CARAMBA seminar at the LORIA

SIKE Channels

Élise Tasso (CEA), `elise.tasso2@cea.fr`
joint work with Luca De Feo (IBM Research), Nadia El Mrabet (EMSE), Aymeric Genêt (EPFL/Nagra Kudelski Group), Novak Kaluđerović (EPFL), Natacha Linard de Guertechin (CY-SEC SA) and Simon Pontié (CEA)

April 5th, 2022

LSCO, SAS joint research team at the Centre of Microelectronics in Provence, Gardanne, France

1. Context: SIKE and hardware attacks
2. Theoretical isogeny computation side-channel attack
3. Side-channel attack in a laboratory on an isogeny computation implementation
4. Countermeasure

Context: SIKE and hardware attacks

SIKE in the NIST PQC Standardization Contest

- Quantum computer threat.
- NIST Post Quantum Cryptography Standardization Contest for asymmetric cryptography algorithms (since 2016).

SIKE is one of the NIST alternate candidates for encryption and key encapsulation.

- The only one based on isogenies between elliptic curves.
- Relatively slow: on an Intel CPU, $(9681 + 10343) \cdot 10^3$ cycles for encapsulation + decapsulation **vs** $(1862 + 1747) \cdot 10^3$ cycles for the slowest among the other candidates at the lowest security level.
- Smallest public key size : 330 bytes (p434, uncompressed) **vs** 672 bytes for the smallest key among the other candidates at the lowest security level.

Finite fields and field extensions

Let p be prime. In SIKE, $p = 2^{e_2}3^{e_3} - 1$ with $e_2 = 216$ and $e_3 = 137$ (p434).

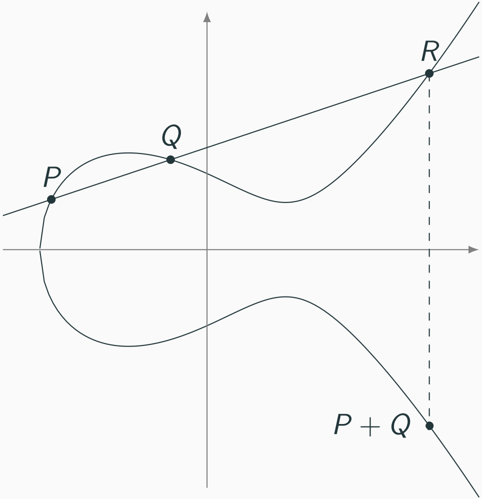
$\mathbb{F}_p = \{0, ..., p-1\}$ is the finite field with p elements.

$\mathbb{F}_{p^2}$ is an extension of $\mathbb{F}_p$. Its elements are of the form $a + ib$ where $i^2 = -1$ and a, $b \in \mathbb{F}_p$. If $a + ib$, $c + id \in \mathbb{F}_{p^2}$, then

$$(a + ib)(c + id) = (ac - bd) + i(ad + bc).$$

Elliptic curves - definition

- Montgomery curve E with an equation of the form $By^2 = x^3 + Ax^2 + x$ defined on $\mathbb{F}_{p^2}$.
- The points of an elliptic curve form a group.
- The neutral element is the point at infinity O.
- The addition law can be defined geometrically.



Elliptic curves - coordinates

Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be points of E such that $P \neq \pm Q$. Then for $R = P + Q$, we have

$$x_R = B\left(\frac{y_p - y_Q}{x_p - x_Q}\right)^2 - (x_P + x_Q) - A$$

and

$$y_R = \left(\frac{y_p - y_Q}{x_p - x_Q}\right)(x_P - x_R) - y_P.$$

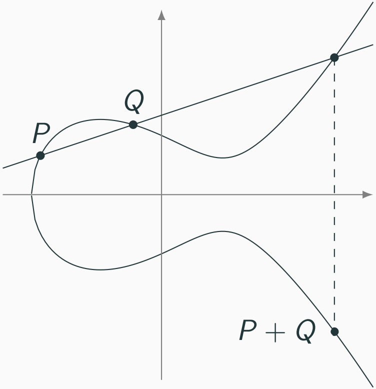
- For efficiency reasons, projective coordinates $(X : Y : Z)$ such that $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$ are used to avoid inversions in the formulas.
- It is also possible to only use the x coordinates, which also improves the performances.

Elliptic curves - projective $(X : Z)$ coordinates

- In projective coordinates, we will represent the curve by coefficients $(A_{24}^{+} : A_{24}^{-})$ such that $A_{24}^{+} \neq A_{24}^{-}$ and they are projectively equivalent to $(A + 2 : A - 2)$.
- $(A_{24}^{+} : A_{24}^{-}) = (0 : 0)$ represents an undefined curve.
- $(A_{24}^{+} : A_{24}^{-})$ such that $A_{24}^{+} = A_{24}^{-}$ represents a degenerate curve.

- $(0 : 0)$ is the undefined point.
- $(X : 0)$ with $X \neq 0$ represents the point at infinity O.

The curves can be represented by a triplet of distinct points P, Q and $P+Q$ with non-zero x coordinates.



$$x_P, x_Q, x_{P+Q}$$

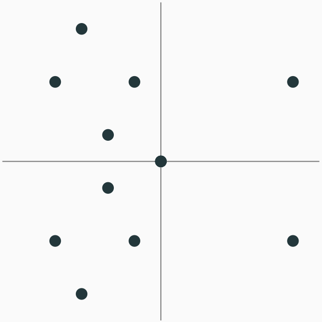
Elliptic curves - useful points

Let $n \in \mathbb{N}$.

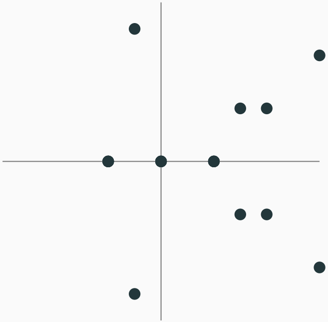
- The n-torsion $E[n]$ is the set of points P of E such that $nP = O$.
- A point P is of order n if n is the smallest integer k such that $kP = O$. We write $\mathrm{ord}(P) = n$.

Isogenies - example with $\mathbb{F}_{11}$

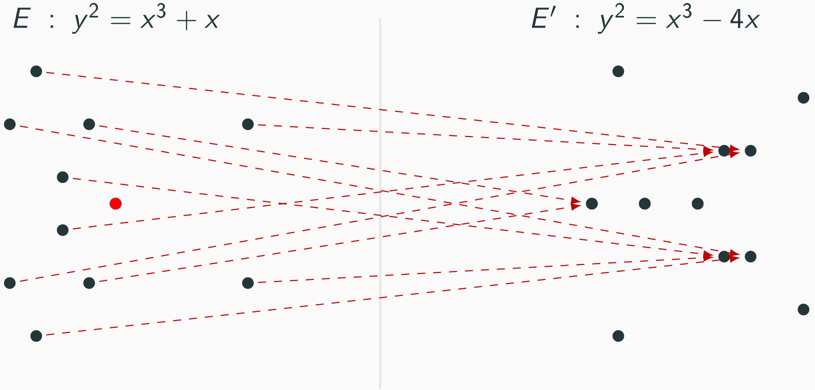
$$E \ : \ y^2 = x^3 + x$$



$$E' \ : \ y^2 = x^3 - 4x$$



$$\phi(x,y) = \left(\frac{x^2+1}{x}, \quad y\frac{x^2-1}{x^2} \right)$$



$$\phi(x,y) = \left(\frac{x^2+1}{x},\quad y\frac{x^2-1}{x^2}\right)$$

Isogenies - how to define them?

- The kernel of an isogeny ϕ is the set of points $P \in E$ such that

$$\phi(P) = O.$$

- In SIKE, the kernel is generated by one point G.
- This generator G **suffices** to define the isogeny.

Isogenies - computation

- The degree of an isogeny is the number of points of its kernel.
- It measures the "complexity" of the isogeny.
- **Problem:** Finding an isogeny of fixed degree knowing its starting curve and target curve.
  - It is easy if only a few points are sent to infinity.
  - It is hard if a lot of points are sent to infinity.
- Isogenies of large degree in SIKE are computed as composition of small-degree isogenies.

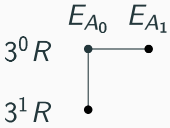
Isogenies - computation with a strategy

In SIKE, an isogeny of degree 3^2 with a kernel generated by a point R is computed as follows:

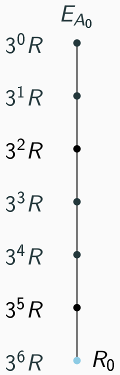
$$\phi = \phi_1 \circ \phi_0$$

where ϕ_0 and ϕ_1 are 3-isogenies such that

- $\mathrm{Ker}(\phi_0) = \langle 3R \rangle$ and
- $\mathrm{Ker}(\phi_1) = \langle \phi_0(R) \rangle$.



Visualization of a tree traversal (3^2-isogeny computation).

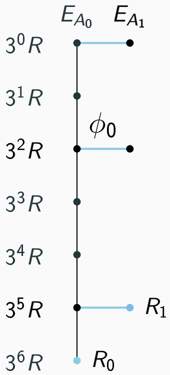


Visualization of a tree traversal (3^7-isogeny computation).

We want to compute the 3^7-isogeny with kernel $\langle R \rangle$.

We compute the point 3^6R and save the points 3^0R, 3^2R and 3^5R.

Isogenies - computation with a strategy



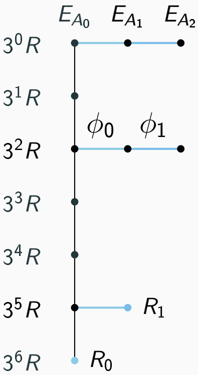
Visualization of a tree traversal (3^7-isogeny computation).

We compute ϕ_0 such that

$$\mathrm{Ker}(\phi_0) = \langle 3^6 R \rangle,$$

$$\phi_0(3^0R),\ \phi_0(3^2R) \text{ and } \phi_0(3^5R).$$

Isogenies - computation with a strategy



Visualization of a tree traversal (3^7-isogeny computation).

We compute ϕ_1 such that

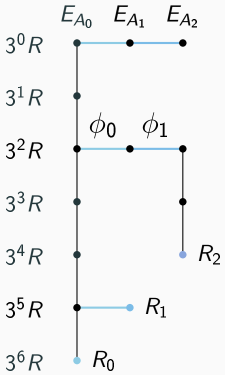
$$\mathrm{Ker}(\phi_1) = \langle 3^5\phi_0(R)\rangle,$$

$$\phi_1 \circ \phi_0(3^0R),$$

and

$$\phi_1 \circ \phi_0(3^2R).$$

Isogenies - computation with a strategy



Visualization of a tree traversal (3^7-isogeny computation).

We want to compute ϕ_2 such that

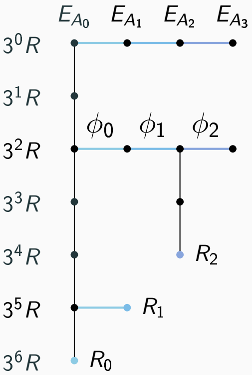
$$\mathrm{Ker}(\phi_2) = \langle 3^4 \phi_1 \circ \phi_0(R) \rangle,$$

but $3^4 R$ is not a saved point. Thus we compute

$$3^4 \phi_1 \circ \phi_0(R) = 3^2 \phi_1 \circ \phi_0(3^2 R)$$

by tripling.

Isogenies - computation with a strategy



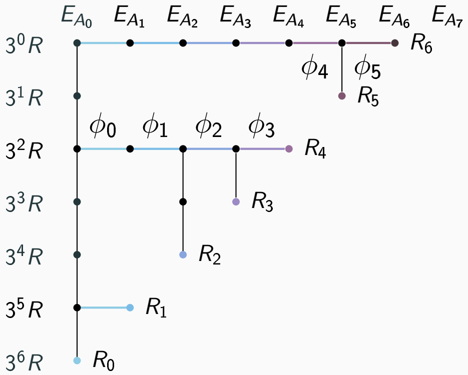
Visualization of a tree traversal (3^7-isogeny computation).

We can now compute ϕ_2,

$$\phi_2 \circ \phi_1 \circ \phi_0(3^0 R)$$

and

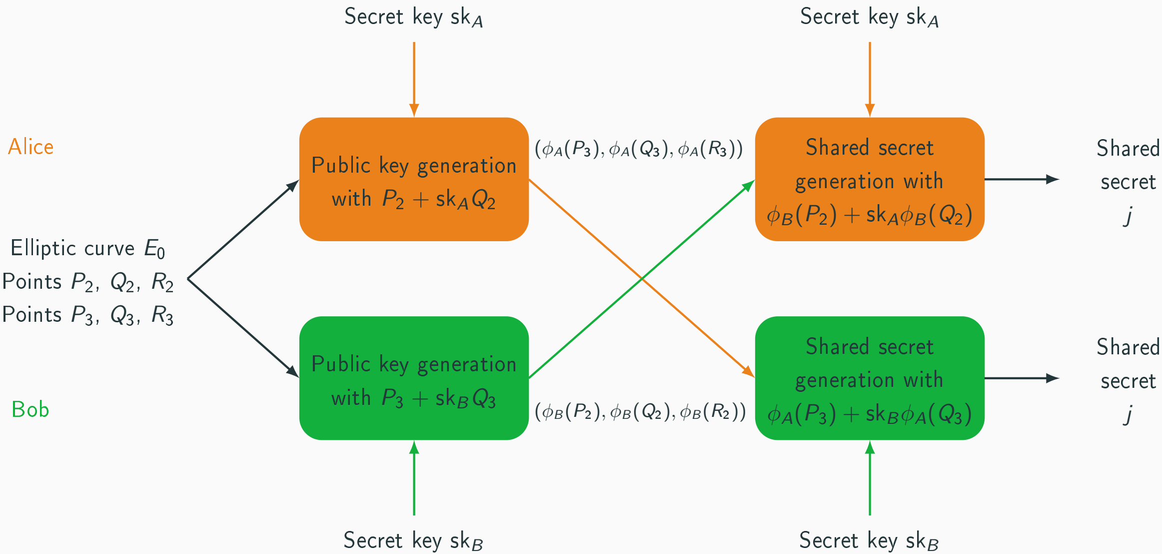
$$\phi_2 \circ \phi_1 \circ \phi_0(3^2 R).$$



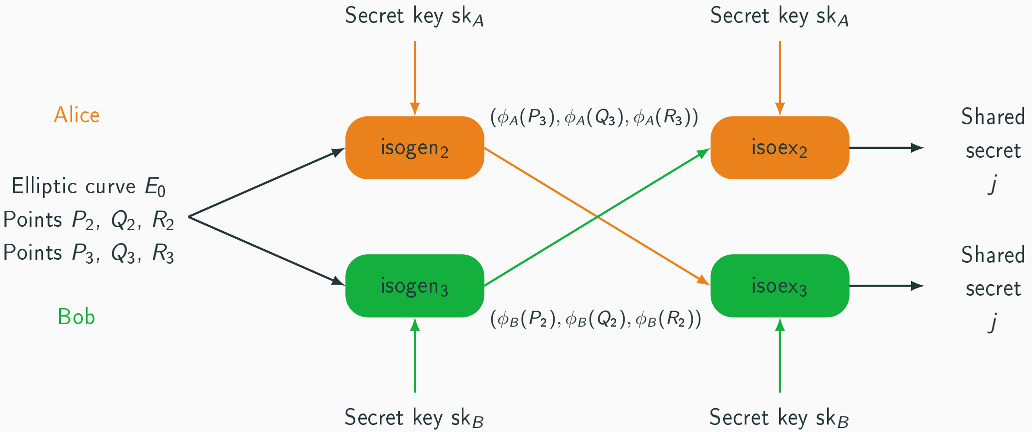
Visualization of a tree traversal (3^7-isogeny computation).

We continue like this until we have computed all isogenies of order 3 that make up the 3^7-isogeny.

The SIDH key exchange

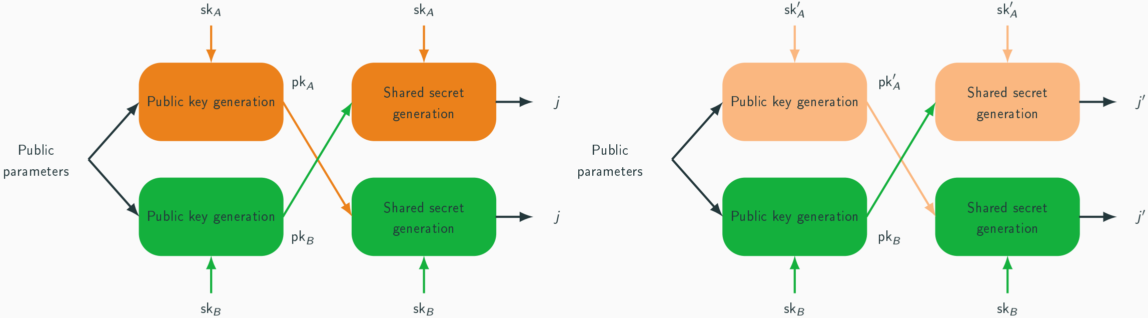


The SIDH key exchange



Why not use SIDH directly ?

SIDH is mathematically insecure if one of the secret keys is static (Galbraith et al., 2016).



SIKE is mathematically secure in "semi-static mode".

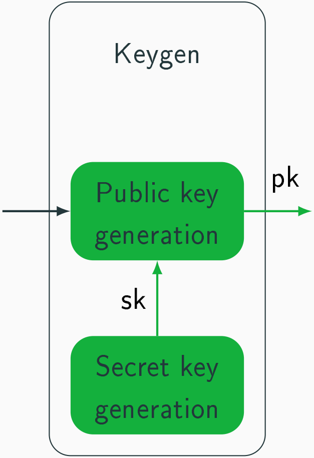
The SIKE mechanism

Public parameters

Elliptic curve E_0

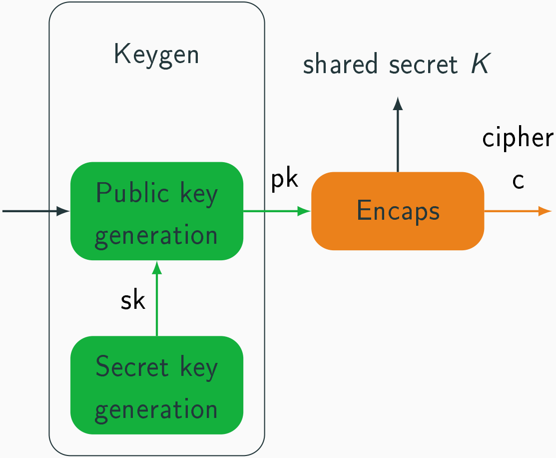
Points (P_2, Q_2, R_2)

Points (P_3, Q_3, R_3)

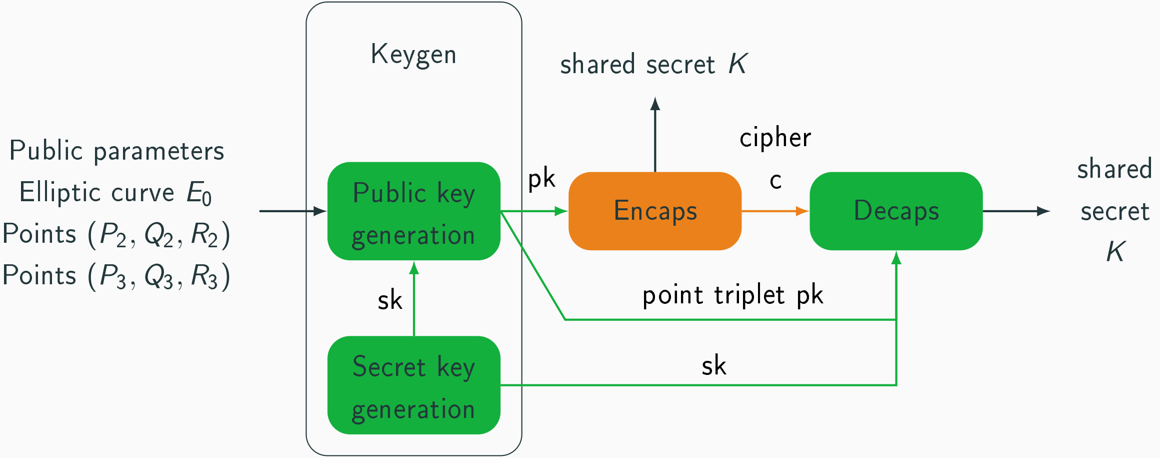


The SIKE mechanism

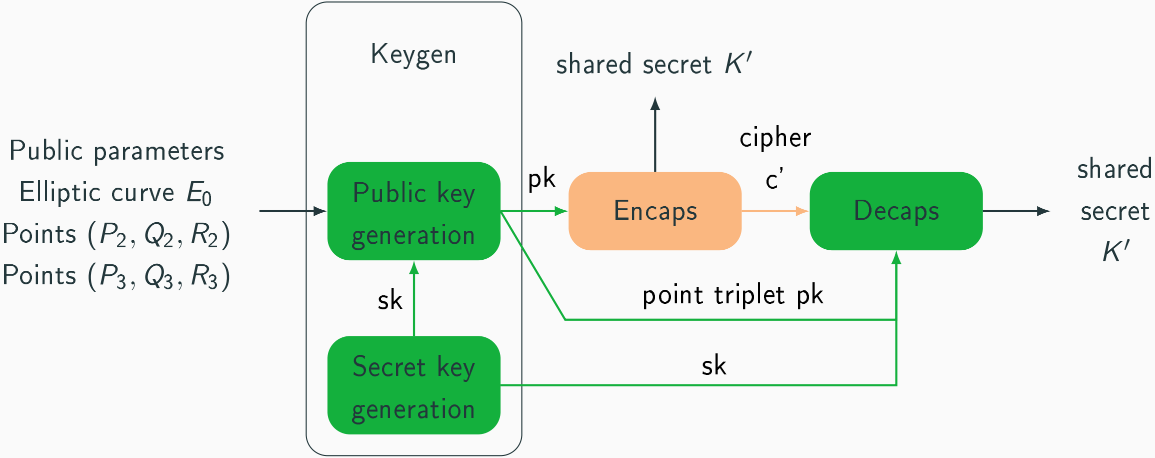
Public parameters
Elliptic curve E_0
Points (P_2, Q_2, R_2)
Points (P_3, Q_3, R_3)



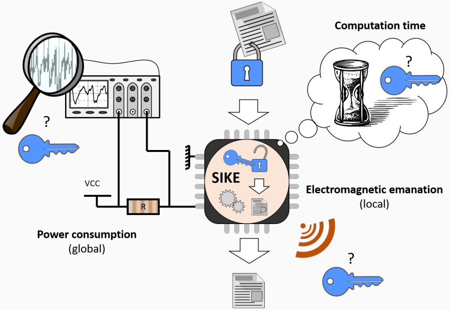
The SIKE mechanism



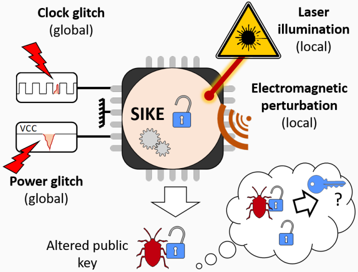
The SIKE mechanism



There are two types of hardware attacks.



Side-channel attacks



Fault attacks

Hardware attacks on SIKE : state of the art

SIKE is believed to be mathematically secure, but hardware attacks may exist depending on the implementation...

- Regularity of SIKE
- Attacks taking advantage of ECC or of the isogeny computation

	Fault injection	**Side-channel attacks**
Theoretical	Yan Bo Ti, 2017	Koziel et al., 2017
Simulated	Gélin et al., 2017	none
Experimentally verified	Tasso et al., 2021	Koppermann et al., 2018 Zhang et al., 2020 Genêt et al., 2021

Focus on side-channel analysis

- Koppermann et al., Zhang et al. and Genêt et al. perform DPAs/CPAs on classical ECC.
- They recommend **projective coordinate randomization** as a countermeasure: if the affine coordinate is x, pick a random Z and use projective coordinates $(xZ : Z)$.
- Zero-value coordinates are not affected by coordinate randomization.
- Koziel et al.: zero-point attacks (ZPA, a form of RPA) are presented but they cannot be applied to the SIKE case.

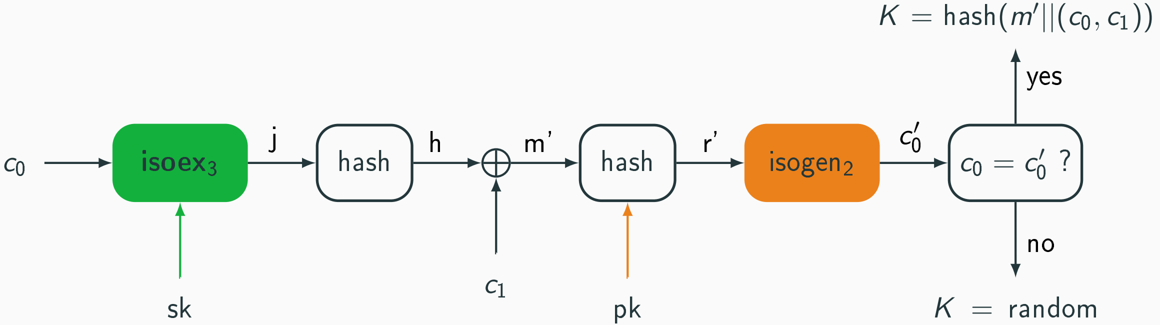
Our work

- Is there a theoretical side-channel attack on SIKE that bypasses coordinate randomization?
- Is this attack exploitable in practice?
- What are fitting countermeasures ?

Theoretical isogeny computation side-channel attack

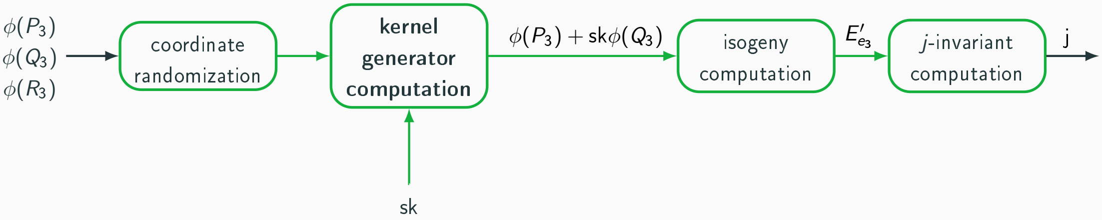
Where do we attack?

In Decaps, during the computation of the j-invariant.



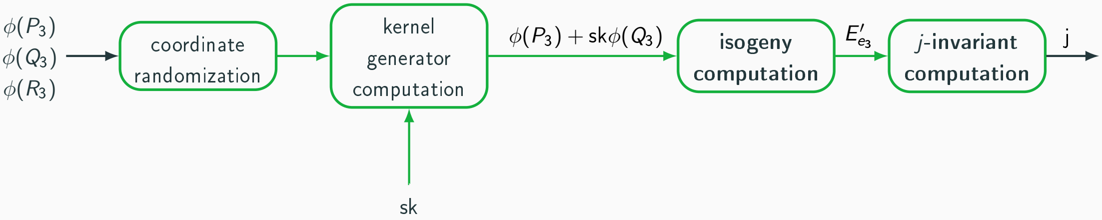
Where do we attack?

In isoex_3, during the computation of the isogeny kernel generator...



Where do we attack?

...or during the isogeny and j-invariant computation.



Attack principle

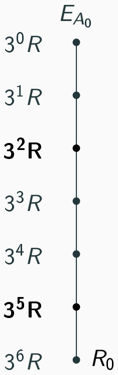
Goal: recover the secret key bit by bit.

Assume that we know bits $sk_0, ..., sk_{k-1}$ of the secret key. We choose a point triplet such that the target's kernel generator R will make

- zero values appear in the computations if $sk_k = 0$ and
- arbitrary values appear if $sk_k \neq 0$.

As 0 is not sensitive to the randomization countermeasure, the two cases will be distinguishable using a side channel.

Example: isogeny computation with a kernel of wrong order

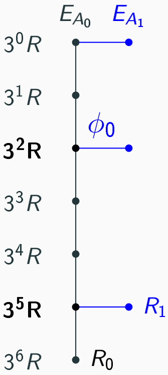


We want to perform a 3^7-isogeny computation with a kernel of incompatible order (i.e. a power of 2) generated by a point R.

Assume that $3^2 R = 3^5 R$.

First, we compute $3^6 R$ on curve E_{A_0} and save points $3^0 R$, $3^2 R$ and $3^5 R$.

Example: isogeny computation with a kernel of wrong order



We want then to compute isogeny ϕ_0 of kernel $\langle R_0 \rangle$ such that $R_0 = 3^6 R$ with expression

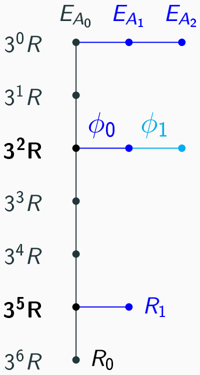
$$\phi_0((X:Z)) = (X(XX_{R_0} - ZZ_{R_0})^2 : Z(XZ_{R_0} - ZX_{R_0})^2).$$

We have $3^2R = 3^5R$. Let $(x\lambda : \lambda)$ with $\lambda \neq 0$ be the coordinates of a point on E_{A_0}. Then

$$\begin{aligned}\phi_0((\lambda x : \lambda)) &= (\lambda x(\lambda x X_{R_0} - \lambda Z_{R_0})^2 : \lambda(\lambda x Z_{R_0} - \lambda X_{R_0})^2) \\ &= (\lambda^3 x(xX_{R_0} - Z_{R_0})^2 : \lambda^3(xZ_{R_0} - X_{R_0})^2) \\ &= (x(xX_{R_0} - Z_{R_0})^2 : (xZ_{R_0} - X_{R_0})^2).\end{aligned}$$

Thus $\phi_0(3^2R) = \phi_0(3^5R)$.

Example: isogeny computation with a kernel of wrong order

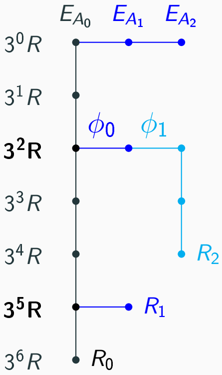


Next, we compute isogeny ϕ_1 with kernel generator R_1 such that $R_1 = 3^5 \phi_0(R)$.

As $\phi_0(3^2 R) = \phi_0(3^5 R)$, we get

$$\phi_1 \circ \phi_0(3^2 R) = O.$$

Example: isogeny computation with a kernel of wrong order



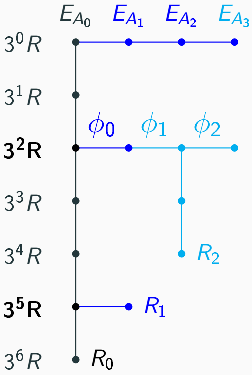
The following isogeny is ϕ_2 with kernel generator R_2 such that $R_2 = 3^4\phi_1 \circ \phi_0(R)$.

We triple $\phi_1 \circ \phi_0(3^2 R)$ on E_{A_2} to compute it. The formula for the tripling of a point is

$$\begin{aligned} 3(X:Z) = ((A_{24}^{+} - A_{24}^{-})(X^4 - 6X^2Z^2 - 3Z^4) \\ - 8(A_{24}^{+} + A_{24}^{-})XZ^3)^2 Z : \\ (A_{24}^{+} - A_{24}^{-})(3X^4 + 6X^2Z^2 - Z^4) \\ + 4(A_{24}^{+} - A_{24}^{-})X^3Z)^2 Z). \end{aligned}$$

So tripling O on E_{A_2} will yield O. Thus $R_2 = O$.

Example: isogeny computation with a kernel of wrong order



The kernel of isogeny ϕ_2 is O. The formula for its target curve coefficients is

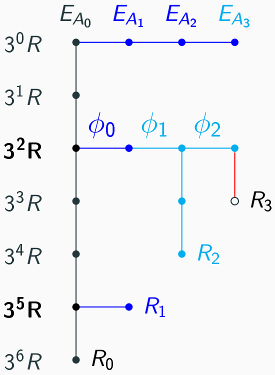
$$(A_{24}^+ : A_{24}^-) = ((3X_{R_2} - Z_{R_2})^3(X_{R_2} + Z_{R_2}) : (3X_{R_2} + Z_{R_2})^3(X_{R_2} - Z_{R_2}))$$

and yields

$$(A_{24}^+ : A_{24}^-) = (1 : 1)$$

. Thus E_{A_3} is a degenerate curve.

Example: isogeny computation with a kernel of wrong order

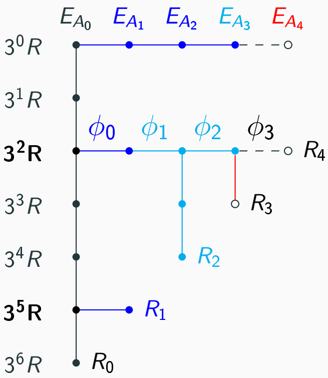


The kernel generator of isogeny ϕ_3 is $3^3\phi_2 \circ \phi_1 \circ \phi_0(R)$. We compute it by tripling $\phi_2 \circ \phi_1 \circ \phi_0(3^2R)$ (equal to O) on the degenerate curve E_{A_3}. As $(A_{24}^+ : A_{24}^-) = (1 : 1)$, the formula

$$\begin{aligned} 3(X : Z) = ((A_{24}^+ - A_{24}^-)(X^4 - 6X^2Z^2 - 3Z^4) \\ - 8(A_{24}^+ + A_{24}^-)XZ^3)^2 Z : \\ (A_{24}^+ - A_{24}^-)(3X^4 + 6X^2Z^2 - Z^4) \\ + 4(A_{24}^+ - A_{24}^-)X^3Z)^2 Z) \end{aligned}$$

yields $(0 : 0)$, which represents an undefined point.

Example: isogeny computation with a kernel of wrong order



With this kernel generator, we compute the coefficients of the target curve of ϕ_3:

$$(A_{24}^+ : A_{24}^-) = ((3X_{R_3} - Z_{R_3})^3(X_{R_3} + Z_{R_3}) : (3X_{R_3} + Z_{R_3})^3(X_{R_3} - Z_{R_3})).$$

We get $(A_{24}^+ : A_{24}^-) = (0 : 0)$, which represents the undefined curve E_{A_4}.

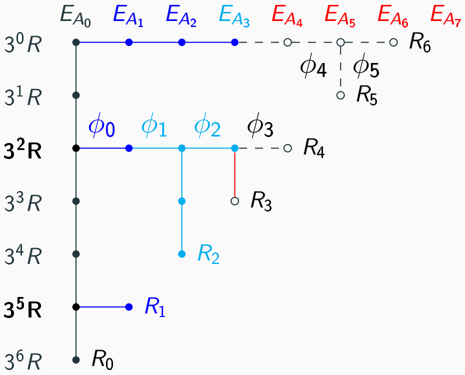
The expression of ϕ_3 is given by

$$\phi_3((X : Z)) = (X(XX_{R_3} - ZZ_{R_3})^2 : Z(XZ_{R_3} - ZX_{R_3})^2)$$

thus $\phi_3((X : Z)) = (0 : 0)$.

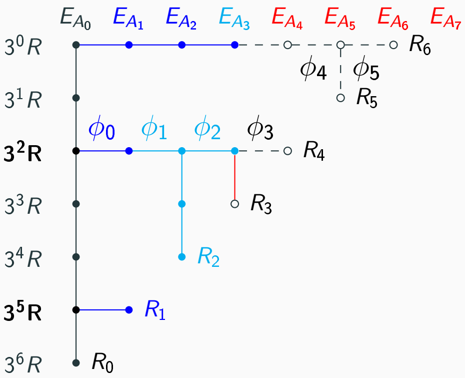
The image of O by ϕ_3 is then $(0 : 0)$.

Example: isogeny computation with a kernel of wrong order



From now on, only undefined points and curves will appear.

Example: isogeny computation with a kernel of wrong order



Black : points of wrong order and supersingular elliptic curves.

Blue : arbitrary points, isogenies with random image and arbitrary (non-supersingular) elliptic curves.

Cyan : the point O, isogenies with image $\{O\}$, triplings of O and degenerate elliptic curves.

Red : the tripling which first creates the undefined point $(0 : 0)$, and undefined elliptic curves.

Choosing R

Purple: saved points on the first branch of the strategy.

Bold: equal points among the saved points.

ord(R)	2	2^2	2^3	2^4	2^5	2^6
3^0 R	**R**	**R**	**R**	R	R	R
3^1 R	R	3R	3R	3R	3R	3R
3^2 R	**R**	**R**	**R**	**9R**	9R	9R
3^3 R	R	3R	3R	11R	27R	27R
3^4 R	R	R	R	R	17R	17R
3^5 R	**R**	3R	3R	3R	19R	51R
3^6 R	**R**	**R**	**R**	**9R**	25R	25R

The break-point exponent o

There is an exponent $o > 0$ such that

L1. if $\mathrm{ord}(R) \big| 2^{o-1}$ then isogenies always lead to undefined points $(0 : 0)$ and

L2. if $2^{o} \big| \mathrm{ord}(R)$ then arbitrary values are computed.

The exponent o depends on

- the isogeny degree and
- the tree-traversal strategy.

Creating a malicious point triplet

Given o, we send the target a triplet of points such that the computed kernel generator R such that $R = P + skQ$ is of order 2^{o-1+sk_k}.

Input: Index of bit being guessed k, known part of secret key $sk_{<k}$, a public parameter o
Assumes: $k \leq e_2 - o$
Output: Public key $pk_k^j = (P, Q, Q - P)$.

$E \leftarrow$ any supersingular elliptic curve
$P_2, Q_2 \leftarrow$ generators of $E[2^{e_2}]$
Assume $[2^{e_2-1}]Q_2 \neq T$ where $x_T = 0$.
$S = [2^{e_2-(o-1)}]P_2$
$Q = [2^{e_2-(k+o)}]Q_2$
$P = S - [sk_{<k}]Q$
return $pk_k^j = (P, Q, Q - P)$

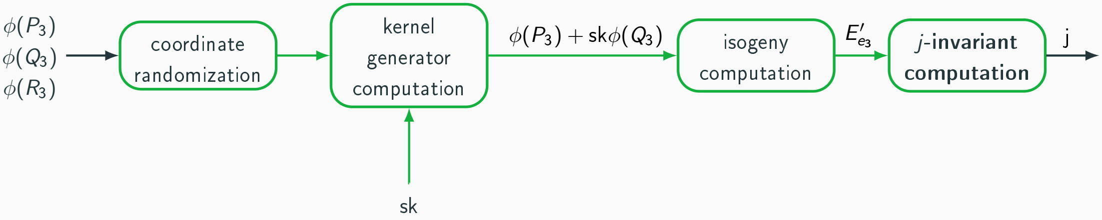
Creating a malicious point triplet

The kernel generator point $R = P + skQ$ generated from the public key pk_k^j of satisfies

$$\mathrm{ord}(R) = \begin{cases} 2^{o-1} & \text{if } sk_k = 0, \\ 2^{o} & \text{if } sk_k \neq 0. \end{cases}$$

Where in SIKE do we perform the zero-value distinction?

Depending on the value of the secret bit, from a certain point in the tree traversal, the victim either computes only zero values or arbitrary values.



We then look at the **field inversion** within the j-invariant computation because

- it is one of the last steps of the "key exchange" and
- there are a lot of field operations because $a^{-1} = a^{p-2}$.

In-lab attack scenario

Input: Breaking point o.
Output: The secret key sk.

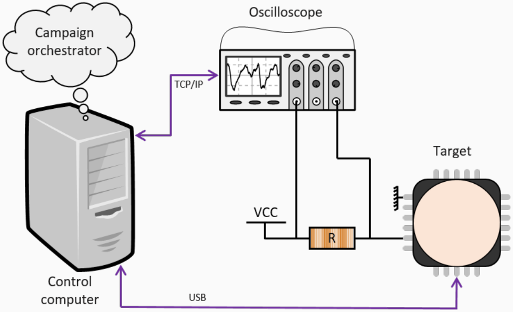
```
for k = 0 to e_2 − o do
    Assume we know sk_{<k} = Σ_{i=0}^{k−1} sk_i 2^i.
    Generate a malicious triplet pk_k^j with (k, sk_{<k}, o).
    Send pk_k^j to the target.
    Side-channel analysis of exponentiation
    if computation of 0^{p−2} is detected then sk_k = 0
    else sk_k = 1
end
Brute-force the remaining bits of the secret key.
return sk
```

Side-channel attack in a laboratory on an isogeny computation implementation

Attacked SIKE implementation

- Cortex-M4 software implementation of the j-invariant computation of SIKE of the NIST PQC Standardization Process round 3 submission with added projective coordinate randomization.
- Target choice: attack in a laboratory of a CW308T-STM32F3 microcontroller featuring an ARM Cortex-M4 (recommended by the NIST) at 44MHz using the ChipWhisperer framework.

Set up of an attack campaign



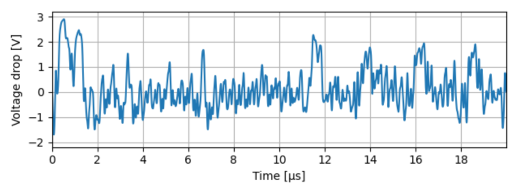
Set up for the realization of a side-channel attack campaign

Goal: recover a bit sk_k of the secret knowing the previous bits $sk_0, \ldots, sk_{k-1}$.

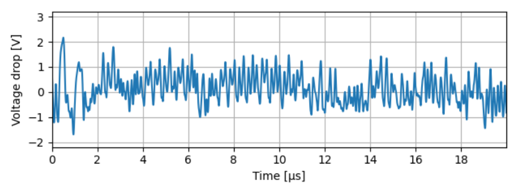
Experimental procedure

We first record baselines of the first field multiplication with two types of input:

- A malicious triplet such that zeros appear during the field multiplication and
- A malicious triplet such that random values appear during the field multiplication.



(a) Zero-valued baseline.



(b) Random-valued baseline.

We then record a trace of the power consumption of the board performing the first field multiplication with as input the malicious triplet presented in the previous section.

Experimental results - Pearson correlation coefficient

For each bit, we measure the similarity

- between the trace and the zero-valued baseline and
- between the trace and the random-valued baseline

with a Pearson correlation coefficient (PCC). The highest PCC yields the correct bit value.

Experimental results - Pearson correlation coefficient

	Target	
Baselines	$j = 0$	$j \neq 0$
$j = 0$	**0.9975**	0.3915
$j \neq 0$	0.3916	**0.9909**

Average PCCs between baselines and target traces ($N = 1,000$).

Thus zero values can be detected by observation of the power comsumption of the first field multiplication.

Countermeasure

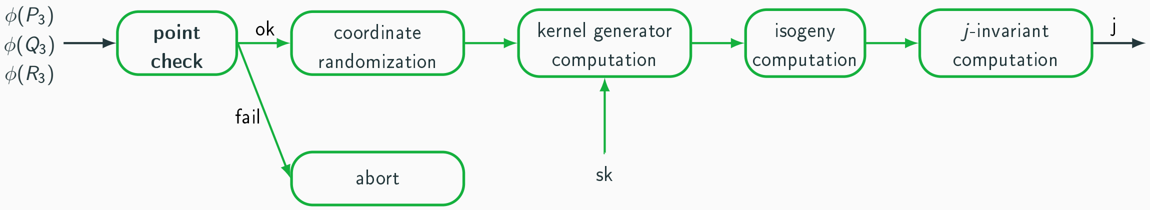
Countermeasure

The attack uses some malformed input points of order 2^n instead of 3^{e_3}. Costello, Longa and Naehrig propose the following test in a 2016 paper: check that

- P and Q are both of order 3^{e_3} and
- they generate the 3^{e_3}-torsion.

This is done by verifying that $3^{e_3-1}P \neq \pm 3^{e_3-1}Q \neq O$ and that $3^{e_3}P = 3^{e_3}Q = O$.

Countermeasure



This countermeasure has a 12.9% overhead (measured on a Cortex-M4).

Conclusion

- Both zero-point attacks enable a bit-by-bit recovery of the secret key.
- We verified them both experimentally using respectively the electromagnetic emissions and the power consumption of a Cortex-M4 core.
- The point check is sufficient to stop both attacks.