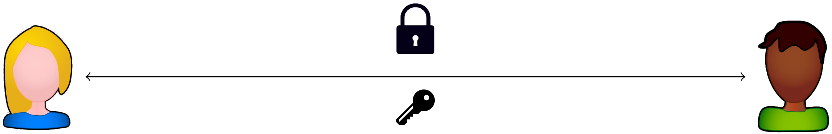
Algorithmique des courbes destinées au contexte de la cryptographie bilinéaire et post-quantique

Simon Masson

4 Décembre 2020

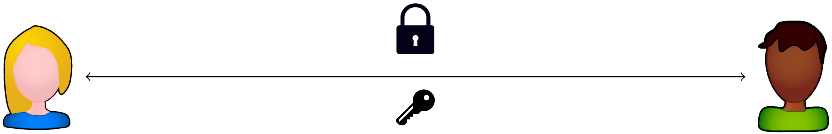
https://members.loria.fr/smasson/slides.pdf

Symmetric and asymmetric cryptography



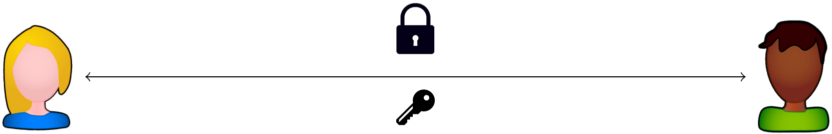
Symmetric and asymmetric cryptography

Symmetric cryptography
(communication with 🔒 and 🔑)



Symmetric and asymmetric cryptography

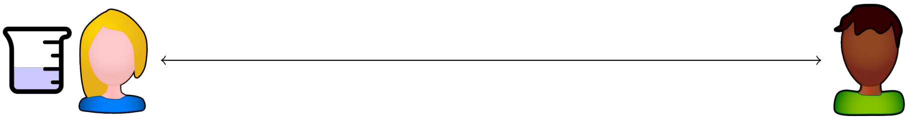
Symmetric cryptography
(communication with 🔒 and 🔑)



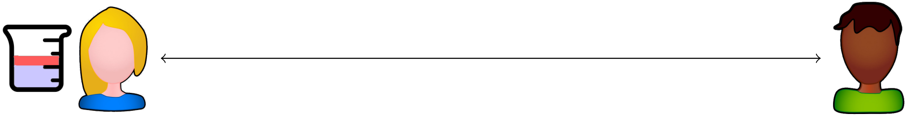
Asymmetric cryptography
(generation of the secret 🔑)

Key exchange with strawberry and mint syrups

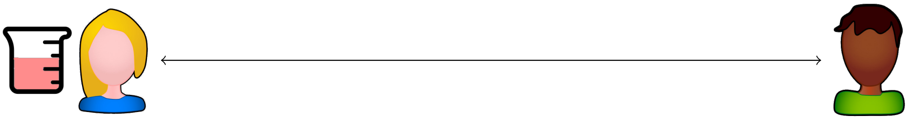
Key exchange with strawberry and mint syrups



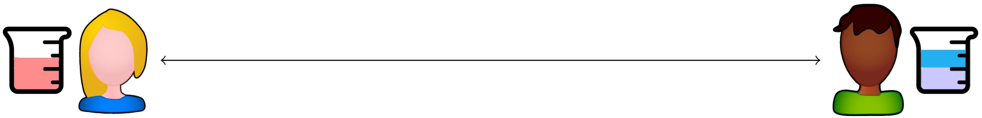
Key exchange with strawberry and mint syrups



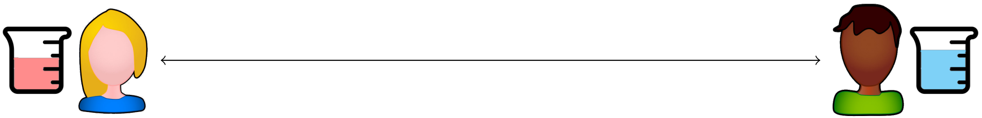
Key exchange with strawberry and mint syrups



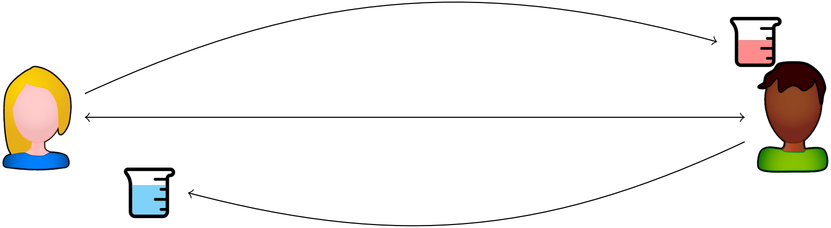
Key exchange with strawberry and mint syrups



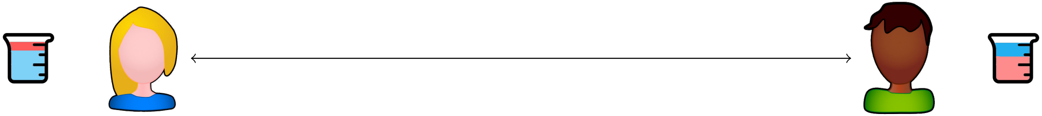
Key exchange with strawberry and mint syrups



Key exchange with strawberry and mint syrups



Key exchange with strawberry and mint syrups



Key exchange with strawberry and mint syrups

Key exchange with strawberry and mint syrups

Common secret

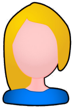
Key exchange with strawberry and mint syrups

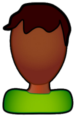
Common secret

Secure if we consider that splitting the syrups is hard.

Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

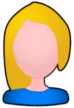




Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

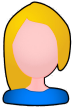
secret integer s_A



Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

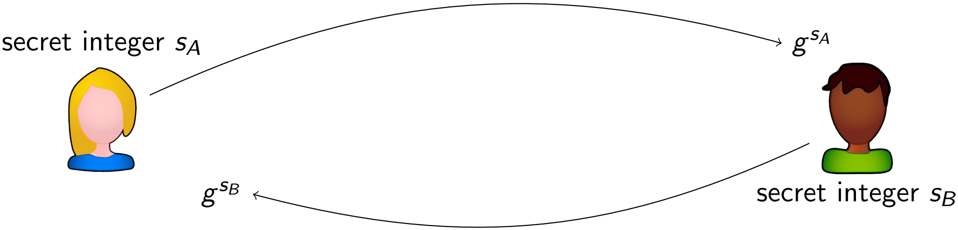
secret integer s_A



secret integer s_B

Diffie–Hellman key exchange (1976)

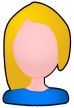
Let $G = \langle g \rangle$ be a cyclic group,



Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

secret integer s_A



$(g^{s_B})^{s_A}$

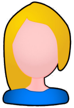
$(g^{s_A})^{s_B}$

secret integer s_B

Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

secret integer s_A



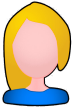
$$(g^{s_B})^{s_A} = g^{s_A s_B} = (g^{s_A})^{s_B}$$

secret integer s_B

Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

secret integer s_A



$$(g^{s_B})^{s_A} = g^{s_A s_B} = (g^{s_A})^{s_B}$$

secret integer s_B

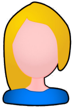
Definition (DLP over G)

Given $h \in G = \langle g \rangle$, find s such that $h = g^s$.

Diffie–Hellman key exchange (1976)

Let $G = \langle g \rangle$ be a cyclic group,

secret integer s_A



$$(g^{s_B})^{s_A} = g^{s_A s_B} = (g^{s_A})^{s_B}$$

secret integer s_B

Definition (DLP over G)

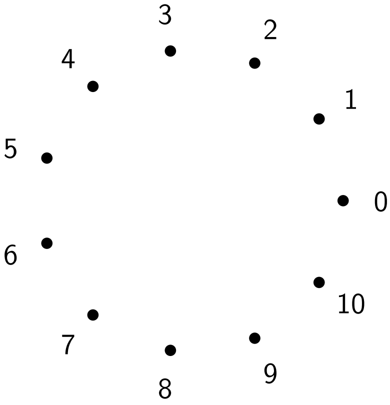
Given $h \in G = \langle g \rangle$, find s such that $h = g^s$.

The different choices of group lead to different security levels.

Finite fields

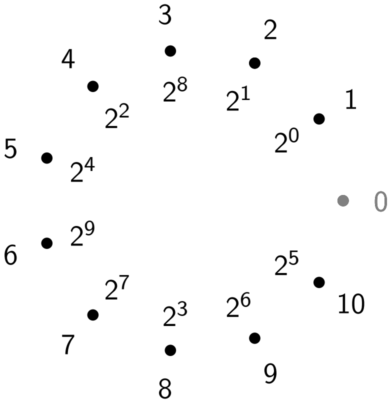
Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.



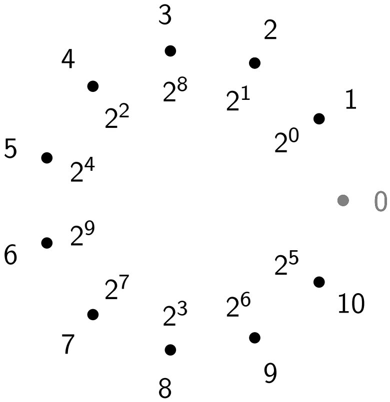
Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



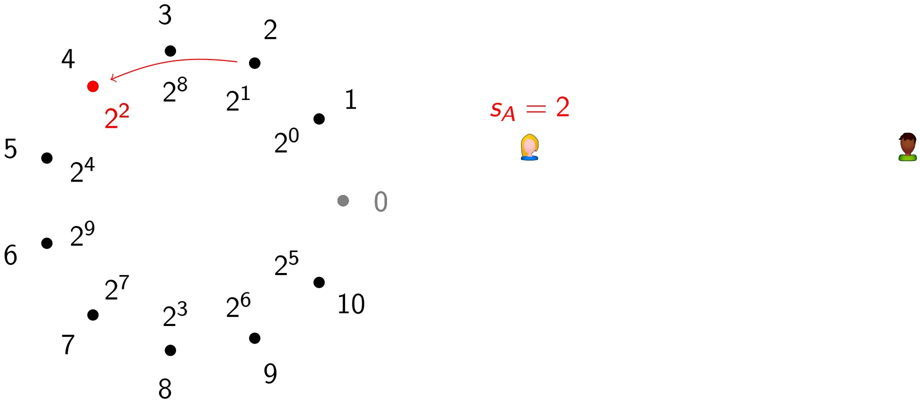
Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



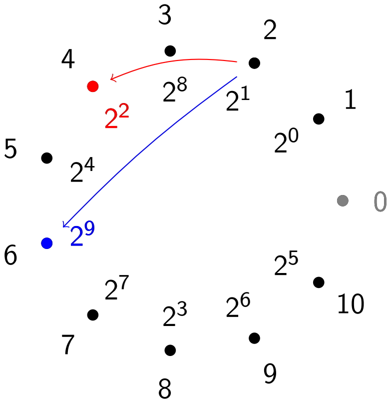
Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.

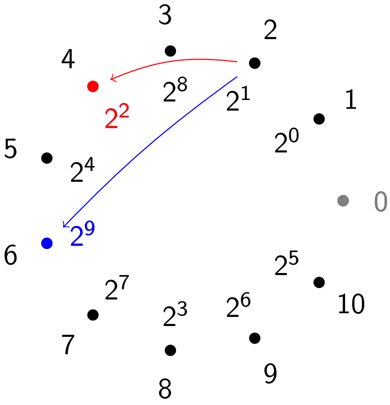


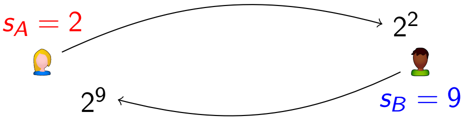
$s_A = 2$

$s_B = 9$

Diffie–Hellman in $\mathbb{F}_p$

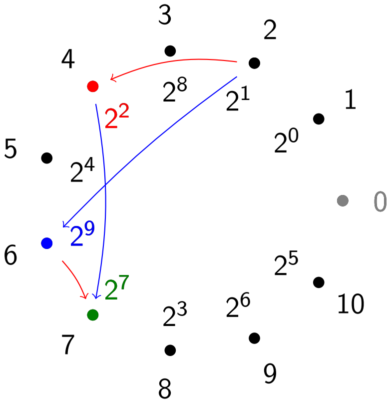
Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.





Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



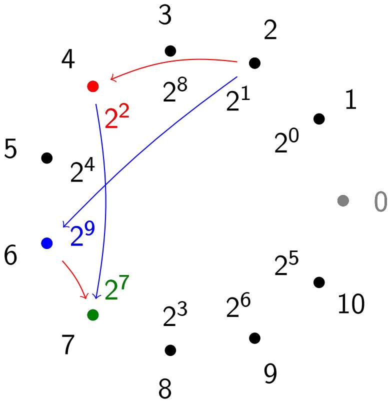
$s_A = 2$

$$2^{2\times 9} = 2^7 \bmod 11$$

$s_B = 9$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

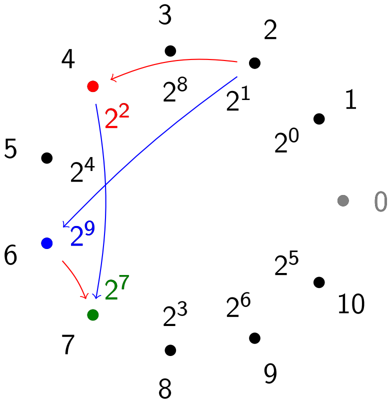
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$?

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

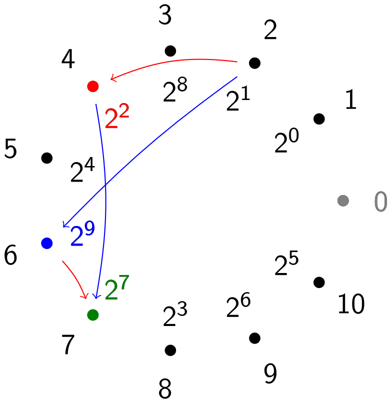
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$DL_2(6)$? $2^1 = 2 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

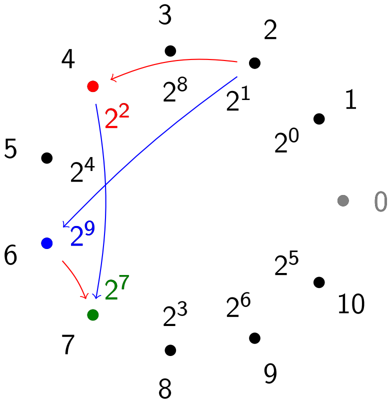
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^2 = 4 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

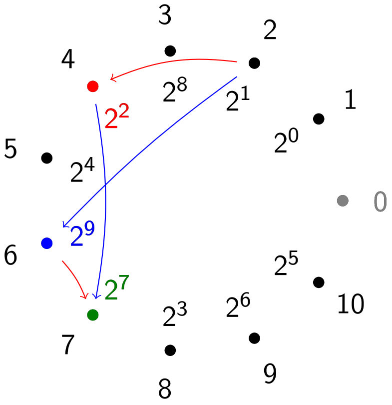
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^3 = 8 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

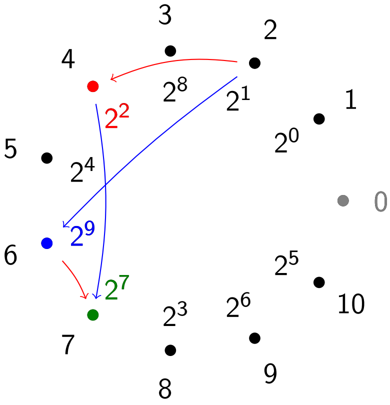
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$DL_2(6)$? $2^4 = 5 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

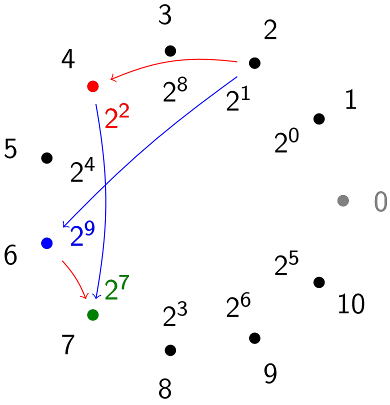
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^5 = 10 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

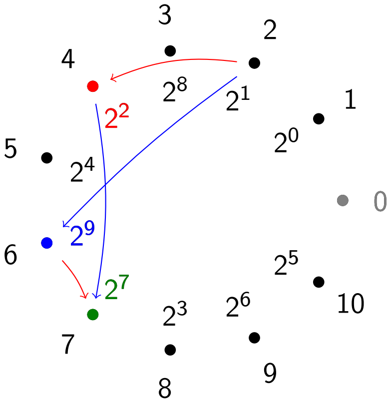
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^6 = 9 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

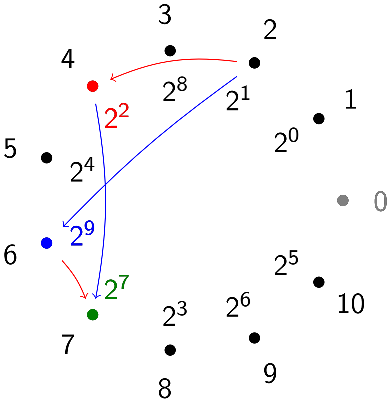
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^7 = 7 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

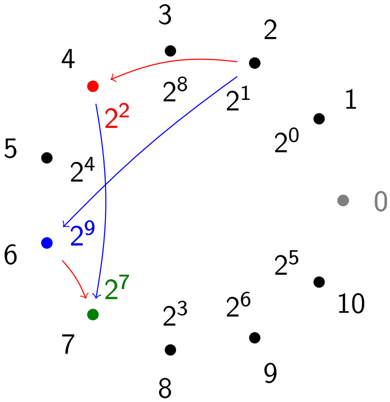
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^8 = 3 \bmod 11$

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

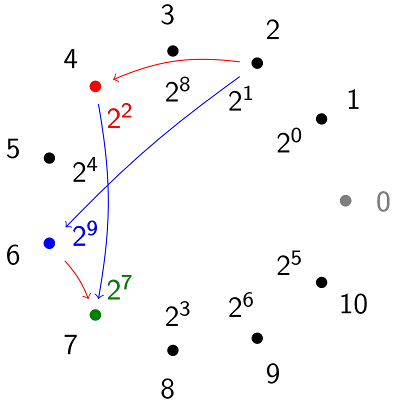
$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^9 = 6 \bmod 11$ ✓

Diffie–Hellman in $\mathbb{F}_p$

Let p be a prime, and $\mathbb{F}_p$ the finite field with p elements.
The set of invertibles $\mathbb{F}_p^*$ is a cyclic group.



$s_A = 2$

$2^{2\times 9} = 2^7 \bmod 11$

$s_B = 9$

$\mathrm{DL}_2(6)$? $2^9 = 6 \bmod 11$ ✓
Complexity $O(\#G)$ operations in G.
Exponential in the size of G.

Index calculus method

We set a factor basis S of *small* elements of G.

Index calculus method

We set a factor basis S of *small* elements of G.

1. *Relation collection.* Find relations of the form

$$g^{a_i} = \prod_{q \in S} q^{e_{q,i}}.$$

Index calculus method

We set a factor basis S of *small* elements of G.

1. *Relation collection.* Find relations of the form

$$g^{a_i} = \prod_{q \in S} q^{e_{q,i}}.$$

2. *Linear algebra.* Solve linear equations modulo ℓ of the form

$$a_i \equiv \sum_{q \in S} e_{q,i} \log q \bmod \ell$$

Index calculus method

We set a factor basis S of *small* elements of G.

1. *Relation collection.* Find relations of the form
$$g^{a_i} = \prod_{q \in S} q^{e_{q,i}}.$$

2. *Linear algebra.* Solve linear equations modulo ℓ of the form
$$a_i \equiv \sum_{q \in S} e_{q,i} \log q \bmod \ell$$

3. *Target discrete logarithm.* Find a relation between h and the elements of S, and recover $\log h$ from solutions of Step 2.

The Number Field Sieve

The Number Field Sieve

$$\mathbb{Z}[x]$$

The Number Field Sieve

$$\begin{tikzcd} & \mathbb{Z}[x] \arrow[dl] \\ \mathbb{Z}[x]/(f_1(x)) & \end{tikzcd}$$

The Number Field Sieve

$$
\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f_1(x)) & & \mathbb{Z}[x]/(f_2(x))
\end{tikzcd}
$$

The Number Field Sieve

$$\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f_1(x)) \arrow[dr, "\text{mod } (p, R(x))"'] & & \mathbb{Z}[x]/(f_2(x)) \arrow[dl, "\text{mod } (p, R(x))"] \\
 & \mathbb{F}_{p^k} &
\end{tikzcd}$$

The Number Field Sieve

$$\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f_1(x)) \arrow[dr, "\text{mod } (p, R(x))"'] & & \mathbb{Z}[x]/(f_2(x)) \arrow[dl, "\text{mod } (p, R(x))"] \\
 & \mathbb{F}_{p^k} &
\end{tikzcd}$$

NFS variant	f_1	f_2	Complexity
Original	$\deg(f_1) = 1$	$\deg(f_2) \approx (3 \log p / \log \log p)^{1/3}$	$L_{p^k}(1/3, \sqrt[3]{64/9} + o(1))$

The Number Field Sieve

$$\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f_1(x)) \arrow[dr, "\text{mod } (p, R(x))"'] & & \mathbb{Z}[x]/(f_2(x)) \arrow[dl, "\text{mod } (p, R(x))"] \\
 & \mathbb{F}_{p^k} &
\end{tikzcd}$$

NFS variant	f_1	f_2	Complexity
Original	$\deg(f_1) = 1$	$\deg(f_2) \approx (3 \log p / \log \log p)^{1/3}$	$L_{p^k}(1/3, \sqrt[3]{64/9} + o(1))$
SNFS	small coeffs, chosen with the structure of p		$L_{p^k}(1/3, \sqrt[3]{32/9} + o(1))$

The Number Field Sieve

$$\begin{tikzcd}[column sep=small]
& \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f_1(x)) \arrow[dr, "\text{mod } (p, R(x))"'] & & \mathbb{Z}[x]/(f_2(x)) \arrow[dl, "\text{mod } (p, R(x))"] \\
& \mathbb{F}_{p^k} &
\end{tikzcd}$$

NFS variant	f_1	f_2	Complexity
Original	$\deg(f_1) = 1$	$\deg(f_2) \approx (3 \log p / \log \log p)^{1/3}$	$L_{p^k}(1/3, \sqrt[3]{64/9} + o(1))$
SNFS	small coeffs, chosen with the structure of p		$L_{p^k}(1/3, \sqrt[3]{32/9} + o(1))$
TNFS	defined over $\mathbb{Z}[x]/(h(x))$		$L_{p^k}(1/3, \sqrt[3]{48/9} + o(1))$

The Number Field Sieve

$$\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f_1(x)) \arrow[dr, "\text{mod } (p, R(x))"'] & & \mathbb{Z}[x]/(f_2(x)) \arrow[dl, "\text{mod } (p, R(x))"] \\
 & \mathbb{F}_{p^k} &
\end{tikzcd}$$

NFS variant	f_1	f_2	Complexity
Original	$\deg(f_1) = 1$	$\deg(f_2) \approx (3 \log p / \log\log p)^{1/3}$	$L_{p^k}(1/3, \sqrt[3]{64/9} + o(1))$
SNFS	small coeffs, chosen with the structure of p		$L_{p^k}(1/3, \sqrt[3]{32/9} + o(1))$
TNFS	defined over $\mathbb{Z}[x]/(h(x))$		$L_{p^k}(1/3, \sqrt[3]{48/9} + o(1))$
STNFS	conditions of SNFS on $\mathbb{Z}[x]/(h(x))$		$L_{p^k}(1/3, \sqrt[3]{32/9} + o(1))$

Finite fields for a 128-bit security level

Estimation of $\log_2(p)$ so that the best NFS variant has complexity $\approx 2^{128}$ operations.

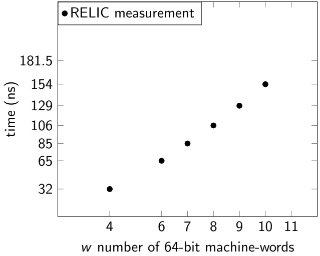
Field	$\mathbb{F}_p$	$\mathbb{F}_{p^5}$	$\mathbb{F}_{p^6}$	$\mathbb{F}_{p^7}$	$\mathbb{F}_{p^8}$	$\mathbb{F}_{p(x_0)^{12}}$	$\mathbb{F}_{p(x_0)^{16}}$
Efficent variants	NFS	NFS	NFS TNFS	NFS	NFS TNFS	NFS TNFS SNFS STNFS	NFS TNFS SNFS STNFS
Field size	3072	3315	4032	3584	4352	5352	5424
$\log_2(p)$	3072	663	672	512	544	446	339

Benchmarks of multiplications in finite fields

Field prop.	**64-bit words for p**	$\mathbb{F}_p$ **mult. timing**
special p, $k = 12$	$\square\square\square\square\square\square\square\square$	
$k = 5$	$\square\square\square\square\square\square\square\square\square\square\square$	
$k = 6$	$\square\square\square\square\square\square\square\square\square\square\square$	
$k = 7$	$\square\square\square\square\square\square\square\square$	
$k = 8$	$\square\square\square\square\square\square\square\square\square$	
$k = 1$	$\square \times 48$	

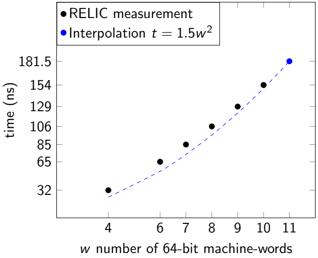
Benchmarks of multiplications in finite fields

Field prop.	**64-bit words for p**	$\mathbb{F}_p$ **mult. timing**
special p, $k = 12$	□□□□□□□□	106ns
$k = 5$	□□□□□□□□□□□	
$k = 6$	□□□□□□□□□□□	
$k = 7$	□□□□□□□□	106ns
$k = 8$	□□□□□□□□□	129ns
$k = 1$	□ $\times$ 48	



Benchmarks of multiplications in finite fields

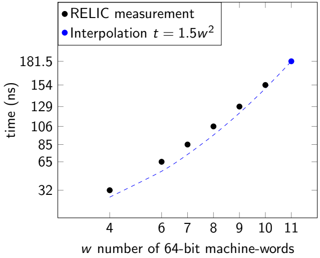
Field prop.	64-bit words for p	$\mathbb{F}_p$ mult. timing
special p, $k=12$	□□□□□□□□	106ns
$k=5$	□□□□□□□□□□□	181.5ns*
$k=6$	□□□□□□□□□□□	181.5ns*
$k=7$	□□□□□□□□	106ns
$k=8$	□□□□□□□□□	129ns
$k=1$	□ $\times$ 48	



* Interpolation from the graph

Benchmarks of multiplications in finite fields

Field prop.	64-bit words for p	$\mathbb{F}_p$ mult. timing
special p, $k = 12$	□□□□□□□□	106ns
$k = 5$	□□□□□□□□□□□	181.5ns*
$k = 6$	□□□□□□□□□□□	181.5ns*
$k = 7$	□□□□□□□□	106ns
$k = 8$	□□□□□□□□□	129ns
$k = 1$	□ $\times$ 48	3800ns**



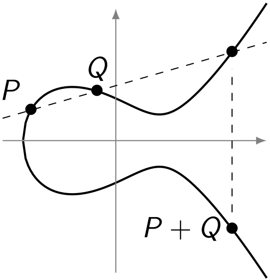
* Interpolation from the graph

**Benchmark with GMP.

Elliptic curves

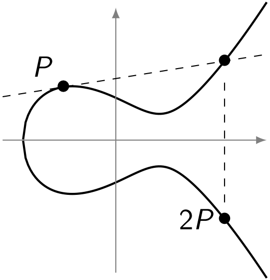
Elliptic curves group law

$E_{a,b} : y^2 = x^3 + ax + b$



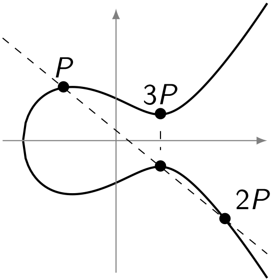
Elliptic curves group law

$$E_{a,b} : y^2 = x^3 + ax + b$$



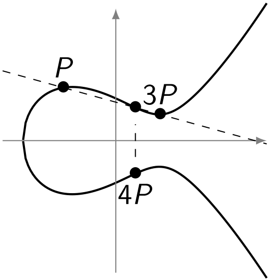
Elliptic curves group law

$$E_{a,b} : y^2 = x^3 + ax + b$$



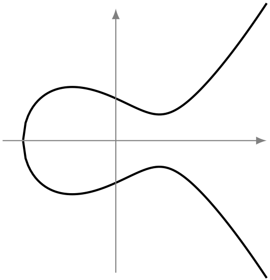
Elliptic curves group law

$E_{a,b} : y^2 = x^3 + ax + b$



Elliptic curves group law

$$E_{a,b} : y^2 = x^3 + ax + b$$



Points on an elliptic curve form a group G (with group law $+$).
Attacking the discrete log costs $O(\sqrt{\#G})$.

Torsion

Let $E_{a,b}$ be an elliptic curve defined over $\mathbb{F}_p$.

$$\begin{array}{rrcl} \pi: & E_{a,b} & \longrightarrow & E_{a,b} \\ & (x,y) & \longmapsto & (x^p, y^p) \end{array}$$

has characteristic polynomial $X^2 - tX + p$ and the order of the curve satisfies

$$\#E(\mathbb{F}_p) = p + 1 - t.$$

Hasse bound: $|t| \leq 2\sqrt{p}$.

Torsion

Let $E_{a,b}$ be an elliptic curve defined over $\mathbb{F}_p$.

$$\begin{array}{rccc} \pi : & E_{a,b} & \longrightarrow & E_{a,b} \\ & (x, y) & \longmapsto & (x^p, y^p) \end{array}$$

has characteristic polynomial $X^2 - tX + p$ and the order of the curve satisfies

$$\#E(\mathbb{F}_p) = p + 1 - t.$$

Hasse bound: $|t| \leq 2\sqrt{p}$.
For an integer ℓ, the ℓ-torsion is

$$\begin{aligned} E[\ell] &:= \left\{P \in E(\bar{\mathbb{F}}_p), \ell P = 0_E\right\} \\ &\simeq \mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z} \text{ if } \gcd(\ell, p) = 1. \end{aligned}$$

Torsion

Let $E : y^2 = x^3 + 6$ defined over $\mathbb{F}_p$ with $p = 27631$.
$\#E(\mathbb{F}_p) = r$ prime, we denote $\mathbb{G}_1 = E[r](\mathbb{F}_p)$.

Torsion

Let $E : y^2 = x^3 + 6$ defined over $\mathbb{F}_p$ with $p = 27631$.
$\#E(\mathbb{F}_p) = r$ prime, we denote $\mathbb{G}_1 = E[r](\mathbb{F}_p)$.
$E[r] \simeq \mathbb{Z}/r\mathbb{Z} \times \mathbb{Z}/r\mathbb{Z}$. Over $\mathbb{F}_{p^{12}}$, E has its full r-torsion rational.

$$\#E(\mathbb{F}_p) = 27481$$

$$\#E(\mathbb{F}_{p^{12}}) = 2^6 3^6 5^2 7^4 13^2 \cdot 73 \cdot 97 \cdot 109 \cdot 127 \cdot 283 \cdot 853 \cdot 2053 \cdot 2137 \cdot 6991 \cdot 27481^2 \cdot 7634397$$

Torsion

Let $E : y^2 = x^3 + 6$ defined over $\mathbb{F}_p$ with $p = 27631$.
$\#E(\mathbb{F}_p) = r$ prime, we denote $\mathbb{G}_1 = E[r](\mathbb{F}_p)$.
$E[r] \simeq \mathbb{Z}/r\mathbb{Z} \times \mathbb{Z}/r\mathbb{Z}$. Over $\mathbb{F}_{p^{12}}$, E has its full r-torsion rational.

$$\#E(\mathbb{F}_p) = 27481$$

$$\#E(\mathbb{F}_{p^{12}}) = 2^6 3^6 5^2 7^4 13^2 \cdot 73 \cdot 97 \cdot 109 \cdot 127 \cdot 283 \cdot 853 \cdot 2053 \cdot 2137 \cdot 6991 \cdot 27481^2 \cdot 7634397$$

We represent $E[r]$ with two subgroups of order r:

- $\mathbb{G}_1$, often chosen over $\mathbb{F}_p$. In the above example, $(21993, 24369)$ has order r.

Torsion

Let $E : y^2 = x^3 + 6$ defined over $\mathbb{F}_p$ with $p = 27631$.
$\#E(\mathbb{F}_p) = r$ prime, we denote $\mathbb{G}_1 = E[r](\mathbb{F}_p)$.
$E[r] \simeq \mathbb{Z}/r\mathbb{Z} \times \mathbb{Z}/r\mathbb{Z}$. Over $\mathbb{F}_{p^{12}}$, E has its full r-torsion rational.

$$\#E(\mathbb{F}_p) = 27481$$

$$\#E(\mathbb{F}_{p^{12}}) = 2^6 3^6 5^2 7^4 13^2 \cdot 73 \cdot 97 \cdot 109 \cdot 127 \cdot 283 \cdot 853 \cdot 2053 \cdot 2137 \cdot 6991 \cdot 27481^2 \cdot 7634397$$

We represent $E[r]$ with two subgroups of order r:

- $\mathbb{G}_1$, often chosen over $\mathbb{F}_p$. In the above example, $(21993, 24369)$ has order r.
- $\mathbb{G}_2$, often defined over $\mathbb{F}_{p^k}$ where k is the embedding degree.

$$\mathbb{F}_{p^2} = \mathbb{F}_p(i) = \mathbb{F}_p[x]/(x^2+1) \qquad \mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}(u) = \mathbb{F}_{p^2}[y]/(y^6 - (1121i + 404))$$

$$x_P = (20678i + 23625)u^5 + (1861i + 10882)u^4 + (16355i + 5810)u^3 + (20962i + 7790)u^2 + (13621i + 26347)u + 19587i + 23498,$$
$$y_P = (11673i + 12944)u^5 + (5902i + 22858)u^4 + (11246i + 24609)u^3 + (802i + 13087)u^2 + (3722i + 15960)u + 8881i + 13552.$$

Twists of curves

- If E has equation $y^2 = x^3 + ax$ for $a \in \mathbb{F}_p$, E has four quartic twists.
- If E has equation $y^2 = x^3 + b$ for $b \in \mathbb{F}_p$, E has six sextic twists.

Twists of curves

- If E has equation $y^2 = x^3 + ax$ for $a \in \mathbb{F}_p$, E has four quartic twists.
- If E has equation $y^2 = x^3 + b$ for $b \in \mathbb{F}_p$, E has six sextic twists.

Twists can lead to a compression of the $\mathbb{G}_2$ subgroup!

Twists of curves

- If E has equation $y^2 = x^3 + ax$ for $a \in \mathbb{F}_p$, E has four quartic twists.
- If E has equation $y^2 = x^3 + b$ for $b \in \mathbb{F}_p$, E has six sextic twists.

Twists can lead to a compression of the $\mathbb{G}_2$ subgroup!

Example (slide 14). a point $P \in E(\mathbb{F}_{p^{12}})$ of order r:

$$x_P = (20678i + 23625)u^5 + (1861i + 10882)u^4 + (16355i + 5810)u^3 + (20962i + 7790)u^2 + (13621i + 26347)u + 19587i + 23498,$$

$$y_P = (11673i + 12944)u^5 + (5902i + 22858)u^4 + (11246i + 24609)u^3 + (802i + 13087)u^2 + (3722i + 15960)u + 8881i + 13552.$$

Twists of curves

- If E has equation $y^2 = x^3 + ax$ for $a \in \mathbb{F}_p$, E has four quartic twists.
- If E has equation $y^2 = x^3 + b$ for $b \in \mathbb{F}_p$, E has six sextic twists.

Twists can lead to a compression of the $\mathbb{G}_2$ subgroup!

Example (slide 14). a point $P \in E(\mathbb{F}_{p^{12}})$ of order r:

$$x_P = (20678i + 23625)u^5 + (1861i + 10882)u^4 + (16355i + 5810)u^3 + (20962i + 7790)u^2 + (13621i + 26347)u + 19587i + 23498,$$
$$y_P = (11673i + 12944)u^5 + (5902i + 22858)u^4 + (11246i + 24609)u^3 + (802i + 13087)u^2 + (3722i + 15960)u + 8881i + 13552.$$

Using the sextic twist, we get a point P of order r with *sparse* coordinates:

$$x_P = (0i + 0)u^5 + (17983i + 9957)u^4 + (0i + 0)u^3 + (0i + 0)u^2 + (0i + 0)u + 0i + 0,$$
$$y_P = (0i + 0)u^5 + (0i + 0)u^4 + (12752i + 19494)u^3 + (0i + 0)u^2 + (0i + 0)u + 0i + 0.$$

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

- $\mathrm{End}(E)$ is an order of a quadratic field $\mathbb{Q}(\sqrt{-D})$.
  E is said to be *ordinary*.

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

- $\mathrm{End}(E)$ is an order of a quadratic field $\mathbb{Q}(\sqrt{-D})$.
  E is said to be *ordinary*. Ex: $\mathbb{Z}$, $\pi \Rightarrow \mathbb{Z}[\pi] \subset \mathbb{Q}(\sqrt{-D})$.

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

- $\mathrm{End}(E)$ is an order of a quadratic field $\mathbb{Q}(\sqrt{-D})$.
  E is said to be *ordinary*. Ex: $\mathbb{Z}$, $\pi \Rightarrow \mathbb{Z}[\pi] \subset \mathbb{Q}(\sqrt{-D})$.
- $\mathrm{End}(E)$ is a maximal order of a quaternion algebra (see later).
  E is said to be *supersingular*.

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

- $\mathrm{End}(E)$ is an order of a quadratic field $\mathbb{Q}(\sqrt{-D})$.
  E is said to be *ordinary*. Ex: $\mathbb{Z}$, $\pi \Rightarrow \mathbb{Z}[\pi] \subset \mathbb{Q}(\sqrt{-D})$.
- $\mathrm{End}(E)$ is a maximal order of a quaternion algebra (see later).
  E is said to be *supersingular*. Ex: $\mathbb{Z}$, π, ψ s.t. $\pi \circ \psi = -\psi \circ \pi$.

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

- $\mathrm{End}(E)$ is an order of a quadratic field $\mathbb{Q}(\sqrt{-D})$.
  E is said to be *ordinary*. Ex: $\mathbb{Z}$, $\pi \Rightarrow \mathbb{Z}[\pi] \subset \mathbb{Q}(\sqrt{-D})$.
- $\mathrm{End}(E)$ is a maximal order of a quaternion algebra (see later).
  E is said to be *supersingular*. Ex: $\mathbb{Z}$, π, ψ s.t. $\pi \circ \psi = -\psi \circ \pi$.

CM method.

Generate a curve E of given $\mathrm{End}(E)$ defined over a number field.
Restricted to discriminants $< 10^{16}$.

Endomorphism ring of elliptic curves

For an elliptic curve E defined over a finite field,

- $\mathrm{End}(E)$ is an order of a quadratic field $\mathbb{Q}(\sqrt{-D})$.
  E is said to be *ordinary*. Ex: $\mathbb{Z}$, $\pi \Rightarrow \mathbb{Z}[\pi] \subset \mathbb{Q}(\sqrt{-D})$.
- $\mathrm{End}(E)$ is a maximal order of a quaternion algebra (see later).
  E is said to be *supersingular*. Ex: $\mathbb{Z}$, π, ψ s.t. $\pi \circ \psi = -\psi \circ \pi$.

CM method.

Generate a curve E of given $\mathrm{End}(E)$ defined over a number field.
Restricted to discriminants $< 10^{16}$.
Given an order $\mathcal{O}$ of discriminant $-D$, $H_D = \texttt{HilbertClassPolynomial}(D)$.
Roots of H_D are invariants leading to curves of endomorphism ring $\mathcal{O}$.

Pairings

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

For some particular $P, Q \in E[r]$ and $a, b \in \mathbb{Z}$,

$$e(aP, bQ)$$

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

For some particular $P, Q \in E[r]$ and $a, b \in \mathbb{Z}$,

$$e(aP, bQ) = e(P, bQ)^a = e(P, Q)^{ab}$$

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

For some particular $P, Q \in E[r]$ and $a, b \in \mathbb{Z}$,

$$e(aP, bQ) = e(P, bQ)^a = e(P, Q)^{ab}$$

elliptic curve

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

For some particular $P, Q \in E[r]$ and $a, b \in \mathbb{Z}$,

$$e(aP, bQ) = e(P, bQ)^a = e(P, Q)^{ab}$$

pairing-friendly elliptic curve

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

For some particular $P, Q \in E[r]$ and $a, b \in \mathbb{Z}$,

$$e(aP, bQ) = e(P, bQ)^a = e(P, Q)^{ab}$$

Secure pairing-friendly elliptic curve

Pairings on elliptic curves

Definition

A pairing on an elliptic curve E is a bilinear non-degenerate application $e : E \times E \longrightarrow \mathbb{F}_{p^k}^{\times}$

For some particular $P, Q \in E[r]$ and $a, b \in \mathbb{Z}$,

$$e(aP, bQ) = e(P, bQ)^a = e(P, Q)^{ab}$$

Secure pairing-friendly elliptic curve with an efficient pairing

Tate and ate pairings

The Tate and ate pairings are computed in two steps:

1. Evaluating a function at a point of the curve (Miller loop)
2. Exponentiating to the power $(p^k - 1)/r$ (final exponentiation).

Tate and ate pairings

The Tate and ate pairings are computed in two steps:

1. Evaluating a function at a point of the curve (Miller loop)
2. Exponentiating to the power $(p^k - 1)/r$ (final exponentiation).

Definition

For $P \in \mathbb{G}_1 = E(\mathbb{F}_p)[r]$, $Q \in \mathbb{G}_2 = E(\mathbb{F}_{p^k})[r]$,

$$\mathrm{Tate}(P, Q) := f_{r,P}(Q)^{(p^k-1)/r} \qquad \mathrm{ate}(P, Q) := f_{t-1,Q}(P)^{(p^k-1)/r}$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$.

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{\boxed{1}01}^{2}$

$$f = 1$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{1\boxed{0}1}^{2}$

$$f = 1$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{1\boxed{0}1}^{2}$

$$f = 1^2$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{1\boxed{0}1}^{2}$

$$f = 1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{10\boxed{1}}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{10\boxed{1}}^{2}$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P)$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{10\boxed{1}}^{2}$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/v_{5Q}(P)$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/v_{5Q}(P)$$

Divisor:

$$4(Q) + 2(-2Q)$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/v_{5Q}(P)$$

Divisor:

$$4(Q) + 2(-2Q) + 2(2Q) + (-4Q)$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \color{red}{\ell_{4Q,Q}(P)}/v_{5Q}(P)$$

Divisor:

$$4(Q) + 2(-2Q) + 2(2Q) + (-4Q) + \color{red}{(Q) + (4Q) + (-5Q)}$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/v_{5Q}(P)$$

Divisor:

$$4(Q) + 2(-2Q) + 2(2Q) + (-4Q) + (Q) + (4Q) + (-5Q)$$

$$-2(2Q) - 2(-2Q) - 2(\mathcal{O})$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/v_{5Q}(P)$$

Divisor:

$$4(Q) + 2(-2Q) + 2(2Q) + (-4Q) + (Q) + (4Q) + (-5Q)$$

$$-2(2Q) - 2(-2Q) - 2(\mathcal{O}) - (4Q) - (-4Q) - (\mathcal{O})$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/\color{red}{v_{5Q}(P)}$$

Divisor:

$$4(Q) + 2(-2Q) + 2(2Q) + (-4Q) + (Q) + (4Q) + (-5Q)$$

$$-2(2Q) - 2(-2Q) - 2(\mathcal{O}) - (4Q) - (-4Q) - (\mathcal{O}) \color{red}{- (5Q) - (-5Q) - (\mathcal{O})}$$

Miller function

Definition

The Miller loop computes the function $f_{s,Q}$ such that Q is a zero of order s, and $[s]Q$ is a pole of order 1, i.e $\mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)\mathcal{O}$.

Example of $f_{5,Q}(P)$. $s = 5 = \overline{101}^2$

$$f = \left(1^2 \cdot \ell_{Q,Q}(P)/v_{2Q}(P)\right)^2 \cdot \ell_{2Q,2Q}(P)/v_{4Q}(P) \cdot \ell_{4Q,Q}(P)/v_{5Q}(P)$$

Divisor:

$$4(Q) + 2(-2Q) + 2(2Q) + (-4Q) + (Q) + (4Q) + (-5Q)$$

$$-2(2Q) - 2(-2Q) - 2(\mathcal{O}) - (4Q) - (-4Q) - (\mathcal{O}) - (5Q) - (-5Q) - (\mathcal{O})$$

$$\mathrm{div}(f) = 5(Q) - (5Q) - 4(\mathcal{O})$$

Final exponentiation

$f_{r,P}(Q)$ and $f_{t-1,Q}(P)$ are cosets modulo r-th powers.
We obtain a unique coset representative by elevating to the power $(p^k - 1)/r$.

Final exponentiation

$f_{r,P}(Q)$ and $f_{t-1,Q}(P)$ are cosets modulo r-th powers.
We obtain a unique coset representative by elevating to the power $(p^k - 1)/r$.

$$\frac{p^k - 1}{r} = \frac{p^k - 1}{\Phi_k(p)} \cdot \frac{\Phi_k(p)}{r}$$

Final exponentiation

$f_{r,P}(Q)$ and $f_{t-1,Q}(P)$ are cosets modulo r-th powers.
We obtain a unique coset representative by elevating to the power $(p^k - 1)/r$.

$$\frac{p^k - 1}{r} = \frac{p^k - 1}{\Phi_k(p)} \cdot \frac{\Phi_k(p)}{r}$$

- First exponentiation: $(p^k - 1)/\Phi_k(p)$.

Final exponentiation

$f_{r,P}(Q)$ and $f_{t-1,Q}(P)$ are cosets modulo r-th powers.
We obtain a unique coset representative by elevating to the power $(p^k - 1)/r$.

$$\frac{p^k - 1}{r} = \frac{p^k - 1}{\Phi_k(p)} \cdot \frac{\Phi_k(p)}{r}$$

- First exponentiation: $(p^k - 1)/\Phi_k(p)$.
  Polynomial in p with very small coefficients.
  Very efficient with Frobenius: if $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(x^k - \alpha)$,
  $a^p = \left(\sum_{i=0}^{k-1} a_i x^i\right)^p = \sum_{i=0}^{k-1} a_i x^{ip}$ and x^{ip} can be precomputed.

Final exponentiation

$f_{r,P}(Q)$ and $f_{t-1,Q}(P)$ are cosets modulo r-th powers.
We obtain a unique coset representative by elevating to the power $(p^k - 1)/r$.

$$\frac{p^k - 1}{r} = \frac{p^k - 1}{\Phi_k(p)} \cdot \frac{\Phi_k(p)}{r}$$

- First exponentiation: $(p^k - 1)/\Phi_k(p)$.
  Polynomial in p with very small coefficients.
  Very efficient with Frobenius: if $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(x^k - \alpha)$,
  $a^p = \left(\sum_{i=0}^{k-1} a_i x^i\right)^p = \sum_{i=0}^{k-1} a_i x^{ip}$ and x^{ip} can be precomputed.
- Second exponentiation: $\Phi_k(p)/r$.

Final exponentiation

$f_{r,P}(Q)$ and $f_{t-1,Q}(P)$ are cosets modulo r-th powers.
We obtain a unique coset representative by elevating to the power $(p^k-1)/r$.

$$\frac{p^k-1}{r} = \frac{p^k-1}{\Phi_k(p)} \cdot \frac{\Phi_k(p)}{r}$$

- First exponentiation: $(p^k-1)/\Phi_k(p)$.
  Polynomial in p with very small coefficients.
  Very efficient with Frobenius: if $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(x^k-\alpha)$,
  $a^p = \left(\sum_{i=0}^{k-1} a_i x^i\right)^p = \sum_{i=0}^{k-1} a_i x^{ip}$ and x^{ip} can be precomputed.
- Second exponentiation: $\Phi_k(p)/r$.
  More expensive. Possible optimizations.

STNFS-resistant pairing-friendly constructions

Before the NFS variants, pairing-friendly curves came from polynomial constructions, with embedding degree $k = 12$ and 16.

STNFS-resistant pairing-friendly constructions

Before the NFS variants, pairing-friendly curves came from polynomial constructions, with embedding degree $k = 12$ and 16.
All threaten by the STNFS variant!

STNFS-resistant pairing-friendly constructions

Before the NFS variants, pairing-friendly curves came from polynomial constructions, with embedding degree $k = 12$ and 16.
All threaten by the STNFS variant!
Solutions.

1. Increase the size of $\log_2(p)$ so that NFS variants are not efficient.
   R. Barbulescu and S. Duquesne, 2019.
   $\Longrightarrow \mathbb{F}_{p^k}$ becomes large.

STNFS-resistant pairing-friendly constructions

Before the NFS variants, pairing-friendly curves came from polynomial constructions, with embedding degree $k = 12$ and 16.
All threaten by the STNFS variant!
Solutions.

1. Increase the size of $\log_2(p)$ so that NFS variants are not efficient.
   R. Barbulescu and S. Duquesne, 2019.
   $\Longrightarrow \mathbb{F}_{p^k}$ becomes large.
2. Choose a $k = 1$ curve with p non-special.
   S. Chatterjee, A. Menezes, and F. Rodríguez-Henríquez, 2017.
   $\Longrightarrow$ Arithmetic over $\mathbb{F}_p$ is not efficient.

STNFS-resistant pairing-friendly constructions

Before the NFS variants, pairing-friendly curves came from polynomial constructions, with embedding degree $k = 12$ and 16.
All threaten by the STNFS variant!
Solutions.

1. Increase the size of $\log_2(p)$ so that NFS variants are not efficient.
   R. Barbulescu and S. Duquesne, 2019.
   $\Longrightarrow \mathbb{F}_{p^k}$ becomes large.
2. Choose a $k = 1$ curve with p non-special.
   S. Chatterjee, A. Menezes, and F. Rodríguez-Henríquez, 2017.
   $\Longrightarrow$ Arithmetic over $\mathbb{F}_p$ is not efficient.
3. Choose a curve with p non-special.
   We investigated this solution for $5 \leq k \leq 8$.
   A. Guillevic, S. Masson, and E. Thomé, 2020.
   $\Longrightarrow$ Competitive pairing?

Barreto-Lynn-Scott family

Curves of embedding degree $k = 12$: TNFS applies!
Discriminant $-D = -3$, twists of degree 6: $\mathbb{G}_2 \simeq {}^{t6}E(\mathbb{F}_{p^2})$.

Barreto-Lynn-Scott family

Curves of embedding degree $k = 12$: TNFS applies!
Discriminant $-D = -3$, twists of degree 6: $\mathbb{G}_2 \simeq {}^{t6}E(\mathbb{F}_{p^2})$.
$p(x) = (x^6 - 2x^5 + 2x^3 + x + 1)/3$

Barreto-Lynn-Scott family

Curves of embedding degree $k = 12$: TNFS applies!
Discriminant $-D = -3$, twists of degree 6: $\mathbb{G}_2 \simeq {}^{t6}E(\mathbb{F}_{p^2})$.
$p(x) = (x^6 - 2x^5 + 2x^3 + x + 1)/3$
p special $\Rightarrow$ fast final exponentiation.

Barreto-Lynn-Scott family

Curves of embedding degree $k = 12$: TNFS applies!
Discriminant $-D = -3$, twists of degree 6: $\mathbb{G}_2 \simeq {}^{t6}E(\mathbb{F}_{p^2})$.
$p(x) = (x^6 - 2x^5 + 2x^3 + x + 1)/3$
p special $\Rightarrow$ fast final exponentiation.
p special $\Rightarrow$ SNFS applies

Barreto-Lynn-Scott family

Curves of embedding degree $k = 12$: TNFS applies!
Discriminant $-D = -3$, twists of degree 6: $\mathbb{G}_2 \simeq {}^{t6}E(\mathbb{F}_{p^2})$.
$p(x) = (x^6 - 2x^5 + 2x^3 + x + 1)/3$
p special $\Rightarrow$ fast final exponentiation.
p special $\Rightarrow$ SNFS applies
128-bit security: NFS, SNFS, TNFS, STNFS apply!
$\log_2(p^k) \geq 5000$ to avoid NFS variants!

Barreto-Lynn-Scott family

Curves of embedding degree $k = 12$: TNFS applies!
Discriminant $-D = -3$, twists of degree 6: $\mathbb{G}_2 \simeq {}^{t6}E(\mathbb{F}_{p^2})$.
$p(x) = (x^6 - 2x^5 + 2x^3 + x + 1)/3$
p special $\Rightarrow$ fast final exponentiation.
p special $\Rightarrow$ SNFS applies
128-bit security: NFS, SNFS, TNFS, STNFS apply!
$\log_2(p^k) \geq 5000$ to avoid NFS variants!

	Estimation of the pairing cost.	**Measurements.**	
Miller loop:	$7805\text{m} \approx 0.7\text{ms}$	0.7ms	
Final expo.:	$7723\text{m} \approx 0.7\text{ms}$	0.7ms	
Total:	$15528\text{m} \approx 1.3\text{ms}$	1.4ms	(error$<$ 10%)

The Cocks–Pinch construction

Given an integer k, and a discriminant $-D$.

Algorithm: $\textsc{Cocks-Pinch}(k, -D)$ – Compute a pairing-friendly curve $E/\mathbb{F}_p$ of trace t with a subgroup of order r, such that $t^2 - 4p = -Dy^2$.

Set a prime r such that $k \mid r - 1$ and $\sqrt{-D} \in \mathbb{F}_r$
Set T such that $r \mid \Phi_k(T)$
$t \leftarrow T + 1$
$y \leftarrow (t - 2)/\sqrt{-D}$
Lift $t, y \in \mathbb{Z}$ such that $t^2 + Dy^2 \equiv 0 \mod 4$
$p \leftarrow (t^2 + Dy^2)/4$
if p is prime **then return** $[p, t, y, r]$ **else** Repeat with another r.

The Cocks–Pinch construction

Given an integer k, and a discriminant $-D$.

Algorithm: $\textsc{Cocks-Pinch}(k, -D)$ – Compute a pairing-friendly curve $E/\mathbb{F}_p$ of trace t with a subgroup of order r, such that $t^2 - 4p = -Dy^2$.

Set a prime r such that $k \mid r - 1$ and $\sqrt{-D} \in \mathbb{F}_r$
Set T such that $r \mid \Phi_k(T)$
$t \leftarrow T + 1$
$y \leftarrow (t - 2)/\sqrt{-D}$
Lift $t, y \in \mathbb{Z}$ such that $t^2 + Dy^2 \equiv 0 \mod 4$
$p \leftarrow (t^2 + Dy^2)/4$
if p is prime **then return** $[p, t, y, r]$ **else** Repeat with another r.

Large trace $t \implies$ the ate pairing is not very efficient 

Our Cocks–Pinch variant

Given an integer k, and a discriminant $-D$.

Algorithm: $\textsc{Cocks-Pinch}(k, -D)$ – Compute a pairing-friendly curve $E/\mathbb{F}_p$ of trace t with a subgroup of order r, such that $t^2 - 4p = -Dy$.

Set a small T
Set a prime r such that $k \mid r-1$, $\sqrt{-D} \in \mathbb{F}_r$ and $r \mid \Phi_k(T)$
$t \leftarrow T + 1$
$y \leftarrow (t-2)/\sqrt{-D}$
Lift $t, y \in \mathbb{Z}$ such that $t^2 + Dy^2 \equiv 0 \mod 4$
$p \leftarrow (t^2 + Dy^2)/4$
if p is prime **then return** $[p, t, y, r]$ **else** Repeat with another r.

Fix: first fix a small T and then choose r. $t = T + 1$ is small 

Our Cocks–Pinch variant

Given an integer k, and a discriminant $-D$.

Algorithm: $\textsc{Cocks-Pinch}(k, -D)$ – Compute a pairing-friendly curve $E/\mathbb{F}_p$ of trace t with a subgroup of order r, such that $t^2 - 4p = -Dy$.

Set a small T
Set a prime r such that $k \mid r-1$, $\sqrt{-D} \in \mathbb{F}_r$ and $r \mid \Phi_k(T)$
$t \leftarrow T+1$
$y \leftarrow (t-2)/\sqrt{-D}$
Lift $t, y \in \mathbb{Z}$ such that $t^2 + Dy^2 \equiv 0 \mod 4$
$p \leftarrow (t^2 + Dy^2)/4$
if p is prime and $p = 1 \bmod k$ **then return** $[p, t, y, r]$ **else** Repeat with another r.

Fix: first fix a small T and then choose r. $t = T+1$ is small 
$\mathbb{F}_{p^k} = \mathbb{F}_p[u]/(u^k - \alpha)$

Properties of our modified Cocks–Pinch curves

Example of generation for $k = 8$.

Code is available at `https://gitlab.inria.fr/smasson/cocks-pinch-variant`.

Properties of our modified Cocks–Pinch curves

Example of generation for $k = 8$.
Code is available at `https://gitlab.inria.fr/smasson/cocks-pinch-variant`.
$D = 4$ (twists of degree 4), $\log_2(T) = 64$ with small Hamming weight
Lift t and y with 16-bit h_t and h_y, and restrict on $\log_2(p) = 544$
Check subgroup-security and twist-subgroup-security.

Properties of our modified Cocks–Pinch curves

Example of generation for $k = 8$.
Code is available at https://gitlab.inria.fr/smasson/cocks-pinch-variant.
$D = 4$ (twists of degree 4), $\log_2(T) = 64$ with small Hamming weight
Lift t and y with 16-bit h_t and h_y, and restrict on $\log_2(p) = 544$
Check subgroup-security and twist-subgroup-security.

```
CocksPinchVariantResult(k=8,D=4,T=0xffffffffeff7c200,i=5,ht=5,
hy=-0xd700,allowed_cofactor=420,allowed_size_cofactor=10,max_B1=600)
```

k	$-D$	NFS	TNFS	$\log(p^k)$	$\log(p)$	Twist	$\mathbb{G}_2$ size
5	34-bit	yes	slower	3315	663	no	3315
6	-3	yes	faster	4032	672	yes	672
7	-20	yes	slower	3584	512	no	3584
8	-4	yes	faster	4352	544	yes	1088

Pairing time Comparison

At the 128-bit security level

Curve	**Miller loop** time estimation	**Exponentiation** time estimation	time estimation	RELIC **Measurement**
$k = 5$				
$k = 6$				
$k = 7$				
$k = 8$				
BN				
BLS12				
KSS16				
$k = 1$				

Pairing time Comparison

At the 128-bit security level

Curve	**Miller loop** time estimation	**Exponentiation** time estimation	time estimation	RELIC **Measurement**
$k = 5$	2.6ms	1.8ms	4.4ms	
$k = 6$	0.8ms	0.7ms	1.5ms	
$k = 7$	1.9ms	1.4ms	3.4ms	
$k = 8$	0.6ms	0.9ms	1.5ms	
BN				
BLS12				
KSS16				
$k = 1$				

Pairing time Comparison

At the 128-bit security level

Curve	**Miller loop** time estimation	**Exponentiation** time estimation	time estimation	RELIC **Measurement**
$k = 5$	2.6ms	1.8ms	4.4ms	
$k = 6$	0.8ms	0.7ms	1.5ms	
$k = 7$	1.9ms	1.4ms	3.4ms	
$k = 8$	0.6ms	0.9ms	1.5ms	
BN	1.0ms	0.5ms	1.4ms	
BLS12	0.7ms	0.7ms	1.3ms	
KSS16	0.5ms	1.2ms	1.7ms	
$k = 1$	17.7ms	15.6ms	33.3ms	

Pairing time Comparison

At the 128-bit security level

Curve	**Miller loop** time estimation	**Exponentiation** time estimation	time estimation	RELIC **Measurement**
$k = 5$	2.6ms	1.8ms	4.4ms	
$k = 6$	0.8ms	0.7ms	1.5ms	
$k = 7$	1.9ms	1.4ms	3.4ms	
$k = 8$	0.6ms	0.9ms	1.5ms	
BN	1.0ms	0.5ms	1.4ms	
BLS12	0.7ms	0.7ms	1.3ms	1.4ms
KSS16	0.5ms	1.2ms	1.7ms	
$k = 1$	17.7ms	15.6ms	33.3ms	

Pairing time Comparison

At the 128-bit security level

Curve	**Miller loop** time estimation	**Exponentiation** time estimation	time estimation	RELIC **Measurement**
$k=5$	2.6ms	1.8ms	4.4ms	
$k=6$	0.8ms	0.7ms	1.5ms	
$k=7$	1.9ms	1.4ms	3.4ms	
$k=8$	0.6ms	0.9ms	1.5ms	2.0ms
BN	1.0ms	0.5ms	1.4ms	
BLS12	0.7ms	0.7ms	1.3ms	1.4ms
KSS16	0.5ms	1.2ms	1.7ms	
$k=1$	17.7ms	15.6ms	33.3ms	

Pairing time Comparison

At the 128-bit security level

Curve	**Miller loop** time estimation	**Exponentiation** time estimation	time estimation	RELIC **Measurement**
$k = 5$	2.6ms	1.8ms	4.4ms	
$k = 6$	0.8ms	0.7ms	1.5ms	
$k = 7$	1.9ms	1.4ms	3.4ms	
$k = 8$	0.6ms	0.9ms	1.5ms	0.6ms + 1.4ms
BN	1.0ms	0.5ms	1.4ms	
BLS12	0.7ms	0.7ms	1.3ms	1.4ms
KSS16	0.5ms	1.2ms	1.7ms	
$k = 1$	17.7ms	15.6ms	33.3ms	

Isogeny of elliptic curve

Definition

An isogeny is a morphism $\varphi : E \to E'$ between elliptic curves such that $\varphi(0_E) = 0_{E'}$.

- We focus here on cyclic separable isogenies.
- A generator of $\ker(\varphi)$ defines the isogeny.
- $\deg(\varphi) \approx \# \ker(\varphi)$. Efficient for small degrees.
- An isogeny $\varphi : E \to E'$ has a dual $\hat{\varphi} : E' \to E$ s.t. $\varphi \circ \hat{\varphi} = \hat{\varphi} \circ \varphi = [\deg \varphi]$.

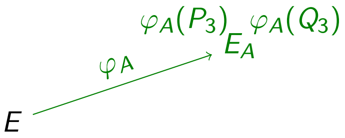
The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.

$$E$$

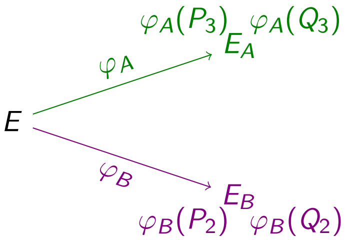
The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Alice chooses an isogeny of kernel of the form $P_2 + s_A Q_2$.
She also computes the image of P_3 and Q_3 by her isogeny.



The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Bob chooses an isogeny of kernel of the form $P_3 + s_B Q_3$.
He also computes the image of P_2 and Q_2 by his isogeny.



The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Alice computes the isogeny of kernel $\varphi_B(P_2) + s_A \varphi_B(Q_2)$.
Bob computes the isogeny of kernel $\varphi_A(P_3) + s_B \varphi_A(Q_3)$.
They arrive at the same curve E_{AB}.

$$\begin{tikzcd}
 & \overset{\varphi_A(P_3)\quad\varphi_A(Q_3)}{E_A} \arrow[dr, "\varphi_B"] & \\
E \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"'] & & E_{AB} \\
 & \underset{\varphi_B(P_2)\quad\varphi_B(Q_2)}{E_B} \arrow[ur, "\varphi_A"'] &
\end{tikzcd}$$

The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Alice computes the isogeny of kernel $\varphi_B(P_2) + s_A \varphi_B(Q_2)$.
Bob computes the isogeny of kernel $\varphi_A(P_3) + s_B \varphi_A(Q_3)$.
They arrive at the same curve E_{AB}.

$$\begin{tikzcd}
 & \varphi_A(P_3)\quad \varphi_A(Q_3) & \\
 & E_A \arrow[dr, "\varphi_B"] & \\
E \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"'] & & E_{AB} \\
 & E_B \arrow[ur, "\varphi_A"'] & \\
 & \varphi_B(P_2)\quad \varphi_B(Q_2) &
\end{tikzcd}$$

Security assumption.

- It is hard to compute φ_A given E, E_A, $\varphi_A(P_3)$ and $\varphi_A(Q_3)$

The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Alice computes the isogeny of kernel $\varphi_B(P_2) + s_A \varphi_B(Q_2)$.
Bob computes the isogeny of kernel $\varphi_A(P_3) + s_B \varphi_A(Q_3)$.
They arrive at the same curve E_{AB}.

$$\begin{tikzcd}
 & \overset{\varphi_A(P_3)\quad \varphi_A(Q_3)}{E_A} \arrow[dr, "\varphi_B"] & \\
E \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"'] & & E_{AB} \\
 & \underset{\varphi_B(P_2)\quad \varphi_B(Q_2)}{E_B} \arrow[ur, "\varphi_A"'] &
\end{tikzcd}$$

Security assumption.

- It is hard to compute φ_A given E, E_A, $\varphi_A(P_3)$ and $\varphi_A(Q_3)$

  .
- (stronger) Given E, E', it is hard to find an isogeny $E \to E'$ of given degree

  .

The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Alice computes the isogeny of kernel $\varphi_B(P_2) + s_A \varphi_B(Q_2)$.
Bob computes the isogeny of kernel $\varphi_A(P_3) + s_B \varphi_A(Q_3)$.
They arrive at the same curve E_{AB}.

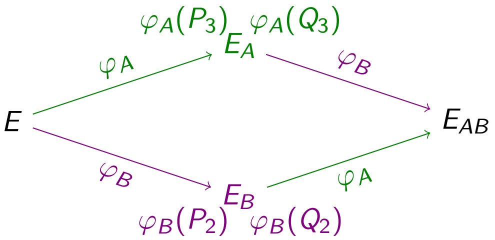
$$\begin{tikzcd}
 & \overset{\varphi_A(P_3)\ \ \varphi_A(Q_3)}{E_A} \arrow[dr, "\varphi_B"] & \\
E \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"'] & & E_{AB} \\
 & \underset{\varphi_B(P_2)\ \ \varphi_B(Q_2)}{E_B} \arrow[ur, "\varphi_A"'] &
\end{tikzcd}$$

Security assumption.

- It is hard to compute φ_A given E, E_A, $\varphi_A(P_3)$ and $\varphi_A(Q_3)$
- (stronger) Given E, E', it is hard to find an isogeny $E \to E'$ of given degree
- (stronger) It is hard to compute the **endomorphism ring** of E_A

The SIDH key exchange

E supersingular curve defined over $\mathbb{F}_{p^2}$.
$E[2^n] = \langle P_2, Q_2 \rangle$, $E[3^m] = \langle P_3, Q_3 \rangle$.
Alice computes the isogeny of kernel $\varphi_B(P_2) + s_A \varphi_B(Q_2)$.
Bob computes the isogeny of kernel $\varphi_A(P_3) + s_B \varphi_A(Q_3)$.
They arrive at the same curve E_{AB}.



Security assumption.

- It is hard to compute φ_A given E, E_A, $\varphi_A(P_3)$ and $\varphi_A(Q_3)$ even with a quantum computer.
- (stronger) Given E, E', it is hard to find an isogeny $E \to E'$ of given degree even with a quantum computer.
- (stronger) It is hard to compute the **endomorphism ring** of E_A even with a quantum computer.

Endomorphism ring of supersingular curves

Proposition

The endomorphism ring of a supersingular curve is a maximal order of a quaternion algebra.

Endomorphism ring of supersingular curves

Proposition

The endomorphism ring of a supersingular curve is a maximal order of a quaternion algebra.

- Quaternion algebra: non-commutative dimension-4 $\mathbb{Q}$-algebras.
  Here, we consider for primes p and q the quaternion algebra
  $H_{-q,-p} = \mathbb{Q}1 + \mathbb{Q}\mathrm{i} + \mathbb{Q}\mathrm{j} + \mathbb{Q}\mathrm{ij}$ where $\mathrm{i}^2 = -q$ and $\mathrm{j}^2 = -p$.

Endomorphism ring of supersingular curves

Proposition

The endomorphism ring of a supersingular curve is a maximal order of a quaternion algebra.

- Quaternion algebra: non-commutative dimension-4 $\mathbb{Q}$-algebras.
  Here, we consider for primes p and q the quaternion algebra $H_{-q,-p} = \mathbb{Q}1 + \mathbb{Q}\mathrm{i} + \mathbb{Q}\mathrm{j} + \mathbb{Q}\mathrm{ij}$ where $\mathrm{i}^2 = -q$ and $\mathrm{j}^2 = -p$.
- Order: full rank lattice which is a subring of $H_{-q,-p}$.

Endomorphism ring of supersingular curves

Proposition

The endomorphism ring of a supersingular curve is a maximal order of a quaternion algebra.

- Quaternion algebra: non-commutative dimension-4 $\mathbb{Q}$-algebras.
  Here, we consider for primes p and q the quaternion algebra
  $H_{-q,-p} = \mathbb{Q}1 + \mathbb{Q}\mathrm{i} + \mathbb{Q}\mathrm{j} + \mathbb{Q}\mathrm{ij}$ where $\mathrm{i}^2 = -q$ and $\mathrm{j}^2 = -p$.
- Order: full rank lattice which is a subring of $H_{-q,-p}$.
- Maximal order: no order contain this order.
  Maximal orders are not unique!

Example of endomorphism ring

Let $p = 3 \bmod 4$.
The curve $E : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular.

Example of endomorphism ring

Let $p = 3 \bmod 4$.
The curve $E : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular.

- $\pi : (x, y) \mapsto (x^p, y^p)$ is an endomorphism of E.

Example of endomorphism ring

Let $p = 3 \bmod 4$.
The curve $E : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular.

- $\pi : (x, y) \mapsto (x^p, y^p)$ is an endomorphism of E.
- $\psi : (x, y) \mapsto (-x, \sqrt{-1}y)$ is an endomorphism of E.

Example of endomorphism ring

Let $p = 3 \bmod 4$.
The curve $E : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular.

- $\pi : (x, y) \mapsto (x^p, y^p)$ is an endomorphism of E.
- $\psi : (x, y) \mapsto (-x, \sqrt{-1}y)$ is an endomorphism of E.

Let $q = 3$, and consider $H_{-q,-p}$,

$$\mathrm{End}(E) = \mathbb{Z} + \mathbb{Z}\mathrm{i} + \mathbb{Z}\frac{1+\mathrm{j}}{2} + \mathbb{Z}\frac{\mathrm{i}+\mathrm{k}}{2} = \mathbb{Z}[1] + \mathbb{Z}\psi + \mathbb{Z}\frac{[1]+\pi}{2} + \mathbb{Z}\frac{\psi + \psi \circ \pi}{2}$$

Example of endomorphism ring

Let $p = 3 \bmod 4$.
The curve $E : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular.

- $\pi : (x, y) \mapsto (x^p, y^p)$ is an endomorphism of E.
- $\psi : (x, y) \mapsto (-x, \sqrt{-1}y)$ is an endomorphism of E.

Let $q = 3$, and consider $H_{-q,-p}$,

$$\mathrm{End}(E) = \mathbb{Z} + \mathbb{Z}\mathrm{i} + \mathbb{Z}\frac{1+\mathrm{j}}{2} + \mathbb{Z}\frac{\mathrm{i}+\mathrm{k}}{2} = \mathbb{Z}[1] + \mathbb{Z}\psi + \mathbb{Z}\frac{[1]+\pi}{2} + \mathbb{Z}\frac{\psi + \psi \circ \pi}{2}$$

Endomorphism ring is easy to compute because it is a particular curve: reduction of a $\mathbb{Q}$-curve of discriminant $-D = -4$.

Computing endomorphism rings

- Small discriminant curves: easy.

Computing endomorphism rings

- Small discriminant curves: easy.
  $y^2 = x^3 + x$: discriminant $-D = -4$ (latter example).

Computing endomorphism rings

- Small discriminant curves: easy.
  $y^2 = x^3 + x$: discriminant $-D = -4$ (latter example).
  $y^2 = x^3 + 1$: discriminant $-D = -3$ (similar).

Computing endomorphism rings

- Small discriminant curves: easy.
  $y^2 = x^3 + x$: discriminant $-D = -4$ (latter example).
  $y^2 = x^3 + 1$: discriminant $-D = -3$ (similar).
- Random supersingular curve: hard.

Computing endomorphism rings

- Small discriminant curves: easy.
  $y^2 = x^3 + x$: discriminant $-D = -4$ (latter example).
  $y^2 = x^3 + 1$: discriminant $-D = -3$ (similar).
- Random supersingular curve: hard.
- Curve + isogeny from a small disc. curve: easy (theoretically).

Computing endomorphism rings

- Small discriminant curves: easy.
  $y^2 = x^3 + x$: discriminant $-D = -4$ (latter example).
  $y^2 = x^3 + 1$: discriminant $-D = -3$ (similar).
- Random supersingular curve: hard.
- Curve + isogeny from a small disc. curve: easy (theoretically).

Here, we focus on computing endomorphism rings given the knowledge of an isogeny (in a practical point of view).

Endomorphism ring through isogenies

$$
\begin{tikzcd}
y^2 = x^3 + x : \; E_0 \arrow[rr, bend left, "\ker\varphi = \langle P \rangle,\, \deg\varphi = 2"] & \varphi : (x,y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right) & E
\end{tikzcd}
$$

Endomorphism ring through isogenies

$$\begin{array}{ccc} y^2 = x^3 + x : \; E_0 & \xrightarrow[\varphi : (x,y) \mapsto \left(\frac{x^2-1}{x},\, y\frac{x^2+1}{x^2}\right)]{\ker\varphi = \langle P \rangle,\, \deg\varphi = 2} & E \\ \mathcal{O}_0 & & \mathcal{O} \end{array}$$

Endomorphism ring through isogenies

$$y^2 = x^3 + x : \; E_0 \xrightarrow[\varphi : (x,y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right)]{\ker\varphi = \langle P \rangle, \deg\varphi = 2} E$$

$$\mathbb{Z}\langle 1, \mathrm{i}, \tfrac{1+\mathrm{j}}{2}, \tfrac{\mathrm{i}+\mathrm{k}}{2} \rangle = \mathcal{O}_0 \qquad\qquad \mathcal{O}$$

Endomorphism ring through isogenies

$$\begin{tikzcd}
y^2 = x^3 + x : \; E_0 \arrow[rr, bend left, "\ker\varphi = \langle P\rangle, \deg\varphi = 2"] & \varphi : (x, y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right) & E \\
\mathbb{Z}\langle 1, \mathrm{i}, \frac{1+\mathrm{j}}{2}, \frac{\mathrm{i}+\mathrm{k}}{2}\rangle = \mathcal{O}_0 \arrow[rr, leftrightarrow, "\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha"'] & & \mathcal{O}
\end{tikzcd}$$

Endomorphism ring through isogenies

$$\begin{tikzcd}[column sep=huge] y^2 = x^3 + x : \ E_0 \arrow[rr, bend left, "\ker\varphi = \langle P \rangle, \deg\varphi = 2"] & \varphi : (x,y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right) & E \end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge] \mathbb{Z}\langle 1, \mathrm{i}, \frac{1+\mathrm{j}}{2}, \frac{\mathrm{i}+\mathrm{k}}{2} \rangle = \mathcal{O}_0 \arrow[r, leftrightarrow, "\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha"'] & \mathcal{O} \end{tikzcd}$$

The endomorphism α can be written $\alpha = n_1 1 + n_2 \mathrm{i} + n_3 \frac{1+\mathrm{j}}{2} + n_4 \frac{\mathrm{i}+\mathrm{k}}{2}$.

Endomorphism ring through isogenies

$$
\begin{tikzcd}
y^2 = x^3 + x : \; E_0 \arrow[rr, bend left, "\ker\varphi = \langle P \rangle,\, \deg\varphi = 2"] & \varphi : (x,y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right) & E \\
\mathbb{Z}\langle 1, \mathrm{i}, \frac{1+\mathrm{j}}{2}, \frac{\mathrm{i}+\mathrm{k}}{2} \rangle = \mathcal{O}_0 \arrow[rr, leftrightarrow, "\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha"'] & & \mathcal{O}
\end{tikzcd}
$$

The endomorphism α can be written $\alpha = n_1 1 + n_2 \mathrm{i} + n_3 \frac{1+\mathrm{j}}{2} + n_4 \frac{\mathrm{i}+\mathrm{k}}{2}$.
We solve $n_1 1(P) + n_2 \mathrm{i}(P) + n_3 (1+\mathrm{j})/2(P) + n_4 (\mathrm{i}+\mathrm{k})/2(P) = 0_{E_0}$.

Endomorphism ring through isogenies

$$
\begin{tikzcd}
y^2 = x^3 + x : \ E_0 \arrow[rrr, bend left, "{\ker\varphi = \langle P \rangle, \deg\varphi = 2}", "{\varphi : (x,y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right)}"'] & & & E
\end{tikzcd}
$$

$$
\mathbb{Z}\langle 1, \mathrm{i}, \tfrac{1+\mathrm{j}}{2}, \tfrac{\mathrm{i}+\mathrm{k}}{2} \rangle = \mathcal{O}_0 \xleftrightarrow[\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha]{\qquad\qquad} \mathcal{O}
$$

The endomorphism α can be written $\alpha = n_1 1 + n_2 \mathrm{i} + n_3 \frac{1+\mathrm{j}}{2} + n_4 \frac{\mathrm{i}+\mathrm{k}}{2}$.
We solve $n_1 1(P) + n_2 \mathrm{i}(P) + n_3(1+\mathrm{j})/2(P) + n_4(\mathrm{i}+\mathrm{k})/2(P) = 0_{E_0}$.
$\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha = \mathbb{Z}\left\langle \frac{1+\mathrm{i}+\mathrm{j}+3\mathrm{k}}{2}, \mathrm{i}+\mathrm{k}, \mathrm{j}+\mathrm{k}, 2\mathrm{k} \right\rangle$ and $\mathrm{End}(E) \simeq \mathcal{O} = O_R(\mathcal{I})$.

Endomorphism ring through isogenies

$$\begin{tikzcd}[column sep=huge] y^2 = x^3 + x : \; E_0 \arrow[r, bend left, "\ker\varphi = \langle P\rangle, \deg\varphi = 2"] \arrow[r, phantom, "\varphi : (x,y) \mapsto \left(\frac{x^2-1}{x}, y\frac{x^2+1}{x^2}\right)"] & E \\ \mathbb{Z}\langle 1, \mathrm{i}, \frac{1+\mathrm{j}}{2}, \frac{\mathrm{i}+\mathrm{k}}{2}\rangle = \mathcal{O}_0 \arrow[r, leftrightarrow, "\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha"'] & \mathcal{O} \end{tikzcd}$$

The endomorphism α can be written $\alpha = n_1 1 + n_2 \mathrm{i} + n_3 \frac{1+\mathrm{j}}{2} + n_4 \frac{\mathrm{i}+\mathrm{k}}{2}$.
We solve $n_1 1(P) + n_2 \mathrm{i}(P) + n_3(1+\mathrm{j})/2(P) + n_4(\mathrm{i}+\mathrm{k})/2(P) = 0_{E_0}$.
$\mathcal{I} = \mathcal{O}_0 \cdot 2 + \mathcal{O}_0 \cdot \alpha = \mathbb{Z}\left\langle \frac{1+\mathrm{i}+\mathrm{j}+3\mathrm{k}}{2}, \mathrm{i}+\mathrm{k}, \mathrm{j}+\mathrm{k}, 2\mathrm{k} \right\rangle$ and $\mathrm{End}(E) \simeq \mathcal{O} = O_R(\mathcal{I})$.
https://gitlab.inria.fr/smasson/endomorphismsthroughisogenies.

Definition

Definition

A *verifiable delay function* (VDF) is a function $f : X \longrightarrow Y$ such that

1. it takes T steps to evaluate, even with massive amounts of parallelism
2. the output can be verified efficiently.

Definition

Definition

A *verifiable delay function* (VDF) is a function $f : X \longrightarrow Y$ such that

1. it takes T steps to evaluate, even with massive amounts of parallelism
2. the output can be verified efficiently.

- $\text{Setup}(\lambda, T) \longrightarrow$ public parameters pp
- $\text{Eval}(pp, x) \longrightarrow$ output y such that $y = f(x)$, and a proof π (requires T steps)
- $\text{Verify}(pp, x, y, \pi) \longrightarrow$ yes or no.

VDF based on RSA

Setup. $\mathbb{Z}/N\mathbb{Z}$ where N is a RSA modulus

VDF based on RSA

Setup. $\mathbb{Z}/N\mathbb{Z}$ where N is a RSA modulus

Evaluation. $y = x^{2^T} \mod N$.

VDF based on RSA

Setup. $\mathbb{Z}/N\mathbb{Z}$ where N is a RSA modulus

Evaluation. $y = x^{2^T} \mod N$.

Verification. The evaluator also sends a proof π to convince the verifier.

VDF based on RSA

Setup. $\mathbb{Z}/N\mathbb{Z}$ where N is a RSA modulus

Evaluation. $y = x^{2^T} \mod N$.

Verification. The evaluator also sends a proof π to convince the verifier.

- **Wesolowski verification.** [Eurocrypt '19]
  π is short
  Verification is fast.
- **Pietrzak verification.** [ITCS '19]
  π computation is more efficient
  Verification is slower.

Different security assumptions.

Generalization of the RSA VDF

If one knows the factorization of N, the evaluation can be computed using

$$x^{2^T} \equiv x^{2^T \bmod \varphi(N)} \mod N$$

Need a *trusted setup* to choose N.

Generalization of the RSA VDF

If one knows the factorization of N, the evaluation can be computed using

$$x^{2^T} \equiv x^{2^T \bmod \varphi(N)} \mod N$$

Need a *trusted setup* to choose N.
This VDF also works in another *group of unknown order*.

Generalization of the RSA VDF

If one knows the factorization of N, the evaluation can be computed using

$$x^{2^T} \equiv x^{2^T \bmod \varphi(N)} \mod N$$

Need a *trusted setup* to choose N.
This VDF also works in another *group of unknown order*.
Generalization to the class group VDF. Let $K = \mathbb{Q}(\sqrt{-D})$ and O_K its ring of integers.

$$\mathrm{ClassGroup}(D) = \mathrm{Ideals}(O_K)/\mathrm{PrincipalIdeals}(O_K)$$

This group is finite and it is hard to compute $\#\mathrm{ClassGroup}(D)$.

Generalization of the RSA VDF

If one knows the factorization of N, the evaluation can be computed using

$$x^{2^T} \equiv x^{2^T \bmod \varphi(N)} \mod N$$

Need a *trusted setup* to choose N.
This VDF also works in another *group of unknown order*.
Generalization to the class group VDF. Let $K = \mathbb{Q}(\sqrt{-D})$ and O_K its ring of integers.

$$\mathrm{ClassGroup}(D) = \mathrm{Ideals}(O_K)/\mathrm{PrincipalIdeals}(O_K)$$

This group is finite and it is hard to compute $\#\mathrm{ClassGroup}(D)$.

VDF	**pro**	**con**
RSA	fast verification	trusted setup not post-quantum
Class group	small parameters	slow verification not post-quantum

VDF constructions with isogenies and pairings

L. De Feo, S. Masson, C. Petit, and A Sanso, Asiacrypt'19.

VDF constructions with isogenies and pairings

L. De Feo, S. Masson, C. Petit, and A Sanso, Asiacrypt'19.

Idea:

- Evaluation. Evaluate T isogenies sequentially at a point.
- Verification. Compute pairings on the domain and the codomain curve.

VDF constructions with isogenies and pairings

L. De Feo, S. Masson, C. Petit, and A Sanso, Asiacrypt'19.

Idea:

- Evaluation. Evaluate T isogenies sequentially at a point.
- Verification. Compute pairings on the domain and the codomain curve.

Two constructions: isogenies over $\mathbb{F}_p$ or over $\mathbb{F}_{p^2}$.

VDF constructions with isogenies and pairings

L. De Feo, S. Masson, C. Petit, and A Sanso, Asiacrypt'19.

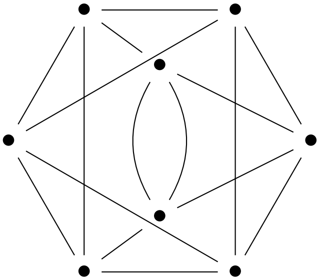
Idea:

- Evaluation. Evaluate T isogenies sequentially at a point.
- Verification. Compute pairings on the domain and the codomain curve.

Two constructions: isogenies over $\mathbb{F}_p$ or over $\mathbb{F}_{p^2}$.

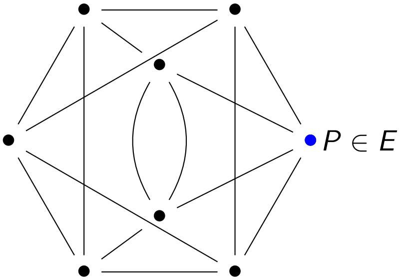
Not post-quantum because of pairings.

VDF over $\mathbb{F}_{p^2}$ supersingular curves



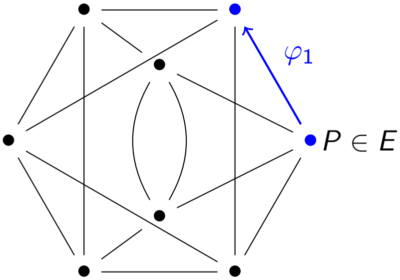
VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.



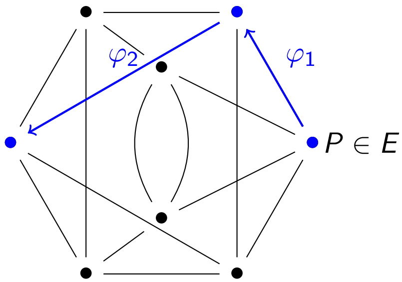
VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.



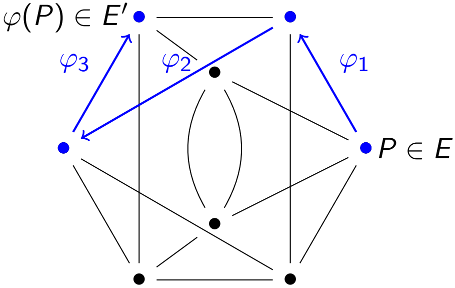
VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.



VDF over $\mathbb{F}_{p^2}$ supersingular curves

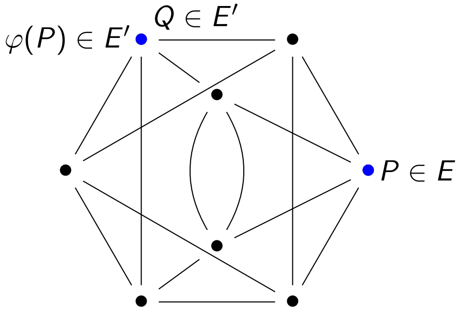
Setup A **public** walk in the isogeny graph.



VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.

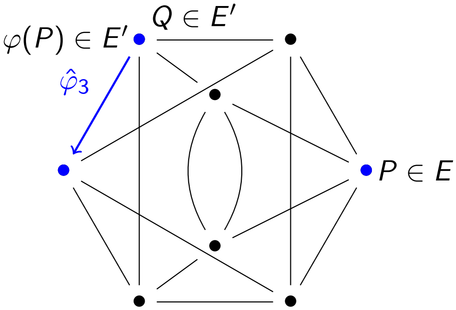
Evaluation For $Q \in E'$, compute $\hat{\varphi}(Q)$ (the backtracking walk).



VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.

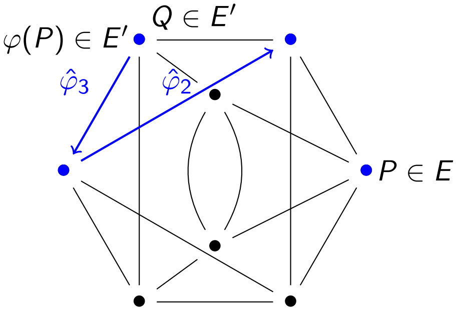
Evaluation For $Q \in E'$, compute $\hat{\varphi}(Q)$ (the backtracking walk).



VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.

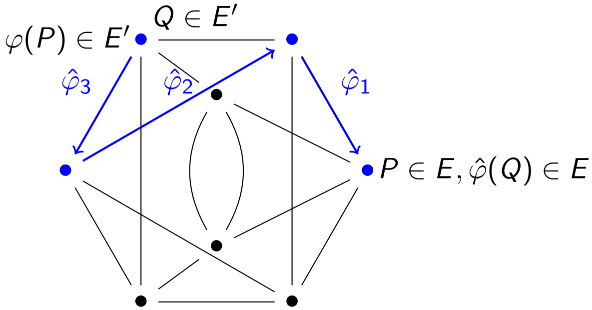
Evaluation For $Q \in E'$, compute $\hat{\varphi}(Q)$ (the backtracking walk).



VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.

Evaluation For $Q \in E'$, compute $\hat{\varphi}(Q)$ (the backtracking walk).

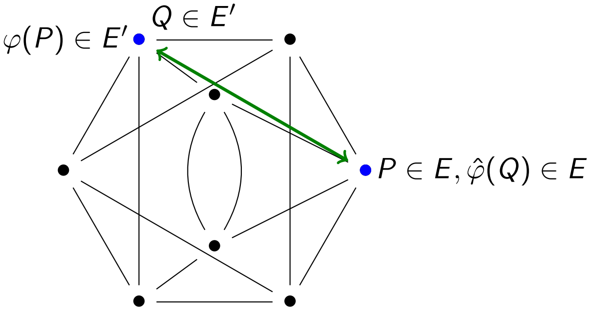


VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup A **public** walk in the isogeny graph.

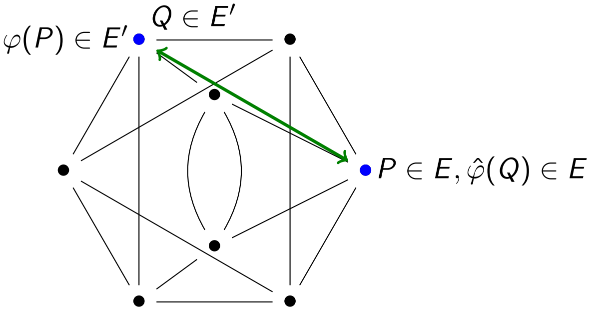
Evaluation For $Q \in E'$, compute $\hat{\varphi}(Q)$ (the backtracking walk).

Verification Check that $e(P, \hat{\varphi}(Q)) = e(\varphi(P), Q)$.



VDF over $\mathbb{F}_{p^2}$ supersingular curves

Setup	A **public** walk in the isogeny graph.
Evaluation	For $Q \in E'$, compute $\hat{\varphi}(Q)$ (the backtracking walk).
Verification	Check that $e(P, \hat{\varphi}(Q)) = e(\varphi(P), Q)$.



Another version with isogenies defined over $\mathbb{F}_p$ in the paper.

Security

What means the VDF is secure ?

Security

What means the VDF is secure ?
One cannot evaluate in less than T steps.

Security

What means the VDF is secure ?
One cannot evaluate in less than T steps.

- Attacking the DLP in $\mathbb{F}_{p^2}$.
  Writing $\mathbb{G}_2 = \langle G \rangle$, find x such that $e(P, G)^x = e(\varphi(P), Q)$.
  Solution: choose a large prime p (1500 bits) such that DLP is hard in $\mathbb{F}_{p^2}$.

Security

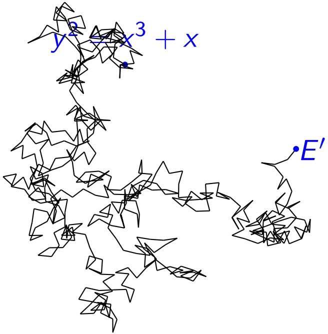
What means the VDF is secure ?
One cannot evaluate in less than T steps.

- Attacking the DLP in $\mathbb{F}_{p^2}$.
  Writing $\mathbb{G}_2 = \langle G \rangle$, find x such that $e(P, G)^x = e(\varphi(P), Q)$.
  Solution: choose a large prime p (1500 bits) such that DLP is hard in $\mathbb{F}_{p^2}$.
- Find a shortcut.
  Find a way to compute the isogeny in less than T steps.

Isogeny shortcut

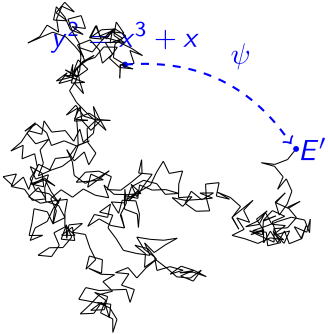
$$y^2 = x^3 + x$$

Isogeny shortcut



Isogeny shortcut

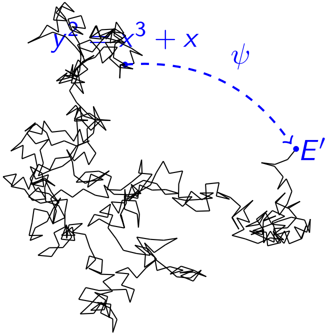
If E has a *known* endomorphism ring, a shortcut can be found.



Isogeny shortcut

If E has a *known* endomorphism ring, a shortcut can be found.

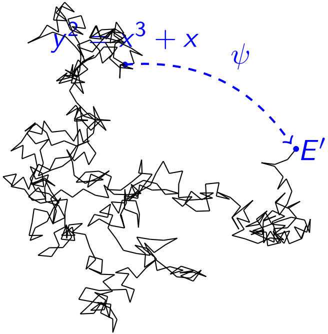
- Convert the isogeny into an ideal of $\mathrm{End}(E)$;
- Find an equivalent ideal J of different (smaller) norm;
- Convert J into another isogeny ψ of smaller degree.



Isogeny shortcut

If E has a *known* endomorphism ring, a shortcut can be found.

- Convert the isogeny into an ideal of $\mathrm{End}(E)$;
- Find an equivalent ideal J of different (smaller) norm;
- Convert J into another isogeny ψ of smaller degree.



Conclusion: **do not use a curve with a known endomorphism ring!**

The need of trusted setup

Do we know curves with an unknown endomorphism ring?

The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves
no

The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves	CM supersingular curves
no	**no**

The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves	CM supersingular curves
no	**no**

Trusted setup (supersingular case)**.**

The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves	CM supersingular curves
no	**no**

$E_0 \bullet$

Trusted setup (supersingular case)**.**

- Start from a well known supersingular curve,

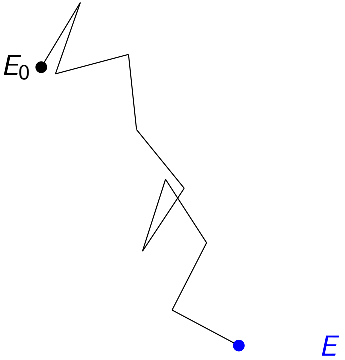
The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves	CM supersingular curves
no	**no**

Trusted setup (supersingular case)**.**

- Start from a well known supersingular curve,
- Do a random walk,



The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves	CM supersingular curves
no	**no**

Trusted setup (supersingular case)**.**

- Start from a well known supersingular curve,
- Do a random walk,
- Forget it.

The need of trusted setup

Do we know curves with an unknown endomorphism ring?

Pairing-friendly ordinary curves	CM supersingular curves
no	**no**

Trusted setup (supersingular case)**.**

- Start from a well known supersingular curve,
- Do a random walk,
- Forget it.

E has an unknown endomorphism ring.

E

Implementation of the VDF

- Proof of concept in SageMath : `https://github.com/isogenies-vdf`.

Implementation of the VDF

- Proof of concept in SageMath : https://github.com/isogenies-vdf.
- Parameters chosen for 128 bits of security

Implementation of the VDF

- Proof of concept in SageMath : `https://github.com/isogenies-vdf`.
- Parameters chosen for 128 bits of security
- Arithmetic of Montgomery curves

Implementation of the VDF

- Proof of concept in SageMath : `https://github.com/isogenies-vdf`.
- Parameters chosen for 128 bits of security
- Arithmetic of Montgomery curves
- Isogeny computation with recursive strategy

Implementation of the VDF

- Proof of concept in SageMath : `https://github.com/isogenies-vdf`.
- Parameters chosen for 128 bits of security
- Arithmetic of Montgomery curves
- Isogeny computation with recursive strategy
- Tate pairing computation.

Implementation of the VDF

- Proof of concept in SageMath : https://github.com/isogenies-vdf.
- Parameters chosen for 128 bits of security
- Arithmetic of Montgomery curves
- Isogeny computation with recursive strategy
- Tate pairing computation.

	Step	**e_k size**	**Time**	**Throughput**
$\mathbb{F}_p$ graph	Setup	238 kb	90s	0.75isog/ms
	Evaluation	–	89s	0.75isog/ms
	Verification	–	0.3s	–
$\mathbb{F}_{p^2}$ VDF	Setup	491 kb	193s	0.35isog/ms
	Evaluation	–	297s	0.23isog/ms
	Verification	–	4s	–

Table: Benchmarks for our VDFs, on a Intel Core i7-8700 @ 3.20GHz, $T \approx 2^{16}$

Comparison of the VDFs

VDF	**pro**	**con**
RSA	fast verification	trusted setup
Class group	no trusted setup small parameters	slow verification
Isogenies over $\mathbb{F}_p$	Fast verification	trusted setup long setup
Isogenies over $\mathbb{F}_{p^2}$	Quantum-annoying Fast verification	trusted setup long setup

Comparison of the VDFs

VDF	**pro**	**con**
RSA	fast verification	trusted setup
Class group	no trusted setup small parameters	slow verification
Isogenies over $\mathbb{F}_p$	Fast verification	trusted setup long setup
Isogenies over $\mathbb{F}_{p^2}$	Quantum-annoying Fast verification	trusted setup long setup

Open problems.

- Hash to the supersingular set (in order to remove the trusted setup);

Comparison of the VDFs

VDF	pro	con
RSA	fast verification	trusted setup
Class group	no trusted setup small parameters	slow verification
Isogenies over $\mathbb{F}_p$	Fast verification	trusted setup long setup
Isogenies over $\mathbb{F}_{p^2}$	Quantum-annoying Fast verification	trusted setup long setup

Open problems.

- Hash to the supersingular set (in order to remove the trusted setup);
- Find a fully post-quantum VDF.

Conclusion

- Construction of new pairing-friendly curves resistant to NFS variants, with an efficient optimal pairing.
  Aurore Guillevic, Simon Masson, and Emmanuel Thomé.
  Designs, Codes and Cryptography (2020).

Conclusion

- Construction of new pairing-friendly curves resistant to NFS variants, with an efficient optimal pairing.
  Aurore Guillevic, Simon Masson, and Emmanuel Thomé.
  Designs, Codes and Cryptography (2020).

- Implementation of endomorphism rings through isogenies.
  Magma code available at
  `https://gitlab.inria.fr/smasson/endomorphismsthroughisogenies`.

Conclusion

- Construction of new pairing-friendly curves resistant to NFS variants, with an efficient optimal pairing.
  Aurore Guillevic, Simon Masson, and Emmanuel Thomé.
  Designs, Codes and Cryptography (2020).

- Implementation of endomorphism rings through isogenies.
  Magma code available at
  https://gitlab.inria.fr/smasson/endomorphismsthroughisogenies.

- Constuction of two verifiable delay functions based on isogenies and pairings.
  Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso
  Asiacrypt 2019.

Conclusion

- Construction of new pairing-friendly curves resistant to NFS variants, with an efficient optimal pairing.
  Aurore Guillevic, Simon Masson, and Emmanuel Thomé.
  Designs, Codes and Cryptography (2020).

- Implementation of endomorphism rings through isogenies.
  Magma code available at
  https://gitlab.inria.fr/smasson/endomorphismsthroughisogenies.

- Constuction of two verifiable delay functions based on isogenies and pairings.
  Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso
  Asiacrypt 2019.

Thank you for your attention.