On the Cost of Computing Isogenies Between Supersingular Elliptic Curves

Gora Adj [1], Daniel Cervantes-Vázquez [2], Jesús-Javier Chi-Domínguez [2], Alfred Menezes [1], and Francisco Rodríguez-Henríquez [2]

[1]Department of Combinatorics & Optimization, University of Waterloo

[2]Computer Science Department, CINVESTAV-IPN

August 17, 2018

Agenda

1. Introduction
2. SIDH overview
3. CSSI problem
4. How to solve Collision Finding Problem?
    - Meet-in-the-middle
    - VW golden collision search
    - Comments about quantum attacks
    - Recommendations
5. Conclusions

Outline

Introduction

The Supersingular Isogeny Diffie-Hellman (SIDH) key agreement scheme was proposed by De Feo and Jao [De Feo & Jao'11, De Feo, Jao and Plût'14].

Introduction

The Supersingular Isogeny Diffie-Hellman (SIDH) key agreement scheme was proposed by De Feo and Jao [De Feo & Jao'11, De Feo, Jao and Plût'14].

- It is one of 69 candidates being considered by the (NIST) for inclusion in a forthcoming standard for quantum-safe cryptography [Jao *et al.*'17].

Introduction

The Supersingular Isogeny Diffie-Hellman (SIDH) key agreement scheme was proposed by De Feo and Jao [De Feo & Jao'11, De Feo, Jao and Plût'14].

- It is one of 69 candidates being considered by the (NIST) for inclusion in a forthcoming standard for quantum-safe cryptography [Jao *et al.*'17].
- Its security is based on the difficulty of the Computational Supersingular Isogeny (CSSI) problem (CSSI problem was introduced in [Charles *et al.*'09]).

Introduction: main contributions

One of our main contributions is the observation that VW golden collision search can be used to solve CSSI.

Introduction: main contributions

One of our main contributions is the observation that VW golden collision search can be used to solve CSSI.
Thus, there are two classical attacks on CSSI:

- Meet-in-the middle, and
- VW golden collision search.

Introduction: main contributions

One of our main contributions is the observation that VW golden collision search can be used to solve CSSI.
Thus, there are two classical attacks on CSSI:

- Meet-in-the middle, and
- VW golden collision search.

We argue that, even though VW is slower than MITM, it is less costly, and thus should be used to select parameters for resistance to *known* classical attacks.

Introduction: main contributions

One of our main contributions is the observation that VW golden collision search can be used to solve CSSI.
Thus, there are two classical attacks on CSSI:

- Meet-in-the middle, and
- VW golden collision search.

We argue that, even though VW is slower than MITM, it is less costly, and thus should be used to select parameters for resistance to *known* classical attacks.

Remarks: two facts about VW golden collision search:

1. it is not well known, and
2. it is different from the “usual” VW collision search.

Introduction

Flow of this presentation

In this talk, we will review the VW golden collision search as it applies to CSSI problem.

Flow of this presentation

In this talk, we will review the VW golden collision search as it applies to CSSI problem.

Remark: we are not accounting for the memory access costs, which are expected to be quite expensive.

Outline

SIDH overview
[De Feo, Jao and Plût'14, Jao *et al.*'17]

SIDH framework:

- $p = \ell_A^{e_A} \ell_B^{e_B} d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A \rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B \rangle$.

SIDH overview [De Feo, Jao and Plût'14, Jao *et al.*'17]

SIDH framework:

- $p = \ell_A^{e_A} \ell_B^{e_B} d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A \rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B \rangle$.

General description SIDH:

$$E$$

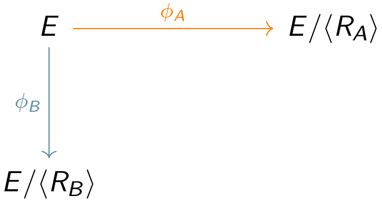
SIDH overview [De Feo, Jao and Plût'14, Jao *et al.*'17]

SIDH framework:

- $p = \ell_A^{e_A} \ell_B^{e_B} d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A \rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B \rangle$.

General description SIDH:

$$R_A \leftarrow [n_A]P_A + [m_A]Q_A$$
$$R_B \leftarrow [n_B]P_B + [m_B]Q_B$$



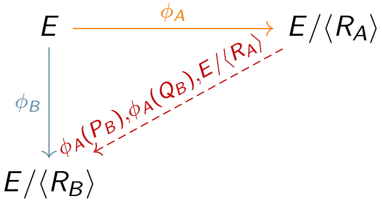
SIDH overview
[De Feo, Jao and Plût'14, Jao *et al.*'17]

SIDH framework:

- $p = \ell_A^{e_A} \ell_B^{e_B} d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A \rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B \rangle$.

General description SIDH:

$$R_A \leftarrow [n_A]P_A + [m_A]Q_A$$
$$R_B \leftarrow [n_B]P_B + [m_B]Q_B$$



SIDH overview
[De Feo, Jao and Plût'14, Jao *et al.*'17]

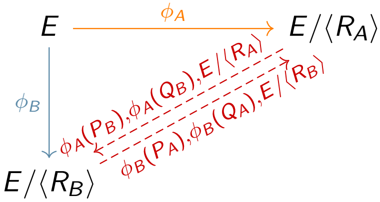
SIDH framework:

- $p = \ell_A^{e_A}\ell_B^{e_B}d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A\rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B\rangle$.

General description SIDH:

$R_A \leftarrow [n_A]P_A + [m_A]Q_A$

$R_B \leftarrow [n_B]P_B + [m_B]Q_B$



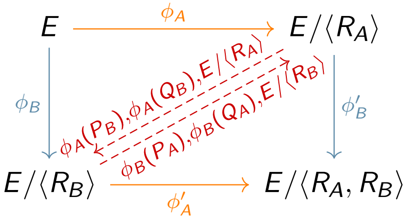
SIDH overview
[De Feo, Jao and Plût'14, Jao *et al.*'17]

SIDH framework:

- $p = \ell_A^{e_A}\ell_B^{e_B}d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A \rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B \rangle$.

General description SIDH:

$$\phi_B(R_A) \leftarrow [n_A]\phi_B(P_A) + [m_A]\phi_B(Q_A)$$
$$\phi_A(R_B) \leftarrow [n_B]\phi_A(P_B) + [m_B]\phi_A(Q_B)$$



SIDH overview [De Feo, Jao and Plût'14, Jao *et al.*'17]

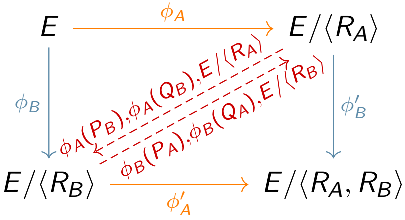
SIDH framework:

- $p = \ell_A^{e_A}\ell_B^{e_B}d - 1$ is a prime number,
- E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- $E[\ell_A^{e_A}](\mathbb{F}_{p^2}) = \langle P_A, Q_A\rangle$ and $E[\ell_B^{e_B}](\mathbb{F}_{p^2}) = \langle P_B, Q_B\rangle$.

General description SIDH:

$\phi_B(R_A) \leftarrow [n_A]\phi_B(P_A) + [m_A]\phi_B(Q_A)$

$\phi_A(R_B) \leftarrow [n_B]\phi_A(P_B) + [m_B]\phi_A(Q_B)$



The shared secret key is $j(E/\langle R_A, R_B\rangle)$.

Outline

CSSI problem

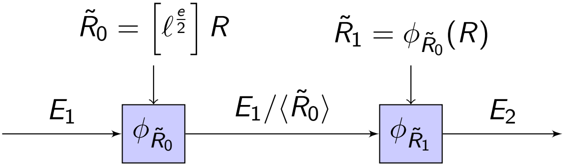
As a consequence, SIDH based its security in the hardness of the following problem

Problem (CSSI)

Given the public parameters ℓ_A, ℓ_B, e_A, e_B, p, E, P_A, Q_A, *and the elliptic curve* $E/\langle R_A \rangle$, *compute a degree-*$\ell_A^{e_A}$ *isogeny* $\phi_A : E \to E/\langle R_A \rangle$.

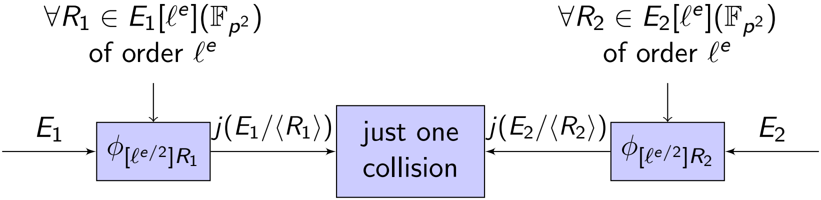
CSSI modeled as Collision Finding Problem

Let's write (R, ℓ, e) to mean either (R_A, ℓ_A, e_A) or (R_B, ℓ_B, e_B), $E_1 = E$, and $E_2 = E/\langle R \rangle$. Notice that the degree-(ℓ^e) isogeny $\phi\colon E \to E/\langle R \rangle$ can be writen as the composition of two degree-$\ell^{e/2}$ isogenies.



CSSI modeled as Collision Finding Problem

Let's write (R, ℓ, e) to mean either (R_A, ℓ_A, e_A) or (R_B, ℓ_B, e_B), $E_1 = E$, and $E_2 = E/\langle R \rangle$. Therefore, E_1 and E_2 satisfies:



Outline

Outline

Meet-in-the-middle attack

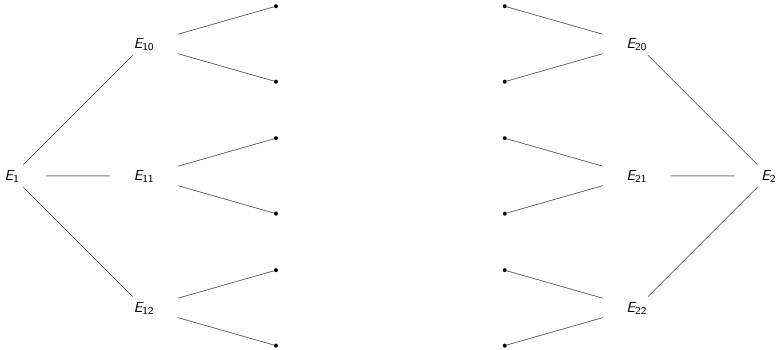
Let's ilustrate how MITM works by an example. Let $\ell_A = 2$, $\ell_B = 3$, $e_A = 4$, $e_B = 2$, $p = 2^4 \cdot 3^2 \cdot 5 - 1$,

$$E_1 \colon y^2 = x^3 + \big(\text{0x040} \cdot i + \text{0x1F0}\big)x + \big(\text{0x1E6} \cdot i + \text{0x0C7}\big),$$
$$P_1 = (\text{0x16E} \cdot i + \text{0x1B4}, \text{0x10B} \cdot i + \text{0x05F}),$$
$$Q_1 = (\text{0x203} \cdot i + \text{0x0CC}, \text{0x047} \cdot i + \text{0x0C5}), \text{ and}$$
$$E_2 \colon y^2 = x^3 + \big(\text{0x1CF} \cdot i + \text{0x047}\big)x + \big(\text{0x1EA} \cdot i + \text{0x00D}\big).$$

Then, the goal is to find a degree-2^4 isogeny from E_1 to E_2.

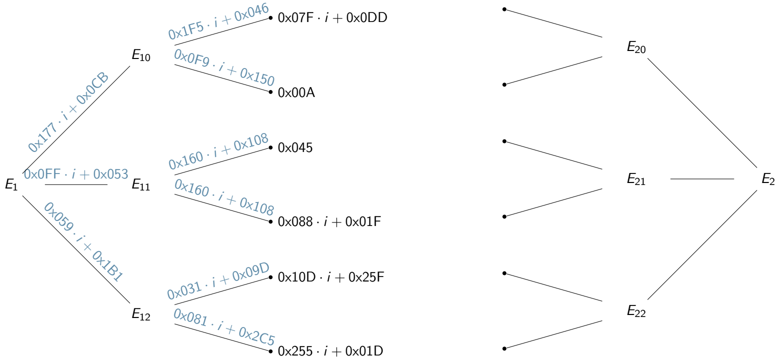
Meet-in-the-middle attack

First, compute the degree-2^2 isogeny tree rooted at E_1, and store its leaves.



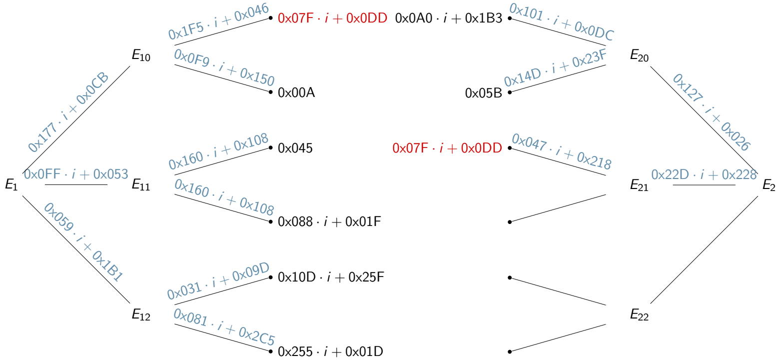
Meet-in-the-middle attack

First, compute the degree-2^2 isogeny tree rooted at E_1, and store its leaves.



Meet-in-the-middle attack

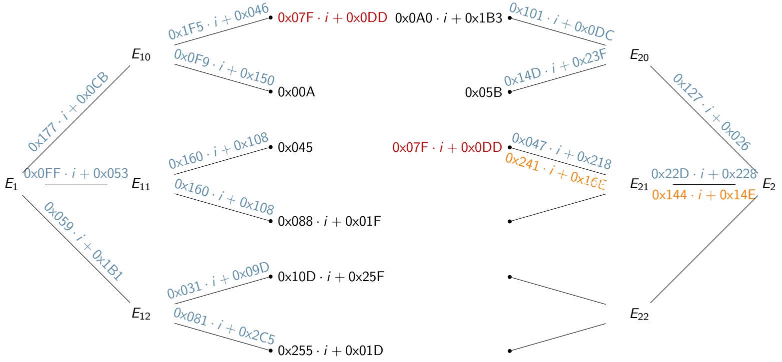
Second, compute degree-2^2 isogenies at E_2 until the match is found.



Meet-in-the-middle attack

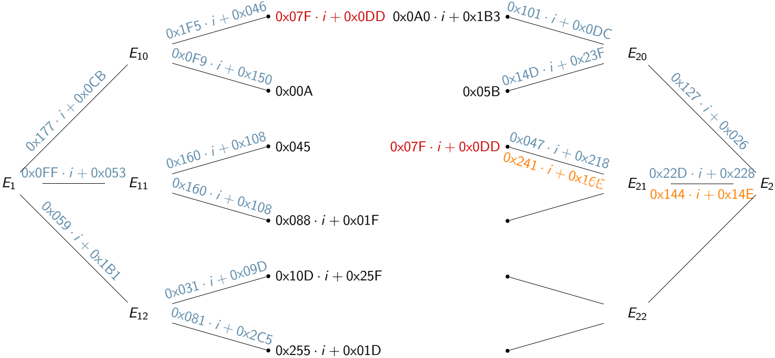
Then, we can reconstruct $\phi_A \colon E_1 \to E_2$ by composing the following isogenies:

$$E_1 \xrightarrow{\phi_0} E_{10} \xrightarrow{\phi_1} E_{100} \xrightarrow[\psi]{\mathbb{F}_{p^2}\text{-isomorphism}} E_{210} \xrightarrow{\hat{\phi}_2} E_{21} \xrightarrow{\hat{\phi}_3} E_2$$



Meet-in-the-middle attack

Now, let λ be the discrete log of $\phi_A(Q_A)$ in base $\phi_A(P_A)$ (or vice versa). Then, the secret kernel of Alice is $\langle Q_A - [\lambda]P_A \rangle$ (or $P_A - [\lambda]Q_A$). In our example, $\lambda = 3$.



Meet-in-the-middle attack

Clearly, The average-case time complexity is $1.5N$ and it has space complexity N, where $N \approx (\ell_A + 1)\ell_A^{e_A/2-1} \approx p^{1/4}$ (Infeasible for $N \geq 2^{80}$).

Meet-in-the-middle attack

Clearly, The average-case time complexity is $1.5N$ and it has space complexity N, where $N \approx (\ell_A + 1)\ell_A^{e_A/2-1} \approx p^{1/4}$ (Infeasible for $N \geq 2^{80}$).
Consequently, using m processors and w cells of memory, the running time of MITM is approximately

$$(w/m + N/m)\frac{N}{w} \approx N^2/(w \cdot m) \approx p^{1/2}/(w \cdot m).$$

Meet-in-the-middle attack: experiments

			MITM-basic				MITM-DFS
e_A	e_B	d	expected time	space	measured time	clock cycles	clock cycles
32	20	23	$2^{17.17}$	$2^{20.72}$	$2^{17.26}$	$2^{34.50}$	$2^{31.73}$
34	21	109	$2^{18.17}$	$2^{21.83}$	$2^{18.24}$	$2^{35.49}$	$2^{32.71}$
36	22	31	$2^{19.17}$	$2^{22.87}$	$2^{19.14}$	$2^{36.43}$	$2^{33.67}$
38	23	271	$2^{20.17}$	$2^{23.99}$	$2^{20.20}$	$2^{37.59}$	$2^{34.60}$
40	25	71	$2^{21.17}$	$2^{25.04}$	$2^{21.15}$	$2^{38.63}$	$2^{35.71}$
42	26	37	$2^{22.17}$	$2^{26.09}$	$2^{22.11}$	$2^{39.83}$	$2^{36.78}$
44	27	37	$2^{23.17}$	$2^{27.14}$	$2^{23.25}$	$2^{41.07}$	$2^{37.87}$

Meet-in-the-middle attacks for finding a 2^{e_A}-isogeny between two supersingular elliptic curves over $\mathbb{F}_{p^2}$ with $p = 2^{e_A} \cdot 3^{e_B} \cdot d - 1$. The 'expected time' and 'measured time' columns give the expected number and the actual number of degree-$2^{e_A/2}$ isogeny computations for MITM-basic. The space is measured in bytes.

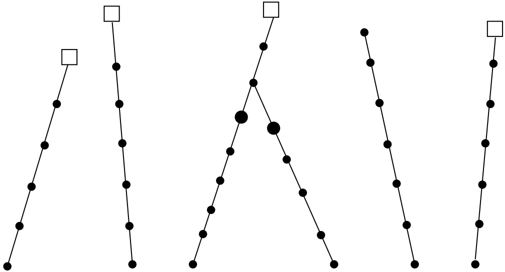
Outline

Collision search problem

Let S be a finite set of size M. The goal is to find a collision for a random function $f : S \to S$.

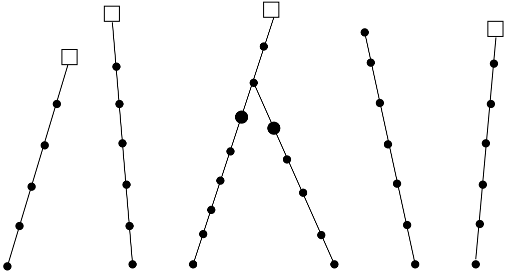
VW collision search

Firstly, let's define an element x of S to be *distinguished* if it has some easily-testable distinguishing property, and let θ be the proportion of elements of S that are distinguished.



VW collision search

Firstly, let's define an element x of S to be *distinguished* if it has some easily-testable distinguishing property, and let θ be the proportion of elements of S that are distinguished.



Then, using m processors, the expected time complexity of the VW method is approximately $\frac{1}{m}\sqrt{\pi M/2} + 2.5/\theta$.

VW golden collision search

A random function $f : S \to S$ is expected to have $(M-1)/2$ unordered collisions.

VW golden collision search

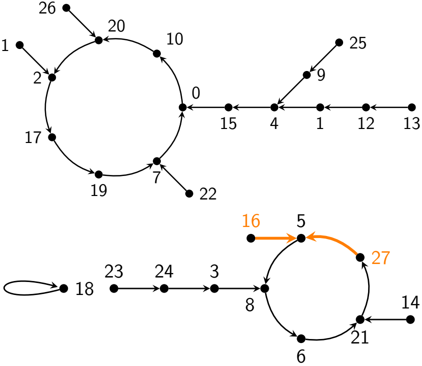
A random function $f : S \to S$ is expected to have $(M-1)/2$ unordered collisions. Suppose that we seek a particular one of these collisions, called a *golden collision*, which can be efficiently recognized.

VW golden collision search

A random function $f : S \to S$ is expected to have $(M-1)/2$ unordered collisions. Suppose that we seek a particular one of these collisions, called a *golden collision*, which can be efficiently recognized.
Consequently, one continues generating distinguished points and collisions until the golden collision is encountered.

VW golden collision search

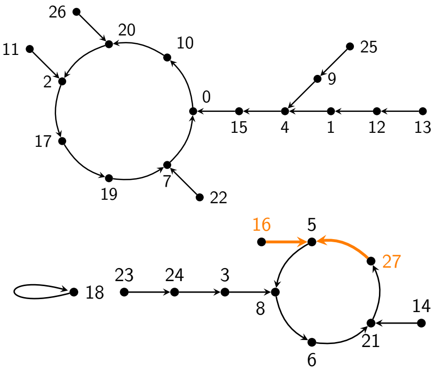
The golden collision might occur with very small probability compared to other collision.



Functional graph of a random function $f\colon \{0,\dots,27\} \to \{0,\dots,27\}$.
The desire golden collision is marked with Orange.

VW golden collision search

The golden collision might occur with very small probability compared to other collision. Thus, it is necessary to change the version of f periodically.



Functional graph of a random function $f\colon \{0,\dots,27\} \to \{0,\dots,27\}$.
The desire golden collision is marked with Orange.

VW golden collision search

Let

- w be the number of elements we can store in memory,
- $\theta = 2.25\sqrt{w/M}$,
- $10w$ be the number of distinguished elements that each version of f produces,
- $2^{10} \leq w \leq M/2^{10}$.

VW golden collision search

Let

- w be the number of elements we can store in memory,
- $\theta = 2.25\sqrt{w/M}$,
- $10w$ be the number of distinguished elements that each version of f produces,
- $2^{10} \leq w \leq M/2^{10}$.

Heuristically, van Oorschot and Wiener saw that each version of f generates approximately $1.3w$ collisions, of which approximately $1.1w$ are distinct.

VW golden collision search

Let

- w be the number of elements we can store in memory,
- $\theta = 2.25\sqrt{w/M}$,
- $10w$ be the number of distinguished elements that each version of f produces,
- $2^{10} \leq w \leq M/2^{10}$.

Heuristically, van Oorschot and Wiener saw that each version of f generates approximately $1.3w$ collisions, of which approximately $1.1w$ are distinct. In addition, the expected running time to find the golden collisions when m processors are employed is

$$\frac{1}{m}\left(2.5\sqrt{M^3/w}\right). \qquad (1)$$

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\dots,\ell\} \times \{0,\dots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

$(c, b, k) \in S$

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

$$(c, b, k) \in S \xmapsto{h_c}$$

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

$$(c,b,k) \in S \xmapsto{h_c} R = \begin{cases} [\ell \cdot k]P_c + Q_c, & \text{if } b = \ell, \\ P_c + [b \cdot \ell^{e/2-1} + k]Q_c, & \text{otherwise.} \end{cases}$$

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

$$(c,b,k) \in S \xmapsto{h_c} R = \begin{cases} [\ell \cdot k]P_c + Q_c, & \text{if } b = \ell, \\ P_c + [b \cdot \ell^{e/2-1} + k]Q_c, & \text{otherwise.} \end{cases}$$

$$\Big\downarrow f_c$$

$$j = j(E_c/\langle R \rangle) \in \mathbb{F}_{p^2}$$

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}]$, $E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

$$(c,b,k) \in S \xrightarrow{h_c} R = \begin{cases} [\ell \cdot k]P_c + Q_c, & \text{if } b = \ell, \\ P_c + [b \cdot \ell^{e/2-1} + k]Q_c, & \text{otherwise.} \end{cases}$$

$$\downarrow f_c$$

$$(c',b',k') \in S \xleftarrow[g_n]{} j = j(E_c/\langle R \rangle) \in \mathbb{F}_{p^2}$$

Here, g_n is defined by using (iteratively) a hash function and returning its $\log_2 \#S$ least significant bits.

Solving CSSI with VW golden collision search

Let $n \in \{0,1\}^{64}$, $S = \{1,2\} \times \{0,\ldots,\ell\} \times \{0,\ldots,\ell^{e/2-1}-1\}$, and $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ be bases for $E_1[\ell^{e/2}], E_2[\ell^{e/2}]$, respectively.

Then, $f : S \to S$ can be described as follows:

$$
\begin{array}{ccc}
(c,b,k) \in S & \xmapsto{h_c} & R = \begin{cases} [\ell \cdot k]P_c + Q_c, & \text{if } b = \ell, \\ P_c + [b \cdot \ell^{e/2-1} + k]Q_c, & \text{otherwise.} \end{cases} \\
\Big\downarrow{\scriptstyle f = g_n \circ f_c \circ h_c} & & \Big\downarrow{\scriptstyle f_c} \\
(c',b',k') \in S & \xleftarrow[g_n]{} & j = j(E_c/\langle R \rangle) \in \mathbb{F}_{p^2}
\end{array}
$$

Here, g_n is defined by using (iteratively) a hash function and returning its $\log_2 \# S$ least significant bits.

Solving CSSI with VW golden collision search

e	p	w	2^8	2^{10}	2^{12}	2^{14}	2^{16}
50	$2^{50}3^{31}179-1$	c_1	1.37	1.36	1.37	1.41	1.49
		c_2	1.14	1.12	1.12	1.11	1.09
60	$2^{60}3^{37}31-1$	c_1	1.37	1.34	1.34	1.35	1.36
		c_2	1.15	1.13	1.13	1.12	1.12
70	$2^{70}3^{32}127-1$	c_1	1.33	1.34	1.34	1.34	1.34
		c_2	1.13	1.14	1.13	1.13	1.13
80	$2^{80}3^{25}71-1$	c_1	1.35	1.32	1.33	1.34	1.33
		c_2	1.14	1.12	1.13	1.13	1.13

Observed number $c_1 w$ of collisions and number $c_2 w$ of distinct collisions per CSSI-based random function f_n. The numbers are averages for 25 function versions (except for $(e, w) \in \{(80, 2^{12}), (80, 2^{14}), (80, 2^{16})\}$ for which 5 function versions were used).

Solving CSSI with VW golden collision search

Therefore, using m processors and w cells of memory, the VW method can be used to find this golden collision in expected time

$$\frac{1}{m}\left(2.5\sqrt{8N^3/w}\right) \approx 7.1p^{3/8}/(w^{1/2}m).$$

Solving CSSI with VW golden collision search: experiments

					median		average	
e_A	e_B	d	w	expected time	measured time	clock cycles	measured time	clock cycles
32	20	23	2^{9}	$2^{23.20}$	$2^{23.55}$	$2^{40.79}$	$2^{24.38}$	$2^{41.62}$
34	21	109	2^{9}	$2^{24.70}$	$2^{24.54}$	$2^{41.89}$	$2^{26.02}$	$2^{43.37}$
36	22	31	2^{10}	$2^{25.70}$	$2^{26.06}$	$2^{43.51}$	$2^{27.25}$	$2^{44.70}$
38	23	271	2^{11}	$2^{26.70}$	$2^{26.15}$	$2^{43.70}$	$2^{27.69}$	$2^{45.23}$
40	25	71	2^{11}	$2^{28.20}$	$2^{26.36}$	$2^{43.99}$	$2^{29.01}$	$2^{46.64}$
42	26	37	2^{12}	$2^{29.20}$	$2^{28.92}$	$2^{46.52}$	$2^{30.95}$	$2^{48.55}$
44	27	37	2^{13}	$2^{30.20}$	$2^{29.78}$	$2^{47.46}$	$2^{30.91}$	$2^{48.58}$

Van Oorschot-Wiener golden collision search for finding a 2^{e_A}-isogeny between two supersingular elliptic curves over $\mathbb{F}_{p^2}$ with $p = 2^{e_A} \cdot 3^{e_B} \cdot d - 1$. The expected and measured times list the number of degree-$2^{e_A/2}$ isogeny computations.

Solving CSSI with VW golden collision search: 128-, 160-, 192-bit security

# processors	space	$p \approx 2^{448}$		$p \approx 2^{512}$		$p \approx 2^{536}$		$p \approx 2^{614}$	
m	w	calendar time	total time	calendar time	total time	calendar time	total time	calendar time	total time
Meet-in-the-middle using Depth-first search									
48	64	106	154	138	186	150	198	188	236
48	80	90	138	122	170	134	182	172	220
64	80	74	138	106	170	118	182	156	220
van Oorschot and Wiener golden collision search									
48	64	88	136	112	160	121	169	149	197
48	80	80	128	104	152	113	161	141	189
64	80	64	128	88	152	97	161	125	189

Time complexity estimates of CSSI attacks for $p \approx 2^{448}$, $p \approx 2^{512}$, $p \approx 2^{536}$ and $p \approx 2^{614}$. All numbers are expressed in their base-2 logarithms. The unit of time is a $2^{e/2}$-isogeny computation [2], and we are ignoring communication costs.

[2] *Calendar time* is the elapsed time taken for a computation, whereas *total time* is the sum of the time expended by all m processors.

Solving CSSI with VW golden collision search: 128-, 160-, 192-bit security

# processors m	space w	$p \approx 2^{448}$ calendar time	$p \approx 2^{448}$ total time	$p \approx 2^{512}$ calendar time	$p \approx 2^{512}$ total time	$p \approx 2^{536}$ calendar time	$p \approx 2^{536}$ total time	$p \approx 2^{614}$ calendar time	$p \approx 2^{614}$ total time
Meet-in-the-middle using Depth-first search									
48	64	106	154	138	186	150	198	188	236
48	80	90	138	122	170	134	182	172	220
64	80	74	138	106	170	118	182	156	220
van Oorschot and Wiener golden collision search									
48	64	88	136	112	160	121	169	149	197
48	80	80	128	104	152	113	161	141	189
64	80	64	128	88	152	97	161	125	189

Time complexity estimates of CSSI attacks for $p \approx 2^{448}$, $p \approx 2^{512}$, $p \approx 2^{536}$ and $p \approx 2^{614}$. All numbers are expressed in their base-2 logarithms. The unit of time is a $2^{e/2}$-isogeny computation [2], and we are ignoring communication costs.

Conclusion: MITM is more costly than VW golden collision search.

[2] *Calendar time* is the elapsed time taken for a computation, whereas *total time* is the sum of the time expended by all m processors.

Outline

Comments about quantum attacks

Tani's algorithm

The fastest known quantum attack on CSSI is Tani's algorithm [Tani'09], which has an running time equal to $O(p^{1/6})$ and requires $O(p^{1/6})$ space.

Comments about quantum attacks

Tani's algorithm

The fastest known quantum attack on CSSI is Tani's algorithm [Tani'09], which has an running time equal to $O(p^{1/6})$ and requires $O(p^{1/6})$ space.

Grover's algorithm

Clearly, CSSI can also be solved by an application of Grover's quantum search [Grover'96], which has a running time equal to $O(p^{1/4})$. However, using m quantum circuits only yields a speedup by a factor of $\sqrt{m}$ [Zalka'99].

Comments about quantum attacks

Tani's algorithm

The fastest known quantum attack on CSSI is Tani's algorithm [Tani'09], which has an running time equal to $O(p^{1/6})$ and requires $O(p^{1/6})$ space.

Grover's algorithm

Clearly, CSSI can also be solved by an application of Grover's quantum search [Grover'96], which has a running time equal to $O(p^{1/4})$. However, using m quantum circuits only yields a speedup by a factor of $\sqrt{m}$ [Zalka'99].

Tani vs Grover: the recent work of Jaques and Schanck argue that Tani's algorithm is more costly than Grover's algorithm using all reasonable cost measures [Jaques & Schank'18].

Comments about quantum attacks

NIST suggests that 2^{40} is the maximum depth of a quantum circuit that can be executed in one year using presently envisioned quantum computing architectures [NIST'16].

Comments about quantum attacks

NIST suggests that 2^{40} is the maximum depth of a quantum circuit that can be executed in one year using presently envisioned quantum computing architectures [NIST'16].

Thus, assuming that the maximum circuit depth is 2^k, the number of quantum circuits needed to perform Grover's search in one year for $p \approx 2^r$ is approximately $\left(\frac{2^{\frac{r}{4}}}{2^k}\right)^2$.

Maximum depth of a quantum circuit	$p \approx 2^{448}$ m	$p \approx 2^{512}$ m	$p \approx 2^{536}$ m	$p \approx 2^{614}$ m
40	144	176	188	227
64	96	128	140	179

Number of quantum circuits needed to perform Grover's search in one year for $p \approx 2^{448}$, $p \approx 2^{512}$, $p \approx 2^{536}$, and $p \approx 2^{614}$. All numbers are expressed in their base-2 logarithms.

Outline

Recommendations

Assuming $m \leq 2^{64}$ and $w \leq 2^{80}$, we suggest

- $p_{434} = 2^{216}3^{137} - 1$ (instead of $p_{751} = 2^{372}3^{239} - 1$ [Costello *et al.*'16]) in order to achieve 128-bit security,
- $p_{546} = 2^{273}3^{172} - 1$ (instead of $p_{964} = 2^{486}3^{301} - 1$ [Jao *et al.*'17]) in order to achieve 160-bit security, and
- $p_{610} = 2^{305}3^{192} - 1$ in order to achieve 192-bit security.

Recommendations

SIDH operations are about 4.8 times faster when p_{434} is used instead of p_{751}.

Protocol phase		CLN library [Costello *et al.*'16]			CLN + enhancements		
		p_{751}	p_{434}	p_{546}	p_{751}	p_{434}	p_{546}
Key Gen.	Alice	35.7	7.51	13.20	26.9	5.3	10.5
	Bob	39.9	8.32	14.84	30.5	6.0	11.7
Shared Secret	Alice	33.6	7.01	12.56	24.9	5.0	10.0
	Bob	38.4	7.94	14.35	28.6	5.8	11.5

Performance of the SIDH protocol. All timings are reported in 10^6 clock cycles, measured on an Intel Core i7-6700 supporting a Skylake micro-architecture. The "CLN + enhancements" columns are for our implementation that incorporates improved formulas for degree-4 and degree-3 isogenies from [Costello & Hisil'17] and Montgomery ladders from [Faz-Hernández *et al.*'17] into the CLN library.

Outline

Conclusions

- We showed that VW Golden Collision search can be used to attack CSSI.
- First implementations of MITM and Golden collision search CSSI attacks reported.
- The implementations confirm that the performance of these attacks is accurately predicted by their heuristic analysis.
- Our concrete cost analysis of the attacks leads to the conclusion that golden collision search is more cost effective that the meet-in-the-middle attack.
- SIDH operations are about 4.8 times faster when p_{434} is used instead of p_{751}.

Conclusions

SIDH parameters with p_{434} could be deemed to meet the security requirements in NIST's Category 2 [NIST'16] (classical and quantum security comparable or greater than that of SHA-256 with respect to collision resistance).

SIDH parameters with p_{610} could be deemed to meet the security requirements in NIST's Category 4 [NIST'16] (classical and quantum security comparable to that of SHA-384).

Thank you for your attention

I look forward to your comments and questions.

e-mail: jjchi@computacion.cs.cinvestav.mx

We thank Steven Galbraith for the suggestion to traverse the MITM trees using depth-first search. We also thank Sam Jaques for the many discussions on Grover's and Tani's algorithms.

Reference I

- D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies", *Post-Quantum Cryptography — PQCrypto 2011*, LNCS 7071 (2011), 19–34.
- D. Charles, E. Goren and K. Lauter, "Cryptographic hash functions from expander graphs", *Journal of Cryptology*, 22 (2009), 93–113.
- J.M. Pollard, "Monte Carlo Methods for Index Computation (mod p)". *Mathematics of Computation*, 32 (1978).
- P. van Oorschot and M. Wiener, "Improving implementable meet-in-the-middle attacks by orders of magnitude", *Advances in Cryptology — CRYPTO '96*, LNCS 1109 (1996), 229–236.

Reference II

- L. De Feo, D. Jao and J. Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies", *Journal of Mathematical Cryptology*, 8 (2014), 209–247.
- D. Jao et al., "Supersingular isogeny key encapsulation", Round 1 submission, NIST Post-Quantum Cryptography Standardization, November 30, 2017.
- Wikipedia, "Sunway TaihuLight", `https://en.wikipedia.org/wiki/Sunway_TaihuLight`.
- Wikipedia, "Exabyte", `https://en.wikipedia.org/wiki/Exabyte#Google`.

Reference III

- National Institute of Standards and Technology, "Submission requirements and evaluation criteria for the post-quantum cryptography standardization process", December 2016. Available from `https://csrc.nist.gov/csrc/media/projects/post-quantum-cryptography/documents/call-for-proposals-final-dec-2016.pdf`.
- L. Grover, "A fast quantum mechanical algorithm for database search", *Proceedings of the Twenty-Eighth Annual Symposium on Theory of Computing — STOC '96*, ACM Press (1996), 212–219.
- S. Tani, "Claw finding algorithms using quantum walk", *Theoretical Computer Science*, 410 (2009), 5285–5297.
- C. Zalka, "Grover's quantum searching algorithm is optimal", *Physical Review A*, 60 (1999), 2746–2751.

Reference IV

- C. Costello and H. Hisil, "A simple and compact algorithm for SIDH with arbitrary degree isogenies", *Advances in Cryptology — ASIACRYPT 2017*, LNCS 10624 (2017), 303–329.
- A. Faz-Hernández, J. López, E. Ochoa-Jiménez and F. Rodríguez-Henríquez, "A faster software implementation of the supersingular isogeny Diffie-Hellman key exchange protocol", *IEEE Transactions on Computers*, to appear; also available from `http://eprint.iacr.org/2017/1015`.
- C. Costello, P. Longa and M. Naehrig, "Efficient algorithms for supersingular isogeny Diffie-Hellman", *Advances in Cryptology — CRYPTO 2016*, LNCS 9814 (2016), 572–601.
- S. Jaques and J. Schanck, "Cost analyses of Tani's algorithm", in preparation.