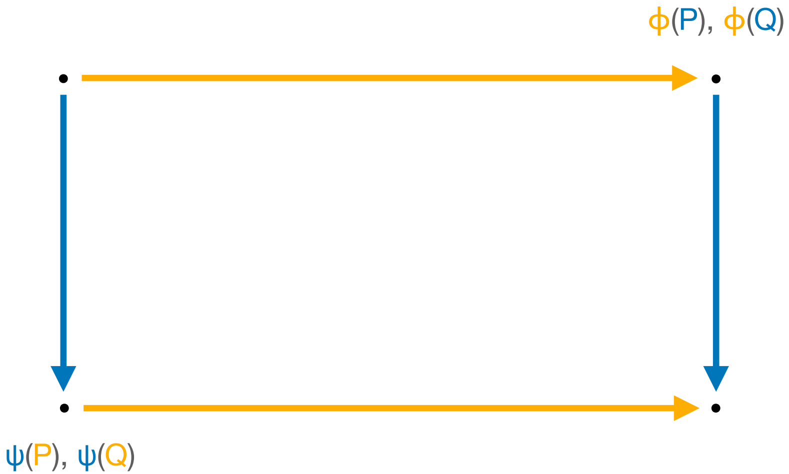
# FESTA: Fast Encryption from Supersingular Torsion Attacks

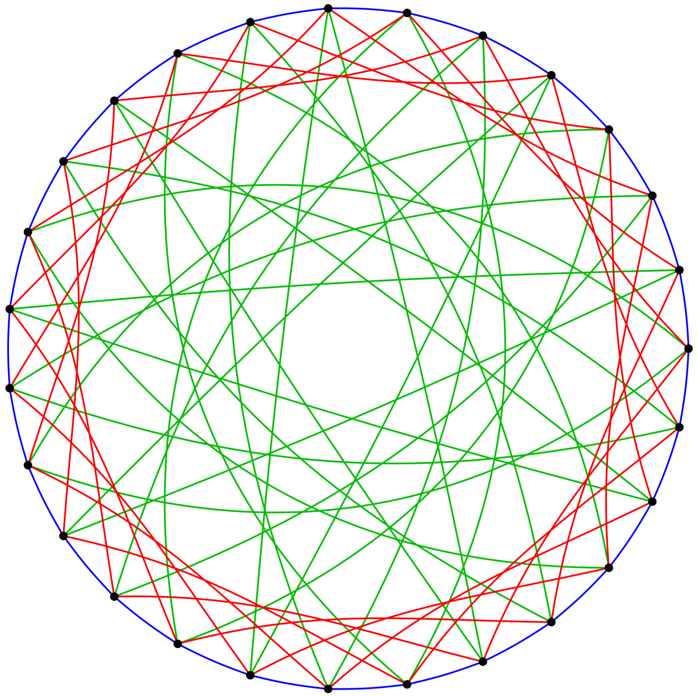
Andrea Basso, Luciano Maino, Giacomo Pope

ASIACRYPT 2023 — December 8th, 2023

# Isogeny-based encryption



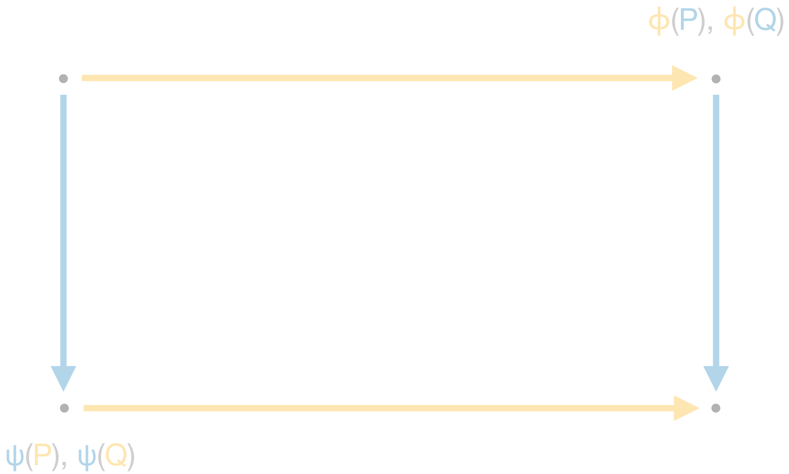

**SIDH**



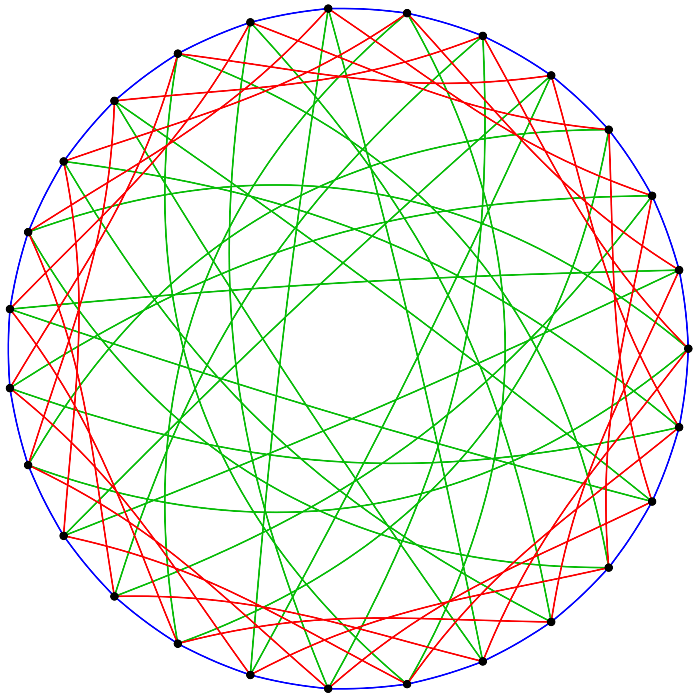
**CSIDH**

The CSIDH image is based on Fig. 1 from *CSIDH: An Efficient Post-Quantum Commutative Group Action*

# Isogeny-based encryption




SIDH



CSIDH

The CSIDH image is based on Fig. 1 from *CSIDH: An Efficient Post-Quantum Commutative Group Action*

# The attacks on SIDH

$E_0, E_1$
$\deg \phi$
$P, Q, \phi(P), \phi(Q)$

SIDH attacks

$\phi : E_0 \to E_1$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \;\bullet \xrightarrow{\quad \phi \quad} \bullet\; \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} \phi(P_0) \\ \phi(Q_0) \end{matrix}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\bullet \xrightarrow{\phi} \bullet$$

$$\begin{array}{l} P_0 \\ Q_0 \end{array} \qquad \begin{array}{l} P_1 \\ Q_1 \end{array} = \begin{array}{l} [\alpha]\ \phi(P_0) \\ [\beta]\ \phi(Q_0) \end{array}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \bullet \xrightarrow{\qquad\qquad \phi \qquad\qquad} \bullet \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\phi(Q_0) \end{matrix}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\quad\phi\quad} \bullet \quad \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix}$$

**An important property**

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \xrightarrow{\phi} \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix}$$

**An important property**

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \xrightarrow{\phi} \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{array}{c} P_0 \\ Q_0 \end{array} \bullet \xrightarrow{\quad\phi\quad} \bullet \begin{array}{c} P_1 \\ Q_1 \end{array} = \begin{array}{c} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{array}$$

**An important property**

$$\begin{array}{c} P_0 \\ Q_0 \end{array} \bullet \xrightarrow{\quad\phi\quad} \bullet \begin{array}{c} P_1 \\ Q_1 \end{array} = \begin{array}{c} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{array} \xrightarrow[\quad\psi\quad]{} \bullet \begin{array}{c} P_2 \\ Q_2 \end{array} = \begin{array}{c} [\beta]\ \psi(P_1) \\ [\beta^{-1}]\ \psi(Q_1) \end{array}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\quad \phi \quad} \bullet \quad \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{matrix}$$

**An important property**

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\quad \phi \quad} \bullet \quad \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{matrix} \qquad \bullet \xrightarrow[\quad \psi \quad]{} \bullet \quad \begin{matrix} P_2 \\ Q_2 \end{matrix} = \begin{matrix} [\beta]\ \psi(P_1) \\ [\beta^{-1}]\ \psi(Q_1) \end{matrix} = \begin{matrix} [\alpha\beta]\ \psi\phi(P_0) \\ [(\alpha\beta)^{-1}]\ \psi\phi(Q_0) \end{matrix}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \xrightarrow{\quad\phi\quad} \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix}$$

**An important property**

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \xrightarrow{\quad\phi\quad} \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix} \xrightarrow{\quad\psi\quad} \begin{matrix} P_2 \\ Q_2 \end{matrix} = \begin{matrix} [\beta]\,\psi(P_1) \\ [\beta^{-1}]\,\psi(Q_1) \end{matrix} = \begin{matrix} [\alpha\beta]\,\psi\phi(P_0) \\ [(\alpha\beta)^{-1}]\,\psi\phi(Q_0) \end{matrix} \xrightarrow{\quad\sigma\quad} \begin{matrix} [\alpha^{-1}]\,\sigma(P_2) \\ [\alpha]\,\sigma(Q_2) \end{matrix}$$

# A new assumption

## Scaling torsion points prevents attacks

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\ \phi\ } \bullet \quad \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix}$$

**An important property**

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\ \phi\ } \bullet \xrightarrow[\ \psi\ ]{} \bullet \xrightarrow[\ \sigma\ ]{} \bullet$$

$$\begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\alpha]\,\phi(P_0) \\ [\alpha^{-1}]\,\phi(Q_0) \end{matrix}$$

$$\begin{matrix} P_2 \\ Q_2 \end{matrix} = \begin{matrix} [\beta]\,\psi(P_1) \\ [\beta^{-1}]\,\psi(Q_1) \end{matrix} = \begin{matrix} [\alpha\beta]\,\psi\phi(P_0) \\ [(\alpha\beta)^{-1}]\,\psi\phi(Q_0) \end{matrix}$$

$$\begin{matrix} [\alpha^{-1}]\,\sigma(P_2) \\ [\alpha]\,\sigma(Q_2) \end{matrix} = \begin{matrix} [\beta]\,\sigma\psi\phi(P_0) \\ [\beta^{-1}]\,\sigma\psi\phi(Q_0) \end{matrix}$$

# The FESTA trapdoor

# The FESTA trapdoor

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\quad \phi \quad} \bullet \quad \begin{matrix} P_A \\ Q_A \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{matrix}$$

# The FESTA trapdoor

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\phi} \bullet \quad \begin{matrix} P_A \\ Q_A \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{matrix}$$

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}($$

# The FESTA trapdoor

$$\begin{matrix} P_0 \\ Q_0 \end{matrix} \quad \bullet \xrightarrow{\quad\phi\quad} \bullet \quad \begin{matrix} P_A \\ Q_A \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{matrix}$$

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\psi, \sigma, \beta)$$

# The FESTA trapdoor

$$\begin{tikzcd} \begin{matrix} P_0 \\ Q_0 \end{matrix} \bullet \arrow[r, "\phi"] \arrow[d, "\psi"'] & \bullet \; \begin{matrix} P_A \\ Q_A \end{matrix} = \begin{matrix} [\alpha] \; \phi(P_0) \\ [\alpha^{-1}] \; \phi(Q_0) \end{matrix} \\ \bullet \; \begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\beta] \; \psi(P_0) \\ [\beta^{-1}] \; \psi(Q_0) \end{matrix} & \end{tikzcd}$$

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\psi, \sigma, \beta)$$

# The FESTA trapdoor

$$\begin{tikzcd} \begin{matrix}P_0\\Q_0\end{matrix} \arrow[r, "\phi"] \arrow[d, "\psi"'] & \begin{matrix}P_A\\Q_A\end{matrix} \arrow[d, "\sigma"] \\ \begin{matrix}P_1\\Q_1\end{matrix} & \begin{matrix}P_2\\Q_2\end{matrix} \end{tikzcd}$$

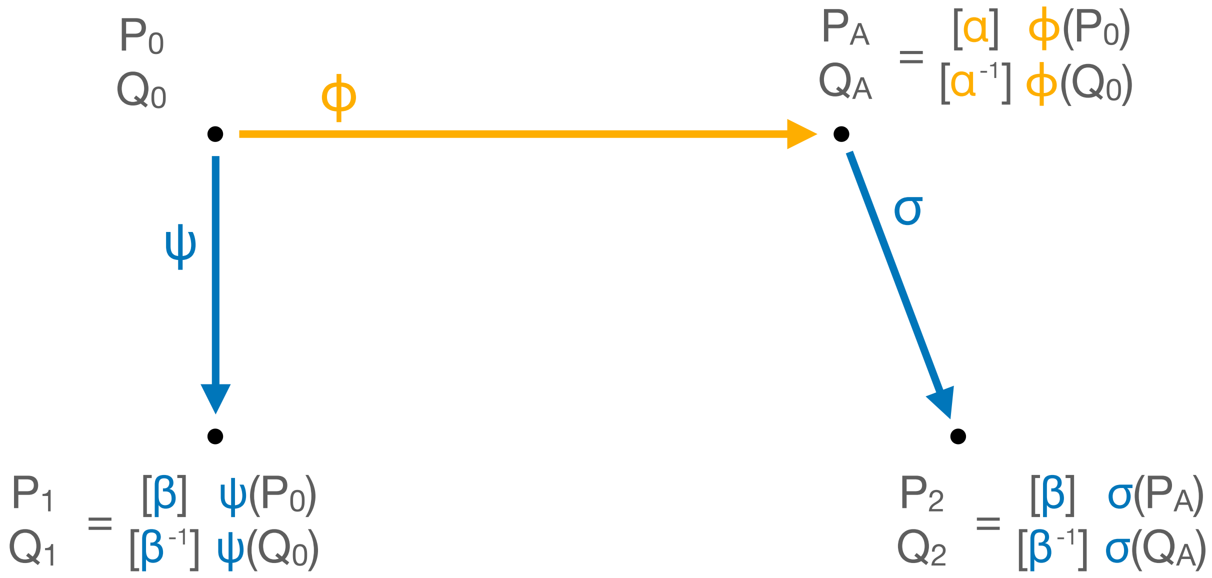
$$\begin{matrix}P_A\\Q_A\end{matrix} = \begin{matrix}[\alpha]\ \phi(P_0)\\ [\alpha^{-1}]\ \phi(Q_0)\end{matrix}$$

$$\begin{matrix}P_1\\Q_1\end{matrix} = \begin{matrix}[\beta]\ \psi(P_0)\\ [\beta^{-1}]\ \psi(Q_0)\end{matrix}$$

$$\begin{matrix}P_2\\Q_2\end{matrix} = \begin{matrix}[\beta]\ \sigma(P_A)\\ [\beta^{-1}]\ \sigma(Q_A)\end{matrix}$$

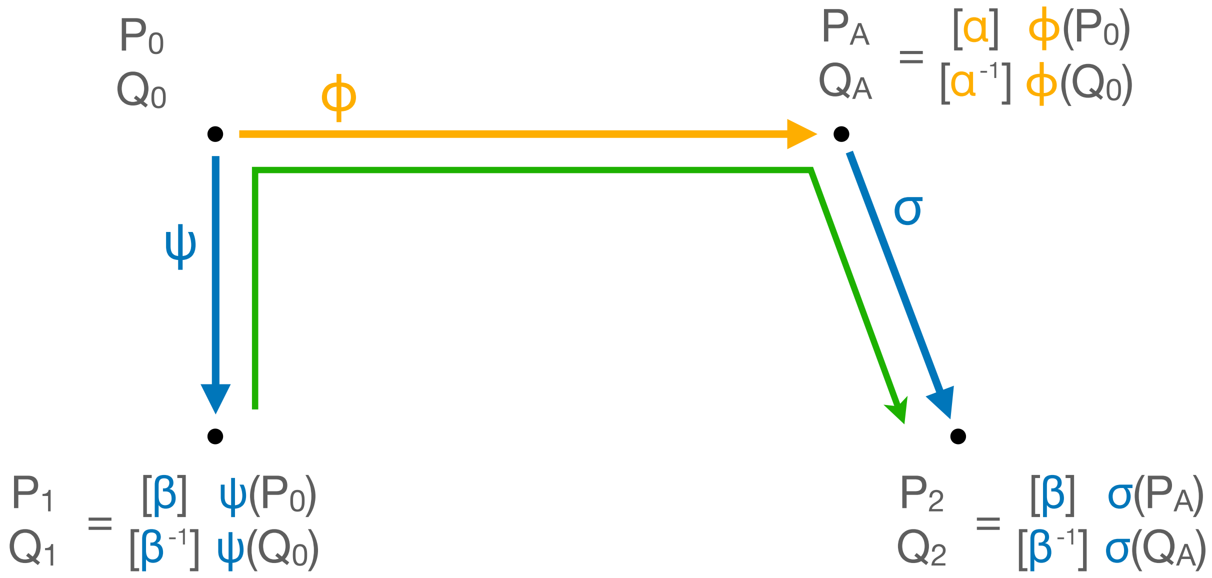
$$f_{E_0,\ P_0,\ Q_0,\ E_A,\ P_A,\ Q_A}(\psi, \sigma, \beta)$$

# The FESTA trapdoor




$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\psi, \sigma, \beta) = E_1, P_1, Q_1, E_2, P_2, Q_2$$

# The FESTA trapdoor




$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\psi, \sigma, \beta) = E_1, P_1, Q_1, E_2, P_2, Q_2$$

# The FESTA trapdoor

$$\begin{tikzcd} \bullet \arrow[r, "\phi"] \arrow[d, "\psi"'] & \bullet \arrow[d, "\sigma"] \\ \bullet & \bullet \end{tikzcd}$$

$P_0$, $Q_0$

$$\begin{matrix} P_A \\ Q_A \end{matrix} = \begin{matrix} [\alpha]\ \phi(P_0) \\ [\alpha^{-1}]\ \phi(Q_0) \end{matrix}$$

$$\begin{matrix} P_1 \\ Q_1 \end{matrix} = \begin{matrix} [\beta]\ \psi(P_0) \\ [\beta^{-1}]\ \psi(Q_0) \end{matrix}$$

$$\begin{matrix} P_2 \\ Q_2 \end{matrix} = \begin{matrix} [\beta]\ \sigma(P_A) \\ [\beta^{-1}]\ \sigma(Q_A) \end{matrix}$$

$$\Rightarrow \begin{matrix} \hat{\psi}\phi\sigma(P_1) = [\deg\psi][\alpha^{-1}]P_2 \\ \hat{\psi}\phi\sigma(Q_1) = [\deg\psi][\alpha]Q_2 \end{matrix}$$

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\psi, \sigma, \beta) = E_1, P_1, Q_1, E_2, P_2, Q_2$$

# The FESTA PKE

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\psi, \sigma, \beta) \;=\; E_1, P_1, Q_1, E_2, P_2, Q_2$$

# The FESTA PKE

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\underbrace{\psi, \sigma, \beta}_{\text{partial-domain one-way}}) \;=\; E_1,\, P_1,\, Q_1,\, E_2,\, P_2,\, Q_2$$

# The FESTA PKE

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\underbrace{\psi, \sigma, \beta}_{\text{partial-domain one-way}}) = E_1, P_1, Q_1, E_2, P_2, Q_2$$

$$\Downarrow$$

**OAEP transform**

IND-CCA2 security in the QROM

# The FESTA PKE

$$f_{E_0,\ P_0,\ Q_0,\ E_A,\ P_A,\ Q_A}(\underbrace{\psi, \sigma, \beta}_{\text{partial-domain one-way}}) = E_1,\ P_1,\ Q_1,\ E_2,\ P_2,\ Q_2$$

$$\Downarrow$$

**OAEP transform**

IND-CCA2 security in the QROM

**Encrypt**

1. Sample random rnd
2. $\psi = (m \,||\, 0\ldots0) + H(\text{rnd})$
3. $\sigma,\ \beta = G(\psi) + \text{rnd}$
4. $\text{ct} = f(\psi,\sigma,\beta)$

# The FESTA PKE

$$f_{E_0,\, P_0,\, Q_0,\, E_A,\, P_A,\, Q_A}(\underbrace{\psi, \sigma, \beta}_{\text{partial-domain one-way}}) = E_1, P_1, Q_1, E_2, P_2, Q_2$$

$$\Downarrow$$

**OAEP transform**

IND-CCA2 security in the QROM

**Encrypt**

1. Sample random rnd
2. $\psi = (m \,||\, 0\ldots0) + H(\text{rnd})$
3. $\sigma, \beta = G(\psi) + \text{rnd}$
4. $\text{ct} = f(\psi, \sigma, \beta)$

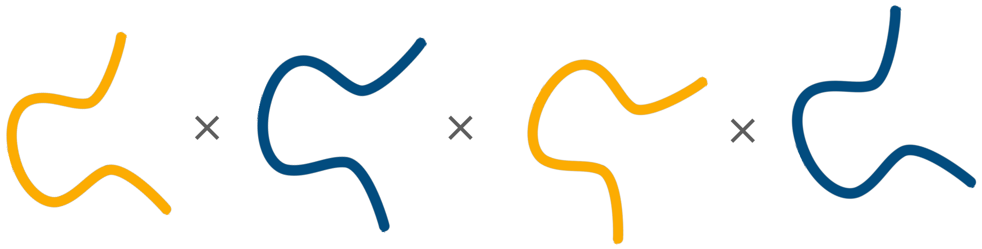
**Decrypt**

1. Compute $\psi, \sigma, \beta$
2. $\text{rnd} = G(\psi) - (\sigma, \beta)$
3. $(m \,||\, 0\ldots0) = \psi - H(\text{rnd})$

# There are attacks and attacks

Dimension two

Dimension four (and higher)

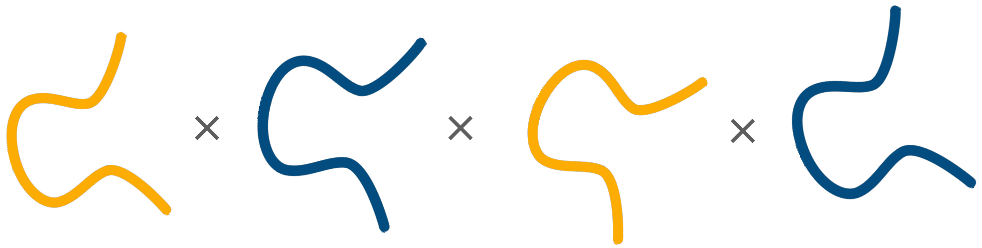


# There are attacks and attacks

**Dimension two**

- Fast and simple implementation
- Strict degree requirements

**Dimension four (and higher)**

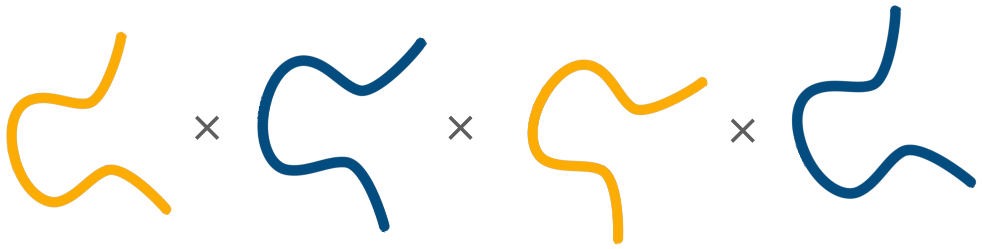


# There are attacks and attacks

**Dimension two**

- Fast and simple implementation
- Strict degree requirements

**Dimension four (and higher)**



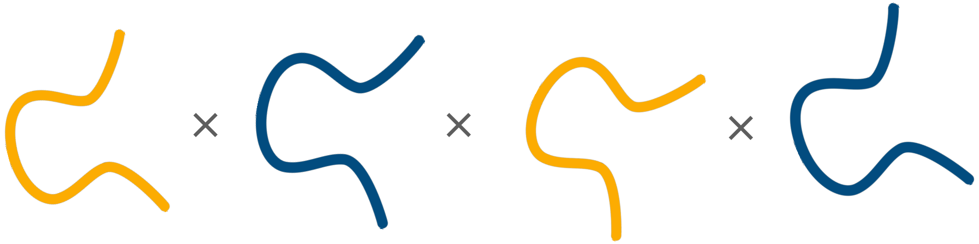
- No degree requirements
- Slow and complex implementation

# There are attacks and attacks

### Dimension two

- Fast and simple implementation
- Strict degree requirements

### Dimension four (and higher)



- No degree requirements
- Slow and complex implementation

$\Downarrow$

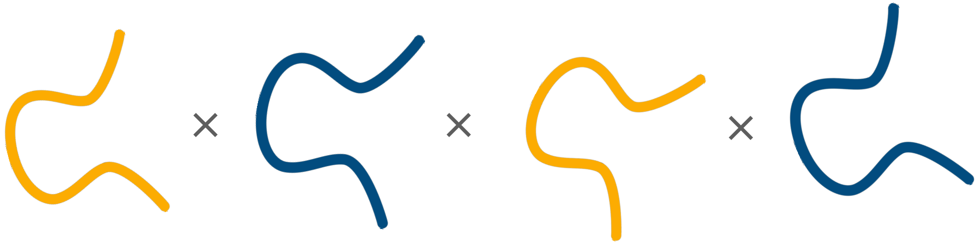
- Small parameters ($p \approx 2^{400}$)
- Fast KeyGen and Encrypt, slow Decrypt

# There are attacks and attacks

**Dimension two**

- Fast and simple implementation
- Strict degree requirements

**Dimension four (and higher)**



- No degree requirements
- Slow and complex implementation

$\Downarrow$

- Small parameters ($p \approx 2^{400}$)
- Fast KeyGen and Encrypt, slow Decrypt

# There are attacks and attacks

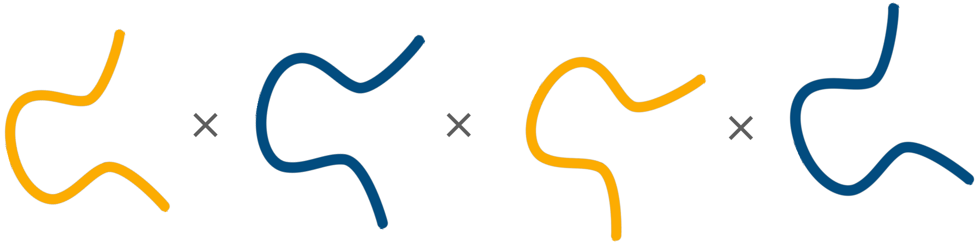
### Dimension two

- Fast and simple implementation
- Strict degree requirements

$\Downarrow$

- Larger parameters ($p \approx 2^{1300}$)
- Practical running times

### Dimension four (and higher)



- No degree requirements
- Slow and complex implementation

$\Downarrow$

- Small parameters ($p \approx 2^{400}$)
- Fast KeyGen and Encrypt, slow Decrypt

# How to find parameters

1

Running times of the attack dominated by the smoothness of the order of torsion points

# How to find parameters

1 Running times of the attack dominated by the smoothness of the order of torsion points $\Rightarrow$ $\text{ord } P = \text{ord } Q = 2^b$

# How to find parameters

1

Running times of the attack dominated by the smoothness of the order of torsion points $\implies$ $\operatorname{ord} P = \operatorname{ord} Q = 2^b$

2

Attacks in dimension two require that

$$\deg(\psi\phi\sigma) + c = 2^b$$

for c smooth and computable

# How to find parameters

1 Running times of the attack dominated by the smoothness of the order of torsion points $\Rightarrow$ $\text{ord}\, P = \text{ord}\, Q = 2^b$

2 Attacks in dimension two require that

$$\deg(\psi\phi\sigma) + c = 2^b$$

for c smooth and computable $\Rightarrow$

$$\deg(\psi\phi_1) + \deg(\phi_2\sigma) = 2^b$$

with $\phi = \phi_1\phi_2$

# How to find parameters

1

Running times of the attack dominated by the smoothness of the order of torsion points

$\Rightarrow$

ord $P$ = ord $Q$ = $2^b$

2

Attacks in dimension two require that

$$\deg(\psi\phi\sigma) + c = 2^b$$

for c smooth and computable

$\Rightarrow$

$$m_1^2 \deg(\psi\phi_1) + m_2^2 \deg(\phi_2\sigma) = 2^b$$

with $\phi = \phi_1\phi_2$

# How to find parameters

1

Running times of the attack dominated by the smoothness of the order of torsion points $\Rightarrow$ ord $P$ = ord $Q$ = $2^b$

2

Attacks in dimension two require that

$$\deg(\psi\phi\sigma) + c = 2^b$$

for c smooth and computable $\Rightarrow$

$$m_1^2 \deg(\psi\phi_1) + m_2^2 \deg(\phi_2\sigma) = 2^b$$

with $\phi = \phi_1\phi_2$

3

Ad-hoc approach based on Cornacchia's algorithm

# How to find parameters

1

Running times of the attack dominated by the smoothness of the order of torsion points

$\Rightarrow$

ord P = ord Q $= 2^b$

2

Attacks in dimension two require that

$$\deg(\psi\phi\sigma) + c = 2^b$$

for c smooth and computable

$\Rightarrow$

$$m_1^2 \deg(\psi\phi_1) + m_2^2 \deg(\phi_2\sigma) = 2^b$$

with $\phi = \phi_1\phi_2$

3

Ad-hoc approach based on Cornacchia's algorithm

$\Rightarrow$

$\deg(\psi)$, $\deg(\sigma)$ are $2^{12}$ smooth, $b = 632$,

$\deg(\phi)$ is $2^{16}$ smooth, $p \approx 2^{1292}$

# Results

```
andrea@MacBook-Pro FESTA-SageMath % sage example_festa.sage
================================================================================
                               Running FESTA_128
================================================================================

================================================================================
                           Keygen took: 4.467  seconds
================================================================================
--------------------------------------------------------------------------------
                        Compressed public key: 561 bytes
--------------------------------------------------------------------------------

================================================================================
                          Encrypt took: 3.057  seconds
================================================================================
--------------------------------------------------------------------------------
                       Compressed ciphertext: 1122 bytes
--------------------------------------------------------------------------------

================================================================================
                          Decrypt took: 10.102 seconds
================================================================================
```

# Conclusion

# Conclusion

1

New constructive framework based on the SIDH attacks

# Conclusion

1. New constructive framework based on the SIDH attacks

2. New isogeny-based PKE scheme from more conservative assumptions

# Conclusion

1. New constructive framework based on the SIDH attacks
2. New isogeny-based PKE scheme from more conservative assumptions
3. With great potential for improvements and advanced applications

# Conclusion

1. New constructive framework based on the SIDH attacks
2. New isogeny-based PKE scheme from more conservative assumptions
3. With great potential for improvements and advanced applications

**Paper**

https://eprint.iacr.org/2023/660.pdf

**Source Code**

https://github.com/FESTA-PKE/FESTA-SageMath