Fault Attacks on Projective-to-Affine Coordinates Conversion

Diana Maimuț (ENS, France)
Cédric Murdica (Secure-IC/Télécom ParisTech, France)
David Naccache (ENS, France)
Mehdi Tibouchi (NTT Secure Platform, Japan)

Our contribution

Attack by Naccache, Smart and Stern at EUROCRYPT'04

Attack on Elliptic Curve Cryptosystems when the returned point of some signature schemes is given in projective coordinates (X, Y, Z).

Feasibility of the attack

In many systems, results are given in affine coordinates (x, y).

Our contribution

Attack by Naccache, Smart and Stern at EUROCRYPT'04

Attack on Elliptic Curve Cryptosystems when the returned point of some signature schemes is given in projective coordinates (X, Y, Z).

Feasibility of the attack

In many systems, results are given in affine coordinates (x, y).

Our fault attack model

Injecting an error during the conversion process to recover the missing Z coordinate.
We propose 3 different ways to recover the missing Z coordinate depending on the fault's precision.

Table of Contents

Table of Contents

Table of Contents

Elliptic Curve

Elliptic Curve on affine coordinates

On a field $\mathbb{F}_p$, $p > 3$, an elliptic curve E is the set of points $(x, y) \in \mathbb{F}_p$, satisfying

$$y^2 = x^3 + ax + b, \text{ with } 4a^3 + 27b^2 \neq 0$$

plus the point at infinity $\mathcal{O}$.
Costly formulæ because of inversions.

Elliptic Curve on Jacobian coordinates

To prevent costly division, represent the point (x, y) by (xZ^2, yZ^3, Z) for any non-zero Z. The curve equation is

$$Y^2 = X^3 + aXZ^4 + bZ^6$$

with $\mathcal{O} = (1, 1, 0)$ and the equivalence relation $(X, Y, Z) \sim (\lambda^2 X, \lambda^3 Y, \lambda Z)$.
To retrieve the affine coordinates from (X, Y, Z), compute
$(x, y) := (x, y, 1) = (X/Z^2, Y/Z^3, 1)$.

Returning the result in Jacobian coordinates

Computation of $Q = [k]P$

Operation called Elliptic Curve Scalar Multiplication (ECSM)

- with k private
- with P public

Is it secure to return the value $Q = (X, Y, Z)$ in Jacobian coordinates?

Returning the result in Jacobian coordinates

Computation of $Q = [k]P$

Operation called Elliptic Curve Scalar Multiplication (ECSM)

- with k private
- with P public

Is it secure to return the value $Q = (X, Y, Z)$ in Jacobian coordinates?

No

"Projective coordinates leak" at Eurocrypt 2004 by Naccache, Smart, Stern.
Some bits of k can be retrieved.

Table of Contents

Group law in Jacobian coordinates

$$P_1 = (X_1, Y_1, Z_1) = (x_1 Z_1^2, y_1 Z_1^3, Z_1), P_2 = (X_2, Y_2, 1) = (x_2, y_2, 1)$$

$$\text{Algorithm } \textsc{ECDBL} = \begin{cases} S &= 4X_1Y_1^2 \\ M &= 3X_1^2 + aZ_1^4 \\ X_3 &= -2S + M^2 \\ Y_3 &= -8Y_1^4 + M(S - X_3) \\ Z_3 &= 2Y_1Z_1 = 2y_1Z_1^4 \\ P_3 &= (X_3, Y_3, Z_3) \end{cases} \qquad \text{return}(P_3 = 2P_1)$$

$$\text{Algorithm } \textsc{ECADD} = \begin{cases} H &= x_2Z_1^2 - X_1 \\ R &= y_2Z_1^3 - Y_1 \\ X_3 &= -H^3 - 2UH^2 + R^2 \\ Y_3 &= -SH^3 + R(UH^2 - X_3) \\ Z_3 &= Z_1H = Z_1^3(x_2 - x_1) \\ P_3 &= (X_3, Y_3, Z_3) \end{cases} \qquad \text{return}(P_3 = P_1 + P_2)$$

Description of the attack

Output result in Jacobian coordinates

$[k]P = (X_0, Y_0, Z_0)$ is computed using the Double-and-Add method.

$A \leftarrow P$
for $i = N - 2$ downto 0 **do**
  $A \leftarrow \text{ECDBL}(A)$
  if $k_i = 1$ then $A \leftarrow \text{ECADD}(A, P)$
end for
return $A = [k]P = (X_0, Y_0, Z_0)$

If $k_0 = 0$

The last operation to obtain (X_0, Y_0, Z_0) was a doubling. Is this possible?

If $k_0 = 1$

The last operations to obtain (X_0, Y_0, Z_0) was a doubling followed by an addition. Is this possible?

Description of the attack

Notation: (X_1, Y_1, Z_1) are the coordinates of the point A at the end of iteration 1

If $k_0 = 0$

The last operation to obtain $Q = (X_0, Y_0, Z_0)$ was a doubling.

$$Z_0 = 2Y_1Z_1 = 2y_1Z_1^4 \Rightarrow Z_1^4 = \frac{Z_0}{2y_1}$$

- Z_0 is given in the output
- Halve the point $Q \Rightarrow (x_1, y_1) = [2^{-1} \bmod \#E]Q$
- Only Z_1 is unknown

Result

- If $\frac{Z_0}{2y_1}$ is not a fourth root, then $k_0 = 1$
- If $\frac{Z_0}{2y_1}$ is a fourth root, then compute the "possible" (X_1, Y_1, Z_1) points

Description of the attack

In an analogous manner, if $k_0 = 1$, the last operation was an addition.
Addition involves a cube for the Z coordinates $\Rightarrow$ try a cubic root.

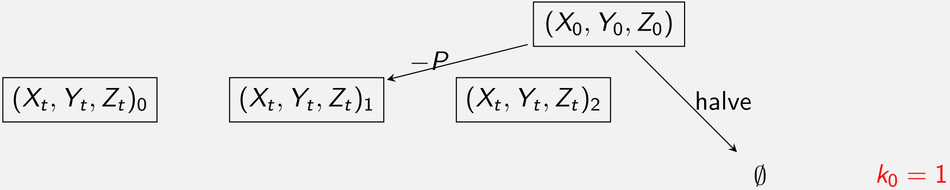
Backtracking Algorithm

$$\boxed{(X_0, Y_0, Z_0)}$$

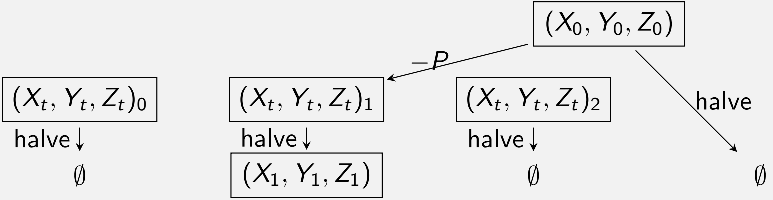
Backtracking Algorithm

$$\boxed{(X_0, Y_0, Z_0)} \xrightarrow{\text{halve}} \emptyset \qquad k_0 = 1$$

Backtracking Algorithm

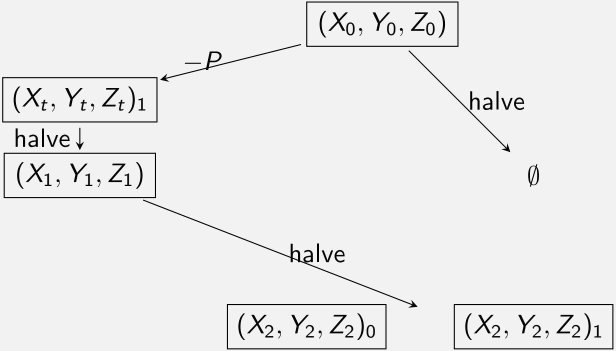


Backtracking Algorithm



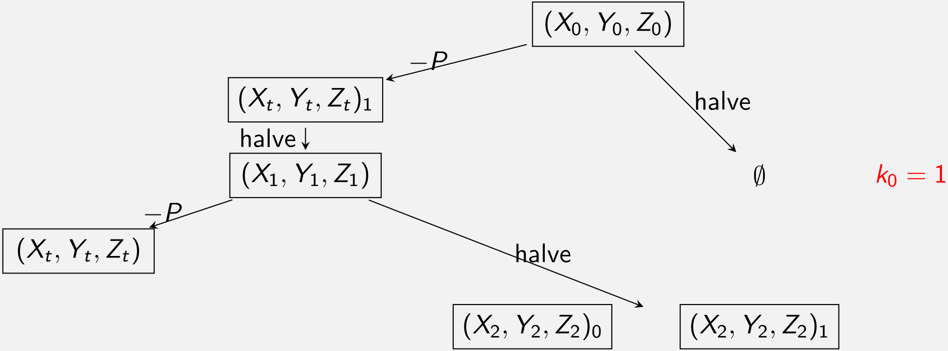
$k_0 = 1$

Backtracking Algorithm

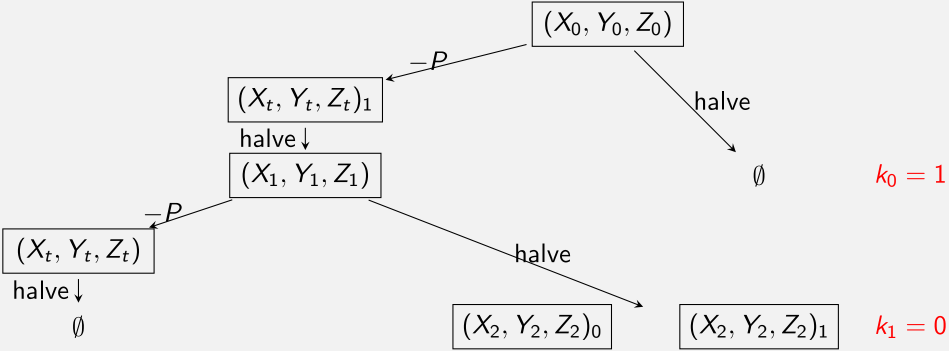


$k_0 = 1$

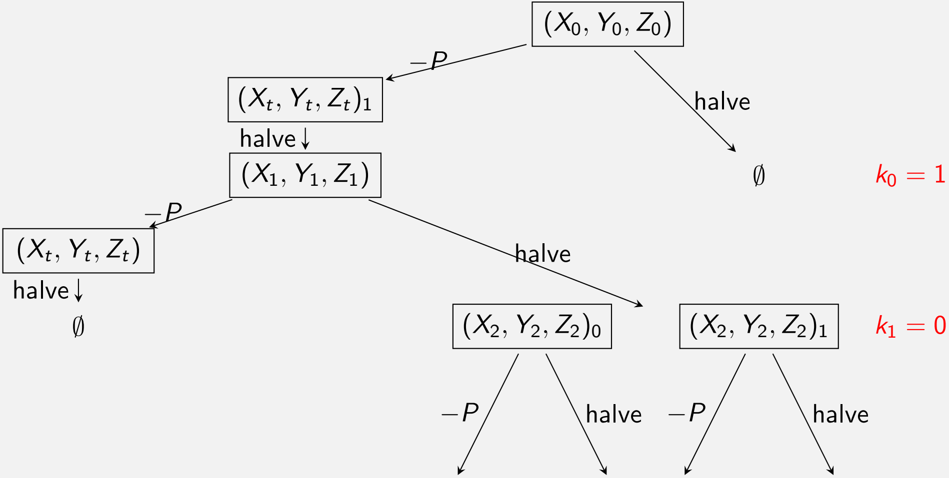
Backtracking Algorithm



Backtracking Algorithm



Backtracking Algorithm



Synthesis of Naccache et al.' attack

- The attack cannot permit to recover all bits of the scalar, only a few. This is enough for some protocols.
- The result must be in Jacobian coordinates (X, Y, Z). In schemes, the results are in affine coordinates (x, y). $[k]P$ is computed in Jacobian coordinates and the point is converted in affine coordinates before returning it.

Synthesis of Naccache et al.' attack

- The attack cannot permit to recover all bits of the scalar, only a few. This is enough for some protocols.
- The result must be in Jacobian coordinates (X, Y, Z). In schemes, the results are in affine coordinates (x, y). $[k]P$ is computed in Jacobian coordinates and the point is converted in affine coordinates before returning it.

Our contribution

Inject a fault during the conversion procedure, so that the faulty result in affine coordinates contains some information on the missing coordinate Z.

Table of Contents

Fault on conversion procedure

Conversion Procedure

The following procedure converts $P = (X, Y, Z) = (xZ^2, yZ^3, Z)$ from Jacobian to affine coordinates (x, y).

$$\textsc{convert}(X, Y, Z) = \begin{cases} r & \leftarrow & Z^{-1} \\ s & \leftarrow & r^2 \\ x & \leftarrow & X \cdot s \\ t & \leftarrow & Y \cdot s \\ y & \leftarrow & t \cdot r \quad \text{return}(x, y) \end{cases}$$

Fault on conversion procedure

Conversion Procedure

The following procedure converts $P = (X, Y, Z) = (xZ^2, yZ^3, Z)$ from Jacobian to affine coordinates (x, y).

$$\textsc{convert}(X, Y, Z) = \begin{cases} r & \leftarrow & Z^{-1} \\ s & \leftarrow & r^2 \\ \tilde{s} = s + \varepsilon & \hookleftarrow & \text{corruption of s} \\ \tilde{x} & \leftarrow & X \cdot \tilde{s} \\ \tilde{t} & \leftarrow & Y \cdot \tilde{s} \\ \tilde{y} & \leftarrow & \tilde{t} \cdot r & & \text{return}(\tilde{x}, \tilde{y}) \end{cases}$$

Equations system

$$\tilde{x} = x + xZ^2\varepsilon \mod p$$
$$\tilde{y} = y + yZ^2\varepsilon \mod p$$

Table of Contents

Large Unknown Faults and a correct result

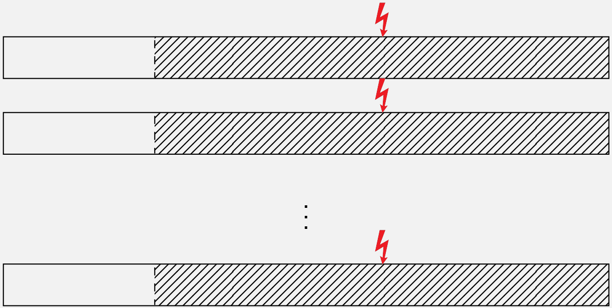
$$\boxed{\qquad\qquad\qquad\qquad\qquad\qquad} = s \quad = Z^{-2}$$

$$\boxed{\qquad\qquad\qquad\qquad\qquad\qquad} = s \quad = Z^{-2}$$

$$\vdots$$

$$\boxed{\qquad\qquad\qquad\qquad\qquad\qquad} = s \quad = Z^{-2}$$

Large Unknown Faults and a correct result



$= \quad \tilde{s}_1 = Z^{-2} + \varepsilon_1$

$= \quad \tilde{s}_2 = Z^{-2} + \varepsilon_2$

$\vdots$

$= \quad \tilde{s}_n = Z^{-2} + \varepsilon_n$

Equations system

Unknown values in red

$$\tilde{x}_i = x + x Z^2 \varepsilon_i \Rightarrow \frac{\tilde{x}_i}{x} - 1 = Z^2 \varepsilon_i \mod p \text{ with } \varepsilon_i < p^a \text{ for some } a < 1$$

Large Unknown Faults and a correct result

Equations system with a known result (x, y)

Unknown values in red

$$\tilde{x}_i = x + xZ^2\varepsilon_i \Rightarrow \frac{\tilde{x}_i}{x} - 1 = Z^2\varepsilon_i \mod p \text{ with } \varepsilon_i < p^a \text{ for some } a < 1$$

$$\Rightarrow u_i = Z^2\varepsilon_i \mod p \text{ with } u_i = \frac{\tilde{x}_i}{x} - 1$$

$$\Rightarrow \boldsymbol{\varepsilon} = \boldsymbol{s} \cdot \mathbf{u} \bmod p \text{ with } \boldsymbol{s} = Z^{-2}, \mathbf{u} = (u_1, \ldots, u_n), \boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$$

Recover $\boldsymbol{\varepsilon}$ using LLL

- Let L be the lattice generated by the vector $\mathbf{u}$ and $p\mathbb{Z}^n$ in $\mathbb{Z}^n$
- Since $\boldsymbol{\varepsilon}$ satisfies $\boldsymbol{\varepsilon} = \boldsymbol{s} \cdot \mathbf{u} \bmod p$, $\boldsymbol{\varepsilon}$ is a vector in L, with $\varepsilon_i < p^a$
- Then, we can recover $\boldsymbol{\varepsilon}$ directly by reducing L using LLL since $\boldsymbol{\varepsilon}$ is a small vector of the lattice.

Large Unknown Faults and a correct result

Equations system with a known result (x, y)

Unknown values in red

$$\tilde{x}_i = x + xZ^2\varepsilon_i \Rightarrow \frac{\tilde{x}_i}{x} - 1 = Z^2\varepsilon_i \mod p \text{ with } \varepsilon_i < p^a \text{ for some } a < 1$$

$$\Rightarrow u_i = Z^2\varepsilon_i \mod p \text{ with } u_i = \frac{\tilde{x}_i}{x} - 1$$

$$\Rightarrow \boldsymbol{\varepsilon} = s \cdot \mathbf{u} \bmod p \text{ with } s = Z^{-2}, \mathbf{u} = (u_1, \ldots, u_n), \boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$$

Recover $\boldsymbol{\varepsilon}$ using LLL

- Let L be the lattice generated by the vector $\mathbf{u}$ and $p\mathbb{Z}^n$ in $\mathbb{Z}^n$
- Since $\boldsymbol{\varepsilon}$ satisfies $\boldsymbol{\varepsilon} = s \cdot \mathbf{u} \bmod p$, $\boldsymbol{\varepsilon}$ is a vector in L, with $\varepsilon_i < p^a$
- Then, we can recover $\boldsymbol{\varepsilon}$ directly by reducing L using LLL since $\boldsymbol{\varepsilon}$ is a small vector of the lattice.
- Simulation (SAGE): with $p \approx 2^{256}$ and $\varepsilon_i \approx 2^{224}$, only 9 faults are necessary to recover $\boldsymbol{\varepsilon}$, in 3ms.

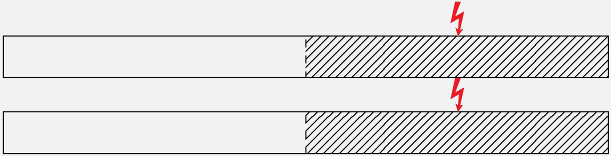
Table of Contents

Two Faults and a correct result

$$\boxed{} = s \quad = Z^{-2}$$

$$\boxed{} = s \quad = Z^{-2}$$

Two Faults and a correct result



$$= \quad \tilde{s}_1 = Z^{-2} + \varepsilon_1$$

$$= \quad \tilde{s}_2 = Z^{-2} + \varepsilon_2$$

Equations system

Unknown values in red

$$\frac{\tilde{x}_1}{x} - 1 = u_1 = Z^2\varepsilon_1 \bmod p \text{ with } \varepsilon_1 < p^{1/2}$$
$$\frac{\tilde{x}_2}{x} - 1 = u_2 = Z^2\varepsilon_2 \bmod p \text{ with } \varepsilon_2 < p^{1/2}$$

Two Faults and a correct result

Equations system

Unknown values in red

$$\frac{\tilde{x}_1}{x} - 1 = u_1 = Z^2 \varepsilon_1 \bmod p \text{ with } \varepsilon_1 < p^{1/2}$$
$$\frac{\tilde{x}_2}{x} - 1 = u_2 = Z^2 \varepsilon_2 \bmod p \text{ with } \varepsilon_2 < p^{1/2}$$

Let $\alpha = u_1 / u_2 = \varepsilon_1 \varepsilon_2^{-1}$

Two Faults and a correct result

Equations system

Unknown values in red

$$\frac{\tilde{x_1}}{x} - 1 = u_1 = Z^2\varepsilon_1 \bmod p \text{ with } \varepsilon_1 < p^{1/2}$$
$$\frac{\tilde{x_2}}{x} - 1 = u_2 = Z^2\varepsilon_2 \bmod p \text{ with } \varepsilon_2 < p^{1/2}$$

Let $\alpha = u_1/u_2 = \varepsilon_1\varepsilon_2^{-1} \Rightarrow$ problem known as the *Rational Number Reconstruction* and is solved using Gauß' algorithm for finding the shortest vector in a bidimensional lattice.

Theorem

Let $\varepsilon_1, \varepsilon_2 \in \mathbb{Z}$ such that $-A \leq \varepsilon_1 \leq A$ and $0 < \varepsilon_2 \leq B$. Let $p > 2AB$ be a prime and $\alpha = \varepsilon_1\varepsilon_2^{-1} \bmod p$. Then $\varepsilon_1, \varepsilon_2$ can be recovered from A, B, α, p in polynomial time.

Recover $\varepsilon_1, \varepsilon_2$ with $A = B = \lfloor\sqrt{p}\rfloor$, $2AB < p$, $0 \leq \varepsilon_1 \leq A$ and $0 \leq \varepsilon_2 \leq B$.

Table of Contents

Known Fault

$$= s \quad = Z^{-2}$$

Known Fault

$$= \quad \tilde{s} = Z^{-2} + \varepsilon$$

Equation

$$\tilde{x} = x + x{\color{red}Z}^2\varepsilon \text{ with } \varepsilon \text{ known}$$

The knowledge of x suffices to recover Z.

Known Fault on ECDSA

G a public generator of order n.
Key pair of an entity (d, P) with $P = [d]G$.

Algorithm 1 ECDSA Signature

Input: Private key d, message m
Output: Signature (r, s)

$k \xleftarrow{\mathcal{R}} \{1, \ldots, n-1\}$
$Q \leftarrow [k]G$
$r \leftarrow x_Q \mod n$
$i \leftarrow k^{-1} \mod n$
$s \leftarrow i(dr + m) \mod n$
return (r, s)

Algorithm 2 ECDSA Verification

Input: Public key P, m, signature (r, s)
Output: true or false

$w \leftarrow s^{-1} \mod n$
$u_1 \leftarrow w \cdot m \mod n$
$u_2 \leftarrow w \cdot r \mod n$
$Q \leftarrow [u_1]G + [u_2]P$
$v \leftarrow x_Q \mod n$
return $(v \stackrel{?}{=} r)$

Known Fault on ECDSA

Algorithm 3 Wrong ECDSA Signature

Input: Private key d, message m
Output: Signature (r, s)

$k \xleftarrow{\mathcal{R}} \{1, \ldots, n-1\}$
$(\tilde{x_Q}, \tilde{y_Q}) \leftarrow [k]G$ ← fault during conversion of Q
$\tilde{r} \leftarrow \tilde{x_Q} \bmod n$
$i \leftarrow k^{-1} \bmod n$
$\tilde{s} \leftarrow i(d\tilde{r} + m) \bmod n$
return $(\tilde{r}, \tilde{s})$

Known Fault on ECDSA

Algorithm 5 Wrong ECDSA Signature

Input: Private key d, message m
Output: Signature (r, s)

$k \xleftarrow{\mathcal{R}} \{1, \ldots, n-1\}$
$(\tilde{x_Q}, \tilde{y_Q}) \leftarrow [k]G$ ← fault during conversion of Q
$\tilde{r} \leftarrow \tilde{x_Q} \bmod n$
$i \leftarrow k^{-1} \bmod n$
$\tilde{s} \leftarrow i(d\tilde{r} + m) \bmod n$
return $(\tilde{r}, \tilde{s})$

Algorithm 6 Recover the x coordinate of Q

Input: P, m, wrong signature $(\tilde{r}, \tilde{s})$
Output: Q

$\tilde{w} \leftarrow \tilde{s}^{-1} \bmod n$
$\tilde{u_1} \leftarrow \tilde{w} \cdot m \bmod n$
$\tilde{u_2} \leftarrow \tilde{w} \cdot \tilde{r} \bmod n$
$\tilde{Q} \leftarrow [\tilde{u_1}]G + [\tilde{u_2}]P$
$\tilde{Q} = \left[\frac{km}{d\tilde{r}+m}\right] G + \left[\frac{k\tilde{r}}{d\tilde{r}+m}\right] P$
$\tilde{Q} = [k]G = Q$
return Q

Recover the true x coordinate of Q

From $(\tilde{r}, \tilde{s})$, we can recover the correct value of x_Q
$\Rightarrow$ recover the Z coordinate of Q $\Rightarrow$ grab a few bits of k

Table of Contents

Conclusion

Contribution

- A new kind of fault attack at the *end* of the ECSM
- The attack permits to perform the Naccache et al.'s attack even when the result is returned in affine coordinates

Feasibility of the attack

- Practical attacks on particular elliptic curve schemes (large and two faults)
- Theoretical attack on ECDSA. Theoretical because the fault model is too strong.

Prevention

Check the validity of the result *after* conversion to affine coordinates.

Thanks for your attention.

Thanks for your attention.

Questions?

Fault Attacks on Projective-to-Affine Coordinates Conversion

Diana Maimuţ (ENS, France)
Cédric Murdica (Secure-IC/Télécom ParisTech, France)
David Naccache (ENS, France)
Mehdi Tibouchi (NTT Secure Platform, Japan)