Elliptic Curve Cryptography on Embedded Devices

Scalar Multiplication and Side-Channel Attacks

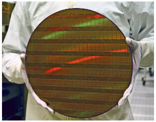
Vincent Verneuil[1,2]

[1] Inside Secure
[2] Institut de Mathématiques de Bordeaux

Séminaire Arithmétique et Théorie de l'Information
Institut de Mathématiques de Luminy
01 / 2011

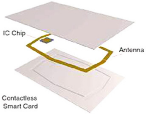
Outline

Inside Secure in (very) short



Manufacturer

→

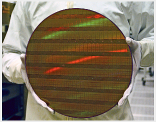


Chip Embedder

→

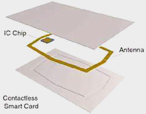
Issuer

Inside Secure in (very) short



Manufacturer

→



Chip Embedder

→

Issuer

inside SECURE

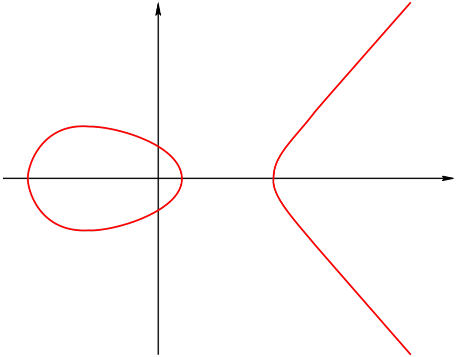
Outline

Outline

1. Elliptic Curve Cryptography
   - Generalities
   - Protocols
   - Points Representation and Formulas
   - Scalar Multiplication Algorithms
2. Side-Channel Analysis
   - Introduction
   - Simple Side-Channel Analysis
   - Differential Side-Channel Analysis
   - Fault Analysis
3. Countermeasures
   - SSCA Countermeasures
   - DSCA Countermeasures
   - FA Countermeasures
4. Conclusion

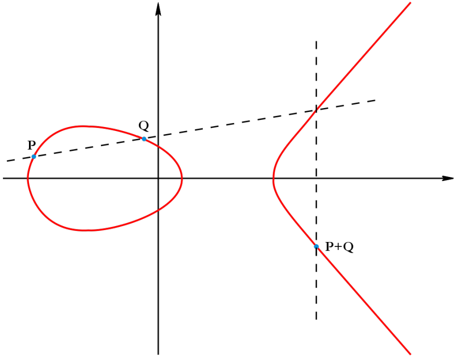
Elliptic Curve Equation

Considering a field $\mathbb{F}_p$, $p > 3$,
the points (x, y) of $\mathcal{E}/\mathbb{F}_p : y^2 = x^3 + ax + b$
and the "point at infinity" $\mathcal{O}$ form a group.



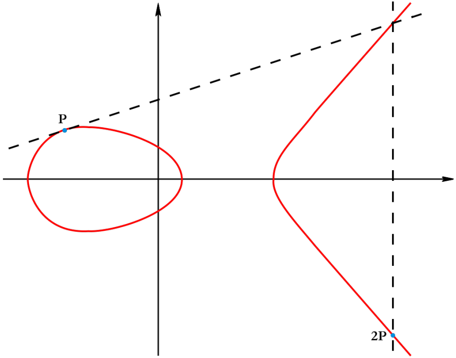
Elliptic Curve Equation

Considering a field $\mathbb{F}_p$, $p > 3$,
the points (x, y) of $\mathcal{E}/\mathbb{F}_p : y^2 = x^3 + ax + b$
and the "point at infinity" $\mathcal{O}$ form a group.



Elliptic Curve Equation

Considering a field $\mathbb{F}_p$, $p > 3$,
the points (x, y) of $\mathcal{E}/\mathbb{F}_p : y^2 = x^3 + ax + b$
and the "point at infinity" $\mathcal{O}$ form a group.



Scalar Multiplication

Given a point P in $\mathcal{E}(\mathbb{F}_p)$ and an integer k, we fix $k \cdot P = \underbrace{P + P + \cdots + P}_{k \text{ times}}$.

Scalar Multiplication

Given a point P in $\mathcal{E}(\mathbb{F}_p)$ and an integer k,
we fix $k \cdot P = \underbrace{P + P + \cdots + P}_{k \text{ times}}$.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given P in $\mathcal{E}(\mathbb{F}_p)$ and $\alpha \cdot P$, $1 \leq \alpha \leq \#\mathcal{E}(\mathbb{F}_p)$, find α ?

Much harder than DLP on finite fields, or factoring.

Scalar Multiplication

Given a point P in $\mathcal{E}(\mathbb{F}_p)$ and an integer k,
we fix $k \cdot P = \underbrace{P + P + \cdots + P}_{k \text{ times}}$.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given P in $\mathcal{E}(\mathbb{F}_p)$ and $\alpha \cdot P$, $1 \leq \alpha \leq \#\mathcal{E}(\mathbb{F}_p)$, find α ?

Much harder than DLP on finite fields, or factoring.

Security	2^{80}	2^{112}	2^{128}	2^{192}
ElGamal p/q	160/1024	224/2048	256/3072	384/8192
RSA	1024	2048	3072	8192
ECC	160	224	256	384

Keylengths for roughly equivalent security

Two Levels Arithmetic

Two Levels Arithmetic

Points group of the elliptic curve

- $\mathcal{E}(\mathbb{F}_p)$: point set
- additive law
- point additions and doublings

Two Levels Arithmetic

Points group of the elliptic curve

- $\mathcal{E}(\mathbb{F}_p)$: point set
- additive law
- point additions and doublings

Base field

- $\mathbb{F}_p$: equivalence classes of integers modulo p
- additive and multiplicative laws
- modular additions and multiplications

Embedded Devices Constraints

Efficiency

Embedded Devices Constraints

Efficiency

- Most transactions have to take less than 500 ms

Embedded Devices Constraints

Efficiency

- Most transactions have to take less than 500 ms
- Small amount of RAM

Embedded Devices Constraints

Efficiency

- Most transactions have to take less than 500 ms
- Small amount of RAM
- Very low power (then frequency) for contactless devices

Embedded Devices Constraints

Efficiency

- Most transactions have to take less than 500 ms
- Small amount of RAM
- Very low power (then frequency) for contactless devices

Arithmetic optimizations

Embedded Devices Constraints

Efficiency

- Most transactions have to take less than 500 ms
- Small amount of RAM
- Very low power (then frequency) for contactless devices

Arithmetic optimizations

- At the base field level (addition formulas, points representation)

Embedded Devices Constraints

Efficiency

- Most transactions have to take less than 500 ms
- Small amount of RAM
- Very low power (then frequency) for contactless devices

Arithmetic optimizations

- At the base field level (addition formulas, points representation)
- At the points group level (scalar multiplication algorithm)

$\mathbb{F}_p$ Operations Theoretical Cost

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion (I)

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion (I)

Significant operations

- Multiplication (M)
- Squaring (S, S/M $\approx$ 0.8)

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion (I)

Significant operations

- Multiplication (M)
- Squaring (S, S/M $\approx$ 0.8)

Negligible operations

- Addition (A)
- Subtraction (S)
- Negation (N)

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion (I)

Significant operations

- Multiplication (M)
- Squaring (S, S/M $\approx$ 0.8)

Negligible operations

- Addition (A) A/M $\approx$ 0.2 on most smart cards
- Subtraction (S)
- Negation (N)

Outline

Elliptic Curve Digital Signature Algorithm (ECDSA)

Public : $\mathcal{E}(a, b, p, n = \#\mathcal{E})$, $P \in \mathcal{E}(\mathbb{F}_p)$, H

INPUT : d and m
OUTPUT : (r, s)

Choose at random k in $[1, n-1]$
$P_1 \leftarrow k \cdot P$
$r \leftarrow x_{P_1} \mod n$
If $r \equiv 0 \mod n$ restart from the beginning
$s \leftarrow k^{-1}\left(H(m) + dr\right) \mod n$
If $s \equiv 0 \mod n$ restart from the beginning
Return (r, s)

Elliptic Curve Digital Signature Algorithm (ECDSA)

Public : $\mathcal{E}(a, b, p, n = \#\mathcal{E})$, $P \in \mathcal{E}(\mathbb{F}_p)$, H

INPUT : d and m
OUTPUT : (r, s)

Choose at random k in $[1, n-1]$
$P_1 \leftarrow k \cdot P$
$r \leftarrow x_{P_1} \mod n$
If $r \equiv 0 \mod n$ restart from the beginning
$s \leftarrow k^{-1} (H(m) + dr) \mod n$
If $s \equiv 0 \mod n$ restart from the beginning
Return (r, s)

Elliptic Curve Digital Signature Algorithm (ECDSA)

Public : $\mathcal{E}(a,b,p,n=\#\mathcal{E})$, $P \in \mathcal{E}(\mathbb{F}_p)$, H

INPUT : d and m
OUTPUT : (r,s)

Choose at random k in $[1,n-1]$
$P_1 \leftarrow k \cdot P$
$r \leftarrow x_{P_1} \mod n$
If $r \equiv 0 \mod n$ restart from the beginning
$s \leftarrow k^{-1}(H(m)+dr) \mod n$
If $s \equiv 0 \mod n$ restart from the beginning
Return (r,s)

$$d = \frac{s \cdot k - H(m)}{r} \mod n$$

Elliptic Curve Diffie-Hellman (ECDH) Key Exchange

$$\mathcal{E}(a,b,p,n),\ P \in \mathcal{E}(\mathbb{F}_p)$$

Alice		Bob
Choose at random $a \in [1, n-1]$		Choose at random $b \in [1, n-1]$
$P_a = a \cdot P$	$\longrightarrow$	P_a
P_b	$\longleftarrow$	$P_b = b \cdot P$
$P_{ab} = a \cdot P_b$		$P_{ab} = b \cdot P_a$

Elliptic Curve Diffie-Hellman (ECDH) Key Exchange

$$\mathcal{E}(a,b,p,n),\ P \in \mathcal{E}(\mathbb{F}_p)$$

Card		Terminal
Choose at random $a \in [1, n-1]$		Choose at random $b \in [1, n-1]$
$P_a = a \cdot P$	$\longrightarrow$	P_a
P_b	$\longleftarrow$	$P_b = b \cdot P$
$P_{ab} = a \cdot P_b$		$P_{ab} = b \cdot P_a$

Elliptic Curve Diffie-Hellman (ECDH) Key Exchange

$$\mathcal{E}(a,b,p,n),\ P \in \mathcal{E}(\mathbb{F}_p)$$

Card		Terminal
Choose at random $a \in [1, n-1]$		Choose at random $b \in [1, n-1]$
$P_a = a \cdot P$	→	P_a
P_b	←	$P_b = b \cdot P$
$P_{ab} = a \cdot P_b$		$P_{ab} = b \cdot P_a$

Elliptic Curve Standards over $\mathbb{F}_p$

Elliptic Curve Standards over $\mathbb{F}_p$

NIST (U.S.)

Keylengths : 192, 224, 256, 384, and 521 bits.

Elliptic Curve Standards over $\mathbb{F}_p$

NIST (U.S.)

Keylengths : 192, 224, 256, 384, and 521 bits.

Brainpool (BSI, Germany)

Keylengths : 160, 192, 224, 256, 320, 384, and 512 bits.

Elliptic Curve Standards over $\mathbb{F}_p$

NIST (U.S.)

Keylengths : 192, 224, 256, 384, and 521 bits.

Brainpool (BSI, Germany)

Keylengths : 160, 192, 224, 256, 320, 384, and 512 bits.

Other standards (ANSI, ISO, IEEE, SECG) $\rightarrow$ NIST curves

Outline

Affine Representation

A point of the curve $\mathcal{E} : y^2 = x^3 + ax + b$ is represented as (x, y).

No representation for $\mathcal{O}$

Add. : 1I + 2M + 1S, Doubl. : 1I + 2M + 2S

Affine Representation

A point of the curve $\mathcal{E} : y^2 = x^3 + ax + b$ is represented as (x, y).

No representation for $\mathcal{O}$

Add. : 1I + 2M + 1S, Doubl. : 1I + 2M + 2S

Homogeneous Projective Representation

A point is represented by an equivalence class $(X : Y : Z)$.

$(X : Y : Z)$ and $(\lambda X : \lambda Y : \lambda Z)$, $\lambda \neq 0$ represent the same point

$\mathcal{O} = (0 : 1 : 0)$

Homogeneous Projective Representation

A point is represented by an equivalence class $(X : Y : Z)$.

$(X : Y : Z)$ and $(\lambda X : \lambda Y : \lambda Z)$, $\lambda \neq 0$ represent the same point

$$\mathcal{O} = (0 : 1 : 0)$$

Aff. $\rightarrow$ Hom. conversion :

$$(x, y) \rightarrow (x : y : 1)$$

Hom. $\rightarrow$ Aff. conversion :

$$(X : Y : Z \neq 0) \rightarrow (X/Z, Y/Z)$$

Homogeneous Projective Representation

A point is represented by an equivalence class $(X : Y : Z)$.

$(X : Y : Z)$ and $(\lambda X : \lambda Y : \lambda Z)$, $\lambda \neq 0$ represent the same point

$$\mathcal{O} = (0 : 1 : 0)$$

Aff. $\rightarrow$ Hom. conversion :
$$(x, y) \rightarrow (x : y : 1)$$

Hom. $\rightarrow$ Aff. conversion :
$$(X : Y : Z \neq 0) \rightarrow (X/Z, Y/Z)$$

Add. : 12M + 2S, Doubl. : 6M + 6S

Jacobian Projective Representation

A point is represented by an equivalence class $(X : Y : Z)$.

$(X : Y : Z)$ and $(\lambda^2 X : \lambda^3 Y : \lambda Z)$, $\lambda \neq 0$ represent the same point

$\mathcal{O} = (1 : 1 : 0)$

Jacobian Projective Representation

A point is represented by an equivalence class $(X : Y : Z)$.

$(X : Y : Z)$ and $(\lambda^2 X : \lambda^3 Y : \lambda Z)$, $\lambda \neq 0$ represent the same point

$\mathcal{O} = (1 : 1 : 0)$

Aff. $\rightarrow$ Jac. conversion :

$(x, y) \rightarrow (x : y : 1)$

Jac. $\rightarrow$ Aff. conversion :

$(X : Y : Z \neq 0) \rightarrow (X/Z^2, Y/Z^3)$

Jacobian Projective Representation

A point is represented by an equivalence class $(X : Y : Z)$.

$(X : Y : Z)$ and $(\lambda^2 X : \lambda^3 Y : \lambda Z)$, $\lambda \neq 0$ represent the same point

$\mathcal{O} = (1 : 1 : 0)$

Aff. $\rightarrow$ Jac. conversion :

$(x, y) \rightarrow (x : y : 1)$

Jac. $\rightarrow$ Aff. conversion :

$(X : Y : Z \neq 0) \rightarrow (X/Z^2, Y/Z^3)$

Add. : 11M + 5S, Doubl. : 2M + 8S

Modified Jacobian Projective Representation

Introduced in [Cohen, Miyaji & Ono, *Efficient elliptic curve exponentiation using mixed coordinates*, Asiacrypt 1998].

Modified Jacobian Projective Representation

Introduced in [Cohen, Miyaji & Ono, *Efficient elliptic curve exponentiation using mixed coordinates*, Asiacrypt 1998].

Based on the Jacobian projective representation.
Plus an extra coordinate $(X : Y : Z : aZ^4)$.

Modified Jacobian Projective Representation

Introduced in [Cohen, Miyaji & Ono, *Efficient elliptic curve exponentiation using mixed coordinates*, Asiacrypt 1998].

Based on the Jacobian projective representation.
Plus an extra coordinate $(X : Y : Z : aZ^4)$.

Faster doubling than Jacobian projective : 3M + 5S
But slower addition : 13M + 7S

Outline

Double & Add Algorithm

Left-to-Right

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$,
$k = (k_{\ell-1} \dots k_1 k_0)_2$

OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$

Return Q

Double & Add Algorithm

Left-to-Right

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$,
$k = (k_{\ell-1} \dots k_1 k_0)_2$
OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$

Return Q

On average :

$$\ell \cdot \text{dbl} + \frac{\ell}{2} \cdot \text{add}$$

NAF Multiplication

NAF Representation

Signed binary representation.
Minimize the number of non-zero digits (1/3 vs 1/2).

Example :

$$187 = 10111011^{(2)} = 10\bar{1}000\bar{1}0\bar{1}^{(NAF)}$$

NAF Multiplication

NAF Representation

Signed binary representation.
Minimize the number of non-zero digits (1/3 vs 1/2).

Example :

$$187 = 10111011^{(2)} = 10\bar{1}000\bar{1}0\bar{1}^{(NAF)}$$

Interest

- Minimize the number of additions
- $P \to -P$ is cheap : $(X : Y : Z) \to (X : -Y : Z)$

NAF Multiplication

Right-to-Left

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$,
$k = (k_{\ell-1} \dots k_1 k_0)_{\mathrm{NAF}}$
OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$
$R \leftarrow P$

For i from 0 to $\ell - 1$ do
  If $k_i = 1$ then
    $Q \leftarrow Q + R$
  If $k_i = -1$ then
    $Q \leftarrow Q + (-R)$
  $R \leftarrow 2R$

Return Q

NAF Multiplication

Right-to-Left

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$,
$k = (k_{\ell-1} \dots k_1 k_0)_{\text{NAF}}$
OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$
$R \leftarrow P$

For i from 0 to $\ell - 1$ do
  If $k_i = 1$ then
    $Q \leftarrow Q + R$
  If $k_i = -1$ then
    $Q \leftarrow Q + (-R)$
  $R \leftarrow 2R$

Return Q

Cost :

$\ell \cdot \text{dbl} + \frac{\ell}{3} \cdot \text{add}$

NAF Multiplication

Right-to-Left

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$,
$k = (k_{\ell-1} \dots k_1 k_0)_{\text{NAF}}$
OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$
$R \leftarrow P$

For i from 0 to $\ell - 1$ do
  If $k_i = 1$ then
    $Q \leftarrow Q + R$
  If $k_i = -1$ then
    $Q \leftarrow Q + (-R)$
  $R \leftarrow 2R$

Return Q

Cost :

$\ell \cdot \text{dbl} + \frac{\ell}{3} \cdot \text{add}$

Variant introduced in [Joye, *Fast point multiplication on elliptic curves without precomputation*, WAIFI 2008] :

NAF Multiplication

Right-to-Left

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$, $k = (k_{\ell-1} \dots k_1 k_0)_{\text{NAF}}$
OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$
$R \leftarrow P$

For i from 0 to $\ell - 1$ do
  If $k_i = 1$ then
    $Q \leftarrow Q + R$
  If $k_i = -1$ then
    $Q \leftarrow Q + (-R)$
  $R \leftarrow 2R$

Return Q

Cost :

$$\ell \cdot \text{dbl} + \frac{\ell}{3} \cdot \text{add}$$

Variant introduced in [Joye, *Fast point multiplication on elliptic curves without precomputation*, WAIFI 2008] :

- Q in Jacobian coordinates

NAF Multiplication

Right-to-Left

INPUT : $P \in \mathcal{E}(\mathbb{F}_p)$,
$k = (k_{\ell-1} \ldots k_1 k_0)_{\text{NAF}}$
OUTPUT : $k \cdot P$

$Q \leftarrow \mathcal{O}$
$R \leftarrow P$

For i from 0 to $\ell - 1$ do
  If $k_i = 1$ then
    $Q \leftarrow Q + R$
  If $k_i = -1$ then
    $Q \leftarrow Q + (-R)$
  $R \leftarrow 2R$

Return Q

Cost :

$\ell \cdot \text{dbl} + \frac{\ell}{3} \cdot \text{add}$

Variant introduced in [Joye, *Fast point multiplication on elliptic curves without precomputation*, WAIFI 2008] :

- Q in Jacobian coordinates
- R in modified Jacobian coordinates

Other algorithms

Other algorithms

Sliding window algorithms

Precompute $3P, 5P, \ldots$ to process several scalar bits at a time.

Can be combined with the NAF method.

Other algorithms

Sliding window algorithms

Precompute $3P, 5P, \ldots$ to process several scalar bits at a time.

Can be combined with the NAF method.

DBNS, multibase NAF...

Heavy precomputations.
Too expensive for the ECDSA in the embedded context.

Other algorithms

Sliding window algorithms

Precompute $3P, 5P, \ldots$ to process several scalar bits at a time.

Can be combined with the NAF method.

DBNS, multibase NAF...

Heavy precomputations.
Too expensive for the ECDSA in the embedded context.

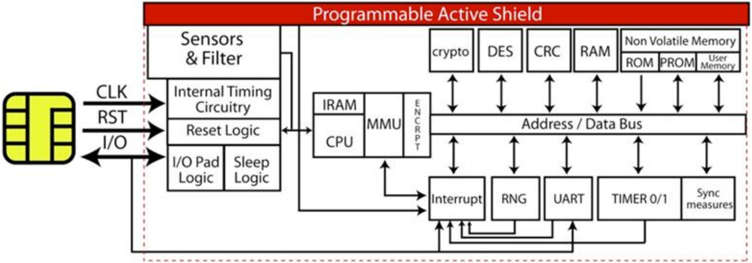
Co-Z Addition

Euclidean Addition Chains [Meloni, WAIFI 2007]
Co-Z binary ladder [Goundar, Joye & Miyaji, CHES 2010]

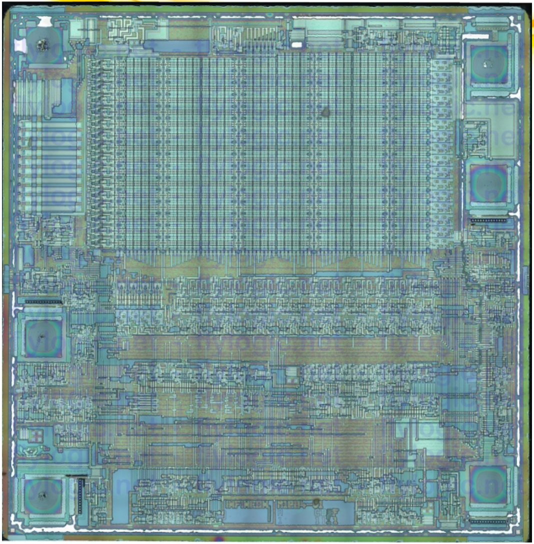
Outline

Outline

A chip in details

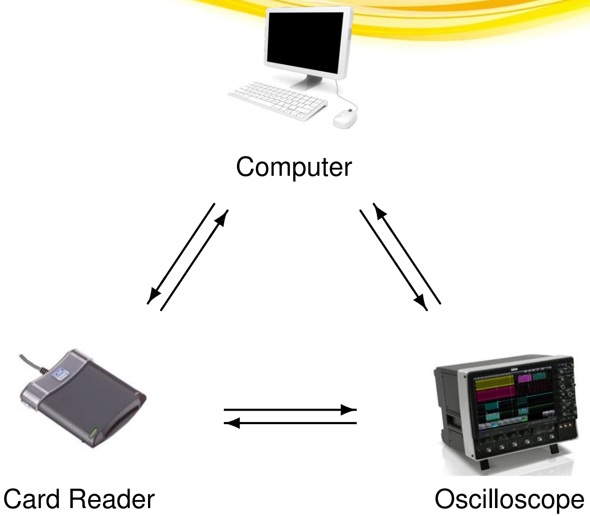


A chip in details



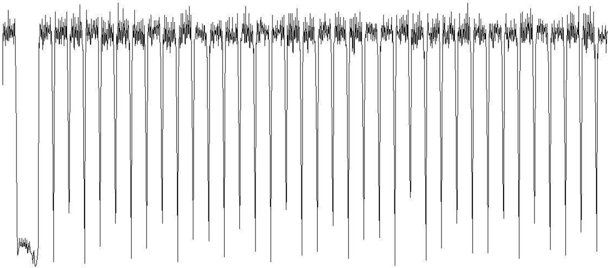
Attack Bench

Non Invasive Attacks



Simple Analyse Example

Leakage on Performed Operations



RSA exponentiation

Simple Analyse Example

Leakage on Manipulated Data

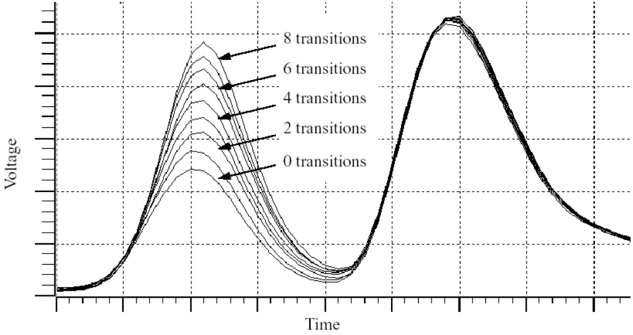


FIGURE 2. Number of Bit Transitions versus Power Consumption
These results show how the data effects the power levels. The nine overlayed waveforms correspond to the power traces of different data being accessed by an LDA instruction. These results were obtained by averaging the power signals across 500 samples in order to reduce the noise content. The difference in voltage between i transitions and i+1 transitions is about 6.5 mV.

Milestones

- Timing Attacks [Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, Crypto 1996]
- Fault Attacks [Boneh et al., *On the Importance of Checking Cryptographic Protocols for Faults*, Eurocrypt 1997]
- SPA and DPA [Kocher et al., *Differential Power Analysis*, Crypto 1999]

Milestones

- Timing Attacks [Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, Crypto 1996]
- Fault Attacks [Boneh et al., *On the Importance of Checking Cryptographic Protocols for Faults*, Eurocrypt 1997]
- SPA and DPA [Kocher et al., *Differential Power Analysis*, Crypto 1999]
- DFA on ECC [Biehl et al., *Differential Fault Attacks on Elliptic Curve Cryptosystems*, Crypto 2000]
- DPA on RSA [den Boer et al., *A DPA Attack Against the Modular Reduction within a CRT Implementation of RSA*, CHES 2002]

Milestones

- Timing Attacks [Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, Crypto 1996]
- Fault Attacks [Boneh et al., *On the Importance of Checking Cryptographic Protocols for Faults*, Eurocrypt 1997]
- SPA and DPA [Kocher et al., *Differential Power Analysis*, Crypto 1999]
- DFA on ECC [Biehl et al., *Differential Fault Attacks on Elliptic Curve Cryptosystems*, Crypto 2000]
- DPA on RSA [den Boer et al., *A DPA Attack Against the Modular Reduction within a CRT Implementation of RSA*, CHES 2002]
- CPA [Brier et al., *Correlation Power Analysis with a Leakage Model*, CHES 2004]
- CPA on PK [Amiel et al., *Power Analysis for Secret Recovering and Reverse Engineering of Public Key Algorithms*, SAC 2007]

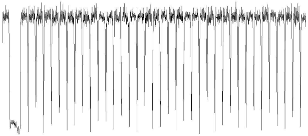
Outline

Simple Analysis Principle

Measure one side-channel leakage s function of t and consider the curve $s(t)$.

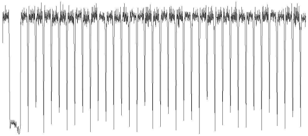
Simple Analysis Principle

Measure one side-channel leakage s function of t and consider the curve $s(t)$.



Simple Analysis Principle

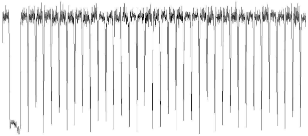
Measure one side-channel leakage s function of t and consider the curve $s(t)$.



SPA/SEMA

Simple Analysis Principle

Measure one side-channel leakage s function of t and consider the curve $s(t)$.

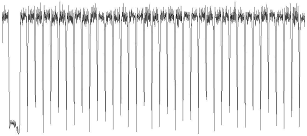


SPA/SEMA

- depicts the behavior of the chip depending on the performed operations / manipulated data

Simple Analysis Principle

Measure one side-channel leakage s function of t and consider the curve $s(t)$.



SPA/SEMA

- depicts the behavior of the chip depending on the performed operations / manipulated data
- each measure enables direct reading

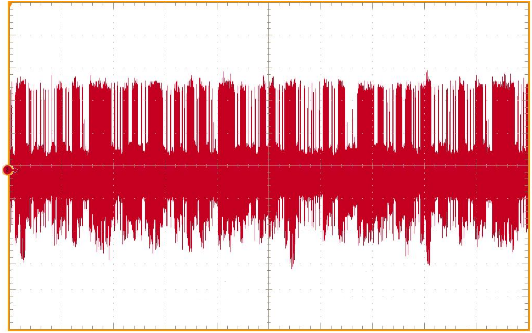
Example

Left-to-Right *Double & add* Algorithm Analysis

$Q \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$

Return Q



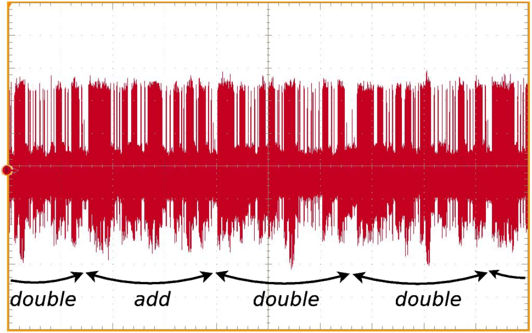
Example

Left-to-Right *Double & add* Algorithm Analysis

$Q \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
$\quad Q \leftarrow 2Q$
$\quad$ If $k_i = 1$ then
$\quad\quad Q \leftarrow Q + P$

Return Q



Outline

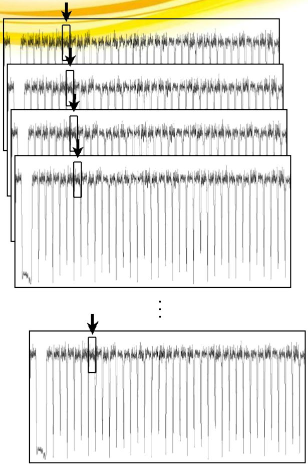
Differential Analysis Principle

Measure n times a side-channel leakage s function of t and consider the curves $s_1(t), s_2(t), \ldots, s_n(t)$.

Differential Analysis Principle

Measure n times a side-channel leakage s function of t and consider the curves $s_1(t), s_2(t), \ldots, s_n(t)$.

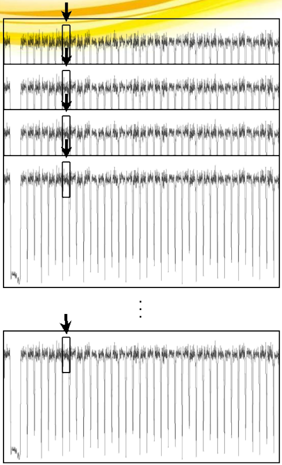
- targets a same operation on all curves but involving different data



Differential Analysis Principle

Measure n times a side-channel leakage s function of t and consider the curves $s_1(t), s_2(t), \ldots, s_n(t)$.

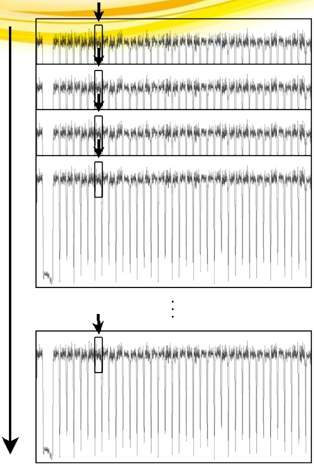
- targets a same operation on all curves but involving different data
- align vertically the curves on the targeted operation



Differential Analysis Principle

Measure n times a side-channel leakage s function of t and consider the curves $s_1(t), s_2(t), \ldots, s_n(t)$.

- targets a same operation on all curves but involving different data
- align vertically the curves on the targeted operation
- process the curves with statistical treatment



Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input.

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input.

Original DPA/DEMA

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input.

Original DPA/DEMA

- For each possible value (guess) :

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input.

Original DPA/DEMA

- For each possible value (guess) :
  - sort the curves into two sets S_0 and S_1 depending of some intermediate result

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input.

Original DPA/DEMA

- For each possible value (guess) :
  - sort the curves into two sets S_0 and S_1 depending of some intermediate result
  - average and subtract : $< S_0 > - < S_1 >$, and look for peaks

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input.

Original DPA/DEMA

- For each possible value (guess) :
  - sort the curves into two sets S_0 and S_1 depending of some intermediate result
  - average and subtract : $< S_0 > - < S_1 >$, and look for peaks
- Iterate until peaks are found

Differential Analysis

Statistical Treatment

Example

Differential Analysis

Statistical Treatment

Example

$$C_1$$
$$C_2$$
$$\vdots$$
$$C_N$$

Differential Analysis

Statistical Treatment

Example

$$\begin{array}{cc} C_1 & P_1 \\ C_2 & P_2 \\ \vdots & \vdots \\ C_N & P_N \end{array}$$

Differential Analysis

Statistical Treatment

Example

Guess : $k_i = 0$

$$\begin{array}{cc} C_1 & P_1 \\ C_2 & P_2 \\ \vdots & \vdots \\ C_N & P_N \end{array}$$

Differential Analysis

Statistical Treatment

Example

Guess : $k_i = 0$

$$\begin{array}{ccc} C_1 & P_1 & Q_1^i \\ C_2 & P_2 & Q_2^i \\ \vdots & \vdots & \vdots \\ C_N & P_N & Q_N^i \end{array}$$

Differential Analysis

Statistical Treatment

Example

Guess : $k_i = 0$

$$\begin{array}{ccccc} C_1 & P_1 & Q_1^i & \rightarrow & S_0 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_1 \end{array}$$

Differential Analysis

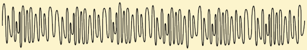
Statistical Treatment

Example

Guess : $k_i = 0$

$$\begin{array}{ccccc} C_1 & P_1 & Q_1^i & \rightarrow & S_0 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_1 \end{array}$$

Compute $< S_0 > - < S_1 >$:



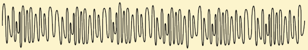
Differential Analysis

Statistical Treatment

Example

Guess : $k_i = 0$

$$\begin{array}{ccccc} C_1 & P_1 & Q_1^i & \rightarrow & S_0 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_1 \end{array}$$

Compute $< S_0 > - < S_1 >$: 

Differential Analysis

Statistical Treatment

Example

Guess : $k_i = 1$

$$\begin{array}{ccc} C_1 & P_1 & Q_1^i \\ C_2 & P_2 & Q_2^i \\ \vdots & \vdots & \vdots \\ C_N & P_N & Q_N^i \end{array}$$

Differential Analysis

Statistical Treatment

Example

$$\text{Guess} : k_i = 1$$

$$\begin{array}{ccccc} C_1 & P_1 & Q_1^i & \rightarrow & S_1 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_0 \end{array}$$

Differential Analysis

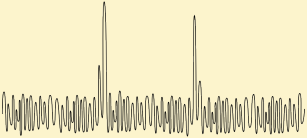
Statistical Treatment

Example

Guess : $k_i = 1$

$$\begin{array}{ccccc} C_1 & P_1 & Q_1^i & \rightarrow & S_1 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_0 \end{array}$$

Compute $< S_0 > - < S_1 >$:



Differential Analysis

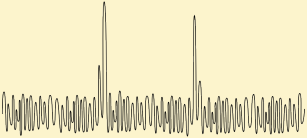
Statistical Treatment

Example

Guess : $k_i = 1$

$$\begin{array}{ccccc} C_1 & P_1 & Q_1^i & \rightarrow & S_1 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_0 \end{array}$$

Compute $< S_0 > - < S_1 >$:



Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input (as DPA).

CPA/CEMA

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input (as DPA).

CPA/CEMA

- For each possible value (guess) :

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input (as DPA).

CPA/CEMA

- For each possible value (guess) :
  - compute correlation curves between s_i and HW of some intermediate result depending on the guess

Differential Analysis

Statistical Treatment

Depending on some known and variable input of the algorithm and of a few bits of the secret input (as DPA).

CPA/CEMA

- For each possible value (guess) :
  - compute correlation curves between s_i and HW of some intermediate result depending on the guess
  - average the correlation curves and apply a threshold
- Iterate until the threshold is reached

Outline

1. Elliptic Curve Cryptography
   - Generalities
   - Protocols
   - Points Representation and Formulas
   - Scalar Multiplication Algorithms
2. Side-Channel Analysis
   - Introduction
   - Simple Side-Channel Analysis
   - Differential Side-Channel Analysis
   - **Fault Analysis**
3. Countermeasures
   - SSCA Countermeasures
   - DSCA Countermeasures
   - FA Countermeasures
4. Conclusion

Fault Attacks on Scalar Multiplication

Fault Attacks on Scalar Multiplication

- Inject a fault : $x_P \leftarrow x_{P'}$

Fault Attacks on Scalar Multiplication

- Inject a fault : $x_P \leftarrow x_{P'}$
- Since b is not involved in the scalar multiplication, $P' \in \mathcal{E}'(\mathbb{F}_p)$, with $\mathcal{E}' : y^2 = x^3 + ax + b'$ and $b' = {y_P}^2 - {x'_P}^3 - ax'_P$

Fault Attacks on Scalar Multiplication

- Inject a fault : $x_P \leftarrow x_{P'}$
- Since b is not involved in the scalar multiplication, $P' \in \mathcal{E}'(\mathbb{F}_p)$, with $\mathcal{E}' : y^2 = x^3 + ax + b'$ and $b' = {y_P}^2 - {x'_P}^3 - ax'_P$
- Then the scalar multiplication $Q' = k \cdot P'$ takes place on $\mathcal{E}'$

Fault Attacks on Scalar Multiplication

- Inject a fault : $x_P \leftarrow x_{P'}$
- Since b is not involved in the scalar multiplication, $P' \in \mathcal{E}'(\mathbb{F}_p)$, with $\mathcal{E}' : y^2 = x^3 + ax + b'$ and $b' = {y_P}^2 - {x'_P}^3 - ax'_P$
- Then the scalar multiplication $Q' = k \cdot P'$ takes place on $\mathcal{E}'$
- DLP for $Q' = k \cdot P'$ is easy to solve if $\text{ord}_{\mathcal{E}'}(P')$ is small

Fault Attacks on Scalar Multiplication

- Inject a fault : $x_P \leftarrow x_{P'}$
- Since b is not involved in the scalar multiplication, $P' \in \mathcal{E}'(\mathbb{F}_p)$, with $\mathcal{E}' : y^2 = x^3 + ax + b'$ and $b' = {y_P}^2 - {x'_P}^3 - ax'_P$
- Then the scalar multiplication $Q' = k \cdot P'$ takes place on $\mathcal{E}'$
- DLP for $Q' = k \cdot P'$ is easy to solve if $\text{ord}_{\mathcal{E}'}(P')$ is small
- Iterate and apply the chinese reminder theorem to recover k.

Outline

Outline

SPA/SEMA Protection

SPA/SEMA Protection

Mostly three kinds of countermeasures :

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas
  - Homogeneous projective coordinates [Brier & Joye, 2002]

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas
  - Homogeneous projective coordinates [Brier & Joye, 2002]
  - Specific curves formulas (Hessian, Edwards, etc.)

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas
  - Homogeneous projective coordinates [Brier & Joye, 2002]
  - Specific curves formulas (Hessian, Edwards, etc.)
- Atomicity

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas
  - Homogeneous projective coordinates [Brier & Joye, 2002]
  - Specific curves formulas (Hessian, Edwards, etc.)
- Atomicity
  - Original ECC pattern [Chevallier et al., 2003]

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas
  - Homogeneous projective coordinates [Brier & Joye, 2002]
  - Specific curves formulas (Hessian, Edwards, etc.)
- Atomicity
  - Original ECC pattern [Chevallier et al., 2003]
  - Longa ECC patterns [Longa, 2007]

SPA/SEMA Protection

Mostly three kinds of countermeasures :

- Regular algorithms
  - Dummy curve operations : Double and Add Always [Coron, 1999]
  - Highly regular : Montgomery ladder [Montgomery, 1987]
- Unified formulas
  - Homogeneous projective coordinates [Brier & Joye, 2002]
  - Specific curves formulas (Hessian, Edwards, etc.)
- Atomicity
  - Original ECC pattern [Chevallier et al., 2003]
  - Longa ECC patterns [Longa, 2007]
  - Improved ECC pattern [Giraud and Verneuil, 2010]

Regular Algorithms

Double & add always

$Q, T \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$
  Else
    $T \leftarrow Q + P$

Return Q

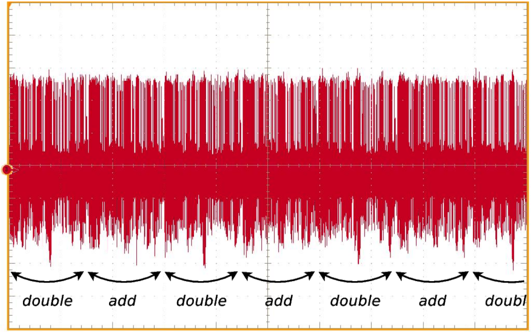
Regular Algorithms

Double & add always

$Q, T \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$
  Else
    $T \leftarrow Q + P$

Return Q



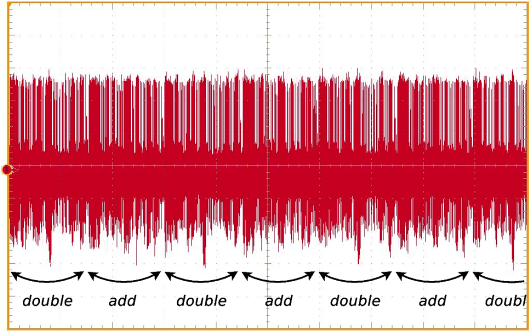
Regular Algorithms

Double & add always

$Q, T \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$
  Else
    $T \leftarrow Q + P$

Return Q



On average :

$\ell \cdot \mathrm{dbl} + \ell \cdot \mathrm{add}$

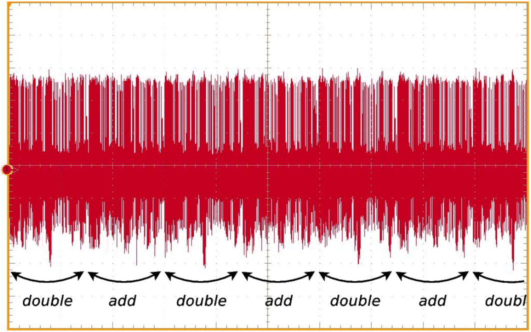
Regular Algorithms

Double & add always

$Q, T \leftarrow \mathcal{O}$

For i from $\ell - 1$ to 0 do
  $Q \leftarrow 2Q$
  If $k_i = 1$ then
    $Q \leftarrow Q + P$
  Else
    $T \leftarrow Q + P$

Return Q



On average :

$\ell \cdot \text{dbl} + \ell \cdot \text{add}$

Prone to safe errors.

Regular Algorithms

Montgomery ladder

$Q_1 \leftarrow P$
$Q_2 \leftarrow 2P$

For i from $l-2$ to 0 do

$Q_{1-k_i} \leftarrow Q_1 + Q_2$
$Q_{k_i} \leftarrow 2Q_i$

Return Q_1

Regular Algorithms

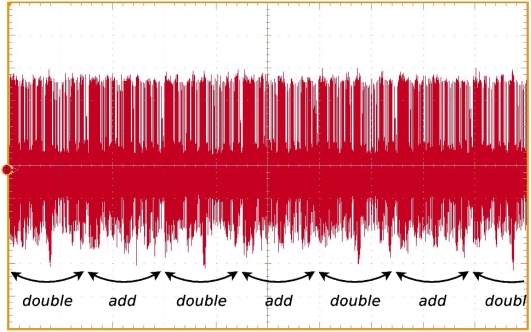
Montgomery ladder

$Q_1 \leftarrow P$
$Q_2 \leftarrow 2P$

For i from $l-2$ to 0 do

$Q_{1-k_i} \leftarrow Q_1 + Q_2$
$Q_{k_i} \leftarrow 2Q_i$

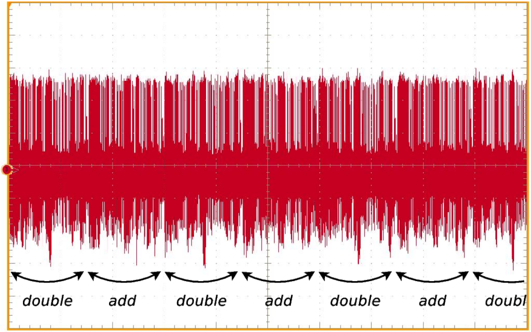
Return Q_1



Regular Algorithms

Montgomery ladder

$Q_1 \leftarrow P$
$Q_2 \leftarrow 2P$
For i from $l-2$ to 0 do
$Q_{1-k_i} \leftarrow Q_1 + Q_2$
$Q_{k_i} \leftarrow 2Q_i$
Return Q_1



Trick :

Y_1 and Y_2 computation can be avoided.

- Brier & Joye, PKC 2002
- Izu & Takagi, PKC 2002
- Fischer et al., ePrint 2002

Unified Formulas

Unified Formulas

A single formula for addition and doubling

Unified Formulas

A single formula for addition and doubling

- Homogeneous projective coordinates : 12M + 6S

Unified Formulas

A single formula for addition and doubling

- Homogeneous projective coordinates : 12M + 6S
- Edwards curves : 10M + 1S

Unified Formulas

A single formula for addition and doubling

- Homogeneous projective coordinates : 12M + 6S
- Edwards curves : 10M + 1S in $\mathbb{F}_{p^6}$ with standard curves :(

Unified Formulas

A single formula for addition and doubling

- Homogeneous projective coordinates : 12M + 6S
- Edwards curves : 10M + 1S in $\mathbb{F}_{p^6}$ with standard curves :(
- Twisted Edwards curves : 9M + 1S

Unified Formulas

A single formula for addition and doubling

- Homogeneous projective coordinates : 12M + 6S
- Edwards curves : 10M + 1S in $\mathbb{F}_{p^6}$ with standard curves :(
- Twisted Edwards curves : 9M + 1S in $\mathbb{F}_{p^3}$ with standard curves :(

Atomicity

Introduced in [Chevallier-Mames, Ciet & Joye, *Low-cost solutions for preventing simple side-channel analysis...*, ePrint 2003].

Atomicity

Introduced in [Chevallier-Mames, Ciet & Joye, *Low-cost solutions for preventing simple side-channel analysis...*, ePrint 2003].

Idea : always repeat the same pattern of operations

Atomicity

Introduced in [Chevallier-Mames, Ciet & Joye, *Low-cost solutions for preventing simple side-channel analysis...*, ePrint 2003].

Idea : always repeat the same pattern of operations

Example : RSA (*square & multiply*)

- S, M, S, S, S, M, S, S, M, S, M, ...

Atomicity

Introduced in [Chevallier-Mames, Ciet & Joye, *Low-cost solutions for preventing simple side-channel analysis...*, ePrint 2003].

Idea : always repeat the same pattern of operations

Example : RSA (*square & multiply*)

- S, M, S, S, S, M, S, S, M, S, M, ...
- M, M, M, M, M, M, M, M, M, M, M, ...

Atomicity

Introduced in [Chevallier-Mames, Ciet & Joye, *Low-cost solutions for preventing simple side-channel analysis...*, ePrint 2003].

Idea : always repeat the same pattern of operations

Example : RSA (*square & multiply*)

- S, M, S, S, S, M, S, S, M, S, M, ...
- M, M, M, M, M, M, M, M, M, M, M, ...

$\rightarrow$ Cost

Atomicity for Elliptic Curves

Atomicity for Elliptic Curves

Principle

Atomicity for Elliptic Curves

Principle

Always repeat the same pattern :

Atomicity for Elliptic Curves

Principle

Always repeat the same pattern :

- Multiplication
- Addition
- Negation
- Addition

Atomicity for Elliptic Curves

Principle

Always repeat the same pattern :

- Multiplication
- Addition
- Negation
- Addition

- Multiplication
- Addition
- Negation
- Addition

Atomicity for Elliptic Curves

Principle

Always repeat the same pattern :

- Multiplication
- Addition
- Negation
- Addition

- Multiplication
- Addition
- Negation
- Addition

...

Atomicity for Elliptic Curves

Principle

Always repeat the same pattern :

- Multiplication
- Addition
- Negation
- Addition

- Multiplication
- Addition
- Negation
- Addition

...

No more squarings :(

Atomicity for Elliptic Curves

Principle

Always repeat the same pattern :

- Multiplication
- Addition
- Negation
- Addition

- Multiplication
- Addition
- Negation
- Addition

...

No more squarings :(

Many dummy additions/negations :(

Longa Atomicity

Longa Atomicity

Other patterns

In [Longa, *Accelerating the Scalar Multiplication on Elliptic Curve Cryptosystems over Prime Fields*, 2007] are proposed 2 new patterns :

Longa Atomicity

Other patterns

In [Longa, *Accelerating the Scalar Multiplication on Elliptic Curve Cryptosystems over Prime Fields*, 2007] are proposed 2 new patterns :

- Multiplication
- Negation
- Addition
- Multiplication
- Negation
- Addition
- Addition

Longa Atomicity

Other patterns

In [Longa, *Accelerating the Scalar Multiplication on Elliptic Curve Cryptosystems over Prime Fields*, 2007] are proposed 2 new patterns :

- Multiplication
- Negation
- Addition
- Multiplication
- Negation
- Addition
- Addition

- Squaring
- Negation
- Addition
- Multiplication
- Negation
- Addition
- Addition

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Two steps

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Two steps

- First define the largest atomic pattern possible

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Two steps

- First define the largest atomic pattern possible
- Then remove as many possible dummy operations

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Two steps

- First define the largest atomic pattern possible
- Then remove as many possible dummy operations

Advantages

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Two steps

- First define the largest atomic pattern possible
- Then remove as many possible dummy operations

Advantages

- Potentially applicable to every algorithm (no curve restriction)

Atomicity Improvement

Full paper : [Giraud & Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*, CARDIS 2010]

Two steps

- First define the largest atomic pattern possible
- Then remove as many possible dummy operations

Advantages

- Potentially applicable to every algorithm (no curve restriction)
- Prevents from the SPA at a lower cost than classical atomicity

Atomic Joye's Multiplication

Best pattern

	Add. 1	Add. 2	Dbl.
Sq.	$R_1 \leftarrow {Z_2}^2$	$R_1 \leftarrow {R_6}^2$	$R_1 \leftarrow {X_1}^2$
Add.	$\star$	$\star$	$R_2 \leftarrow Y_1 + Y_1$
Mult.	$R_2 \leftarrow Y_1 \cdot Z_2$	$R_4 \leftarrow R_5 \cdot R_1$	$Z_2 \leftarrow R_2 \cdot Z_1$
Add.	$\star$	$\star$	$R_4 \leftarrow R_1 + R_1$
Mult.	$R_5 \leftarrow Y_2 \cdot Z_1$	$R_5 \leftarrow R_1 \cdot R_6$	$R_3 \leftarrow R_2 \cdot Y_1$
Add.	$\star$	$\star$	$R_6 \leftarrow R_3 + R_3$
Mult.	$R_3 \leftarrow R_1 \cdot R_2$	$R_1 \leftarrow Z_1 \cdot R_6$	$R_2 \leftarrow R_6 \cdot R_3$
Add.	$\star$	$\star$	$R_1 \leftarrow R_4 + R_1$
Add.	$\star$	$\star$	$R_1 \leftarrow R_1 + W_1$
Sq.	$R_4 \leftarrow {Z_1}^2$	$R_6 \leftarrow {R_2}^2$	$R_3 \leftarrow {R_1}^2$
Mult.	$R_2 \leftarrow R_5 \cdot R_4$	$Z_3 \leftarrow R_1 \cdot Z_2$	$R_4 \leftarrow R_6 \cdot X_1$
Add.	$\star$	$R_1 \leftarrow R_4 + R_4$	$R_5 \leftarrow W_1 + W_1$
Sub.	$R_2 \leftarrow R_2 - R_3$	$R_6 \leftarrow R_6 - R_1$	$R_3 \leftarrow R_3 - R_4$
Mult.	$R_5 \leftarrow R_1 \cdot X_1$	$R_1 \leftarrow R_5 \cdot R_3$	$W_2 \leftarrow R_2 \cdot R_5$
Sub.	$\star$	$X_3 \leftarrow R_6 - R_5$	$X_2 \leftarrow R_3 - R_4$
Sub.	$\star$	$R_4 \leftarrow R_4 - X_3$	$R_6 \leftarrow R_4 - X_2$
Mult.	$R_6 \leftarrow X_2 \cdot R_4$	$R_3 \leftarrow R_4 \cdot R_2$	$R_4 \leftarrow R_6 \cdot R_1$
Sub.	$R_6 \leftarrow R_6 - R_5$	$Y_3 \leftarrow R_3 - R_1$	$Y_2 \leftarrow R_4 - R_2$

Outline

DPA/DEMA Protection

DPA/DEMA Protection

Classical countermeasures :

DPA/DEMA Protection

Classical countermeasures :

- Scalar blinding : $k' = k + r\#\mathcal{E}(\mathbb{F}_p)$

DPA/DEMA Protection

Classical countermeasures :

- Scalar blinding : $k' = k + r\#\mathcal{E}(\mathbb{F}_p)$
- Point coordinates blinding : $(X : Y : Z) = (r^2X : r^3Y : rZ)$, $r \neq 0$

DPA/DEMA Protection

Classical countermeasures :

- Scalar blinding : $k' = k + r\#\mathcal{E}(\mathbb{F}_p)$
- Point coordinates blinding : $(X : Y : Z) = (r^2X : r^3Y : rZ)$, $r \neq 0$
- Random curve isomorphism :
  $a' \leftarrow r^4a$
  $b' \leftarrow r^6b$
  $P' \leftarrow (r^2X_P, r^3Y_P, rZ_P)$
  $Q \leftarrow (x_{Q'}/r^2, y_{Q'}/r^3)$

Outline

Fault Protection

Fault Protection

Classical countermeasures :

Fault Protection

Classical countermeasures :

- Redundancy, verification...

Fault Protection

Classical countermeasures :

- Redundancy, verification...
- Verify that $P, Q \in \mathcal{E}(\mathbb{F}_p)$.

Outline

Conclusion

Conclusion

- Scalar multuplication efficiency has been extensively studied.

Conclusion

- Scalar multuplication efficiency has been extensively studied.
- Edwards curve standardization ?

Conclusion

- Scalar multuplication efficiency has been extensively studied.
- Edwards curve standardization ?
- Research on side-channel attacks keeps progressing.

Conclusion

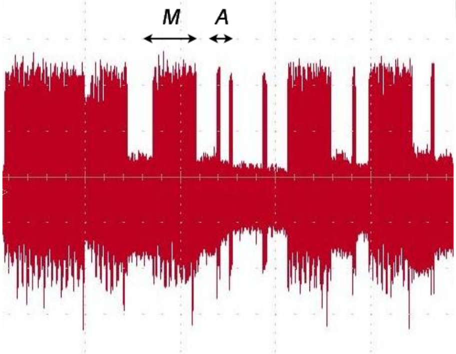
- Scalar multuplication efficiency has been extensively studied.
- Edwards curve standardization ?
- Research on side-channel attacks keeps progressing.
- Using security models for proving the resistance against attacks ?

Thank you for your attention !

Contact :

vverneuil@insidefr.com
www.math.u-bordeaux1.fr/~vverneui/

Additions Cost on a Chip



192-bit integers

A/M $\approx$ 0.2, S $=$ A, and N/M $\approx$ 0.1