Sign Change Fault Attacks On Elliptic Curve Cryptosystems

Martin Otto

Codes and Cryptography Group / PaSCo graduate school

University of Paderborn, Paderborn, Germany

joint work with Johannes Blömer[1] and Jean-Pierre Seifert[2]

[1] University of Paderborn, Germany [2] Intel Corporation, Hillsboro (OR), USA

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Content

Fault Model

- Sign Change Faults

Attack

- on Elliptic Curve Scalar Multiplication

Countermeasure

- against Sign Change Attacks

Elliptic Curves: Notation

- Set of points $(x : y : z)$ satisfying

$$\Rightarrow E_p : y^2 z \equiv x^3 + Axz^2 + Bz^3 \bmod p \qquad (1)$$

(Weierstraß-Equation in projective coordinates)

- We only consider E defined over $\mathbb{F}_p$, p prime
- Group of points: All points satisfying (1)
  - $(x : y : z) = (\lambda x : \lambda y : \lambda z)$ for $\lambda \neq 0$
  - $\mathcal{O} = (0 : 1 : 0) \longrightarrow$ “point at infinity“

Previous Work

Approaches:

- Analysis of faults in curve and field parameters (Biehl, Meyer, Müller 2000 & Ciet, Joye 2003)

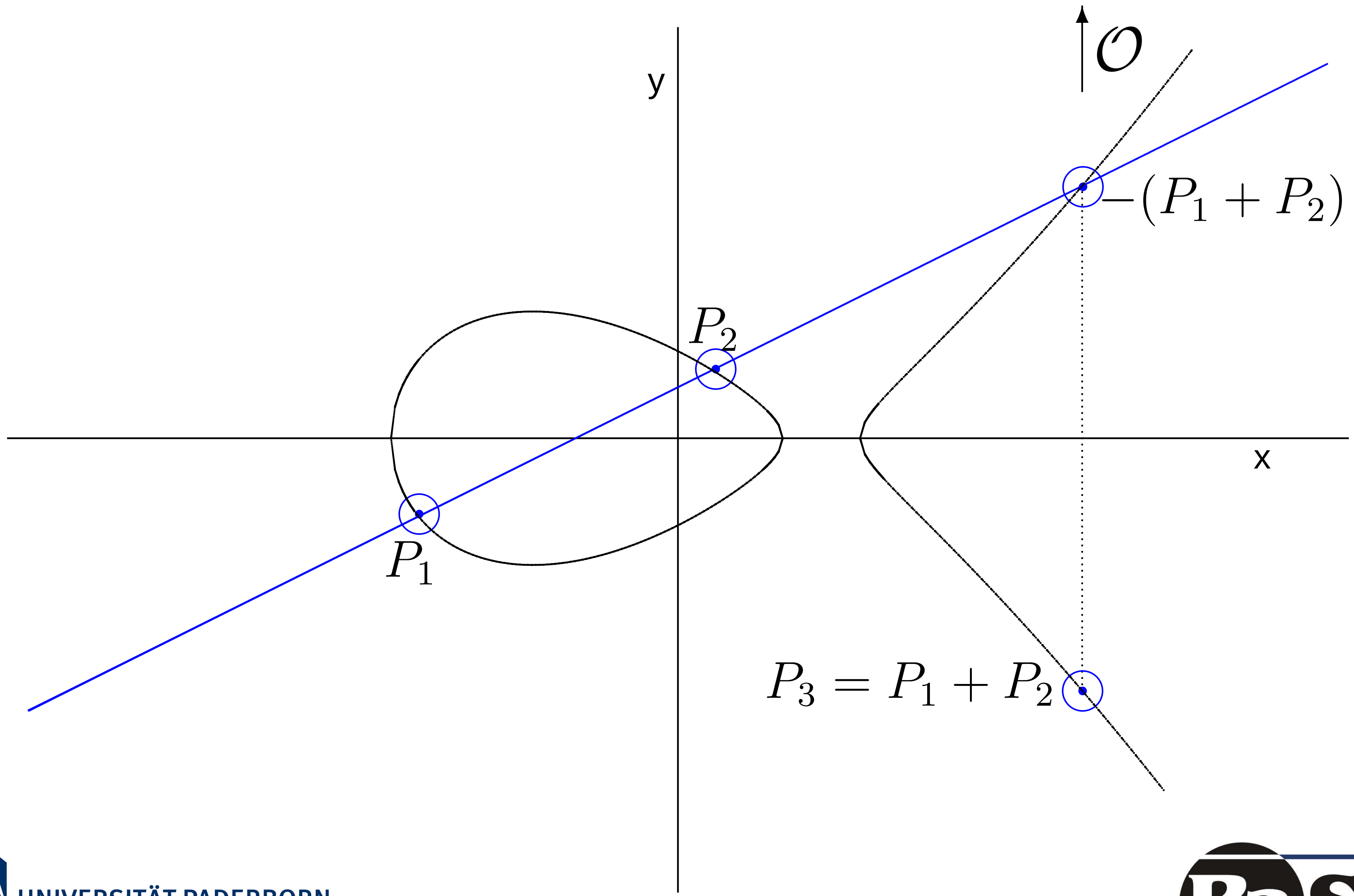
Results:

- With overwhelming probability, a (random) fault results in a final result that is not on the curve.

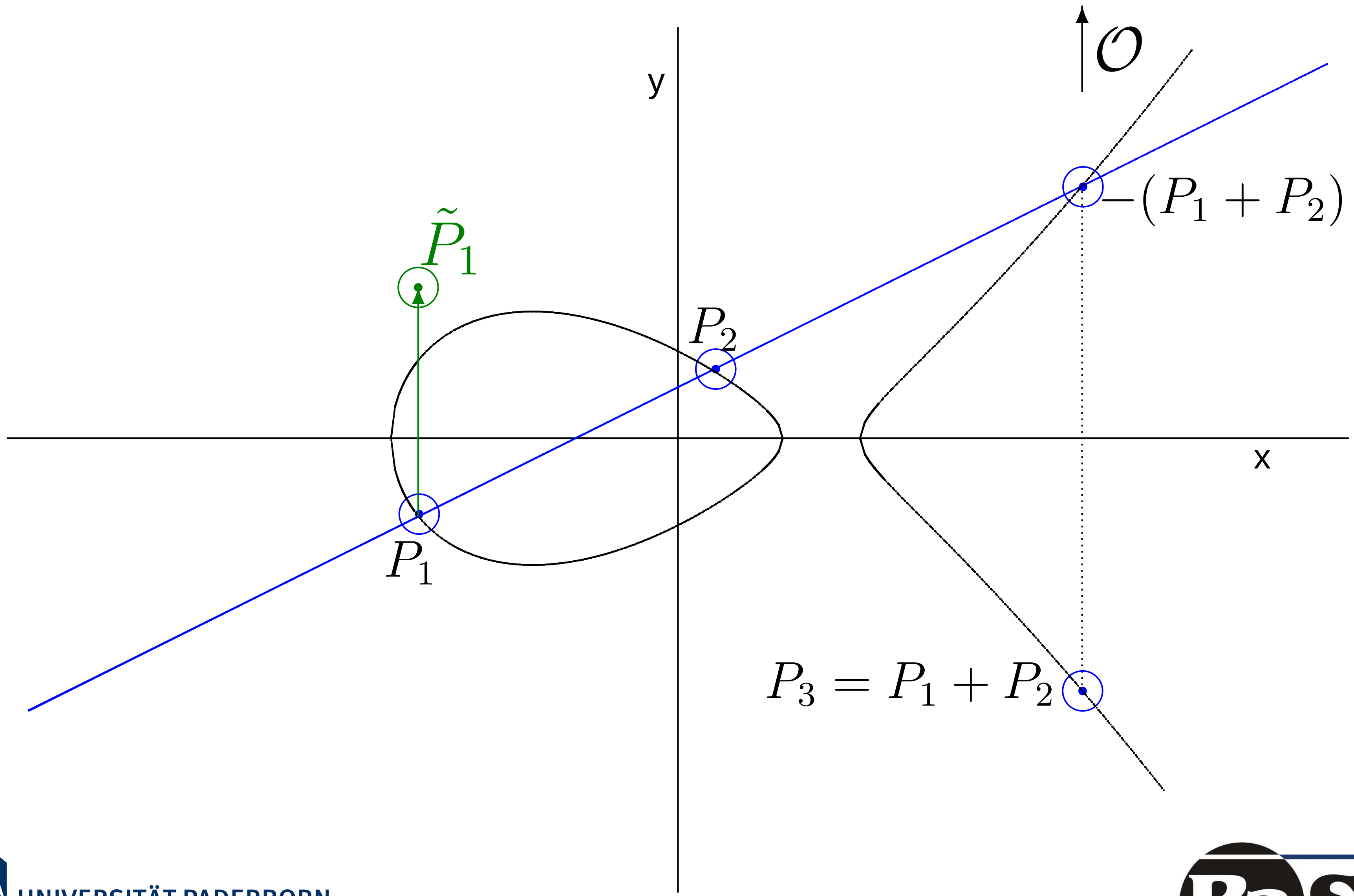
Natural Countermeasure:

- Check result before output: Is result a valid point on curve?

Fault Attacks on Elliptic Curve Addition



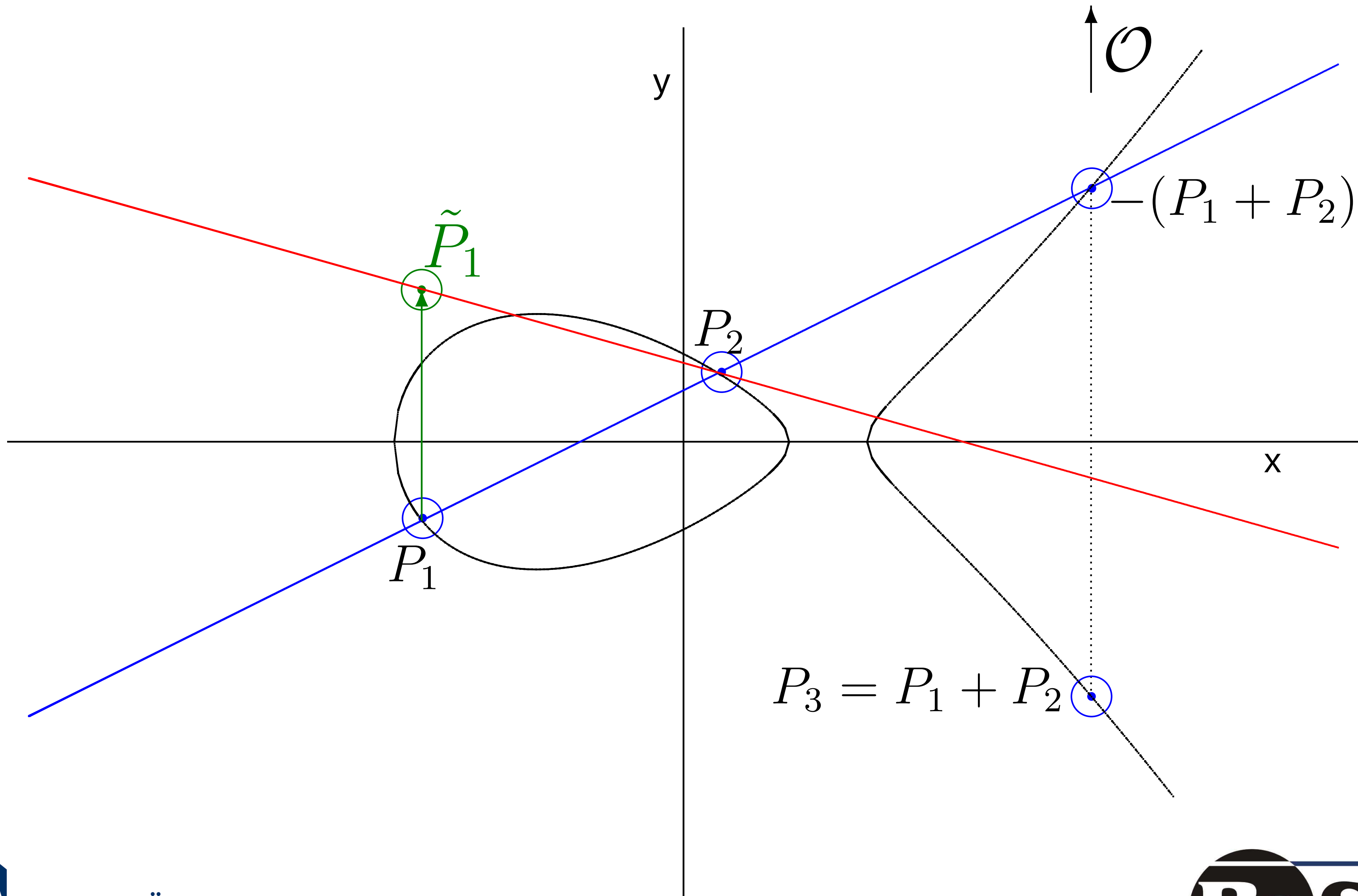
Fault Attacks on Elliptic Curve Addition



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

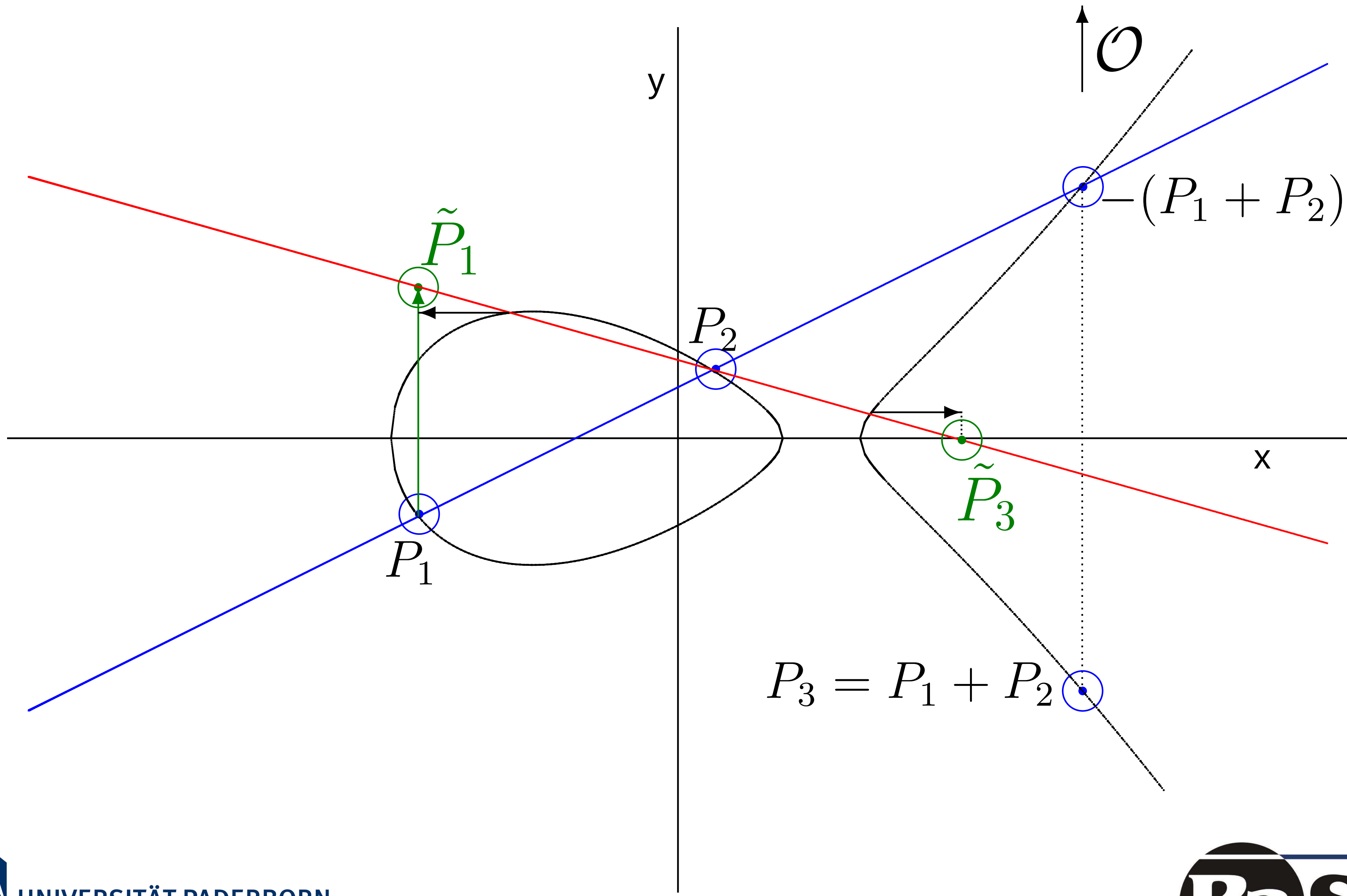
Fault Attacks on Elliptic Curve Addition



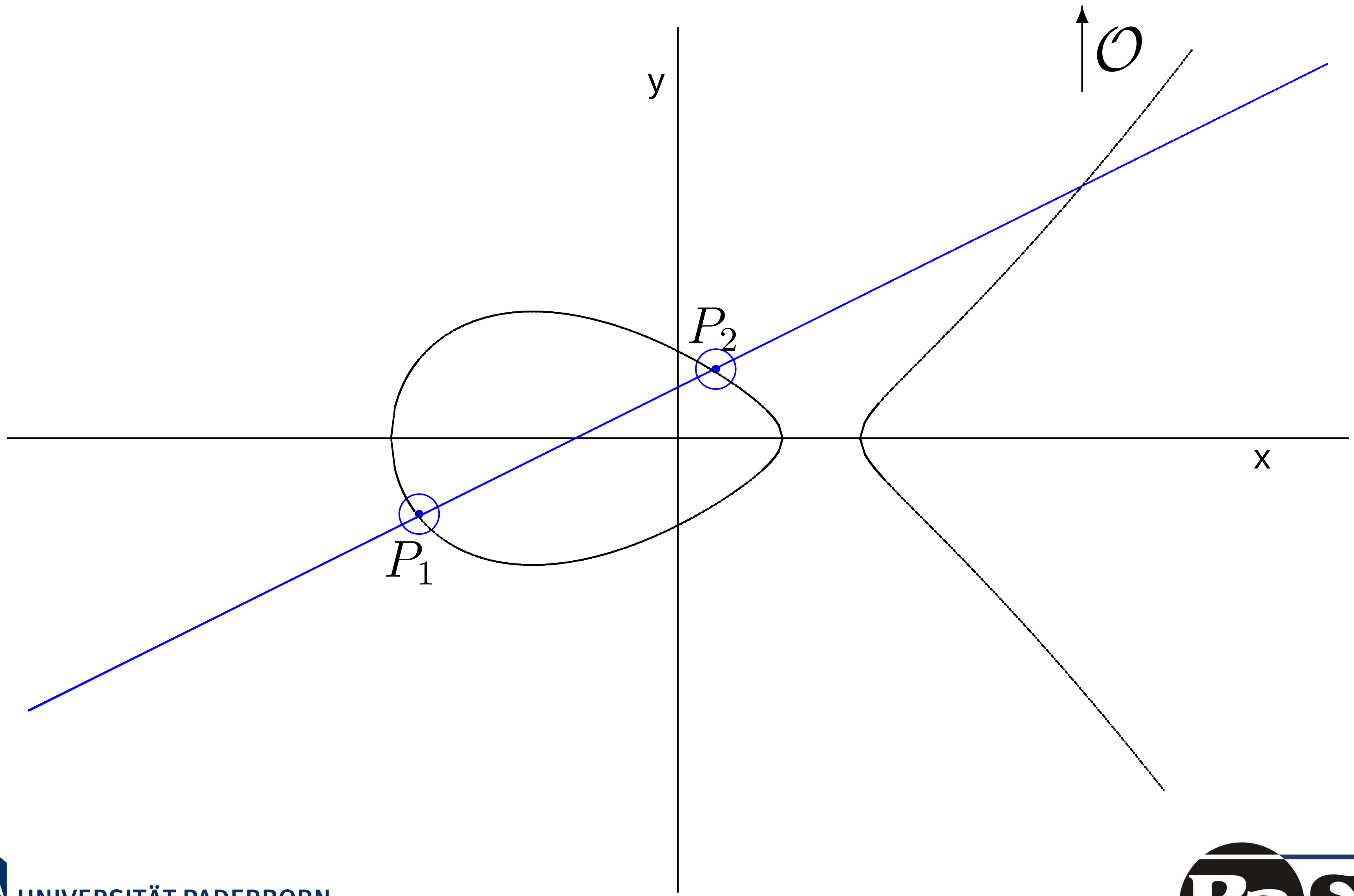
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

GK PaSCo
Paderborn Institute for Scientific Computation

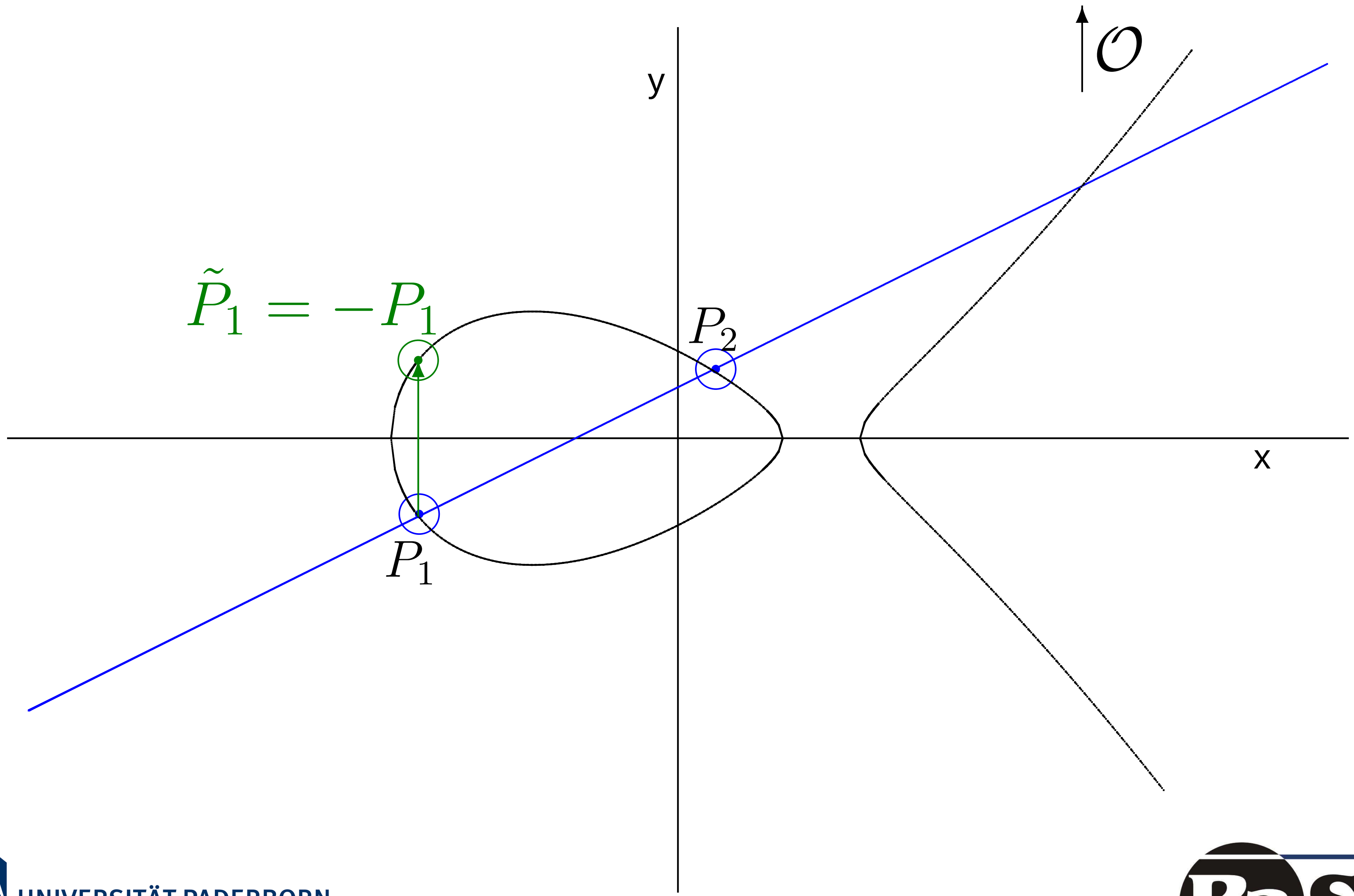
Fault Attacks on Elliptic Curve Addition



Sign Change Faults



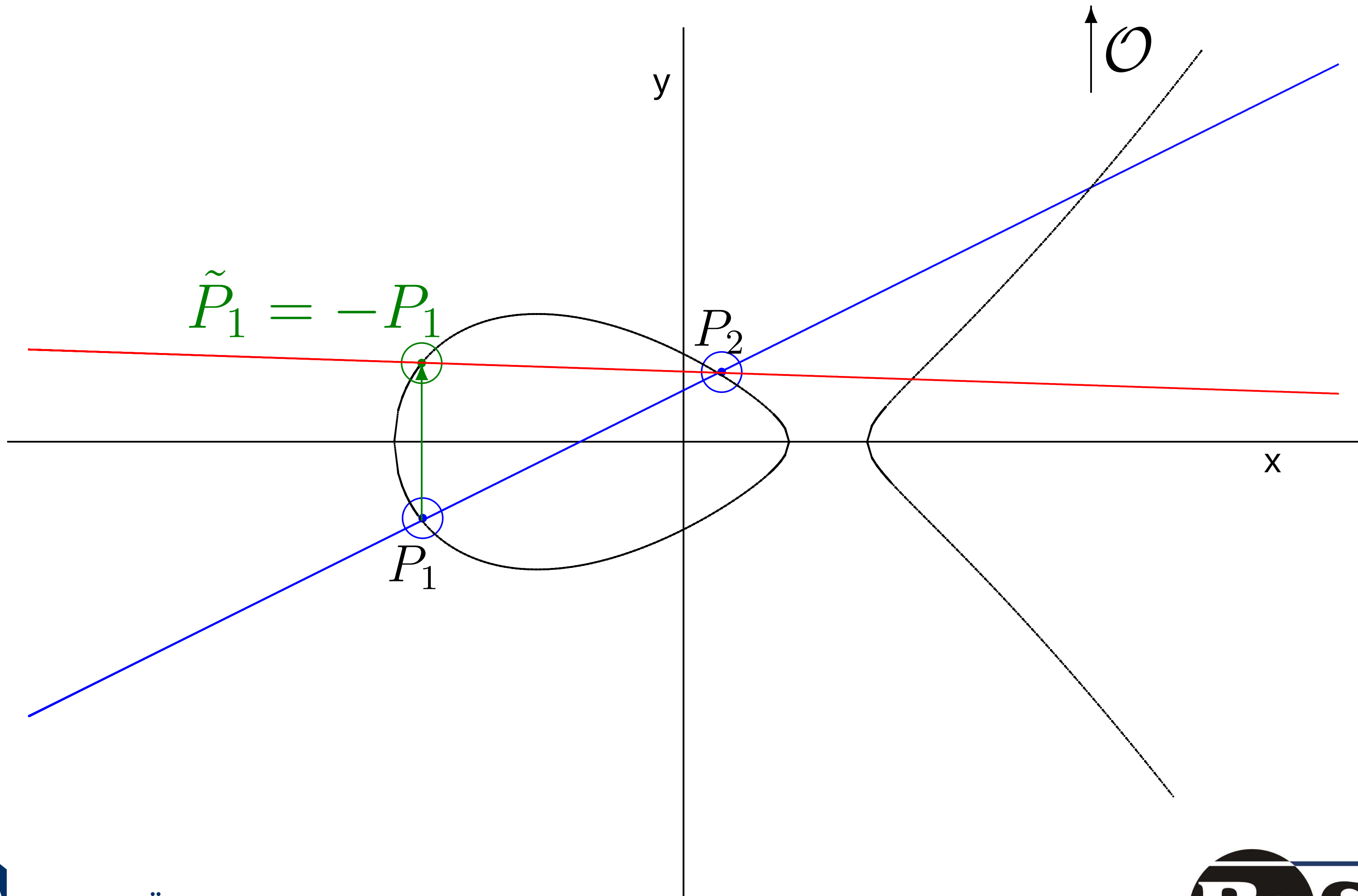
Sign Change Faults



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

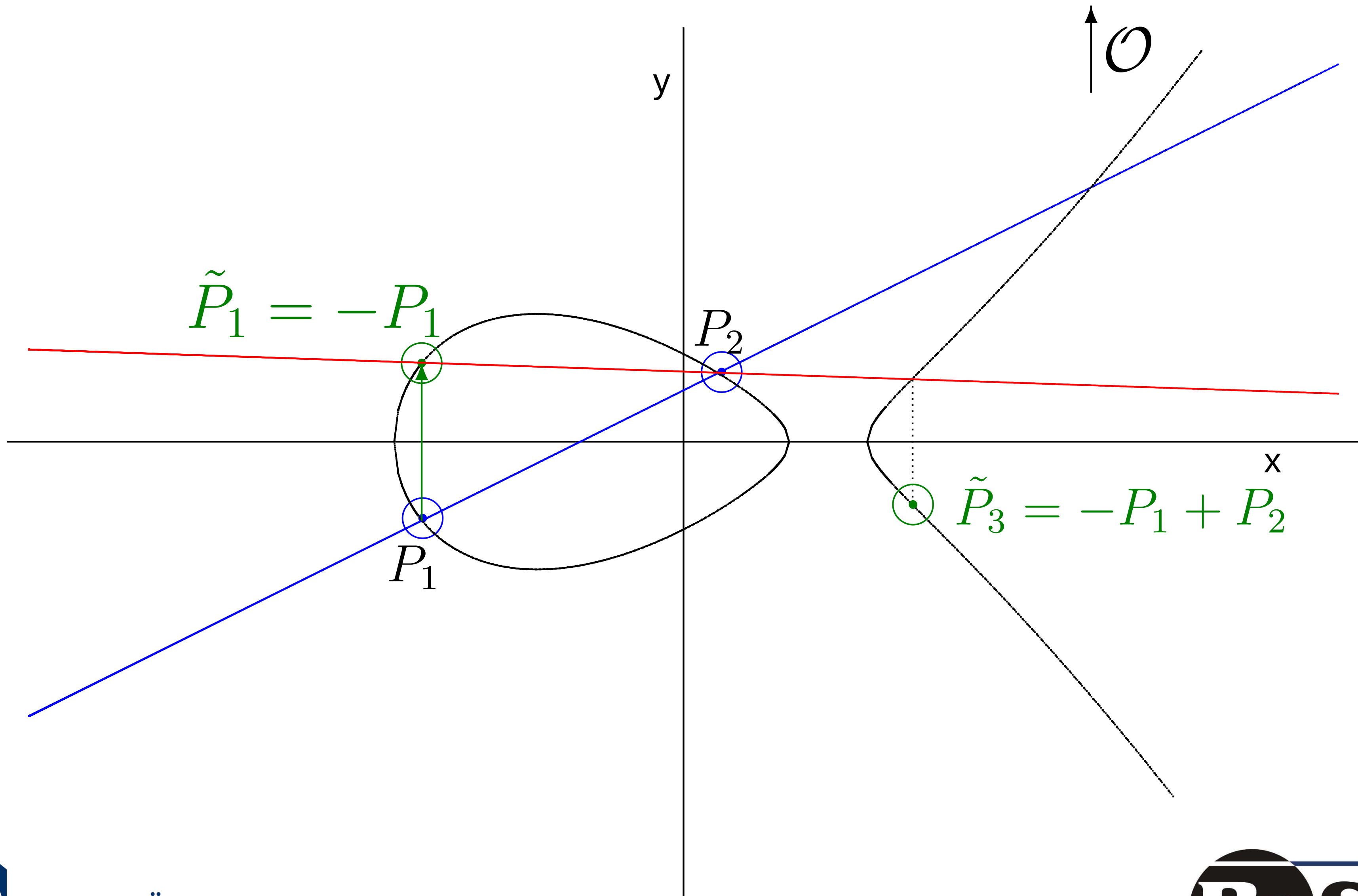
Sign Change Faults



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Sign Change Faults



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Can we achieve Sign Change Faults?

Yes, we can! Examples:

- NAF-based scalar multiplication: attack a key bit
- Attacks during point addition, e.g., affine addition:

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2}, \qquad \begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= -y_1 + \lambda \cdot (x_1 - x_3) \end{aligned}$$

  - Change y_2 to achieve a Sign Change Fault
  - Attack ALU s.t. argument is inverted
- Success depends on the implementation (hardware and software)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Fault Attacks With Sign Change Faults (1)

Compute $Q = kP$ on E_p:		
0 Set $n := l(k)$		
1 Set $Q_n := \mathcal{O}$		
2 For i from n-1 to 0 do		
3 Set $Q_i' := 2 \cdot Q_{i+1}$		
4 If (k_i = 1) then	set $Q_i := Q_i' + P$	
else	set $Q_i := Q_i'$	
5 Return Q_0		

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Fault Attacks With Sign Change Faults (1)

Compute $Q = kP$ on E_p:

```
Set n := l(k)
Set Q_n := O
For i from n-1 to 0 do
    Set Q'_i := 2 · Q_{i+1}
    If (k_i = 1) then  set Q_i := Q'_i + P
                 else  set Q_i := Q'_i
Return Q_0
```

Attacking

$$Q'_i \overset{⚡}{\longmapsto} -Q'_i$$

at random iteration i

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Fault Attacks With Sign Change Faults (1)

Compute $Q = kP$ on E_p:
0 Set $n := l(k)$
1 Set $Q_n := \mathcal{O}$
2 For i from n-1 to 0 do
3 Set $Q'_i := 2 \cdot Q_{i+1}$
4 If (k_i = 1) then set $Q_i := Q'_i + P$
else set $Q_i := Q'_i$
5 Return Q_0

Attacking

$Q'_i \overset{\not\;}{\longmapsto} -Q'_i$

at random iteration i

Step 1: Describe faulty final result

$\tilde{Q} = -Q + 2 \cdot L_i(k)$, where $L_i(k) := \sum_{j=0}^{i} k_j 2^j \cdot P$

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Fault Attacks With Sign Change Faults (2)

Step 2: Collect many faulty final results

- Choose block size $m \Rightarrow O(2^m)$ operations:
- Mount $(n/m)\log(2n)$ many attacks to hit every possible block with Prob. at least $1/2$

Fault Attacks With Sign Change Faults (2)

Step 2: Collect many faulty final results

- Choose block size $m \Rightarrow O(2^m)$ operations:
- Mount $(n/m)\log(2n)$ many attacks to hit every possible block with Prob. at least $1/2$

Step 3: incremental computation of k

- Assumption: all s lowest bits of k are known
- try all possibilities with up to $s+m$ bits:

$$\tilde{Q} \stackrel{?}{=} -Q + 2 \cdot L_{s+m-1}(k)$$

- Compare to gathered faulty final results

Fault Attacks With Sign Change Faults (3)

Step 4: Proof of correctness

- guessed pattern describes a faulty final result $\tilde{Q}$: we show: pattern correct
- if no pattern describes $\tilde{Q}$:
  - “Zero Block Failure“: k has block of zeros

Summary of Attack

Theorem: Secret scalar k of length n is recovered with

$$O(n \cdot 2^m \cdot t)$$

scalar multiplications with probability at least $1/2$ inducing $t = (n/m)\log(2n)$ Sign Change Faults.

- This attack also applies to other scalar multiplication algorithms (NAF-LR/RL, Montgomery Ladder)
- Attack layout derived from first fault attack on RSA (Boneh, DeMillo, Lipton 1997)

Countermeasure Against Sign Change Attacks

What we want

- check final result efficiently for correctness

Idea

- Check using a “small“ curve:
  Choose prime t and (A_t, x_t, y_t) to define

$$E_t : y^2 z \equiv x^3 + A_t x z^2 + B_t z^3 \bmod t$$
$$P_t = (x_t : y_t : 1)$$

  such that order of E_t is prime

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

GK PaSCo Paderborn Institute for Scientific Computation

A “combined“ curve

- Determine $E_{pt} : y^2 z \equiv x^3 + A_{pt} x z^2 + B_{pt} z^3 \mod pt$

$$P_{pt} = (x_{pt} : y_{pt} : 1)$$

- Requirement: $A_{pt} \equiv A \mod p$

$$A_{pt} \equiv A_t \mod t \text{ etc.}$$

- Using the Chinese Remainder Theorem (CRT): $A_{pt} :=$ CRT(A, A_t) etc.

- First, compute $Q_{pt} := kP_{pt}$ on E_{pt}

We have $Q_{pt} \equiv kP \mod p$ and

$$Q_{pt} \equiv kP_t \mod t$$

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

A New SCF-Secure Algorithm for $k \cdot P$

SCF-Secure Scalar Multiplication $Q = kP$	
#	Precomputation (during production time of device)
1	Choose prime t and “small“ curve E_t
2	Determine the “combined“ curve E_{pt}
#	Main (computations on the device)
3	Set $Q := kP_{pt}$ on E_{pt}
4	Set $R := kP_t$ on E_t
5	If $R \neq Q \bmod t$ then **output** „failure“ else **output** Q on E_p

- Analysis: Order of E_t is security parameter
- undetectable faults: Adversary needs $O(2^{\mathrm{ord}(E_t)})$ guesses

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Conclusion

Summary:

- New Sign Change Attacks
- New Countermeasure

Open Problems:

- Extend the idea to curves over binary fields
- Other specialized fault types?

Thank you!

Appendix

On Choosing E_t

Theorem 1: (Hasse)
Given $E_p : y^2 z \equiv x^3 + Axz^2 + Bz^3 \mod p$, it is

$$p + 1 - 2\sqrt{p} \leq \#E_p \leq p + 1 + 2\sqrt{p}.$$

Fact 2: $\exists\ p^2 - p$ different elliptic curves over $\mathbb{F}_p$.

Theorem 6: (Deurich)
There exists a constant $c > 0$ such that there are at least $c \cdot (p\sqrt{p})/\log(p)$ many elliptic curves for every given group order.

On Choosing E_t

Conjecture 7: (Cramer and Goldwasser/Kilian)
There exist constants $c_1, c_2 > 0$ such that
$\pi(t + 2\sqrt{t}) - \pi(t - 2\sqrt{t}) \geq c_2\sqrt{t}/\log^{c_1}(t)$.

Theorem 8: Choose $(A_t, x_t, y_t) \in \mathbb{Z}_t^3$ uniformly at random. (A_t, x_t, y_t) defines E_t uniquely. If Conjecture 7 is true, then $\exists c > 0$ such that the probability that E_t has prime order is at least

$$\frac{c \cdot c_2}{\log^{1+c_1}(t)},$$

where c_1, c_2 are as in Conjecture 7.

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Montgomery's skalar Multiplikation

Montgomery Algorithm: $Q = k \cdot P$

init $P1_{(n-1)} := P$ and $P2_{(n-1)} := 2P$ and $n := bits(k)$

main for i from n-2 downto 0 do

if ($k_i = 0$) then set $P2_{(i)} := P1_{(i+1)} + P2_{(i+1)}$

$P1_{(i)} := 2P1_{(i+1)}$

if ($k_i = 1$) then set $P1_{(i)} := P1_{(i+1)} + P2_{(i+1)}$

$P2_{(i)} := 2P2_{(i+1)}$

output $Q = P1_{(0)}$

$\tilde{Q} = \left(\frac{l_i(k)}{2^i} - 1\right) \cdot (Q - l_i(k)P) + l_i(k)P$, where $l_i(k) = \sum_{j=0}^{i} k_j 2^j$

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK
Paderborn Institute for Scientific Computation

Projektive Addition

$$P_1 = (x_1 : y_1 : z_1),\ P_2 = (x_2 : y_2 : z_2),\ P_1 + P_2 = P_3 = (x_3 : y_3 : z_3)$$

$$\begin{aligned} x_3 &:= r^2 - tw^2 \\ 2y_3 &:= vr - mw^3 \\ z_3 &:= z_1 z_2 w, \end{aligned}$$

where $u_1 := x_1 z_2^2, \quad s_1 := y_1 z_2^3, \quad w := u_1 - u_2, \quad r := s_1 - s_2,$

$u_2 := x_2 z_1^2, \quad s_2 := y_2 z_1^3, \quad t := u_1 + u_2, \quad m := s_1 + s_2,$

and $v := tw^2 - 2x_3$

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

PaSCo GK Paderborn Institute for Scientific Computation