Side-Channel Resistant Scalar Multiplication Algorithms over Finite Fields

Alexandre VENELLI[1,2]

François DASSANCE[1]

1 - ATMEL

Secure Microcontroller Solutions

Rousset, FRANCE

2 - IML – ERISCS

Université de la Méditerranée

Marseille, FRANCE

Outline

- Elliptic Curve Cryptosystems (ECC)
- Side-channel attacks against ECC
- Classical side-channel resistant scalar multiplication algorithms
- Our proposed alternatives

Background on ECC (1)

- **Public Key (Asymmetric) cryptosystem**

- **Based on a hard problem :**
  - **Elliptic Curve Discrete Logarithm Problem (ECDLP)**
  - **Given an elliptic curve, points P and Q, find k such that Q=kP**
  - **Hardness of ECDLP = Security level of ECC protocols**
  - **No sub-exponential algorithms known for ECDLP**

Background on ECC (2)

- At the base of ECC operations is finite field algebra with either :
  - Prime finite fields (GF(p)) or
  - Binary extension finite fields (GF(2^m))
- ECC depends on :
  - Finite field selection,
  - Elliptic curve type,
  - Point representation,
  - Protocol,
  - Hardware/software breakdown,
  - Memory available,
  - …

Elliptic Curve

- **Short Weierstrass curves**
  - **Curves used in norms: FIPS, ANSI, …**

- **Elliptic curve on binary field :**

$$E:\ y^2 + xy = x^3 + ax^2 + b \quad (a, b \in GF(2^n), b \neq 0)$$

- **Elliptic curve on prime field :**

$$E:\ y^2 = x^3 + ax + b \quad (a, b \in GF(p), 4a^3 + 27b^2 \neq 0, p > 3)$$

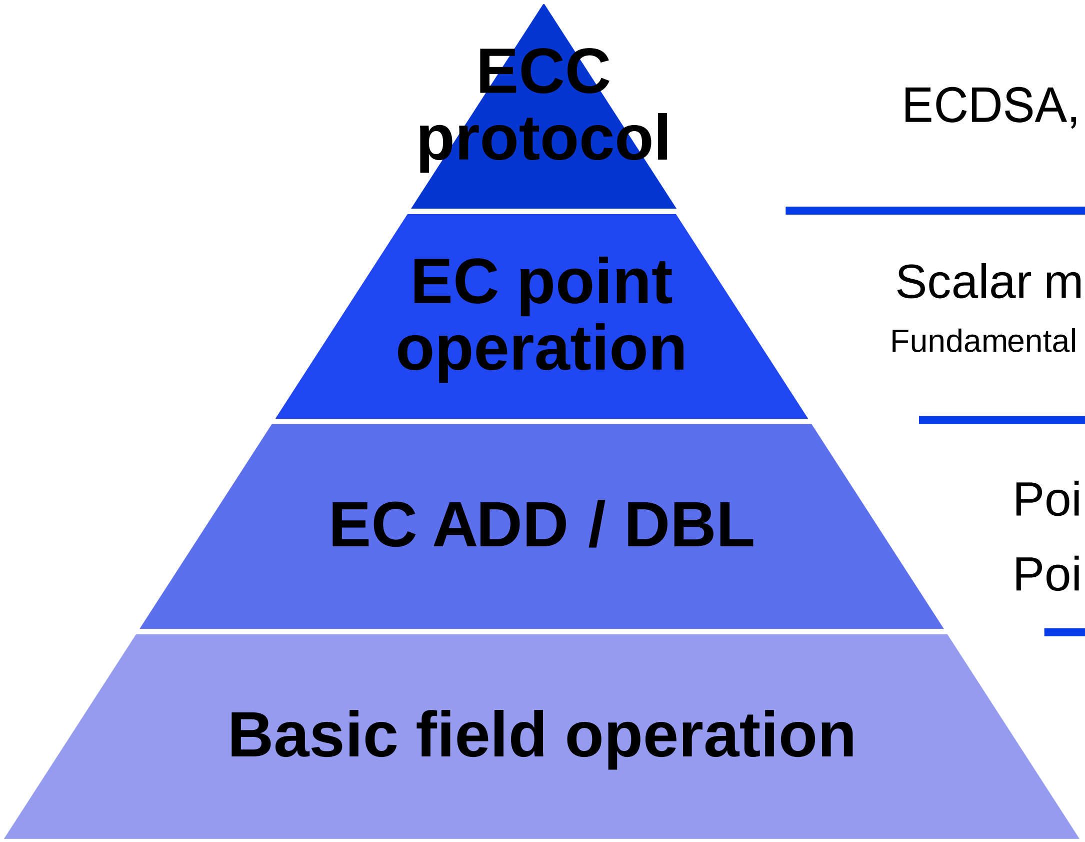
All points satisfying E and infinity point O → Abelian group with addition law

Generic Addition on EC

- **Let** $P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3) \in E$
- **EC Doubling (ECDBL) :** $P_3 = P_1 + P_1 = 2P_1$
- **EC Addition (ECADD) :** $P_3 = P_1 + P_2 \quad (P_1 \neq P_2)$
- **On GF(p), Jacobian coordinates :**
  - **ECDBL = 4M + 5S**
  - **ECADD = 14M + 5S**
- **On GF(2^m), López-Dahab coordinates :**
  - **ECDBL = 3M + 5S**
  - **ECADD = 13M + 4S**

HTTP://WWW.HYPERELLIPTIC.ORG/EFD/

ECC Operations Hierarchy



ECDSA, ECDH, ECIES, …

Scalar multiplication : kP

Fundamental and most time consuming operation

Point addition : $P_3 = P_1 + P_2$

Point doubling : $P_3 = 2P_1$

GF addition :	a + b mod p
GF subtraction :	a – b mod p
GF multiplication :	a * b mod p
GF inversion :	1 / a mod p

Institut de Mathématiques de Luminy

eRISCS

ATMEL®

‘Simplified’ Addition on EC

- Let $P_1 = (X_1, Y_1, Z), P_2 = (X_2, Y_2, Z) \in E$

$$SimpleAdd\ (P_1, P_2) \rightarrow (\widetilde{P}_1, P_1 + P_2) \text{ with } Z_{\widetilde{P}_1} = Z_{P_1+P_2}$$

- On GF(p), Jacobian coordinates :
  - 5M + 2S (Meloni 2007)

- On GF(2^m), Jacobian coordinates :
  - 7M + 2S (this work)

- Formulae not interesting with a standard scalar multiplication algorithm → our propositions

Institut de Mathématiques de Luminy

eRISCS

ATMEL®

Scalar Multiplication on EC

- **Scalar Multiplication** kP
  - **Double-and-add** $P \in E, \quad k = \underbrace{(k_{n-1} \cdots k_0)_2}_{\text{binary representation}}, k_{n-1} = 1$

binary representation

1. $Q \leftarrow P$
2. **From** $i = n-2$ **downto** 0

   $Q \leftarrow 2Q$ **ECDBL**

   if $k_i = 1$ **then** $Q \leftarrow Q + P$ **ECADD**

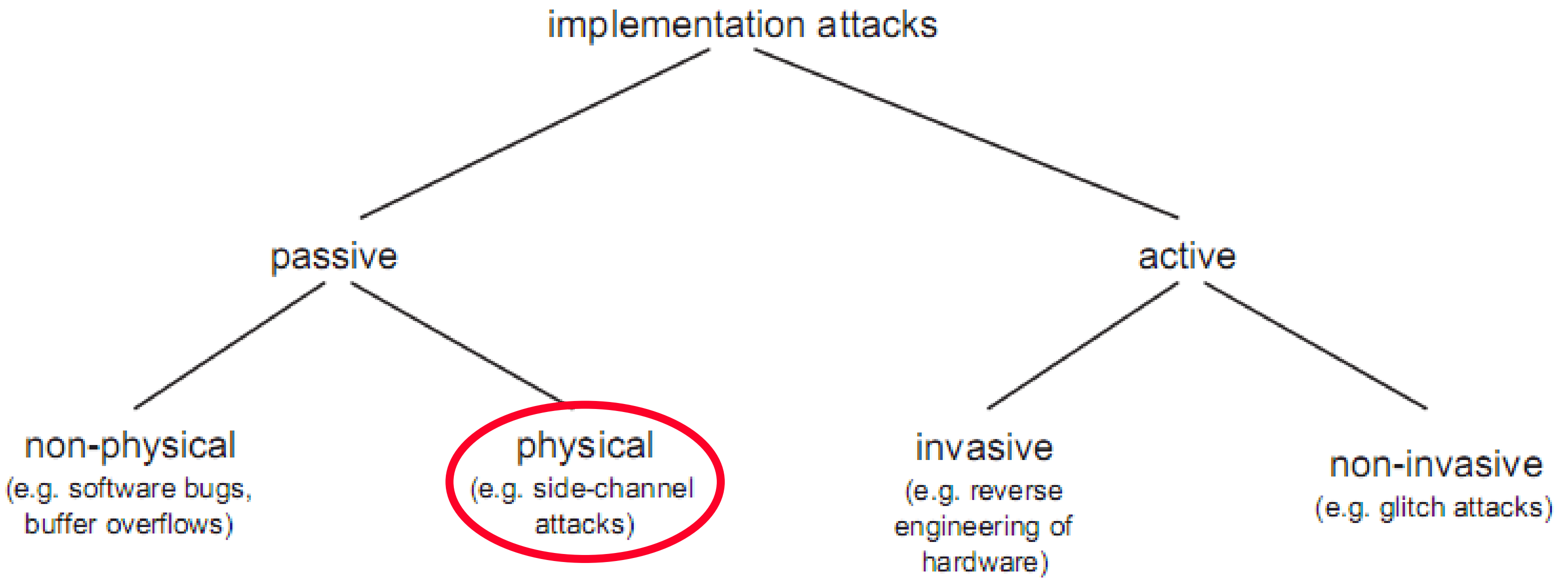
3. **Return** Q

- **Ex :** $51P = (110011)_2 P$

$P \xrightarrow[D]{} 2P \xrightarrow[A]{} 3P \xrightarrow[D]{} 6P \xrightarrow[D]{} 12P \xrightarrow[D]{} 24P \xrightarrow[A]{} 25P$

$\xrightarrow[D]{} 50P \xrightarrow[A]{} 51P$

Implementation Attacks



Families of Side-Channel Attacks

- **Simple Power Analysis (SPA)**
  Observe the power consumption of devices in a single computation and detect the secret key

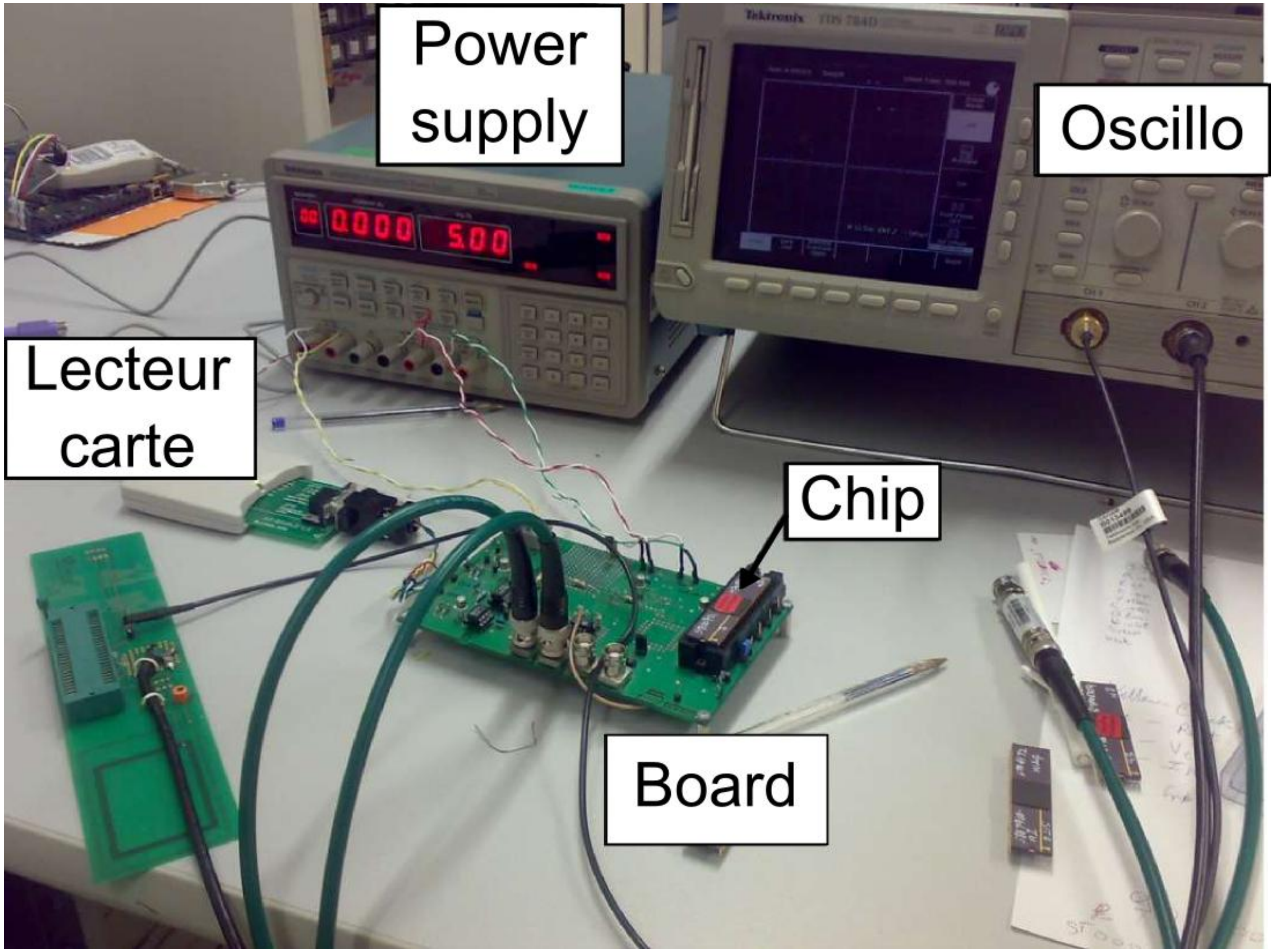
- **Differential Power Analysis (DPA)**
  Observe many power consumptions and analyze these information together with statistic tools

- **Fault Analysis (FA)**
  Using the knowledge of correct results, faulted results and the precise place of induced faults an adversary is able to compute the secret key

Brief History of SCA

- **1996 :**
  - **Kocher et al. → Timing attacks**
  - **Boneh et al. → Fault injection**
- **1998 :**
  - **Kocher et al. → Power analysis**
- **2000 :**
  - **Quisquater et al. → Electromagnetic analysis**

Power Analysis : Cheap and Easy



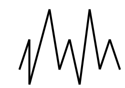
SPA against ECC (Coron 1999)

Algorithm 1: Left-to-right double-and-add

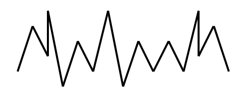
input : $P \in E$ and $k = (k_{n-1} \ldots k_1 k_0)_2$
output: $[k]P \in E$

```
Q ← P
for i ← n − 2 to 0 do
    Q ← [2]P                 ECDBL
    if k_i = 1 then          ECADD
        Q ← Q + P
return Q
```

- **ECDBL**

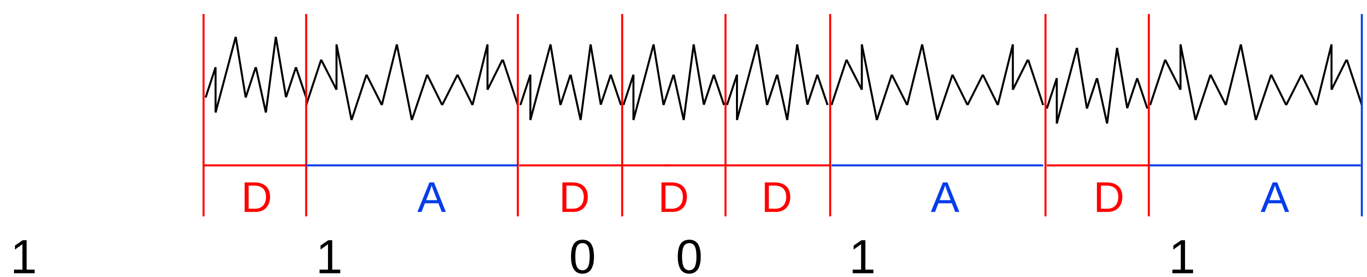


- **ECADD**



Ex : $51P = (110011)_2 P$

Secret revealed !



Double-and-add-always (Coron 1999)

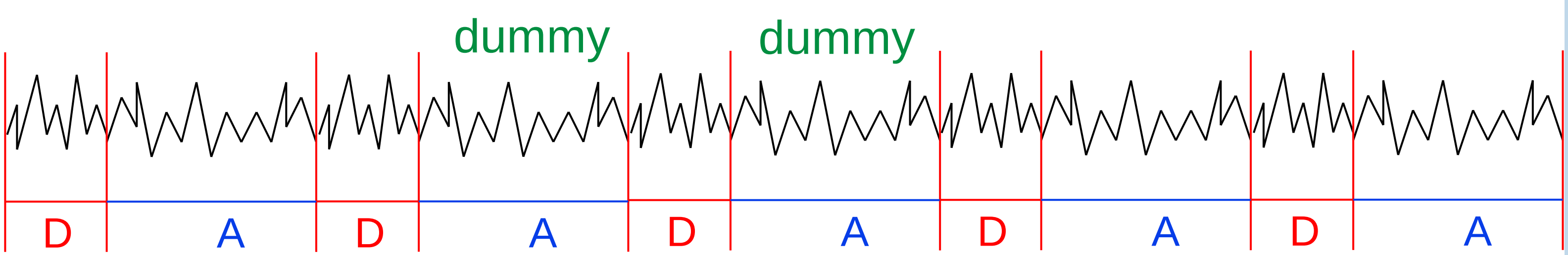
Algorithm 2: Double-and-always-add

input : $P \in E$ and $k = (k_{n-1} \dots k_1 k_0)_2$
output: $[k]P \in E$

$Q_0 \leftarrow P$
for $i \leftarrow n-2$ **to** 0 **do**
  $Q_0 \leftarrow [2]Q_0$ — ECDBL
  $Q_1 \leftarrow Q_0 + P$ — ECADD
  $Q_0 \leftarrow Q_{k_i}$
return Q_0

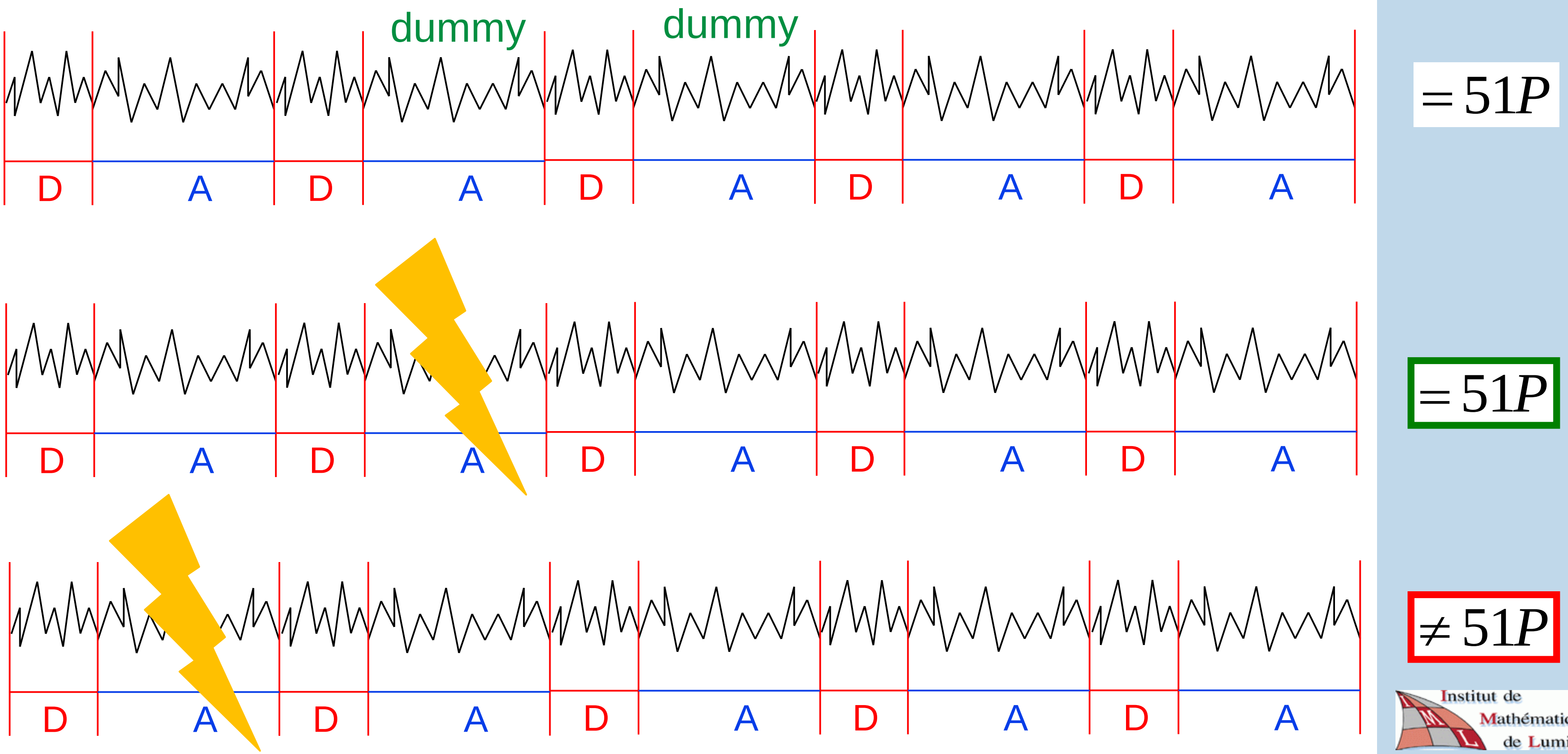
Ex :

$$51P = (110011)_2 P$$



1 0 or 1? 0 or 1? 0 or 1? 0 or 1? 0 or 1?

SPA Resistant but not FA Resistant



Montgomery Ladder (Brier, Joye 2002)

Algorithm 3: Montgomery ladder

input : $P \in E$ and $k = (k_{n-1} \ldots k_1 k_0)_2$
output: $[k]P \in E$

1 $P_0 \leftarrow P$
2 $P_1 \leftarrow [2]P$
3 **for** $i \leftarrow n-2$ **to** 0 **do**
    `// `k_i` = either `0` or `1` and `$\bar{k}_i = 1 - k_i$
4     $P_{\bar{k}_i} \leftarrow P_0 + P_1$
5     $P_{k_i} \leftarrow [2]P_{k_i}$
6 **return** P_0

Montgomery Ladder, it works !

- **Ex :** $51P = (110011)_2 P$

$k_5 = 1$	
P_0	$= P$
P_1	$= 2P$

$k_4 = 1$	
$P_0=P_0+P_1$	$= 3P$
$P_1=2P_1$	$= 4P$

$k_3 = 0$	
$P_1=P_0+P_1$	$= 7P$
$P_0=2P_0$	$= 6P$

$k_2 = 0$	
$P_1=P_0+P_1$	$= 13P$
$P_0=2P_0$	$= 12P$

$k_1 = 1$	
$P_0=P_0+P_1$	$= 25P$
$P_1=2P_1$	$= 26P$

$k_0 = 1$	
$P_0=P_0+P_1$	$= 51P$
$P_1=2P_1$	$= 52P$

Algorithm 3: Montgomery ladder

input : $P \in E$ and $k = (k_{n-1} \dots k_1 k_0)_2$

output: $[k]P \in E$

$P_0 \leftarrow P$

$P_1 \leftarrow [2]P$

for $i \leftarrow n-2$ **to** 0 **do**

// k_i = either 0 or 1 and $\bar{k_i} = 1 - k_i$

$P_{\bar{k_i}} \leftarrow P_0 + P_1$

$P_{k_i} \leftarrow [2]P_{k_i}$

return P_0

Institut de Mathématiques de Luminy

eRISCS

ATMEL

Our Proposition

- **Montgomery ladder idea + 'simplified' addition**

  = side-channel resistant + efficient algorithm

- **<u>Problem :</u>**
  - **Montgomery ladder needs a EC doubling each round**
  - **In the next round, we need for the 'simplified' addition points with the same Z-coordinate**
  - **We would need to transform the output of the doubling so that it has the correct Z-coordinate**
  - **Extremely inefficient**

- **We need to get rid of EC doubling in the algorithm**

  → only use fast 'simplified' additions

Modified Montgomery Ladder

Algorithm 4: Montgomery ladder with additions

input : $P \in E$ and $k = (k_{n-1} \ldots k_1 k_0)_2$

output: $[k]P \in E$

$P_1 \leftarrow P$;

$P_2 \leftarrow [2]P$;

for $i \leftarrow n-2$ **to** 0 **do**

  $P_1 \leftarrow P_1 + P_2$;

  $P_2 \leftarrow P_1 + (-1)^{\bar{k}_i} P$;

return P_1

Modified Montgomery Ladder, still works !

- **Ex :** $51P = (110011)_2 P$

$k_5 = 1$	
P_1	$= P$
P_2	$= 2P$

$k_4 = 1$	
$P_1=P_1+P_2$	$= 3P$
$P_2=P_1+P$	$= 4P$

$k_3 = 0$	
$P_1=P_1+P_2$	$= 7P$
$P_2=P_1-P$	$= 6P$

$k_2 = 0$	
$P_1=P_1+P_2$	$= 13P$
$P_2=P_1-P$	$= 12P$

$k_1 = 1$	
$P_1=P_1+P_2$	$= 25P$
$P_2=P_1+P$	$= 26P$

$k_0 = 1$	
$P_1=P_1+P_2$	$= 51P$
$P_2=P_1+P$	$= 52P$

Algorithm 4: Montgomery ladder with additions

input : $P \in E$ and $k = (k_{n-1} \dots k_1 k_0)_2$
output: $[k]P \in E$

$P_1 \leftarrow P$;
$P_2 \leftarrow [2]P$;
for $i \leftarrow n-2$ **to** 0 **do**
  $P_1 \leftarrow P_1 + P_2$;
  $P_2 \leftarrow P_1 + (-1)^{\bar{k}_i} P$;
return P_1

Institut de Mathématiques de Luminy

Tweak 'Simplified' Addition

- **Problem :** we need the point P with the correct Z-coordinate at each round
- Computing both addition and subtraction in a modified 'simplified' addition

$$SimpledAddSub \rightarrow (\widetilde{P}_1, P_1 + P_2, P_1 - P_2)$$

Complexity in field operations

	GF(p)	GF(2^m)
SimpleAdd	5M+2S	7M+2S
SimpleAddSub	6M+3S	11M+2S

Proposed Algorithm

Algorithm 5: BasicScalarMult

input : $P \in E$ and $k = (k_{n-1} \dots k_1 k_0)_2$

output: $[k]P \in E$

$P_1 \leftarrow [2]P$

$P_2 \leftarrow P$

// We assume $Z_{P_1} = Z_{P_2}$

for $i \leftarrow n-2$ **to** 0 **do**

  $Q \leftarrow$ SimpleAddSub(P_1, P_2)

  $P_1 \leftarrow Q[1]$ /* $P_1 \leftarrow (P_1 + P_2)$ */

  $P_2 \leftarrow Q[2]$ /* $P_2 \leftarrow (P_1 - P_2) = P$ */

  $Q \leftarrow$ SimpleAddSub(P_1, P_2)

  $P_1 \leftarrow Q[k_i]$ /* $P_1 \leftarrow \tilde{P}_1$ or $P_1 \leftarrow P_1 + P_2$ */

  $P_2 \leftarrow Q[2\bar{k}_i]$ /* $P_2 \leftarrow \tilde{P}_1$ or $P_2 \leftarrow P_1 - P_2$ */

return P_2

Efficiency Evaluation on GF(2^m)

Algorithm	*Complexity (per bit of scalar)*
Generic Montgomery Ladder	18M+10S ≈ 28M
Lopez et al. (1999)	6M+5S ≈ 11M
BasicScalarMult	**22M+4S ≈ 26M**

Institut de Mathématiques de Luminy

eRISCS

ATMEL®

Efficiency Evaluation on GF(p)

Algorithm	*Complexity (per bit of scalar)*
Generic Montgomery Ladder	12M+13S ≈ 25M
Brier et al. (2002)	15M+5S ≈ 20M
Izu et al. (2002)	13M+4S ≈ 17M
BasicScalarMult	**12M+6S ≈ 18M**
OptScalarMult	**10M+6S ≈ 16M**

Conclusion

- Side-channel resistance is a major issue in constrained devices…

- … however efficiency should not suffer

- We wanted to improve scalar multiplication, the main part of ECC, on these 2 points

- Our results :
  - an alternative algorithm on $GF(2^m)$,
  - very interesting replacement on GF(p)

Thank you. Questions ?