Post-Quantum Adaptor Signature for Privacy-Preserving Off-Chain Payments

Erkan Tairi [1] Pedro Moreno-Sanchez [2] Matteo Maffei [1]

de·crypto seminar, December 9, 2020

[1]TU Wien

[2]IMDEA Software Institute

Introduction and Motivation

Adaptor Signatures

- Previously known as **scriptless scripts**, and first introduced by Poelstra in 2017
- Recently formalized by Aumayr et al. in [AEE+20]

Adaptor Signatures

- Previously known as **scriptless scripts**, and first introduced by Poelstra in 2017
- Recently formalized by Aumayr et al. in [AEE+20]
- Advantages: low on-chain cost, improved fungibility of transactions, advanced functionality beyond blockchain's scripting language

Adaptor Signatures

- Previously known as **scriptless scripts**, and first introduced by Poelstra in 2017
- Recently formalized by Aumayr et al. in [AEE+20]
- Advantages: low on-chain cost, improved fungibility of transactions, advanced functionality beyond blockchain's scripting language
- Applications: payment channel networks, payment channel hubs, atomic swaps, etc.

Adaptor Signature

- Extends ordinary signatures
- Requires a hard relation compatible with the signature scheme
    - e.g., knowledge of discrete logarithm relation and Schnorr/ECDSA signature

Adaptor Signature

- Extends ordinary signatures
- Requires a hard relation compatible with the signature scheme
  - e.g., knowledge of discrete logarithm relation and Schnorr/ECDSA signature
- Can be viewed as a **two party "conditional signature"**
  - But, the final output is an ordinary signature from a single party (or a joint signature in the threshold setting)

Adaptor Signature - Formal Definition

- Let $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ be a digital signature scheme and $(Y, y) \in R$ be a hard relation (y secret witness, Y public statement)

Adaptor Signature - Formal Definition

- Let $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ be a digital signature scheme and $(Y, y) \in R$ be a hard relation (y secret witness, Y public statement)
  - $\hat{\sigma} \leftarrow \mathsf{PreSig}(\mathsf{sk}, m, Y)$

Adaptor Signature - Formal Definition

- Let $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ be a digital signature scheme and $(Y, y) \in R$ be a hard relation (y secret witness, Y public statement)
  - $\hat{\sigma} \leftarrow \mathsf{PreSig}(\mathsf{sk}, m, Y)$
  - $0/1 \leftarrow \mathsf{PreVer}(\mathsf{pk}, m, Y, \hat{\sigma})$

Adaptor Signature - Formal Definition

- Let $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ be a digital signature scheme and $(Y, y) \in R$ be a hard relation (y secret witness, Y public statement)
  - $\hat{\sigma} \leftarrow \mathsf{PreSig}(\mathsf{sk}, m, Y)$
  - $0/1 \leftarrow \mathsf{PreVer}(\mathsf{pk}, m, Y, \hat{\sigma})$
  - $\sigma \leftarrow \mathsf{Adapt}(\hat{\sigma}, y)$

Adaptor Signature - Formal Definition

- Let $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ be a digital signature scheme and $(Y, y) \in R$ be a hard relation (y secret witness, Y public statement)
  - $\hat{\sigma} \leftarrow \mathsf{PreSig}(\mathsf{sk}, m, Y)$
  - $0/1 \leftarrow \mathsf{PreVer}(\mathsf{pk}, m, Y, \hat{\sigma})$
  - $\sigma \leftarrow \mathsf{Adapt}(\hat{\sigma}, y)$
  - $y/\perp \leftarrow \mathsf{Ext}(\sigma, \hat{\sigma}, Y)$ such that $(Y, y) \in R$

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

$(\mathsf{sk}, \mathsf{pk})$
Y, m

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

(sk, pk)
Y, m

$\hat{\sigma} \leftarrow \mathrm{PreSig}(\mathrm{sk}, m, Y)$

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

(sk, pk)
Y, m

$\hat{\sigma} \leftarrow \text{PreSig}(\text{sk}, m, Y)$

$\hat{\sigma}$

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

$(\mathsf{sk}, \mathsf{pk})$
Y, m

$\hat{\sigma} \leftarrow \mathsf{PreSig}(\mathsf{sk}, m, Y)$

$\hat{\sigma}$

$\sigma \leftarrow \mathsf{Adapt}(\hat{\sigma}, y)$

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

$(\mathsf{sk}, \mathsf{pk})$
Y, m

$\hat{\sigma} \leftarrow \mathsf{PreSig}(\mathsf{sk}, m, Y)$

$\hat{\sigma}$

$\sigma \leftarrow \mathsf{Adapt}(\hat{\sigma}, y)$

σ

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

(sk, pk)
Y, m

$\hat{\sigma} \leftarrow \text{PreSig}(\text{sk}, m, Y)$

$\hat{\sigma}$

$\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y)$

σ

$y \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y)$

Adaptor Signature - Protocol View

$(Y, y) \in R$
pk, m

(sk, pk)
Y, m

$\hat{\sigma}$

$\hat{\sigma} \leftarrow \text{PreSig}(\text{sk}, m, Y)$

$\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y)$

σ

$y \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y)$

Observation:

- PreSig is like a **commitment**, such that Alice with a valid witness can **complete** the signature. And any valid $(\sigma, \hat{\sigma})$ pair **reveals** a witness.

Adaptor Signature - Properties

- **Unforgeability:** infeasible to forge a signature even when pre-signature is given without knowing a witness to R

Adaptor Signature - Properties

- **Unforgeability:** infeasible to forge a signature even when pre-signature is given without knowing a witness to R
- **Pre-signature Adaptability:** anyone that knows a witness to Y can complete a pre-signature conditioned on Y

Adaptor Signature - Properties

- **Unforgeability:** infeasible to forge a signature even when pre-signature is given without knowing a witness to R
- **Pre-signature Adaptability:** anyone that knows a witness to Y can complete a pre-signature conditioned on Y
- **Witness Extractability:** any given (pre-signature, signature) pair on the same statement Y reveals a witness to Y

Instantiation and Application

Schnorr Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure KGen(1^λ)

$x \leftarrow_\$ \mathbb{Z}_p$

return $(\mathsf{sk} := x, \mathsf{pk} := g^x)$

Schnorr Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure KGen(1^λ)
$x \leftarrow_\$ \mathbb{Z}_p$
return $(\mathsf{sk} := x, \mathsf{pk} := g^x)$

procedure Sig(sk, m)
$k \leftarrow_\$ \mathbb{Z}_p, R := g^k$
$e := H(\mathsf{pk} \| R \| m)$
$s := k + e \cdot \mathsf{sk}$
return $\sigma := (e, s)$

Schnorr Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure KGen(1^λ)
$x \leftarrow_\$ \mathbb{Z}_p$
return $(\mathsf{sk} := x, \mathsf{pk} := g^x)$

procedure Sig(sk, m)
$k \leftarrow_\$ \mathbb{Z}_p, R := g^k$
$e := H(\mathsf{pk} \| R \| m)$
$s := k + e \cdot \mathsf{sk}$
return $\sigma := (e, s)$

procedure Ver(pk, m, σ)
Parse σ as (e, s)
$e' := H(\mathsf{pk} \| g^s \cdot \mathsf{pk}^{-e} \| m)$
return $(e = e')$

Schnorr Adaptor Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure Sig(sk, m)
- $k \leftarrow\!\!\$\ \mathbb{Z}_p$, $R := g^k$
- $e := H(\mathsf{pk} \| R \| m)$
- $s := k + e \cdot \mathsf{sk}$
- **return** $\sigma := (e, s)$

procedure Ver(pk, m, σ)
- Parse σ as (e, s)
- $e' := H(\mathsf{pk} \| g^s \cdot \mathsf{pk}^{-e} \| m)$
- **return** $(e = e')$

Schnorr Adaptor Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure PreSig(sk, m, Y)
- $k \leftarrow\!\!\$\ \mathbb{Z}_p$, $R := g^k$
- $e := H(\mathsf{pk} \| R \cdot Y \| m)$
- $\hat{s} := k + e \cdot \mathsf{sk}$
- **return** $\hat{\sigma} := (e, \hat{s})$

procedure Ver(pk, m, σ)
- Parse σ as (e, s)
- $e' := H(\mathsf{pk} \| g^s \cdot \mathsf{pk}^{-e} \| m)$
- **return** $(e = e')$

Schnorr Adaptor Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure PreSig(sk, m, Y)
- $k \leftarrow\!\!\$\ \mathbb{Z}_p$, $R := g^k$
- $e := H(\mathsf{pk} \| R \cdot Y \| m)$
- $\hat{s} := k + e \cdot \mathsf{sk}$
- **return** $\hat{\sigma} := (e, \hat{s})$

procedure PreVer(pk, m, Y, $\hat{\sigma}$)
- Parse $\hat{\sigma}$ as $(e, \hat{s})$
- $e' := H(\mathsf{pk} \| g^{\hat{s}} \cdot \mathsf{pk}^{-e} \cdot Y \| m)$
- **return** $(e = e')$

Schnorr Adaptor Signature

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure PreSig(sk, m, Y)
- $k \leftarrow\!\!\$\ \mathbb{Z}_p$, $R := g^k$
- $e := H(\mathsf{pk} \| R \cdot Y \| m)$
- $\hat{s} := k + e \cdot \mathsf{sk}$
- **return** $\hat{\sigma} := (e, \hat{s})$

procedure PreVer(pk, m, Y, $\hat{\sigma}$)
- Parse $\hat{\sigma}$ as $(e, \hat{s})$
- $e' := H(\mathsf{pk} \| g^{\hat{s}} \cdot \mathsf{pk}^{-e} \cdot Y \| m)$
- **return** $(e = e')$

procedure Adapt($\hat{\sigma}$, y)
- Parse $\hat{\sigma}$ as $(e, \hat{s})$
- $s := \hat{s} + y$
- **return** $\sigma := (e, s)$

procedure Ext(σ, $\hat{\sigma}$, Y)
- Parse σ as (e, s), $\hat{\sigma}$ as $(e, \hat{s})$
- $y' := s - \hat{s}$
- **if** $(Y, y') \in R$ **return** y'
- **else return** $\perp$

Payment Channel Networks (PCNs)

- Malavolta et al. [MMS+19] introduced Anonymous Multi-Hop Locks (AMHLs) as a building block for secure and privacy-preserving PCNs

Payment Channel Networks (PCNs)

$y_1 \xleftarrow{\$} \mathbb{Z}_p$; $Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \xleftarrow{\$} \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

A

E_1

B

E_2

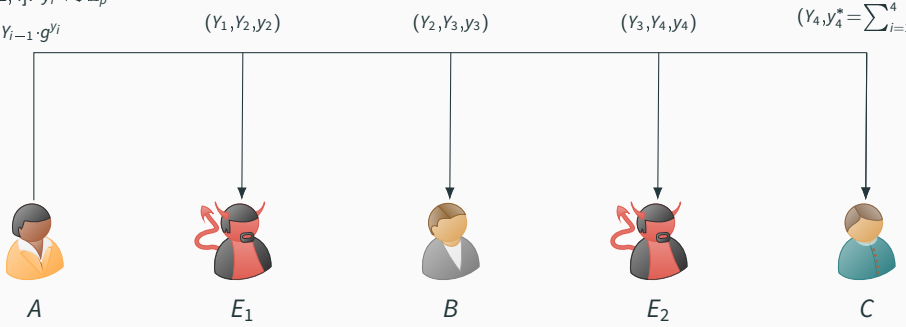
C

Payment Channel Networks (PCNs)

$y_1 \xleftarrow{\$} \mathbb{Z}_p;\ Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \xleftarrow{\$} \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

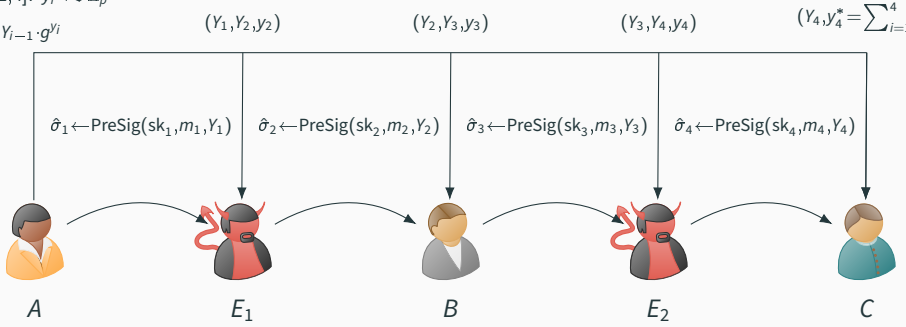


Payment Channel Networks (PCNs)

$y_1 \leftarrow\!\$\ \mathbb{Z}_p;\ Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \leftarrow\!\$\ \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

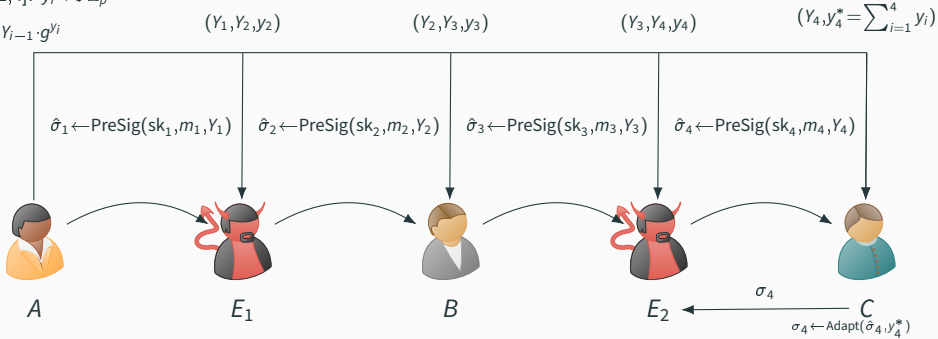


Payment Channel Networks (PCNs)

$y_1 \xleftarrow{\$} \mathbb{Z}_p;\ Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \xleftarrow{\$} \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

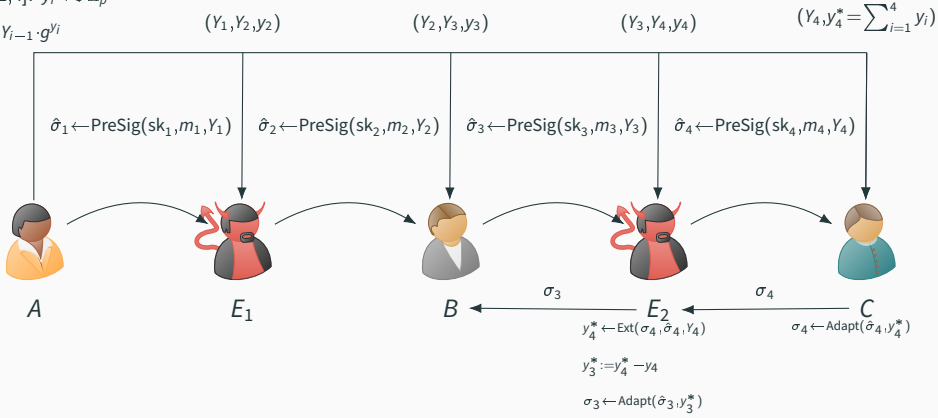


Payment Channel Networks (PCNs)

$y_1 \leftarrow\$ \mathbb{Z}_p;\ Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \leftarrow\$ \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

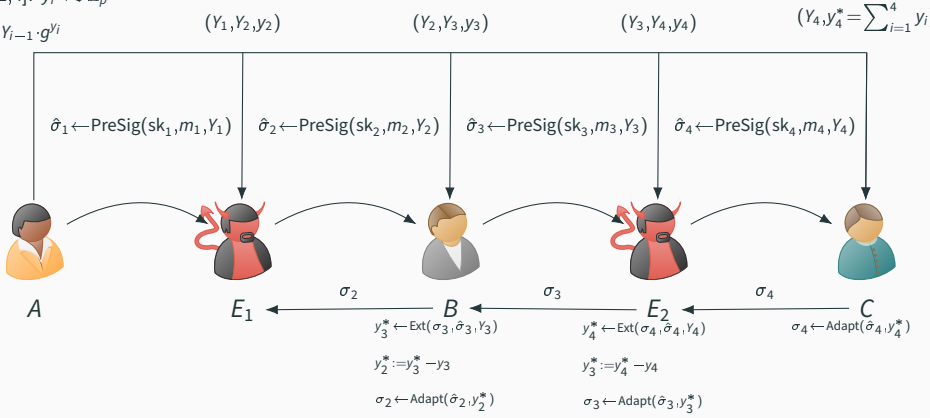


Payment Channel Networks (PCNs)

$y_1 \leftarrow\$ \mathbb{Z}_p;\ Y_1 := g^{y_1}$

for $i \in [2,4]:\ y_i \leftarrow\$ \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

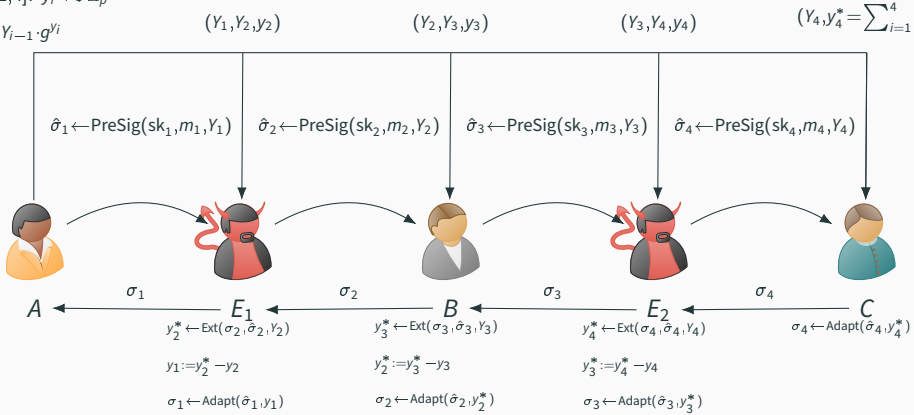


Payment Channel Networks (PCNs)

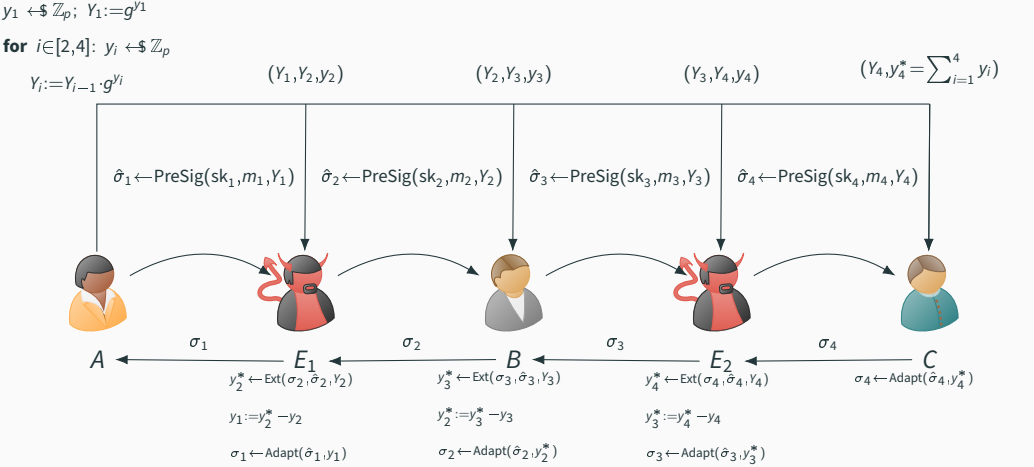
$y_1 \leftarrow\$ \mathbb{Z}_p$; $Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \leftarrow\$ \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$



Payment Channel Networks (PCNs)



- Additionally, the sender (A) needs to send a ZK proof for the well-formedness of objects to each intermediary.

Why post-quantum security?

- Existing adaptor signature (i.e., Schnorr and ECDSA) are broken with a quantum computer due to Shor's algorithm.

Why post-quantum security?

- Existing adaptor signature (i.e., Schnorr and ECDSA) are broken with a quantum computer due to Shor's algorithm.
- Ongoing standardization process by NIST.

Why post-quantum security?

- Existing adaptor signature (i.e., Schnorr and ECDSA) are broken with a quantum computer due to Shor's algorithm.
- Ongoing standardization process by NIST.

  Can we construct adaptor signatures that remain secure against quantum adversaries?

Why post-quantum security?

- Existing adaptor signature (i.e., Schnorr and ECDSA) are broken with a quantum computer due to Shor's algorithm.
- Ongoing standardization process by NIST.

Can we construct adaptor signatures that remain secure against quantum adversaries?

Yes!

Lattice-based Adaptor Signature

Lattice Adaptor Signature (LAS)

- Introduced by Esgin et al. [EEE20]
- Builds upon the Dilithium signature scheme [DKL$^+$18], which uses Fiat-Shamir with Aborts technique (rejection sampling)
- Based on well-known Module-SIS and Module-LWE problems
- Hard relation: knowledge of Module-SIS preimage
- Post-quantum PCNs and atomic swaps using LAS

Schnorr

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure KGen(1^λ)

$x \leftarrow\!\!\$\ \mathbb{Z}_p$

return $(\mathsf{sk} := x, \mathsf{pk} := g^x)$

Schnorr

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure KGen(1^λ)

$x \leftarrow_\$ \mathbb{Z}_p$

return $(\mathsf{sk} := x, \mathsf{pk} := g^x)$

LAS

Public parameters: a random matrix $\mathbf{A} \leftarrow_\$ \mathcal{R}^{n \times (n+\ell)}$, hash function $H\colon \{0,1\}^* \to \mathcal{C}$

Schnorr

Public parameters: group parameters $(\mathbb{G}, g, p)$, hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$

procedure KGen(1^λ)

$x \leftarrow_\$ \mathbb{Z}_p$

return $(\mathsf{sk} := x, \mathsf{pk} := g^x)$

LAS

Public parameters: a random matrix $\mathbf{A} \leftarrow_\$ \mathcal{R}^{n \times (n+\ell)}$, hash function $H\colon \{0,1\}^* \to \mathcal{C}$

procedure KGen(1^λ)

$\mathbf{x} \leftarrow_\$ \mathbb{S}_1^{n+\ell}$

return $(\mathsf{sk} := \mathbf{x}, \mathsf{pk} := \mathbf{Ax})$

- $\mathbb{S}_1^{n+\ell}$ is a distribution that samples vectors of size $n + \ell$ and of norm 1

Schnorr

procedure PreSig(sk, m, Y)

$k \leftarrow_{\$} \mathbb{Z}_p, R := g^k$

$e := H(\mathsf{pk} \| R \cdot Y \| m)$

$\hat{s} := k + e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{s})$

Schnorr

procedure PreSig(sk, m, Y)

$k \leftarrow\!\!\$\ \mathbb{Z}_p, R := g^k$

$e := H(\mathsf{pk} \| R \cdot Y \| m)$

$\hat{s} := k + e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{s})$

LAS

procedure PreSig(sk, m, $\mathbf{Y}$)

$\mathbf{k} \leftarrow\!\!\$\ \mathbb{S}_\gamma^{n+\ell}, \mathbf{R} := \mathbf{Ak}$

$e := H(\mathsf{pk} \| \mathbf{R} + \mathbf{Y} \| m)$

$\hat{\mathbf{s}} := \mathbf{k} + e \cdot \mathsf{sk}$

if $\|\hat{\mathbf{s}}\|_\infty > \gamma - \kappa - 1$ **then**

Restart

return $\hat{\sigma} := (e, \hat{\mathbf{s}})$

LAS

Schnorr

procedure PreSig(sk, m, Y)

$k \leftarrow\!\!\$\ \mathbb{Z}_p, R := g^k$

$e := H(\mathsf{pk} \| R \cdot Y \| m)$

$\hat{s} := k + e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{s})$

LAS

procedure PreSig(sk, m, $\mathbf{Y}$)

$\mathbf{k} \leftarrow\!\!\$\ \mathbb{S}_\gamma^{n+\ell}, \mathbf{R} := \mathbf{Ak}$

$e := H(\mathsf{pk} \| \mathbf{R} + \mathbf{Y} \| m)$

$\hat{\mathbf{s}} := \mathbf{k} + e \cdot \mathsf{sk}$

if $\|\hat{\mathbf{s}}\|_\infty > \gamma - \kappa - 1$ **then**

Restart

return $\hat{\sigma} := (e, \hat{\mathbf{s}})$

- Range of H (challenge set $\mathcal{C}$) is composed of elements from the ring $\mathcal{R}$ with certain norm restrictions
- If statement is used to perform rejection sampling, in order not to leak information about the secret key

Schnorr

procedure PreVer($\mathsf{pk}, m, Y, \hat{\sigma}$)

Parse $\hat{\sigma}$ as $(e, \hat{s})$

$e' = H(\mathsf{pk} \| g^{\hat{s}} \cdot \mathsf{pk}^{-e} \cdot Y \| m)$

return $(e = e')$

Schnorr

procedure PreVer($\mathsf{pk}, m, Y, \hat{\sigma}$)
  Parse $\hat{\sigma}$ as $(e, \hat{s})$
  $e' = H(\mathsf{pk} \| g^{\hat{s}} \cdot \mathsf{pk}^{-e} \cdot Y \| m)$
  return $(e = e')$

LAS

procedure PreVer($\mathsf{pk}, m, \mathbf{Y}, \hat{\sigma}$)
  Parse $\hat{\sigma}$ as $(e, \hat{\mathbf{s}})$
  if $\|\hat{\mathbf{s}}\|_\infty > \gamma - \kappa - 1$ **then**
    return 0
  $\mathbf{w} := \mathbf{A}\hat{\mathbf{s}} - e \cdot \mathsf{pk}$
  $e' = H(\mathsf{pk} \| \mathbf{w} + \mathbf{Y} \| m)$
  return $(e = e')$

LAS

- Adapt and Ext algorithms are analogous to Schnorr-based construction.
- The hard relation R considered so far is knowledge of Module-SIS with a small preimage (i.e., $(\mathbf{Y}, \mathbf{y}) \in R$ iff $\mathbf{Y} = \mathbf{A}\mathbf{y}$ and $\|\mathbf{y}\|_\infty \leq 1$)

procedure Adapt($\hat{\sigma}$, $\mathbf{y}$)
  Parse $\hat{\sigma}$ as $(e, \hat{\mathbf{s}})$
  $\mathbf{s} := \hat{\mathbf{s}} + \mathbf{y}$
  return $\sigma := (e, \mathbf{s})$

procedure Ext(σ, $\hat{\sigma}$, Y)
  Parse σ as $(e, \mathbf{s})$, $\hat{\sigma}$ as $(e, \hat{\mathbf{s}})$
  $\mathbf{y}' := \mathbf{s} - \hat{\mathbf{s}}$
  if $(\mathbf{Y}, \mathbf{y}') \in R$ **return** $\mathbf{y}'$
  else return $\perp$

LAS

- Adapt and Ext algorithms are analogous to Schnorr-based construction.
- The hard relation R considered so far is knowledge of Module-SIS with a small preimage (i.e., $(\mathbf{Y}, \mathbf{y}) \in R$ iff $\mathbf{Y} = \mathbf{A}\mathbf{y}$ and $\|\mathbf{y}\|_\infty \leq 1$)

procedure Adapt($\hat{\sigma}$, $\mathbf{y}$)
  Parse $\hat{\sigma}$ as $(e, \hat{\mathbf{s}})$
  $\mathbf{s} := \hat{\mathbf{s}} + \mathbf{y}$
  return $\sigma := (e, \mathbf{s})$

procedure Ext(σ, $\hat{\sigma}$, Y)
  Parse σ as $(e, \mathbf{s})$, $\hat{\sigma}$ as $(e, \hat{\mathbf{s}})$
  $\mathbf{y}' := \mathbf{s} - \hat{\mathbf{s}}$
  if $(\mathbf{Y}, \mathbf{y}') \in R$ **return** $\mathbf{y}'$
  else return $\perp$

Observation:
The witness $\mathbf{y}'$ output by Ext can have norm at most $\leq 2(\gamma - \kappa)$.

Drawbacks of LAS

- Due to inherent **knowledge gap** in lattice-based ZK proofs, it requires an **extended** relation R' such that $R \subseteq R'$ (i.e., witnesses can have bigger norm in R')

Drawbacks of LAS

- Due to inherent **knowledge gap** in lattice-based ZK proofs, it requires an **extended** relation R' such that $R \subseteq R'$ (i.e., witnesses can have bigger norm in R')
- **Weak Pre-signature Adaptability:** anyone that knows a y with $(Y, y) \in R$ can complete a pre-signature conditioned on Y
  - $\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y)$ where $(Y, y) \in R$
- **Witness Extractability:** any given (pre-signature, signature) pair on the same statement Y reveals a witness y' such that $(Y, y') \in R'$
  - $y'/\perp \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y)$ such that $(Y, y') \in R'$

Drawbacks of LAS

- Due to inherent **knowledge gap** in lattice-based ZK proofs, it requires an **extended** relation R' such that $R \subseteq R'$ (i.e., witnesses can have bigger norm in R')
- **Weak Pre-signature Adaptability:** anyone that knows a y with $(Y, y) \in R$ can complete a pre-signature conditioned on Y
  - $\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y)$ where $(Y, y) \in R$
- **Witness Extractability:** any given (pre-signature, signature) pair on the same statement Y reveals a witness y' such that $(Y, y') \in R'$
  - $y'/\perp \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y)$ such that $(Y, y') \in R'$

Observation:
- Extracted witnesses do **NOT** guarantee adaptability.

Drawbacks of LAS

- Due to inherent **knowledge gap** in lattice-based ZK proofs, it requires an **extended** relation R' such that $R \subseteq R'$ (i.e., witnesses can have bigger norm in R')
- **Weak Pre-signature Adaptability:** anyone that knows a y with $(Y, y) \in R$ can complete a pre-signature conditioned on Y
  - $\sigma \leftarrow \mathsf{Adapt}(\hat{\sigma}, y)$ where $(Y, y) \in R$
- **Witness Extractability:** any given (pre-signature, signature) pair on the same statement Y reveals a witness y' such that $(Y, y') \in R'$
  - $y'/\perp \leftarrow \mathsf{Ext}(\sigma, \hat{\sigma}, Y)$ such that $(Y, y') \in R'$

Observation:
- Extracted witnesses do **NOT** guarantee adaptability.

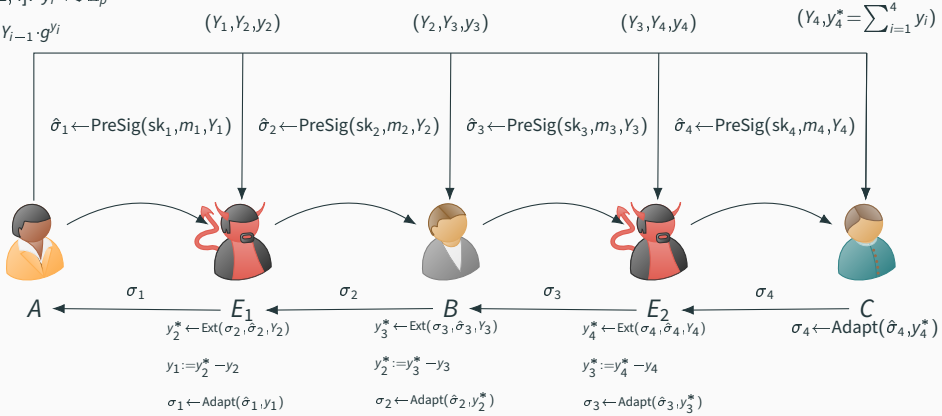
Solution:
- Use a lengthy ZK proof (e.g., the proof from [ENS20] is 47KB).

LAS-based PCN

$y_1 \leftarrow\$ \mathbb{Z}_p$; $Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \leftarrow\$ \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

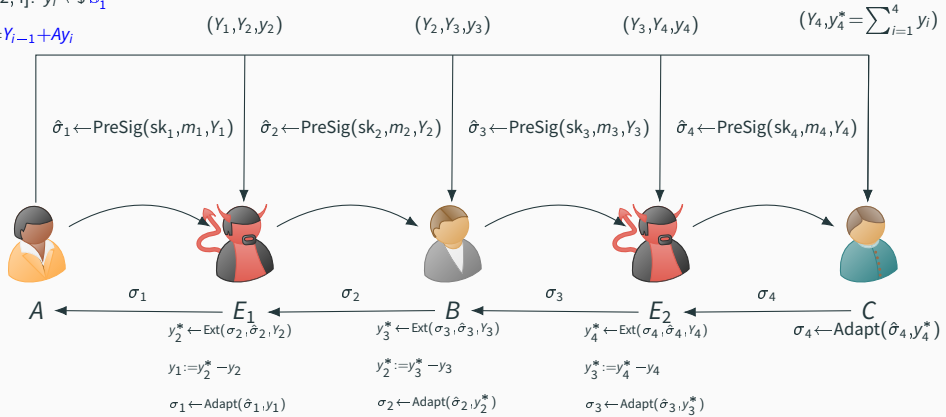


LAS-based PCN

$y_1 \leftarrow\$ \mathbb{S}_1^{n+\ell}$; $Y_1 := Ay_1$

for $i \in [2,4]$: $y_i \leftarrow\$ \mathbb{S}_1^{n+\ell}$

$Y_i := Y_{i-1} + Ay_i$

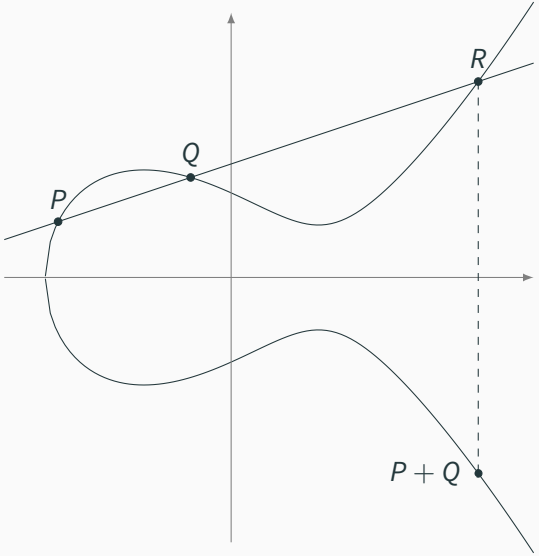


Observation:
The norm of the objects increases as we go along the path.

Background on Isogenies

Elliptic curves[1]

Define a group law such that any three colinear points add up to zero.

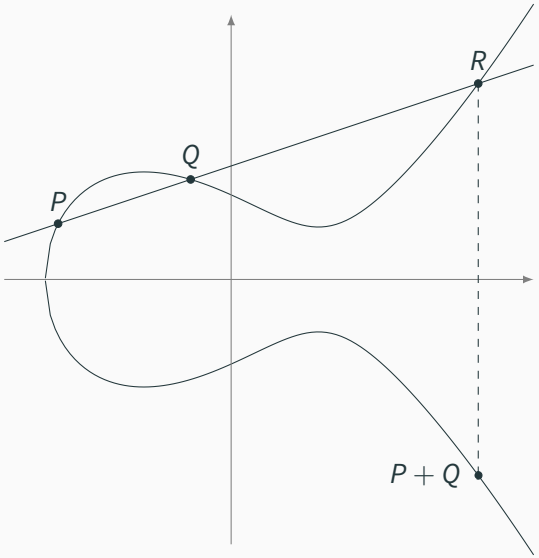


[1] Slides of this section are taken from Luca De Feo

Elliptic curves[1]

Define a group law such that any three colinear points add up to zero.

- The law is algebraic (it has *formulas*);

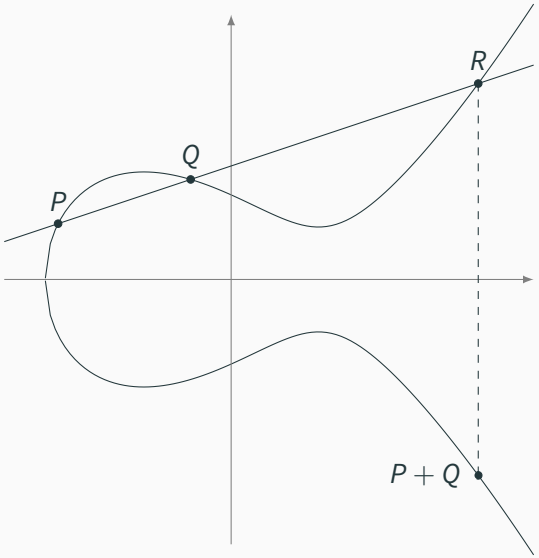


[1] Slides of this section are taken from Luca De Feo

Elliptic curves[1]

Define a group law such that any three colinear points add up to zero.

- The law is algebraic (it has *formulas*);
- The law is commutative;
- $\mathcal{O}$ is the group identity;
- Opposite points have the same x-value.



[1] Slides of this section are taken from Luca De Feo

Theorem
Let $\phi : E \to E'$ be a map between elliptic curves. These conditions are equivalent:

- ϕ is a surjective group morphism,
- ϕ is a group morphism with finite kernel,
- ϕ is a non-constant algebraic map of projective varieties sending the point at infinity of E onto the point at infinity of E'.

If they hold ϕ is called an isogeny.

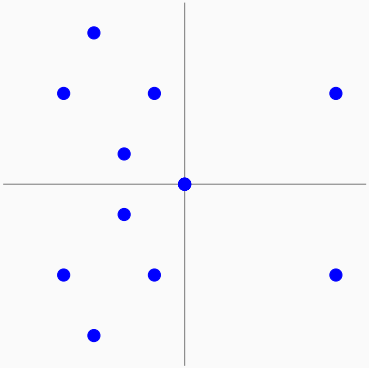
Definition
Let $E : y^2 = x^3 + ax + b$, its ***j*-invariant** is

$$j(E) = 1728\frac{4a^3}{4a^3 + 27b^2}.$$

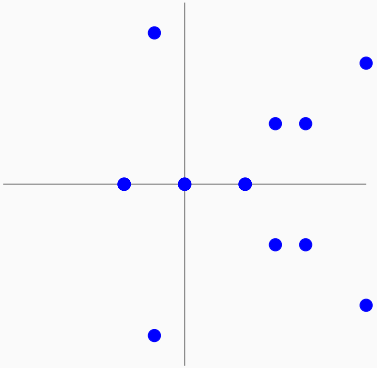
Two elliptic curves E, E' are **isomorphic** if and only if $j(E) = j(E')$.

Isogenies: an example over $\mathbb{F}_{11}$

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



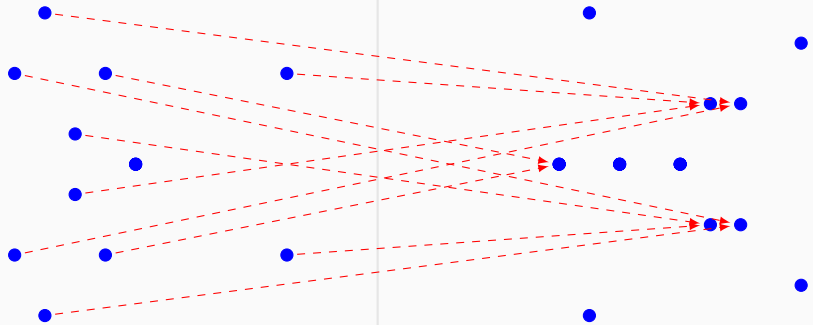
$E' : y^2 = x^3 - 4x$



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

Isogenies: an example over $\mathbb{F}_{11}$

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

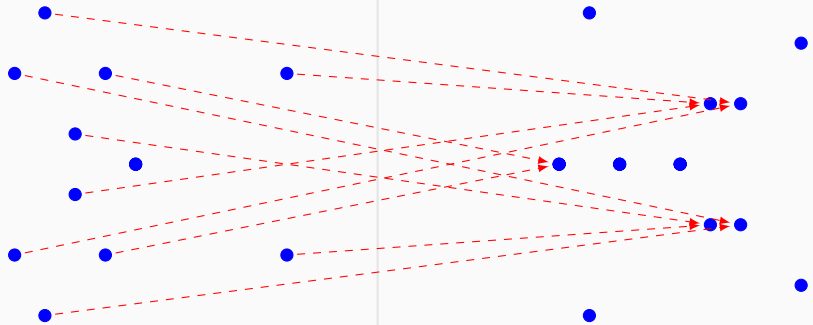


$$\phi(x, y) = \left(\frac{x^2 + 1}{x}, \quad y\frac{x^2 - 1}{x^2} \right)$$

Isogenies: an example over $\mathbb{F}_{11}$

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

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



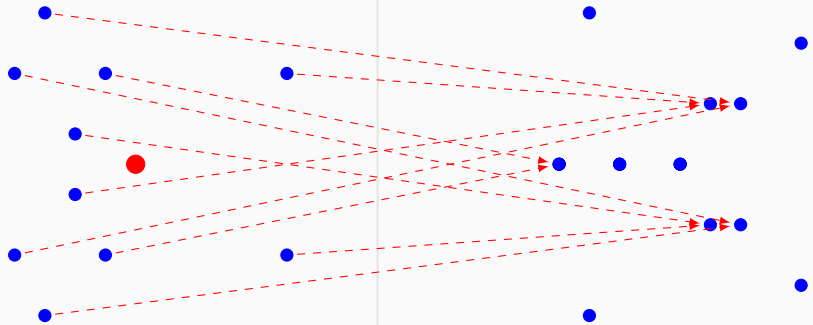
$$\phi(x, y) = \left(\frac{x^2 + 1}{x}, \quad y \frac{x^2 - 1}{x^2} \right)$$

- Kernel generator in red.
- This is a degree 2 map.

Isogenies: an example over $\mathbb{F}_{11}$

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

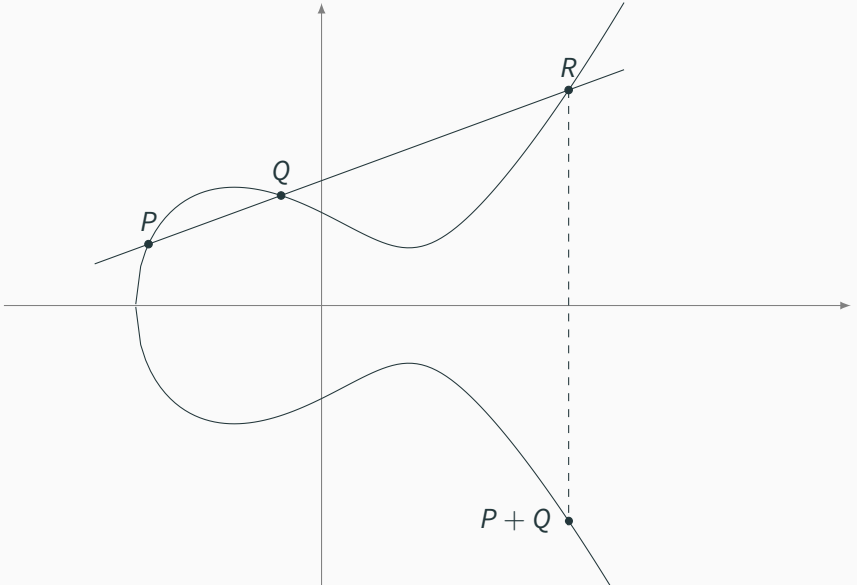
$E' : y^2 = x^3 - 4x$



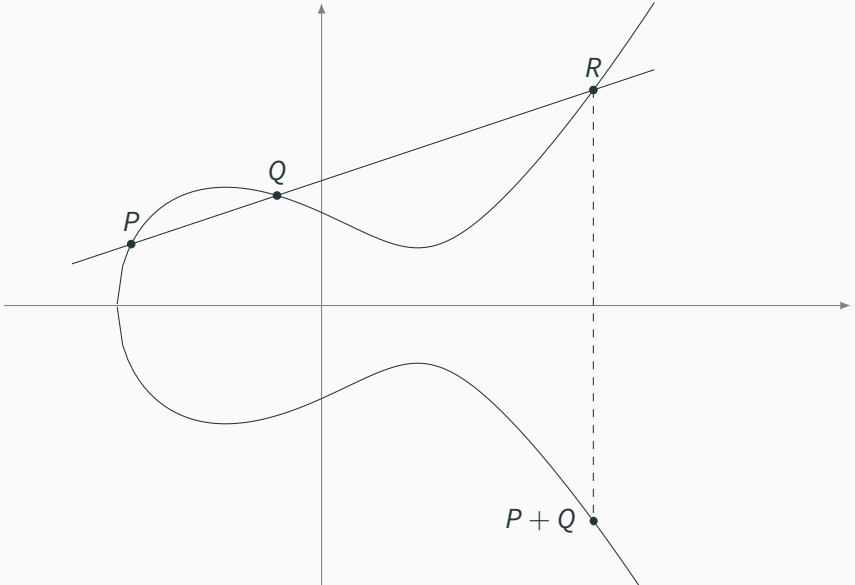
$$\phi(x, y) = \left(\frac{x^2 + 1}{x}, \quad y \frac{x^2 - 1}{x^2} \right)$$

- Kernel generator in red.
- This is a degree 2 map.

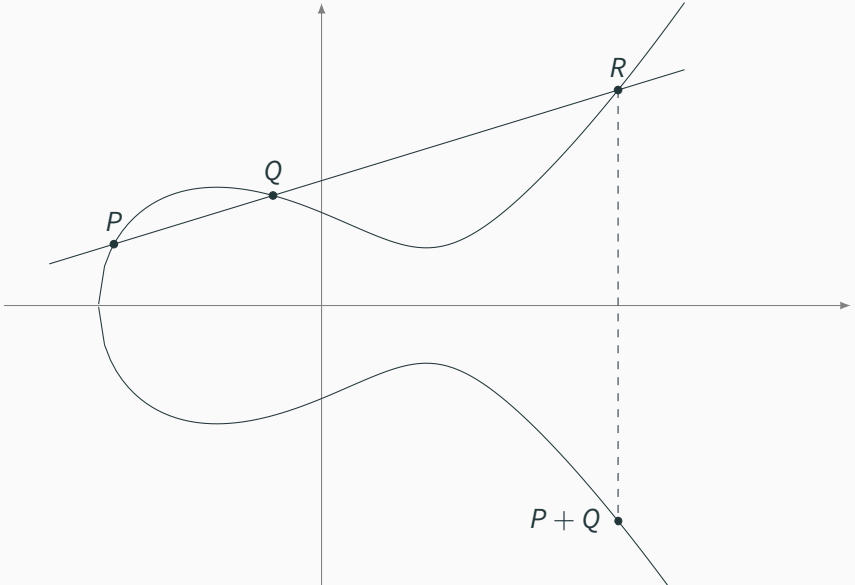
Up to isomorphism



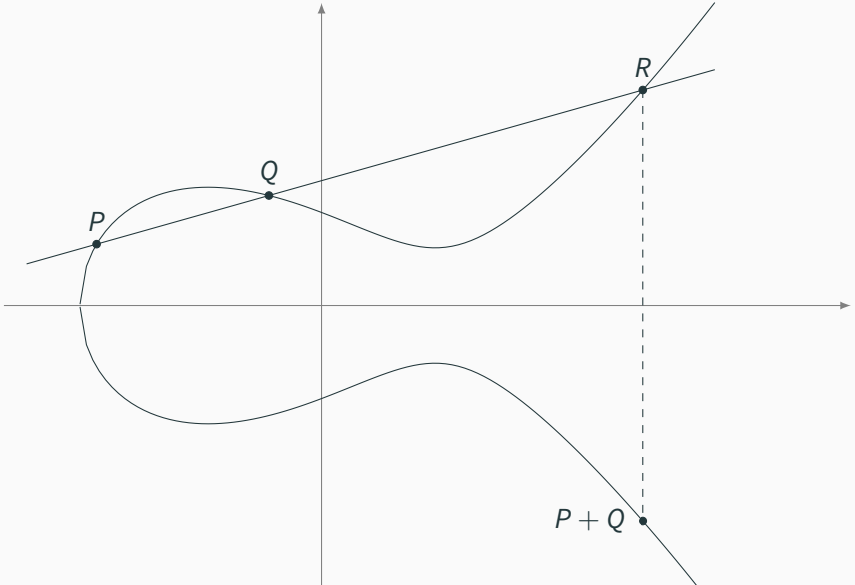
Up to isomorphism



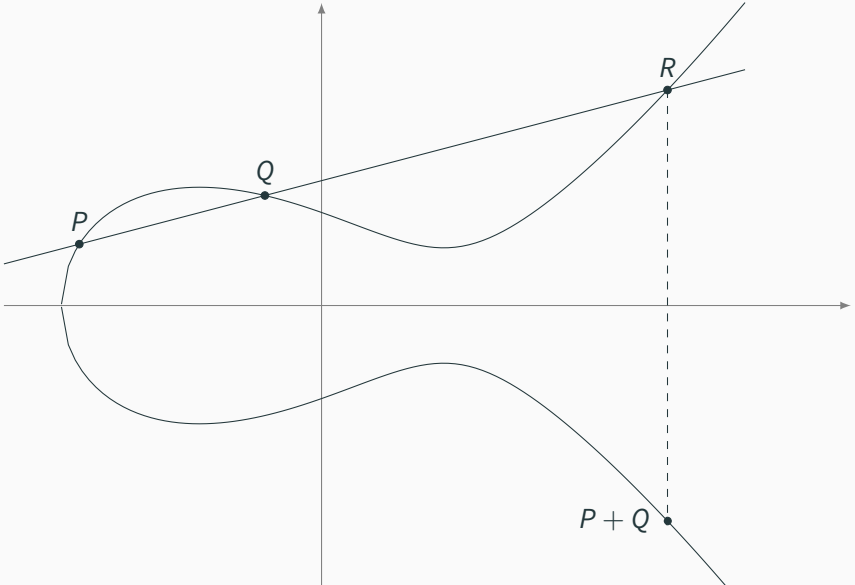
Up to isomorphism



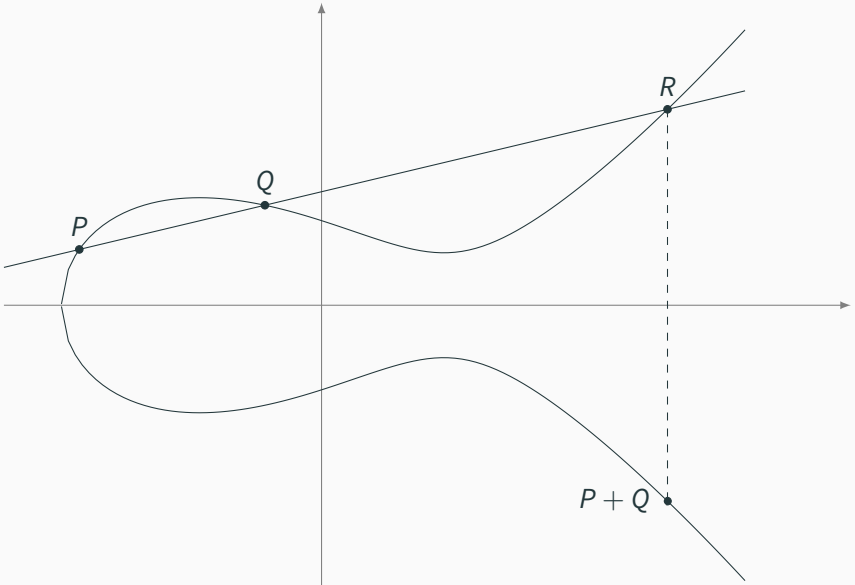
Up to isomorphism



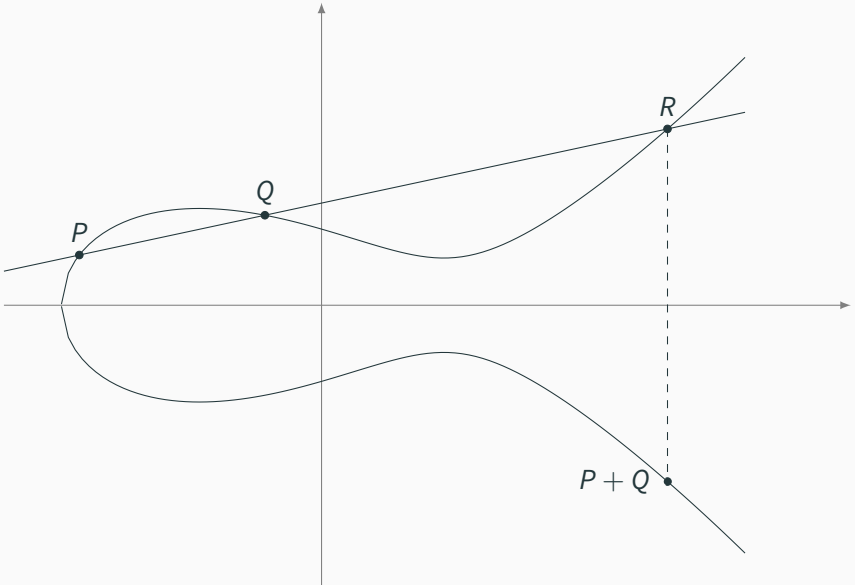
Up to isomorphism



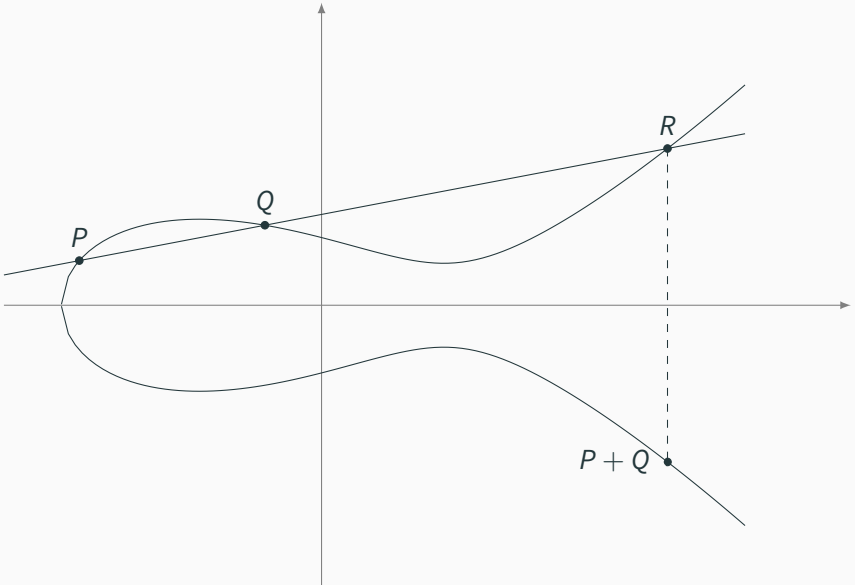
Up to isomorphism



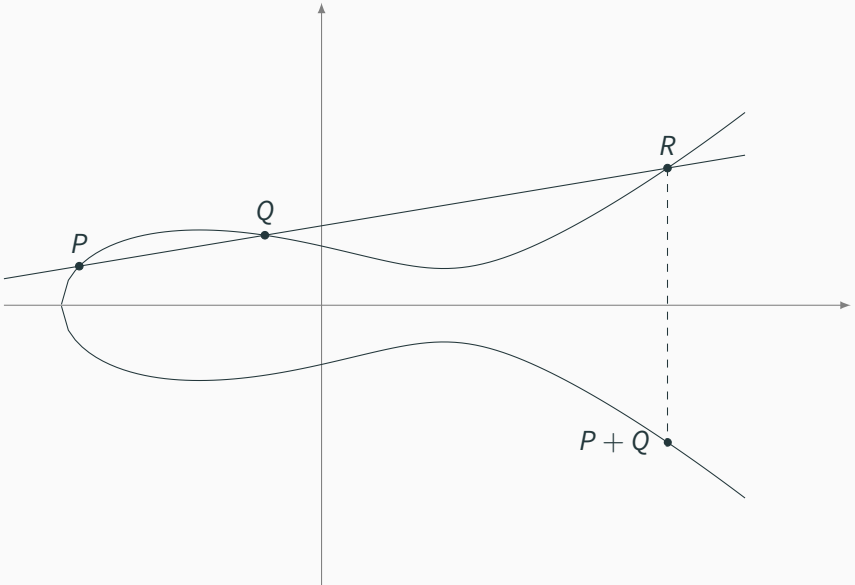
Up to isomorphism



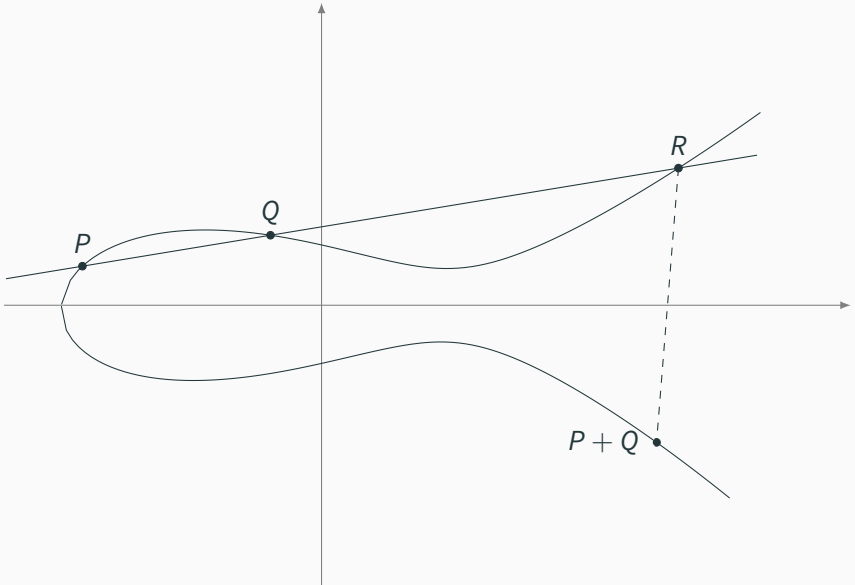
Up to isomorphism



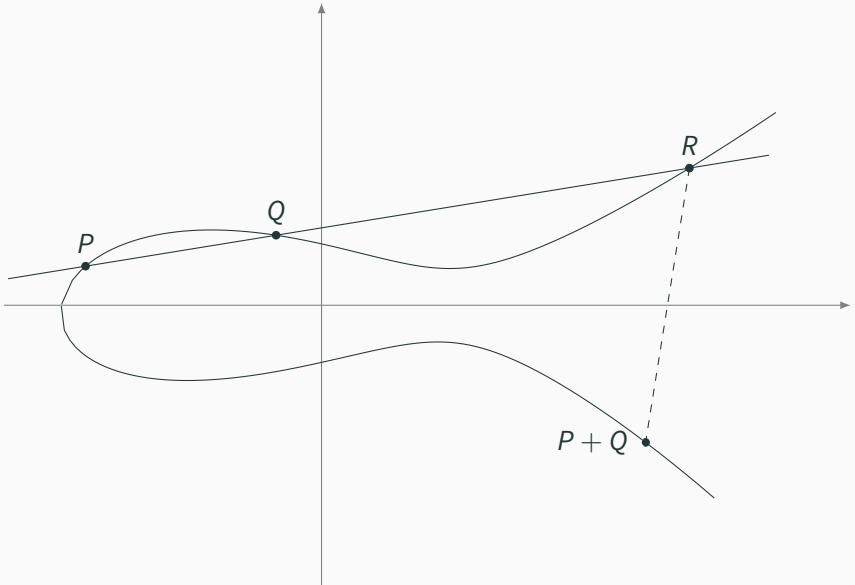
Up to isomorphism



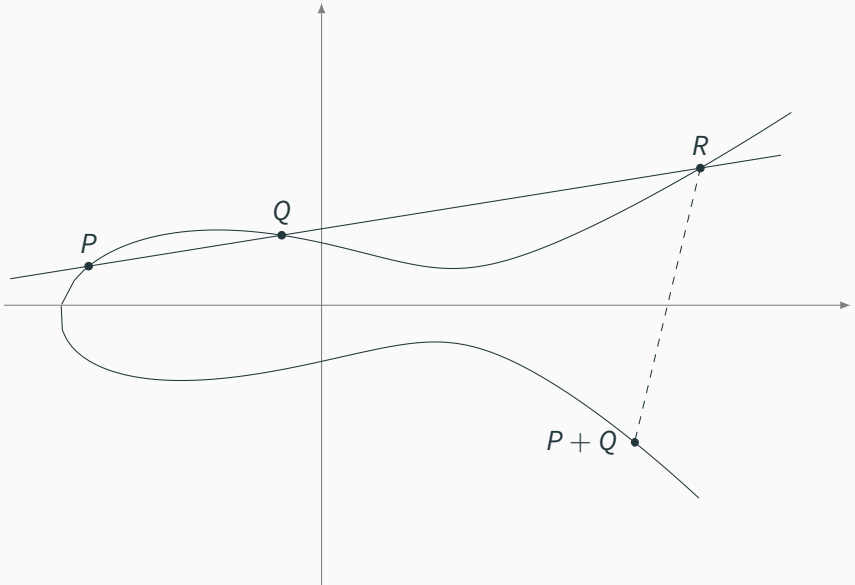
Up to isomorphism



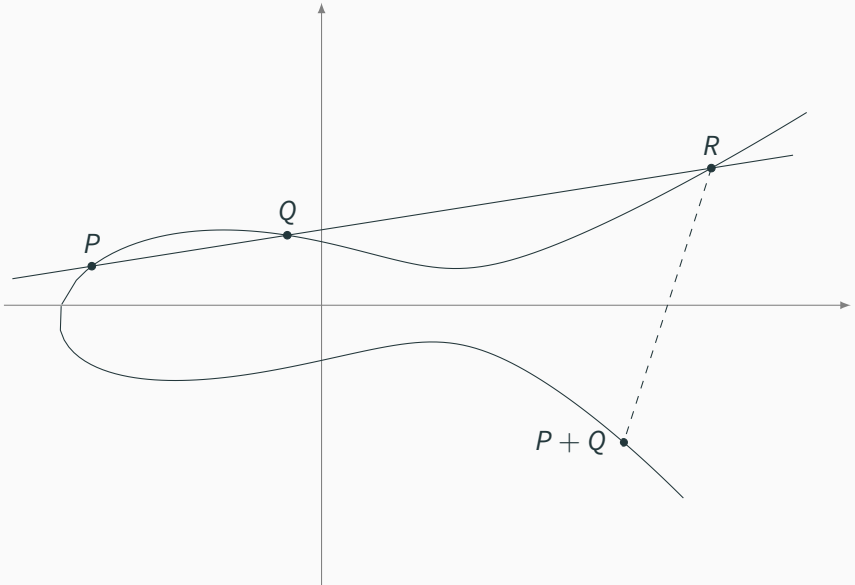
Up to isomorphism



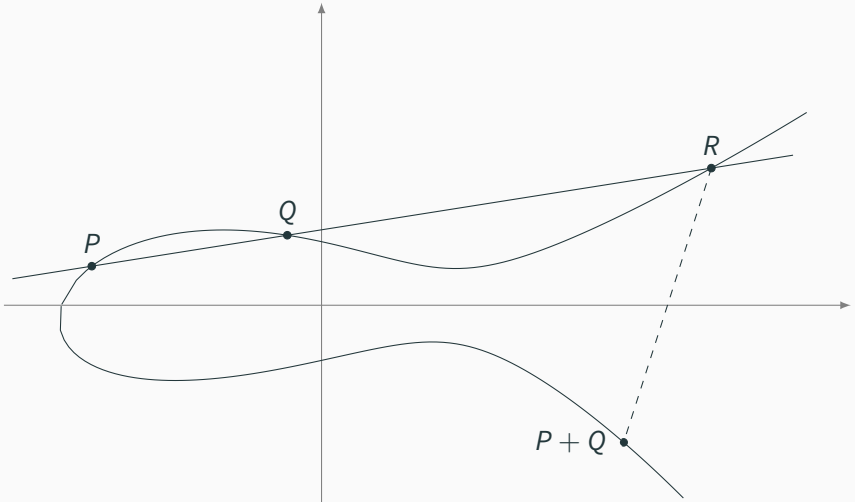
Up to isomorphism



Up to isomorphism

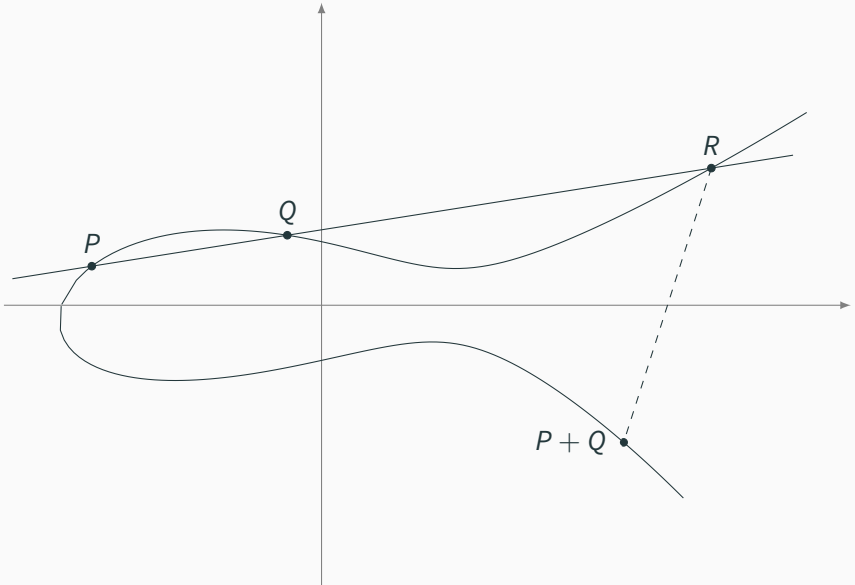


Up to isomorphism

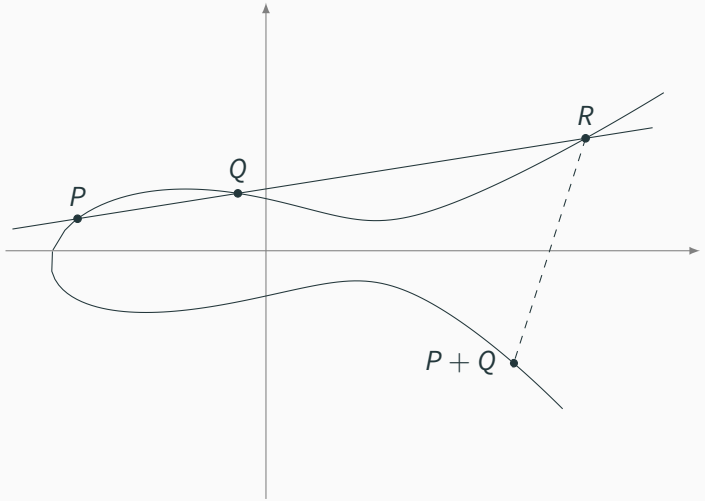


$$y^2 = x^3 + ax + b \quad \longrightarrow \quad j \equiv 1728 \frac{4a^3}{4a^3 + 27b^2}$$

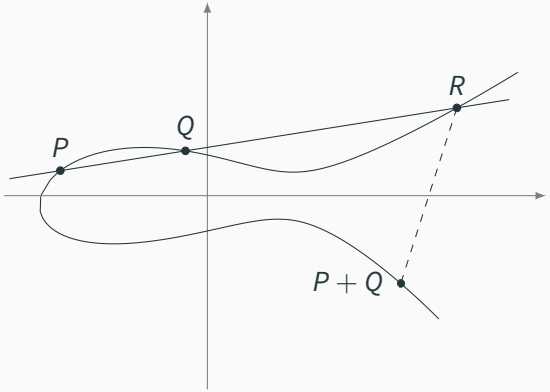
Up to isomorphism



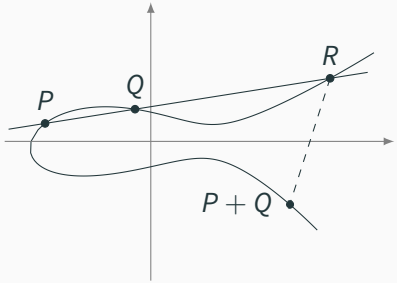
Up to isomorphism



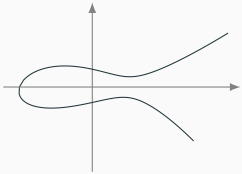
Up to isomorphism



Up to isomorphism



Up to isomorphism

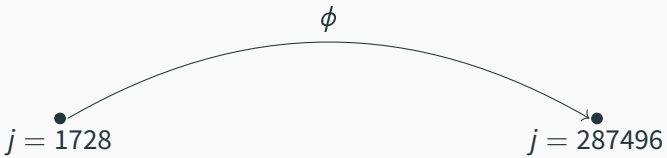


Up to isomorphism

Up to isomorphism

$j = \overset{\bullet}{1}728$

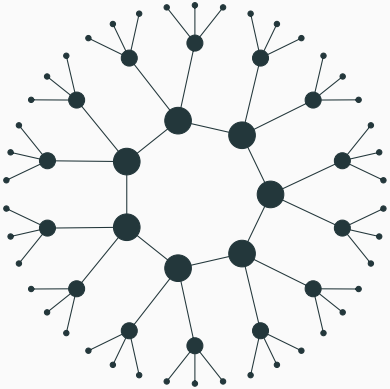
Up to isomorphism



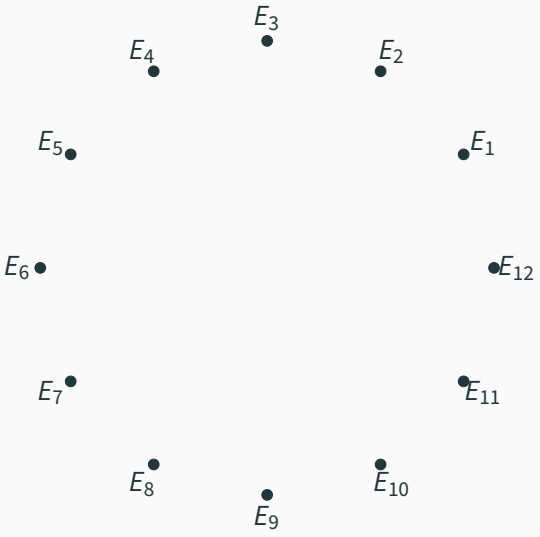
Isogeny graphs

Isogeny graphs

- Vertices are curves up to isomorphism,
- Edges are isogenies up to isomorphism.

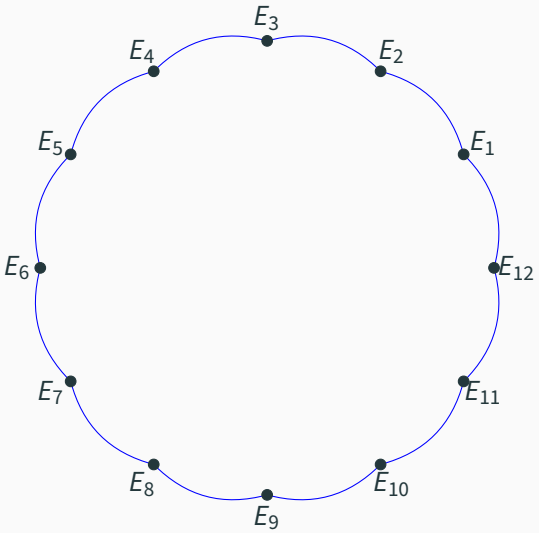


Isogeny graph



Vertices are elliptic curves **with complex multiplication by** $\mathcal{O}_K$ (i.e., $\mathrm{End}(E) \simeq \mathcal{O}_K \subset \mathbb{Q}(\sqrt{-D})$).

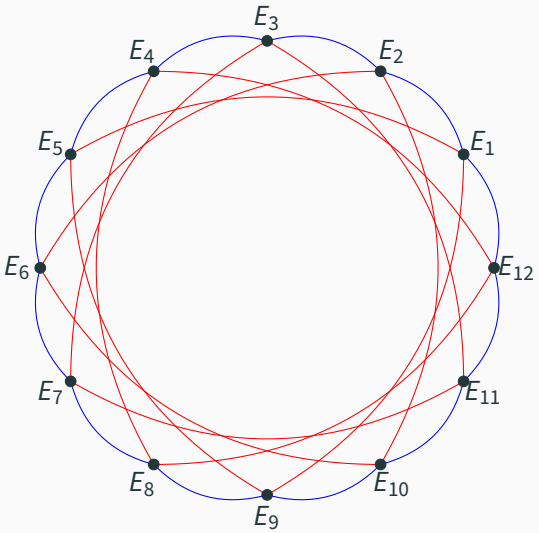
Isogeny graph



Vertices are elliptic curves **with complex multiplication by** $\mathcal{O}_K$ (i.e., $\mathrm{End}(E) \simeq \mathcal{O}_K \subset \mathbb{Q}(\sqrt{-D})$).
Edges are **isogenies** of bounded prime degree.

— degree 2

Isogeny graph

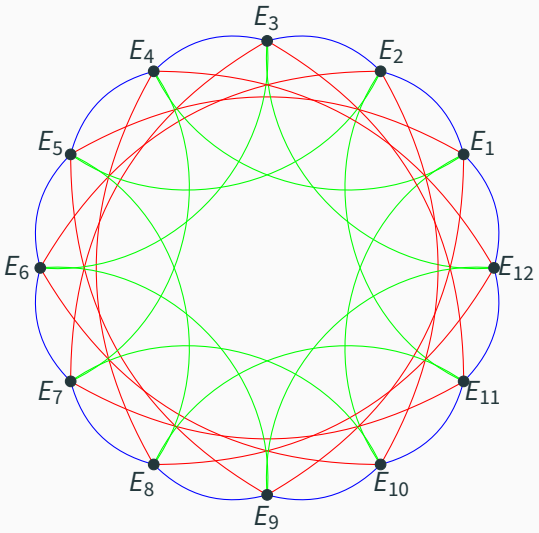


Vertices are elliptic curves **with complex multiplication by** $\mathcal{O}_K$ (i.e., $\mathrm{End}(E) \simeq \mathcal{O}_K \subset \mathbb{Q}(\sqrt{-D})$).
Edges are **isogenies** of bounded prime degree.

— degree 2

— degree 3

Isogeny graph



Vertices are elliptic curves **with complex multiplication by** $\mathcal{O}_K$ (i.e., $\mathrm{End}(E) \simeq \mathcal{O}_K \subset \mathbb{Q}(\sqrt{-D})$).
Edges are **isogenies** of bounded prime degree.

— degree 2

— degree 3

— degree 5

Class group action

Definition
The **class group** of an order $\mathcal{O} \subset \mathbb{Q}(\sqrt{-D})$ is the quotient $\mathrm{Cl}(\mathcal{O}) = \mathcal{I}(\mathcal{O})/\mathcal{P}(\mathcal{O})$. It is a **finite abelian** group.

Theorem (Main theorem of complex multiplication)
The class group of $\mathcal{O}$ acts **faithfully and transitively** on the set of elliptic curves with CM by $\mathcal{O}$ (denoted by $\mathrm{Ell}(\mathcal{O})$) with the **group action** as:

$$\mathrm{Cl}(\mathcal{O}) \times \mathrm{Ell}(\mathcal{O}) \to \mathrm{Ell}(\mathcal{O})$$
$$\mathfrak{a} \star E = E'$$

Class group action

Definition
The **class group** of an order $\mathcal{O} \subset \mathbb{Q}(\sqrt{-D})$ is the quotient $\mathrm{Cl}(\mathcal{O}) = \mathcal{I}(\mathcal{O})/\mathcal{P}(\mathcal{O})$. It is a **finite abelian** group.

Theorem (Main theorem of complex multiplication)
The class group of $\mathcal{O}$ acts **faithfully and transitively** on the set of elliptic curves with CM by $\mathcal{O}$ (denoted by $\mathrm{Ell}(\mathcal{O})$) with the **group action** as:

$$\mathrm{Cl}(\mathcal{O}) \times \mathrm{Ell}(\mathcal{O}) \to \mathrm{Ell}(\mathcal{O})$$
$$\mathfrak{a} \star E = E'$$

Observation:
There is no meaningful operation between two elements of $\mathrm{Ell}(\mathcal{O})$.

Definition (Group Action Inversion Problem (GAIP) [DFG19])
Given two elliptic curves E and E' over the same finite field and with $\mathrm{End}(E) = \mathrm{End}(E') = \mathcal{O}$, find an ideal $\mathfrak{a} \subset \mathcal{O}$ such that $E' = \mathfrak{a} \star E$.

Definition (Group Action Inversion Problem (GAIP) [DFG19])
Given two elliptic curves E and E' over the same finite field and with $\mathrm{End}(E) = \mathrm{End}(E') = \mathcal{O}$, find an ideal $\mathfrak{a} \subset \mathcal{O}$ such that $E' = \mathfrak{a} \star E$.

Definition (Isogeny walk problem)
Given two isogenous elliptic curves E and E' over the same finite field, find a path $E \to E'$ in an isogeny graph.

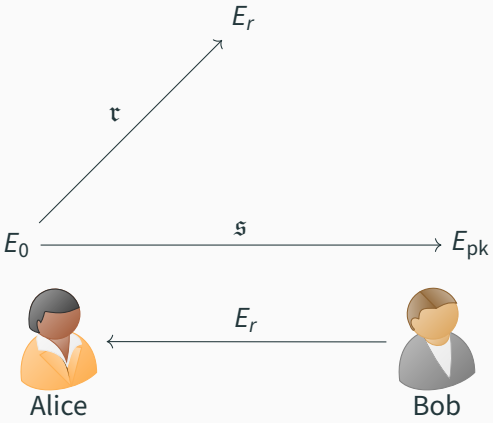
Isogeny-based Adaptor Signature

Identification scheme from isogenies

E_0 is a designated base (starting) curve that is part of public parameters.

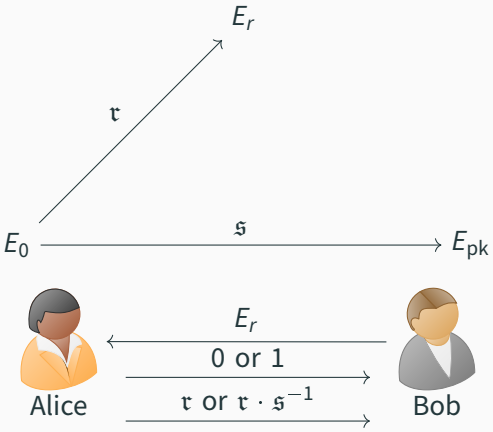
Identification scheme from isogenies

E_0 is a designated base (starting) curve that is part of public parameters.



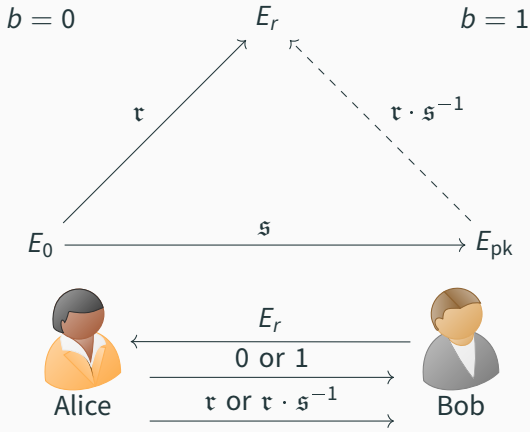
Identification scheme from isogenies

E_0 is a designated base (starting) curve that is part of public parameters.



Identification scheme from isogenies

E_0 is a designated base (starting) curve that is part of public parameters.

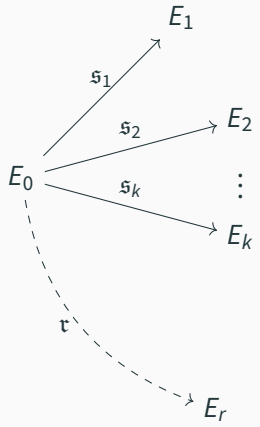


Increasing soundness

- The previous method only has soundness of $\frac{1}{2}$ due to limiting algebraic structure

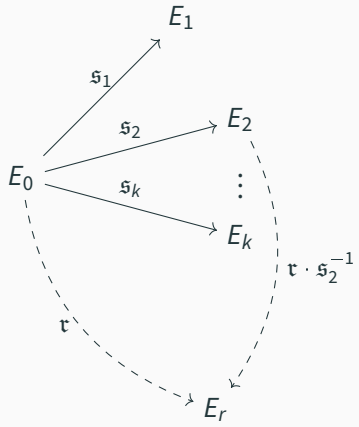
Increasing soundness

- The previous method only has soundness of $\frac{1}{2}$ due to limiting algebraic structure
- We can increase soundness to $\frac{1}{S}$ by using S public keys as described in [DFG19]



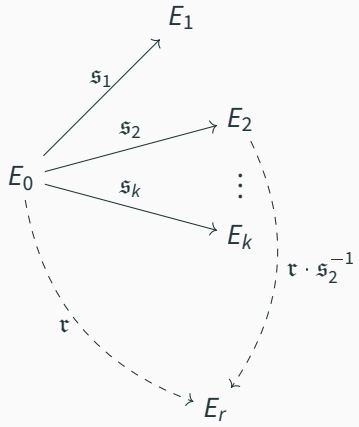
Increasing soundness

- The previous method only has soundness of $\frac{1}{2}$ due to limiting algebraic structure
- We can increase soundness to $\frac{1}{S}$ by using S public keys as described in [DFG19]



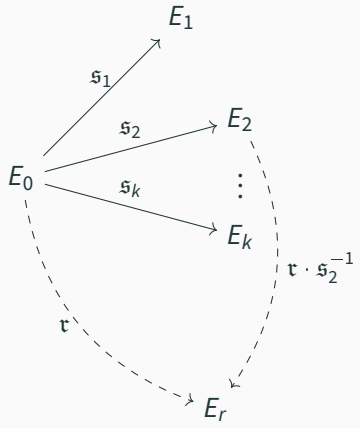
Increasing soundness

- The previous method only has soundness of $\frac{1}{2}$ due to limiting algebraic structure
- We can increase soundness to $\frac{1}{S}$ by using S public keys as described in [DFG19]
- Using quadratic twists of the curves as in [BKV19], we can further reduce the cheating probability to $\frac{1}{2S-1}$



Increasing soundness

- The previous method only has soundness of $\frac{1}{2}$ due to limiting algebraic structure
- We can increase soundness to $\frac{1}{S}$ by using S public keys as described in [DFG19]
- Using quadratic twists of the curves as in [BKV19], we can further reduce the cheating probability to $\frac{1}{2S-1}$
- To achieve security level λ, we require $t = \frac{\lambda}{\log_2 S}$ iterations.



Isogeny Adaptor Signature (IAS)

- Combining the previous approaches with Fiat-Shamir transform we obtain the isogeny-based signature scheme CSI-FiSh [BKV19]
- We can construct an adaptor signature from CSI-FiSh using GAIP as the hard relation (for simplicity we consider the base scheme with challenge space $\{0, 1\}$).

Isogeny Adaptor Signature (IAS)

- Combining the previous approaches with Fiat-Shamir transform we obtain the isogeny-based signature scheme CSI-FiSh [BKV19]
- We can construct an adaptor signature from CSI-FiSh using GAIP as the hard relation (for simplicity we consider the base scheme with challenge space $\{0, 1\}$).

Schnorr

procedure PreSig(sk, m, Y)

$k \leftarrow_\$ \mathbb{Z}_p, R := g^k$

$e := H(\mathsf{pk} \| R \cdot Y \| m)$

$\hat{s} := k + e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{s})$

IAS [TMSM20]

procedure PreSig(sk, m, E_Y)

$\mathfrak{k} \leftarrow_\$ \mathrm{Cl}(\mathcal{O}), E_R := \mathfrak{k} \star E_0$

$e := H(\mathsf{pk} \| E_R \cdot E_Y \| m)$

$\hat{\mathfrak{s}} := \mathfrak{k} - e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{\mathfrak{s}})$

Isogeny Adaptor Signature (IAS)

- Combining the previous approaches with Fiat-Shamir transform we obtain the isogeny-based signature scheme CSI-FiSh [BKV19]
- We can construct an adaptor signature from CSI-FiSh using GAIP as the hard relation (for simplicity we consider the base scheme with challenge space $\{0, 1\}$).

Schnorr

procedure PreSig(sk, m, Y)

$k \leftarrow_\$ \mathbb{Z}_p, R := g^k$

$e := H(\mathsf{pk} \| R \cdot Y \| m)$

$\hat{s} := k + e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{s})$

IAS [TMSM20]

procedure PreSig(sk, m, E_Y)

$\mathfrak{k} \leftarrow_\$ \mathrm{Cl}(\mathcal{O}), E_R := \mathfrak{k} \star E_0$

$e := H(\mathsf{pk} \| E_R \cdot E_Y \| m)$

$\hat{\mathfrak{s}} := \mathfrak{k} - e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{\mathfrak{s}})$

Observation:
We cannot combine E_R and E_Y as there is no meaningful operation between two elliptic curves from $\mathrm{Ell}(\mathcal{O})$

Solution:
Randomize the statement E_Y with the group action of E_R and prove the relation between E_R and E_Y in ZK (i.e., a DH-tuple proof for isogenies) [TMSM20].

Solution:
Randomize the statement E_Y with the group action of E_R and prove the relation between E_R and E_Y in ZK (i.e., a DH-tuple proof for isogenies) [TMSM20].

procedure PreSig(sk, m, E_Y)

$\mathfrak{k} \leftarrow\!\!\$\ \mathsf{Cl}(\mathcal{O})$; $E_R := \mathfrak{k} \star E_0$; $\hat{E}_R := \mathfrak{k} \star E_Y$

Set $x := (E_0, E_R, E_Y, \hat{E}_R)$

$\pi \leftarrow \mathsf{P}_{\mathsf{NIZK}}(x, \mathfrak{k})$

$e := H(\mathsf{pk} \| \hat{E}_R \| m)$

$\hat{\mathfrak{s}} := \mathfrak{k} - e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{\mathfrak{s}}, \hat{E}_R, \pi)$

Solution:
Randomize the statement E_Y with the group action of E_R and prove the relation between E_R and E_Y in ZK (i.e., a DH-tuple proof for isogenies) [TMSM20].

procedure PreSig(sk, m, E_Y)

$\mathfrak{k} \leftarrow\!\!\$\ \mathrm{Cl}(\mathcal{O})$; $E_R := \mathfrak{k} \star E_0$; $\hat{E}_R := \mathfrak{k} \star E_Y$

Set $x := (E_0, E_R, E_Y, \hat{E}_R)$

$\pi \leftarrow \mathsf{P}_{\mathsf{NIZK}}(x, \mathfrak{k})$

$e := H(\mathsf{pk} \| \hat{E}_R \| m)$

$\hat{\mathfrak{s}} := \mathfrak{k} - e \cdot \mathsf{sk}$

return $\hat{\sigma} := (e, \hat{\mathfrak{s}}, \hat{E}_R, \pi)$

procedure PreVer(pk, m, E_Y, $\hat{\sigma}$)

Parse pk as (E_0, E_1)

Parse $\hat{\sigma}$ as $(e, \hat{\mathfrak{s}}, \hat{E}_R, \pi)$

$E_R := \hat{\mathfrak{s}} \star \mathsf{pk}_e$

Set $x := (E_0, E_R, E_Y, \hat{E}_R)$

if $\pi \leftarrow \mathsf{V}_{\mathsf{NIZK}}(x, \pi) \neq 1$ **then**

return 0

$e' = H(\mathsf{pk} \| \hat{E}_R \| m)$

return $(e = e')$

Solution:
Randomize the statement E_Y with the group action of E_R and prove the relation between E_R and E_Y in ZK (i.e., a DH-tuple proof for isogenies) [TMSM20].

procedure PreSig(sk, m, E_Y)
- $\mathfrak{k} \leftarrow_{\$} \mathrm{Cl}(\mathcal{O})$; $E_R := \mathfrak{k} \star E_0$; $\hat{E}_R := \mathfrak{k} \star E_Y$
- Set $x := (E_0, E_R, E_Y, \hat{E}_R)$
- $\pi \leftarrow \mathsf{P}_{\mathsf{NIZK}}(x, \mathfrak{k})$
- $e := H(\mathsf{pk} \| \hat{E}_R \| m)$
- $\hat{\mathfrak{s}} := \mathfrak{k} - e \cdot \mathsf{sk}$
- **return** $\hat{\sigma} := (e, \hat{\mathfrak{s}}, \hat{E}_R, \pi)$

procedure PreVer(pk, m, E_Y, $\hat{\sigma}$)
- Parse pk as (E_0, E_1)
- Parse $\hat{\sigma}$ as $(e, \hat{\mathfrak{s}}, \hat{E}_R, \pi)$
- $E_R := \hat{\mathfrak{s}} \star \mathsf{pk}_e$
- Set $x := (E_0, E_R, E_Y, \hat{E}_R)$
- **if** $\pi \leftarrow \mathsf{V}_{\mathsf{NIZK}}(x, \pi) \neq 1$ **then**
  - **return** 0
- $e' = H(\mathsf{pk} \| \hat{E}_R \| m)$
- **return** $(e = e')$

Adapt and Ext algorithms are analogous to Schnorr-based construction.

Zero-knowledge proof for group action

Cozzo and Smart [CS20] showed how to prove knowledge of a secret isogeny generically. More precisely, they showed a zero-knowledge proof for the following relation (i.e., knowledge of a witness $\mathfrak{s}$ for j simultaneous instances of the GAIP):

$$L_j := \left\{ \left((E_1, E_1', \ldots, E_j, E_j'), \mathfrak{s} \right) : \bigwedge_{i=1}^{j} (E_i' = \mathfrak{s} \star E_i) \right\}.$$

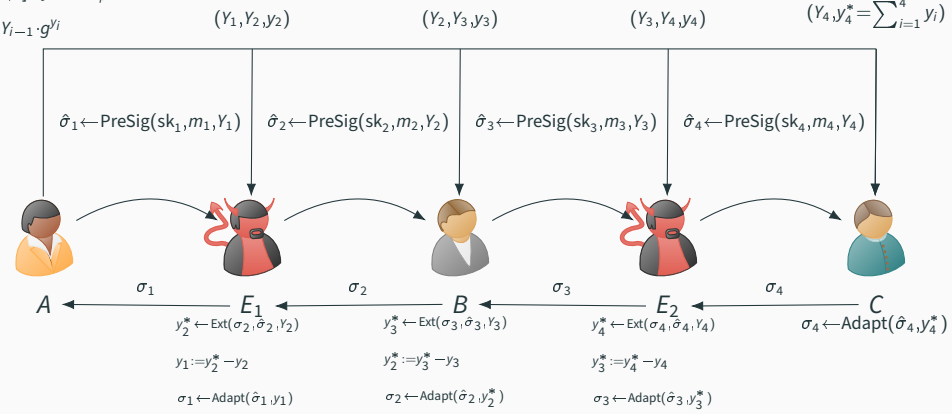
The proof simply involves running multiple iterations of the basic identification scheme to amplify the soundness. We note that we only need this for L_2.

IAS-based PCN

$y_1 \leftarrow\$ \mathbb{Z}_p$; $Y_1 := g^{y_1}$

for $i \in [2,4]$: $y_i \leftarrow\$ \mathbb{Z}_p$

$Y_i := Y_{i-1} \cdot g^{y_i}$

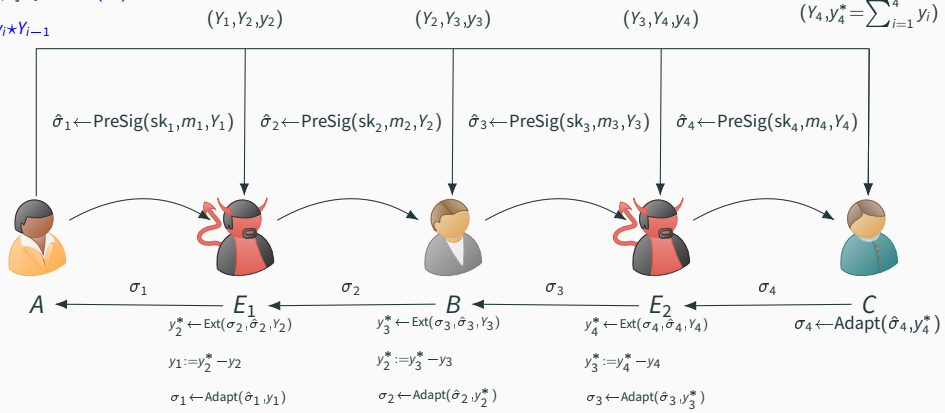


IAS-based PCN

$y_1 \leftarrow\$ \mathrm{Cl}(\mathcal{O})$; $Y_1 := y_1 \star E_0$

for $i \in [2,4]$: $y_i \leftarrow\$ \mathrm{Cl}(\mathcal{O})$

$Y_i := y_i \star Y_{i-1}$



Performance Evaluation

- C implementation, parallelized with OpenMP. Benchmarked on a KVM-based VM with 2.0GHz AMD EPYC 7702 processor with 16 cores and 32GB RAM, running Ubuntu 18.04 LTS
- Source code: `https://github.com/etairi/Adaptor-CSI-FiSh`

S	t_S	k	$\lvert sk \rvert$	$\lvert pk \rvert$	$\lvert \hat{\sigma} \rvert$	$\lvert \sigma \rvert$	KGen	Sig	Ver	PreSig	PreVer	Ext	Adapt
2^1	56	16	16	128	19944	1880	0.05	0.24	0.23	3.59	3.55	0.005	0.005
2^2	38	14	16	256	19672	1286	0.06	0.16	0.16	2.75	2.68	0.005	0.005
2^3	28	16	16	512	19020	956	0.07	0.13	0.14	2.21	2.15	0.005	0.005
2^4	23	13	16	1024	19338	791	0.07	0.11	0.11	1.99	1.94	0.005	0.005
2^6	16	16	16	4096	18624	560	0.29	0.08	0.09	1.61	1.56	0.005	0.005
2^8	13	11	16	16384	19330	461	1.00	0.08	0.08	1.50	1.44	0.005	0.005
2^{10}	11	7	16	65536	19908	395	3.21	0.06	0.06	1.40	1.36	0.005	0.005
2^{12}	9	11	16	262144	19198	329	12.89	0.06	0.06	1.30	1.25	0.005	0.005
2^{15}	7	16	16	2097152	18327	263	102.02	0.06	0.06	1.16	1.11	0.005	0.005

Open Problems

- Can we construct adaptor signatures from other post-quantum assumptions (e.g., multivariate quadratics, code-based, hash-based)?
- Can we transform SQISign [DFKL$^+$20] into an adaptor signature to obtain a more efficient isogeny-based construction?


Thank you!
@erkantairi

References i

- Lukas Aumayr, Oguzhan Ersoy, Andreas Erwig, Sebastian Faust, Kristina Hostakova, Matteo Maffei, Pedro Moreno-Sanchez, and Siavash Riahi, Generalized bitcoin-compatible channels, Cryptology ePrint Archive, Report 2020/476, 2020, `https://eprint.iacr.org/2020/476`.
- Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren, Csi-fish: Efficient isogeny based signatures through class group computations, ASIACRYPT, 2019.
- Daniele Cozzo and Nigel P. Smart, Sashimi: Cutting up csi-fish secret keys to produce an actively secure distributed signing protocol, PQCrypto, 2020.
- Luca De Feo and Steven D. Galbraith, Seasign: Compact isogeny signatures from class group actions, EUROCRYPT (Yuval Ishai and Vincent Rijmen, eds.), 2019.

References ii

- Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, Sqisign: Compact post-quantum signatures from quaternions and isogenies, Advances in Cryptology – ASIACRYPT 2020 (Cham) (Shiho Moriai and Huaxiong Wang, eds.), Springer International Publishing, 2020, pp. 64–93.
- Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé, Crystals-dilithium: A lattice-based digital signature scheme, IACR Transactions on Cryptographic Hardware and Embedded Systems **2018** (2018), no. 1, 238–268.
- Muhammed F. Esgin, Oguzhan Ersoy, and Zekeriya Erkin, Post-quantum adaptor signatures and payment channel networks, Cryptology ePrint Archive, Report 2020/845, 2020, `https://eprint.iacr.org/2020/845`.

References iii

- Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler, Practical exact proofs from lattices: New techniques to exploit fully-splitting rings.
- Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei, Anonymous multi-hop locks for blockchain scalability and interoperability, 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019, 2019.
- Erkan Tairi, Pedro Moreno-Sanchez, and Matteo Maffei, Post-quantum adaptor signature for privacy-preserving off-chain payments, Cryptology ePrint Archive, Report 2020/1345, 2020, `https://eprint.iacr.org/2020/1345`.