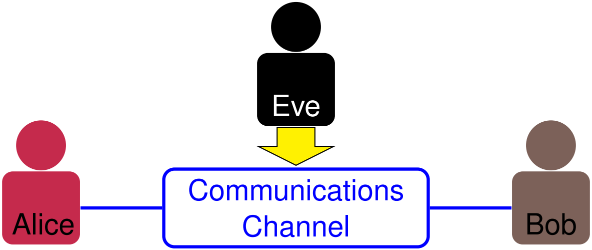
Using Modular Extension to Provably Protect Edwards Curves Against Fault Attacks

Margaux Dugardin, Sylvain Guilley, Martin Moreau, Zakaria Najm, Pablo Rauzy

PROOFS 2016 - Santa Barbara, CA

Introduction

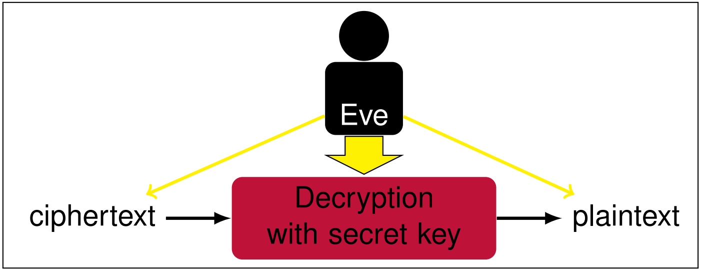


We need :

- Encryption/Decryption
- Key exchange
- Signature

$\Rightarrow$ Asymmetric cryptography

Introduction



Eve is able to:

- observe the Alice's computation
- change the input
- have the output
- inject a fault during the computation

Fault attacks

Fault attacks:

- Safe-error attacks
- Cryptosystems parameters alteration
- Differential Fault Analysis (DFA) e.g. BellCoRe attack, sign-change attacks.

Fault model:

- Randomizing faults (Boneh et al, EUROCRYPT 1997)
- Zeroing faults (Clavier, CHES 2007)
- Instruction skip faults (Moro et al, JCE 2014)

Classical Algorithm Scalar Multiplication

Algorithm 1 Double and Add Left-to-Right

Input: $P \in E(\mathbb{F}_p)$, $k = (k_{n-1}k_{n-2}\dots k_0)_2, \forall i,\ k_i \in \{0,1\}$
Output: $[k]P$

1: $Q \leftarrow \mathcal{O}$ ▷ the point at infinity
2: **for** $i = n-1$ **downto** 0 **do**
3: $Q \leftarrow 2Q$ ▷ EC-DBL
4: **if** $k_i = 1$ **then**
5: $Q \leftarrow Q + P$ ▷ EC-ADD
6: **end if**
7: **end for**

Fault Attack: Invalid input point

Biehl et al, CRYPTO 2000

Algorithm 1 Double and Add Left-to-Right

Input: $\boxed{P \in \text{weak curve}}$, $k = (k_{n-1}k_{n-2}\dots k_0)_2, \forall i,\ k_i \in \{0,1\}$
Output: $[k]P$

1: $Q \leftarrow \mathcal{O}$ ▷ the point at infinity
2: **for** $i = n-1$ **downto** 0 **do**
3: $\quad Q \leftarrow 2Q$ ▷ EC-DBL
4: $\quad$ **if** $k_i = 1$ **then**
5: $\qquad Q \leftarrow Q + P$ ▷ EC-ADD
6: $\quad$ **end if**
7: **end for**

Fault Attack: Invalid input point

Biehl et al, CRYPTO 2000

Algorithm 1 Double and Add Left-to-Right

Input: $\boxed{P \in \text{weak curve}}$, $k = (k_{n-1}k_{n-2}\dots k_0)_2, \forall i,\ k_i \in \{0, 1\}$
Output: $[k]P$

1: if P is not on the curve $E(\mathbb{F}_p)$ then `error`
2: $Q \leftarrow \mathcal{O}$ ▷ the point at infinity
3: **for** $i = n - 1$ **downto** 0 **do**
4: $\quad Q \leftarrow 2Q$ ▷ EC-DBL
5: $\quad$ **if** $k_i = 1$ **then**
6: $\quad\quad Q \leftarrow Q + P$ ▷ EC-ADD
7: $\quad$ **end if**
8: **end for**
9: if Q is not on the curve $E(\mathbb{F}_p)$ then `error` else return Q

Countermeasure: Verify the input/output point and the curve parameters

Sign-change fault attack

Blömer et al, LNCS 2006

Algorithm 1 Double and Add Left-to-Right

Input: $P \in E(\mathbb{F}_p)$, $k = (k_{n-1}k_{n-2}\dots k_0)_2, \forall i,\ k_i \in \{0,1\}$
Output: $[k]P$

```
if P is not on the curve E(F_p) then error
Q ← O                                          ▷ the point at infinity
for i = n − 1 downto 0 do
    Q ← 2Q                                     ▷ Sign-change fault at i = 0
    if k_i = 1 then
        Q ← Q + P                              ▷ EC-ADD
    end if
end for
if Q is not on the curve E(F_p) then error else return Q
```

Countermeasure: Verify the input/output point and the curve parameters

$\Rightarrow$ INEFFECTIVE

Sign-change fault attack

Blömer et al, LNCS 2006

Algorithm 1 Double and Add Left-to-Right

Input: $P \in E(\mathbb{F}_p)$, $k = (k_{n-1}k_{n-2}\dots k_0)_2, \forall i,\ k_i \in \{0,1\}$
Output: $[k]P$

if P is not on the curve $E(\mathbb{F}_p)$ then `error`
$Q \leftarrow \mathcal{O}$ ▷ the point at infinity
for $i = n-1$ **downto** 0 **do**
    $\boxed{Q \leftarrow 2Q}$ ▷ Sign-change fault at $i = 0$
    if $k_i = 1$ **then**
        $Q \leftarrow Q + P$ ▷ EC-ADD
    end if
end for
if Q is not on the curve $E(\mathbb{F}_p)$ then `error` else return Q

$$\left\{\begin{array}{lcl} Q & = & [k_0 + 2\sum_{i=1}^{n-1} k_i 2^{i-1}]P \\ Q^* & = & [k_0 - 2\sum_{i=1}^{n-1} k_i 2^{i-1}]P \end{array}\right. \implies Q + Q^* = [2k_0]P.$$

Sign-change fault attack

Blömer et al, LNCS 2006

Algorithm 1 Double and Add Left-to-Right

Input: $P \in E(\mathbb{F}_p)$, $k = (k_{n-1}k_{n-2}\dots k_0)_2, \forall i,\ k_i \in \{0,1\}$
Output: $[k]P$

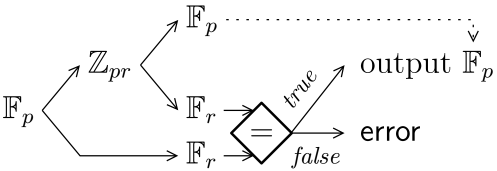
```
if P is not on the curve E(F_p) then error
Q ← O                                          ▷ the point at infinity
for i = n − 1 downto 0 do
    Q ← 2Q                                     ▷ Sign-change fault at i = 1
    if k_i = 1 then
        Q ← Q + P                              ▷ EC-ADD
    end if
end for
if Q is not on the curve E(F_p) then error else return Q
```

$$\begin{cases} Q &= [2k_1 + k_0 + 4\sum_{i=2}^{n-1} k_i 2^{i-2}]P \\ Q^* &= [2k_1 + k_0 - 4\sum_{i=2}^{n-1} k_i 2^{i-2}]P \end{cases} \implies Q + Q^* = [2(2k_1 + k_0)]P.$$

Shamir countermeasures

- Computional protections against fault injection:

$\Rightarrow$ Modular extension



BOS countermeasure

Blömer et al, LNCS 2006

Algorithm 2 ECSM protected with BOS countermeasure

Input: $P \in E(\mathbb{F}_p)$, $k \in \{1, \ldots, ord(P) - 1\}$
Output: $Q = [k]P \in E(\mathbb{F}_p)$

1: Choose a small prime r, a curve $E(\mathbb{F}_r)$, and a point P_r on that curve.
2: Determine the combined curve $E(\mathbb{Z}_{pr})$ and point P_{pr} using the CRT.
3: $(X_{pr} : Y_{pr} : Z_{pr}) = \text{ECSM}(P_{pr}, k, pr)$
4: $(X_r : Y_r : Z_r) = \text{ECSM}(P_r, k, r)$
5: **if** $(X_{pr} \bmod r : Y_{pr} \bmod r : Z_{pr} \bmod r) = (X_r : Y_r : Z_r)$ **then**
6: return $(X_{pr} \bmod p : Y_{pr} \bmod p : Z_{pr} \bmod p)$
7: **else**
8: return `error`
9: **end if**

BOS countermeasure

Blömer et al, LNCS 2006

Algorithm 3 ECSM protected with BOS countermeasure

Input: $P \in E(\mathbb{F}_p)$, $k \in \{1, \ldots, ord(P) - 1\}$
Output: $Q = [k]P \in E(\mathbb{F}_p)$

1: Choose a small prime r, a curve $E(\mathbb{F}_r)$, and a point P_r on that curve.
2: Determine the combined curve $E(\mathbb{Z}_{pr})$ and point P_{pr} using the CRT.
3: $(X_{pr} : Y_{pr} : Z_{pr}) = \text{ECSM}(P_{pr}, k, pr)$
4: $(X_r : Y_r : Z_r) = \text{ECSM}(P_r, k, r)$
5: **if** $\boxed{(X_{pr} \bmod r : Y_{pr} \bmod r : Z_{pr} \bmod r) = (X_r : Y_r : Z_r)}$ **then**
6: return $(X_{pr} \bmod p : Y_{pr} \bmod p : Z_{pr} \bmod p)$
7: **else**
8: return `error`
9: **end if**

THALES
TELECOM ParisTech

BOS countermeasure

Blömer et al, LNCS 2006

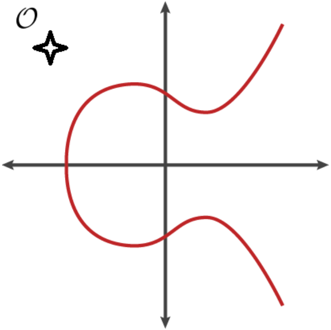
Algorithm 4 ECSM protected with BOS countermeasure

Input: $P \in E(\mathbb{F}_p)$, $k \in \{1, \ldots, ord(P) - 1\}$
Output: $Q = [k]P \in E(\mathbb{F}_p)$

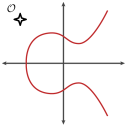
1: Choose a small prime r, a curve $E(\mathbb{F}_r)$, and a point P_r on that curve.
2: Determine the combined curve $E(\mathbb{Z}_{pr})$ and point P_{pr} using the CRT.
3: $(X_{pr} : Y_{pr} : Z_{pr}) = \text{ECSM}(P_{pr}, k, pr)$
4: $(X_r : Y_r : Z_r) = \text{ECSM}(P_r, k, r)$ ▷ without test in EC-ADD
5: **if** $(X_{pr} \bmod r : Y_{pr} \bmod r : Z_{pr} \bmod r) = (X_r : Y_r : Z_r)$ **then**
6:     return $(X_{pr} \bmod p : Y_{pr} \bmod p : Z_{pr} \bmod p)$
7: **else**
8:     return `error`
9: **end if**

BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

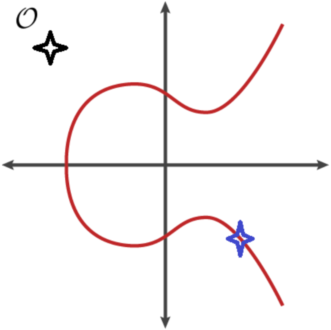


Elliptic curve on $\mathbb{F}_r$

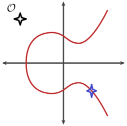


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

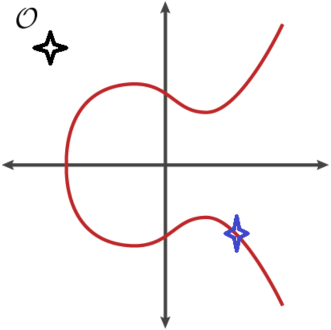


Elliptic curve on $\mathbb{F}_r$

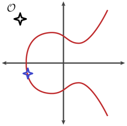


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

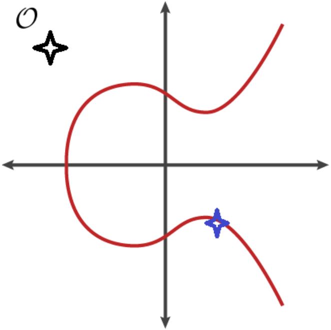


Elliptic curve on $\mathbb{F}_r$

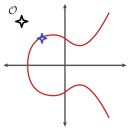


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

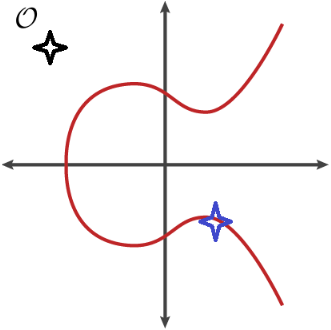


Elliptic curve on $\mathbb{F}_r$

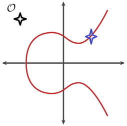


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

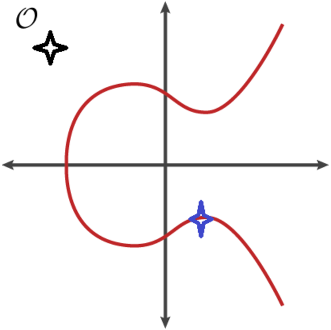


Elliptic curve on $\mathbb{F}_r$

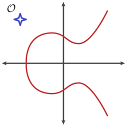


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

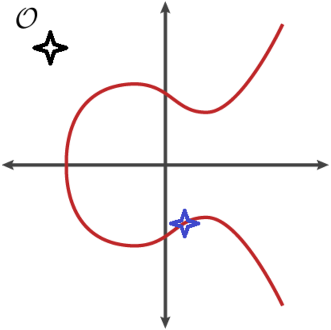


Elliptic curve on $\mathbb{F}_r$

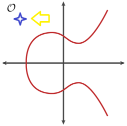


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

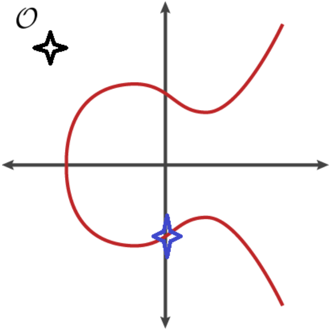


Elliptic curve on $\mathbb{F}_r$

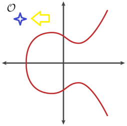


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

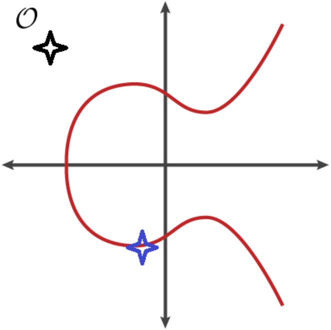


Elliptic curve on $\mathbb{F}_r$

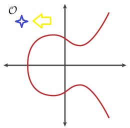


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

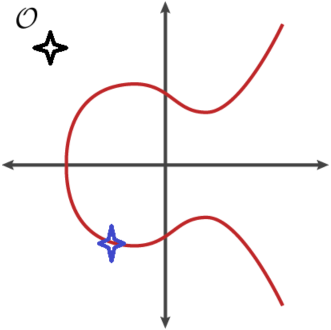


Elliptic curve on $\mathbb{F}_r$

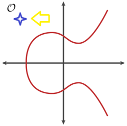


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

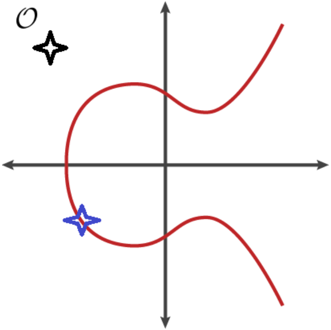


Elliptic curve on $\mathbb{F}_r$

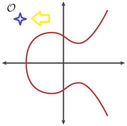


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$

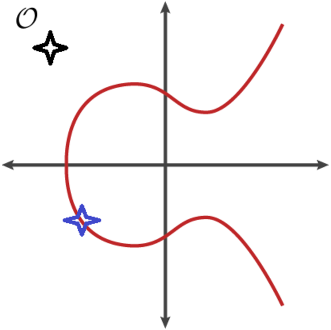


Elliptic curve on $\mathbb{F}_r$

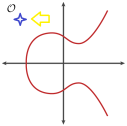


BOS is incorrect in Weierstrass curve

Elliptic curve on $\mathbb{Z}_{pr}$



Elliptic curve on $\mathbb{F}_r$



Without fault injection, there are an error because $\mathcal{O} \neq [k]P \mod r$

Our contributions

- Security analysis of modular extension countermeasure
- Correct the BOS countermeasure using Edwards and Twisted Edward curve

Security Analysis of Modular Extension

Definition 1: Fault model

We consider an attacker who can fault data by randomizing or zeroing any intermediate variable, and fault code by skipping any number of consecutive instructions.

Definition 2: Attack order

We call order of the attack the number of faults (in the sense of Def. 1) injected during the target execution.

Definition 3: Secure algorithm

An algorithm is said secure if it is correct and if it either returns the right result or an error constant when faults have been injected, with an overwhelming probability.

Security Analysis of Modular Extension

Theorem 1: Security of test-free modular extension

Test-free algorithms protected using the modular extension technique, are secure as per Def. 3 . In particular, the probability of non-detection is inversely proportional to the security parameter r.

Faulted results are polynomials of faults.

- We give the formal name $\widehat{x}$ to any faulted variable x.
- For convenience, we denote them by $\widehat{x_i}$, $1 \leq i \leq n$, where $n \geq 1$ is the number of injected faults.
- The result of asymmetric computation consists in additions, subtractions, and multiplications of those formal variables (and inputs). Such expression is a multivariate polynomial.
- If the inputs are fixed, then the polynomial has only n formal variables. We call it $P(\widehat{x_1}, \ldots, \widehat{x_n})$.
- For now, let us assume that $n = 1$, i.e., that we face a single fault. Then P is a monovariate polynomial. Its degree d is the multiplicative depth of $\widehat{x_1}$ in the result.

Non-detection probability is inversely proportional to r

A fault is not detected if and only if $P(\widehat{x_1}) = P(x_1) \mod r$, whereas $P(\widehat{x_1}) \neq P(x_1) \mod p$.

As the faulted variable $\widehat{x_1}$ can take any value in $\mathbb{Z}_{pr}$, the non-detection probability $\mathbb{P}_{\text{n.d.}}$ is given by:

$$\begin{aligned}
\mathbb{P}_{\text{n.d.}} &= \frac{1}{pr-1} \cdot \sum_{\widehat{x_1} \in \mathbb{Z}_{pr} \setminus \{x_1\}} 1_{P(\widehat{x_1}) = P(x_1) \mod r} \\
&= \frac{1}{pr-1} \cdot \Big(-1 + p \sum_{\widehat{x_1}=0}^{r-1} 1_{P(\widehat{x_1}) = P(x_1) \mod r} \Big). \qquad (1)
\end{aligned}$$

Let $\widehat{x_1} \in \mathbb{Z}_r$, if $P(\widehat{x_1}) = P(x_1) \mod r$, then $\widehat{x_1}$ is a root of the polynomial $\Delta P(\widehat{x_1}) = P(\widehat{x_1}) - P(x_1)$ in $\mathbb{Z}_r$. We denote by $\#\text{roots}(\Delta P)$ the number of roots of ΔP over $\mathbb{Z}_r$. Thus (1) computes $(p \times \#\text{roots}(\Delta P) - 1)/(pr-1) \approx \#\text{roots}(\Delta P)/r$.

Theoretical Upper-Bound for #roots

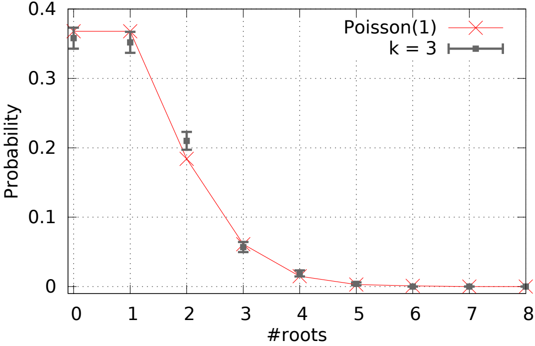
$\#$roots(ΔP) can be as high as the degree d of ΔP in $\mathbb{Z}_r$, i.e., $\min(d, r-1)$. However, in practice, ΔP looks like a random polynomial over the finite field $\mathbb{Z}_r$, for several reasons:

- inputs are random numbers in most cryptographic algorithms, such as probabilistic signature schemes,
- the coefficients of ΔP in $\mathbb{Z}_r$ are randomized due to the reduction modulo r.

Theoretical Upper-Bound for #roots

Leont'ev proved in Mathematical Notes 2006 that if P is a random polynomial in $\mathbb{F}_p$ then $\#\text{roots}(P) \sim \text{Poisson}(\lambda = 1)$, i.e., $\mathbb{P}(\#\text{roots}(P) = n) = \frac{1}{en!}$.

In the case of $\Delta P \bmod r$, we know that there is always at least one root, when $\widehat{x_1} = x_1$



Non-detection probability is inversely proportional to r.

Correct BOS countermeasure

Definition 4: Edwards curves

On the finite field $\mathbb{F}_p$ with p a prime number, an elliptic curve in Edwards form has parameters c, d in the finite field $\mathbb{F}_p$ and coordinates (x, y) satisfying the following equation:

$$x^2 + y^2 = c^2(1 + dx^2y^2), \tag{2}$$

with $cd(1 - c^4d) \neq 0$.

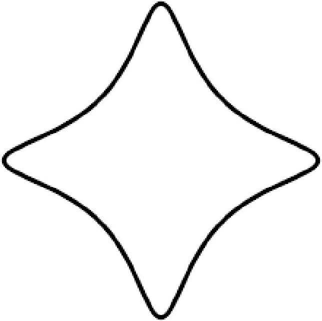
The main advantage to use the Edwards curves is that addition formulas ECADD-complete are :

- complete
- unified

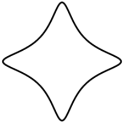
$\Rightarrow$ no test in EC-ADD-unified formula

Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

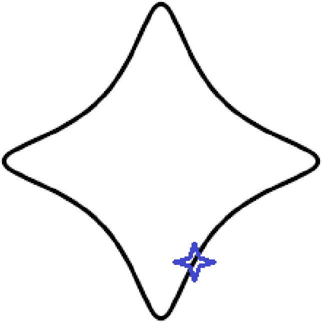


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

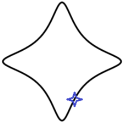


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

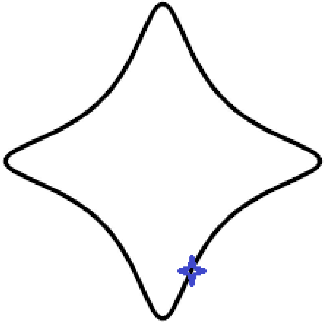


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

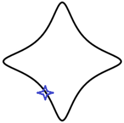


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

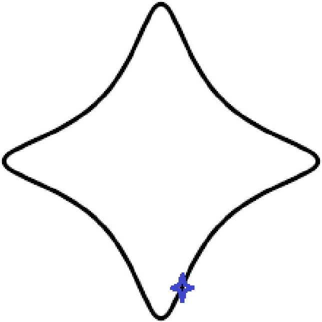


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

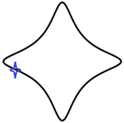


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

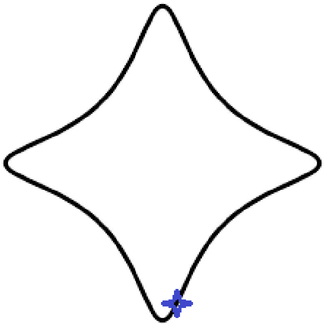


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

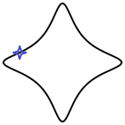


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

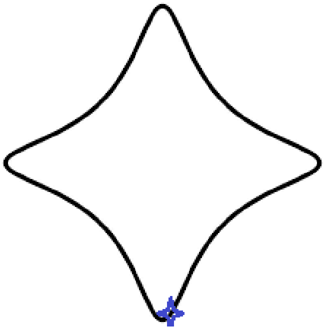


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

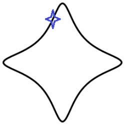


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

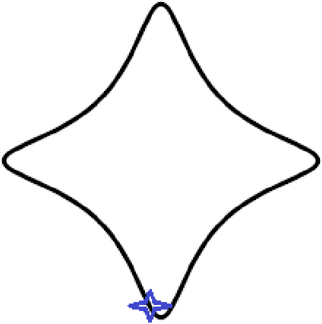


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

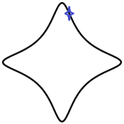


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

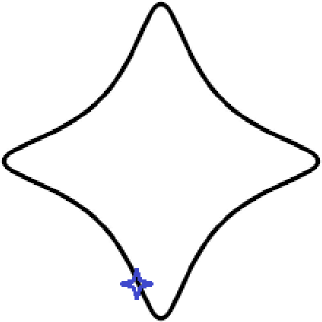


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

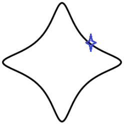


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

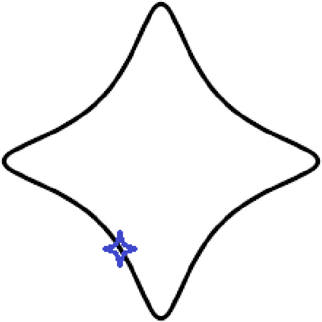


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

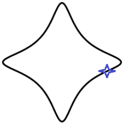


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

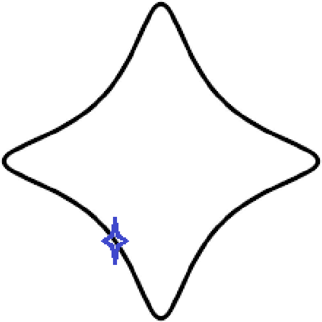


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

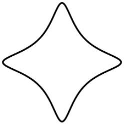


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

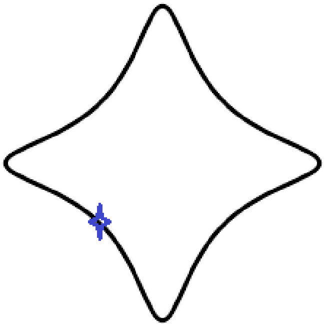


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

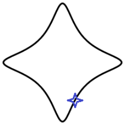


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

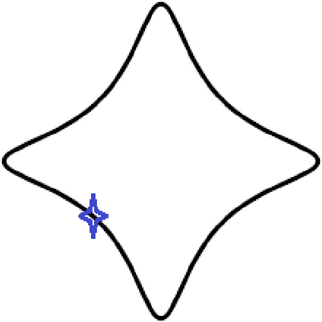


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

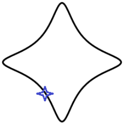


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

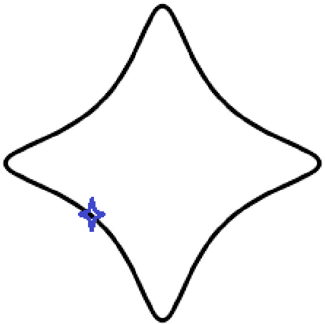


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

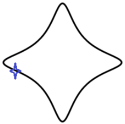


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

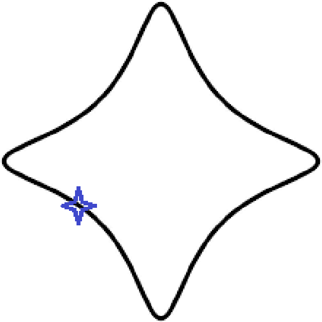


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

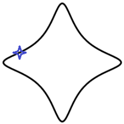


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$ (Edwards form)

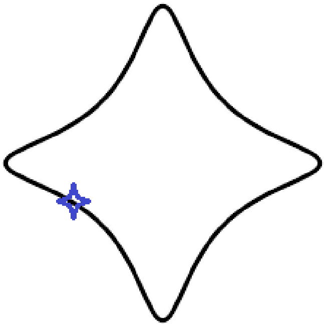


Elliptic curve on $\mathbb{F}_r$ (Edwards form)

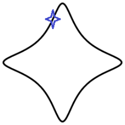


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

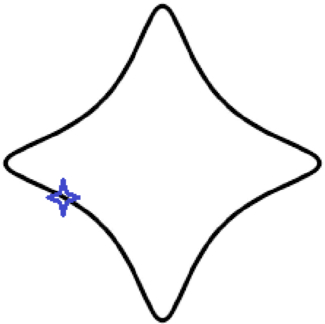


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

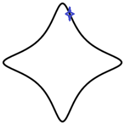


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

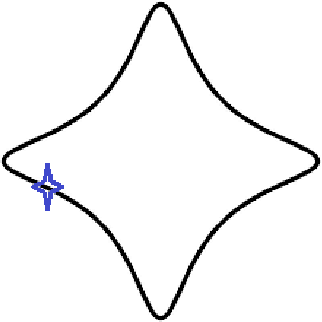


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

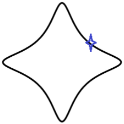


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

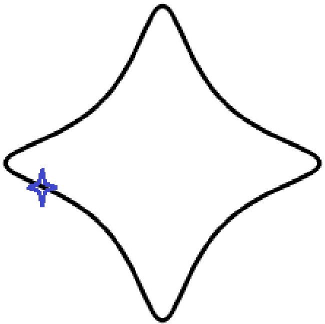


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

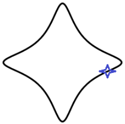


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$
(Edwards form)

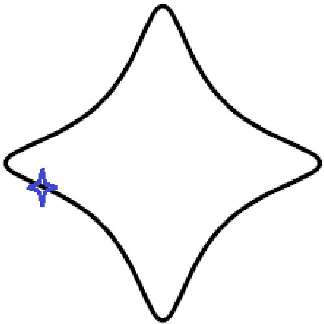


Elliptic curve on $\mathbb{F}_r$
(Edwards form)

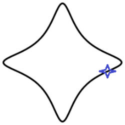


Correct BOS countermeasure

Elliptic curve on $\mathbb{Z}_{pr}$ (Edwards form)



Elliptic curve on $\mathbb{F}_r$ (Edwards form)



No problem with the point at infinity

Correct BOS countermeasure

Twisted Edwards curves are a generalization of Edwards curves.

Definition 5: Twisted Edwards curves

Let p a prime number. On the finite field $\mathbb{F}_p$, an elliptic curve in twisted Edwards form has parameters a, d in the finite field $\mathbb{F}_p$ and coordinates (x, y) satisfying the following equation:

$$ax^2 + y^2 = 1 + dx^2y^2, \tag{3}$$

with $ad(a - d) \neq 0$.

Like Edwards curves, the addition formulas are unified and complete.

Input : $P \in \mathcal{E}(\mathbb{F}_p)$, $k \in \mathbb{Z}$
Output : $Q = [k]P \in \mathcal{E}(\mathbb{F}_p)$

Offline phase

Edwards Curves:

Compute $\lambda p = x_G^2 + y_G^2 - c^2(1 + cx^2y^2)$
repeat
  Choose a random prime $r < p$
  Compute $x'_G = X_G \mod r$
  Compute $y'_G = y_G \mod r$
  Compute $c' = c^2 + \lambda p \mod r$
  Compute $d' = \frac{dc^2}{c^2+\lambda p} \mod r$
until $x'_G \neq 0$ *and* $y'_G \neq 0$ *and* $c'd'(1 - c'^4 d') \neq 0$ *and* c' *a square and* d' *a no-square*
▷ r verifies the lemma 1

Twisted Edwards Curves:

Compute $\lambda = (1 + dx_G^2 y_G^2 - ax_G^2 + y_G^2) \div p$
Find all the factor r smaller than p of λ
for *each factor* r **do**
  Compute $x'_G = x_G \mod r$
  Compute $y'_G = y_G \mod r$
  Compute $a' = a \mod r$
  Compute $d' = d \mod r$
  if $x'_G \neq 0$ *and* $y'_G \neq 0$ *and* $a'd'(a' - d') \neq 0$ *and* a' *a square and* d' *a no-square* **then**
    break ▷ r verifies the lemma 2
  else
    r does not work

Determine the small curve $\mathcal{E}(\mathbb{F}_r)$ with parameter c' (or a') and d', and a point $P'(x'_G, y'_G)$ is on that curve.
Determine the combined curve $\mathcal{E}(\mathbb{Z}_{pr})$ with parameter $C = CRT(c, c')$ (or $A = CRT(a, a')$) and $D = CRT(d, d')$ ▷ using properties 1 and 2.

Online phase

$(X_{pr} : Y_{pr} : Z_{pr}) = \mathrm{ECSM}(P, k, \mathcal{E}(\mathbb{Z}_{pr}))$ ▷ without test on the point and on the scalar value
$(X_r : Y_r : Z_r) = \mathrm{ECSM}(P', k, \mathcal{E}(\mathbb{F}_r))$ ▷ without test on the point and on the scalar value
if $(X_{pr} \mod r : Y_{pr} \mod r : Z_{pr} \mod r) = (X_r : Y_r : Z_r)$ **then**
  return $(X_{pr} \mod p : Y_{pr} \mod p : Z_{pr} \mod p)$
else
  return error

THALES

TELECOM ParisTech

Edwards Curve example

We generate a Edwards curve on the finite field $\mathbb{F}_{2^{255}-19}$ defined by $x^2 + y^2 = 1 - 6x^2y^2 \mod 2^{255} - 19$.
The number of elements defined on the curve computed by MAGMA tool is:

$$\#\mathcal{E}(2^{255} - 19) = 2^{255} + 138694172605265013181071149003381840660.$$

We find a generator point (x_G, y_G) on the Edwards curve with:

$$x_G = 53746514586250388770967951861766021561817370662802863797712166095360241234126,$$
$$y_G = 19570081233560550597987439135529516381390903225319934175948181057081969418594.$$

For the small curve $\mathcal{E}(\mathbb{F}_r)$, we can choose $r = 2147499037$; hence we have $c' = 1800340494, d' = 1430405543, x'_G = 28751952$ and $y'_G = 1290929995$.
Remark: The probability that a random prime r meets the requirement of lemma 1 is closed to $1/4$.

Twisted Edwards Curve example

The twisted Edwards Curves `Ed25519` defined by equation $-x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$ on $\mathbb{F}_{2^{255}-19}$, with:

$$x_G = 24727413235106541002554574571675588834622768167397638456726423682521233608206 3,$$

$$y_G = 15549675580280190176352668710449542251549572066445060580507079593062643049417.$$

The prime factor smaller than p of λ is :

Prime factors r	2	3	17	47	78857	843229	159962189299
Length in bit of r	2	2	5	7	16	19	40
r verifies the lemma 2	False	False	False	False	**True**	**True**	False

Important remark: we notice that the small verification field $\mathbb{F}_r$ cannot be chosen at random.

Performance

- Projective unified addition version takes $10\mathcal{M} + 1\mathcal{S} + 1\mathcal{C} + 1\mathcal{D} + 7\mathcal{A}$
- The bitwidth of the modulus is denoted by n (e.g., $n = 256$ for `Ed25519`).
- We denote by n' the number of CPU words of the modulus

Curves type	ECADD-complete on $\mathbb{F}_p$	ECADD-complete on $\mathbb{Z}_{pr}$	ECADD-complete on $\mathbb{F}_r$	Total cost of the countermeasure
Edwards	$11.8{n'}^2 + 7n'$	$11.8{n'}^2 + 30.6n' + 18.8$	19.8	$11.8{n'}^2 + 30.6n' + 38.6$
Twisted Edwards	$11.8{n'}^2 + 7n'$	$12.8{n'}^2 + 32.6n' + 29.8$	19.8	$12.8{n'}^2 + 32.6n' + 49.6$

Curves type	Computational overhead with:	
	$n' = 8$	$n' = 16$
Edwards	$\approx +28\%$	$\approx +13\%$
Twisted Edwards	$\approx +39\%$	$\approx +21\%$

Conclusion

- Using complete and unified elliptic curve formula is recommended to implemente the BOS countermeasure
- Choose a small curve is not trivial !
  (Other work: Neves and Tibouchi, PKC 2016)
- Another advantage of (Twisted) Edwards curve is the Simple Side Channel Analysis resistance of unified formulas (no difference between a doubling and adding)
- The ECSM computation on the small curve can be reduced by the modulo of the order of the small curve

Thank you !