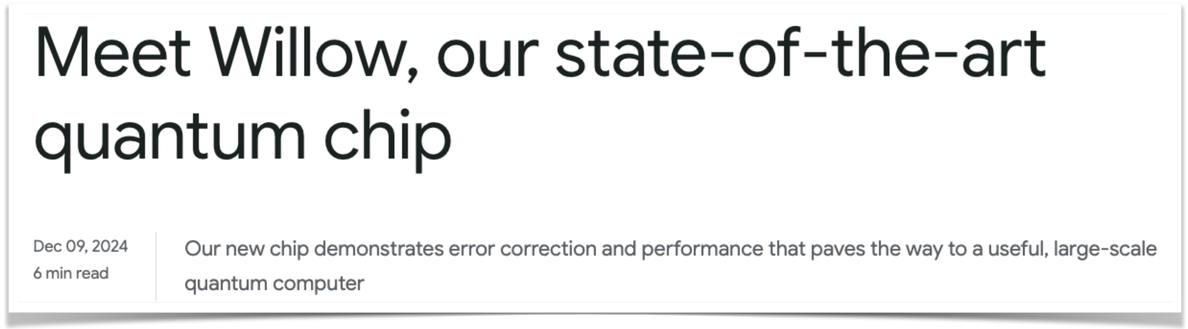
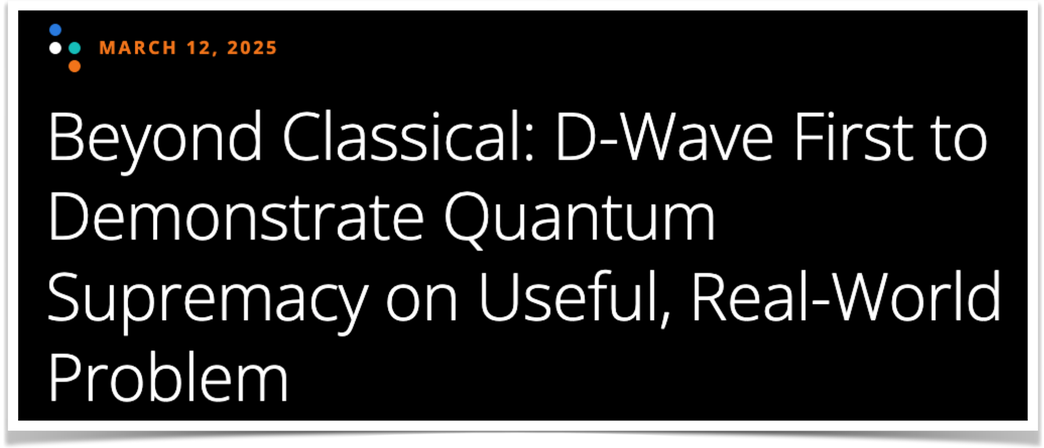
Practical Cryptographic Systems

Post-Quantum Cryptography

Instructor: Matthew Green

Quantum Computers





Quantum Computers

Quantum Computers

Quantum Computers

- What is it?
  - A different model of computation: exploits wave superposition and interference
  - Different physical model: qubits, unitary gates, measurements
- Is it more powerful than classical computation?
  - Every program for a classical computer can be run on a quantum computer with linear overhead
  - We don't know if quantum programs can be efficiently run on classical computers
  - Converse is widely believed to be true: We currently have exponentially faster quantum algorithms for some problems

Source: Google Quantum AI

Relevant Quantum Algorithms

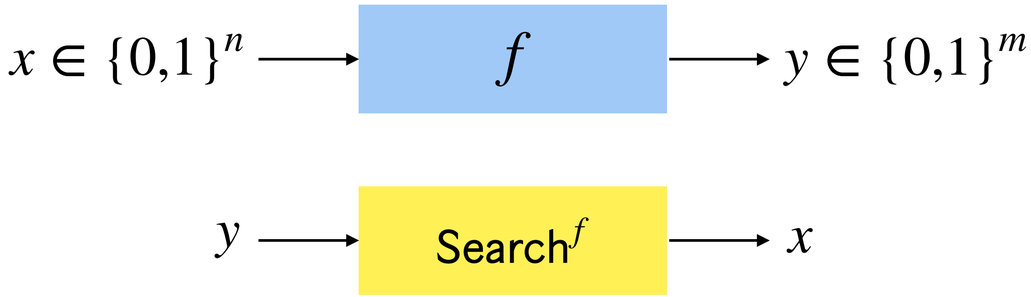
Grover’s Algorithm

Relevant Quantum Algorithms

Grover's Algorithm

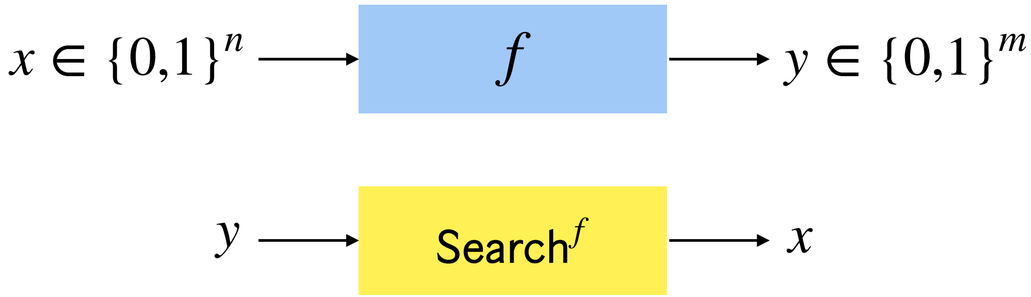
Relevant Quantum Algorithms

Grover's Algorithm



Relevant Quantum Algorithms

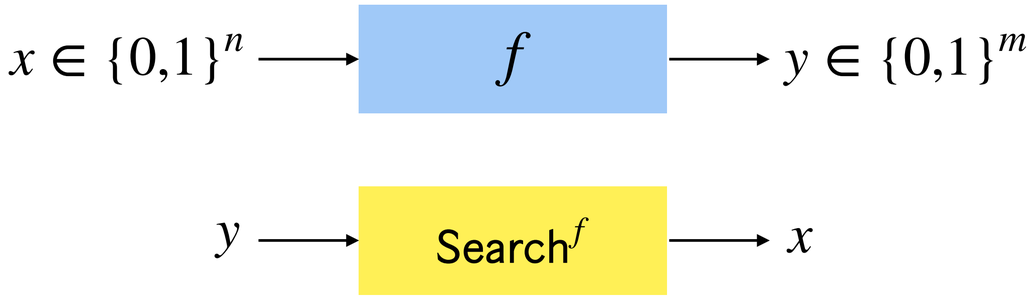
Grover's Algorithm



Classical:

Relevant Quantum Algorithms

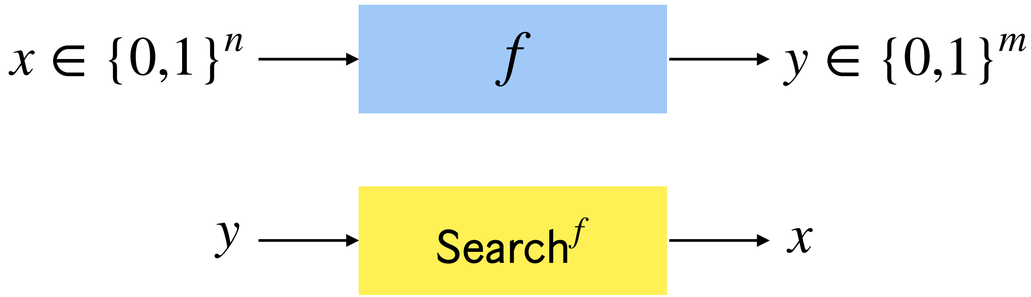
Grover's Algorithm



Classical: 2^n evaluations of f

Relevant Quantum Algorithms

Grover's Algorithm

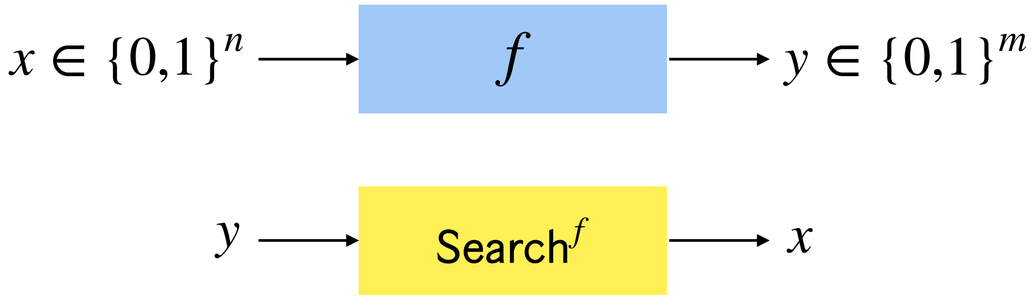


Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

Relevant Quantum Algorithms

Grover's Algorithm



Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

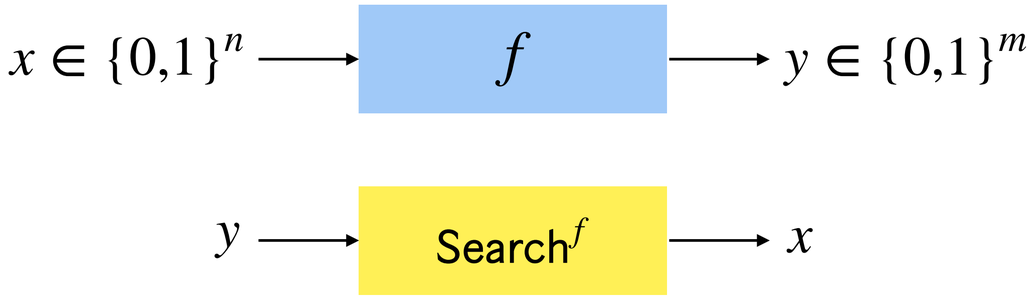
Unstructured search

Works for any function f

Quadratic Speedup

Relevant Quantum Algorithms

Grover's Algorithm



Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

Unstructured search

Works for any function f

Quadratic Speedup

Shor's Algorithm

$\mathbb{Z}_p$

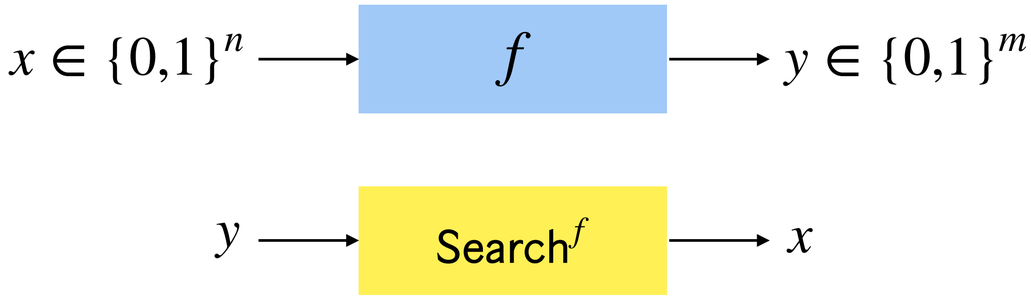
Elliptic Curves

Factoring N

Discrete log over any finite cyclic group

Relevant Quantum Algorithms

Grover's Algorithm



Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

Unstructured search

Quadratic Speedup

Works for any function f

Shor's Algorithm

$\mathbb{Z}_p$

Elliptic Curves

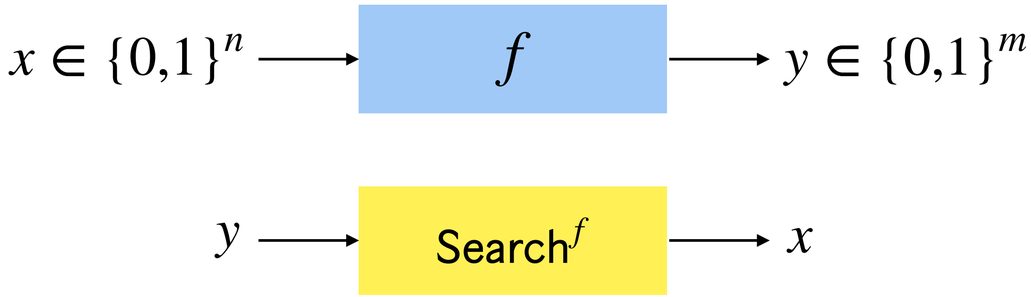
Discrete log over any finite cyclic group

Factoring N

Classical: $\tilde{O}\left(e^{(\log N)^{1/3}}\right)$

Relevant Quantum Algorithms

Grover's Algorithm



Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

Unstructured search

Quadratic Speedup

Works for any function f

Shor's Algorithm

$\mathbb{Z}_p$

Elliptic Curves

Discrete log over any finite cyclic group

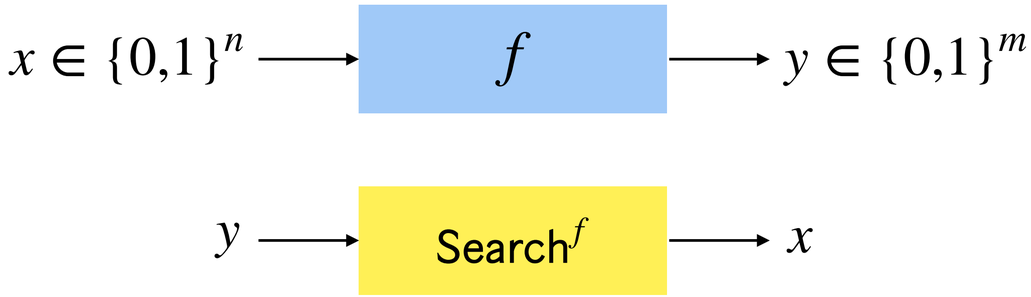
Factoring N

Classical: $\tilde{O}\left(e^{(\log N)^{1/3}}\right)$

Quantum: $O\left((\log N)^3\right)$

Relevant Quantum Algorithms

Grover's Algorithm



Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

Unstructured search

Quadratic Speedup

Works for any function f

Shor's Algorithm

$\mathbb{Z}_p$

Elliptic Curves

Factoring N

Classical: $\tilde{O}\left(e^{(\log N)^{1/3}}\right)$

Quantum: $O\left((\log N)^3\right)$

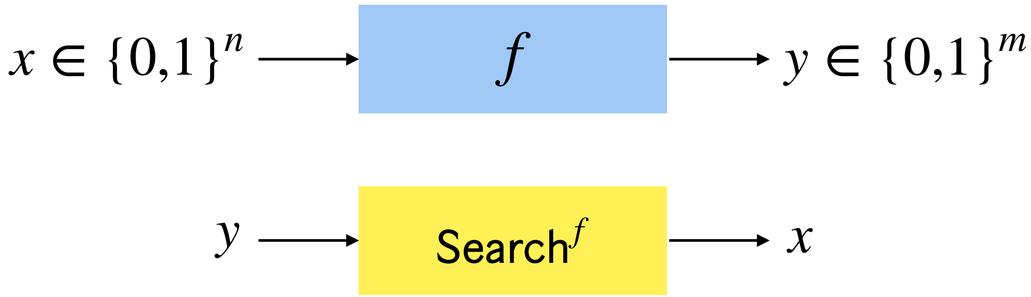
Discrete log over any finite cyclic group

Classical: $O\left(\sqrt{|G|}\right)$

Quantum: $O\left((\log |G|)^3\right)$

Relevant Quantum Algorithms

Grover's Algorithm



Classical: 2^n evaluations of f

Grover's Algorithm: (Quantum) $2^{n/2}$ evaluations of f

Unstructured search

Works for any function f

Quadratic Speedup

Shor's Algorithm

$\mathbb{Z}_p$

Elliptic Curves

Factoring N

Classical: $\tilde{O}\left(e^{(\log N)^{1/3}}\right)$

Quantum: $O\left((\log N)^3\right)$

Discrete log over any finite cyclic group

Classical: $O\left(\sqrt{|G|}\right)$

Quantum: $O\left((\log |G|)^3\right)$

Structured search

Works only for specific problems with compatible algebraic structure

Exponential Speedup

Relevant Quantum Algorithms

Grover's Algorithm

Unstructured search

Quadratic Speedup

Works for any function f

Shor's Algorithm

Structured search

Exponential Speedup

Works only for specific problems with compatible algebraic structure

Relevant Quantum Algorithms

Grover's Algorithm

Unstructured search

Quadratic Speedup

Works for any function f

Baby-step Giant-step

Any cyclic group

Quadratic Speedup

Runtime:

$O(\sqrt{|G|})$ integer ops

Shor's Algorithm

Structured search

Exponential Speedup

Works only for specific problems with compatible algebraic structure

Relevant Quantum Algorithms

Grover's Algorithm

Unstructured search

Quadratic Speedup

Works for any function f

Baby-step Giant-step

Any cyclic group

Quadratic Speedup

Runtime:

$O(\sqrt{|G|})$ integer ops

Shor's Algorithm

Structured search

Exponential Speedup

Works only for specific problems with compatible algebraic structure

Pohlig-Hellman

Smooth order groups

Small prime factors

Exponential Speedup

$|G| = p_1^{e_1} \cdot \ldots \cdot p_n^{e_n}$

Runtime: $\sum_i e_i \cdot \sqrt{p_i}$ integer ops

Status of Primitives We've Discussed

- Symmetric-key Cryptography

Status of Primitives We've Discussed

- Symmetric-key Cryptography
  - Hash functions, block ciphers
  - Only affected by Grover's algorithm
  - Double the key-length to retain same security level e.g., use AES-256 and SHA-512.
  - In practice, safe to use current primitives (AES-128, SHA-256)
    - Unlike classical algorithms, running Grover search in parallel does not give significant speedup

Status of Primitives We've Discussed

- Symmetric-key Cryptography
  - Hash functions, block ciphers
  - Only affected by Grover's algorithm
  - Double the key-length to retain same security level e.g., use AES-256 and SHA-512.
  - In practice, safe to use current primitives (AES-128, SHA-256)
    - Unlike classical algorithms, running Grover search in parallel does not give significant speedup

- Public-key Cryptography

Status of Primitives We've Discussed

- Symmetric-key Cryptography
  - Hash functions, block ciphers
  - Only affected by Grover's algorithm
  - Double the key-length to retain same security level e.g., use AES-256 and SHA-512.
  - In practice, safe to use current primitives (AES-128, SHA-256)
    - Unlike classical algorithms, running Grover search in parallel does not give significant speedup

- Public-key Cryptography
  - Public-key encryption, key-exchange, efficient signature schemes (Schnorr, ECDSA)
  - Depend on hardness of discrete log and factoring $\Longrightarrow$ broken due to Shor's algorithm
  - We need post-quantum secure constructions

Why Worry About Quantum Computers Today?

Researchers say it might take one million or more qubits to crack RSA. The largest quantum machine available today ... has 433 qubits.

Scientific American
Davide Castelvecchi

January 10, 2023

A 2017 study by the Quantum Computing Report estimated that a quantum computer with 4 million error-corrected Qubits could potentially crack a 256-bit elliptic curve private key in about 8 hours. Currently, the largest quantum computers have just a few hundred Qubits with high error rates.

The Hacker Wire
Mohamed Nabil Ali

March 14, 2024

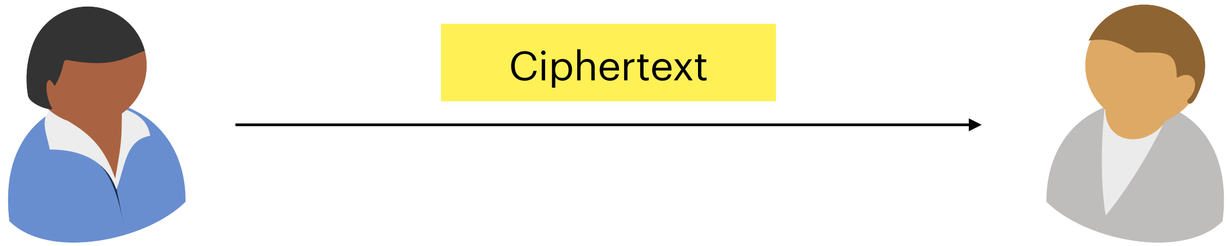
The Willow chip is not capable of breaking modern cryptography.

Charina Chou
Google Quantum AI Director

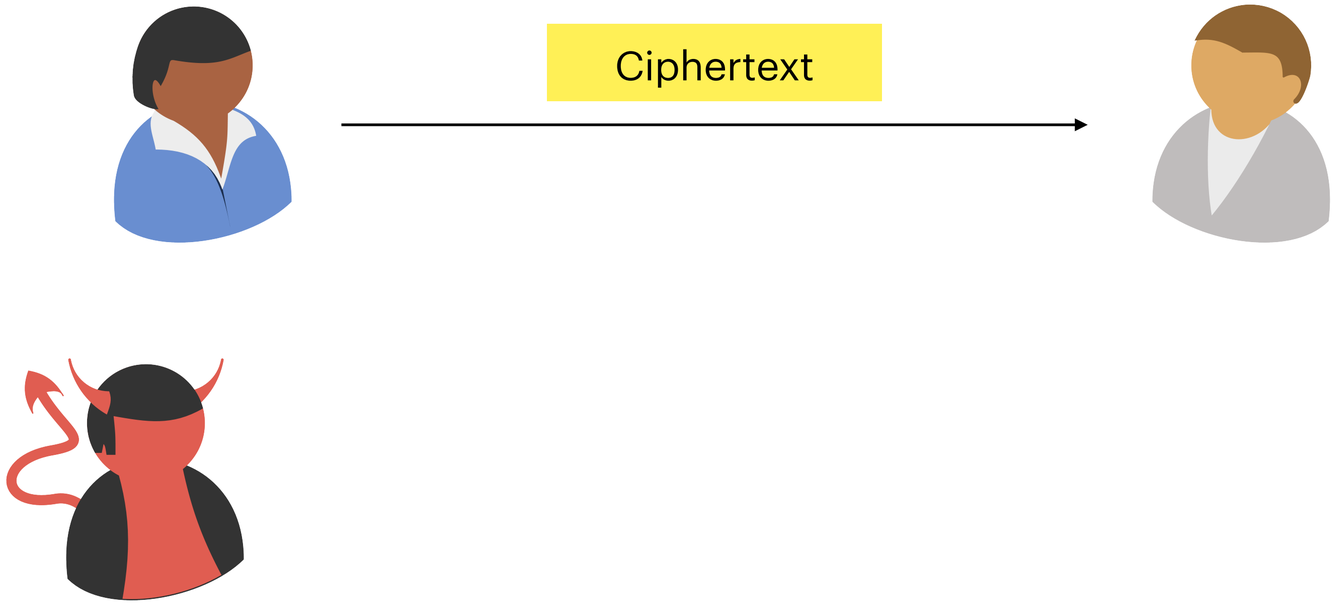
December 12, 2024

Why Worry About Quantum Computers Today?

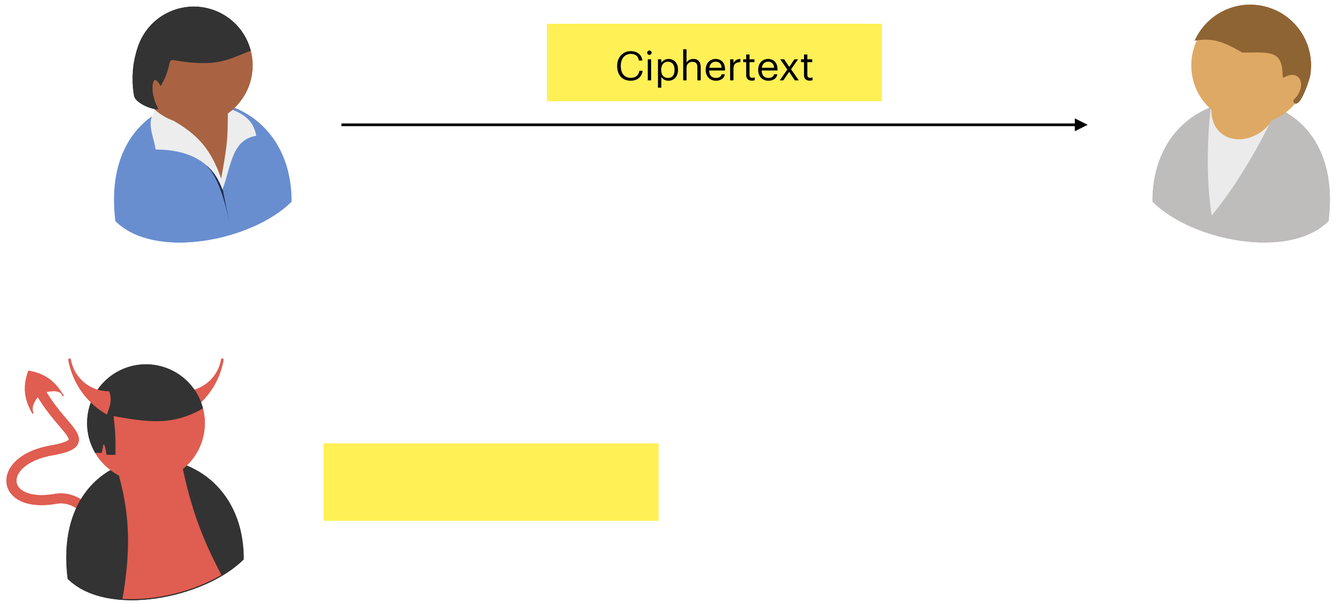
Why Worry About Quantum Computers Today?



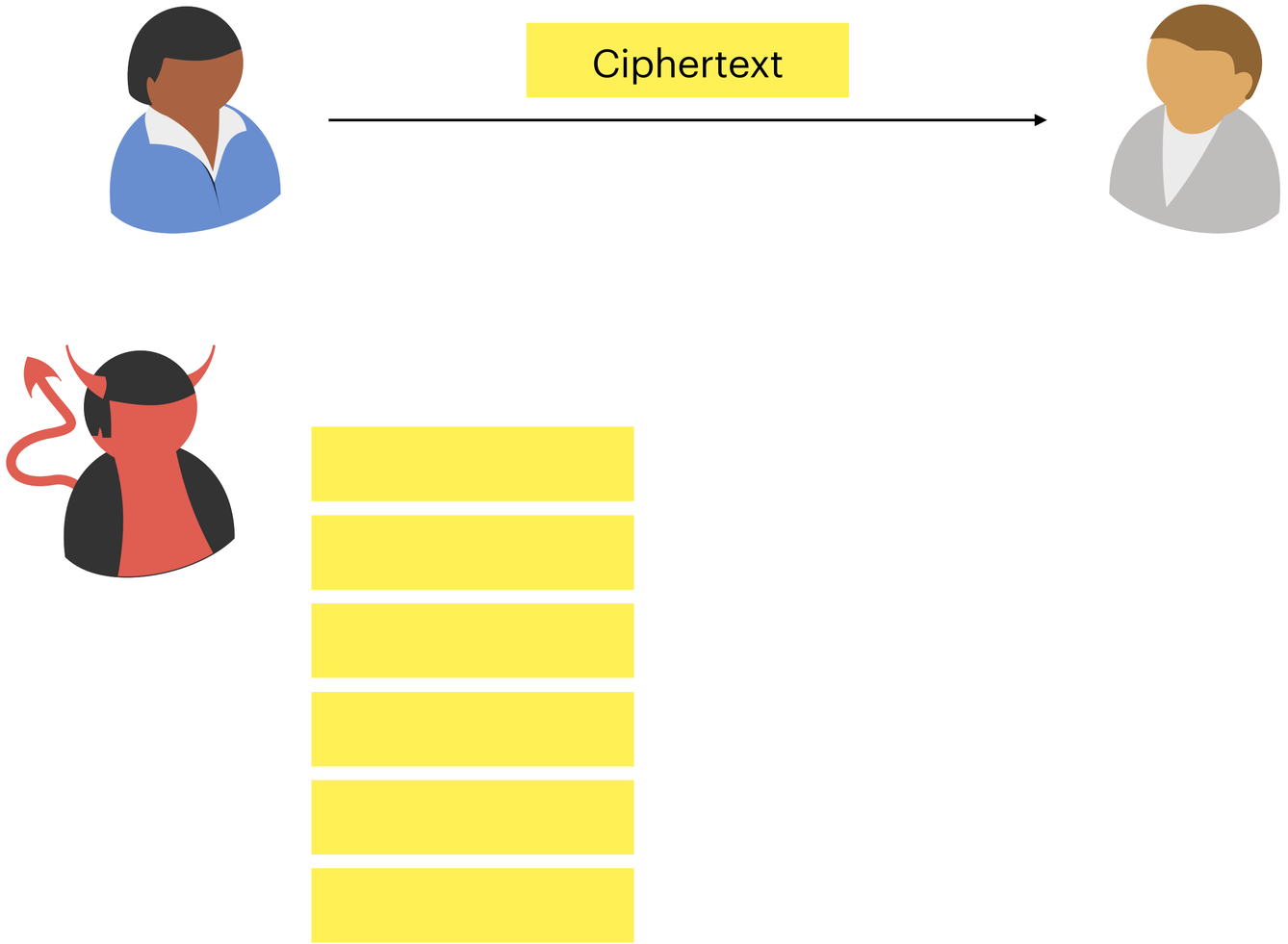
Why Worry About Quantum Computers Today?



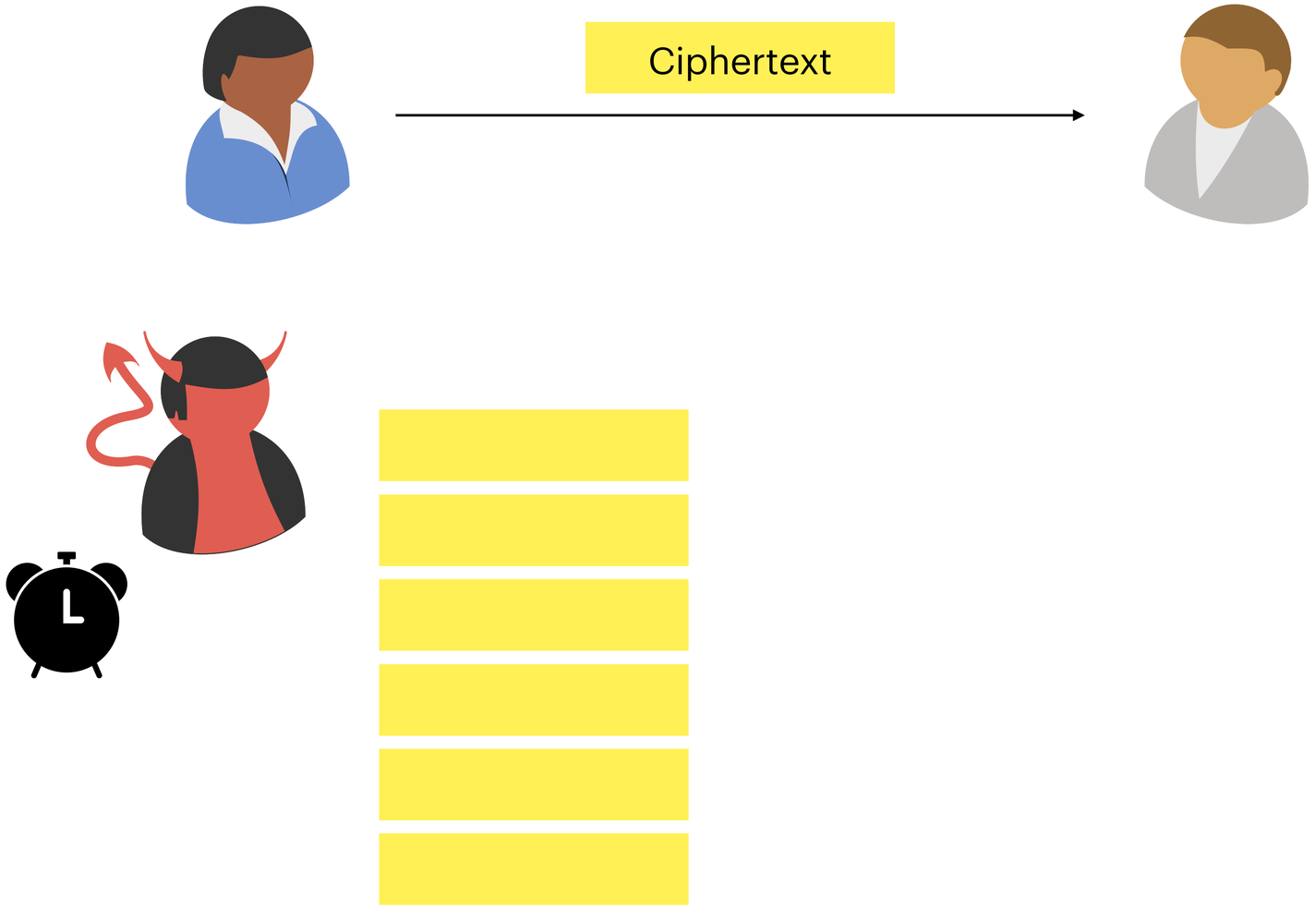
Why Worry About Quantum Computers Today?



Why Worry About Quantum Computers Today?



Why Worry About Quantum Computers Today?

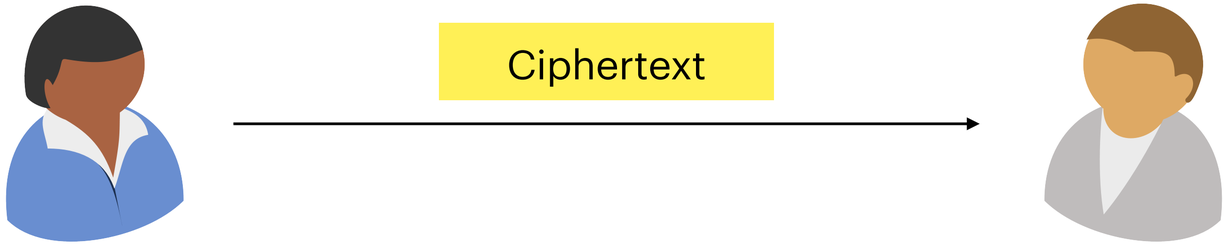


Why Worry About Quantum Computers Today?

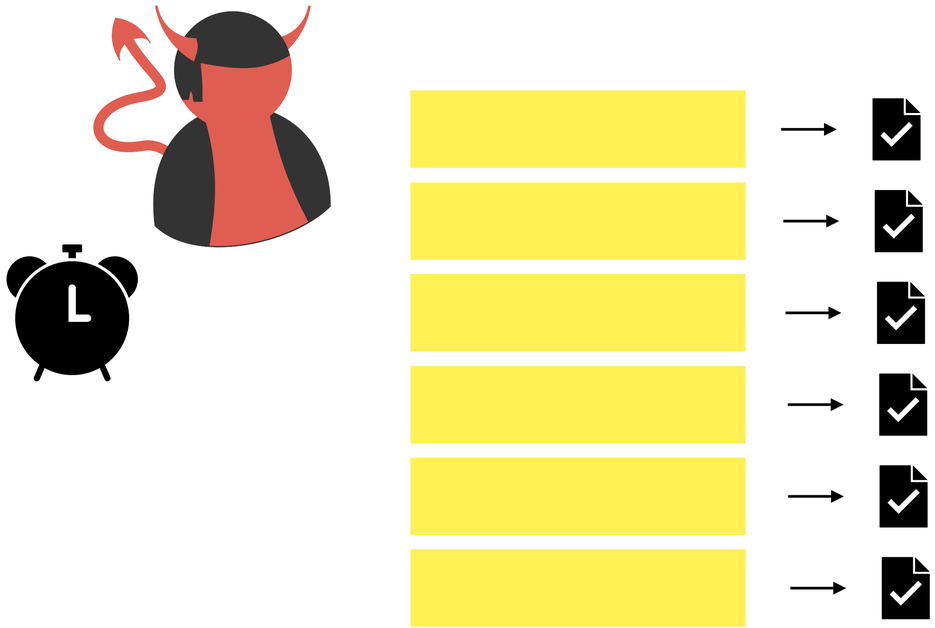
Ciphertext

Source: Google Quantum AI

Why Worry About Quantum Computers Today?

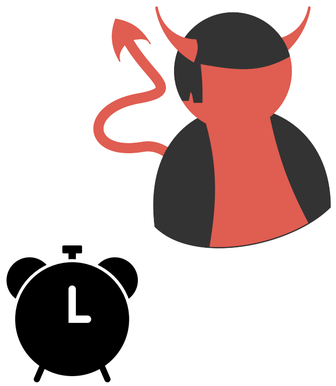


Source: Google Quantum AI



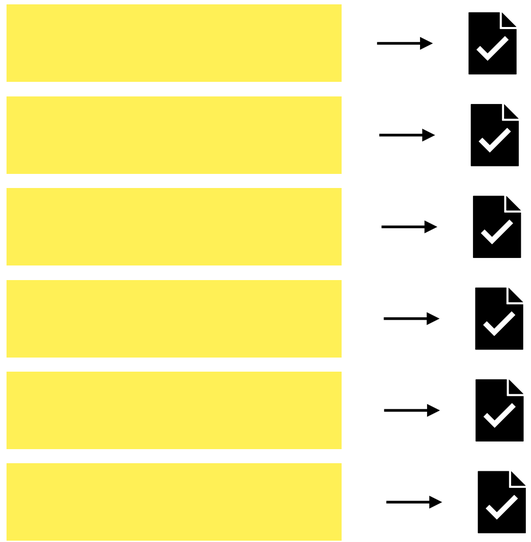
Why Worry About Quantum Computers Today?

Source: Google Quantum AI



Ciphertext

Harvest Now, Decrypt Later



Military and State Secrets

Venona Project

Intellectual property

Privacy and Regulation

GDPR fines for not keeping data confidential for mandated lifetime

Post-Quantum Cryptography

How to build post-quantum secure public-key primitives?

Post-Quantum Cryptography

How to build post-quantum secure public-key primitives?

Algebraic Structure in Assumptions

Post-Quantum Cryptography

How to build post-quantum secure public-key primitives?

Hash functions

Algebraic Structure in Assumptions

Post-Quantum Cryptography

How to build post-quantum secure public-key primitives?

Hash functions

Digital signature schemes

Does not give public-key encryption

Algebraic Structure in Assumptions

Post-Quantum Cryptography

How to build post-quantum secure public-key primitives?

Hash functions

Digital signature schemes

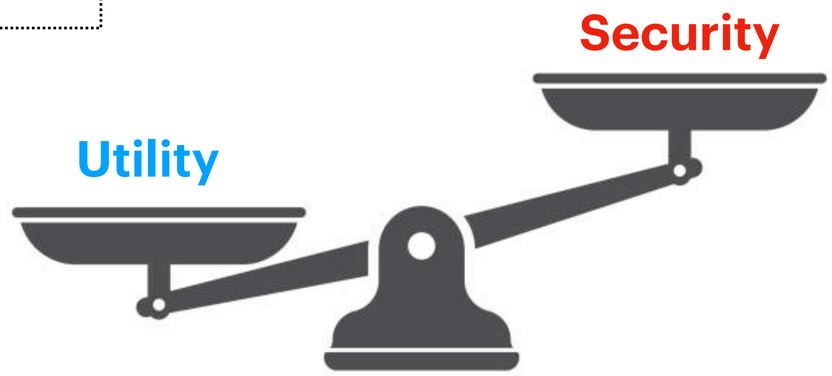
Does not give public-key encryption

Lattice-based

Code-based

Isogeny-based

Multivariate polynomial-based



Algebraic Structure in Assumptions

Post-Quantum Cryptography

How to build post-quantum secure public-key primitives?

Hash functions

Digital signature schemes

Does not give public-key encryption

Lattice-based

Code-based

Isogeny-based

Multivariate polynomial-based

Algebraic Structure in Assumptions

Post-Quantum Cryptography

- NIST PQC Standardization
  - Announced in 2016
  - Public-key Encryption, Key-encapsulation Mechanisms (KEM), and Digital Signatures
  - Three schemes finalized in August 2024
    - Module-Lattice-Based KEM (ML-KEM)
    - Module-Lattice-Based Digital Signature Algorithm (ML-DSA)
    - Stateless Hash-Based Digital Signature Algorithm (SLH-DSA)
      - Backup, in case ML-DSA proves vulnerable

Post-Quantum Cryptography

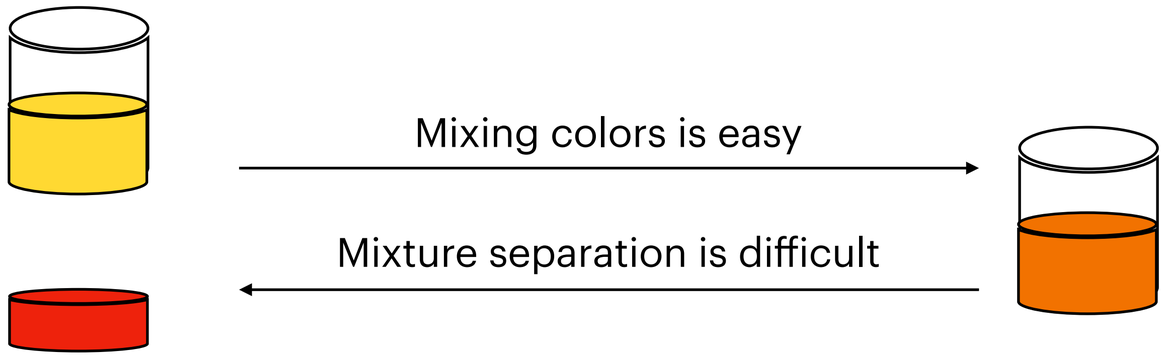
- NIST PQC Standardization
  - Announced in 2016
  - Public-key Encryption, Key-encapsulation Mechanisms (KEM), and Digital Signatures
  - Three schemes finalized in August 2024
    - Module-Lattice-Based KEM (ML-KEM)
    - Module-Lattice-Based Digital Signature Algorithm (ML-DSA)
    - Stateless Hash-Based Digital Signature Algorithm (SLH-DSA)
      - Backup, in case ML-DSA proves vulnerable

Key Exchange

Key Exchange

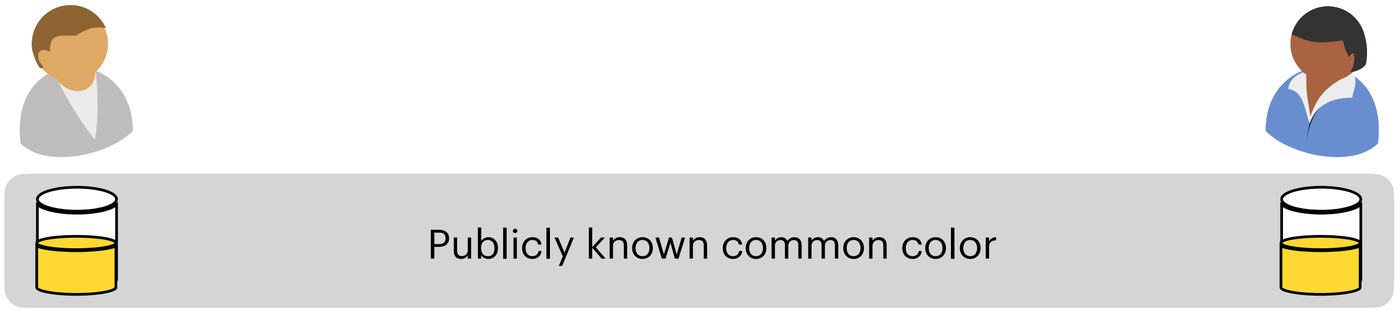
Agreeing on a common secret over an untrusted/public channel

Key Idea: Exploiting asymmetry

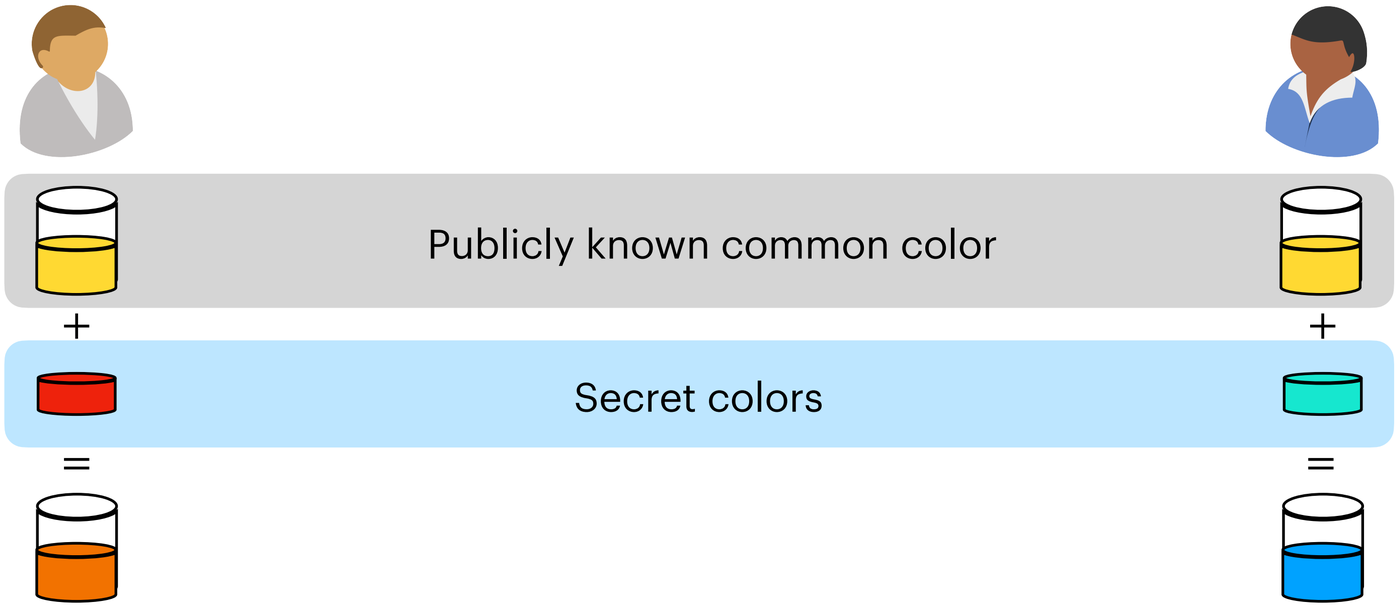


Basic Linear Algebra Review

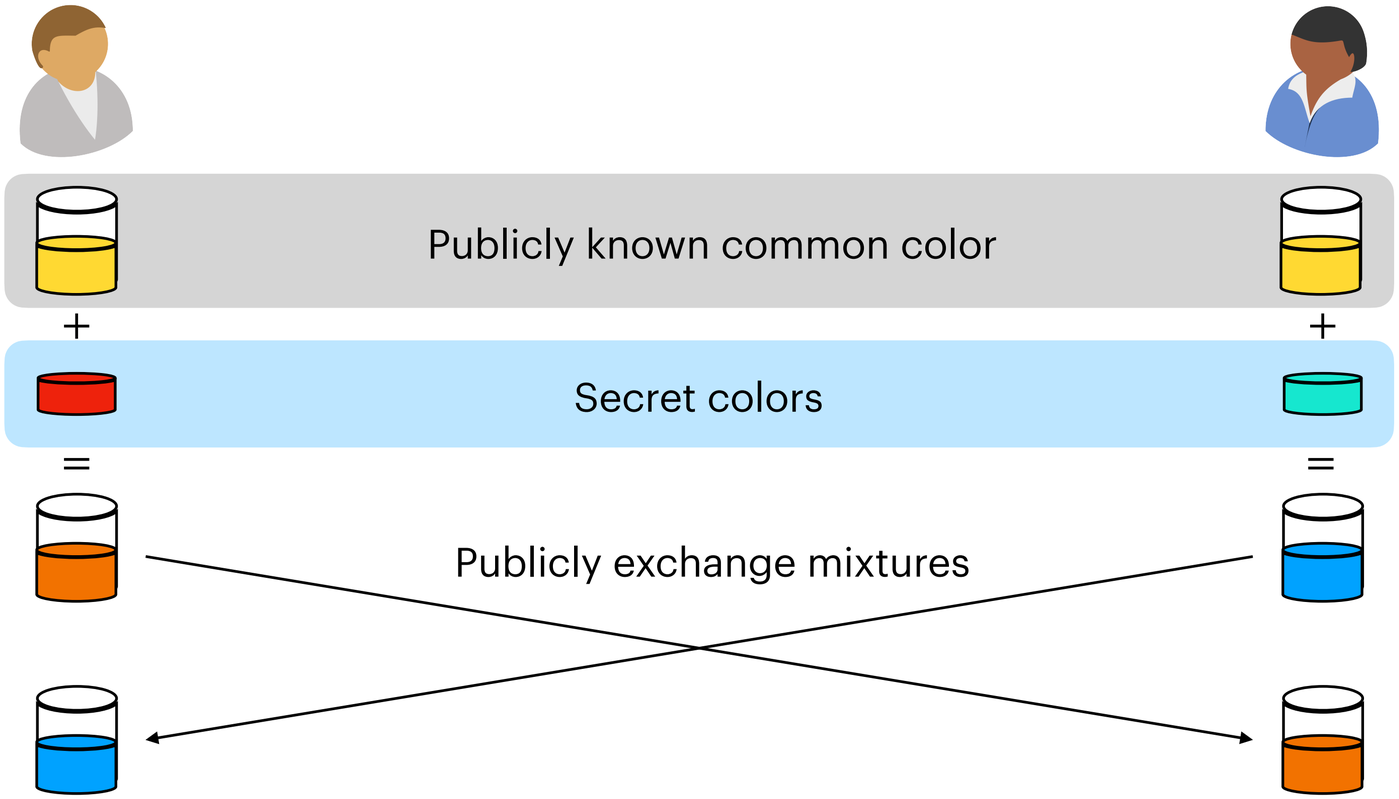
Template for Key Exchange



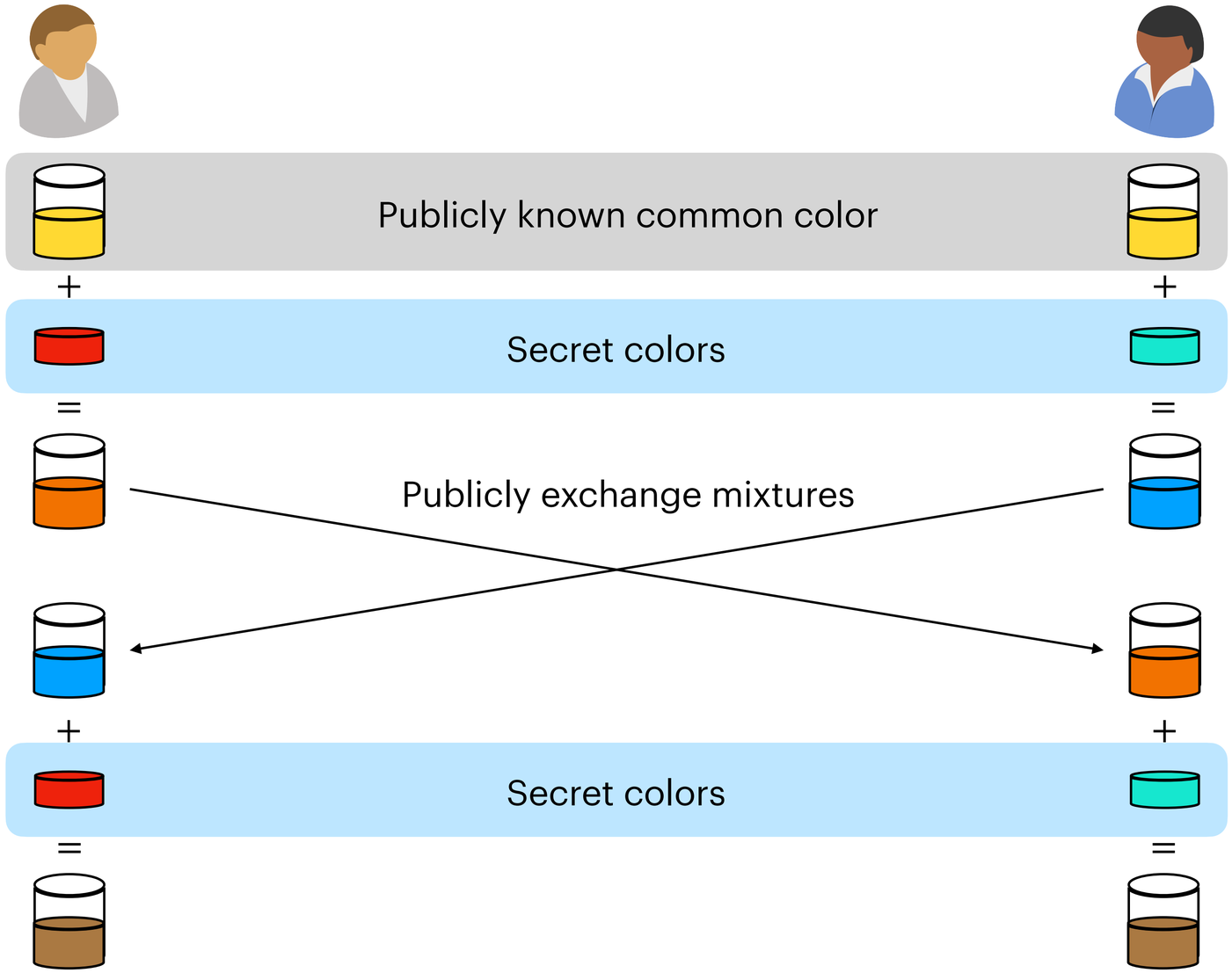
Template for Key Exchange



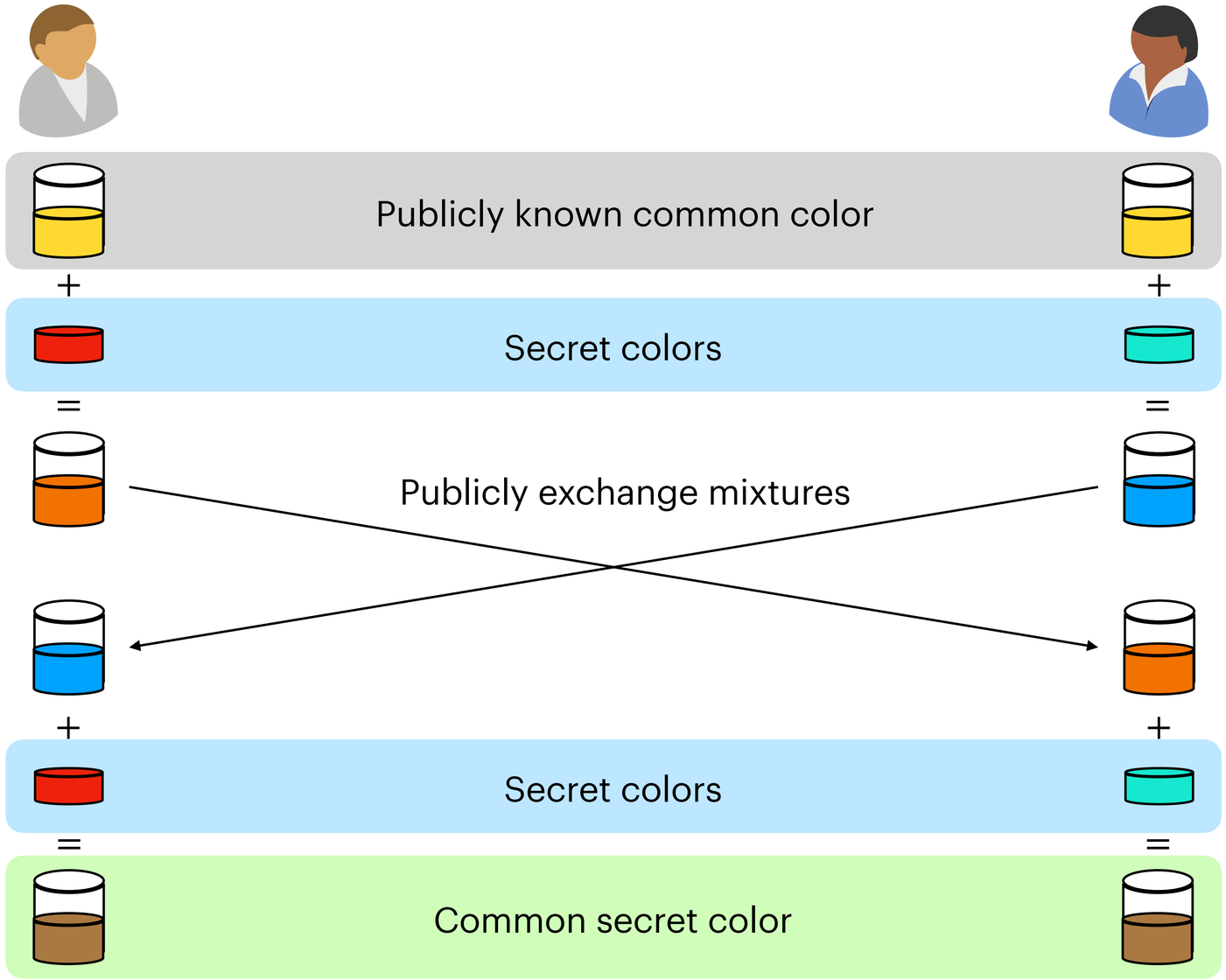
Template for Key Exchange



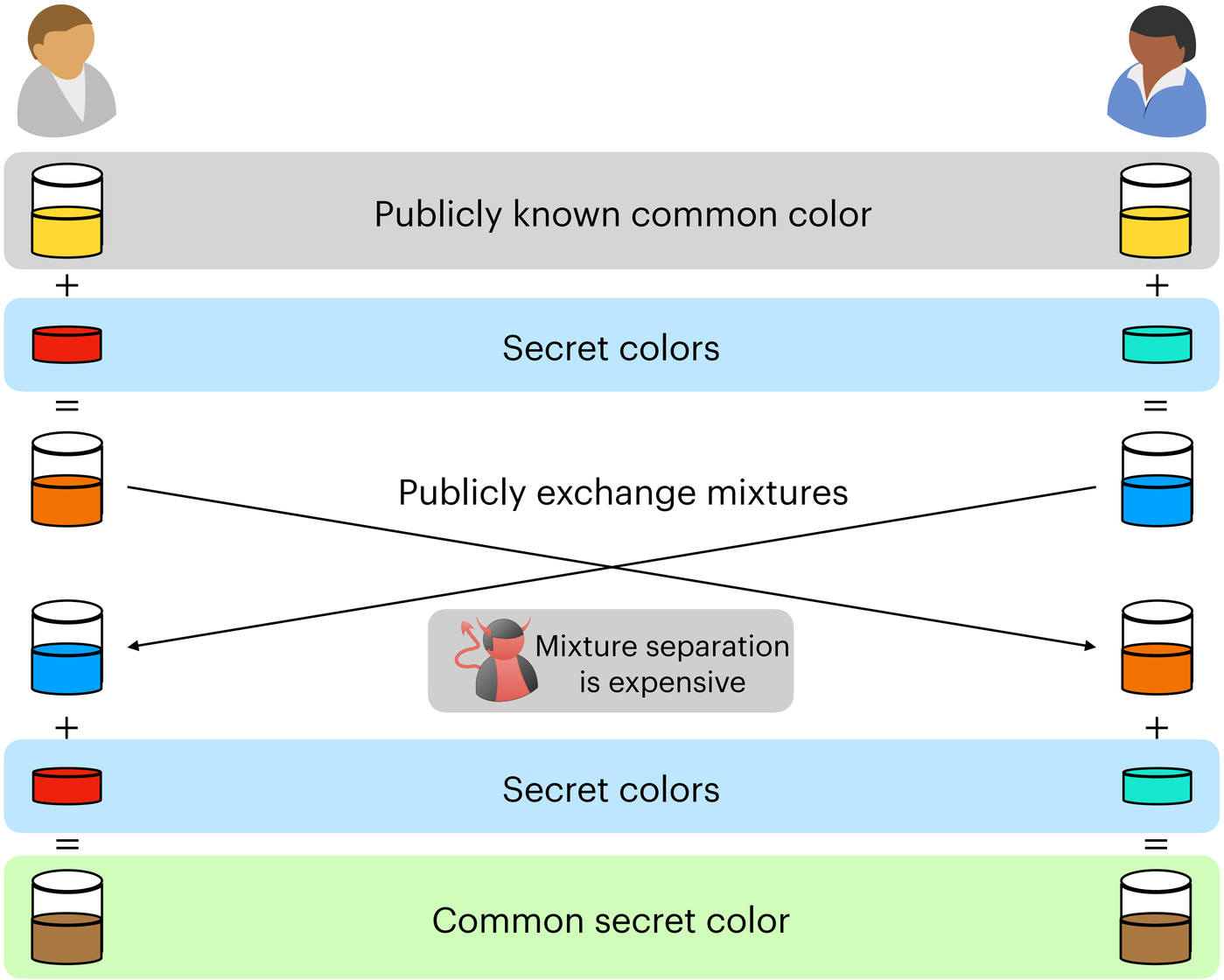
Template for Key Exchange



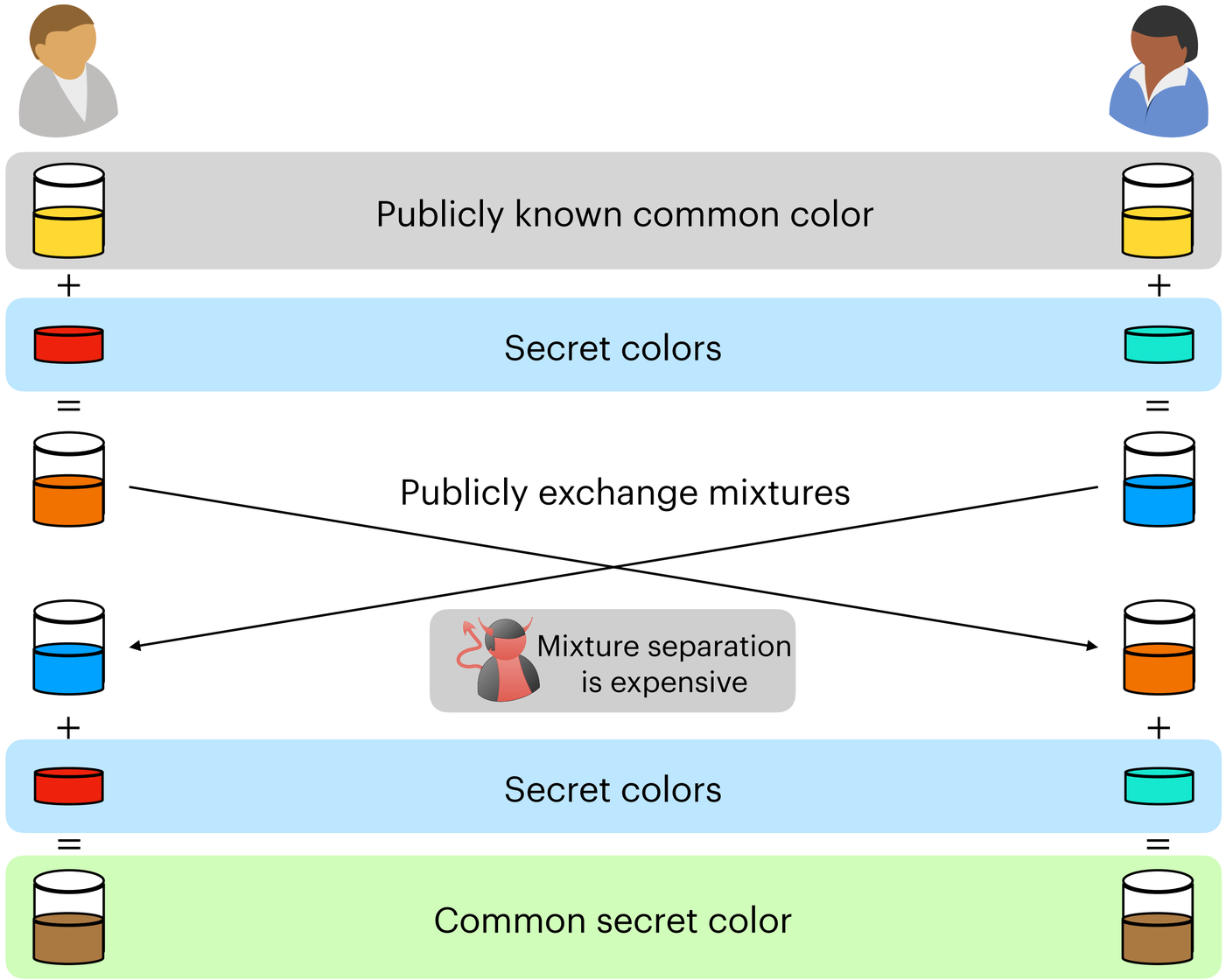
Template for Key Exchange



Template for Key Exchange

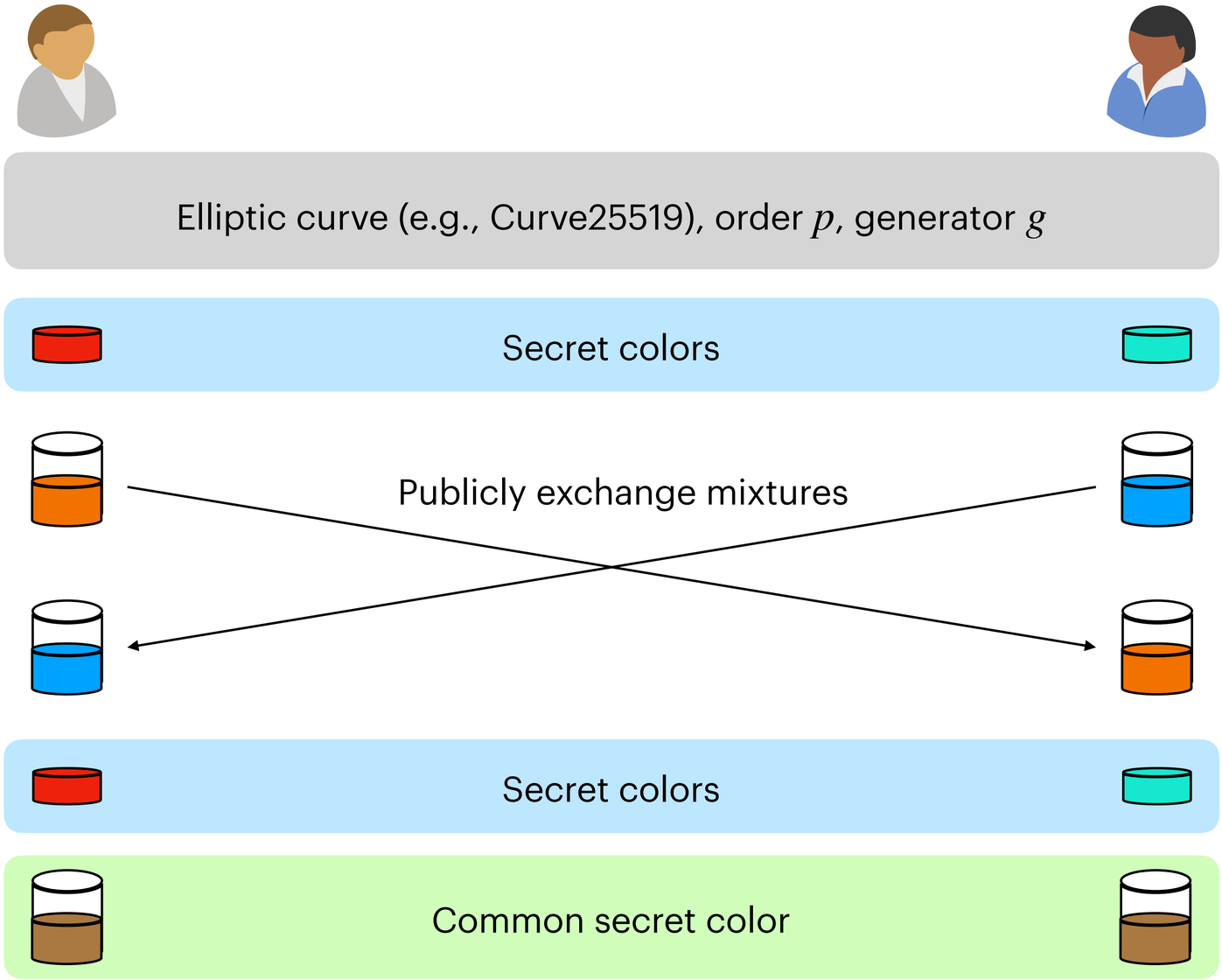


Template for Key Exchange

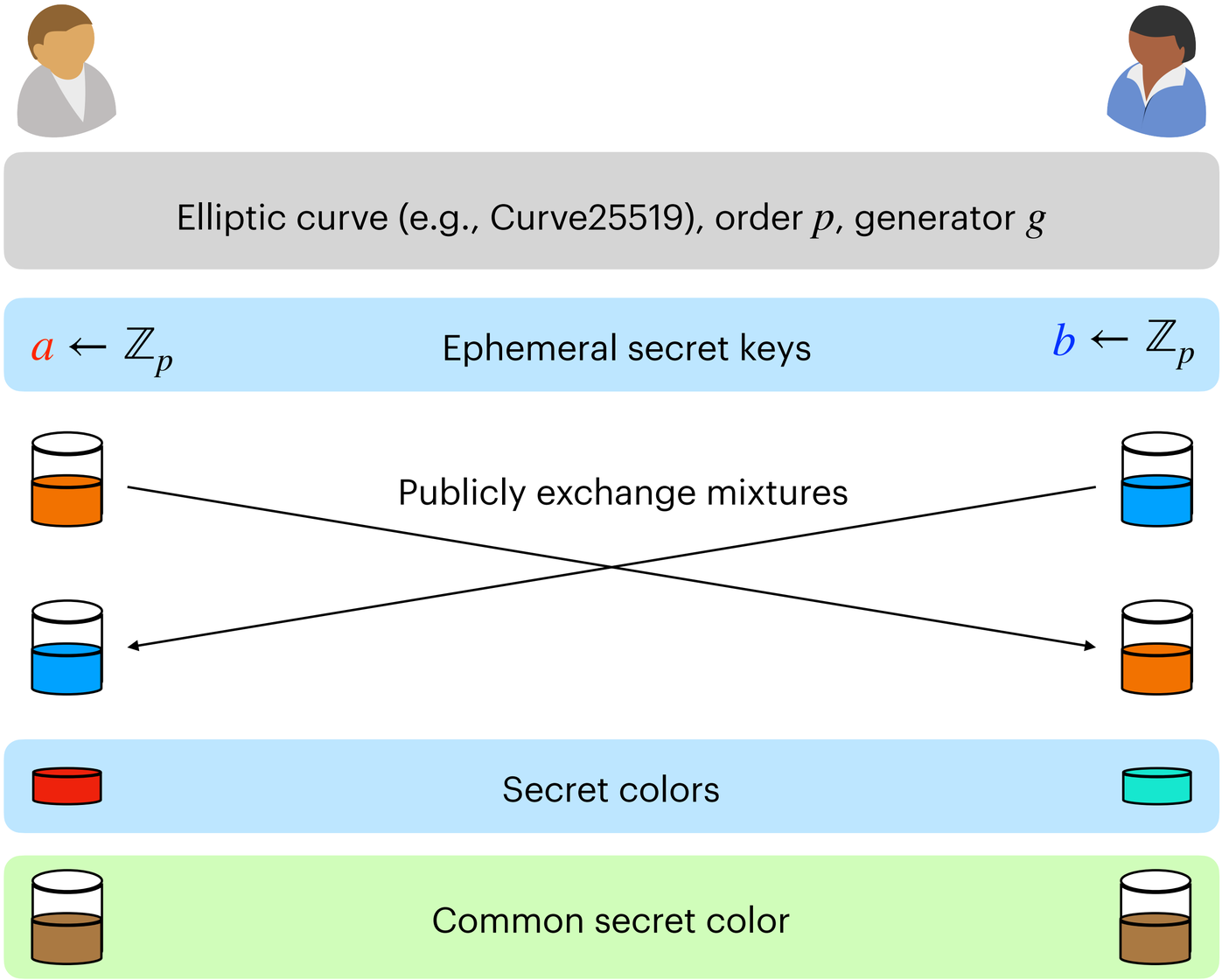


Mathematical equivalent of hardness of mixture separation?

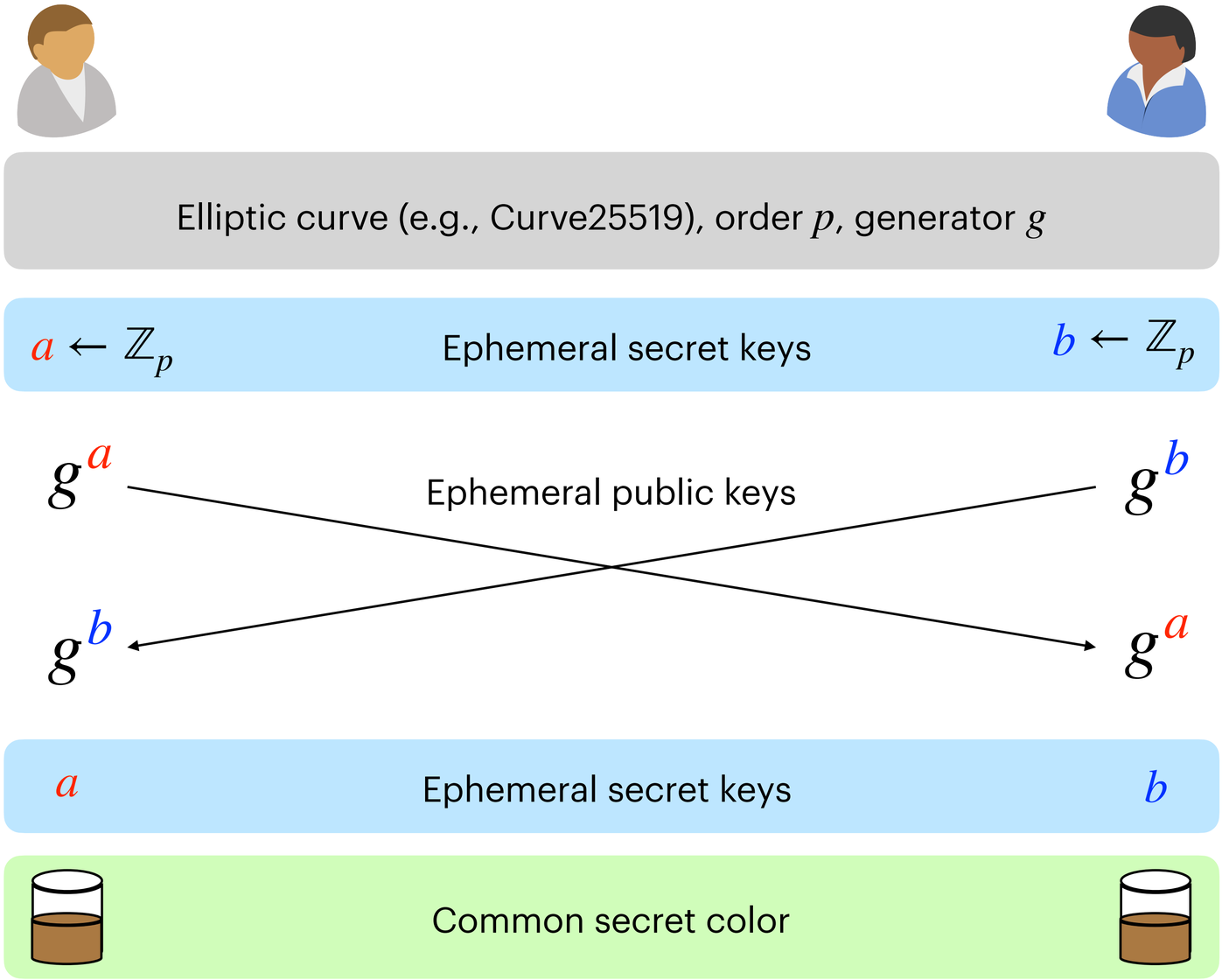
Diffie-Hellman Key Exchange



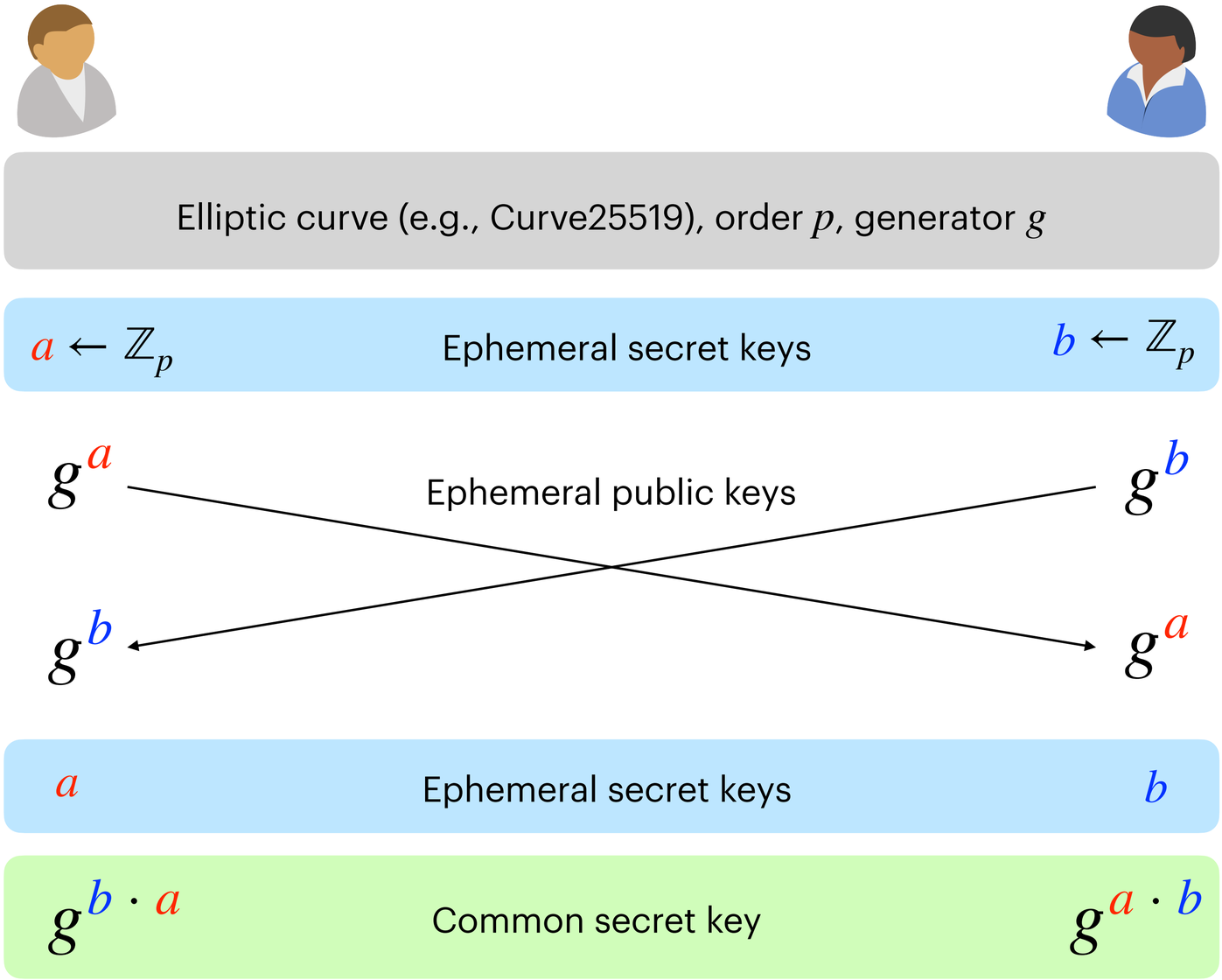
Diffie-Hellman Key Exchange



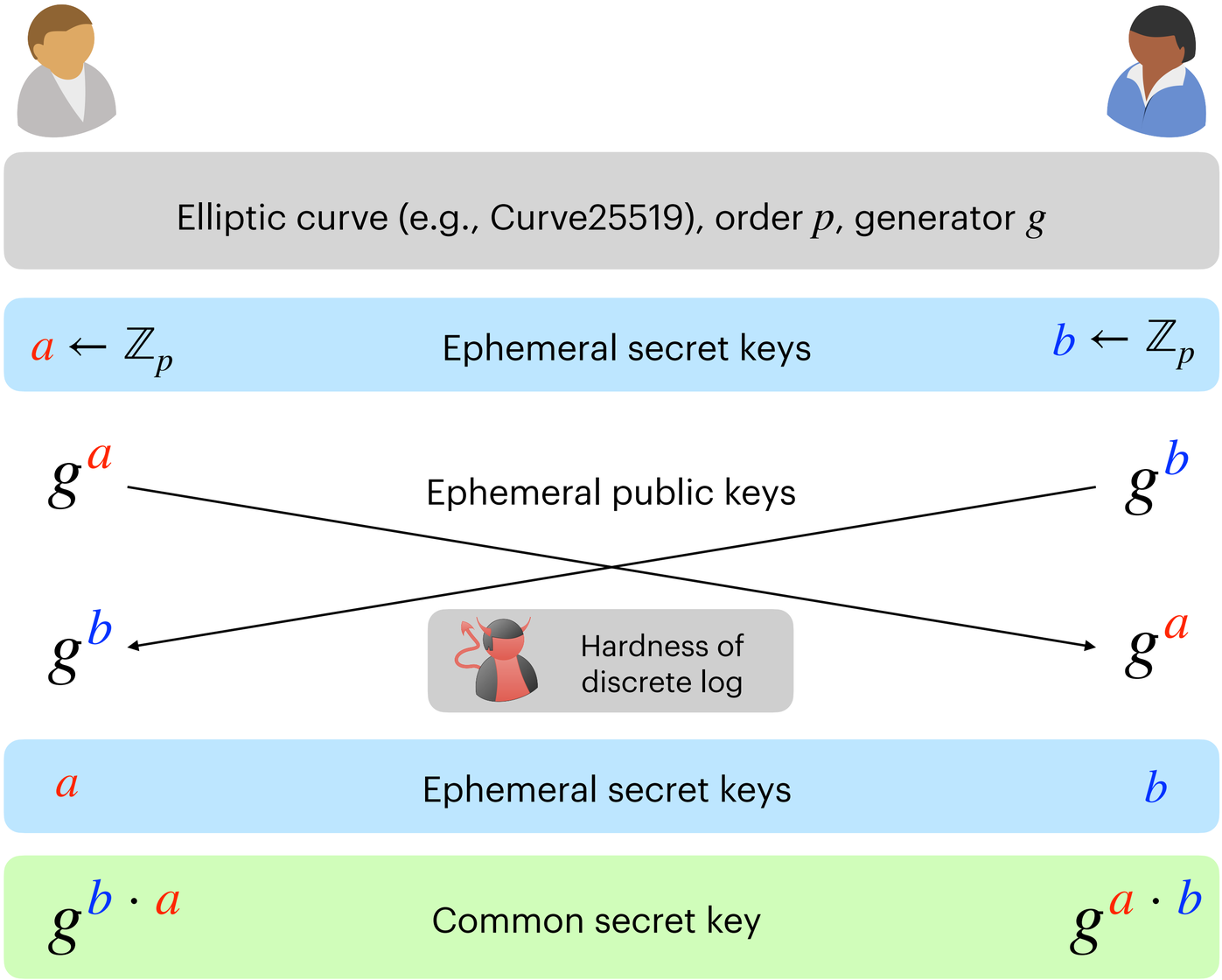
Diffie-Hellman Key Exchange



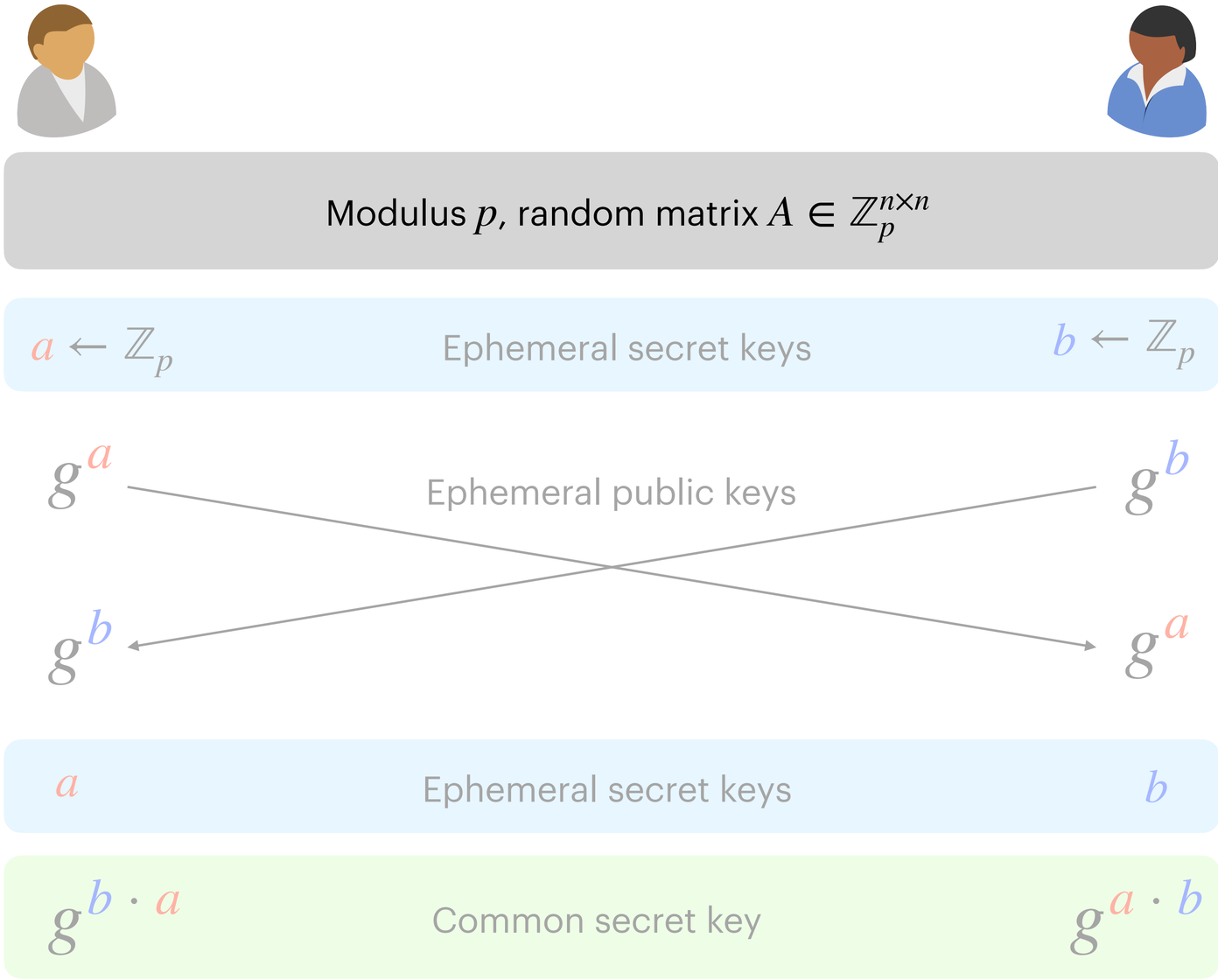
Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange



Towards Post-Quantum Secure Key Exchange



Towards Post-Quantum Secure Key Exchange

Modulus p, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$A \cdot a$ Ephemeral public keys $b^T \cdot A$

$b^T \cdot A$ $A \cdot a$

a Ephemeral secret keys b

$g^{b \cdot a}$ Common secret key $g^{a \cdot b}$

Towards Post-Quantum Secure Key Exchange

Modulus p, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$A \cdot a$ Ephemeral public keys $b^T \cdot A$

$b^T \cdot A$ $A \cdot a$

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

Modulus p, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$A \cdot a$ Ephemeral public keys $b^T \cdot A$

Insecure!

$b^T \cdot A$ $A \cdot a$

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

Modulus p, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$A^{-1} \cdot (A \cdot a)$

$A \cdot a$ Ephemeral public keys $b^T \cdot A$

$\left(b^T \cdot A\right) \cdot A^{-1}$

$b^T \cdot A$ $A \cdot a$

Insecure!

a Ephemeral secret keys b

$\left(b^T \cdot A\right) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

χ is a noise distribution over $\mathbb{Z}_p^n$

Short vectors: Every element in $x \leftarrow \chi$ is much smaller than p

Ephemeral secret keys

 $\leftarrow \chi$

$A \cdot a +$ 

Ephemeral public keys

$b^T \cdot A +$ 

 $\leftarrow \chi$

Towards Post-Quantum Secure Key Exchange

χ is a noise distribution over $\mathbb{Z}_p^n$

Short vectors: Every element in $x \leftarrow \chi$ is much smaller than p

Ephemeral secret keys

$\blacksquare \leftarrow \chi$ $\quad A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$ $\quad \blacksquare \leftarrow \chi$

$A^{-1} \cdot (A \cdot a) + A^{-1} \cdot \blacksquare$

Towards Post-Quantum Secure Key Exchange

χ is a noise distribution over $\mathbb{Z}_p^n$

Short vectors: Every element in $x \leftarrow \chi$ is much smaller than p

Ephemeral secret keys

$\blacksquare \leftarrow \chi$ $\quad A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$ $\quad \blacksquare \leftarrow \chi$

$$A^{-1} \cdot (A \cdot a) + A^{-1} \cdot \blacksquare$$

$$= a + A^{-1} \cdot \blacksquare$$

Towards Post-Quantum Secure Key Exchange

χ is a noise distribution over $\mathbb{Z}_p^n$

Short vectors: Every element in $x \leftarrow \chi$ is much smaller than p

Ephemeral secret keys

$\blacksquare \leftarrow \chi$ $\quad A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$ $\quad \blacksquare \leftarrow \chi$

$A^{-1} \cdot (A \cdot a) + A^{-1} \cdot \blacksquare$

$= a + A^{-1} \cdot \blacksquare$

$\left(b^T \cdot A\right) \cdot A^{-1} + \blacksquare A^{-1}$

$= b^T + \blacksquare A^{-1}$

Towards Post-Quantum Secure Key Exchange

χ is a noise distribution over $\mathbb{Z}_p^n$

Short vectors: Every element in $x \leftarrow \chi$ is much smaller than p

Ephemeral secret keys

$\blacksquare \leftarrow \chi$ $\quad A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$ $\quad \blacksquare \leftarrow \chi$

Not an assumption, basic probability analysis

$A^{-1} \cdot (A \cdot a) + A^{-1} \cdot \blacksquare$

$= a + A^{-1} \cdot \blacksquare$

Property of χ

Random matrix times noise vector is a random vector in $\mathbb{Z}_p^n$

$(b^T \cdot A) \cdot A^{-1} + \blacksquare A^{-1}$

$= b^T + \blacksquare A^{-1}$

Towards Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$A \cdot a$ Ephemeral public keys $b^T \cdot A$

$b^T \cdot A$ $A \cdot a$

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$A \cdot a$ Ephemeral public keys $b^T \cdot A$

$b^T \cdot A$ $A \cdot a$

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$\blacksquare \leftarrow \chi$ $\blacksquare \leftarrow \chi$

$A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$

$b^T \cdot A + \blacksquare$ $A \cdot a + \blacksquare$

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$\blacksquare \leftarrow \chi$ $\blacksquare \leftarrow \chi$

$A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$

$b^T \cdot A + \blacksquare$ $A \cdot a + \blacksquare$

Property of χ

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a$ Common secret key $b^T \cdot (A \cdot a)$

Towards Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$\blacksquare \leftarrow \chi$ $\blacksquare \leftarrow \chi$

$A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$

$b^T \cdot A + \blacksquare$ $A \cdot a + \blacksquare$

Property of χ

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a + \blacksquare \cdot a$ Common secret key $b^T \cdot (A \cdot a) + b^T \cdot \blacksquare$

Towards Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

$a \leftarrow \mathbb{Z}_p^n$ Ephemeral secret keys $b \leftarrow \mathbb{Z}_p^n$

$\blacksquare \leftarrow \chi$ $\blacksquare \leftarrow \chi$

$A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$

$b^T \cdot A + \blacksquare$ $A \cdot a + \blacksquare$

Property of χ

Not the same common key!

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a + \blacksquare \cdot a$ Common secret key $b^T \cdot (A \cdot a) + b^T \cdot \blacksquare$

Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

Sample short keys

$a \leftarrow \chi$ Ephemeral secret keys $b \leftarrow \chi$

$\blacksquare \leftarrow \chi$ $\blacksquare \leftarrow \chi$

$A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$

$b^T \cdot A + \blacksquare$ $A \cdot a + \blacksquare$

Property of χ

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a + \blacksquare \cdot a$ Common secret key $b^T \cdot (A \cdot a) + b^T \cdot \blacksquare$

Post-Quantum Secure Key Exchange

Modulus p, distribution χ, random matrix $A \in \mathbb{Z}_p^{n \times n}$

Sample short keys

$a \leftarrow \chi$ Ephemeral secret keys $b \leftarrow \chi$

$\blacksquare \leftarrow \chi$ $\blacksquare \leftarrow \chi$

$A \cdot a + \blacksquare$ Ephemeral public keys $b^T \cdot A + \blacksquare$

$b^T \cdot A + \blacksquare$ $A \cdot a + \blacksquare$

Property of χ

a Ephemeral secret keys b

$(b^T \cdot A) \cdot a + \blacksquare \cdot a$ Common secret key $b^T \cdot (A \cdot a) + b^T \cdot \blacksquare$

Post-Quantum Secure Key Exchange

Common secret key

$\left(b^T \cdot A\right) \cdot a$ + [redacted] $\cdot a$

Common random value

Small error

If this value is closer to $p/2$ output 0

Else output 1

$b^T \cdot (A \cdot a)$ + $b^T \cdot$ [redacted]

Common random value

Small error

If this value is closer to $p/2$ output 0

Else output 1

- Because the errors are small, rounding outputs the same bit with very high probability.
- Security is based on the Learning with Errors assumption

$$(A,\ A \cdot x\ +\ \text{[redacted]}\ ) \approx (A,\ u)$$

Where A is a uniformly random matrix in $\mathbb{Z}_p^{n \times n}$, $x \leftarrow \chi$, $u \leftarrow \mathbb{Z}_p^n$, and [redacted] $\leftarrow \chi$

Post-Quantum KEM

- **Lot of parameters:** modulus p, noise distribution χ, dimension of matrix n
  - Needs to be set carefully to ensure security
  - **Module-LWE:** Practical schemes work over polynomial rings for efficiency instead of $\mathbb{Z}_p$
  - Ephemeral public-key size for Module-LWE key exchange: 221 KB
  - Ephemeral public-key size for Curve25519 key exchange: 32 bytes
- **Key encapsulation mechanism:** Similar to Public-key encryption of a random key; key is used to encrypt message under symmetric-key encryption scheme
  - Also ensures integrity of ciphertext
  - ML-KEM public-key size: 1,184 bytes
  - ML-KEM ciphertext size: 1088 bytes