Post-Quantum Cryptography in ARM Processors

Reza Azarderakhsh, Ph.D.

Florida Atlantic University and PQSecure Technologies

ARM Research Summit

Austin, TX

Sept. 2019

Quantum Threat to Information Security

PQ Secure

Large-scale quantum computers could break some encryption schemes

Need to migrate encryption to quantum-resistant algorithms

When we should start the process?

Questions:

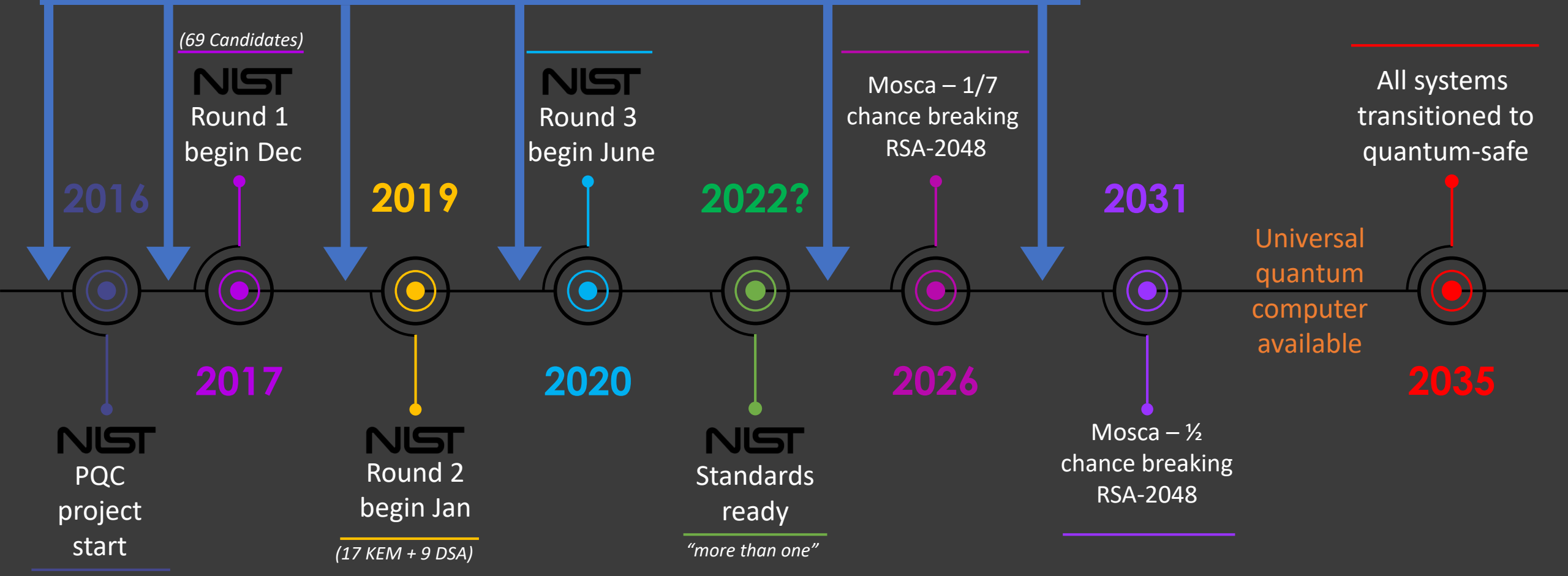
- When will quantum computers arrive?
- When will they be a threat?
- When do we switch?
- How do we have live and agile transition?
- What do we transition to?
- …

“In 25 years we’ll be working on quantum resistant crypto standardization based on elliptic curves in some way!”

—Dustin Moody, NIST (NSF CCC Workshop 2019)

PQ Secure

Retroactive decryption:
Record encrypted communication now,
Decrypt once quantum computers are available



Open Questions about Post-Quantum Cryptography

- Design better post-quantum cryptosystems
- Improve classical and quantum attacks
- Pick parameter sizes
- Develop fast, efficient, and secure implementations
- Integrate them into the existing infrastructures

PQ Secure

Post-Quantum Key-Exchange

- Lattice-based
- Code-based
- Isogeny-based

Post-Quantum Signatures

- Lattice-based
- Hash-based
- Multivariate-based
- Zero-Knowledge based

Quick Overview

PQ Secure

- **[2006]:** Birth of a supersingular isogeny-based cryptosystem
  - **Charles** – **Goren** – **Lauter**
  - built hash function from supersingular isogeny graph
- **[2011]:** Supersingular isogeny key exchange
  - Jao – De Feo
- **[2017]:** Supersingular isogeny key encapsulation
  - SIKE Team

SIKE Team


UNIVERSITY OF
WATERLOO


UNIVERSITÉ DE
VERSAILLES
SAINT-QUENTIN-EN-YVELINES


FAU
FLORIDA ATLANTIC
UNIVERSITY


IN·DEI·NOMINE·FELICITER
Radboud
Universiteit


UNIVERSITY OF
TORONTO


Microsoft Research


amazon


LinkedIn


TEXAS
INSTRUMENTS


INFOSEC
GLOBAL

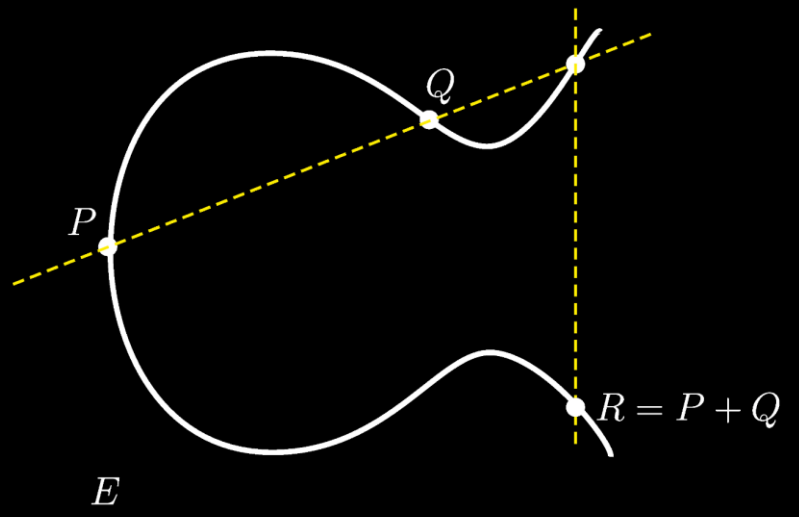

PQSecure

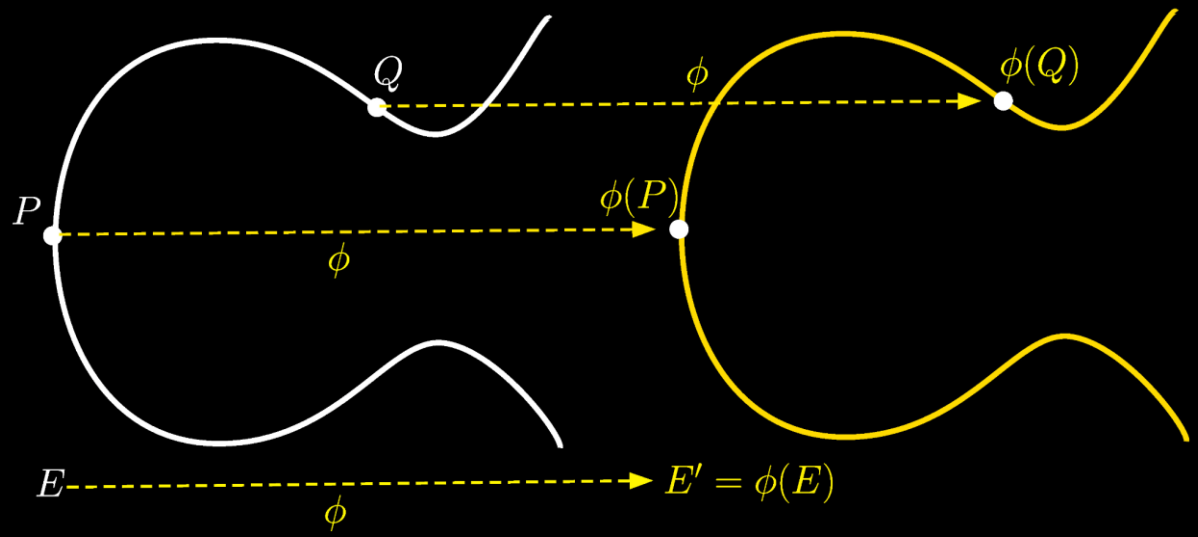

evolution

Isogeny-Based Cryptography

PQ Secure

- Isogeny-based cryptography is constructed on a set of curves.
- Given two curve E and $E' = \phi(E)$ find ϕ?





Supersingular Isomorphism Classes

PQ Secure

- We are interested in the set of supersingular curves (up to isomorphism) over a specific field
- Prime $p = 2^{e_A} \cdot 3^{e_B} \cdot f \pm 1$
- Elliptic curves over $\mathbb{F}_{p^2}$, $\#E = (p \mp 1)^2$
- Supersingular j-invariants: $\#S_{p^2} \approx \left\lfloor \frac{p}{12} \right\rfloor$ (isogenous elliptic curves)

Prime $p = 2^3 \cdot 3^2 - 1 = 71$, $\#E = 72^2$, $\#S_{p^2} = 7$

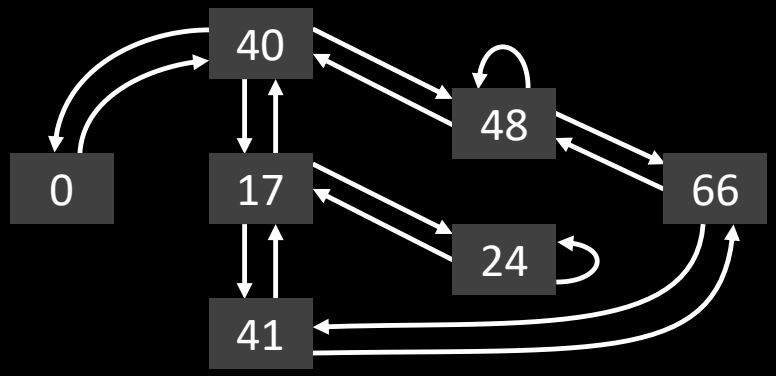
Isogeny Graphs

PQ Secure

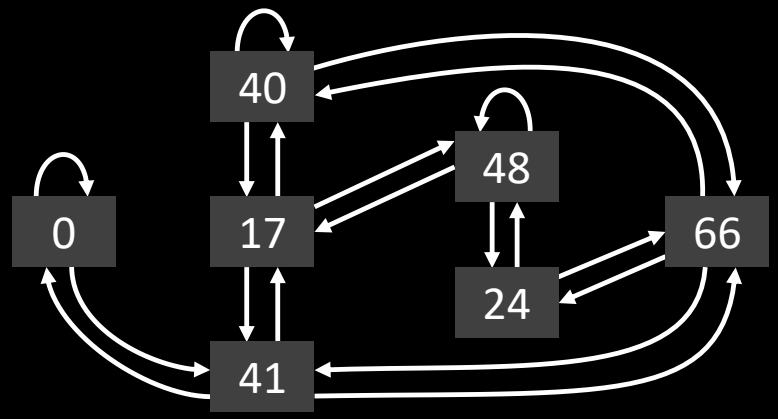
Vertices: All isogenous elliptic curves over $\mathbb{F}_{p^2}$.

Edges: Isogenies of degree ℓ

With isogeny of degree ℓ, we get a connected $(\ell + 1)$-regular graph.



2-isogeny graph

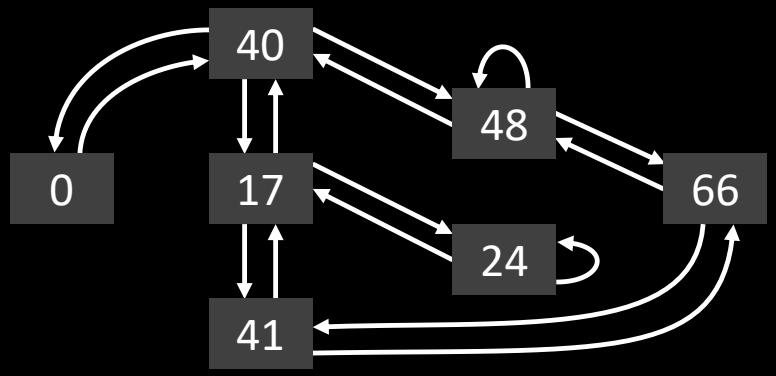


3-isogeny graph

Key Exchange based on Isogeny Graphs

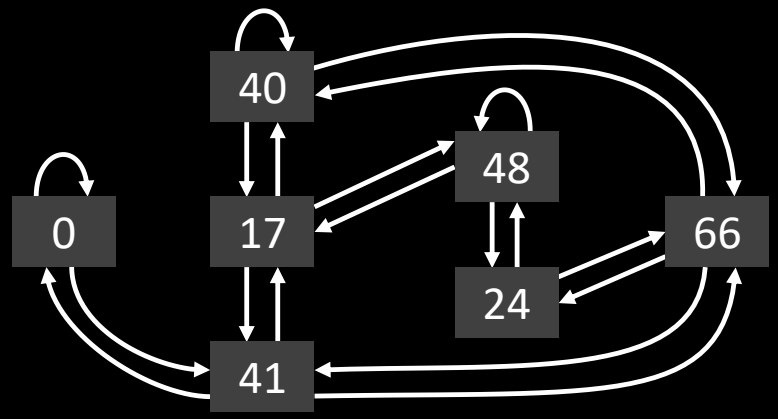
PQ Secure

Alice



2-isogeny graph

Bob



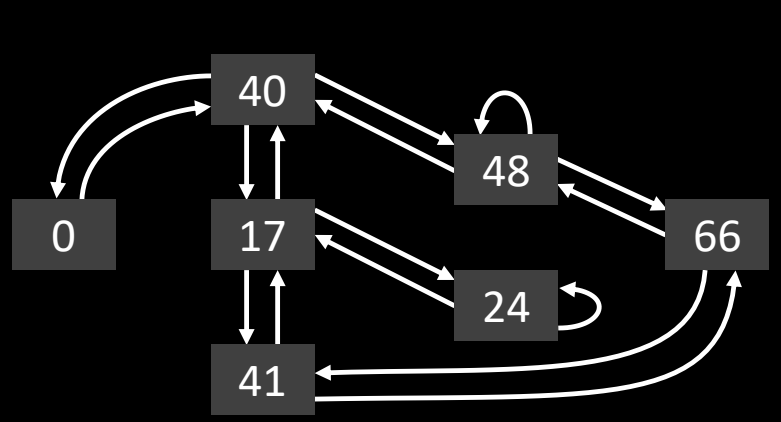
3-isogeny graph

Public Parameters

PQ Secure

$$E_0/\mathbb{F}_{p^2}$$
$$\{P_A, Q_A\} \in E_0[2^{e_A}]$$
$$\{P_B, Q_B\} \in E_0[3^{e_B}]$$

Alice

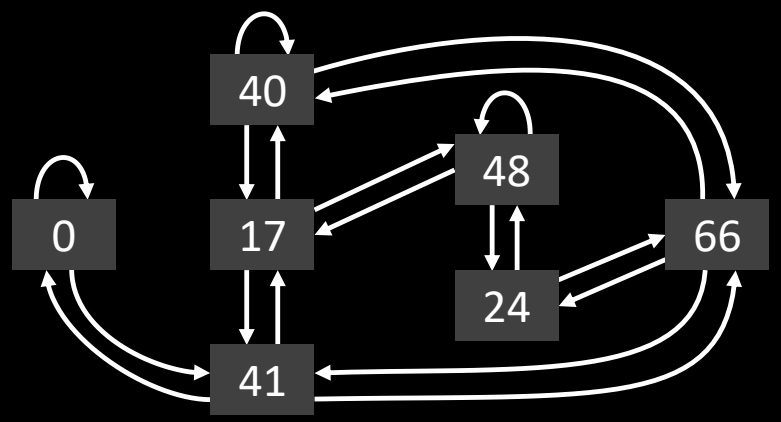


$P_A = (53, 55)$

$Q_A = (18, 27w + 44)$

$E_0: y^2 = x^3 + x$

Bob



$P_B = (7w + 20, 31w + 50)$

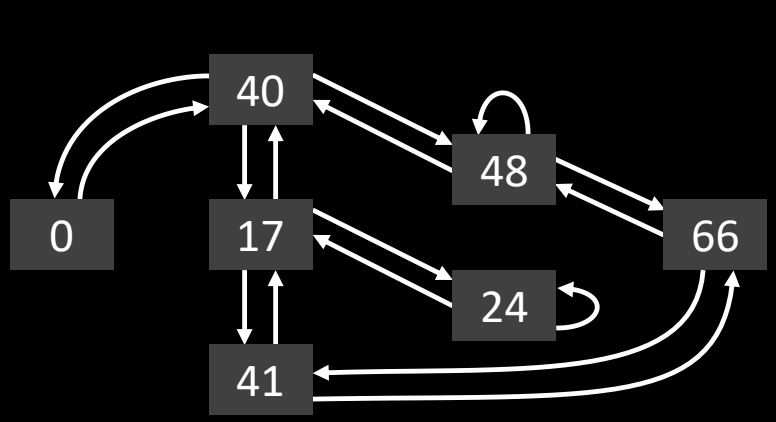
$Q_B = (21w + 64, 38w + 13)$

Secret Key

PQ Secure

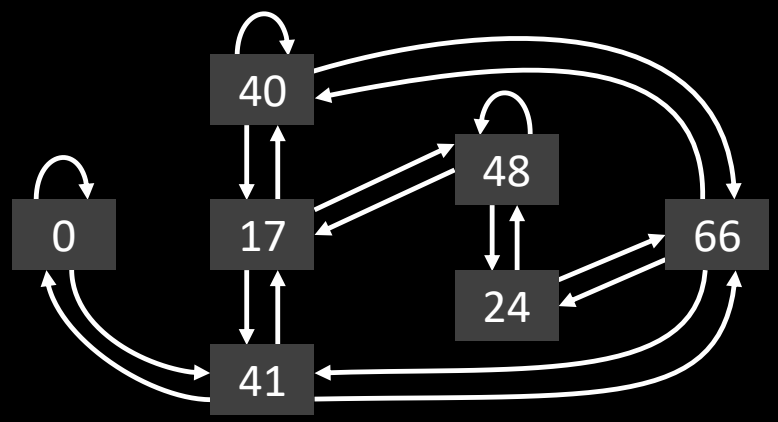
$s_A \in [0,2^{e_A})$
$s_B \in [0,3^{e_B})$

Alice



$s_A = 6$

Bob



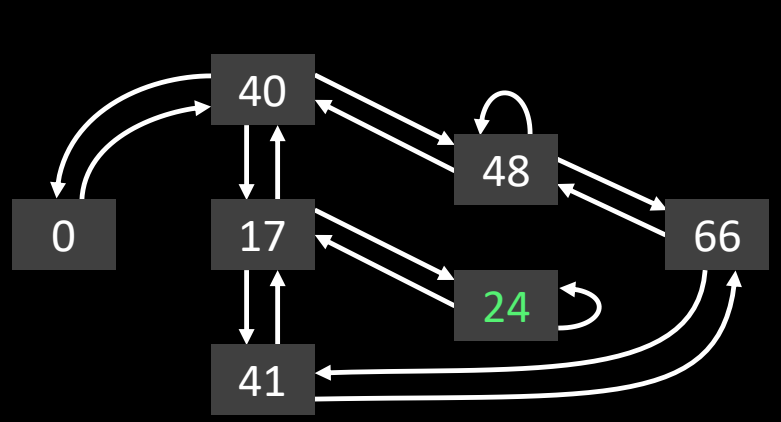
$s_B = 3$

Public Key Generation

E_0

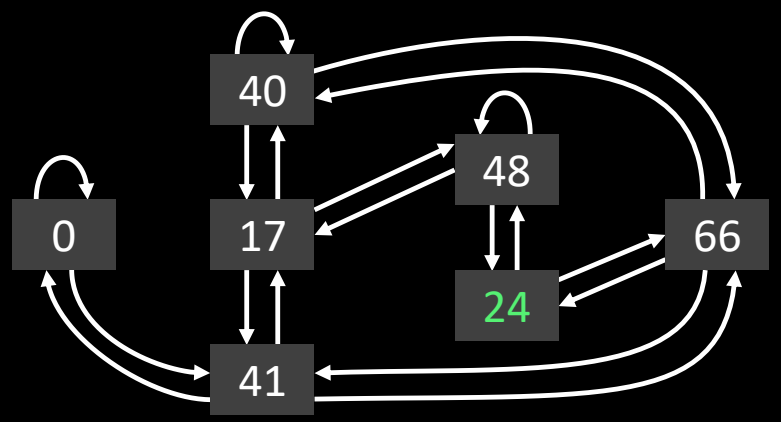
PQ Secure

Alice



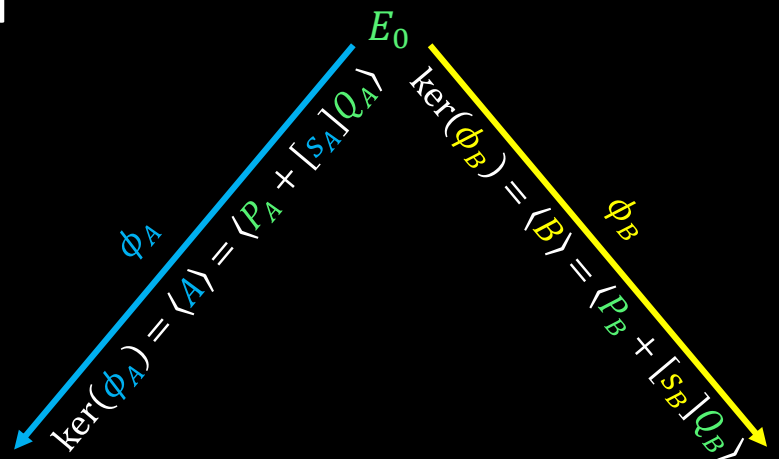
$E_0: y^2 = x^3 + x$

Bob

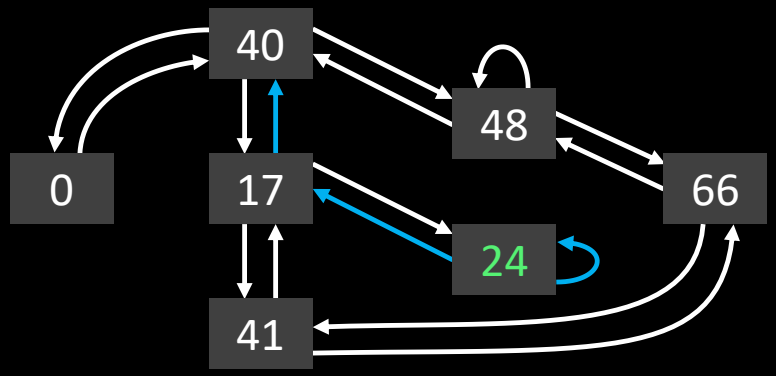


$E_0: y^2 = x^3 + x$

Public Key Generation



Alice

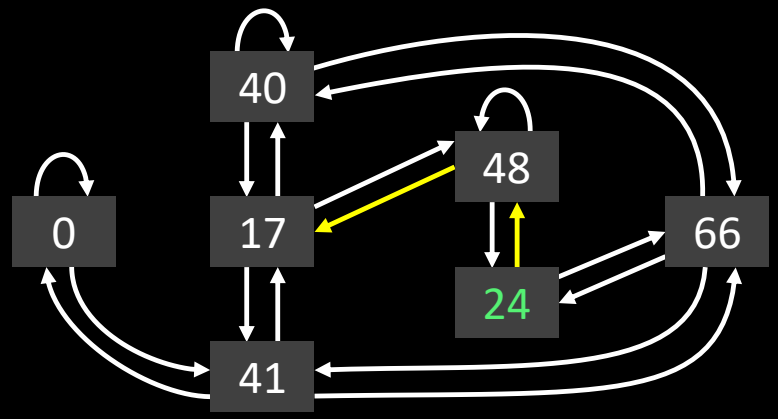


$E_0: y^2 = x^3 + x$

$\phi_A: E_0 \rightarrow E_A$

$p = 2^3 \cdot 3^2 - 1 = 71$

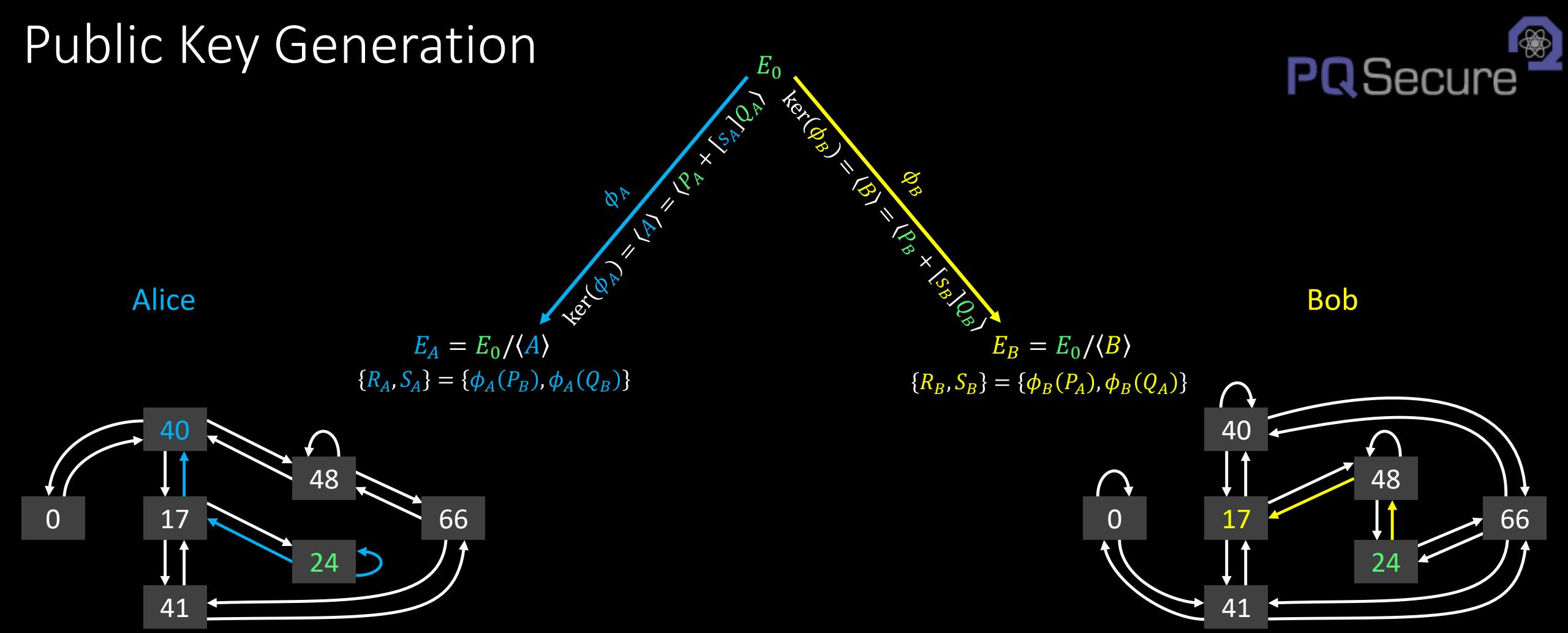
Bob



$E_0: y^2 = x^3 + x$

$\phi_B: E_0 \rightarrow E_B$

Public Key Generation



$E_0: y^2 = x^3 + x$

$\phi_A: E_0 \to E_A$

$E_A: y^2 = x^3 + 22x + 35$

$E_0: y^2 = x^3 + x$

$\phi_B: E_0 \to E_B$

$E_B: y^2 = x^3 + 63x + (55w + 16)$

Key Exchange

PQ Secure

E_0

ϕ_A

$\ker(\phi_A) = \langle A \rangle = \langle P_A + [s_A]Q_A \rangle$

ϕ_B

$\ker(\phi_B) = \langle B \rangle = \langle P_B + [s_B]Q_B \rangle$

Alice

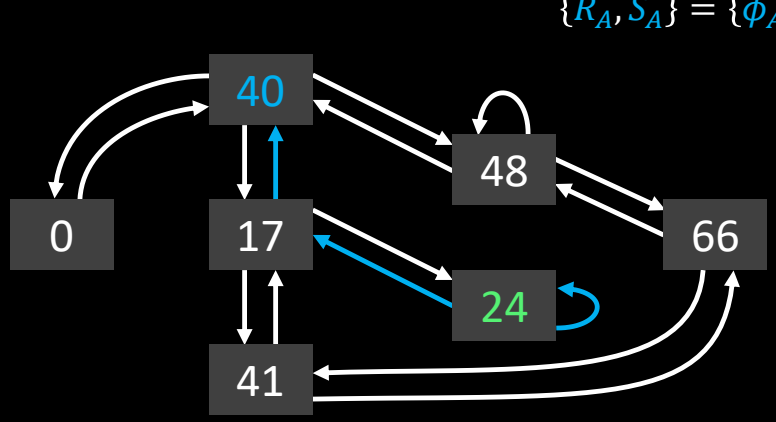
Bob

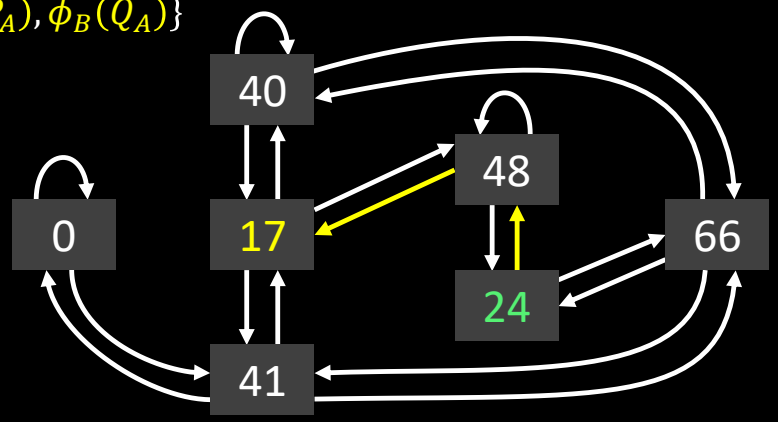
$E_A = E_0/\langle A \rangle$

$\{R_A, S_A\} = \{\phi_A(P_B), \phi_A(Q_B)\}$

$E_B = E_0/\langle B \rangle$

$\{R_B, S_B\} = \{\phi_B(P_A), \phi_B(Q_A)\}$





$E_0: y^2 = x^3 + x$

$\phi_A: E_0 \to E_A$

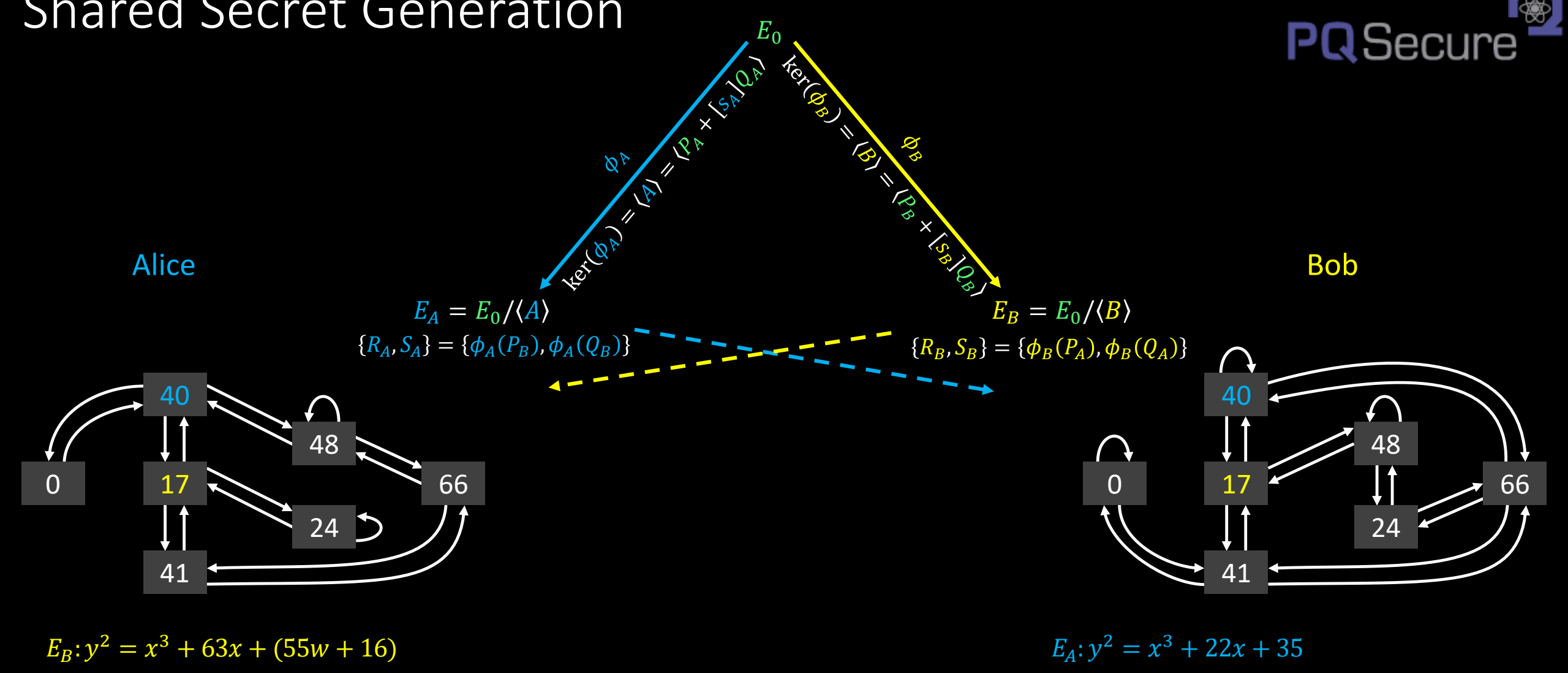
$E_A: y^2 = x^3 + 22x + 35$

$E_0: y^2 = x^3 + x$

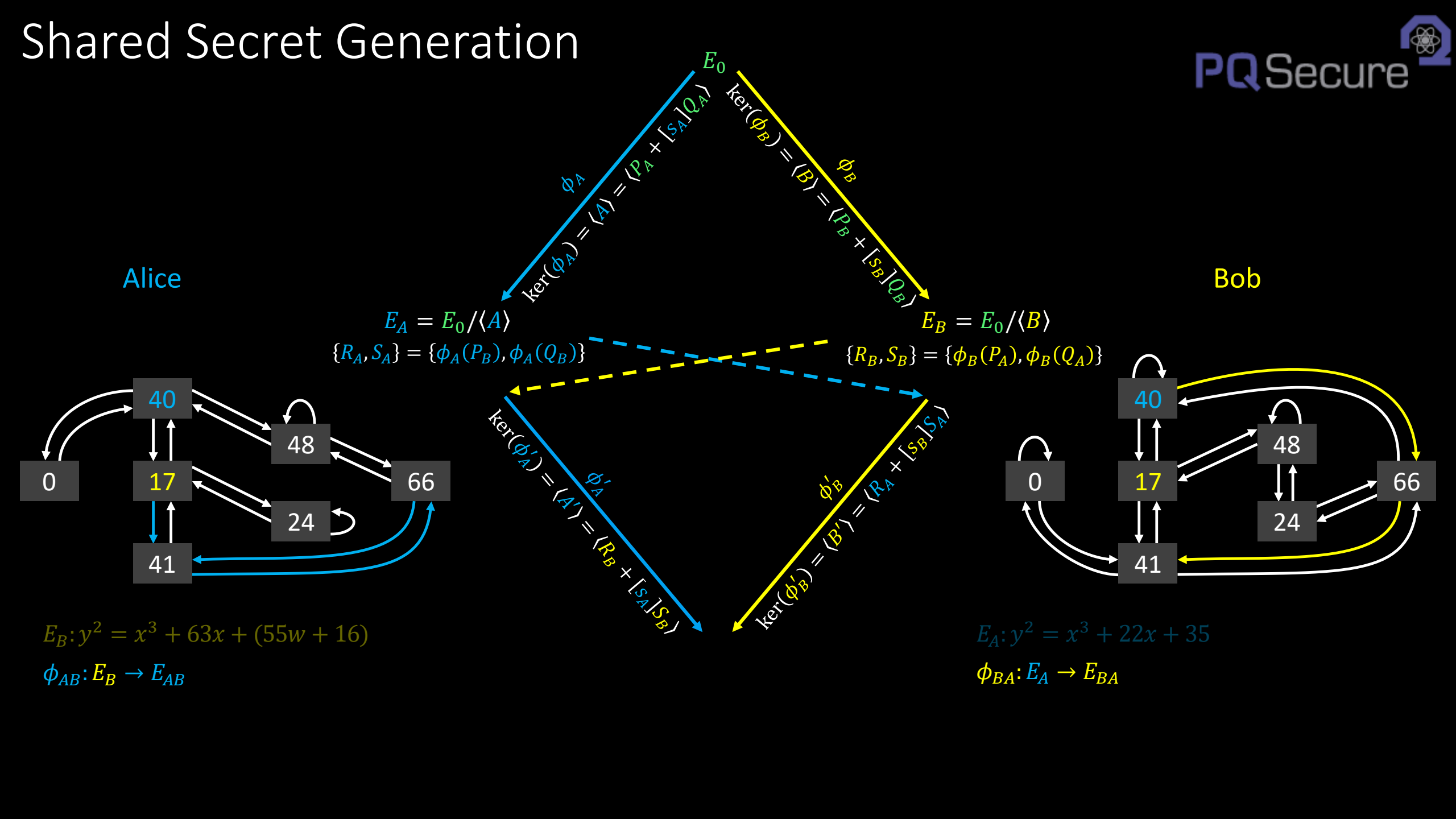
$\phi_B: E_0 \to E_B$

$E_B: y^2 = x^3 + 63x + (55w + 16)$

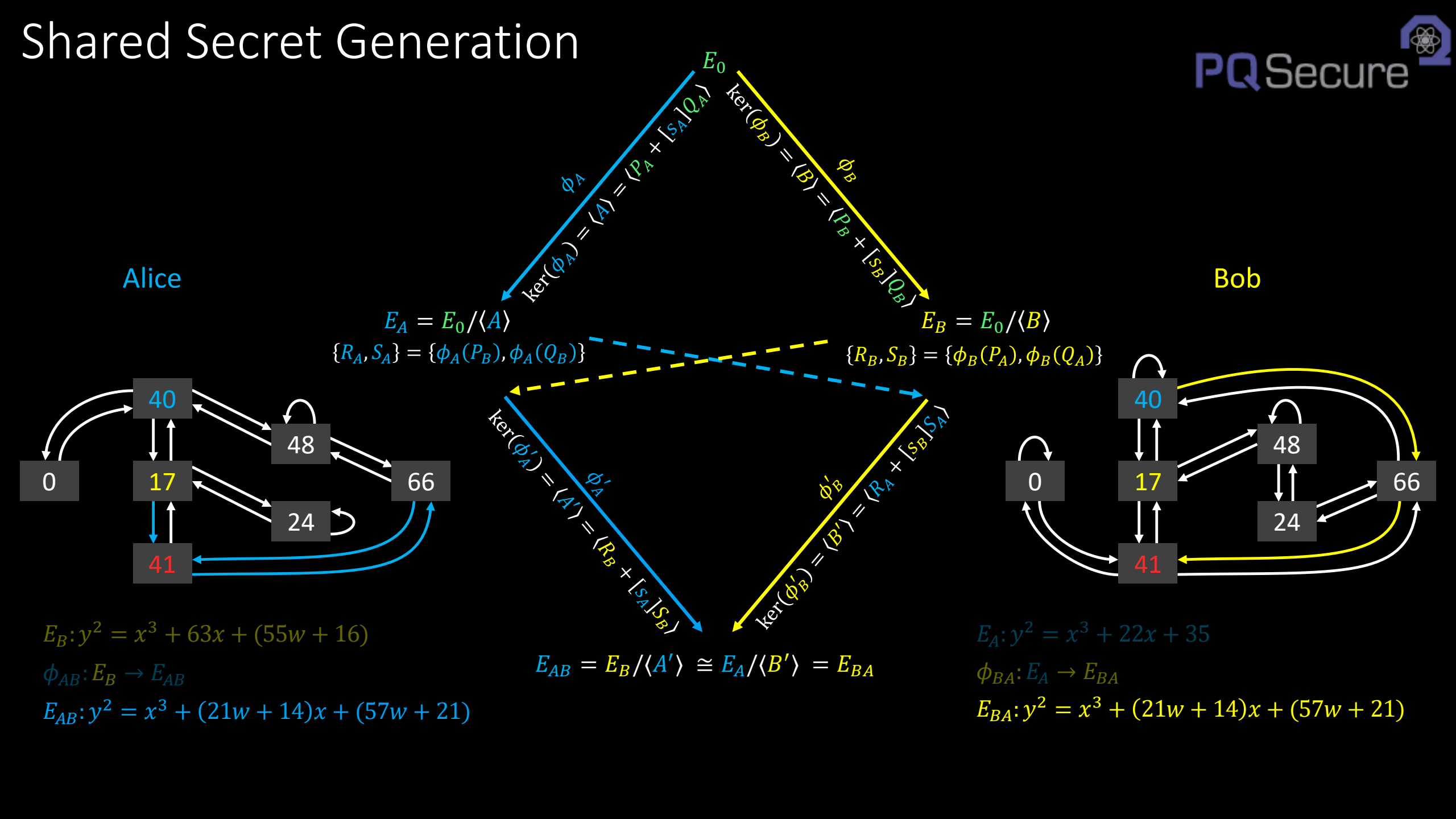
Shared Secret Generation



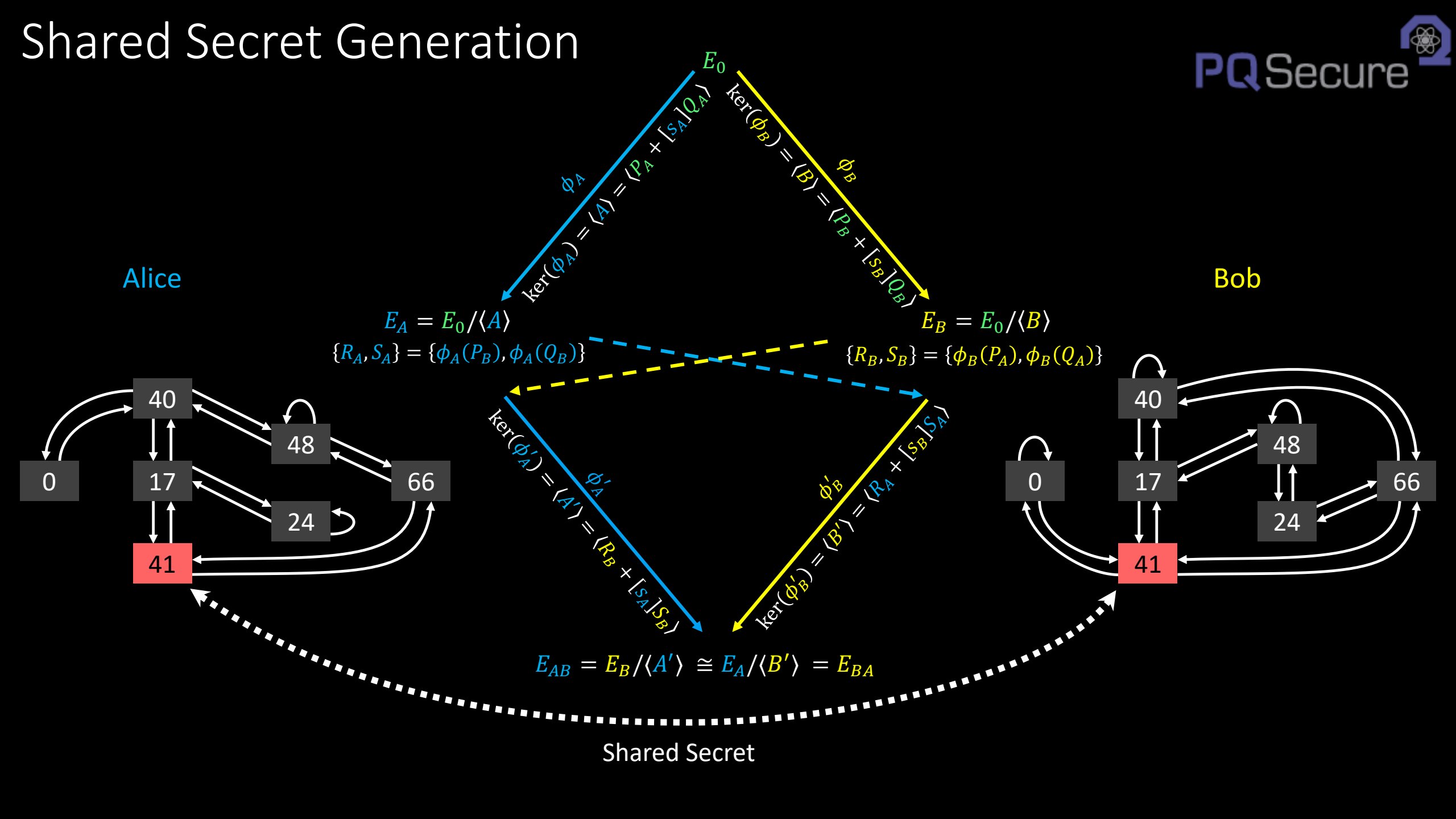
Shared Secret Generation



Shared Secret Generation



Shared Secret Generation



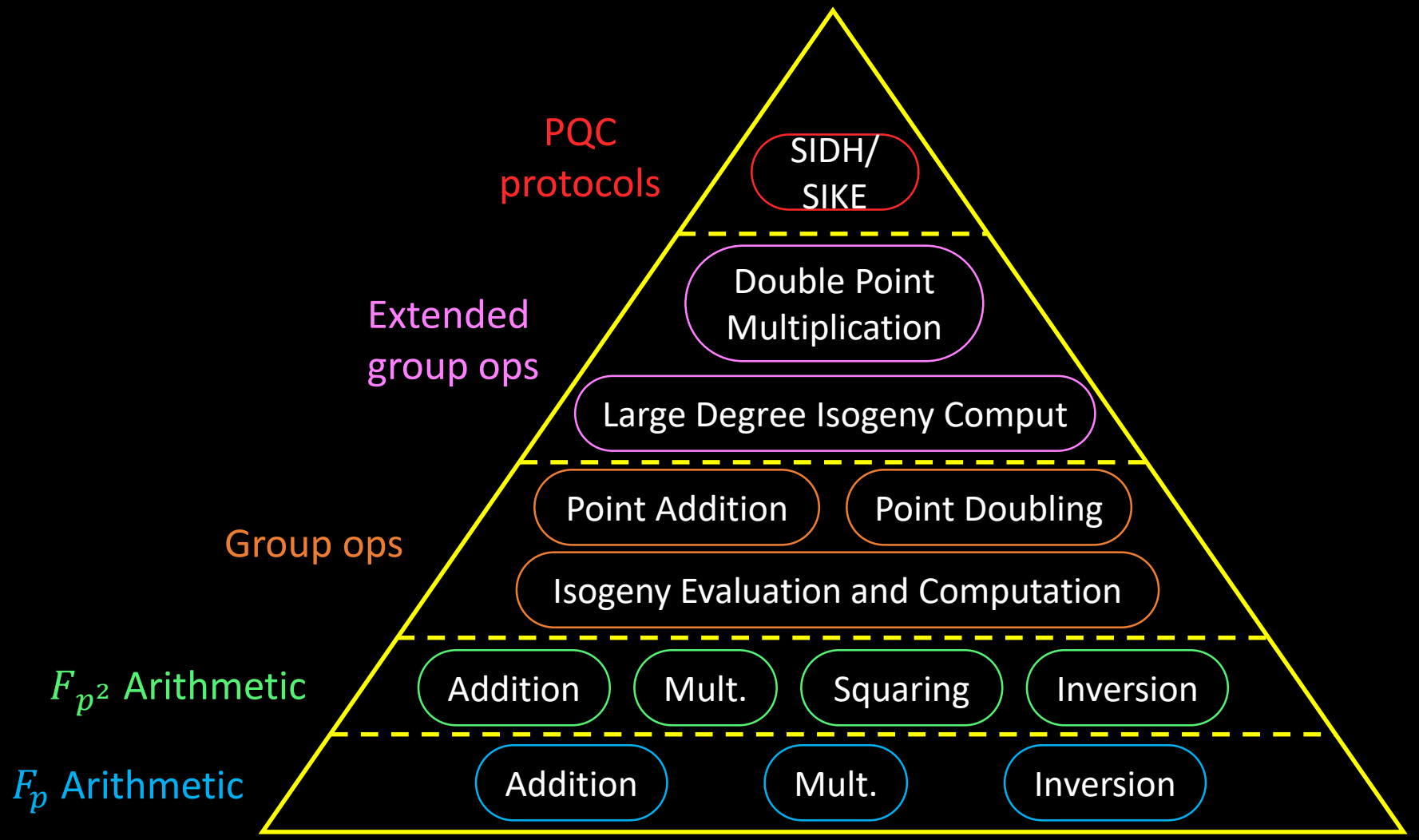
SIKE Key sizes

PQ Secure

NIST Level	Prime size (bits)	Prime	Public key size (bytes)	Compressed PK size (bytes)
1	434	$2^{216}3^{137}-1$	330	196
2	503	$2^{250}3^{159}-1$	378	224
3	610	$2^{305}3^{192}-1$	462	273
5	751	$2^{372}3^{239}-1$	564	331

SIDH Computations

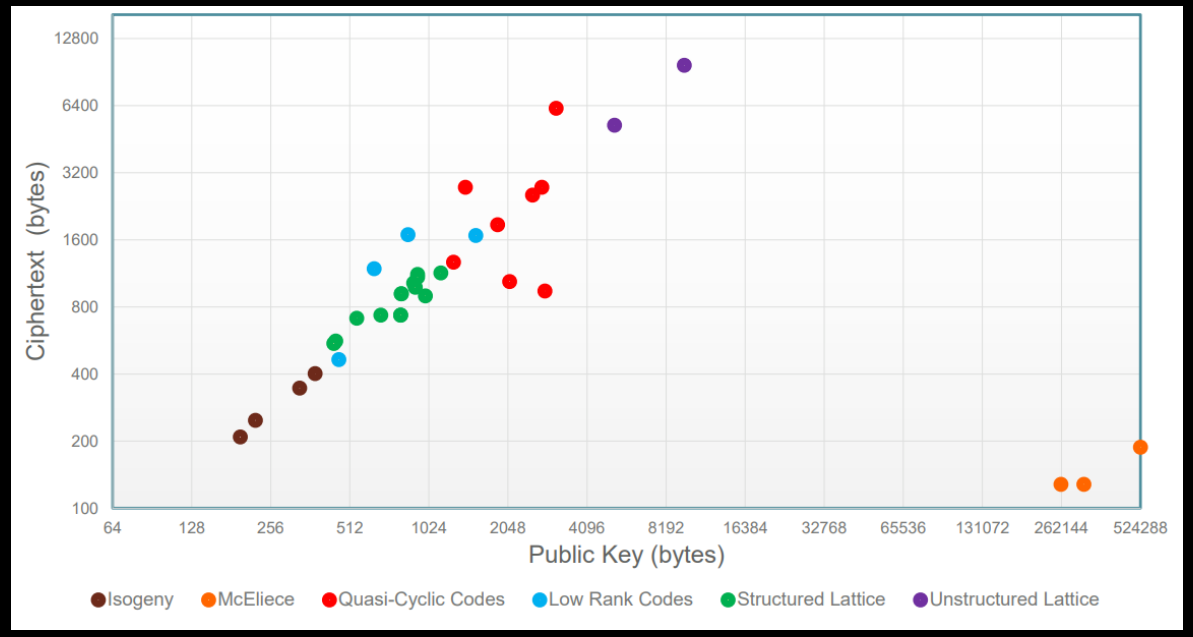
PQ Secure



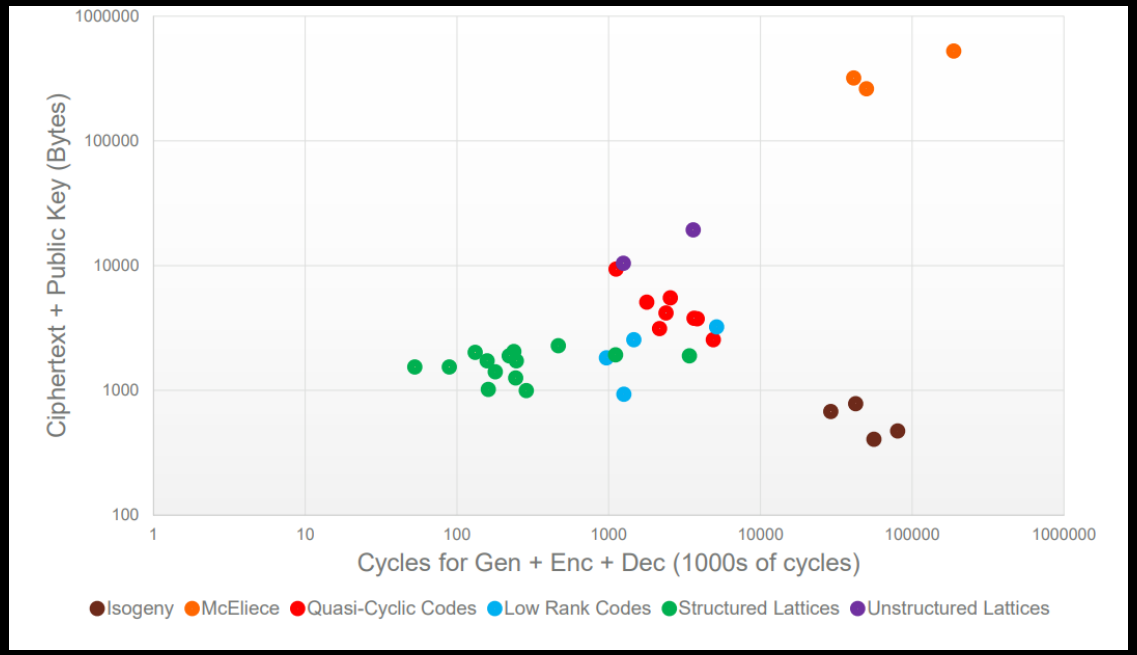
Comparisons

PQ Secure

Public Key vs Ciphertext Size



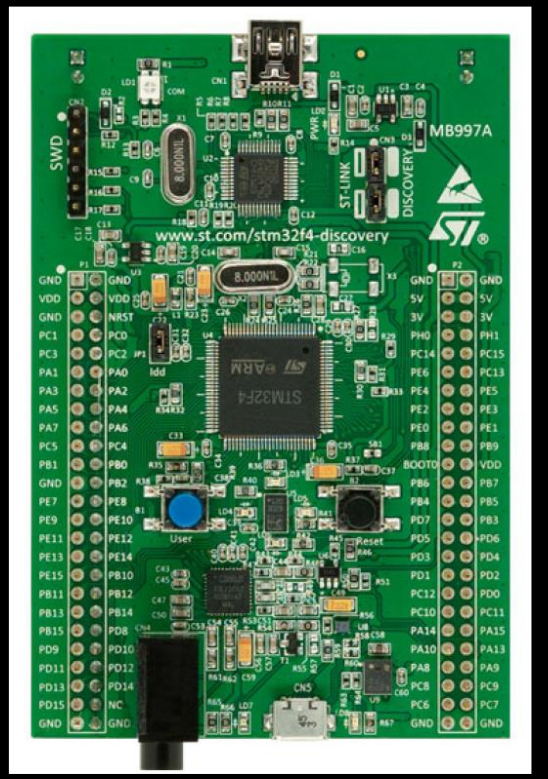
Speed vs Size



Post-quantum on Small Devices

PQ Secure

- STM32F4DISCOVERY
  - Recommended by NIST for PQC evaluation
  - ARM Cortex-M4
  - 32-bit, ARMv7E-M
  - 192 KiB RAM, 168 MHZ

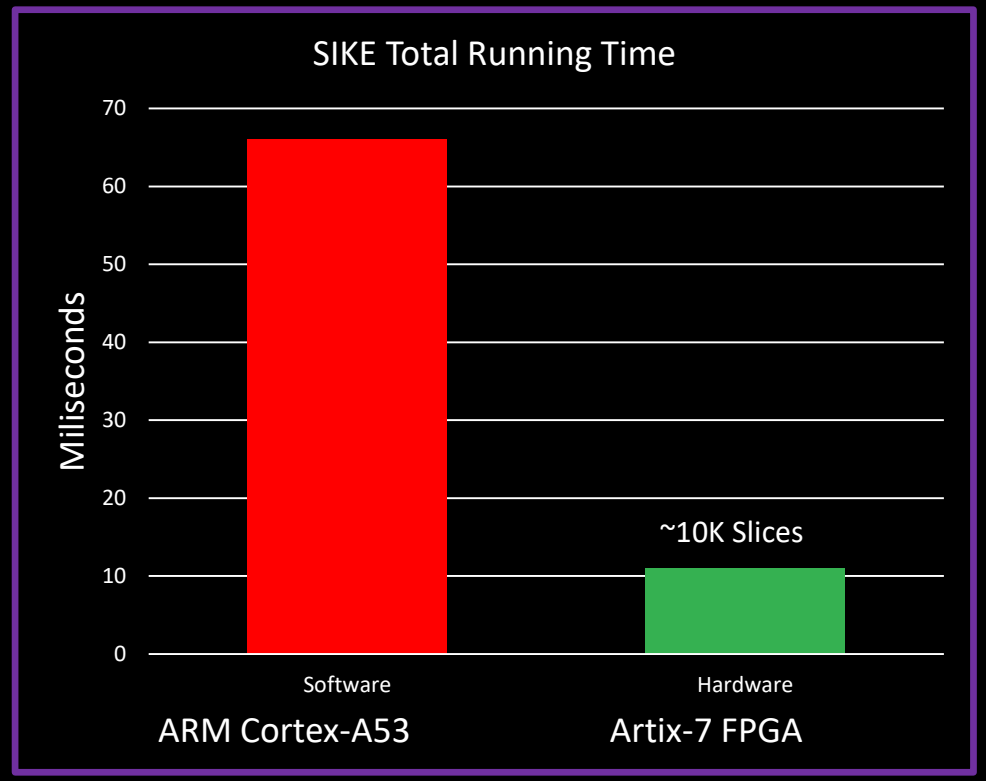


STM32 F4

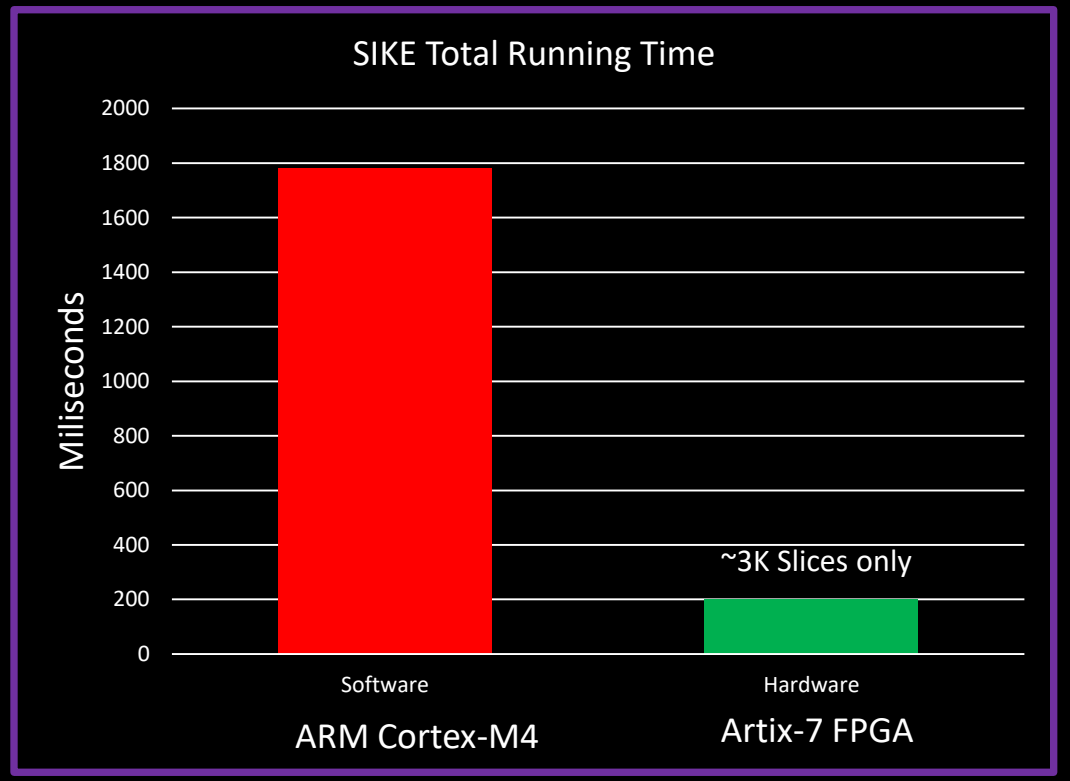
SIKE: Results for NIST level 1

PQ Secure

Target: High Performance Edge



Target: Resource-constrained IoT



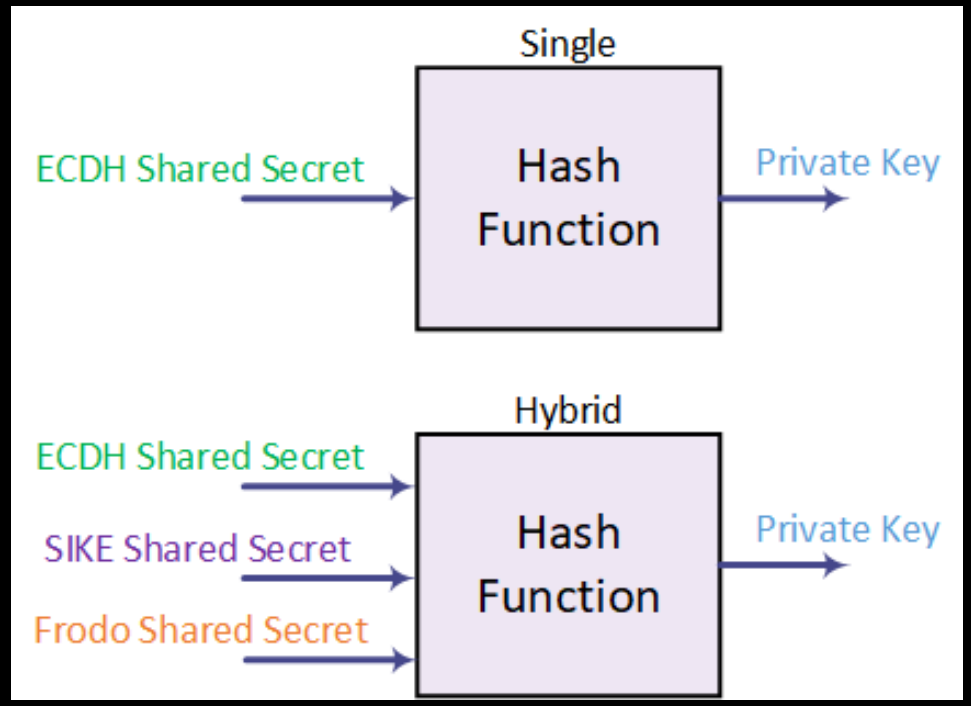
Hybrid Key Exchange and Signatures

PQ Secure

- Hybrid → combine multiple cryptosystems at once
  - Classical PKC still in use
  - Classical is Smaller/Faster/Efficient
  - PQC still being studied: i.e. may be broken
  - Multiple choices for different environments
- Goal: Leverage risk among multiple schemes

- Example:
  - Hash shared secret from multiple cryptosystems to get private key for AES

BROKEN if ECDH is broken



BROKEN if ECDH, SIKE, AND Frodo are broken

The case for SIKE

PQ Secure

- The post-quantum landscape is uncharted territory:
  - The smallest scheme is the slowest, and the fastest scheme is the largest.
  - Compare with traditional cryptography, where the fastest scheme (ECC) is also the smallest.

- This situation introduces a new set of tradeoffs.
  - SIKE's advantages will become more pronounced over time.
  - SIKE's disadvantages will become less pronounced over time.

The future of SIKE: Computational Costs

- Hardware gets faster over time.
- Software also gets faster over time.
- The above happens naturally, without effort or expenditure.
- An across-the-board performance increase reduces the performance penalty of SIKE (in absolute terms).
- We can also spend more money for faster hardware.
- Certain expenditures (e.g. hardware acceleration) provide good value per unit cost.

The future of SIKE: Communication Costs

PQ Secure

- As hardware and software gets faster, attacks get faster.
- Faster attacks require larger keys to counteract.
- An across-the-board key size increase enlarges the communication cost benefits of SIKE (in absolute terms).
- Variance in communication channels is much higher than variance in cycle counts. SIKE already wins today on desktop browsers when including variance.

PQ Secure

Thank you!

Questions?