or can guess) the resulting faulty modulus $\hat{\mathbf{p}}$
- If $\gcd(\mathbf{k}, \hat{\mathbf{p}} - 1) = 1$, we have:

$$\hat{u} \equiv \mathbf{g}^{\mathbf{k}} \bmod \hat{\mathbf{p}} \quad \text{and} \quad \hat{v} \equiv \frac{h(m) - \mathbf{x}\hat{u}}{\mathbf{k}} \bmod (\hat{\mathbf{p}} - 1) \quad (3)$$

■ Fault Analysis

- Let \dagger s.t. $\dagger \mid \hat{\mathbf{p}}$ and $\varphi(\dagger) \mid (\hat{\mathbf{p}} - 1)$

$$\hat{u}^{\hat{v}} \equiv \mathbf{g}^{\mathbf{k} \frac{h(m) - \mathbf{x}\hat{u}}{\mathbf{k}}} \equiv \mathbf{g}^{h(m) - \mathbf{x}\hat{u}} \bmod \dagger$$

$$\frac{\hat{u}^{\hat{v}}}{\mathbf{g}^{h(m)}} \equiv (\mathbf{g}^{-\hat{u}})^{\mathbf{x}} \bmod \dagger$$

- So, each fault analysis makes the attacker recover $\mathbf{x} \bmod \mathbf{r}$, where \mathbf{r} denotes the order of $(\mathbf{g}^{-\hat{u}})$ modulo \dagger

Outline

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ●○○	○○ ○○○	

- 1 Introduction
- 2 IFP-based algorithms
- 3 DLP-based algorithms
 - ElGamal Signature Scheme
 - DSA Signature Scheme
- 4 ECDLP-based algorithms
- 5 Conclusion

DSA Signature Scheme

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ●○○	○○ ○○○○	

■ Key generation

- Pick a random prime p, q s.t. $q \mid (p - 1)$, $g \in \mathbb{Z}/p\mathbb{Z}^*$ s.t. $\text{ord}(g) = q$
- Then, pick a random x s.t. $0 < x < q$ and compute:

$$y = g^x \bmod p \quad (4)$$

- The public key is (y, g, p, q)
- The private key is x

DSA Signature Scheme

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○○

■ Key generation

- Pick a random prime p, q s.t. $q \mid (p - 1)$, $g \in \mathbb{Z}/p\mathbb{Z}^*$ s.t. $\text{ord}(g) = q$
- Then, pick a random x s.t. $0 < x < q$ and compute:

$$y = g^x \bmod p \quad (4)$$

- The public key is (y, g, p, q)
- The private key is x

■ Signature

- Pick a random k s.t. $\text{gcd}(k, p - 1) = 1$
- Compute $u \equiv (g^k \bmod p) \bmod q$ and $v \equiv \frac{h(m) + xu}{k} \bmod q$
- Return the couple (u, v)

DSA Signature Scheme

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○○

■ Key generation

- Pick a random prime p, q s.t. $q \mid (p - 1)$, $g \in \mathbb{Z}/p\mathbb{Z}^*$ s.t. $\text{ord}(g) = q$
- Then, pick a random x s.t. $0 < x < q$ and compute:

$$y = g^x \bmod p \quad (4)$$

- The public key is (y, g, p, q)
- The private key is x

■ Signature

- Pick a random k s.t. $\text{gcd}(k, p - 1) = 1$
- Compute $u \equiv (g^k \bmod p) \bmod q$ and $v \equiv \frac{h(m) + xu}{k} \bmod q$
- Return the couple (u, v)

■ Signature verification

- Compute $w = v^{-1} \bmod q$
- Check that $(g^{wh(m)} y^{wu}) \bmod q = u$

Fault Attack (Reference (KBPJJ08))

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○●	○○ ○○○	

■ Fault Model

- The attacker can generate random faults on \mathbf{p} and \mathbf{q}
- He knows (or can guess) resulting faulty moduli \hat{p} and \hat{q}
- If $\gcd(k, \hat{q}) = 1$, we have:

$$\hat{u} \equiv (g^k \bmod \hat{p}) \bmod \hat{q} \quad \text{and} \quad \hat{v} \equiv \frac{h(m) + x\hat{u}}{k} \bmod \hat{q} \quad (5)$$

Fault Attack (Reference (KBPJJ08))

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○●

○○
○○○○

■ Fault Model

- The attacker can generate random faults on \mathbf{p} and \mathbf{q}
- He knows (or can guess) resulting faulty moduli $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$
- If $\gcd(k, \hat{\mathbf{q}}) = 1$, we have:

$$\hat{u} \equiv (g^k \bmod \hat{\mathbf{p}}) \bmod \hat{\mathbf{q}} \quad \text{and} \quad \hat{v} \equiv \frac{h(m) + \mathbf{x}\hat{u}}{k} \bmod \hat{\mathbf{q}} \quad (5)$$

■ Fault Analysis

- Let \mathbf{t} s.t. $\mathbf{t} \mid \hat{\mathbf{p}}, \mathbf{t} \mid \hat{\mathbf{q}}$ and $\varphi(\mathbf{t}) \mid (\hat{\mathbf{p}} - 1)$

$$\hat{u}^{\hat{v}} \equiv g^{k \frac{h(m) + \mathbf{x}\hat{u}}{k}} \equiv g^{h(m) + \mathbf{x}\hat{u}} \bmod \mathbf{t}$$

$$\frac{\hat{u}^{\hat{v}}}{g^{h(m)}} \equiv (g^{\hat{u}})^{\mathbf{x}} \bmod \mathbf{t}$$

- So, each fault analysis makes the attacker recover $\mathbf{x} \bmod \mathbf{r}$, where \mathbf{r} denotes the order of $(g^{\hat{u}})$ modulo \mathbf{t}

Outline

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	●○ ○○○○	

- 1 Introduction
- 2 IFP-based algorithms
- 3 DLP-based algorithms
- 4 ECDLP-based algorithms
 - Introduction
 - Fault Attacks
- 5 Conclusion

Elliptic Curves

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○● ○○○○	

■ Definition

- An elliptic curve $\mathcal{E}(\mathbf{a}, \mathbf{b})$ defined over a finite field \mathbb{F}_p , where $p > 3$ can be given as:

$$\mathcal{E}(\mathbb{F}_p) : y^2 = x^3 + \mathbf{a}x + \mathbf{b} \quad \mathbf{a}, \mathbf{b} \in \mathbb{F}_p \quad (6)$$

where the associated discriminant $\Delta = -16(4\mathbf{a}^3 + 27\mathbf{b}^2) \neq 0$

Elliptic Curves

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	●● ○○○	

■ Definition

- An elliptic curve $\mathcal{E}(\mathbf{a}, \mathbf{b})$ defined over a finite field \mathbb{F}_p , where $p > 3$ can be given as:

$$\mathcal{E}(\mathbb{F}_p) : y^2 = x^3 + \mathbf{a}x + \mathbf{b} \quad \mathbf{a}, \mathbf{b} \in \mathbb{F}_p \quad (6)$$

where the associated discriminant $\Delta = -16(4\mathbf{a}^3 + 27\mathbf{b}^2) \neq 0$

■ Algebraic Structure

- We can define a law $+$ over the elliptic curve field that performs a point addition
- An elliptic curve $\mathcal{E}(\mathbb{F}_p)$ with this law $+$ forms an abelian group

Elliptic Curves

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

●●
○○○

■ Definition

- An elliptic curve $\mathcal{E}(\mathbf{a}, \mathbf{b})$ defined over a finite field \mathbb{F}_p , where $p > 3$ can be given as:

$$\mathcal{E}(\mathbb{F}_p) : y^2 = x^3 + \mathbf{a}x + \mathbf{b} \quad \mathbf{a}, \mathbf{b} \in \mathbb{F}_p \quad (6)$$

where the associated discriminant $\Delta = -16(4\mathbf{a}^3 + 27\mathbf{b}^2) \neq 0$

■ Algebraic Structure

- We can define a law $+$ over the elliptic curve field that performs a point addition
- An elliptic curve $\mathcal{E}(\mathbb{F}_p)$ with this law $+$ forms an abelian group

■ Scalar Multiplication

- Let $\mathbf{P} \in \mathcal{E}(\mathbb{F}_p)$ and $d \in \mathbb{F}_p$ be a random value:

$$\mathbf{Q} = d \cdot \mathbf{P} = \mathbf{P} + \mathbf{P} \dots + \mathbf{P} \quad d - \text{times} \quad (7)$$

Outline

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ●○○○	

- 1 Introduction
- 2 IFP-based algorithms
- 3 DLP-based algorithms
- 4 ECDLP-based algorithms
 - Introduction
 - Fault Attacks
- 5 Conclusion

Biehl-Meyer-Müller Attack (BMM00)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ●○○○	

■ Fault model

- Faults on the Input Point \mathbf{P} ($\hat{\mathbf{P}}$ known)
- \mathbf{P} is changed s.t $\hat{\mathbf{P}} \in \mathcal{E}'(\mathbf{a}, \hat{\mathbf{b}})$ whose order has a small divisor r
- $\hat{\mathbf{b}}$ may not be use to perform the point addition (ANSI X9.63 and IEEE 1363)

Biehl-Meyer-Müller Attack (BMM00)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○	

■ Fault model

- Faults on the Input Point \mathbf{P} (\hat{P} known)
- \mathbf{P} is changed s.t $\hat{P} \in \mathcal{E}'(\mathbf{a}, \hat{b})$ whose order has a small divisor \mathbf{r}
- \hat{b} may not be use to perform the point addition (ANSI X9.63 and IEEE 1363)

■ Fault Analysis

- $ord(\hat{P}) = r$ and $\hat{Q} = \mathbf{d} \cdot \hat{P}$ is computed over $\mathcal{E}'(\mathbf{a}, \hat{b})$
- Since \mathbf{r} is small, compute the D.L. in $\langle \hat{P} \rangle$ and so find $\mathbf{d} \bmod \mathbf{r}$
- Repeat the process and get \mathbf{d} by the Chinese Remainder Theorem

Biehl-Meyer-Müller Attack (BMM00)

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
●○○○

■ Fault model

- Faults on the Input Point \mathbf{P} (\hat{P} known)
- \mathbf{P} is changed s.t $\hat{P} \in \mathcal{E}'(\mathbf{a}, \hat{b})$ whose order has a small divisor r
- \hat{b} may not be use to perform the point addition (ANSI X9.63 and IEEE 1363)

■ Fault Analysis

- $ord(\hat{P}) = r$ and $\hat{Q} = \mathbf{d} \cdot \hat{P}$ is computed over $\mathcal{E}'(\mathbf{a}, \hat{b})$
- Since r is small, compute the D.L. in $\langle \hat{P} \rangle$ and so find $\mathbf{d} \bmod r$
- Repeat the process and get \mathbf{d} by the Chinese Remainder Theorem

■ Additional Fault Model

- Placing Register Faults – Random bit fault on \mathbf{P}

Ciet-Joye Attack (CJ05)

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○●○	

■ Fault Model

- An unknown bit of the x-coordinate of \mathbf{P} is permanently corrupted
- $\hat{P}(\hat{x}, y) \in \mathcal{E}'(\alpha, \hat{b})$ whose order has a small divisor r , and

$$\hat{Q} = d \cdot \hat{P} = (\hat{x}_Q, \hat{y}_Q) \quad (8)$$

Ciet-Joye Attack (CJ05)

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○●○

■ Fault Model

- An unknown bit of the x -coordinate of \mathbf{P} is permanently corrupted
- $\hat{P}(\hat{x}, y) \in \mathcal{E}'(a, \hat{b})$ whose order has a small divisor r , and

$$\hat{Q} = d \cdot \hat{P} = (\hat{x}_Q, \hat{y}_Q) \quad (8)$$

■ Fault Analysis

- First, recover \hat{b} by noticing that $\hat{Q} \in \mathcal{E}'(a, \hat{b})$: $\hat{b} = y^2 - \hat{x}_Q^3 - a\hat{x}_Q$
- Then, since $\hat{P}(\hat{x}, y) \in \mathcal{E}'(a, \hat{b})$, \hat{x} is a root of $X^3 + aX + \hat{b} - y^2$
- The root that has most matching bits with \mathbf{x} is taken as \hat{x}
- If $\text{ord}(\hat{P}) = r$ is small, compute the D.L. in $\langle \hat{P} \rangle$ and find $d \bmod r$

Ciet-Joye Attack (CJ05)

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○●○

■ Fault Model

- An unknown bit of the x -coordinate of \mathbf{P} is permanently corrupted
- $\hat{P}(\hat{x}, y) \in \mathcal{E}'(a, \hat{b})$ whose order has a small divisor r , and

$$\hat{Q} = d \cdot \hat{P} = (\hat{x}_Q, \hat{y}_Q) \quad (8)$$

■ Fault Analysis

- First, recover \hat{b} by noticing that $\hat{Q} \in \mathcal{E}'(a, \hat{b})$: $\hat{b} = y^2 - \hat{x}_Q^3 - a\hat{x}_Q$
- Then, since $\hat{P}(\hat{x}, y) \in \mathcal{E}'(a, \hat{b})$, \hat{x} is a root of $X^3 + aX + \hat{b} - y^2$
- The root that has most matching bits with \mathbf{x} is taken as \hat{x}
- If $\text{ord}(\hat{P}) = r$ is small, compute the D.L. in $\langle \hat{P} \rangle$ and find $d \bmod r$

■ Additional Fault Model

- Permanent faults on y -coordinates
- Bit-error on the field parameter \mathbf{q}

"Twist Attack" (FLRV08)

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○●

■ Definition

- The twist of \mathcal{E} by \mathbf{c} defined over \mathbb{F}_p where $p > 3$ can be given as:

$$\mathcal{E}_c(\mathbb{F}_p) : y^2 = x^3 + \mathbf{a}c^2x + \mathbf{b}c^3 \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{F}_p \quad (9)$$

- The number of points on the twist is *smooth*

"Twist Attack" (FLRV08)

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○●

■ Definition

- The twist of \mathcal{E} by \mathbf{c} defined over \mathbb{F}_p where $p > 3$ can be given as:

$$\mathcal{E}_c(\mathbb{F}_p) : y^2 = x^3 + \mathbf{a}c^2x + \mathbf{b}c^3 \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{F}_p \quad (9)$$

- The number of points on the twist is *smooth*

■ Fault Model

- The attackers modifies the x -coordinate of \mathbf{P} s.t. $\hat{\mathbf{P}} \in \mathcal{E}_c$
- The fault is induced s.t. $\hat{\mathbf{Q}} = \mathbf{d} \cdot \hat{\mathbf{P}} \in \mathcal{E}_c$
- The attack targets the Montgomery Ladder implementation of the scalar multiplication (y -coordinates not used)

"Twist Attack" (FLRV08)

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○●

■ Definition

- The twist of \mathcal{E} by \mathbf{c} defined over \mathbb{F}_p where $p > 3$ can be given as:

$$\mathcal{E}_c(\mathbb{F}_p) : y^2 = x^3 + \mathbf{a}c^2x + \mathbf{b}c^3 \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{F}_p \quad (9)$$

- The number of points on the twist is *smooth*

■ Fault Model

- The attackers modifies the x -coordinate of \mathbf{P} s.t. $\hat{\mathbf{P}} \in \mathcal{E}_c$
- The fault is induced s.t. $\hat{\mathbf{Q}} = \mathbf{d} \cdot \hat{\mathbf{P}} \in \mathcal{E}_c$
- The attack targets the Montgomery Ladder implementation of the scalar multiplication (y -coordinates not used)

■ Fault Analysis

- From $\hat{\mathbf{Q}}$, the attacker recovers the parameter of the twist \mathbf{c}
- The attackers solve D.L. and recover $\mathbf{d} \bmod \text{ord}(\hat{\mathbf{P}})$

Conclusion

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	oo ooo	ooo ooo	oo oooo	

■ "Structure" Fault Attack

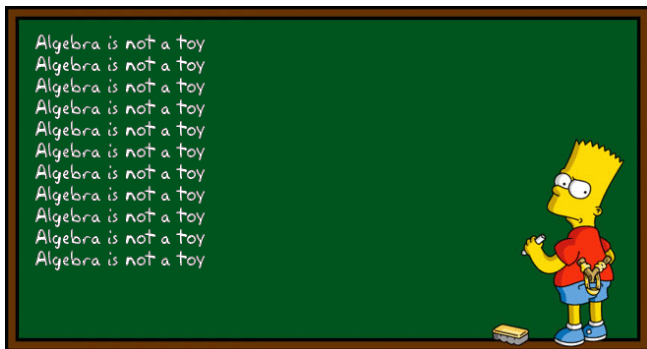
- Use fault to compute cryptographic functions in weaker finite fields
- Perturbation of public elements
- Different algebraic structure targeted

■ Consequence

- Protection of public key elements ...
- ... and also the algebraic structure

Thank you !

Introduction	IFP-based algorithms	DLP-based algorithms	ECDLP-based algorithms	Conclusion
	○○ ○○○	○○○ ○○○	○○ ○○○○	



References I

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○○



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. Springer, 2009.



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.



I. Biehl, B. Meyer, and V. Müller.

Differential Fault Attacks on Elliptic 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.



J. Blömer, M. Otto, and J-P. Seifert.

Sign Change Fault Attacks on Elliptic Curve Cryptosystems.

In L. Breveglieri, I. Koren, D. Naccache, and J-P. Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography*, volume 4236 of *Lecture Notes in Computer Science*, pages 36–52. Springer-Verlag, 2006.



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.

References II

Introduction IFP-based algorithms DLP-based algorithms ECDLP-based algorithms Conclusion

○○
○○○

○○○
○○○

○○
○○○○



P-A. Fouque, R. Lercier, D. Réal, and F. Valette.

Fault attack on elliptic curve montgomery ladder implementation.

In L. Breveglieri, S. Gueron, I. Koren, D. Naccache, and J-P. Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC 2008)*, pages 92–98. IEEE Computer Society, 2008.



C.H. Kim, P. Bullens, C. Petit, and J-J. Quisquater.

Fault Attacks 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.

Biehl-Meyer-Müller Attack (BMM00) (1/2)

Fault Attacks against ECDLP

- Placing Register Faults – Random bit fault on P
 - The fault is injected after checking that P is on the curve $\mathcal{E}(a, b)$
 - $\hat{P} \in \mathcal{E}'(a, \hat{b})$ differs from P in one bit at an unknown position
 - If $\mathcal{E}'(a, \hat{b})$ is weak, find \hat{b} from \hat{Q}
 - Check for all possible \hat{P} candidates and try to compute the D.L. to find a residue of d

Biehl-Meyer-Müller Attack (BMM00) (2/2)

Fault Attacks against ECDLP

■ Faults at Random moments of the Multiplication

- A bit-flip is induced on an internal register during the multiplication
- If the “*Right-to-Left*” binary method is used:

$$\hat{Q} = \hat{Q}_j + d_{[j..(n-1)]} \cdot P \quad (10)$$

where Q_j denotes the internal register value at the j -th step and $d_{[j..(n-1)]}$ the j most significant bits of d

- For all candidate values $d'_{[j..(n-1)]}$, compute

$$Q'_j = Q - d'_{[j..(n-1)]} \cdot P \quad (11)$$

Then, from Q'_j , generate all possible faulty values \tilde{Q}'_j and test if the following equation is satisfied:

$$\tilde{Q}'_j + d'_{[j..(n-1)]} \cdot P = \hat{Q} \quad (12)$$

- In case of success a part of d is recovered

■ Additional Fault Model

- Sign Change Fault Attacks (BOS06)