Elliptic Curve Cryptography on Embedded Devices

Scalar Multiplication and Side-Channel Attacks

Vincent Verneuil

NXP Semiconductors

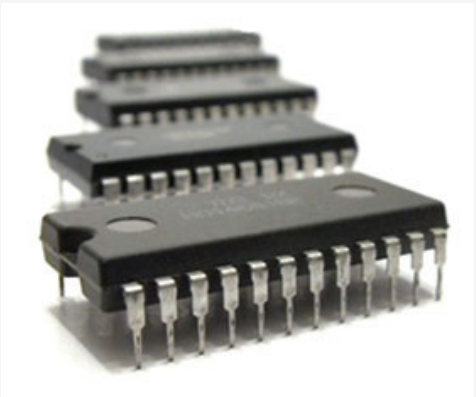
November 18, 2016

Outline

Outline

Outline

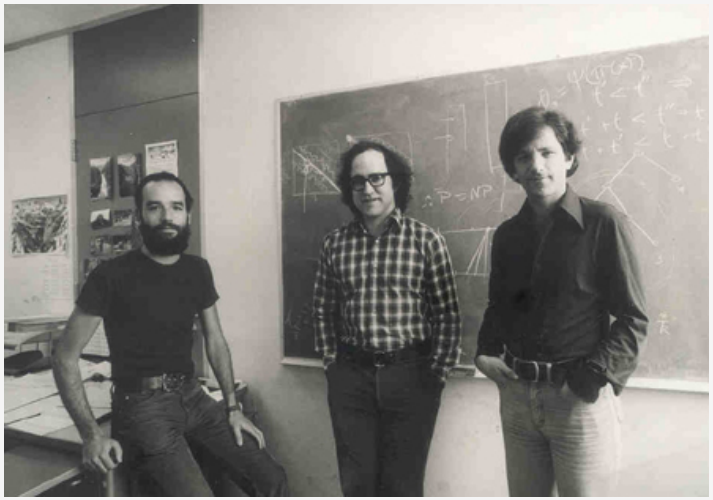
Cryptography



"Practice and study of techniques for secure communication in the presence of third parties."

- Symmetric cryptography
  - Known since the Antiquity
  - Secret key
  - Complex manipulations between the message and the key
  - Problem: key exchange
- Asymmetric cryptography
  - Invented during the 70's
  - Public key and private key
  - Rely on "difficult" mathematicals problems
  - Computations require more resources

RSA (Rivest-Shamir-Adleman)



A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, 1978.

Key generation

- pick at random two primes p and q, and compute $N = p \times q$
- choose e and compute d such that: $e \times d = 1 \mod (p-1)(q-1)$

Public key

🔑 $= \{N, e\}$

Private key

🔑 $= \{p, q, d\}$

RSA (Rivest-Shamir-Adleman)

Encryption / Decryption

To encrypt a message m:

$$c = m^e \mod N$$

To decrypt c:

$$m = c^d \mod N$$

Security assumptions

1. Given 🔑 $= \{N, e\}$, recovering $d = e^{-1} \mod (p-1)(q-1)$ requires to factorize $N = p \times q$
2. Factorization is a difficult problem

Group structure

Set $\mathbb{G}$ provided with an internal operation $\oplus$ such that:

- $\oplus$ is associative: $(x \oplus y) \oplus z = x \oplus (y \oplus z)$
- there is an identity element $0_{\mathbb{G}}$: $\forall x \in \mathbb{G}, x \oplus 0_{\mathbb{G}} = 0_{\mathbb{G}} \oplus x = x$
- each element x has an inverse y in $\mathbb{G}$: $x \oplus y = 0_{\mathbb{G}}$

Group structure

Set $\mathbb{G}$ provided with an internal operation $\oplus$ such that:

- $\oplus$ is associative: $(x \oplus y) \oplus z = x \oplus (y \oplus z)$
- there is an identity element $0_{\mathbb{G}}$: $\forall x \in \mathbb{G}, x \oplus 0_{\mathbb{G}} = 0_{\mathbb{G}} \oplus x = x$
- each element x has an inverse y in $\mathbb{G}$: $x \oplus y = 0_{\mathbb{G}}$

Examples

- $(\mathbb{R}, +)$, identity element : 0
- $(\mathbb{R} \setminus \{0\}, \times)$, identity element : 1
- $(\mathbb{Z}/n\mathbb{Z}, +) \cong \{0, 1, 2, 3, ..., n-1\}$, identity element : 0
- $(\mathbb{Z}/p\mathbb{Z} \setminus \{0\}, \times) \cong \{1, 2, 3, ..., p-1\}$, identity element : 1

Discrete Logarithm Problem (DLP)

Statement

Let $(\mathbb{G}, \times)$ be a multiplicative group, $x \in \mathbb{G}$, $y \in \langle x \rangle$.
Given x and y, how to compute α s.t. $y = x^{\alpha}$?

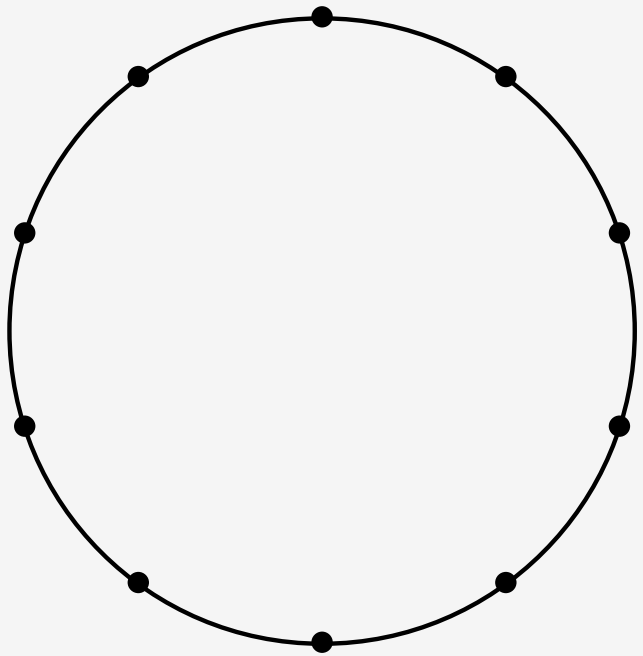
Computational complexity

Open problem: general case is NP-intermediate.

Discrete Logarithm Problem (DLP)

Example

$\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z} \setminus \{0\}$
$p = 11$, $x = 8$

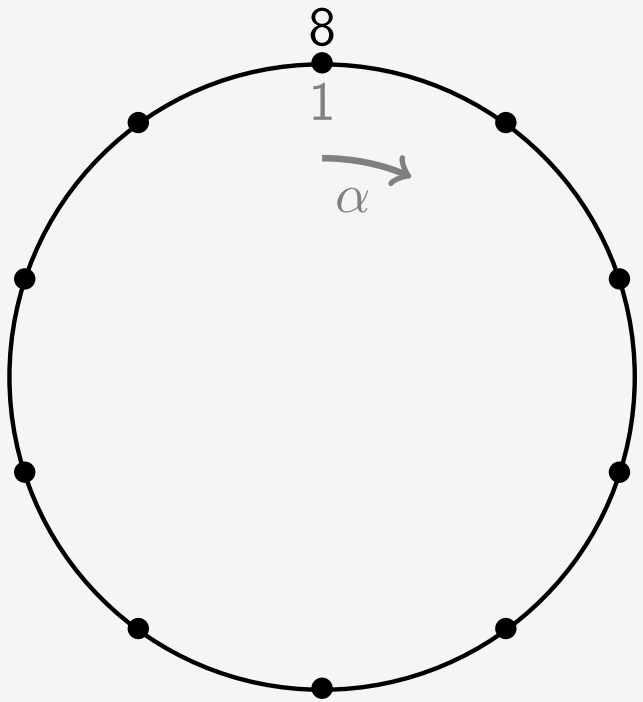


Discrete Logarithm Problem (DLP)

Example

$\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z} \setminus \{0\}$
$p = 11$, $x = 8$

- $8^1 = 8 \mod 11$

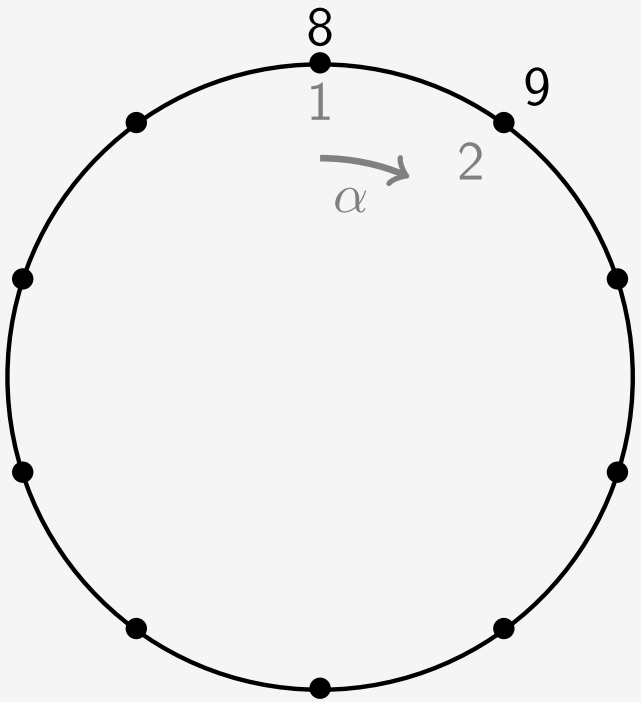


Discrete Logarithm Problem (DLP)

Example

$\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z} \setminus \{0\}$
$p = 11$, $x = 8$

- $8^1 = 8 \mod 11$
- $8^2 = 9 \mod 11$

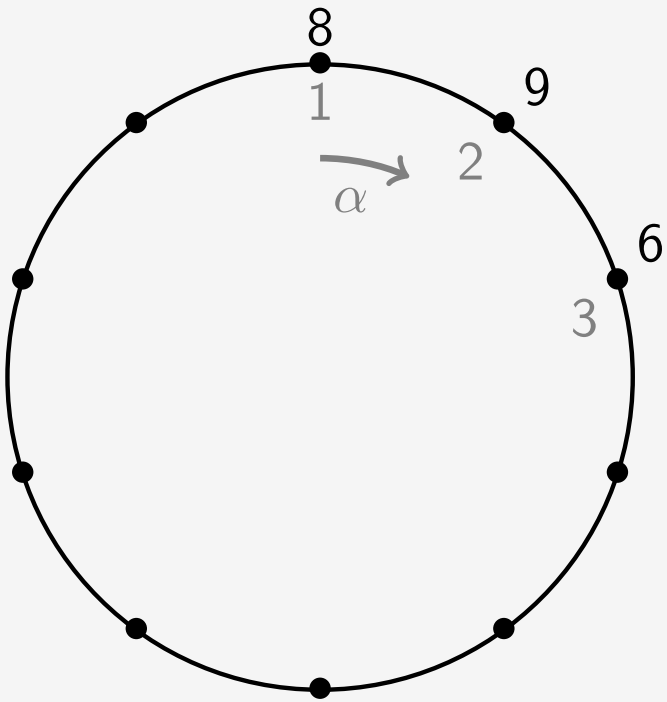


Discrete Logarithm Problem (DLP)

Example

$\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z} \setminus \{0\}$
$p = 11$, $x = 8$

- $8^1 = 8 \mod 11$
- $8^2 = 9 \mod 11$
- $8^3 = 6 \mod 11$

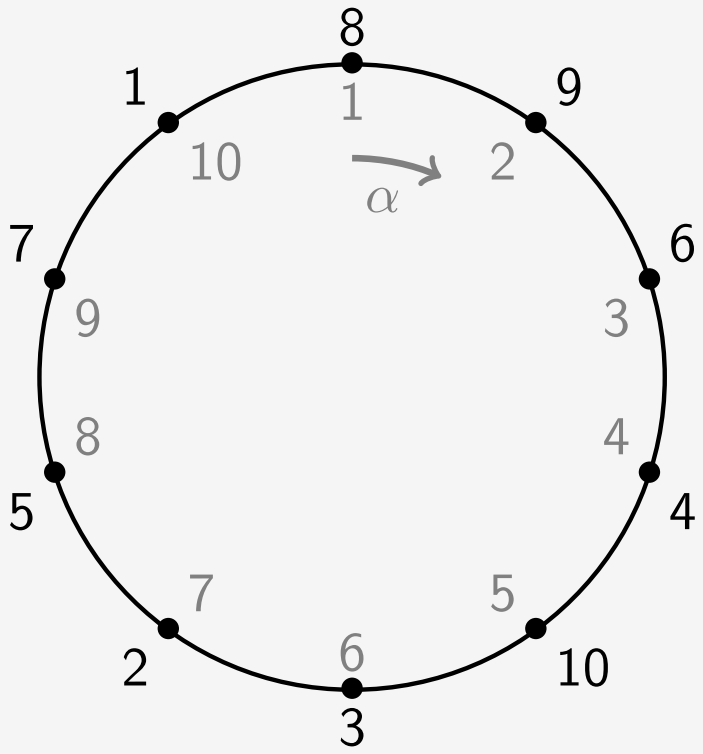


Discrete Logarithm Problem (DLP)

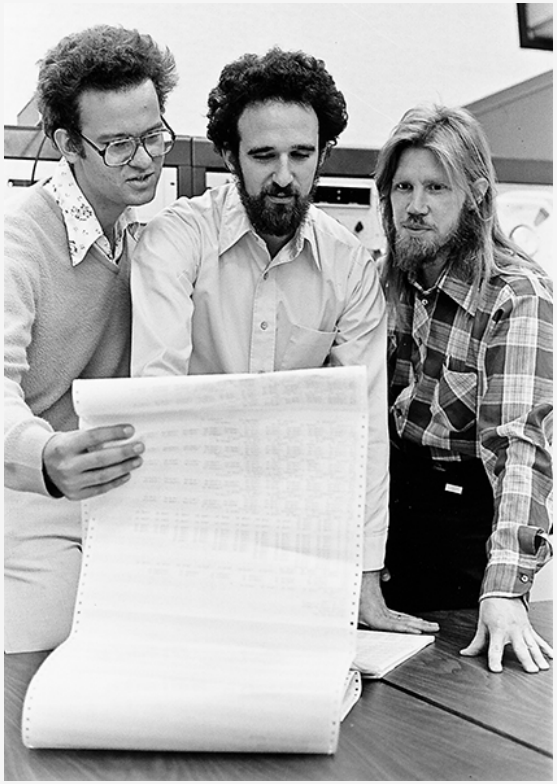
Example

$\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z} \setminus \{0\}$
$p = 11$, $x = 8$

- $8^1 = 8 \mod 11$
- $8^2 = 9 \mod 11$
- $8^3 = 6 \mod 11$
- $8^4 = 4 \mod 11$
- $8^5 = 10 \mod 11$
- $8^6 = 3 \mod 11$
- $8^7 = 2 \mod 11$
- $8^8 = 5 \mod 11$
- $8^9 = 7 \mod 11$
- $8^{10} = 1 \mod 11$
- $\mathbf{8^{\alpha+10} = 8^{\alpha} \mod 11}$

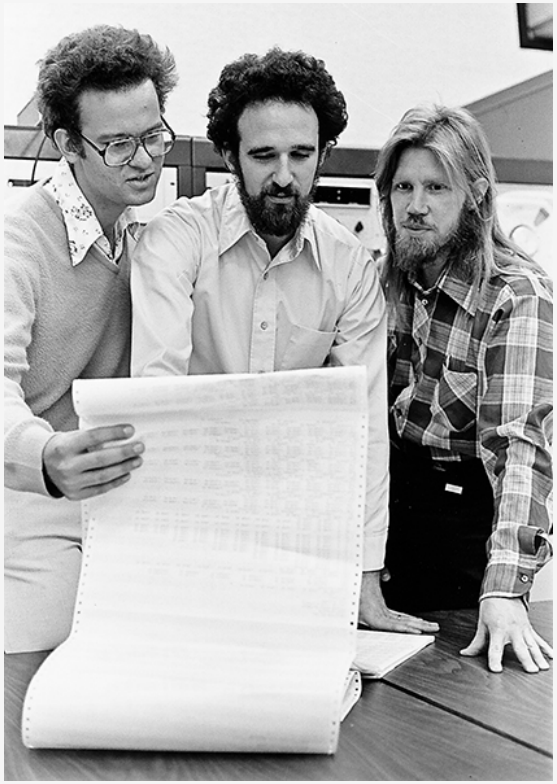


Diffie-Hellman(-Merkle) key exchange



New Directions in Cryptography, 1976.

Diffie-Hellman(-Merkle) key exchange



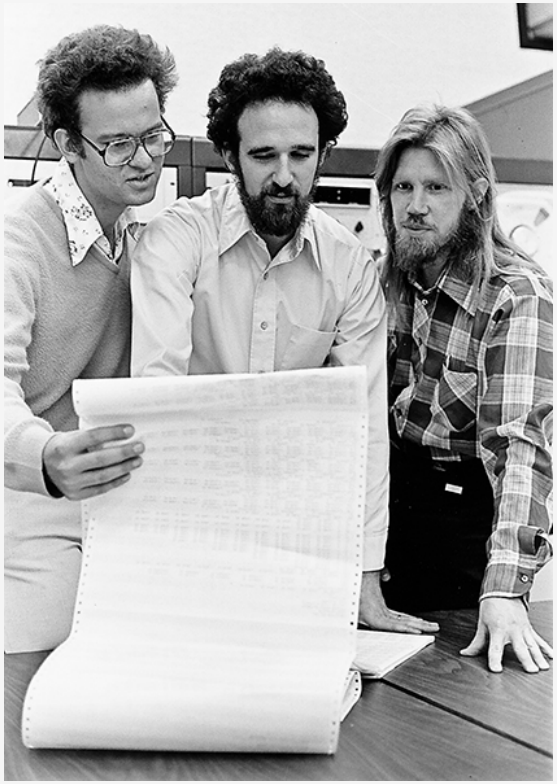
New Directions in Cryptography, 1976.

Key exchange

Given $g \in \mathbb{F}_p^*$, $g \neq 1$:

Alice	Bob
$a = \text{rand}(1, p-1)$	$b = \text{rand}(1, p-1)$
$g_a = g^a \mod p$	$g_b = g^b \mod p$

Diffie-Hellman(-Merkle) key exchange



New Directions in Cryptography, 1976.

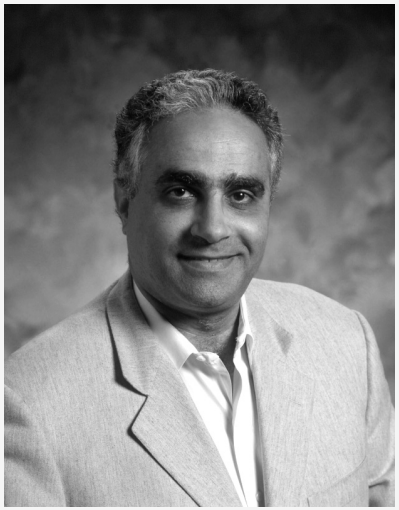
Key exchange

Given $g \in \mathbb{F}_p^*$, $g \neq 1$:

Alice		Bob
$a = \text{rand}(1, p-1)$		$b = \text{rand}(1, p-1)$
$g_a = g^a \mod p$		$g_b = g^b \mod p$
	$\xrightarrow{g_a}$	
	$\xleftarrow{g_b}$	
$g_{ab} = {g_b}^a \mod p$		$g_{ab} = {g_a}^b \mod p$

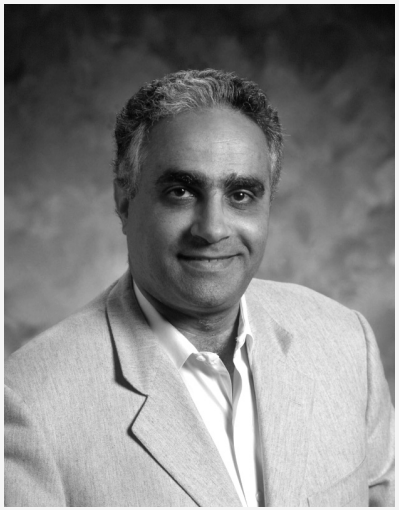
Alice and Bob now share the secret g_{ab}.

The Elgamal cryptosystem



A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, 1984.

The Elgamal cryptosystem



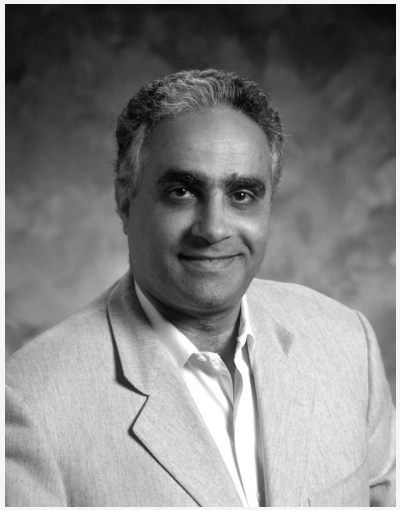
A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, 1984.

Key generation

Given $g \in \mathbb{F}_p^*$, $g \neq 1$:

- pick at random α in $[1, p-1]$
- compute $x = g^\alpha \mod p$

The Elgamal cryptosystem



A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, 1984.

Key generation

Given $g \in \mathbb{F}_p^*$, $g \neq 1$:

- pick at random α in $[1, p-1]$
- compute $x = g^\alpha \mod p$

Public key

 $= \{x\}$

Private key

 $= \{\alpha\}$

Outline

Elliptic Curve Cryptography

Independently introduced by Koblitz and Miller in 1985.

Field structure

Set $\mathbb{K}$ provided with two internal operations $\oplus, \otimes$ such that:

- $(\mathbb{K}, \oplus)$ is an additive group with identity elt. $0_{\mathbb{K}}$
- $(\mathbb{K} \setminus \{0_{\mathbb{K}}\}, \otimes)$ is a multiplicative group
- $\otimes$ is distributive over $\oplus$: $\forall a, b, c \in \mathbb{K}, \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Field structure

Set $\mathbb{K}$ provided with two internal operations $\oplus, \otimes$ such that:

- $(\mathbb{K}, \oplus)$ is an additive group with identity elt. $0_{\mathbb{K}}$
- $(\mathbb{K} \setminus \{0_{\mathbb{K}}\}, \otimes)$ is a multiplicative group
- $\otimes$ is distributive over $\oplus$: $\forall a, b, c \in \mathbb{K}, \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Examples

- $(\mathbb{R}, +, \times)$

Field structure

Set $\mathbb{K}$ provided with two internal operations $\oplus, \otimes$ such that:

- $(\mathbb{K}, \oplus)$ is an additive group with identity elt. $0_{\mathbb{K}}$
- $(\mathbb{K} \setminus \{0_{\mathbb{K}}\}, \otimes)$ is a multiplicative group
- $\otimes$ is distributive over $\oplus$: $\forall a, b, c \in \mathbb{K}, \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Examples

- $(\mathbb{R}, +, \times)$
- $(\mathbb{Z}/n\mathbb{Z}, +, \times)$?

Field structure

Set $\mathbb{K}$ provided with two internal operations $\oplus, \otimes$ such that:

- $(\mathbb{K}, \oplus)$ is an additive group with identity elt. $0_{\mathbb{K}}$
- $(\mathbb{K} \setminus \{0_{\mathbb{K}}\}, \otimes)$ is a multiplicative group
- $\otimes$ is distributive over $\oplus$: $\forall a, b, c \in \mathbb{K}, \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Examples

- $(\mathbb{R}, +, \times)$
- $(\mathbb{Z}/n\mathbb{Z}, +, \times)$? Not in the general case !

Field structure

Set $\mathbb{K}$ provided with two internal operations $\oplus, \otimes$ such that:

- $(\mathbb{K}, \oplus)$ is an additive group with identity elt. $0_{\mathbb{K}}$
- $(\mathbb{K} \setminus \{0_{\mathbb{K}}\}, \otimes)$ is a multiplicative group
- $\otimes$ is distributive over $\oplus$: $\forall a, b, c \in \mathbb{K}, \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Examples

- $(\mathbb{R}, +, \times)$
- $(\mathbb{Z}/n\mathbb{Z}, +, \times)$? Not in the general case !
- $(\mathbb{F}_p, +, \times)$

Elliptic Curve Equation

General equation

An elliptic curve over a field $\mathbb{K}$ has an equation:

$$\mathcal{E}/\mathbb{K} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Elliptic Curve Equation

General equation

An elliptic curve over a field $\mathbb{K}$ has an equation:

$$\mathcal{E}/\mathbb{K} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Characteristic 2 (ordinary curves)

If $\mathbb{K} = \mathbb{F}_{2^n}$, $a_1 = 1$, $a_3 = a_4 = 0$:

$$\mathcal{E}/\mathbb{K} : y^2 + xy = x^3 + a_2x^2 + a_6$$

Elliptic Curve Equation

General equation

An elliptic curve over a field $\mathbb{K}$ has an equation:

$$\mathcal{E}/\mathbb{K} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Characteristic 2 (ordinary curves)

If $\mathbb{K} = \mathbb{F}_{2^n}$, $a_1 = 1$, $a_3 = a_4 = 0$:

$$\mathcal{E}/\mathbb{K} : y^2 + xy = x^3 + a_2x^2 + a_6$$

Large characteristic p

If $\mathbb{K} = \mathbb{F}_p$ with $p > 3$, $a_1 = a_2 = a_3 = 0$:

$$y^2 = x^3 + a_4x + a_6$$

Elliptic Curve Equation

General equation

An elliptic curve over a field $\mathbb{K}$ has an equation:

$$\mathcal{E}/\mathbb{K} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Characteristic 2 (ordinary curves)

If $\mathbb{K} = \mathbb{F}_{2^n}$, $a_1 = 1$, $a_3 = a_4 = 0$:

$$\mathcal{E}/\mathbb{K} : y^2 + xy = x^3 + a_2x^2 + a_6$$

Large characteristic p

If $\mathbb{K} = \mathbb{F}_p$ with $p > 3$, $a_1 = a_2 = a_3 = 0$:

$$y^2 = x^3 + ax + b$$

Elliptic Curve Equation

General equation

An

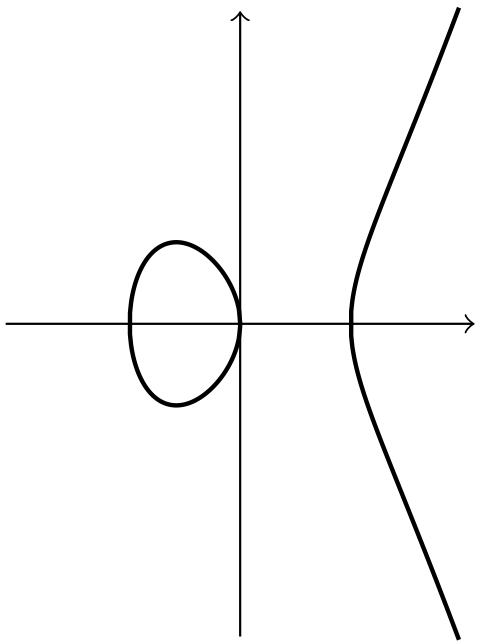
Ch

If $\mathbb{K}$

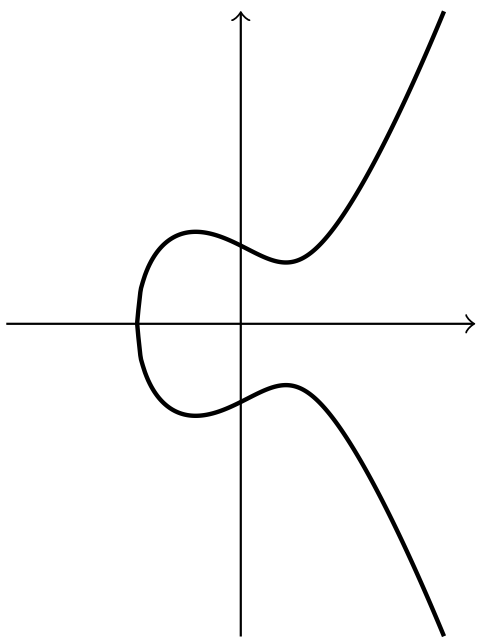
La

If $\mathbb{K}$

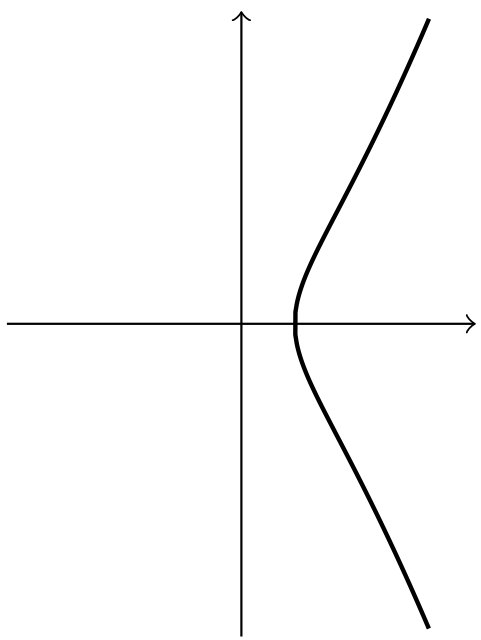
$$\mathbb{K} = \mathbb{R}$$



$y^2 = x^3 - 2x$



$y^2 = x^3 - x + 1$



$y^2 = x^3 + x - 1$

$$y^2 = x^3 + ax + b$$

Elliptic Curve Group Law

Group of points

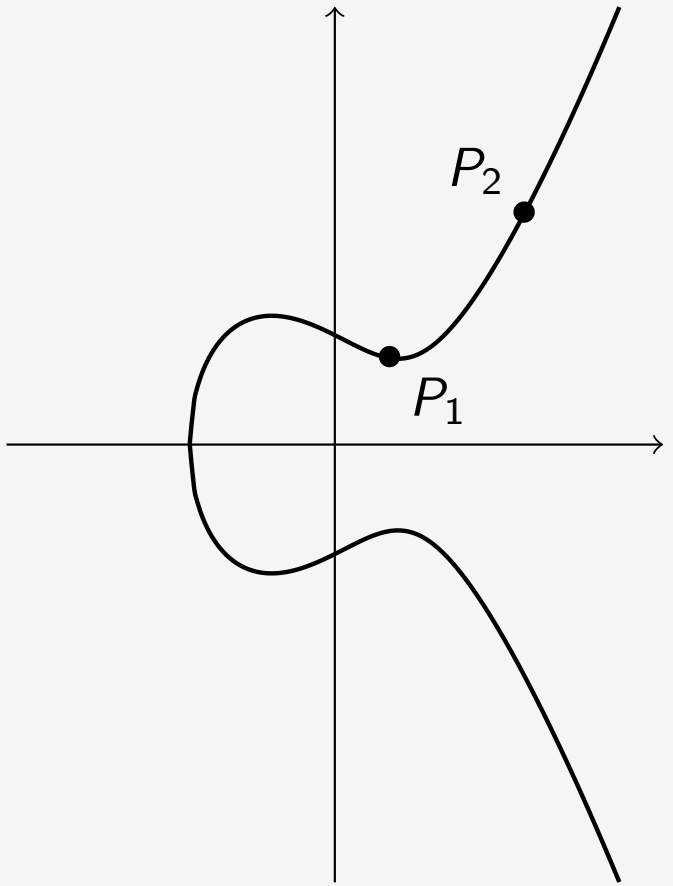
Let $\mathcal{E}(\mathbb{K})$ be the set of points $(x, y) \in \mathbb{K}^2$ satisfying the equation of $\mathcal{E}$ together with the *point at infinity* $\mathcal{O}$.

- $\mathcal{E}(\mathbb{K})$ is a group with neutral element $\mathcal{O}$
- Group law is denoted $+$

Elliptic Curve Group Law

$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.



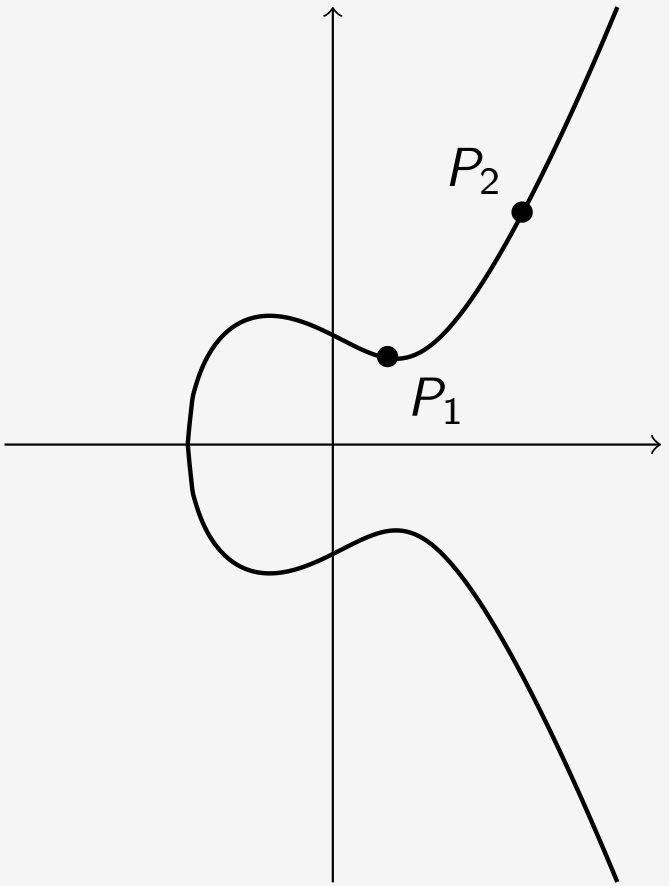
Elliptic Curve Group Law

$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$



Elliptic Curve Group Law

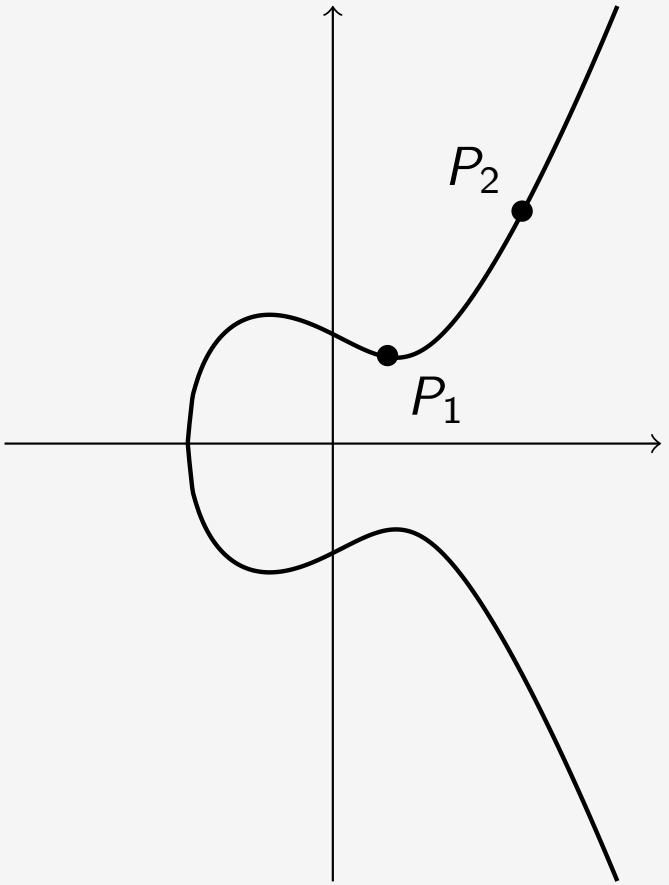
$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{if} \quad P_1 \neq \pm P_2$$



Elliptic Curve Group Law

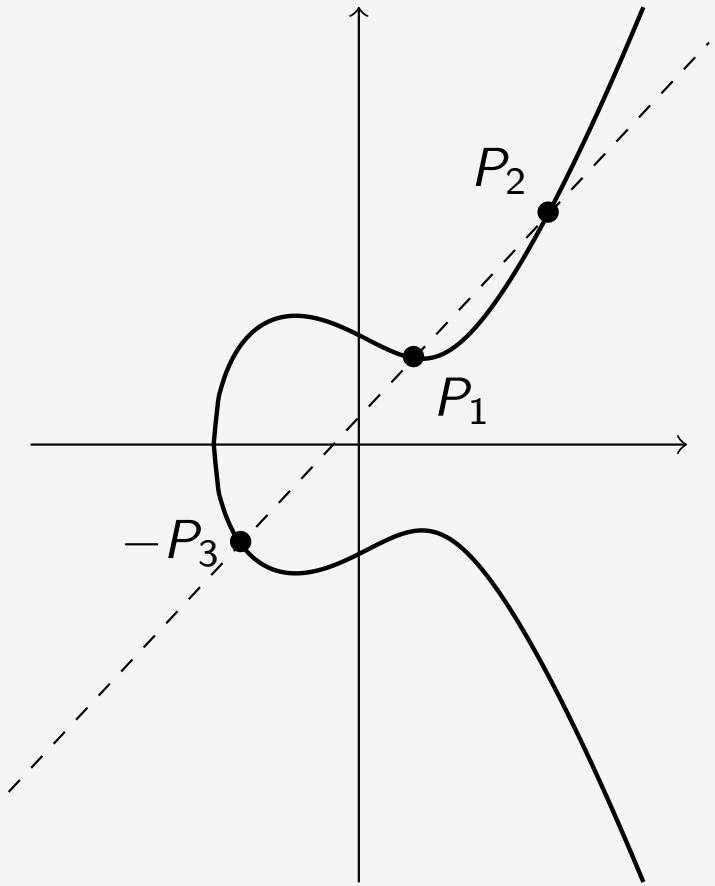
$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{if} \quad P_1 \neq \pm P_2$$



Elliptic Curve Group Law

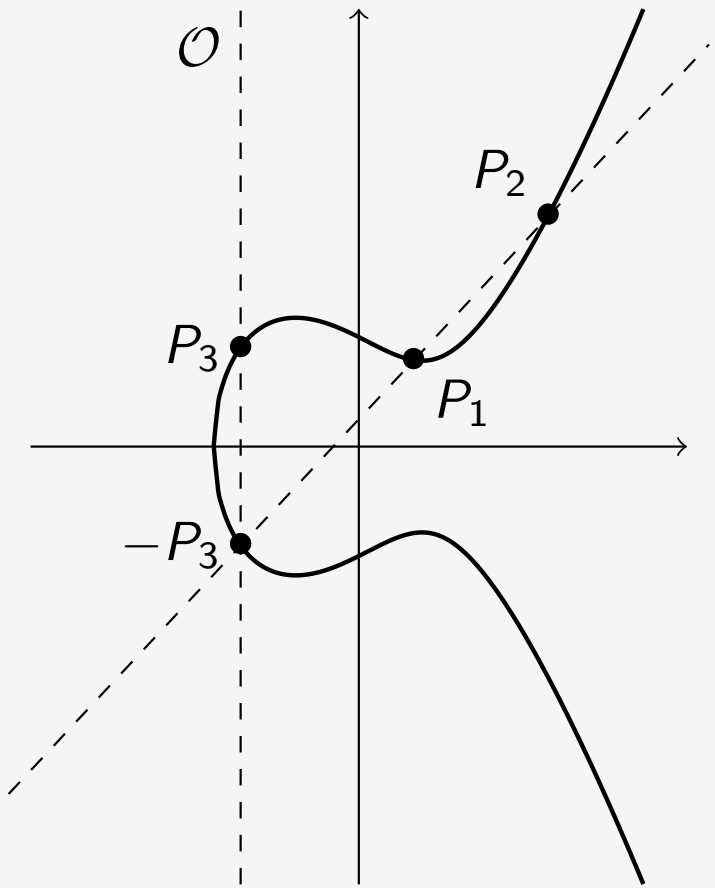
$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{if} \quad P_1 \neq \pm P_2$$



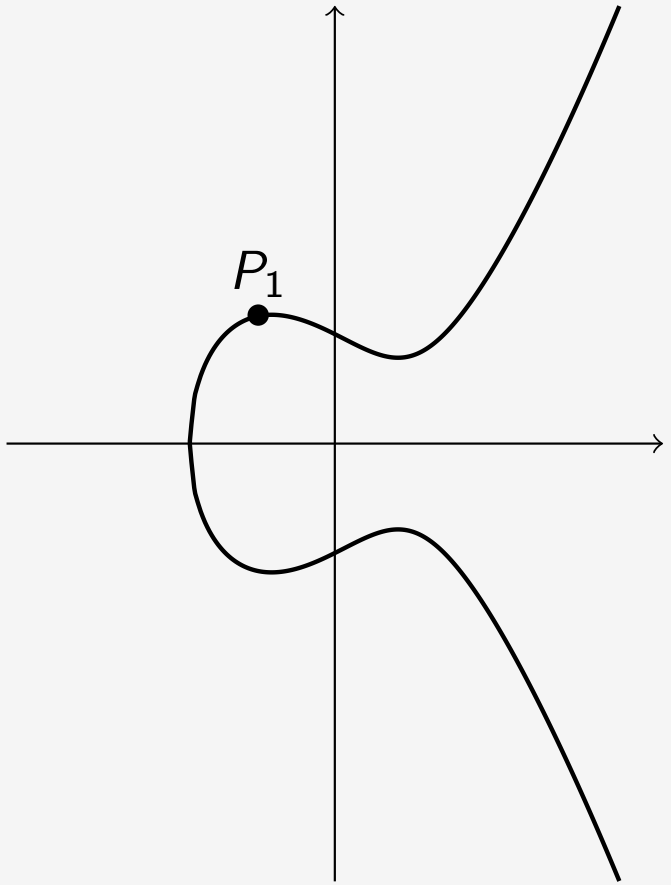
Elliptic Curve Group Law

$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$



Elliptic Curve Group Law

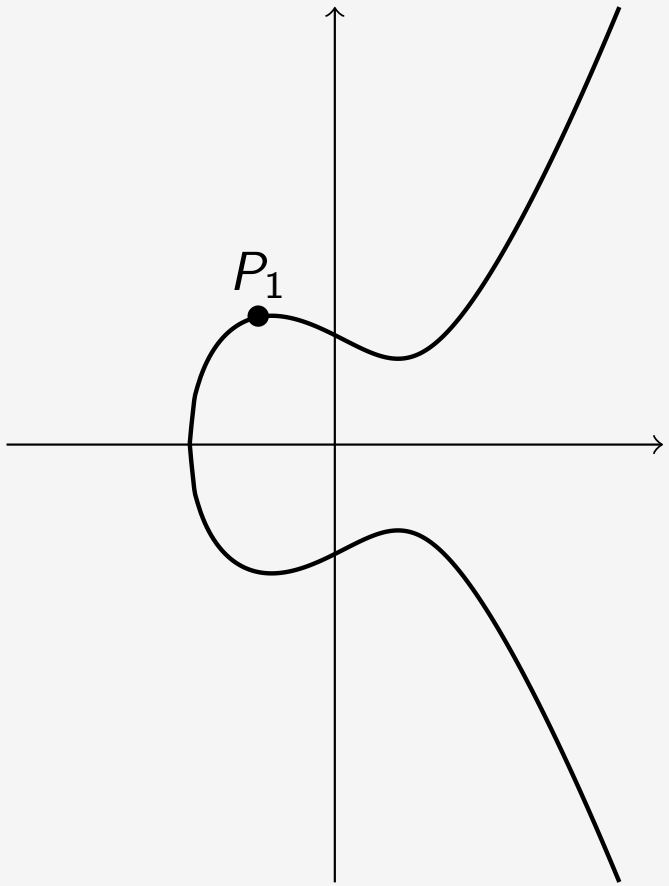
$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$

$$m = \frac{3{x_1}^2 + a}{2y_1} \quad \text{if} \quad P_1 = P_2$$



Elliptic Curve Group Law

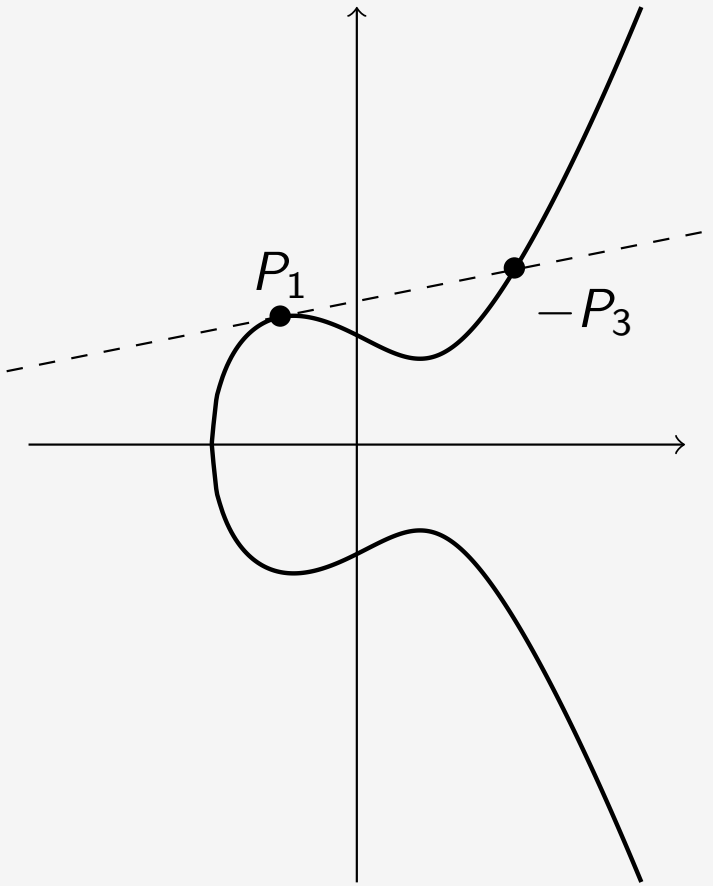
$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$

$$m = \frac{3{x_1}^2 + a}{2y_1} \quad \text{if} \quad P_1 = P_2$$



Elliptic Curve Group Law

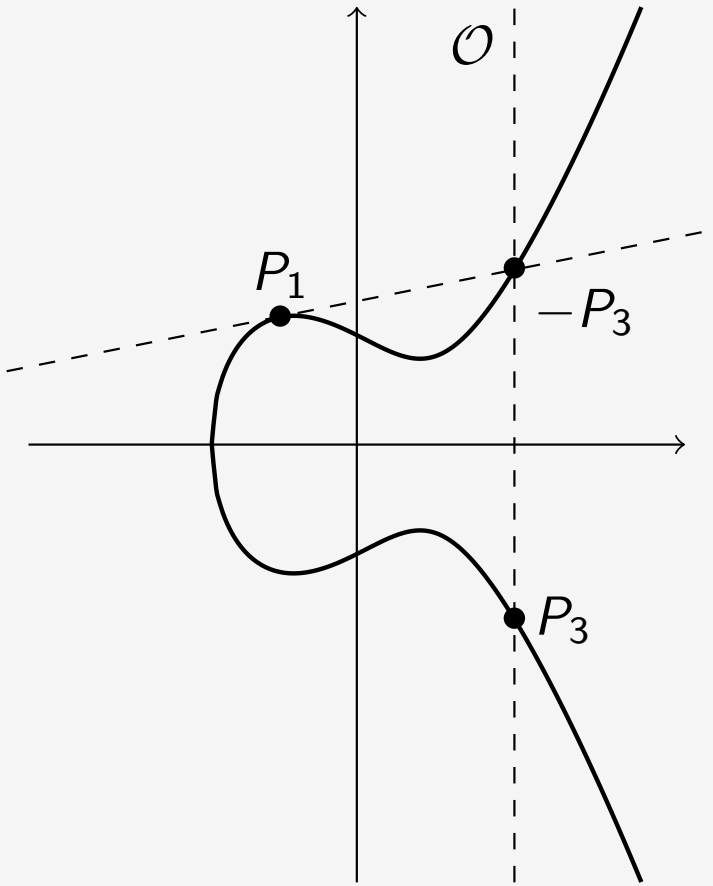
$$\mathbb{K} = \mathbb{F}_p$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, $P_1, P_2 \neq \mathcal{O}$.

$P_3 = P_1 + P_2$ is given by:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m\,(x_1 - x_3) - y_1 \end{cases}$$

$$m = \frac{3{x_1}^2 + a}{2y_1} \quad \text{if} \quad P_1 = P_2$$



Scalar Multiplication

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

Scalar Multiplication

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

Elliptic Curve Discrete Logarithm Problem (ECDLP)

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

Scalar Multiplication

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

Elliptic Curve Discrete Logarithm Problem (ECDLP)

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

Seems to be harder than DLP on finite fields or factorization.

Cryptosystems comparison

Diffie-H. Elgamal on GF

- multiplicative group $\mathbb{F}_p^* = \{1, 2, \ldots, p-1\}$
- DLP on $\mathbb{F}_p^*$
- keylength: 1024 to 16384 bits (eq. 2^{80} to 2^{256})

RSA

- ring $\mathbb{Z}/N\mathbb{Z} = \{1, 2, \ldots, N-1\}$, $N = p \cdot q$
- integer factorisation
- keylength: 1024 to 16384 bits

Cryptosystems comparison

Diffie-H. Elgamal on GF

- multiplicative group $\mathbb{F}_p^* = \{1, 2, \ldots, p-1\}$
- DLP on $\mathbb{F}_p^*$
- keylength: 1024 to 16384 bits (eq. 2^{80} to 2^{256})

RSA

- ring $\mathbb{Z}/N\mathbb{Z} = \{1, 2, \ldots, N-1\}$, $N = p \cdot q$
- integer factorisation
- keylength: 1024 to 16384 bits

ECC

- group of points $\mathcal{E}(\mathbb{K})$
- DLP on $\mathcal{E}(\mathbb{K})$
- keylength: 160 to 512 bits

Outline

Outline

Exponentiation algorithms

Considering integers m, n, and $d = (d_{\ell-1} d_{\ell-2} \dots d_0)_2$,
how to compute $m^d \mod n$?

Exponentiation algorithms

Considering integers m, n, and $d = (d_{\ell-1} d_{\ell-2} \dots d_0)_2$,
how to compute $m^d \mod n$?

Binary decomposition

- $m^d = m^{d_0} \times \left(m^{d_1} \times \left(\dots \left(m^{d_{\ell-1}} \right)^2 \dots \right)^2 \right)^2$
- $m^d = m^{d_{\ell-1} 2^{\ell-1}} \times m^{d_{\ell-2} 2^{\ell-2}} \times \dots \times m^{d_0}$

Exponentiation algorithms

Considering integers m, n, and $d = (d_{\ell-1} d_{\ell-2} \dots d_0)_2$,
how to compute $m^d \mod n$?

Binary decomposition

- $m^d = m^{d_0} \times \left(m^{d_1} \times \left(\dots \left(m^{d_{\ell-1}} \right)^2 \dots \right)^2 \right)^2$
- $m^d = m^{d_{\ell-1} 2^{\ell-1}} \times m^{d_{\ell-2} 2^{\ell-2}} \times \dots \times m^{d_0}$

Example with $d = 13 = (1101)_2$

$$m^{13} = m^1 \times \left(m^0 \times \left(m^1 \times \left(m^1 \right)^2 \right)^2 \right)^2$$

Exponentiation algorithms

Considering integers m, n, and $d = (d_{\ell-1} d_{\ell-2} \dots d_0)_2$,
how to compute $m^d \mod n$?

Binary decomposition

- $m^d = m^{d_0} \times \left(m^{d_1} \times \left(\dots \left(m^{d_{\ell-1}} \right)^2 \dots \right)^2 \right)^2$
- $m^d = m^{d_{\ell-1} 2^{\ell-1}} \times m^{d_{\ell-2} 2^{\ell-2}} \times \dots \times m^{d_0}$

Example with $d = 13 = (1101)_2$

$$\begin{aligned} m^{13} &= m^1 \times \left(m^0 \times \left(m^1 \times \left(m^1 \right)^2 \right)^2 \right)^2 \\ &= m^{1\times 8} \times m^{1\times 4} \times m^{0\times 2} \times m^{1\times 1} \end{aligned}$$

Left-to-right

$$m^d = m^{d_0} \times \left(m^{d_1} \times \left(\ldots \left(m^{d_{\ell-1}} \right)^2 \ldots \right)^2 \right)^2$$

Input: $m, n, d \in \mathbb{N}$
Output: $m^d \mod n$

```
a ← 1
for i = ℓ − 1 to 0 do
    a ← a^2 mod n
    if d_i = 1 then
        a ← a × m mod n
return a
```

Exponentiation algorithms

Square and multiply

Left-to-right

$$m^d = m^{d_0} \times \left(m^{d_1} \times \left(\ldots \left(m^{d_{\ell-1}} \right)^2 \ldots \right)^2 \right)^2$$

Input: $m, n, d \in \mathbb{N}$
Output: $m^d \mod n$

$a \leftarrow 1$
for $i = \ell - 1$ **to** 0 **do**
  $a \leftarrow a^2 \mod n$
  if $d_i = 1$ **then**
    $a \leftarrow a \times m \mod n$
return a

Right-to-left

$$m^d = m^{d_{\ell-1} 2^{\ell-1}} \times m^{d_{\ell-2} 2^{\ell-2}} \times \ldots \times m^{d_0}$$

Input: $m, n, d \in \mathbb{N}$
Output: $m^d \mod n$

$a \leftarrow 1$; $b \leftarrow m$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $a \leftarrow a \times b \mod n$
  $b \leftarrow b^2 \mod n$
return a

What about scalar multiplication ?

Considering a point $P \in \mathcal{E}(\mathbb{K})$ and an integer $d = (d_{\ell-1} d_{\ell-2} \dots d_0)_2$, how to compute $d \cdot P$?

What about scalar multiplication ?

Considering a point $P \in \mathcal{E}(\mathbb{K})$ and an integer $d = (d_{\ell-1} d_{\ell-2} \dots d_0)_2$, how to compute $d \cdot P$?

Binary decomposition

- $d \cdot P = d_0 P + 2\left(d_1 P + 2\left(\dots + 2\left(d_{\ell-1} P\right)\dots\right)\right)$
- $d \cdot P = d_{\ell-1} 2^{\ell-1} P + d_{\ell-2} 2^{\ell-2} P + \dots + d_0 P$

Left-to-right

$$d \cdot P = d_0 P + 2\left(d_1 P + 2\left(\ldots + 2\left(d_{\ell-1} P\right)\ldots\right)\right)$$

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

```
R ← O
for i = ℓ − 1 to 0 do
    R ← 2R
    if d_i = 1 then
        R ← R + P
return R
```

Scalar multiplication algorithms

Double and add

Left-to-right

$$d \cdot P = d_0 P + 2\left(d_1 P + 2\left(\ldots + 2\left(d_{\ell-1} P\right)\ldots\right)\right)$$

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
return R

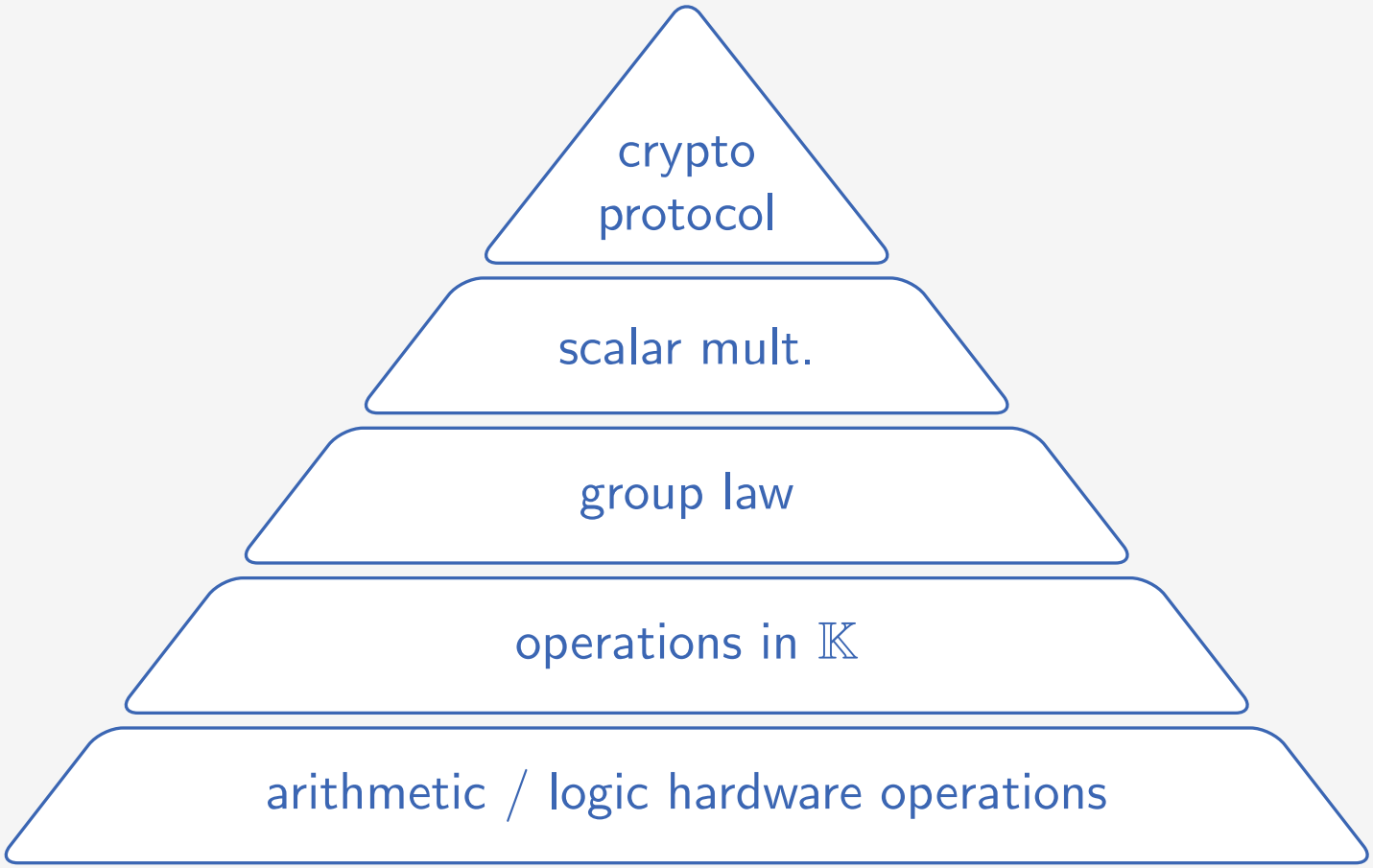
Right-to-left

$$d \cdot P = d_{\ell-1} 2^{\ell-1} P + d_{\ell-2} 2^{\ell-2} P + \ldots + d_0 P$$

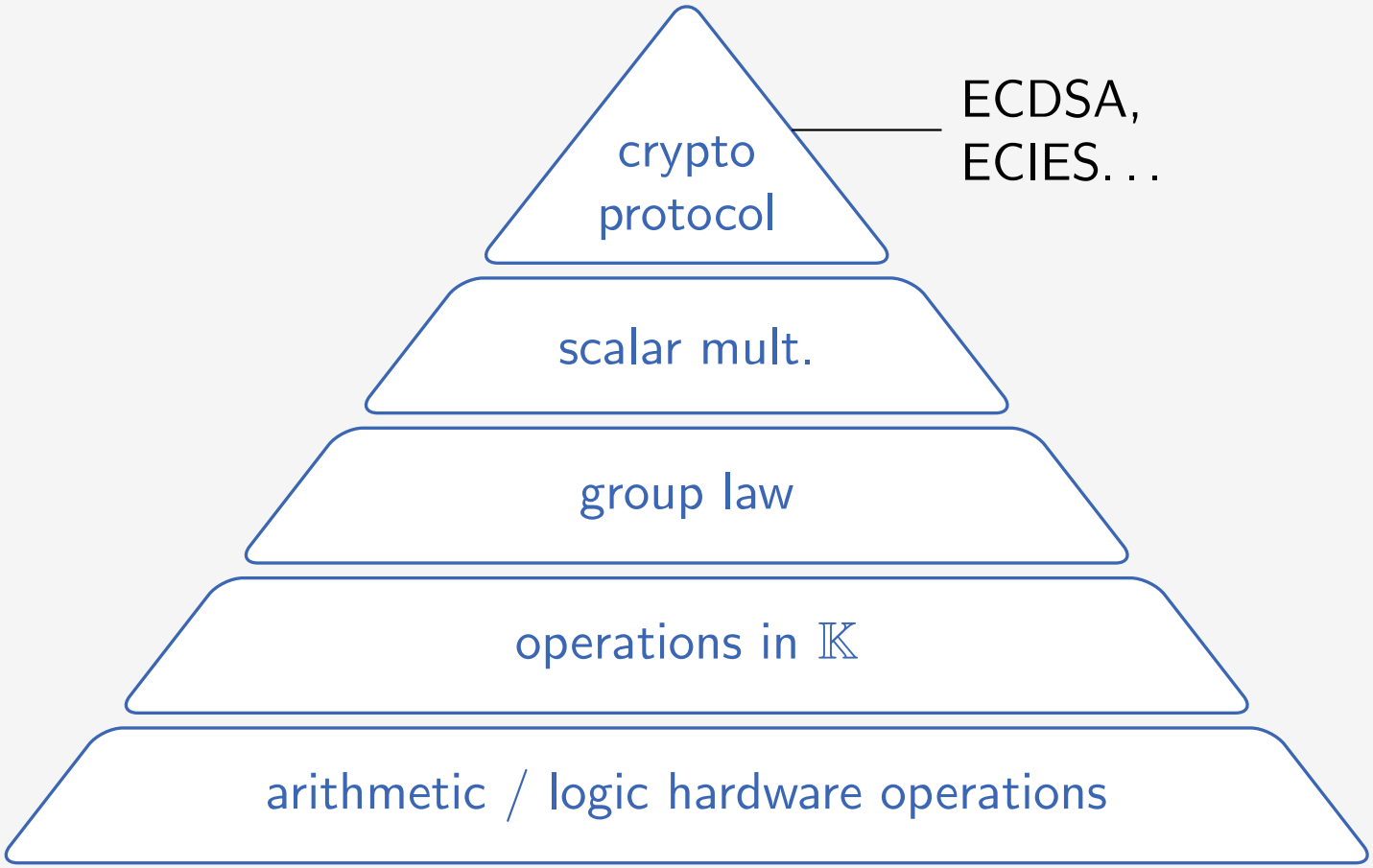
Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$; $Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  $Q \leftarrow 2Q$
return R

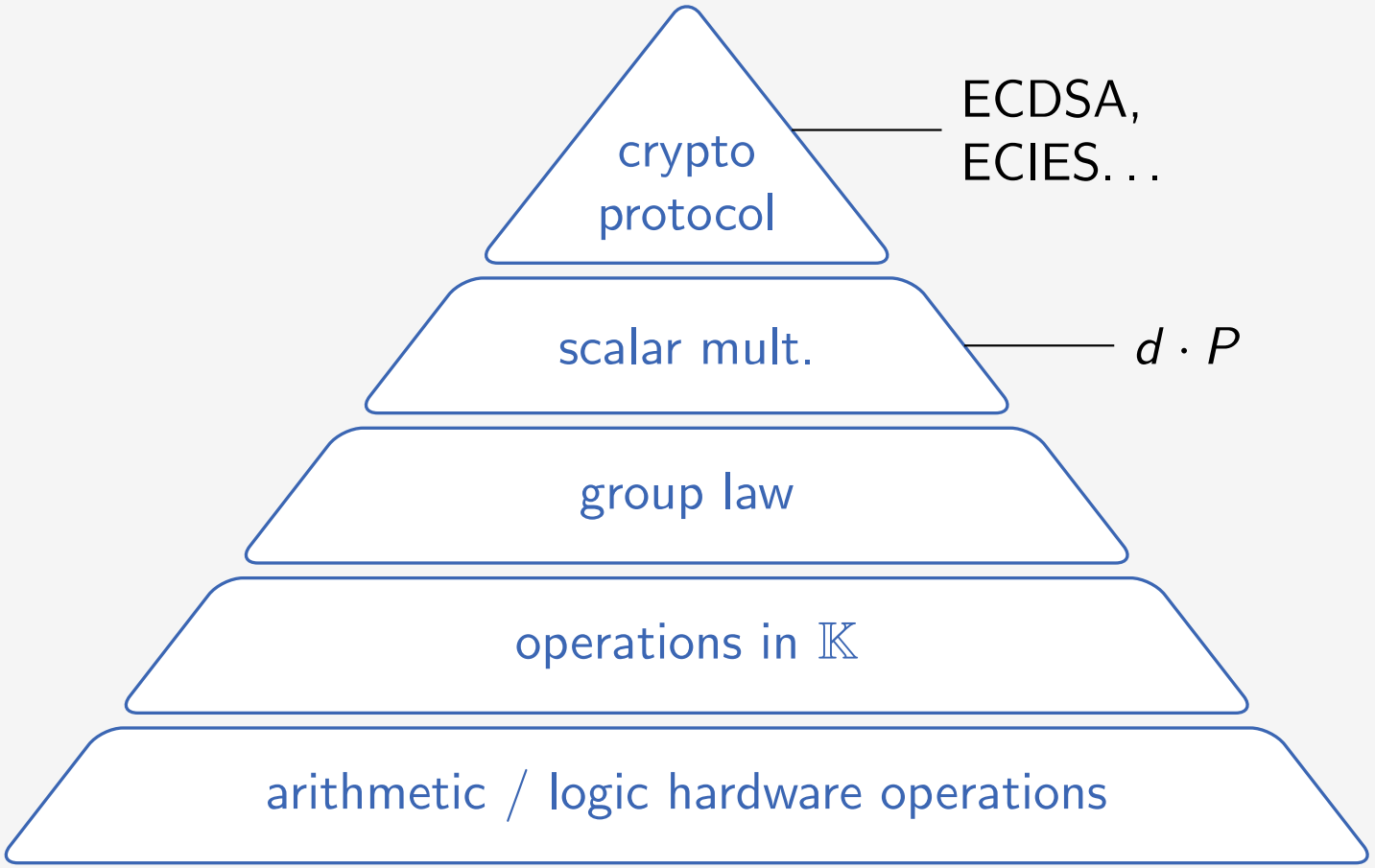
ECC Implementation Layers



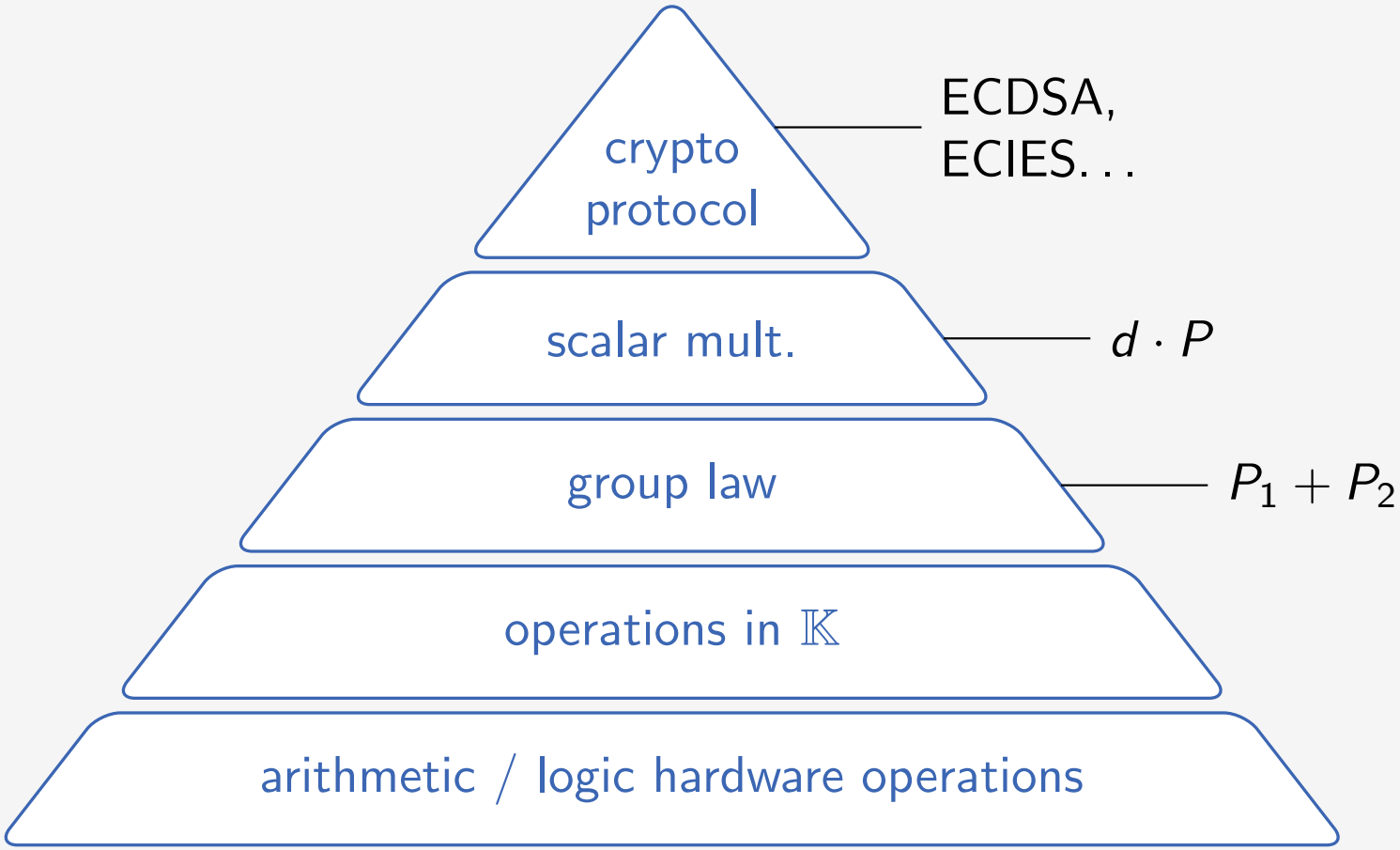
ECC Implementation Layers



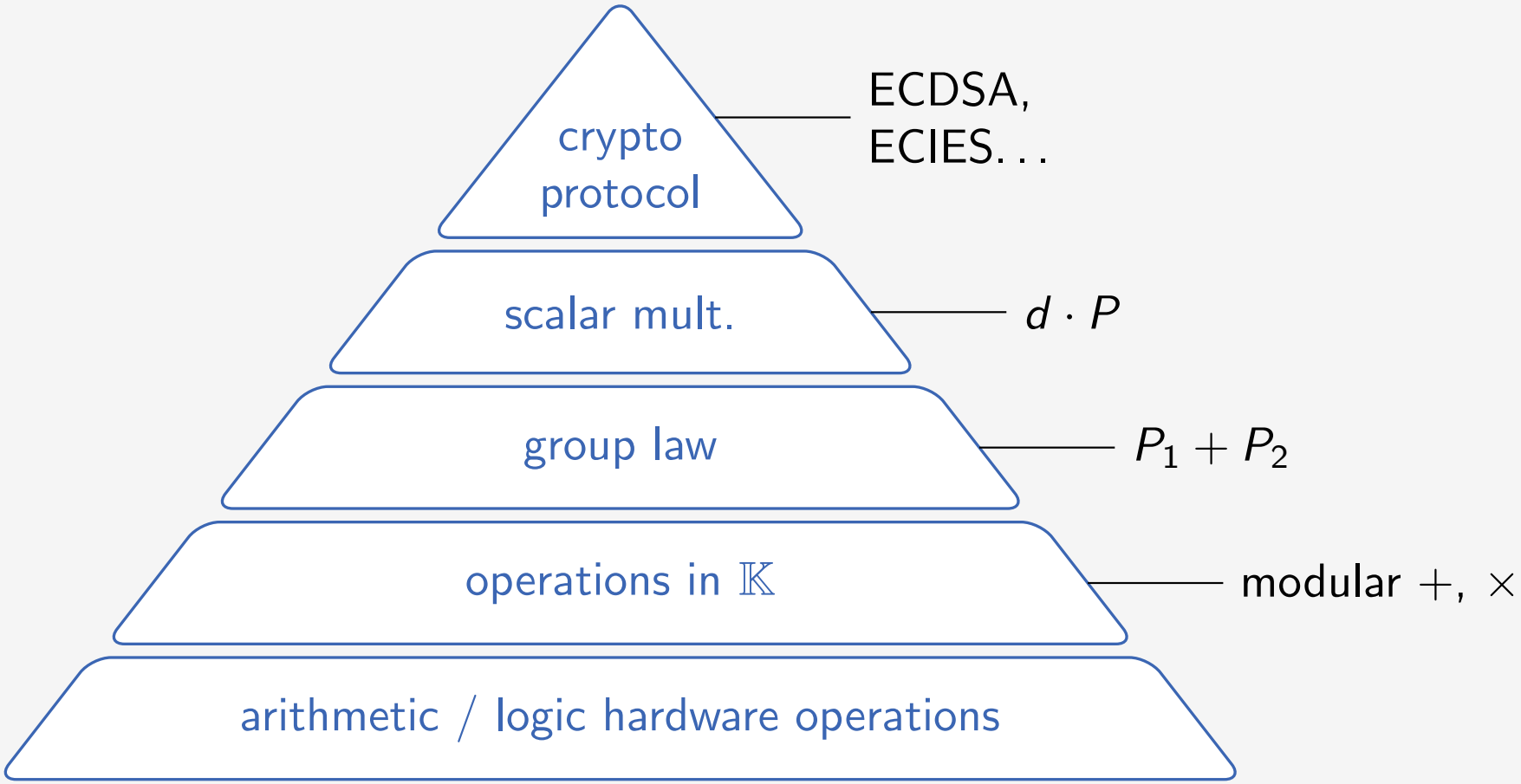
ECC Implementation Layers



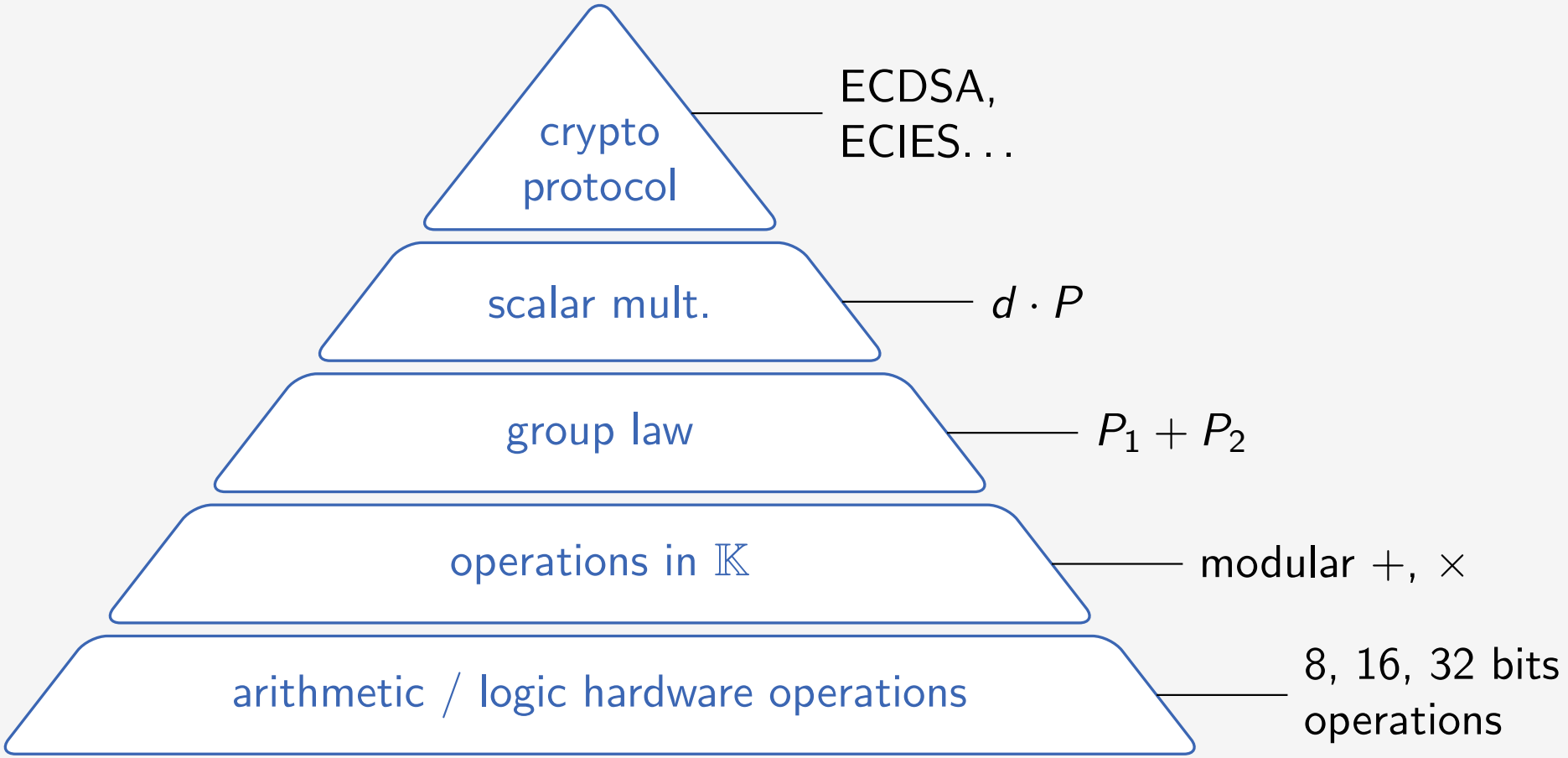
ECC Implementation Layers



ECC Implementation Layers



ECC Implementation Layers



Outline

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

- 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

Expensive operations

- Inversion, GCD (I)

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion, GCD (I)

Significant operations

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

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion, GCD (I)

Significant operations

- Multiplication (M)
- Squaring (S, S/M ≈ 0.8)

Negligible operations

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

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion, GCD (I)

Significant operations

- Multiplication (M)
- Squaring (S, S/M ≈ 0.8)

Negligible operations

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

$\mathbb{F}_p$ Operations Theoretical Cost

Expensive operations

- Inversion, GCD (I)

Significant operations

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

Negligible operations

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

A/M $\approx$ 0.2 on some smart cards

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 and Add Algorithm

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  $Q \leftarrow 2Q$
return R

On average :

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

Double and Add Algorithm

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  $Q \leftarrow 2Q$
return R

On average :

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

Proj. hom. : 17,6M/bit
Proj. jac. : 15,9M/bit
Proj. jac. mod. : 16,3M/bit

Double and Add Algorithm

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  $Q \leftarrow 2Q$
return R

On average :

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

Proj. hom. : 17,6M/bit
Proj. jac. : 15,9M/bit
Proj. jac. mod. : 16,3M/bit

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

Example :

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

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

Example :

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

Interest

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

NAF Multiplication

Right-to-Left

Input:
$P \in \mathcal{E}(\mathbb{K}), d = (d_{\ell-1} \dots d_1 d_0)_{\text{NAF}}$
Output: $d \cdot P$
$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  else if $d_i = -1$ **then**
    $R \leftarrow R + (-Q)$
  $Q \leftarrow 2Q$
return R

NAF Multiplication

Right-to-Left

Input:
$P \in \mathcal{E}(\mathbb{K}), d = (d_{\ell-1} \dots d_1 d_0)_{\text{NAF}}$
Output: $d \cdot P$
$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  else if $d_i = -1$ **then**
    $R \leftarrow R + (-Q)$
  $Q \leftarrow 2Q$
return R

Cost :

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

Proj. jac. : 13,4M/bit

NAF Multiplication

Right-to-Left

Input:
$P \in \mathcal{E}(\mathbb{K}), d = (d_{\ell-1} \dots d_1 d_0)_{\text{NAF}}$
Output: $d \cdot P$
$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  else if $d_i = -1$ **then**
    $R \leftarrow R + (-Q)$
  $Q \leftarrow 2Q$
return R

Cost :

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

Variant from [Joye, WAIFI 2008]:

Proj. jac. : 13,4M/bit

NAF Multiplication

Right-to-Left

Input:
$P \in \mathcal{E}(\mathbb{K}), d = (d_{\ell-1} \dots d_1 d_0)_{\text{NAF}}$
Output: $d \cdot P$
$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  else if $d_i = -1$ **then**
    $R \leftarrow R + (-Q)$
  $Q \leftarrow 2Q$
return R

Cost :

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

Variant from [Joye, WAIFI 2008]:

- R in Jacobian coordinates

Proj. jac. : 13,4M/bit

NAF Multiplication

Right-to-Left

Input:
$P \in \mathcal{E}(\mathbb{K}), d = (d_{\ell-1} \dots d_1 d_0)_{\text{NAF}}$
Output: $d \cdot P$
$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  else if $d_i = -1$ **then**
    $R \leftarrow R + (-Q)$
  $Q \leftarrow 2Q$
return R

Cost :

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

Variant from [Joye, WAIFI 2008]:

- R in Jacobian coordinates
- Q in modified Jacobian coordinates

Proj. jac. : 13,4M/bit

NAF Multiplication

Right-to-Left

Input:
$P \in \mathcal{E}(\mathbb{K}), d = (d_{\ell-1} \dots d_1 d_0)_{\text{NAF}}$
Output: $d \cdot P$
$R \leftarrow \mathcal{O}$
$Q \leftarrow P$
for $i = 0$ **to** $\ell - 1$ **do**
  if $d_i = 1$ **then**
    $R \leftarrow R + Q$
  else if $d_i = -1$ **then**
    $R \leftarrow R + (-Q)$
  $Q \leftarrow 2Q$
return R

Cost :

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

Variant from [Joye, WAIFI 2008]:

- R in Jacobian coordinates
- Q in modified Jacobian coordinates

Proj. jac. : 13,4M/bit
Mixed coord. : 12,0M/bit

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

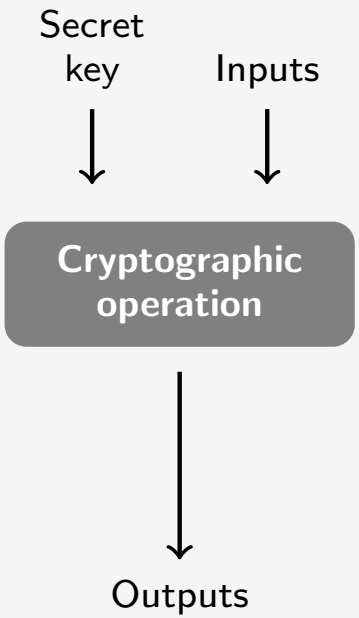
Euclidean Addition Chains [Meloni, WAIFI 2007]
Co-Z binary ladder [Goundar, Joye & Miyaji, CHES 2010]

Outline

Outline

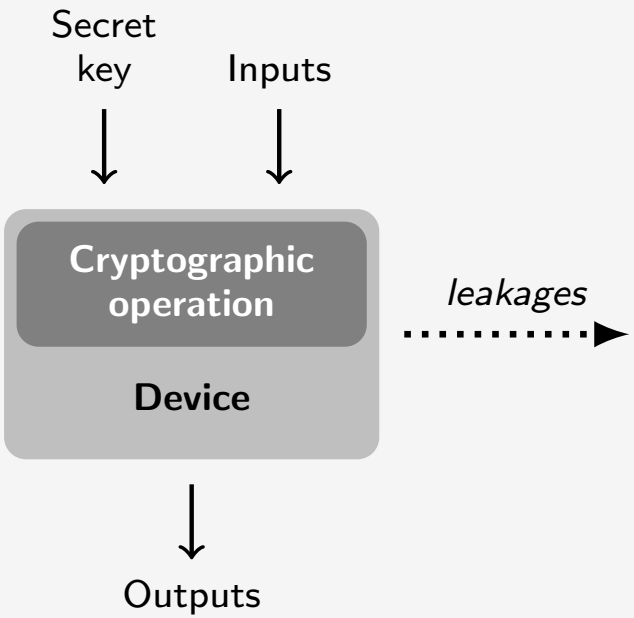
Physical Analysis Framework

Passive Attacks



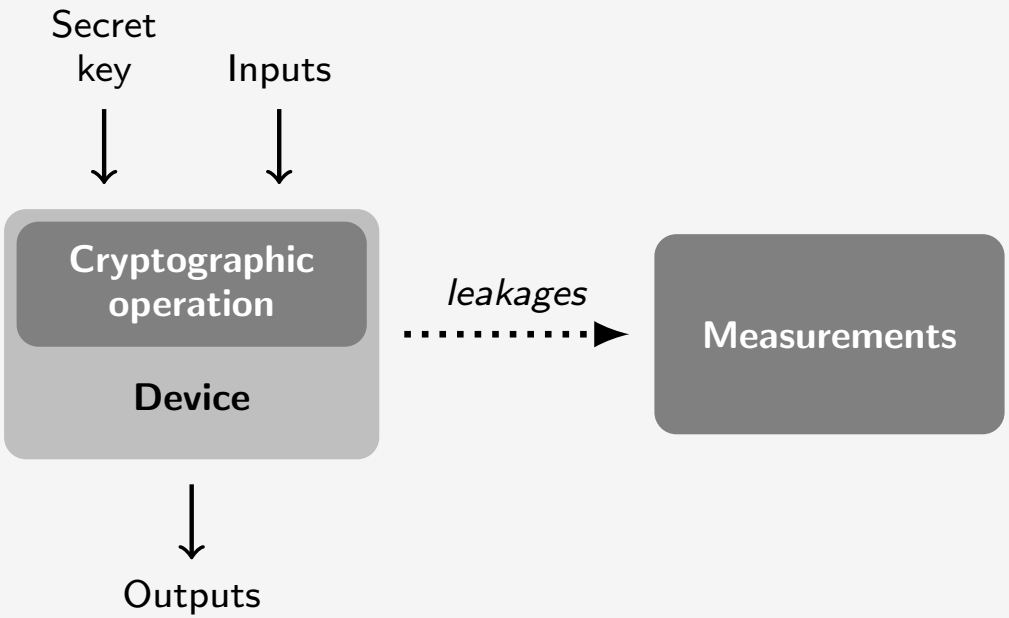
Physical Analysis Framework

Passive Attacks



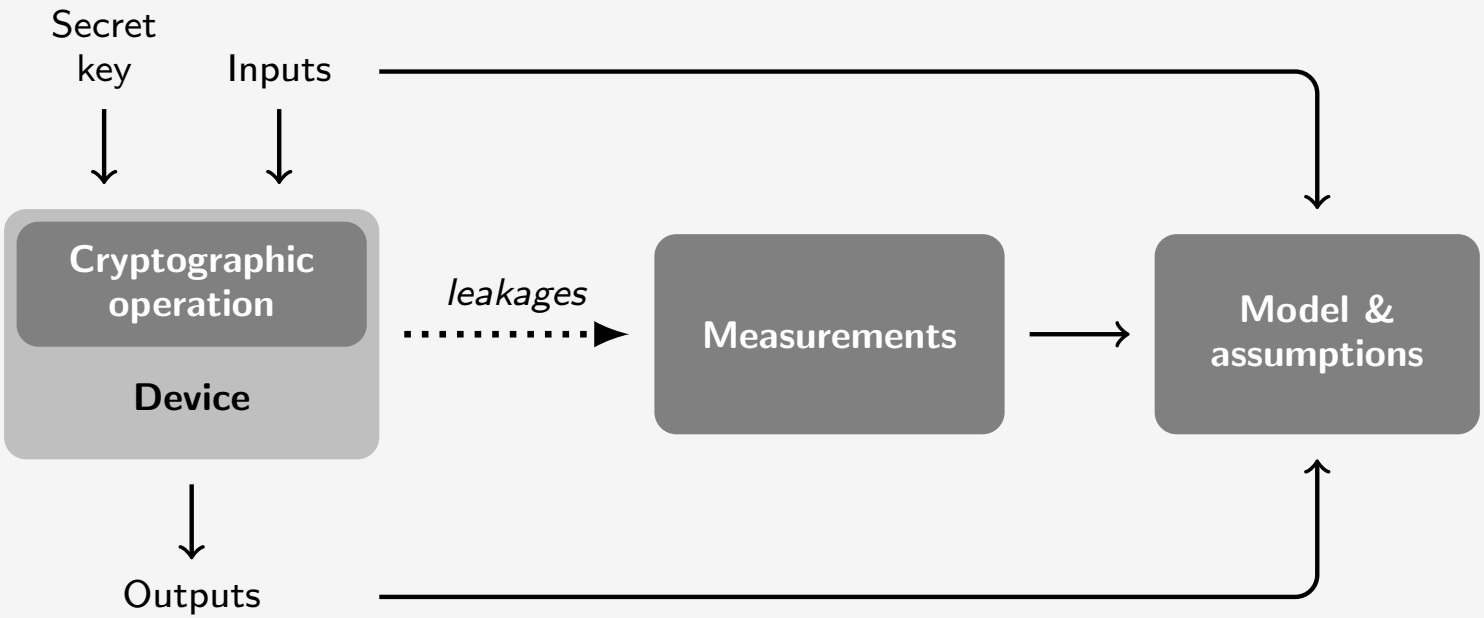
Physical Analysis Framework

Passive Attacks



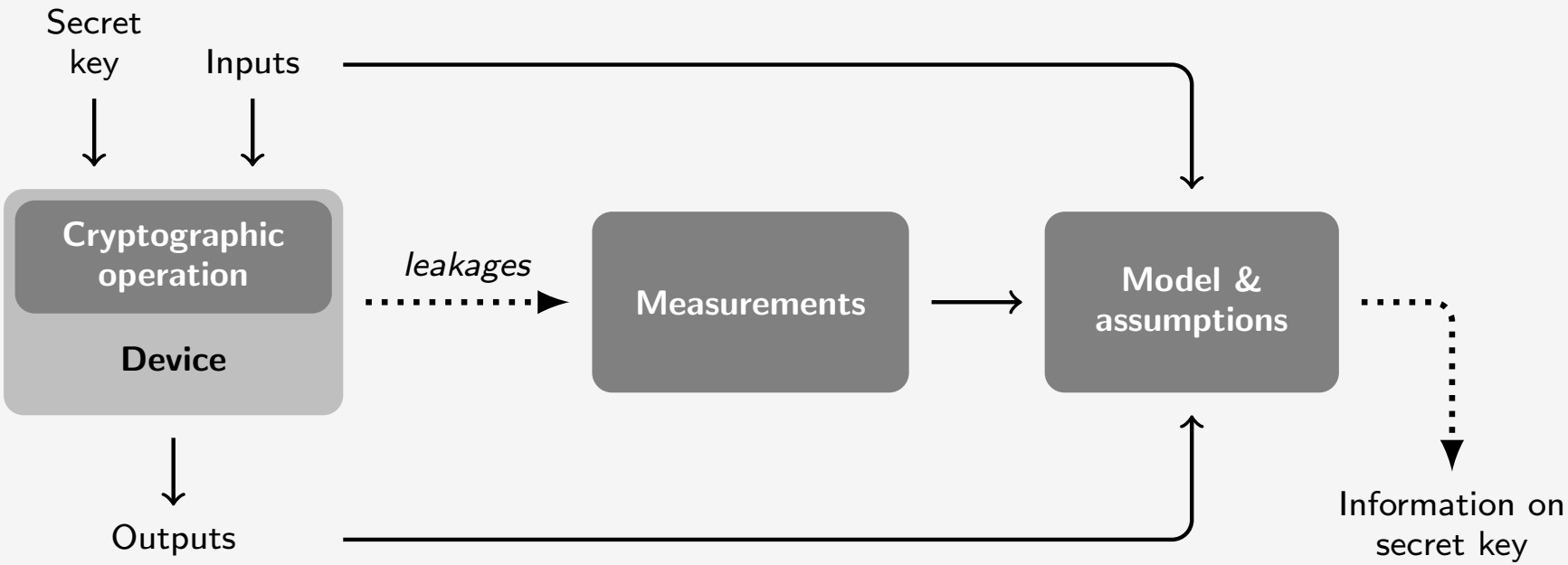
Physical Analysis Framework

Passive Attacks



Physical Analysis Framework

Passive Attacks

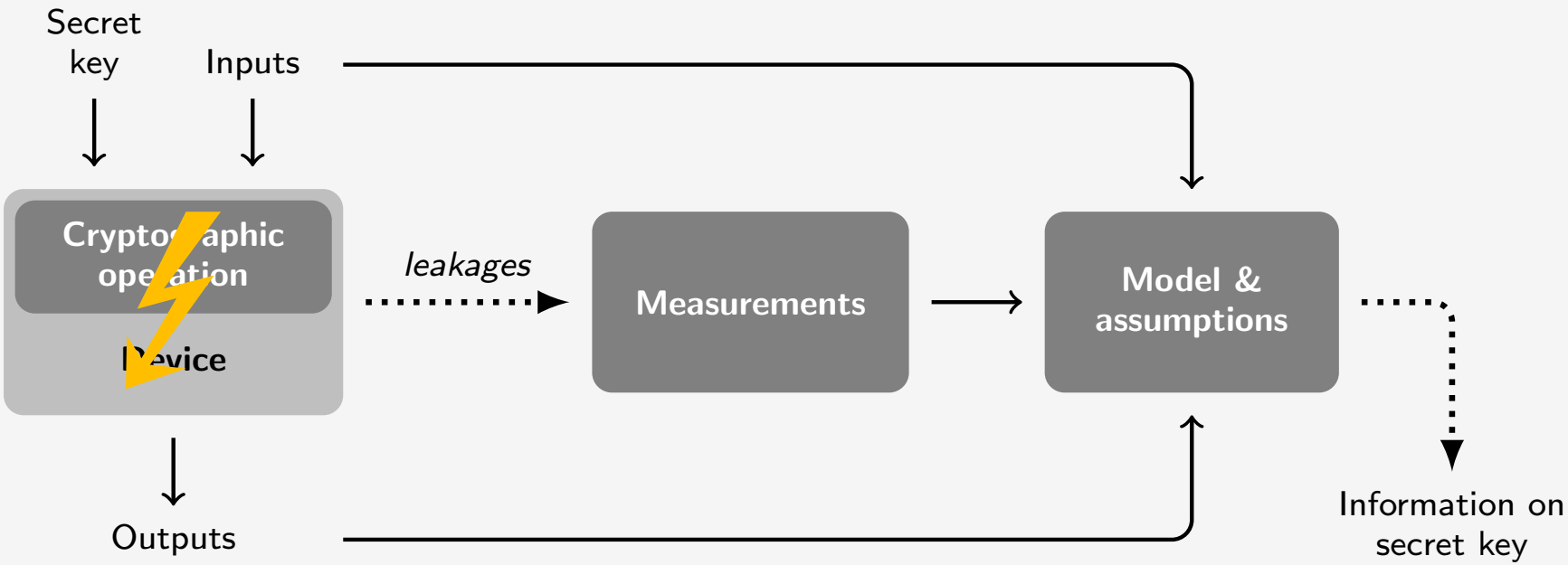


Physical Analysis Framework

Active Attacks

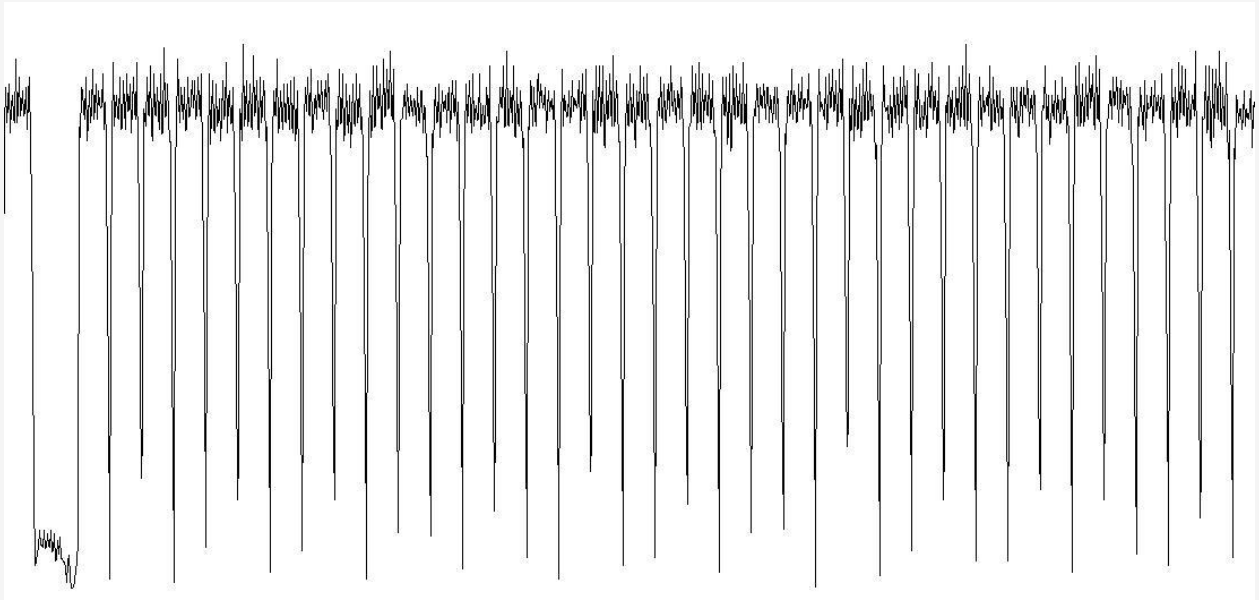
Physical Analysis Framework

Combined Attacks



Simple Analyse Example

Leakage on Performed Operations



RSA exponentiation

Simple Analyse Example

Leakage on Manipulated Data

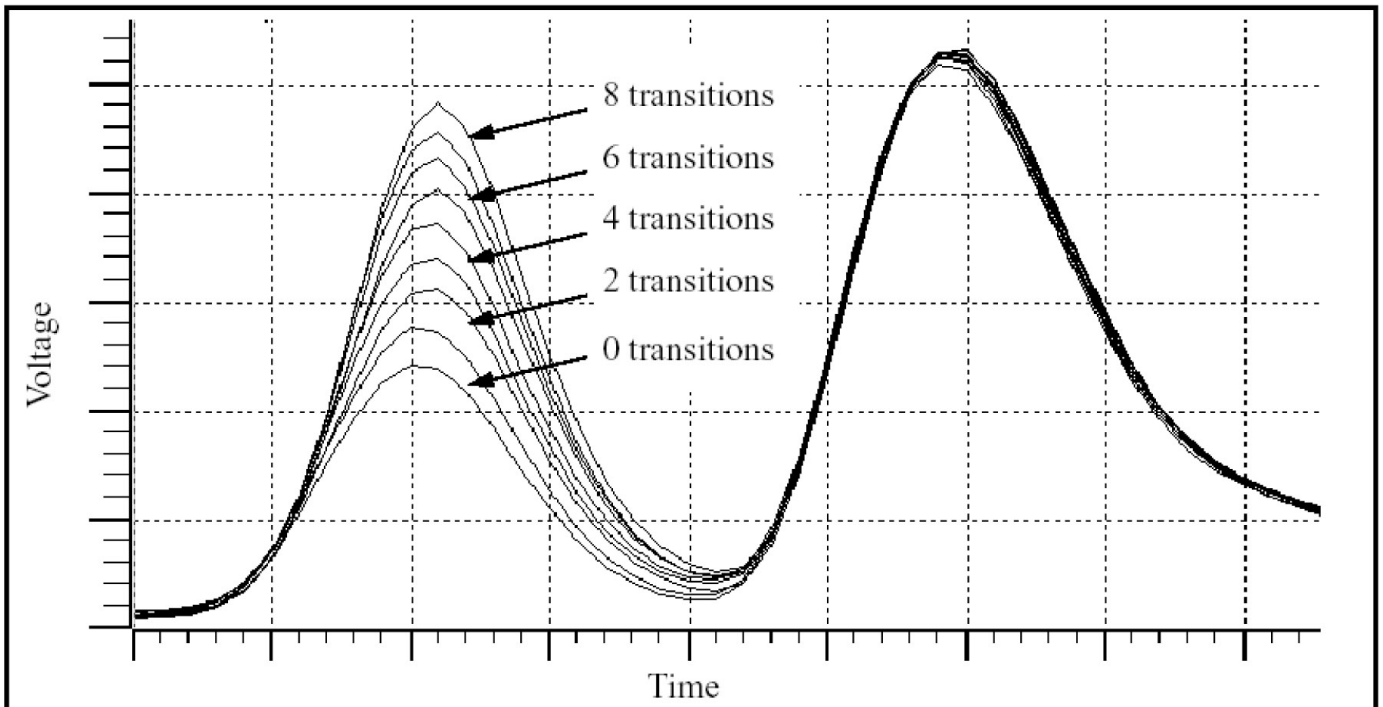


FIGURE 2. **Number of Bit Transitions versus Power Consumption**

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]
- DPA on RSA [Messerges et al., *Power Analysis Attacks of Modular Exponentiation in Smartcards*, CHES 99]
- DFA on ECC [Biehl et al., *Differential Fault Attacks on Elliptic Curve Cryptosystems*, Crypto 2000]

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]
- DPA on RSA [Messerges et al., *Power Analysis Attacks of Modular Exponentiation in Smartcards*, CHES 99]
- DFA on ECC [Biehl et al., *Differential Fault Attacks on Elliptic Curve Cryptosystems*, Crypto 2000]
- 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]

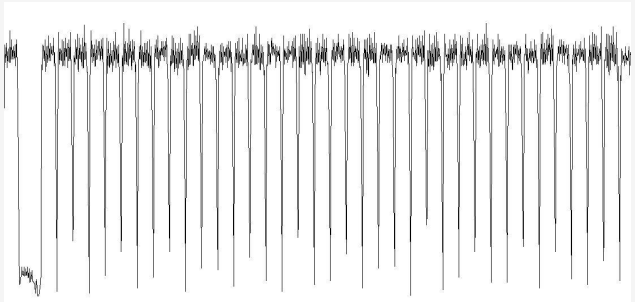
Outline

Simple Analysis Principle

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

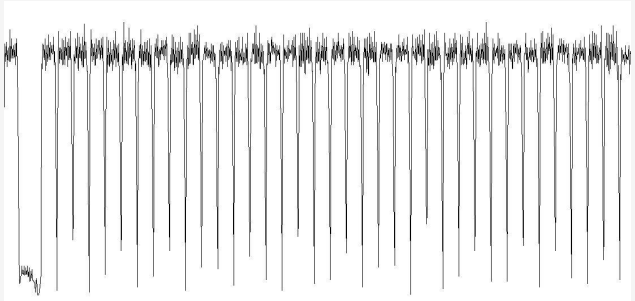
Simple Analysis Principle

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



Simple Analysis Principle

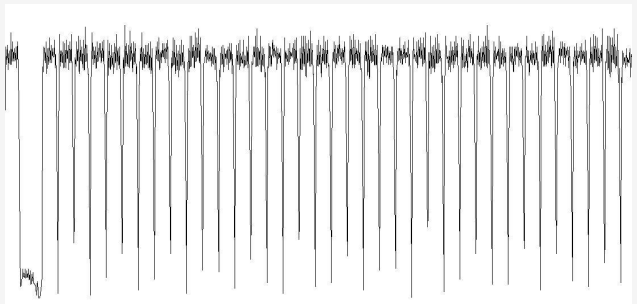
Measure one side-channel leakage s function of t and consider the curve $s(t)$.



SPA/SEMA

Simple Analysis Principle

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

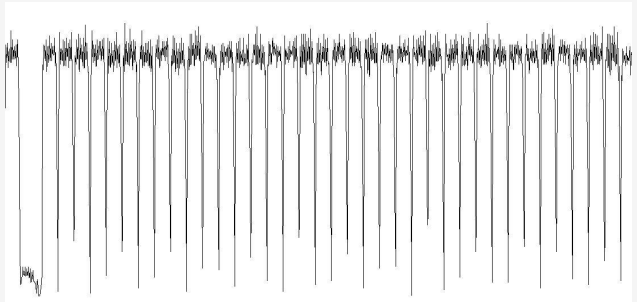


SPA/SEMA

- 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

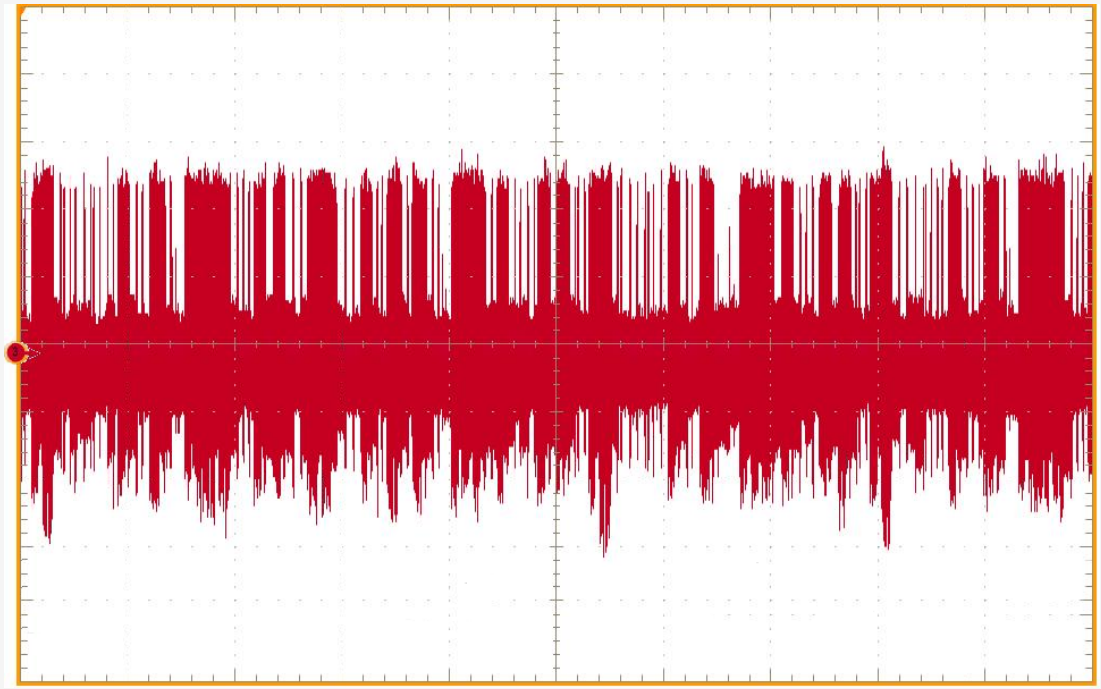
- depicts the behavior of the chip depending on the performed operations / manipulated data
- each measure enables direct reading

Example

Left-to-Right Double and add Algorithm Analysis

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
return R

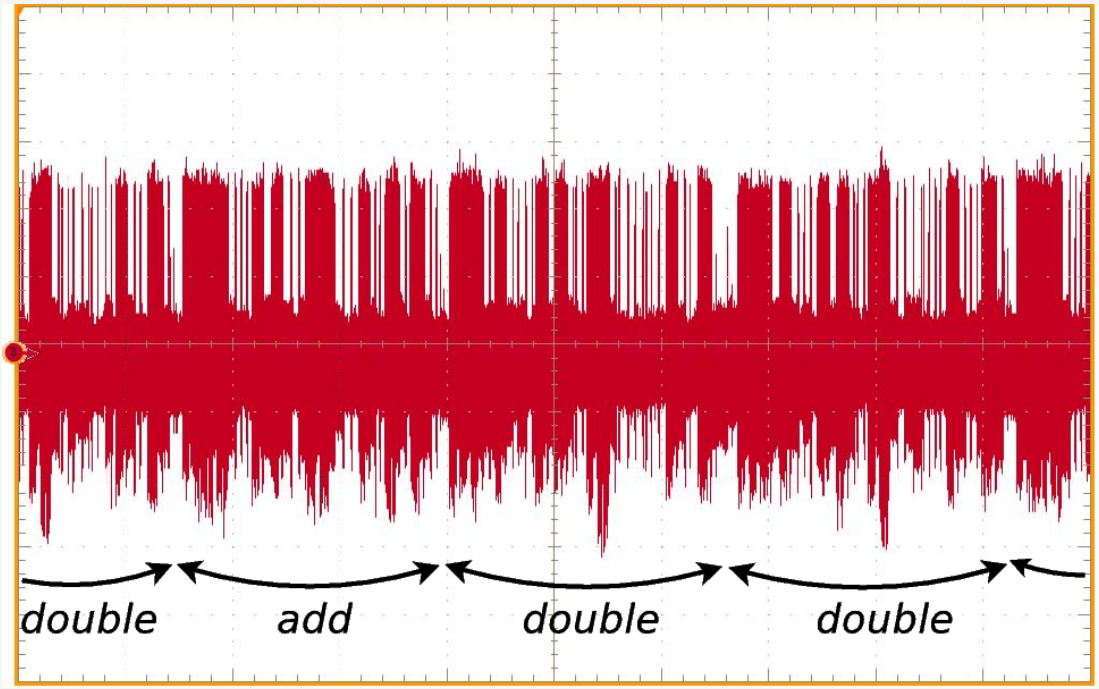


Example

Left-to-Right Double and add Algorithm Analysis

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
return R



Outline

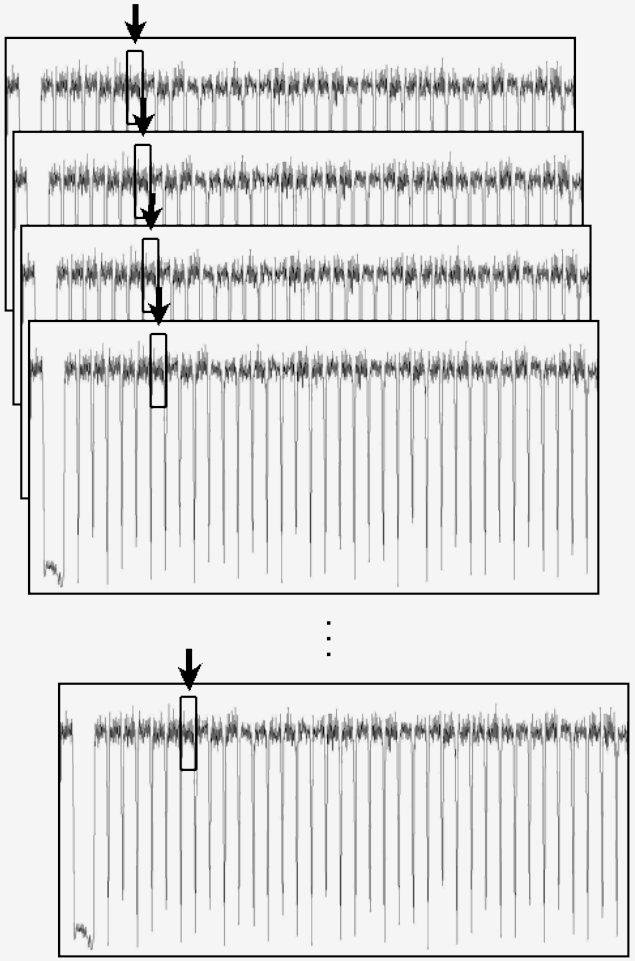
Differential Analysis Principle

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

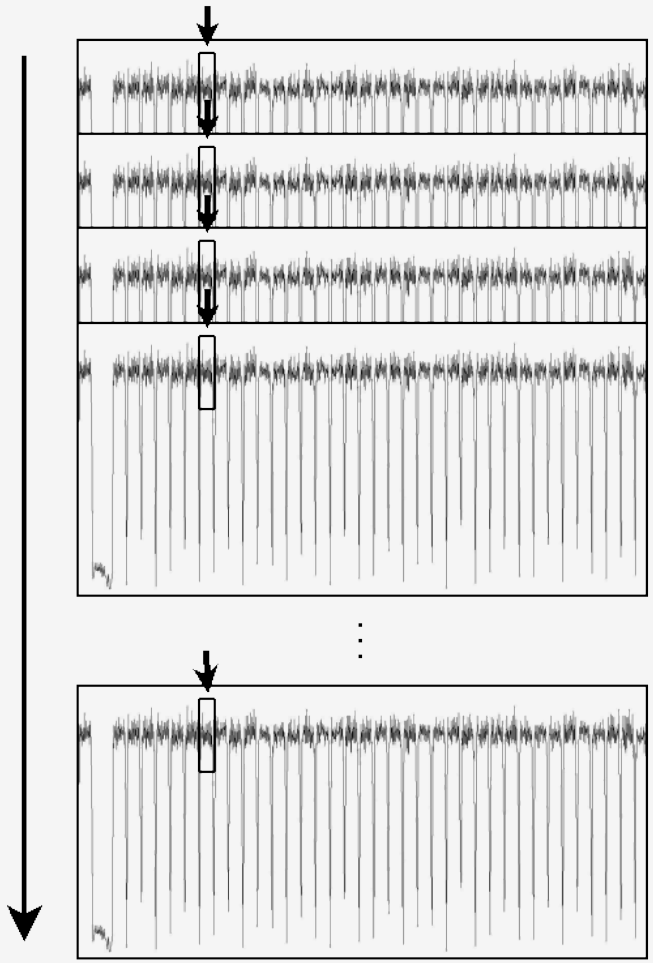
Measure n times a side-channel leakage s function of t and consider the curves $s_1(t), s_2(t), \ldots, s_n(t)$.

- 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

Original DPA/DEMA

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

Original DPA/DEMA

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

- For each possible value (guess) :

Original DPA/DEMA

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

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

Original DPA/DEMA

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

- 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

Original DPA/DEMA

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

- 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

Example

Example

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

Example

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

Example

$$\text{Guess}: \ d_i = 0$$

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

Example

$$\text{Guess}: \ d_i = 0$$

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

Example

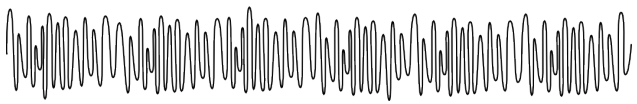
$$\text{Guess} : d_i = 0$$

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

Example

$$\text{Guess} : d_i = 0$$

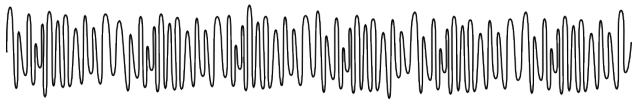
$$\begin{array}{cccc} C_1 & P_1 & Q_1^i & \rightarrow & S_0 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_1 \end{array}$$

Compute $< S_0 > - < S_1 >$: 

Example

Guess : $d_i = 0$

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

Compute $< S_0 > - < S_1 >$: 

Example

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

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

Example

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

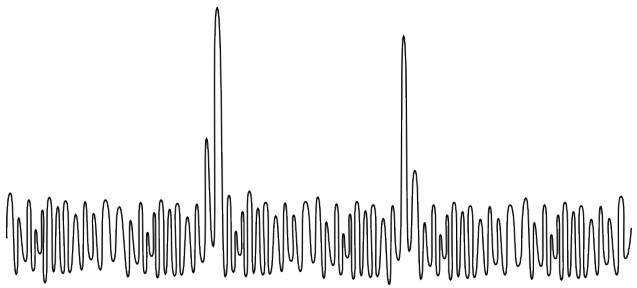
$$\begin{array}{cccc} C_1 & P_1 & Q_1^i & \rightarrow & S_1 \\ C_2 & P_2 & Q_2^i & \rightarrow & S_0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_N & P_N & Q_N^i & \rightarrow & S_0 \end{array}$$

Example

Guess : $d_i = 1$

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

Compute $< S_0 > - < S_1 >$:



CPA/CEMA

Similar to the differential analysis, but uses the Pearson correlation factor :

$$\hat{\rho}_{C,\mathrm{HW}}(t) = \frac{\mathrm{cov}(C(t), \mathrm{HW})}{\sigma_{C(t)}\sigma_{\mathrm{HW}(t)}}$$

CPA/CEMA

Similar to the differential analysis, but uses the Pearson correlation factor :

$$\hat{\rho}_{C,\mathrm{HW}}(t) = \frac{\mathrm{cov}(C(t), \mathrm{HW})}{\sigma_{C(t)}\sigma_{\mathrm{HW}(t)}}$$

- For each possible value (guess) :

CPA/CEMA

Similar to the differential analysis, but uses the Pearson correlation factor :

$$\hat{\rho}_{C,\mathrm{HW}}(t) = \frac{\mathrm{cov}(C(t), \mathrm{HW})}{\sigma_{C(t)}\sigma_{\mathrm{HW}(t)}}$$

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

CPA/CEMA

Similar to the differential analysis, but uses the Pearson correlation factor :

$$\hat{\rho}_{C,\mathrm{HW}}(t) = \frac{\mathrm{cov}(C(t), \mathrm{HW})}{\sigma_{C(t)}\sigma_{\mathrm{HW}(t)}}$$

- 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

Fault Attack on Scalar Multiplication

Fault Attack on Scalar Multiplication

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

Fault Attack 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 Attack 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 Attack 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 $\mathrm{ord}_{\mathcal{E}'}(P')$ is small

Fault Attack 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 $\mathrm{ord}_{\mathcal{E}'}(P')$ is small
- Iterate and apply the chinese reminder theorem to recover k.

Outline

Outline

SPA/SEMA Protection

Mostly three kinds of countermeasures :

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
  - 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
  - Original ECC pattern [Chevallier et al., 2003]
  - Longa ECC patterns [Longa, 2007]
  - Improved ECC pattern [Giraud and Verneuil, 2010]

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

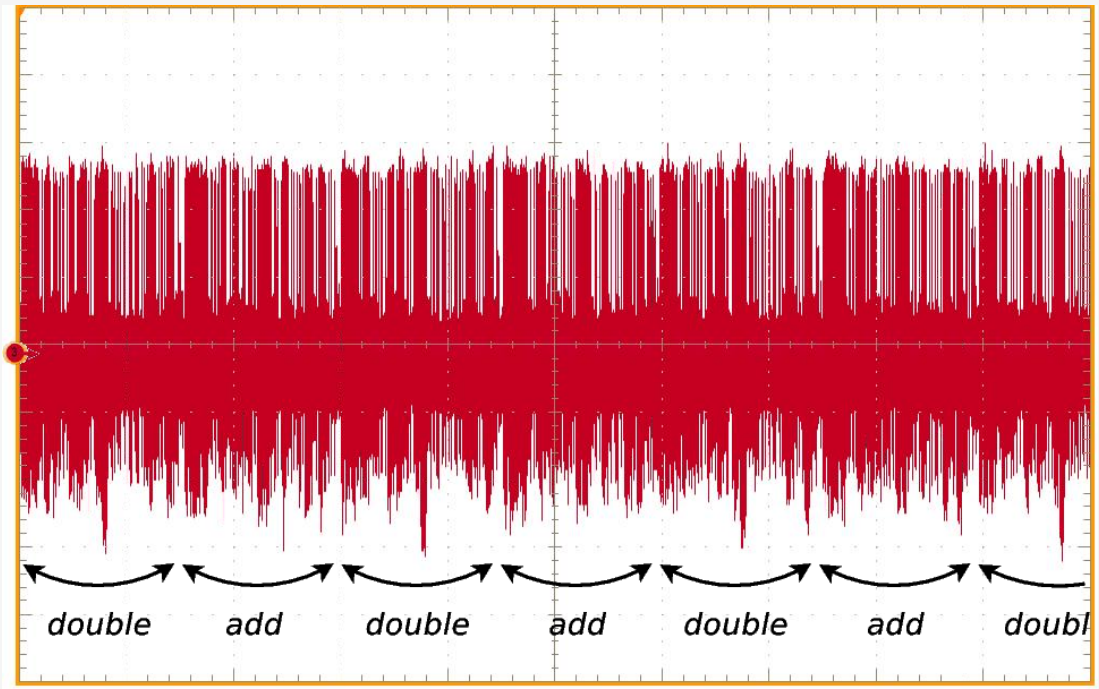
$R, T \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
  else
    $T \leftarrow R + P$
return R

Regular Algorithms

Double and add always

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R, T \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
  else
    $T \leftarrow R + P$
return R



Regular Algorithms

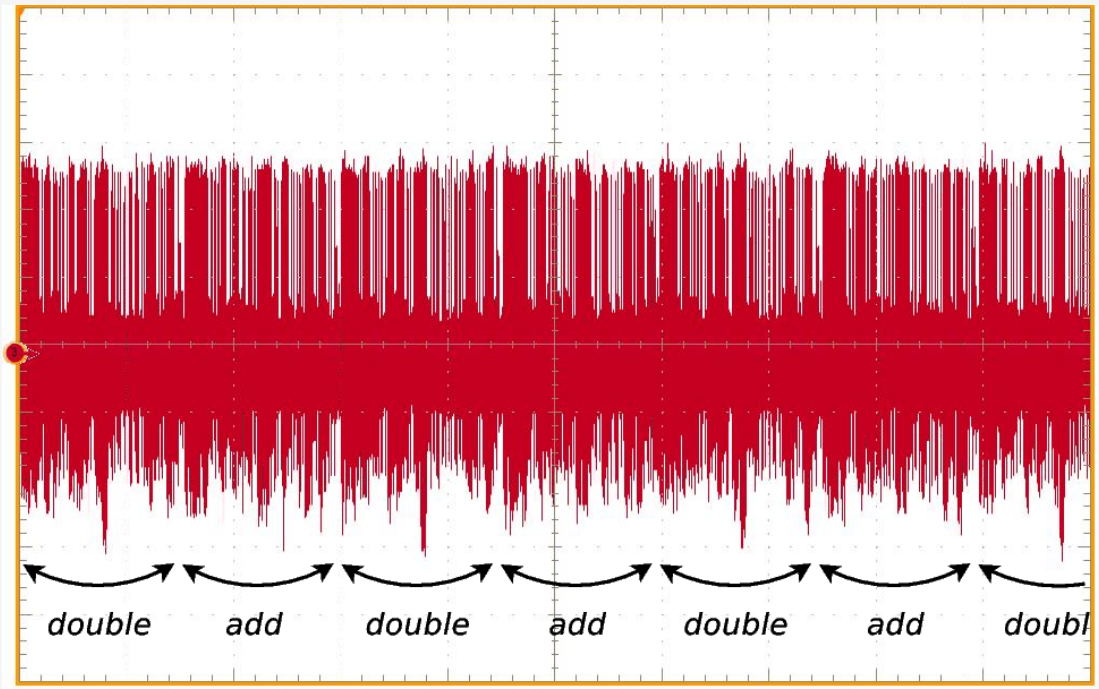
Double and add always

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R, T \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
  else
    $T \leftarrow R + P$
return R

On average :

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



Regular Algorithms

Double and add always

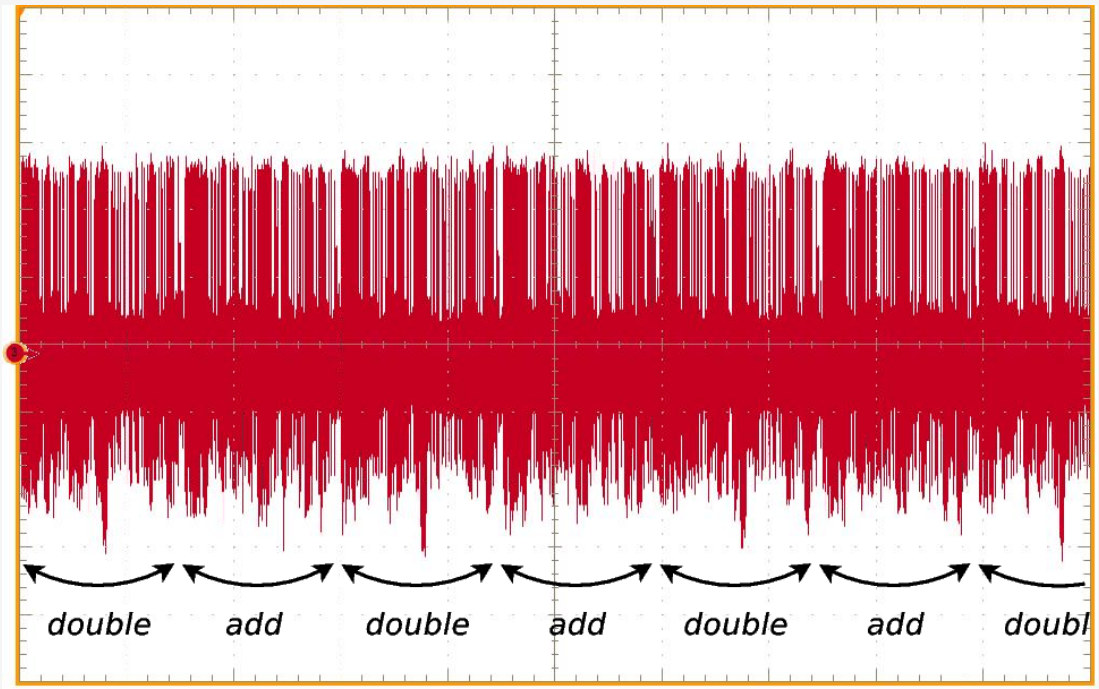
Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R, T \leftarrow \mathcal{O}$
for $i = \ell - 1$ **to** 0 **do**
  $R \leftarrow 2R$
  if $d_i = 1$ **then**
    $R \leftarrow R + P$
  else
    $T \leftarrow R + P$
return R

On average :

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

Prone to safe errors.

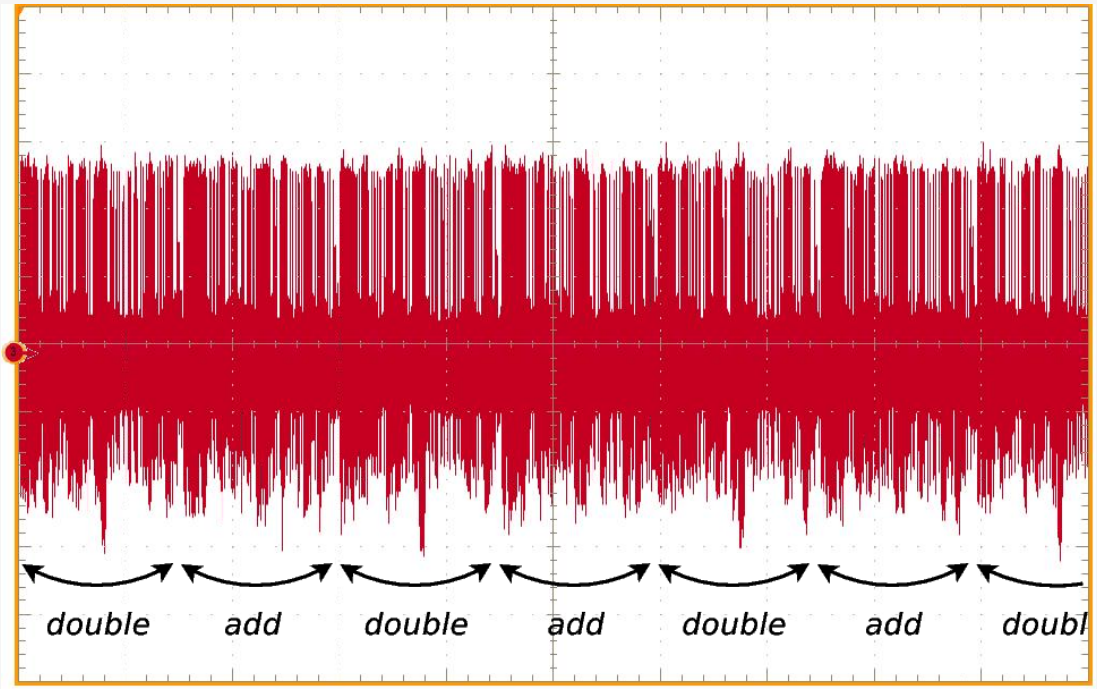


Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R_0 \leftarrow \mathcal{O}$
$R_1 \leftarrow P$
for $i = \ell - 1$ **to** 0 **do**
    $R_{1-d_i} \leftarrow R_0 + R_1$
    $R_{d_i} \leftarrow 2R_{d_i}$
return R_0

Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R_0 \leftarrow \mathcal{O}$
$R_1 \leftarrow P$
for $i = \ell - 1$ **to** 0 **do**
    $R_{1-d_i} \leftarrow R_0 + R_1$
    $R_{d_i} \leftarrow 2R_{d_i}$
return R_0



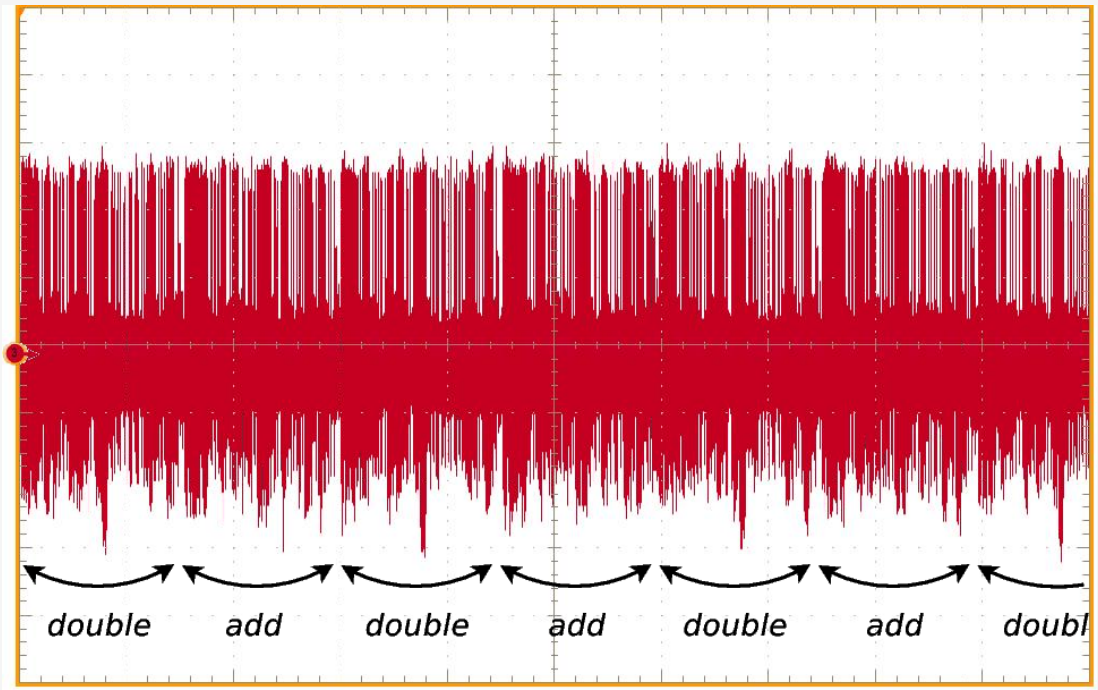
Input: $P \in \mathcal{E}(\mathbb{K}), d \in \mathbb{N}$
Output: $d \cdot P$

$R_0 \leftarrow \mathcal{O}$
$R_1 \leftarrow P$
for $i = \ell - 1$ **to** 0 **do**
  $R_{1-d_i} \leftarrow R_0 + R_1$
  $R_{d_i} \leftarrow 2R_{d_i}$
return R_0

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

A single formula for addition and doubling

Unified Formulas

A single formula for addition and doubling

- Homogeneous projective coordinates : 12M + 6S
- Edwards curves : 10M + 1S
- Twisted Edwards curves : 9M + 1S

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 and 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 and 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 and 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

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

...

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

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]

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

- 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

Classical countermeasures :

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

Outline

Fault Protection

Classical countermeasures

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

Fault Protection

Classical countermeasures

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

Coherence tests

- Montgomery ladder invariant: $R_1 - R_0 = P$
- Right-to-left double-and-add-always

References

Thank you for your attention !

vincent.verneuil@nxp.com
http://vverneuil.net

[1] R.-M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Verkauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. 2006.

[2] J.-L. Danger, S. Guilley, P. Hoogvorst, C. Murdica, and D. Naccache. "A synthesis of side-channel attacks on elliptic curve cryptography in smart-cards". In: *J. Cryptographic Engineering* 4 (2013).

[3] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer, 2003.