The Computational Supersingular Isogeny Problem

Alfred Menezes

NutMiC 2019

Goals of this talk

1. Highlight some of the complications with assessing the "cost" of known attacks on computational problems.

2. Highlight some of the difficulties in comparing the costs of classical and quantum attacks.

3. Justify key size recommendations for SIDH (and SIKE).

Assessing hardness of comp. problems

1. Assess the cost of known attacks.

There are many factors to consider:

- Running time (number of arithmetic operations)
- Parallelizability
- Space requirements
- Communication costs
- Possibility of custom-designed machines
- Quantum resources

2. Assess the possibility of new attacks in the future.

RSA vs. ECC key sizes

Running time of NFS for factoring n:

$$O(\text{exp}^{(1.923+o(1))(\log n)^{1/3}(\log\log n)^{2/3}}).$$

Cost assessment is complicated:

- Communication costs for sieving (best done in cache/RAM)
- Linear algebra does not parallelize well
- Possibility of specialized hardware (TWINKLE, TWIRL)

In contrast, the cost of Pollard's rho attack on the ECDLP in $E(\mathbb{F}_p)$ is straightforward to assess:

- Expected running time is $\sqrt{\pi n}/2 \quad (n = \#E(\mathbb{F}_p) \approx p)$
- Perfectly parallelizable (van Oorschot-Wiener (VW))
- Negligible storage
- Negligible communication costs

RSA vs. ECC key sizes

After much debate, NIST issued the following key size recommendations in 2005 (SP 800-57) based on the running time of the fastest known (classical) attacks:

Bits of security	Block cipher	Hash function	RSA $\log_2 n$	ECC $\log_2 p$
80	SKIPJACK	(SHA-1)	1024	160
112	Triple-DES	SHA-224	2048	224
128	AES-128	SHA-256	3072	256
192	AES-192	SHA-384	7680	384
256	AES-256	SHA-512	15360	512

TLS 1.2: 2048-bit RSA or 256-bit ECC for key agreement.

Grover's search and AES

Let $F : \{0,1\}^\ell \to \{0,1\}$ be a function such that:

(i) F is efficiently computable; and

(ii) $F(x) = 1$ for exactly p inputs $x \in \{0,1\}^\ell$.

Grover's Search (1996) is a quantum algorithm that finds an $x \in \{0,1\}^\ell$ with $F(x) = 1$ in $2^{\ell/2}/p^{1/2}$ evaluations of F.

Key recovery: Consider AES with an ℓ-bit key. Suppose that we have r known plaintext-ciphertext pairs (m_i, c_i), where r is such that the expected number of false keys is very close to 0.

Define $F : \{0,1\}^\ell \to \{0,1\}$ by $F(k) = 1$ if $\text{AES}_k(m_i) = c_i$ for all $1 \le i \le r$; and $F(k) = 0$ otherwise.

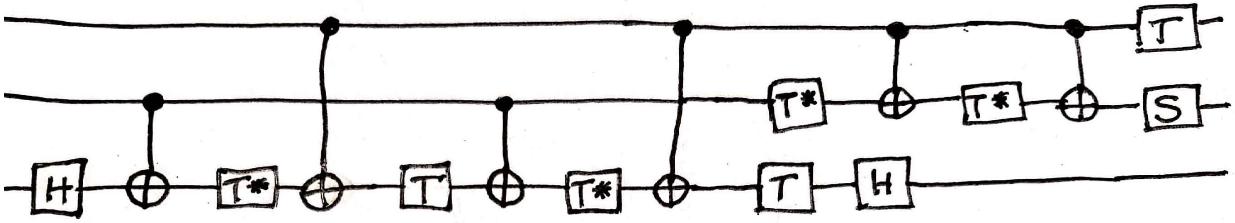
Then Grover's search (with $p = 1$) can find the secret key k in $2^{\ell/2}$ operations.

Grover's search is often used to justify moving from AES-128 to AES-256.

Quantum resource estimates (AES-128)

Grassl-Langenberg-Roetteler-Steinwandt (PQCrypto 2016)

- # circuits: 1
- # qubits: 2,953
- # gates: 2^{87}
- depth: 2^{81}



NIST: Quantum attacks are restricted to a fixed circuit depth, called MAXDEPTH. Plausible values for MAXDEPTH:

- 2^{40} gates (approx. # of gates that presently envisioned quantum computing architectures are expected to serially perform in a year).
- 2^{64} gates (approx. # of gates that current classical computing architectures can perform serially in a decade).
- 2^{96} gates (approx. # of gates that atomic scale qubits with speed of light propagation times could perform in a millennium).

The attack needs to be parallelized.

Grover's search doesn't parallelize well

Optimal strategy (Zalka 1999): Divide the search space into M subsets, each of size $2^\ell/M$. Each of the M processors performs Grover's search on one subset.

Running time (per processor): $2^{\ell/2}/\sqrt{M}$.

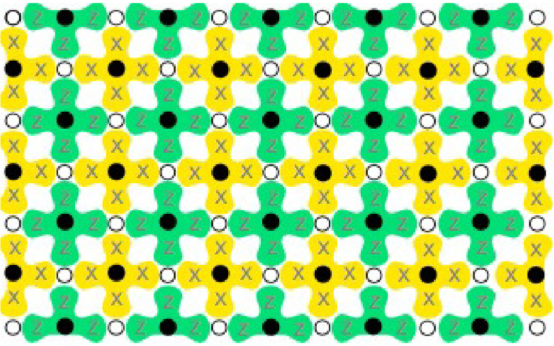
depth:	2^{81}	2^{40}	2^{48}	2^{64}
# circuits:	1	2^{82}	2^{66}	2^{34}
# qubits/circuit:	2,953	2,953	2,953	2,953
# gates/circuit:	2^{87}	2^{46}	2^{54}	2^{70}
Total # gates:	2^{87}	2^{128}	2^{120}	2^{104}

Quantum error correction

Self-correcting quantum memory may not exist.

Actively-controlled quantum memories:

- To protect a circuit of depth D and width W, a surface code requires $\Theta(\log^2(DW))$ physical qubits per logical qubit.



arxiv.org/abs/1208.0928

- The active error correction is applied with a classical processor in a regular cycle (e.g. once every 200*ns*).
- The overall cost of surface code computation is $\Omega(\log^2(DW))$ RAM operations per logical qubit per layer of logical circuit depth.
- Quantum error correction has large overhead.
- This explains why DW-cost is a realistic cost measure for a quantum algorithm.

AES-128 security, revisited

Quantum	#	Classical	#
depth:	2^{40}	depth:	2^{35} AES ops
circuits:	2^{82}	processors:	2^{93}
qubits/circuit:	2,953		
gates/circuit:	2^{46}	gates/processor:	2^{50}
Total gates:	2^{128}	Total gates:	2^{143}

- The 2^{93} classical processors used for error correction could be repurposed to perform exhaustive key search in time 2^{35} AES operations.
- It isn't clear then that Grover's search is more effective than classical exhaustive search in breaking AES-128.
- Nevertheless, since AES-256 is only marginally slower than AES-128, it is reasonable to move from AES-128 to AES-256.

NIST Category 1

- Any attack must require computational resources comparable to or greater than those required for key search on AES-128.
- ...with respect to *all* metrics that NIST deems to be potentially relevant to practical security.
- NIST intends to consider a variety of possible metrics, reflecting different predictions about the future development of quantum and classical computing technology.

- Fixed circuit depth (MAXDEPTH)
- Cost metric: Number of gates
  - 2^{143} classical gates
  - 2^{170}/MAXDEPTH quantum gates
    (2^{130} quantum gates if MAXDEPTH = 2^{40})

- Category 3 (AES-192):
  - 2^{207} classical gates, 2^{233}/MAXDEPTH quantum gates

Hash function collisions: Grover

Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be an ℓ-bit hash function.

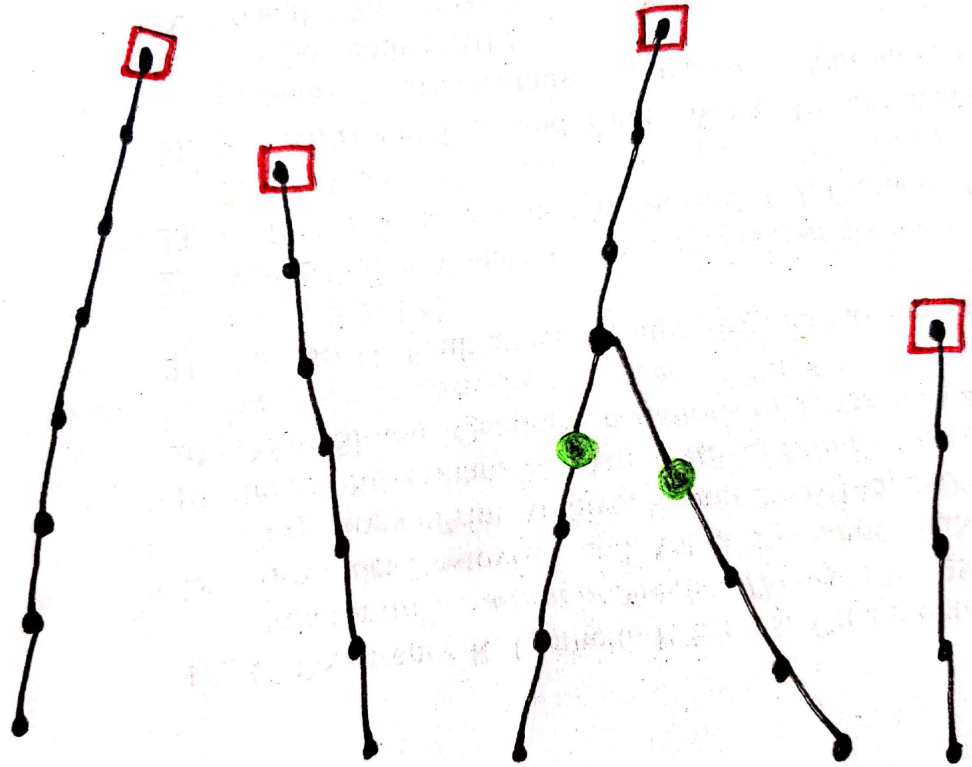
- A collision is a pair (x, y) with $H(x) = H(y)$ and $x \neq y$.
- Define $F : \{0,1\}^{\ell+c} \times \{0,1\}^{\ell+c} \to \{0,1\}$ by

  $$F(x,y) = \begin{cases} 0, & \text{if } H(x) \neq H(y), \\ 1, & \text{if } H(x) = H(y) \text{ and } x \neq y. \end{cases}$$

  The expected number of collisions is $\approx 2^{\ell+2c}$.
- Grover's search with M processors can find a collision in time $2^{\ell/2}/\sqrt{M}$.
- If $M = 2^{\ell/3}$, the time is $2^{\ell/3}$.
- So, collisions for SHA-256 can be found in time $2^{85.3}$.

Collision finding: Classical (VW)

- The fastest generic classic finding algorithm for finding a collision for $f : S \to S$ (where $\#S = N$) is due to van Oorschot-Wiener (VW).
- Let θ be the distinguishing probability for elements in S.

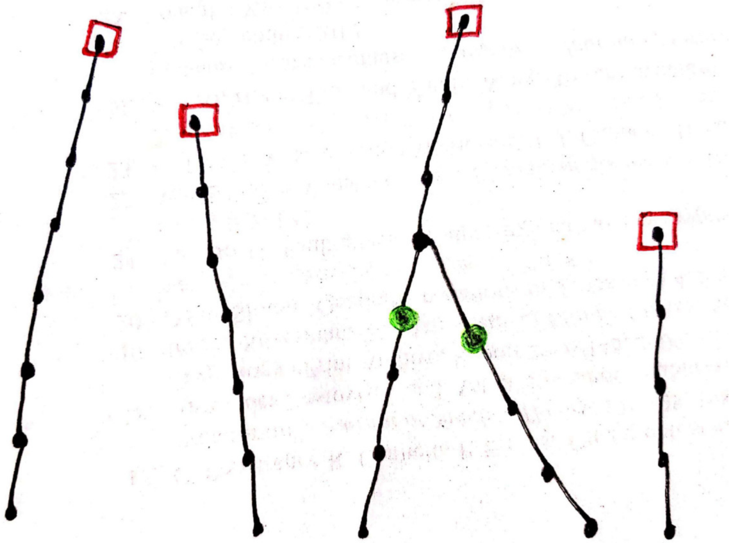


- Expected time $\approx \sqrt{\pi N/2} + \frac{2.5}{\theta}$, Space $\approx \theta\sqrt{\pi N/2}$.

Hash function collisions: VW

The VW algorithm for finding a collision for $H : \{0,1\}^\ell \to \{0,1\}^\ell$:

- Has expected running time $\sqrt{\pi 2^\ell/2} \approx 2^{\ell/2}$
- Is perfectly parallelizable
- Has negligible storage
- Has negligible communication costs



With $M = 2^{\ell/3}$ processors, a collision can be found in time $2^{\ell/6}$. (Grover's search takes time $2^{\ell/3}$.)

Hash function collisions: BHT

Brassard-Høyer-Tapp (BHT) (1998)

Fix $x_1, x_2, \ldots, x_N \in \{0,1\}^{\ell+c}$. Define $F : \{0,1\}^{\ell+c} \to \{0,1\}$ by

$$F(y) = \begin{cases} 1, & \text{if } H(y) = H(x_i) \text{ and } y \neq x_i \text{ for some } i, \\ 0, & \text{otherwise.} \end{cases}$$

Grover's search (one processor) finds a collision in time

$$N + 2^{\ell/2}/N^{1/2}.$$

If $N = 2^{\ell/3}$, this time is $2^{\ell/3}$.

Bernstein (2009) argued that BHT is inferior to VW since:

- ▶ Memory access is expensive (on the order of $N^{1/2}$).
- ▶ Quantum memory is expensive.

NIST Category 2

- Any attack must require computational resources comparable to or greater than those required for collision search on SHA-256.
- Cost metric: Number of gates
  - 2^{146} classical gates

- Category 1:
  - 2^{143} classical gates, 2^{170}/MAXDEPTH quantum gates.
- "...NIST will assume that the five security strengths are correctly ordered in terms of practical security."

- Category 4 (SHA-384):
  - 2^{210} classical gates

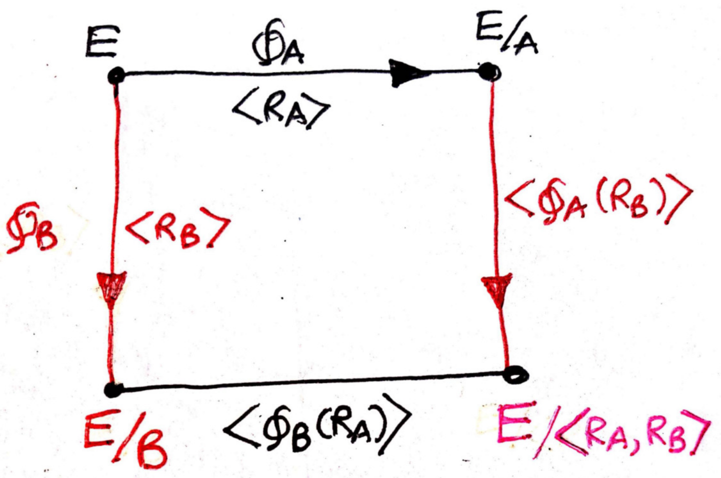
SIDH parameters

Unauthenticated key agreement scheme (Jao & De Feo, 2011).

- Let $p = 2^{e_A}3^{e_B} - 1$ be a prime with $2^{e_A} \approx 3^{e_B} \approx p^{1/2}$.
- Let E be a (supersingular) elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p+1)^2$.
- Then $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}_{p+1} \oplus \mathbb{Z}_{p+1}$, whence $E[2^{e_A}], E[3^{e_B}] \subseteq E(\mathbb{F}_{p^2})$. Let $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ be bases for $E[2^{e_A}], E[3^{e_B}]$.
- Write (ℓ, e) to mean either $(2, e_A)$ or $(3, e_B)$. Similarly for $\{P, Q\}$.
- For each order-ℓ^e subgroup S of $E[\ell^e]$, there exists a degree-ℓ^e (separable) isogeny $\phi_S : E \to E/S$ over $\mathbb{F}_{p^2}$ with kernel S. The isogeny is unique up to isomorphism and can be efficiently computed.
- Hence, the number of degree-ℓ^e isogenies $\phi : E \to E'$ is $(\ell+1)\ell^{e-1} \approx p^{1/2}$.
- SIDH parameters: $e_A, e_B, p, E, P_A, Q_A, P_B, Q_B$.

SIDH

1. Alice selects a random order-2^{e_A} point $R_A = m_A P_A + n_A Q_A$ and computes the isogeny $\phi_A : E \to E/A$, where $A = \langle R_A \rangle$. Alice transmits E/A, $\phi_A(P_B)$, $\phi_A(Q_B)$ to Bob.
2. Bob similarly transmits E/B, $\phi_B(P_A)$, $\phi_B(Q_A)$ to Alice.
3. Alice computes $\phi_B(R_A) = m_A \phi_B(P_A) + n_A \phi_B(Q_A)$ and $(E/B)/\langle \phi_B(R_A) \rangle$.
4. Similarly, Bob computes $(E/A)/\langle \phi_A(R_B) \rangle$.
5. The compositions of isogenies $E \to E/A \to (E/A)/\langle \phi_A(R_B) \rangle$ and $E \to E/B \to (E/B)/\langle \phi_B(R_A) \rangle$ have kernel $\langle R_A, R_B \rangle$.
6. The shared secret is the j-invariant of these curves.



CSSI

- Hardness of the Computational SuperSingular Isogeny problem (CSSI) is necessary for the security of SIDH:
- Given the SIDH parameters e_A, e_B, p, E, P_A, Q_A, P_B, Q_B, and E/A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute a degree-2^{e_A} isogeny $\phi_A : E \to E/A$.
- We will study a simplification of the problem that omits the auxiliary points $\phi_A(P_B)$ and $\phi_A(Q_B)$:

  CSSI: Given the SIDH parameters e_A, e_B, p, E, P_A, Q_A, P_B, Q_B, and E/A, compute a degree-2^{e_A} isogeny $\phi_A : E \to E/A$.

- CSSI was first formulated by Charles, Goren and Lauter in 2005.

Supersingular isogeny graphs

- Let R denote the set of all j-invariants of supersingular elliptic curves over $\mathbb{F}_{p^2}$; then $\#R \approx p/12 \approx \ell^{2e}$.
- The supersingular isogeny graph $\mathcal{G}_\ell(\mathbb{F}_{p^2})$ has vertex set R, and edges (j_1, j_2) with multiplicity equal to the multiplicity of j_2 as a root of the modular polynomial $\Phi_\ell(j_1, Z)$ over $\mathbb{F}_{p^2}$.
- $\mathcal{G}_\ell$ is $(\ell + 1)$-regular.
- Pizer showed that $\mathcal{G}_\ell(\mathbb{F}_{p^2})$ is a Ramanujan graph:
  - Optimal expander graph.
  - The endpoint of a random walk approximates the uniform distribution after $O(\log v)$ steps, where $v \approx \ell^{2e}$.

- Let $E_1 = E, \quad j_1 = j(E_1), \qquad E_2 = E/A, \quad j_2 = j(E_2)$.
- The CSSI problem is to find a path of length e from j_1 to j_2 in $\mathcal{G}_\ell(\mathbb{F}_{p^2})$.

CSSI attacks

The fastest CSSI attacks that were first identified were:

- Classical: Meet-in-the-middle $O(p^{1/4})$.
- Quantum: Tani's algorithm $O(p^{1/6})$.

Consequently, primes p of bitlength ≈ 768 were recommended to attain the 128-bit security level.

However, both attacks have significant storage requirements: $p^{1/4}$ and $p^{1/6}$, respectively.

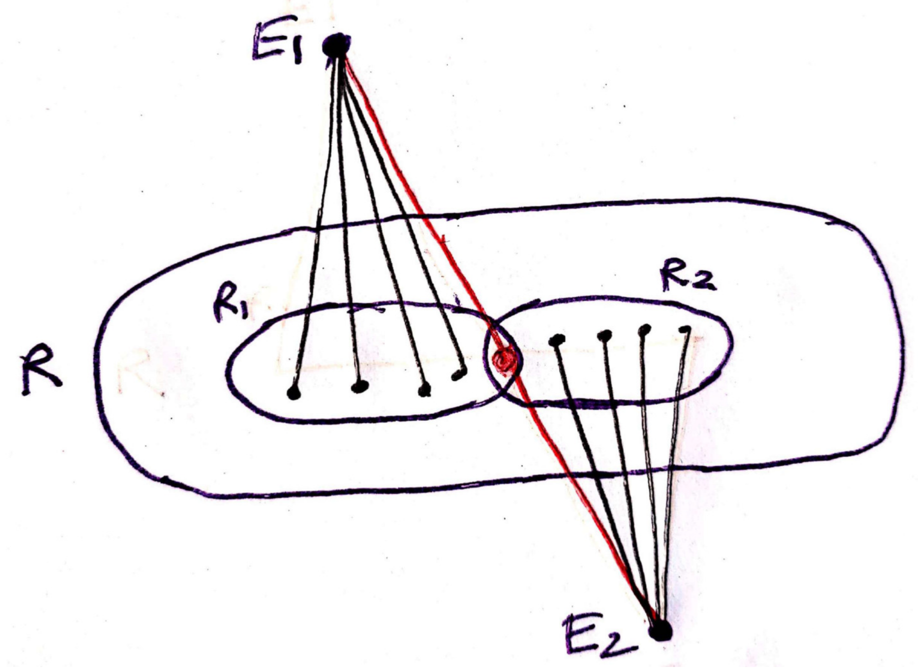
Thus, a concrete cost analysis might justify using smaller p while still attaining the 128-bit security level.

Meet-in-the-middle (MITM) attack

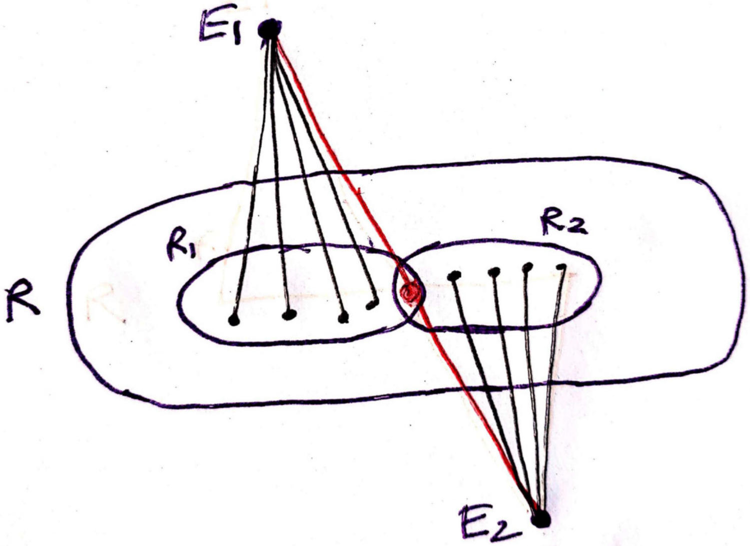
- Denote the number of order-$\ell^{e/2}$ subgroups of $E[\ell^e]$ by $N \approx p^{1/4}$.
- For $i = 1, 2$, let R_i denote that set of j-invariants of elliptic curves over $\mathbb{F}_{p^2}$ that are $\ell^{e/2}$-isogenous to E_i.
- Then one expects that $\#R_1 \approx \#R_2 \approx N \ll \#R$.
  It is also reasonable to assume that $\#(R_1 \cap R_2) = 1$.

MITM

Time: $2N$
Space: N



VW golden collision finding



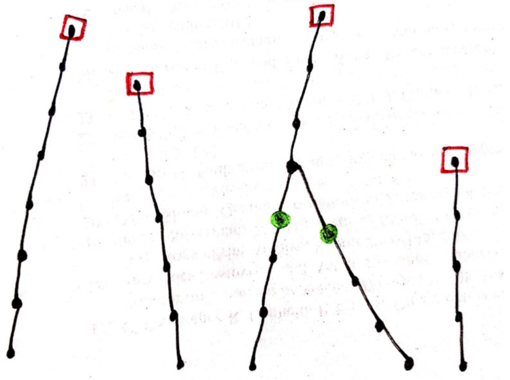
van Oorschot & Wiener, 1996

Adj et al., 2018

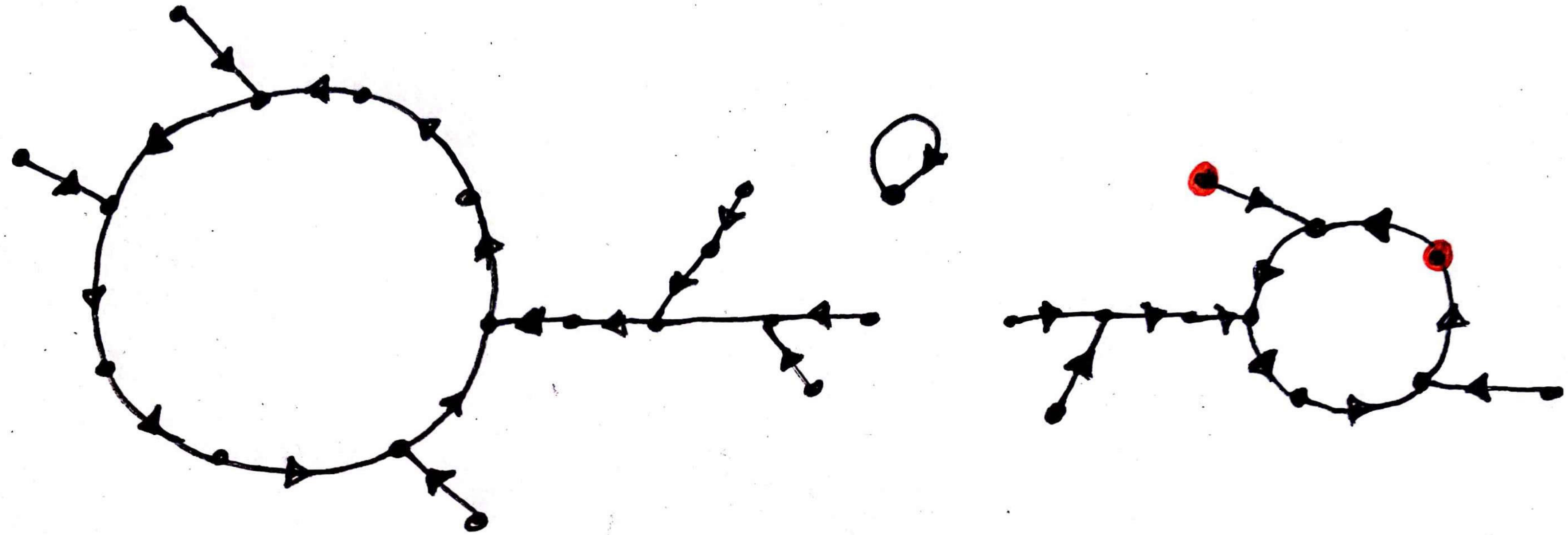
- Let $I = \{1, 2, \ldots\ldots, N\}$ and $S = \{1, 2\} \times I$.
- For $i = 1, 2$, let: $\mathcal{A}_i$ = all order-$\ell^{e/2}$ subgroups of $E_i[\ell^e]$.
  $h_i : I \to \mathcal{A}_i$ bijections.
  $f_i : \mathcal{A}_i \to R_i$, $f_i(A_i) = j(E_i/A_i)$.
- Let $g : R \to S$ be a random function
- Define $f : S \to S$ by $f : (i, y) \mapsto g(f_i(h_i(y)))$
- The expected number of (unordered) collisions for f is $\approx N$.
- Suppose $j(E_1/A_1) = j(E_2/A_2)$, $y_1 = h_1^{-1}(A_1)$, $y_2 = h_2^{-1}(A_2)$.
- We seek the golden collision $(1, y_1)$, $(2, y_2)$.

VW golden collision finding

Main idea: Find many collisions, until the gold. collision is obtained.



Problem: The golden collision might be hard to find.



Solution: Change f periodically (by changing g).

Finding the golden collision

- Storage: Space for w triples $(x_{i,a}, a, x_{i,0})$.
- Set $\theta = \alpha\sqrt{w/(2N)}$.
- Use each version of f to produce βw distinguished points.
- Store a distinguished point in a memory cell determined by hashing it.
- For $\alpha = 2.25$, $\beta = 10$:
  - One expects $1.3w$ collisions per function version.
  - One expects $1.1w$ distinct collisions per function version.
  - The expected time to find the golden collision is

$$\approx \frac{N}{1.1w} \cdot 10w \cdot \frac{2N}{2.25\sqrt{w}} \approx N^{3/2}/w^{1/2} \quad \approx p^{3/8}/w^{1/2}.$$

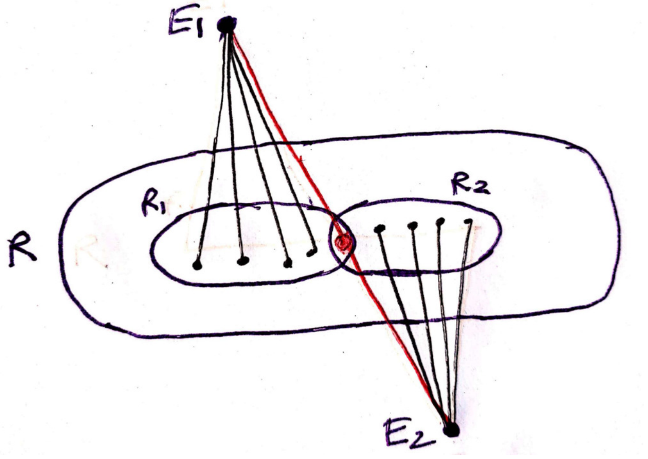
- The algorithm parallelizes well.

MITM vs. VW

- MITM (time-memory tradeoff):
  Time: N^2/w Space: w

- VW golden collision search:
  Time: $N^{3/2}/w^{1/2}$ Space: w

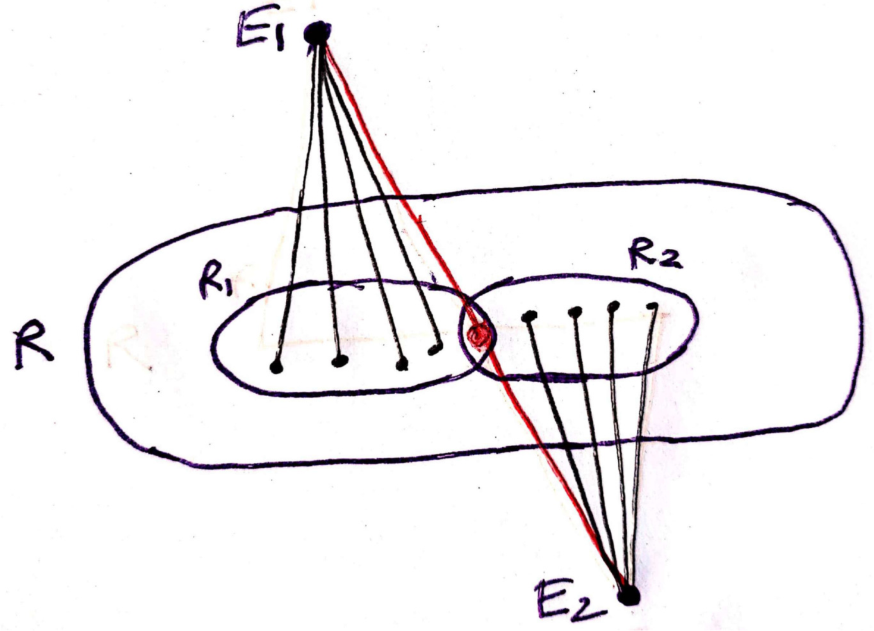
- Conclusion: VW is superior to MITM for $w < N$.



Quantum attacks

CSSI can be viewed as an instance of the claw finding problem:

Consider $f : X \to Z$, $g : Y \to Z$ with $|X| = |Y| = N \ll |Z|$. Given black-box access to f and g, find $(x, y) \in X \times Y$ with $f(x) = g(y)$.



In CSSI: X = degree-$\ell^{e/2}$ isogenies originating at E_1,

Y = degree-$\ell^{e/2}$ isogenies originating at E_2,

Z = set of j-invariants of all supersingular elliptic curves,

f, g record the j-invariants of the image curves,

and there is exactly one claw.

Grover's search

- Define $F : X \times Y \to Z$ by $F(x, y) = 1$ if $f(x) = g(y)$, and $F(x, y) = 0$ otherwise.
- Grover's search can be used to find a claw in time $\sqrt{N^2} \approx p^{1/4}$.
- VW: $N^{3/2}/(Mw^{1/2})$, Grover: $N/\sqrt{M}$.

- Example: Consider $\ell = 2, \quad e = 216, \quad N \approx 2^{108}, \quad w = 2^{80}$, MAXDEPTH=$2^{64}$.
  - Then VW total run time is $2^{125.7}$ degree-2^{108} isogeny computations.
  - An optimistic estimate for the depth of a quantum circuit for a degree-2^{108} isogeny computation is 2^{14}.
  - One quantum circuit can perform 2^{50} isogeny computations, so $M = 2^{116}$ circuits are required for Grover.
  - So, NIST's Category 1 requirements are met.

Tani's algorithm

- The vertices of the Johnson graph $J(X,T)$ are the T-subsets of X, with two subsets begin adjacent iff their intersection has size $T-1$.
- Tani: Perform a quantum random walk (with uniform probabilities) in $G = J(X,T) \times J(Y,T)$.
- The walk on G is a Markov process with uniform probabilities and spectral gap $\delta \approx \frac{1}{T}$.
- The proportion of vertices that contain a claw is

$$\epsilon = \left(\frac{\binom{N-1}{T-1}}{\binom{N}{T}} \right)^2 = \frac{T^2}{N^2}.$$

Quantum random walk

Classical:

```
Construct a random vertex.    (S)
Repeat $O(\frac{1}{\epsilon})$ times:
        Repeat $O(\frac{1}{\delta})$ times:
                Take one random step in $G$.    (U)
        Check if the current vertex contains a claw.    (C)
```

Cost: $O\left(S + \frac{1}{\epsilon}\left(\frac{1}{\delta}U + C\right)\right)$.

Quantum (Magniez-Nayak-Roland-Santha):

```
Create a superposition of random vertices.    (S)
Repeat $O(\frac{1}{\sqrt{\epsilon}})$ times:
        Repeat $O(\frac{1}{\sqrt{\delta}})$ times:
                Take one “quantum” random step in $G$.    (U)
        “Quantum” check for a claw.    (C)
```

Cost: $O\left(S + \frac{1}{\sqrt{\epsilon}}\left(\frac{1}{\sqrt{\delta}}U + C\right)\right)$.

Tani: query optimal

Cost: $O\left(S + \frac{1}{\sqrt{\epsilon}}\left(\frac{1}{\sqrt{\delta}}U + C\right)\right), \quad \epsilon = \frac{T^2}{N^2}, \quad \delta \approx \frac{1}{T}$.

Jaques & Schanck (CRYPTO 2019)

- Cost = $O(T + \frac{N}{T^{1/2}})$.
- The cost is optimized when $T \approx N^{2/3}$, yielding a running time $\approx N^{2/3} = p^{1/6}$ degree-$\ell^{e/2}$ isogeny computations.
- A vertex has size $2T$, so $p^{1/6}$ classical processors are needed in the active error control model.
- These $p^{1/6}$ processors (and $p^{1/6}$ classical memory) can be used with VW golden collision search with running time

$$\frac{p^{3/8}}{p^{1/6} \cdot p^{1/12}} = p^{1/8}.$$

Tani: Non-asymptotic cost estimates

Jaques & Schanck (CRYPTO 2019)

- The optimal T is chosen based on memory access costs and oracle costs.
- Tani suffers from the same parallelization issues as Grover (however, the naive parallelization strategy may not be optimal).
- Note that Tani's algorithm with $T = 1$ is essentially the same as Grover's algorithm.

- Conclusion: Tani is costlier than VW
  - with MAXDEPTH = 2^{64}
  - DW-cost
  - G-cost

Concrete parameters for SIDH

- 128-bit security-level (also: NIST Categories 1 and 2)
  - $p = p434 = 2^{216}3^{137} - 1.$
  - VW: $w = 2^{80}$, $\theta \approx 1/2^{13.6}$, Time = $2^{125.7}$ (isog.).

Protocol phase		CLN + enhancements	
		p_{751}	p_{434}
Key Gen.	Alice	26.9	5.3
	Bob	30.5	6.0
Key Gen.	Alice	24.9	5.0
	Bob	28.6	5.8

(Times are in 10^6 clock cycles on an Intel Core i7-6700)

- 192-bit security level (also: NIST Categories 3 and 4)
  - $p = p610 = 2^{305}3^{192} - 1.$
  - VW: $w = 2^{80}$, $\theta \approx 1/2^{35.9}$, Time = $2^{192.6}$ (isog.).
- $p434$ and $p610$ have been included in the Round 2 SIKE submission to the NIST PQC competition.

Questions

- Can the analysis of VW golden collision finding be made more rigorous?
- Can the CSSI problem be formulated as one of finding a single collision (not a golden collision)?
- Are the assumptions on classical resources and quantum resources reasonable for making long-term key-size recommendations?
- Can Tani's algorithm be parallelized in a cost-effective way?

References

1. G. Adj et al.
   “On the cost of computing isogenies between supersingular elliptic curves”
   SAC 2018.

2. S. Jaques and J. Schanck
   “Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE”
   CRYPTO 2019.

3. S. Jaques
   “Quantum cost models for cryptanalysis of isogenies”
   Master’s thesis, `http://hdl.handle.net/10012/14612`