Supersingular Isogeny Key Encapsulation (SIKE)

Yannick Bormuth, Dario Kermanschah,
Michael Burkhalter, Lauro Böni

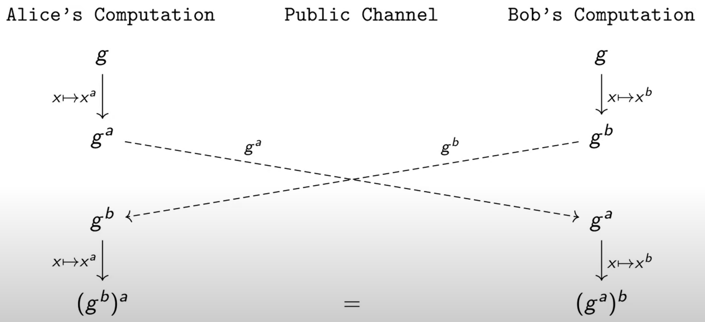
Krypt Det

Bern, 3. September 2020

Overview

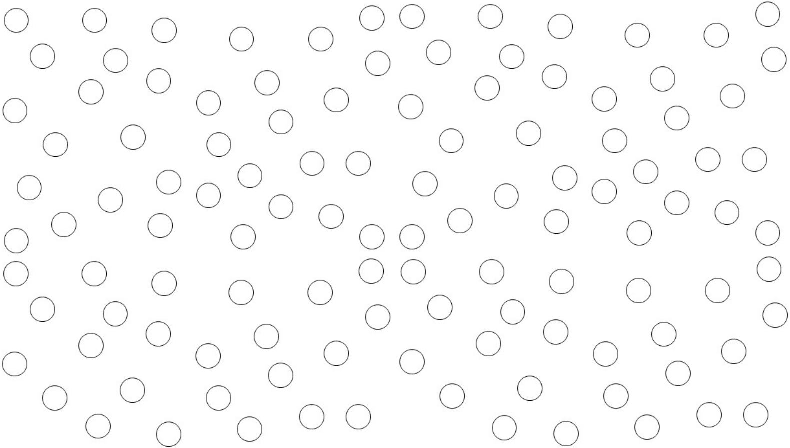
Recap: Classical Diffie-Hellman (DH) protocol

Setup: Fix a group G and $g \in G$.

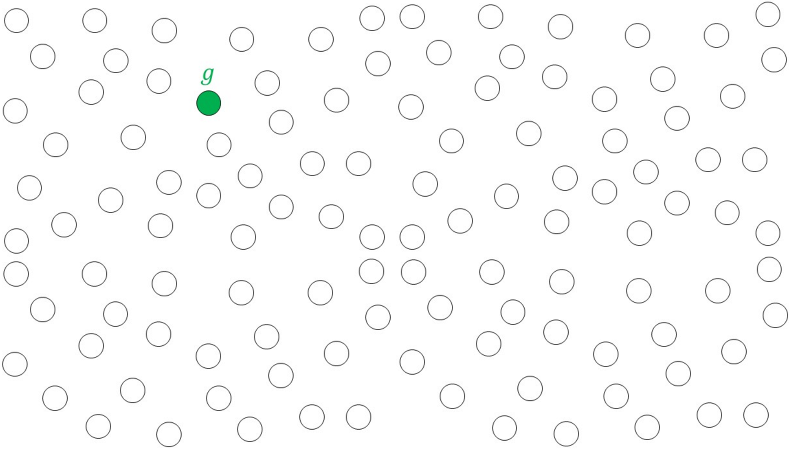


Hard problem: Given g, g^a, and g^b, determine g^{ab}.

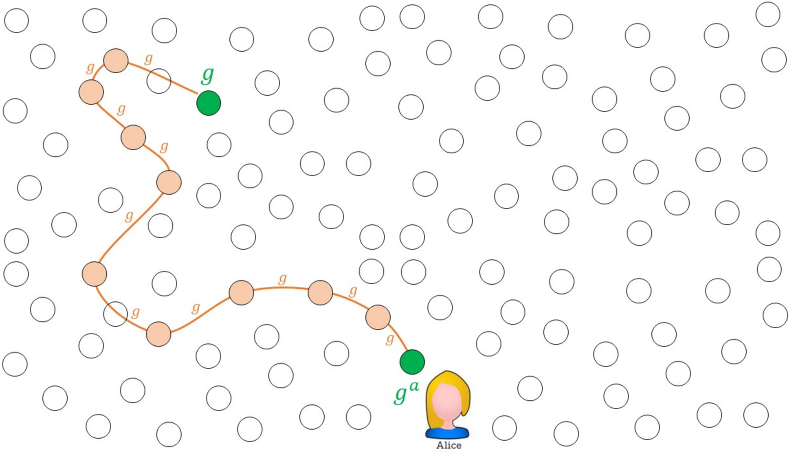
Recap: Classical Diffie-Hellman (DH) protocol



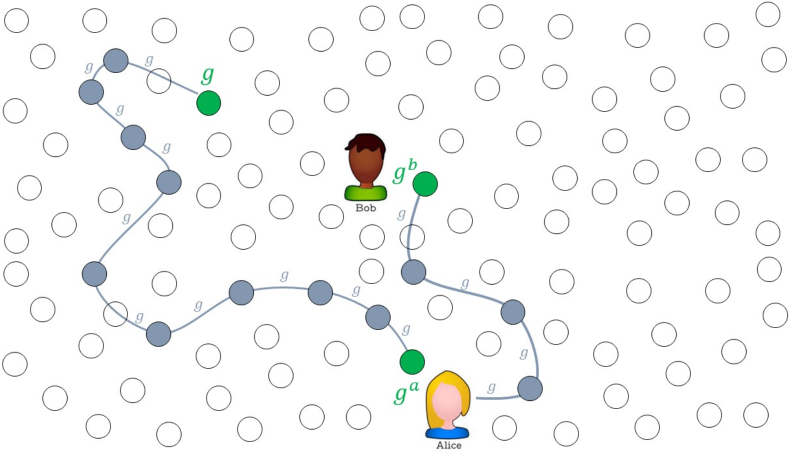
Recap: Classical Diffie-Hellman (DH) protocol



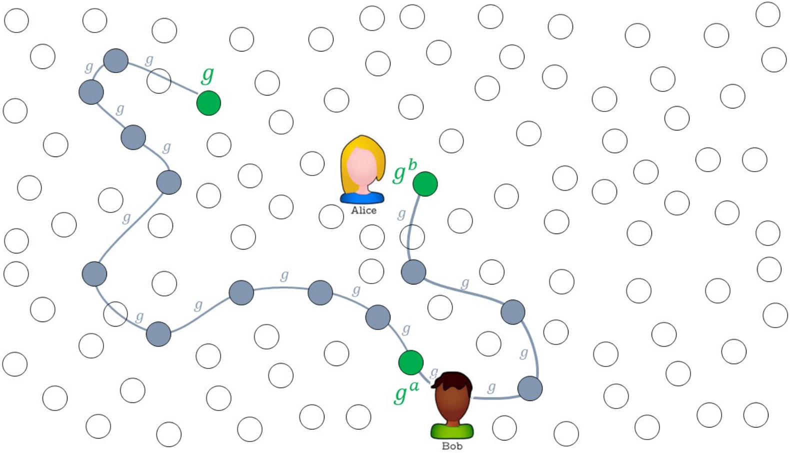
Recap: Classical Diffie-Hellman (DH) protocol



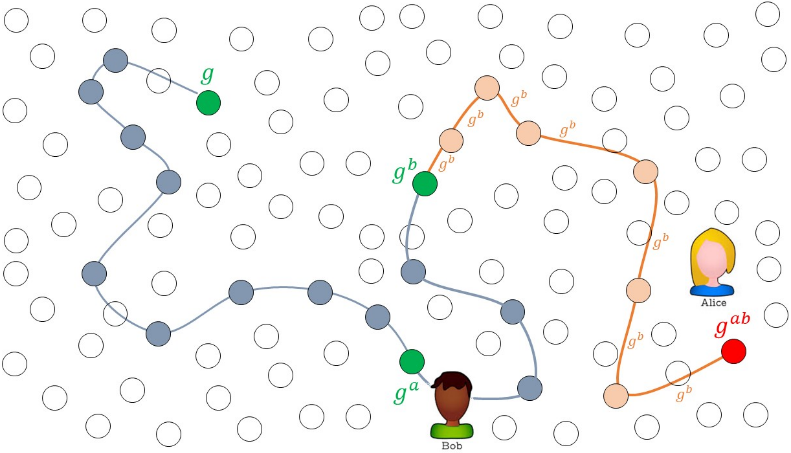
Recap: Classical Diffie-Hellman (DH) protocol



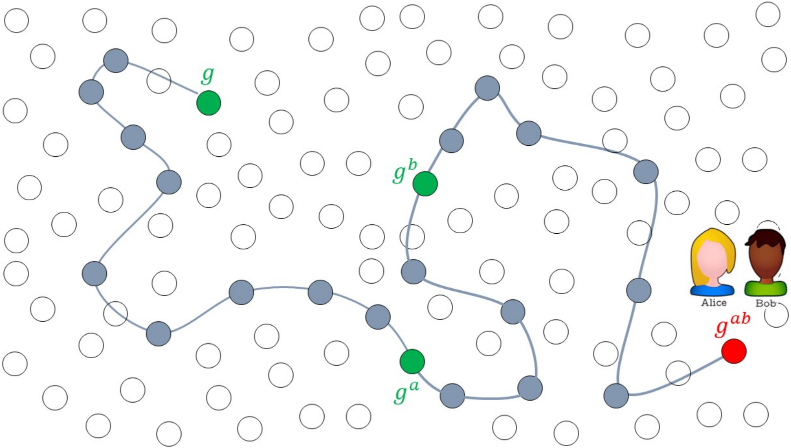
Recap: Classical Diffie-Hellman (DH) protocol



Recap: Classical Diffie-Hellman (DH) protocol



Recap: Classical Diffie-Hellman (DH) protocol



From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

Let us turn our attention to the SIDH protocol:

From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

Let us turn our attention to the SIDH protocol:

- We are working on $\mathbb{F}_{p^2}$ for some prime p of the form

$$p = 2^{e_A} 3^{e_B} - 1.$$

From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

Let us turn our attention to the SIDH protocol:

- We are working on $\mathbb{F}_{p^2}$ for some prime p of the form

$$p = 2^{e_A} 3^{e_B} - 1.$$

- We consider the set of all **supersingular** Elliptic Curves and fix an initial curve E_0.

From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

Let us turn our attention to the SIDH protocol:

- We are working on $\mathbb{F}_{p^2}$ for some prime p of the form

$$p = 2^{e_A} 3^{e_B} - 1.$$

- We consider the set of all **supersingular** Elliptic Curves and fix an initial curve E_0.
  - leads to a *directed* and *regular* graph
  - harder problem than non-supersingular (i.e. ordinary)

From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

Let us turn our attention to the SIDH protocol:

- We are working on $\mathbb{F}_{p^2}$ for some prime p of the form

$$p = 2^{e_A} 3^{e_B} - 1.$$

- We consider the set of all **supersingular** Elliptic Curves and fix an initial curve E_0.
  - leads to a *directed* and *regular* graph
  - harder problem than non-supersingular (i.e. ordinary)
- Alice
  - P_A, Q_A such that $\langle P_A, Q_A \rangle = E[2^{e_A}] \cong \mathbb{Z}_{2^{e_A}} \times \mathbb{Z}_{2^{e_A}}$.
  - Computes $S_A = P_A + [k_A]Q_A$ (Note: S_A has order 2^{e_A})

From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

Let us turn our attention to the SIDH protocol:

- We are working on $\mathbb{F}_{p^2}$ for some prime p of the form
$$p = 2^{e_A}3^{e_B} - 1.$$
- We consider the set of all **supersingular** Elliptic Curves and fix an initial curve E_0.
  - leads to a *directed* and *regular* graph
  - harder problem than non-supersingular (i.e. ordinary)
- Alice
  - P_A, Q_A such that $\langle P_A, Q_A \rangle = E[2^{e_A}] \cong \mathbb{Z}_{2^{e_A}} \times \mathbb{Z}_{2^{e_A}}$.
  - Computes $S_A = P_A + [k_A]Q_A$ (Note: S_A has order 2^{e_A})
- Bob
  - P_B, Q_B such that $\langle P_B, Q_B \rangle = E[3^{e_B}] \cong \mathbb{Z}_{3^{e_B}} \times \mathbb{Z}_{3^{e_B}}$.
  - Computes $S_B = P_B + [k_B]Q_B$ (Note: S_B has order 3^{e_B})

From DH to Supersingular Isogeny Diffie-Hellman (SIDH)

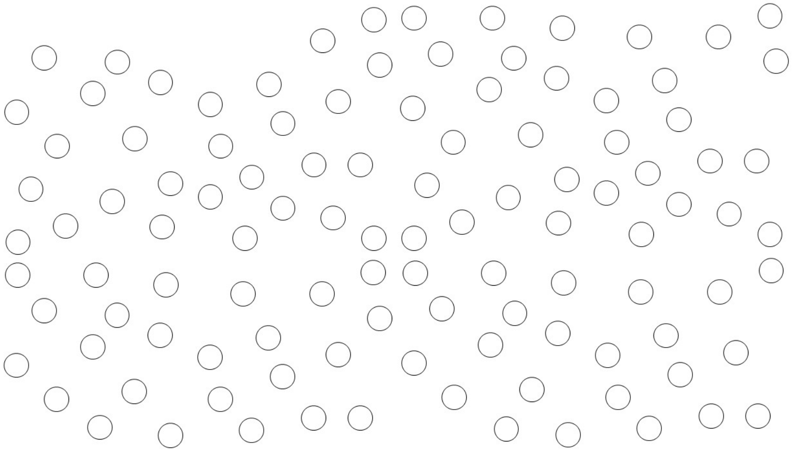
Let us turn our attention to the SIDH protocol:

- We are working on $\mathbb{F}_{p^2}$ for some prime p of the form

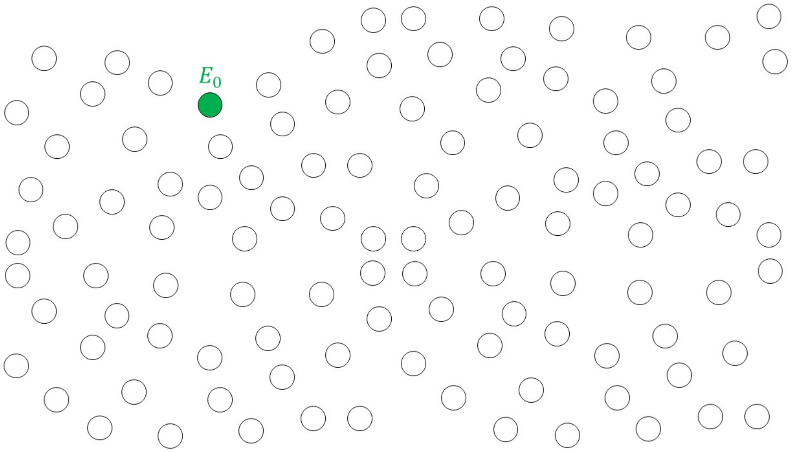
$$p = 2^{e_A} 3^{e_B} - 1.$$

- We consider the set of all **supersingular** Elliptic Curves and fix an initial curve E_0.
    - leads to a *directed* and *regular* graph
    - harder problem than non-supersingular (i.e. ordinary)
- Alice
    - P_A, Q_A such that $\langle P_A, Q_A \rangle = E[2^{e_A}] \cong \mathbb{Z}_{2^{e_A}} \times \mathbb{Z}_{2^{e_A}}$.
    - Computes $S_A = P_A + [k_A] Q_A$ (Note: S_A has order 2^{e_A})
- Bob
    - P_B, Q_B such that $\langle P_B, Q_B \rangle = E[3^{e_B}] \cong \mathbb{Z}_{3^{e_B}} \times \mathbb{Z}_{3^{e_B}}$.
    - Computes $S_B = P_B + [k_B] Q_B$ (Note: S_B has order 3^{e_B})

Supersingular Isogeny Diffie-Hellman (SIDH) protocol

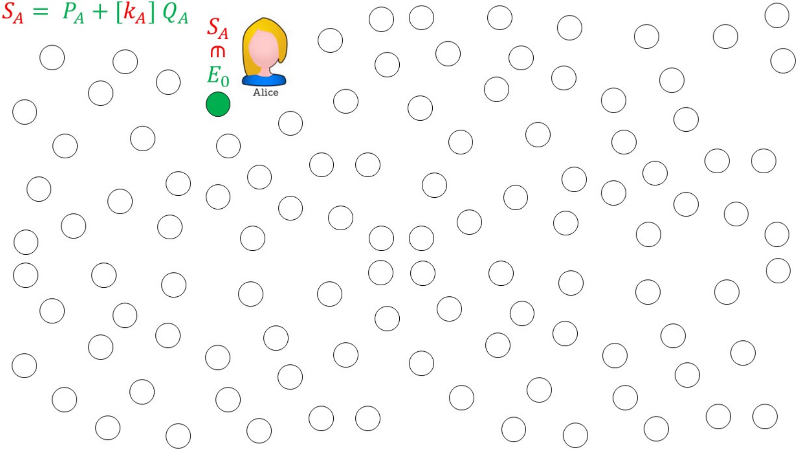


Supersingular Isogeny Diffie-Hellman (SIDH) protocol

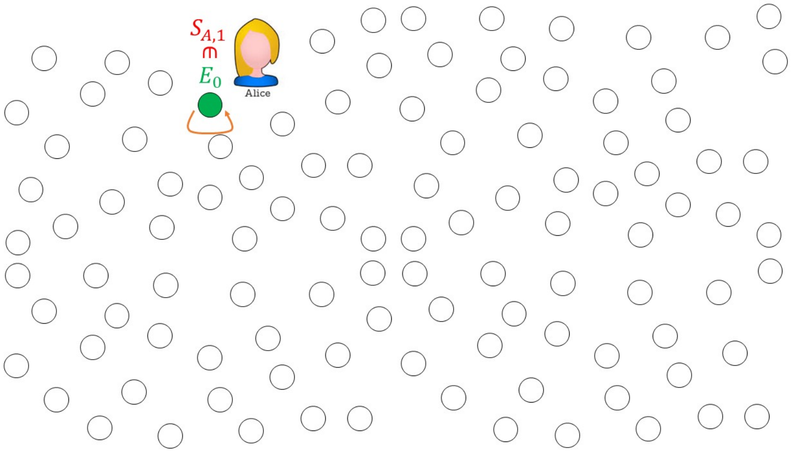


Supersingular Isogeny Diffie-Hellman (SIDH) protocol

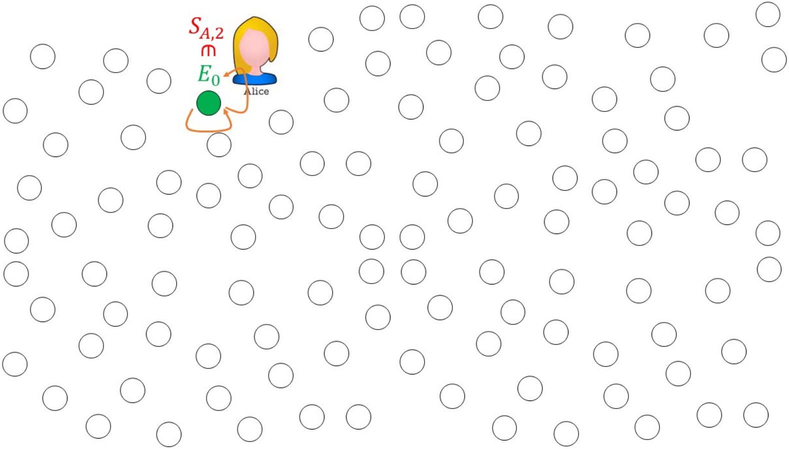
$S_A = P_A + [k_A]\, Q_A$



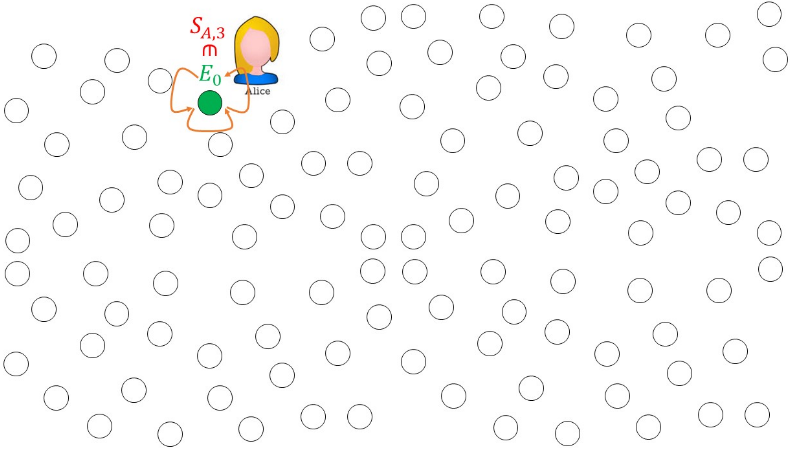
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



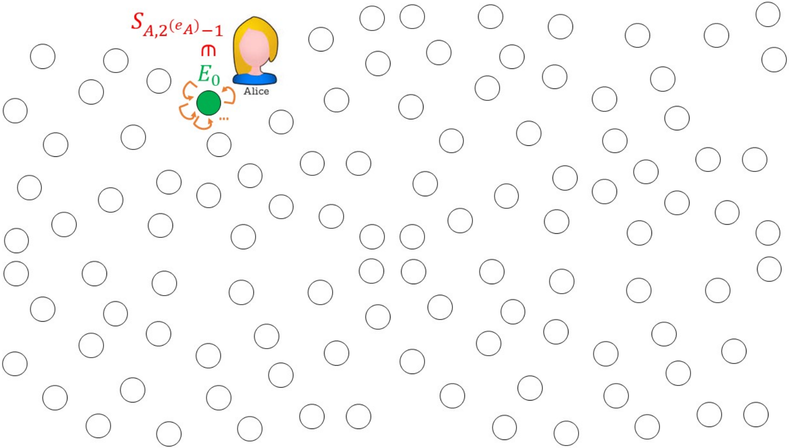
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



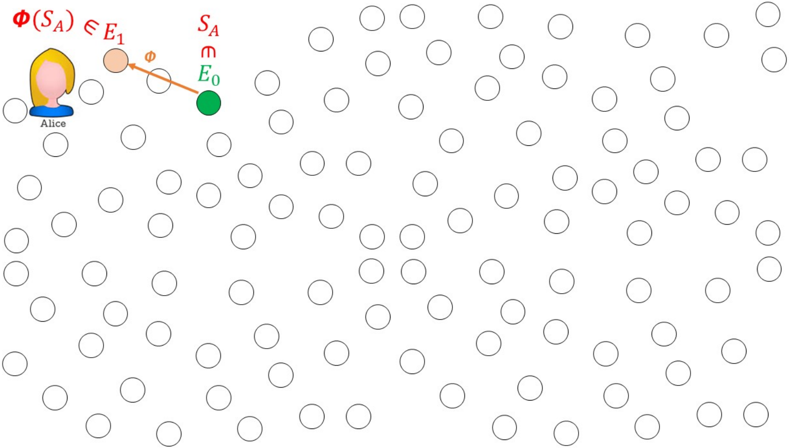
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



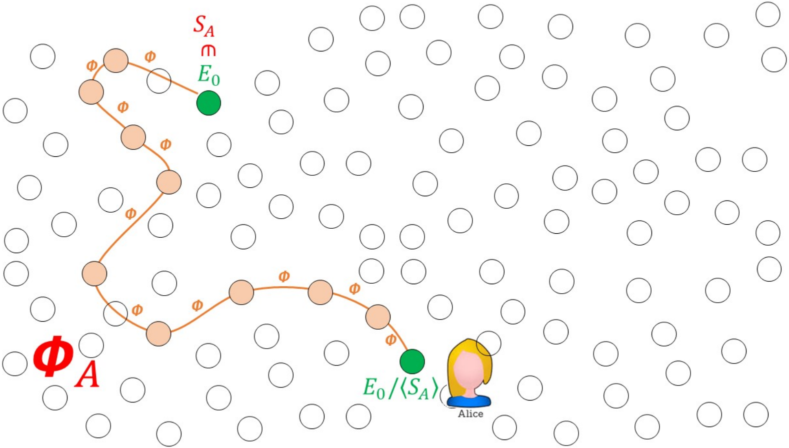
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



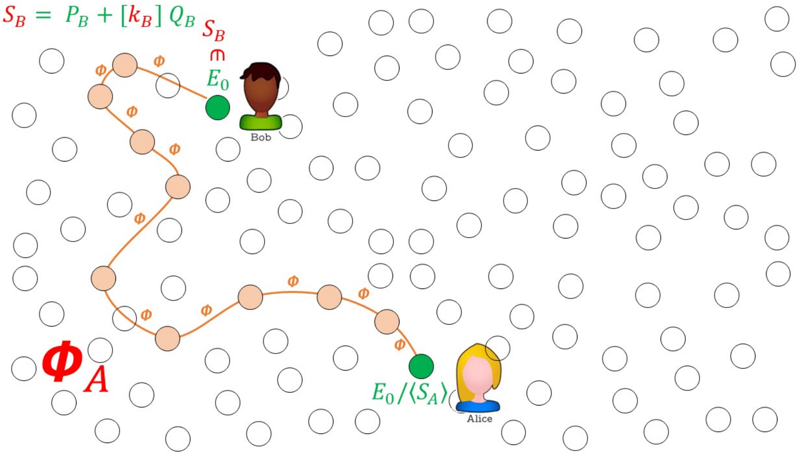
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



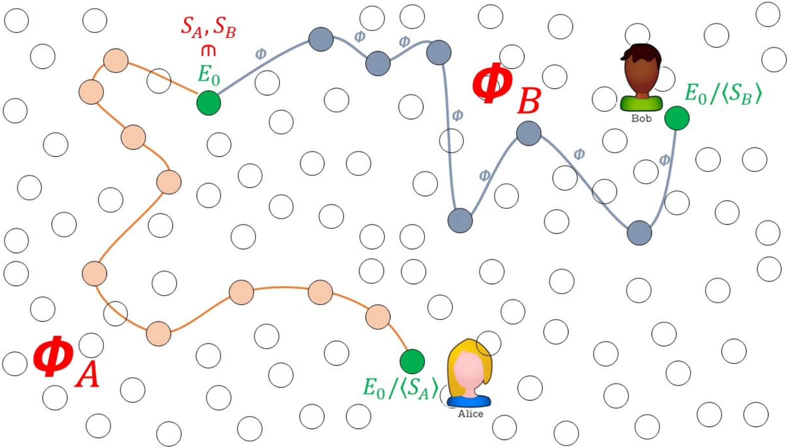
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



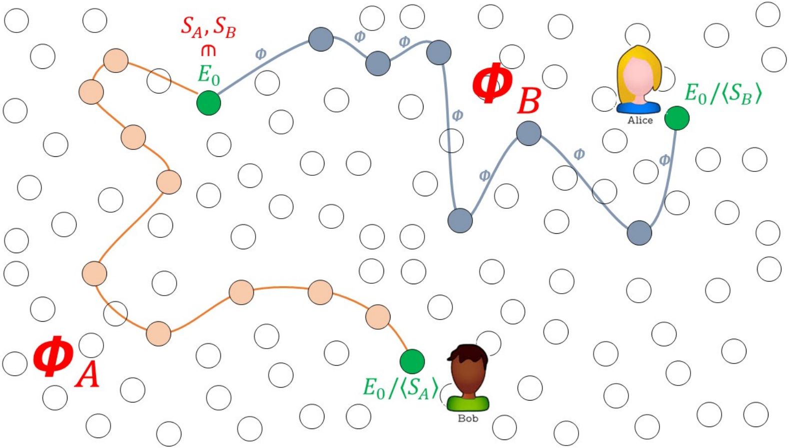
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



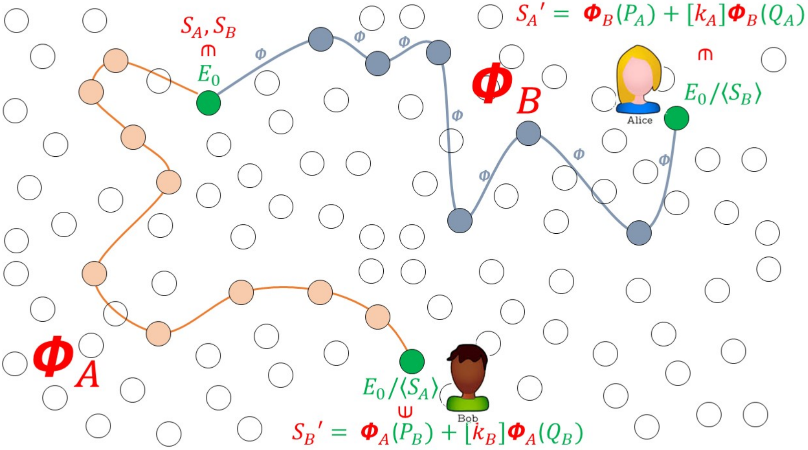
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



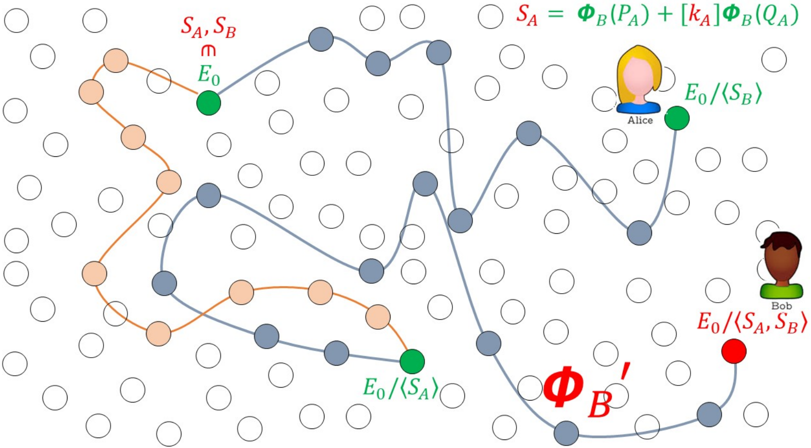
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



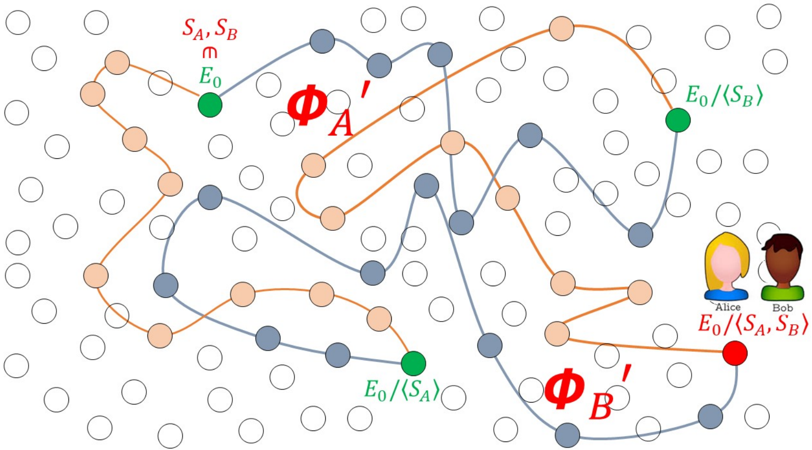
Supersingular Isogeny Diffie-Hellman (SIDH) protocol



Supersingular Isogeny Diffie-Hellman (SIDH) protocol

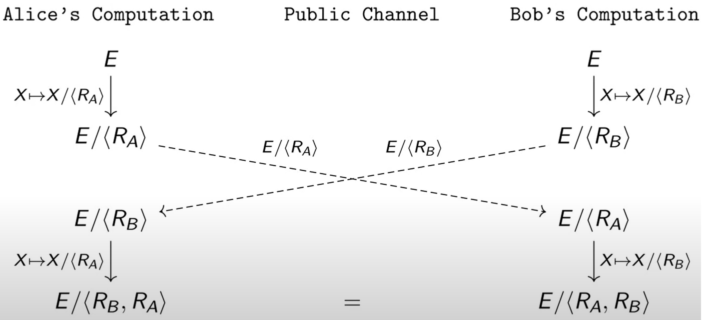


Supersingular Isogeny Diffie-Hellman (SIDH) protocol



Supersingular Isogeny Diffie-Hellman (SIDH) protocol

Setup: Fix a supersingular isogeny class $\mathcal{C}$ and $E \in \mathcal{C}$.



Hard problem: Given E, $E/\langle R_A \rangle$, $E/\langle R_B \rangle$ *, determine $E/\langle R_A, R_B \rangle$.
* Some extra information is also available.

From SIDH to SIKE

- SIKE stands for **Supersingular Isogeny Key Encapsulation**.

From SIDH to SIKE

- SIKE stands for **Supersingular Isogeny Key Encapsulation**.
- It is motivated by a potential security flaw in SIDH, where Bob can reconstruct Alice's secret key k_A.

From SIDH to SIKE

- SIKE stands for **Supersingular Isogeny Key Encapsulation**.
- It is motivated by a potential security flaw in SIDH, where Bob can reconstruct Alice's secret key k_A.

In a nutshell

$$\begin{aligned} \text{SIKE} &= \text{SIDH} \\ &\quad + \text{some mechanism preventing Bob} \\ &\qquad \text{from fooling Alice} \\ &\quad + \text{compression.} \end{aligned}$$

From SIDH to SIKE

- SIKE stands for **Supersingular Isogeny Key Encapsulation**.
- It is motivated by a potential security flaw in SIDH, where Bob can reconstruct Alice's secret key k_A.

In a nutshell

$$\begin{aligned} \text{SIKE} \quad &= \quad \text{SIDH} \\ &\quad + \text{some mechanism preventing Bob} \\ &\qquad \text{from fooling Alice} \\ &\quad + \text{compression.} \end{aligned}$$

Consequence:

- Protocol is no longer symmetric.
- Allows using key pairs more than once.

From SIDH to SIKE

- SIKE stands for **Supersingular Isogeny Key Encapsulation**.
- It is motivated by a potential security flaw in SIDH, where Bob can reconstruct Alice's secret key k_A.

In a nutshell

$$\begin{aligned} \text{SIKE} &= \text{SIDH} \\ &\quad + \text{some mechanism preventing Bob} \\ &\qquad \text{from fooling Alice} \\ &\quad + \text{compression.} \end{aligned}$$

Consequence:

- Protocol is no longer symmetric.
- Allows using key pairs more than once.

Practical implementation

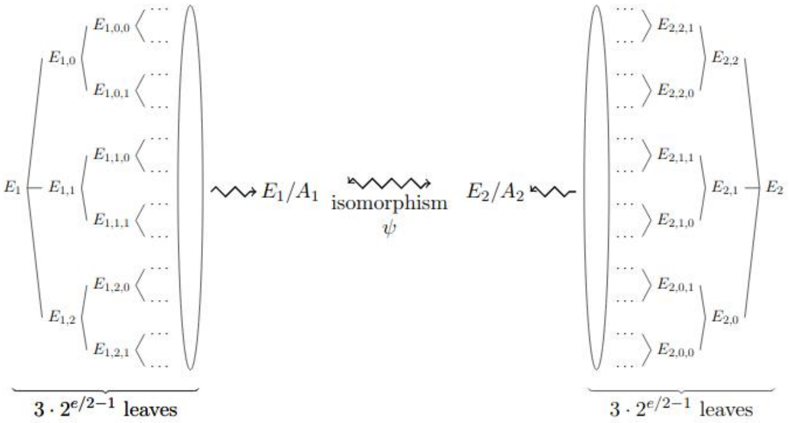
Meet-In-The-Middle

Underlying Math Problem:

Given public parameters $l_A, l_B, e_A, e_B, p, E, P_A, Q_A$ and $E/\langle S_A \rangle$: Compute the $l_A^{e_A}$-isogeny $E \to E/\langle S_A \rangle$

- e_A steps in the l_A-isogeny graph are much fewer than the average number of steps necessary to join any two nodes
- Very likely that the e_A steps represent the shortest path between E and $E/\langle S_A \rangle$
- Build list of all destination nodes taking $e_A/2$ steps from E
- For each destination of length-$e_A/2$ walks from $E/\langle S_A \rangle$, compare to list until match is found

Schematic Of Meet-In-The-Middle Attack



Costs of Classical Attacks

- Classical run time $\mathcal{O}(p^{1/4})$
- $\mathcal{O}(p^{1/4})$ memory needed to build all walks from E
- Smallest SIKE prime has 434 bits makes memory needs prohibitively large
- Technical enhancements give slower algorithms when memory is limited (e.g. to $\sim 2^{80}$)

Costs of Classical Attacks

- Classical run time $\mathcal{O}(p^{1/4})$
- $\mathcal{O}(p^{1/4})$ memory needed to build all walks from E
- Smallest SIKE prime has 434 bits makes memory needs prohibitively large
- Technical enhancements give slower algorithms when memory is limited (e.g. to $\sim 2^{80}$)

EXPONENTIAL IN TIME AND SPACE

PQC security definition

NIST security strength categories

Computational resources required to break security definition

$\geq$

resources for key/collision search on AES/SHA3

NIST level	classical gates	reference algorithms	factoring	discrete logarithm key	discrete logarithm group	Elliptic curve	SIKE
1	2^{143}	AES-128	3 072	256	3 072	256	SIKEp434
3	2^{207}	AES-192	7 680	384	7 680	384	SIKEp610
5	2^{272}	AES-256	15 360	512	15 360	512	SIKEp751

Quantum complexity is . . .

- expressed in terms of classical gates
- based on NIST's restriction on a maximal running time of a quantum circuit

Performance & resources

Comparison

classical Elliptic Curve with 256-bit prime $\iff$ `SIKEp434`
(both corresponding to security level 1, `AES-128`)

	prime bits	secret key bytes	public key bytes	shared secret bytes	cycles
EC	256	32	64	64	$\sim 4\,000\,000$
SIKE	434	330	374	16	$\sim 25\,000\,000$

executed on a 2.7 GHz Intel Core i5-5350U (Broadwell) processor

other resources for SIKE protocol

- between $O(10^7)$ and $O(10^8)$ cycles
- timings of $O(1)$ ms
- 70-80 mW energy consumption (on efficient ARM M4-Cortex processor)

The race for a new quantum-safe standard

What position does SIKE take?

small key sizes

- 564B public keys/48B private keys (for security level 5) compared to kB/MB range for other quantum-safe protocols

increased runtime by a factor of around 100

- seconds instead of miliseconds

Reason for SIKE to still be in the race

- EC theory well-proven in crytographic theory
- quantum attack algorithms not yet investigated enough
- desire for broad range of hardness assumption