Post Quantum Cryptography

Brief Introduction

Hee Ryang Choi

SNU SQRT

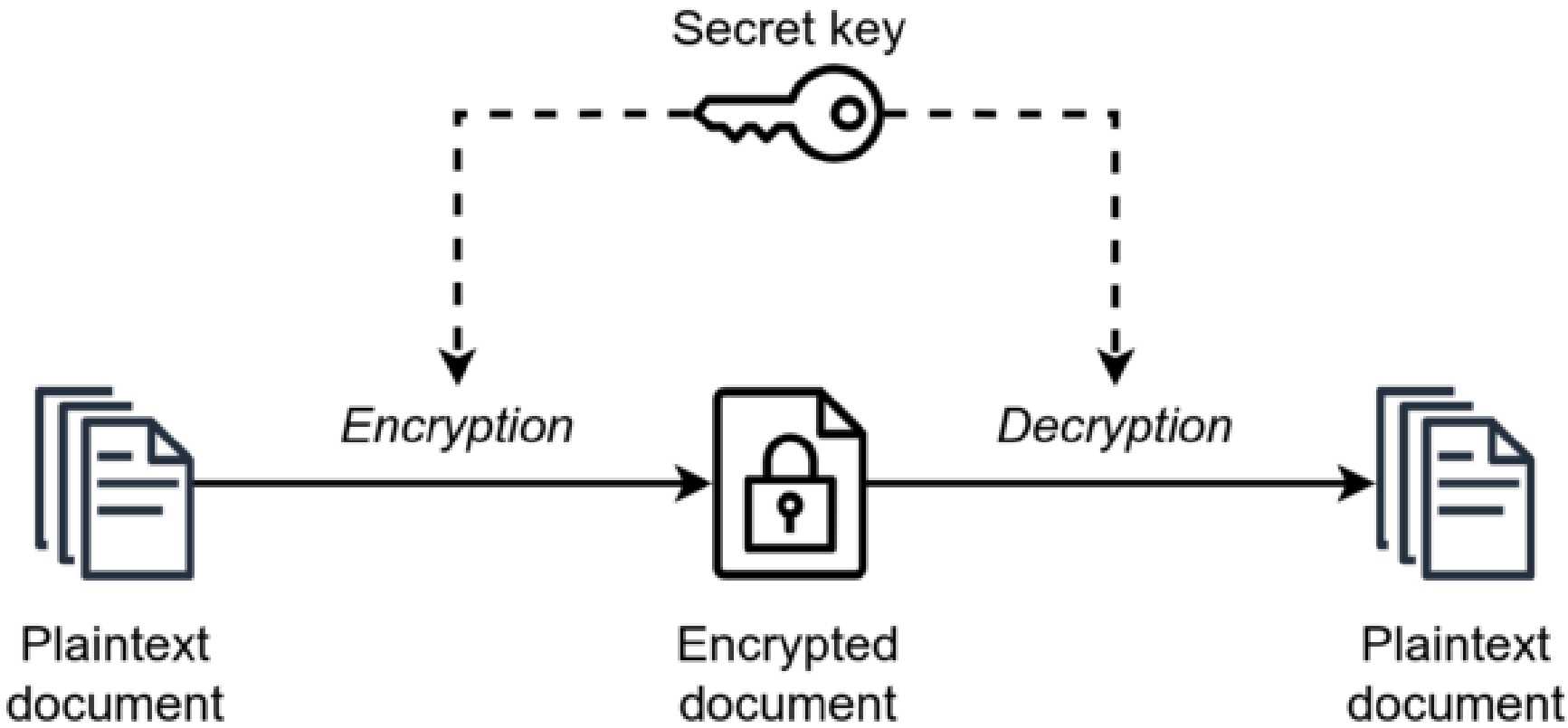
November 8, 2023

Overview

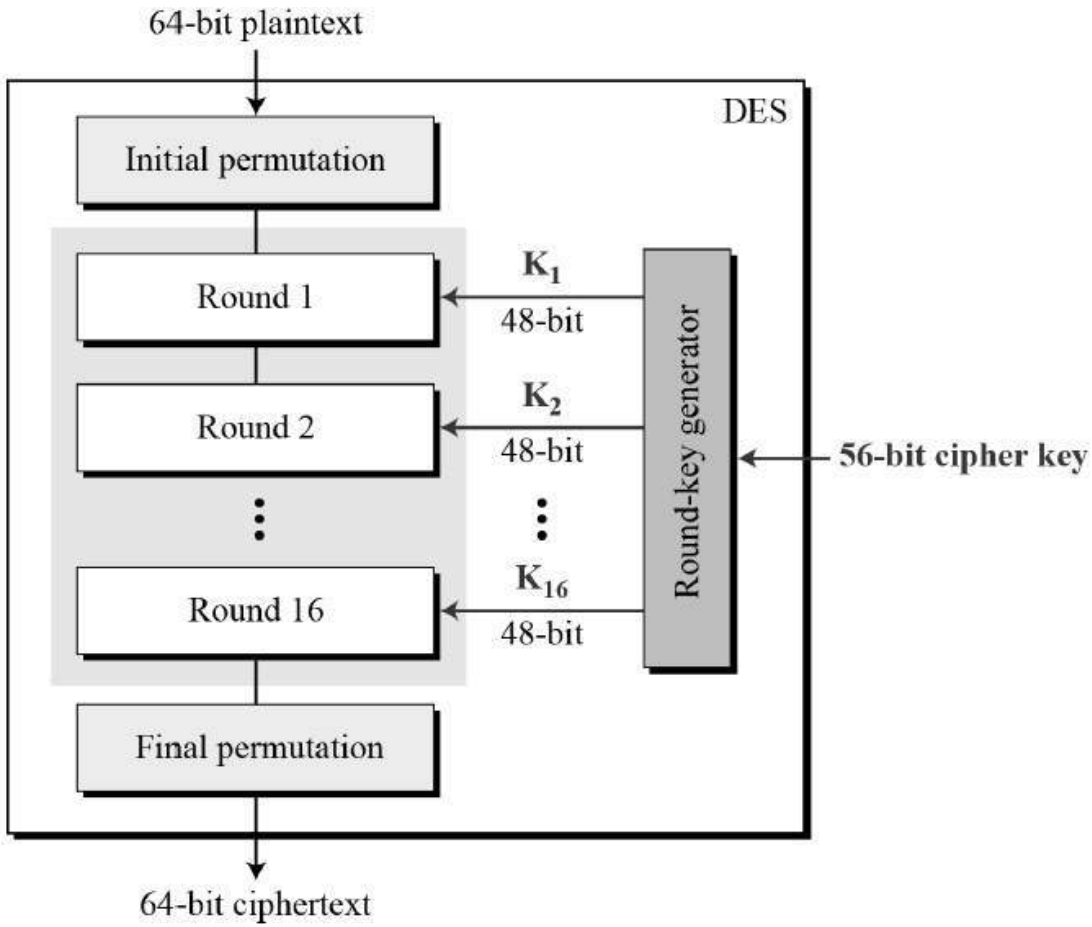
1. Description of Pre-Quantum Cryptography
   - Symmetric Key Encryption
   - Public Key Encryption
   - Elliptic Curve Cryptography
2. Security with Quantum Computer
   - Grover's Algorithm with Collision Problem
   - Shor's Algorithm with DLP
   - Simon's Algorithm with Collision Problem
   - Quantum Key Exchange
3. Post Quantum Cryptography
   - Hash Collision Problem
   - Multivariate Problem
   - Lattice Cryptography

Overview

Symmetric Key Encrpytion

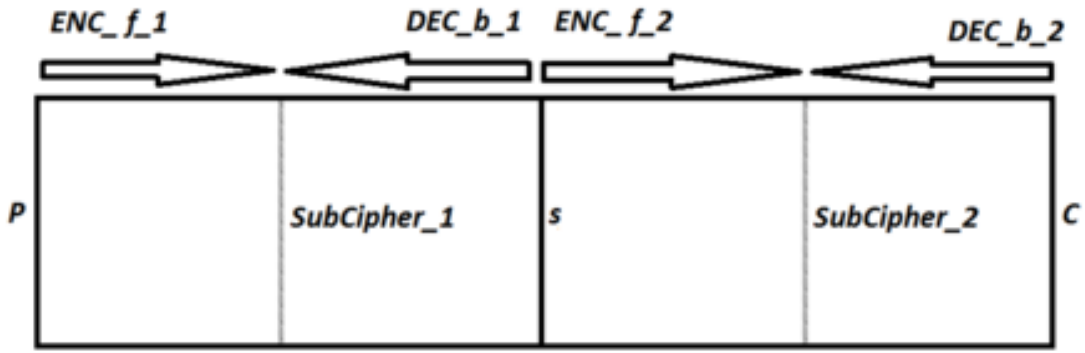


Data Encryption Standard

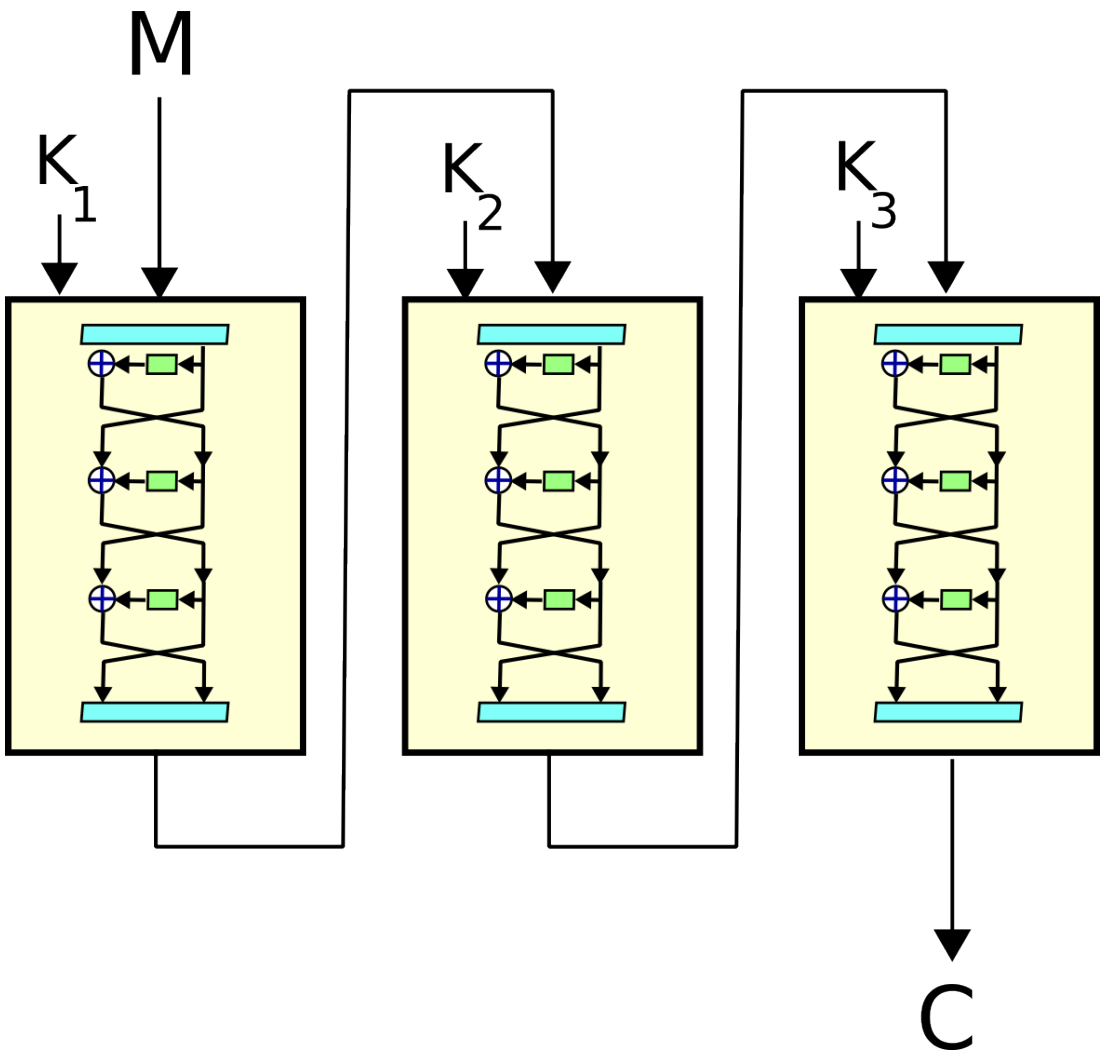


Data Encryption Standard

DES is a problem that is very easily solved by classical computers, not quantum computers. This method is known as Meet in the Middle.



3-DES

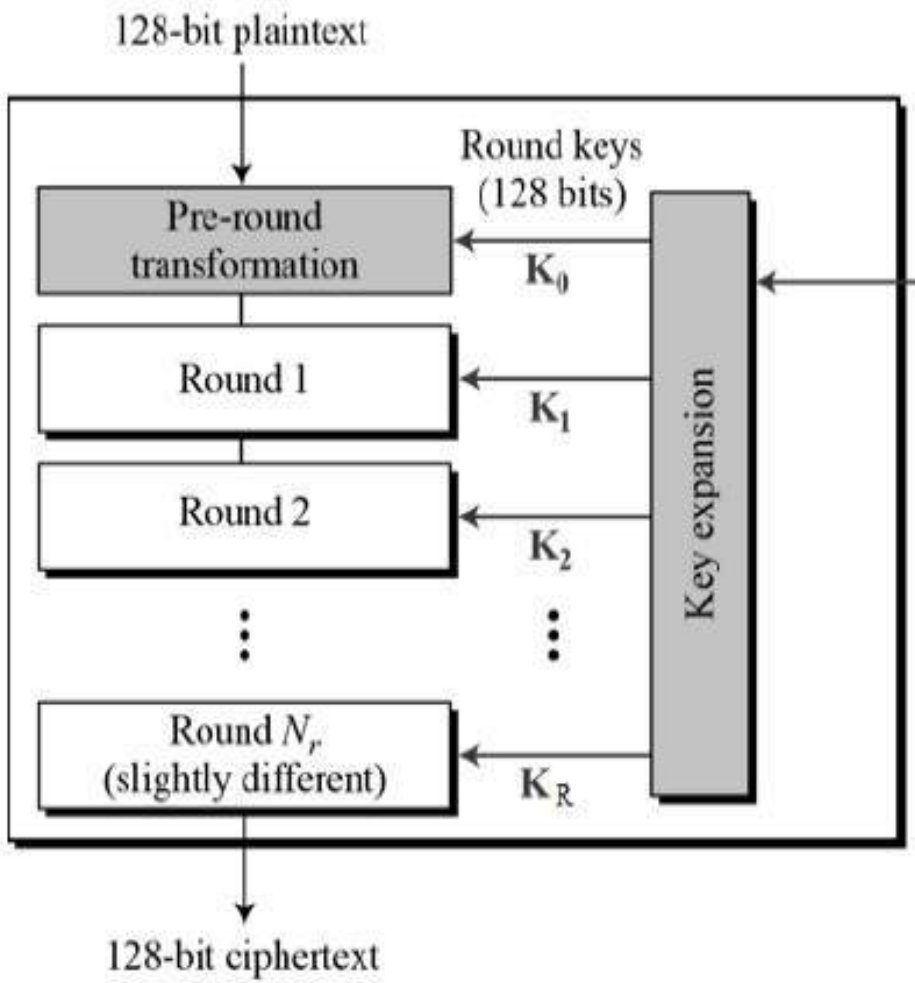


3-DES

It can be easily attacked using the concept of the birthday paradox; known as the Sweet-32 attack, this method statistically analyzes 2^{32} blocks to determine the value.

Advanced Encryption Standard

Also called **Rijndael block cipher**



R	Key size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size

Advanced Encryption Standard

- SubByte: In this stage, each byte is substituted with a value given by a predefined relation. These substitutions are carried out using predetermined S-boxes consisting of 256 values
- ShiftRows: In this stage, the elements obtained from the S-boxes are then shifted right by guidelines specified in the NIST standard, or chosen independently on a need-by-need basis.

Advanced Encryption Standard

- MixColumns: Each column is then read as a polynomial and then multiplied by an encoding polynomial. The result is then divided by an irreducible
- AddRoundKey: A round key generated by a simple key schedule is added to the state array.

The Decryption Algorithm is the same algorithm with the same number of rounds, in reverse order.

Advanced Encryption Standard

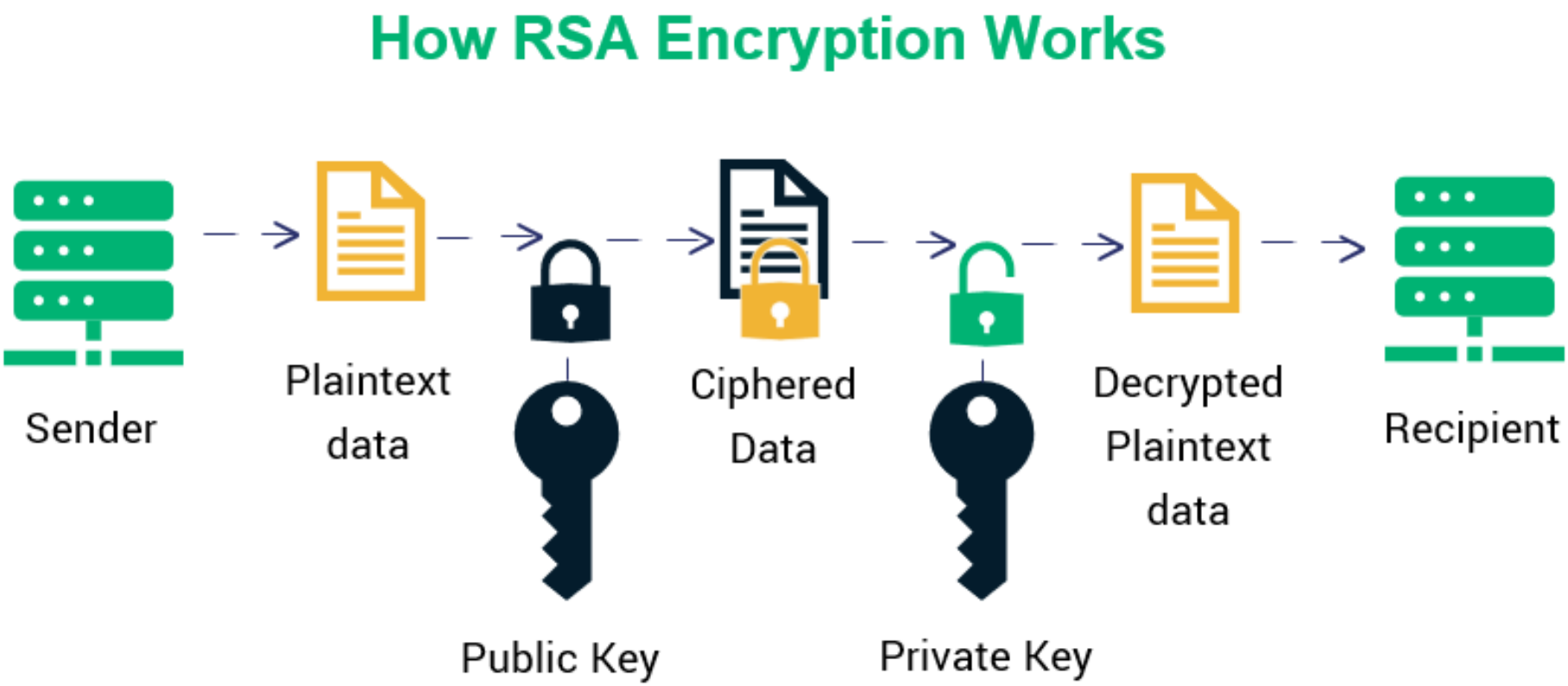
Security of AES is verified by NSA

NSA review

The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either the 192 or 256 key lengths.

Overview

Public Key Encrpytion



Rivest-Shamir-Adelman Crpytosystem

RSA Key Generation

Output: public key: $k_{pub} = (n, e)$ and private key: $k_{pr} = (d)$

1. Choose two large primes p and q.
2. Compute $n = p \cdot q$.
3. Compute $\Phi(n) = (p-1)(q-1)$.
4. Select the public exponent $e \in \{1, 2, \ldots, \Phi(n) - 1\}$ such that

$$\gcd(e, \Phi(n)) = 1.$$

5. Compute the private key d such that

$$d \cdot e \equiv 1 \bmod \Phi(n)$$

Rivest-Shamir-Adelman Crpytosystem

The RSA encryption and decryption operations are the following steps:

1. Encryption
   - Represent the plaintext message M as an integer m in the range $[0, n-1]$.
   - Compute the ciphertext c as $c \equiv m^e \mod n$.
2. Decryption
   - Compute the original message m as $m \equiv c^d \mod n$.

The security of RSA comes from the **difficulty of factoring the large number** into its prime factors. So it can be attacked by Shor's algorithm. But if bit size is bigger enough, NISQ can't beat RSA.

The Discrete Logarithm Problem (DLP)

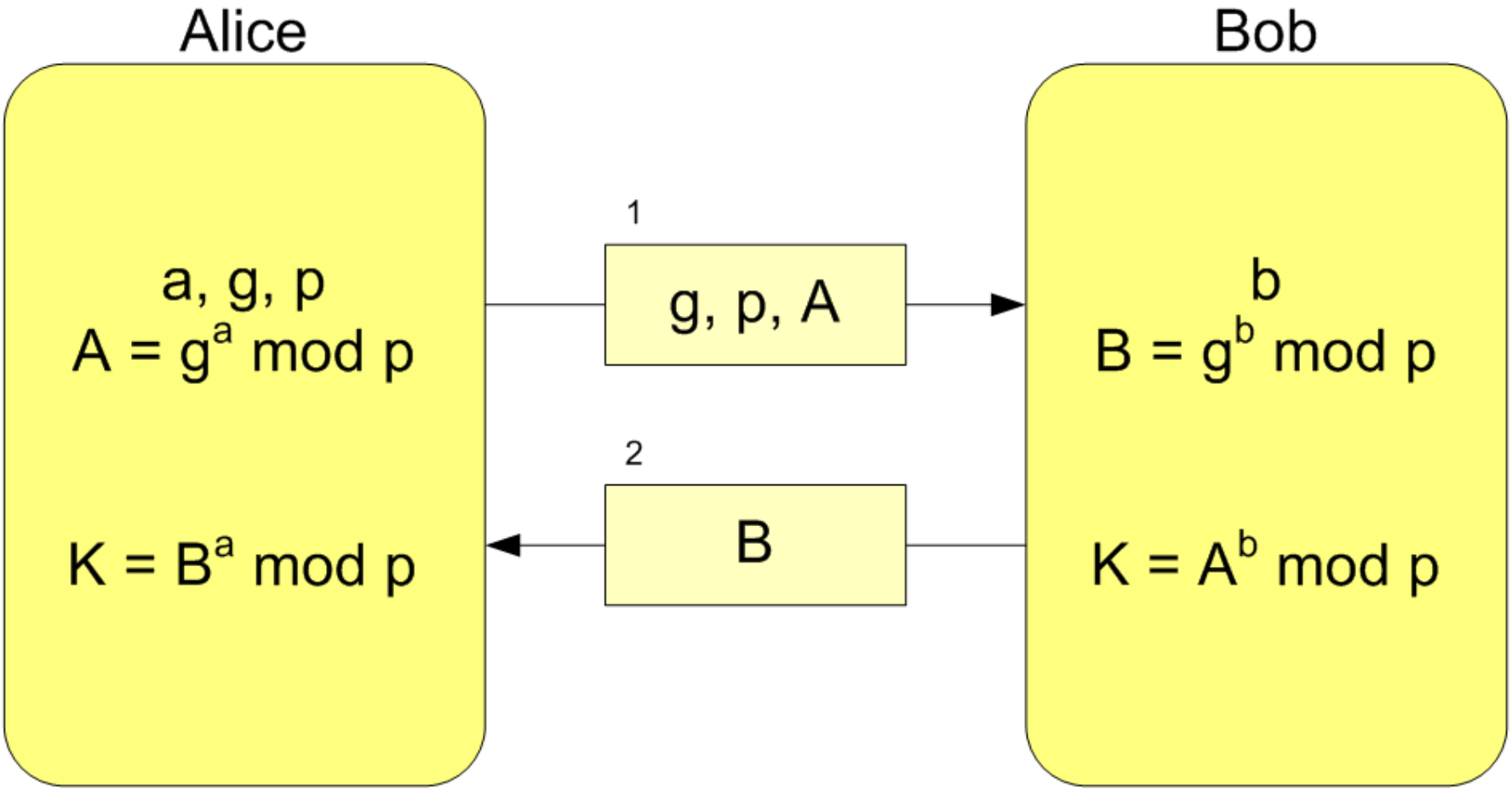
Definition (Discrete Logarithm Problem)

Let G be a finite group of order n, and let g be a generator of G. For any element h in G, the discrete logarithm problem is to find an integer x, $0 \leq x < n$, such that

$$g^x = h.$$

The value x is called the discrete logarithm of h to the base g, denoted as $x = \log_g h$

Diffie-Hellman Key Exchange Protocol



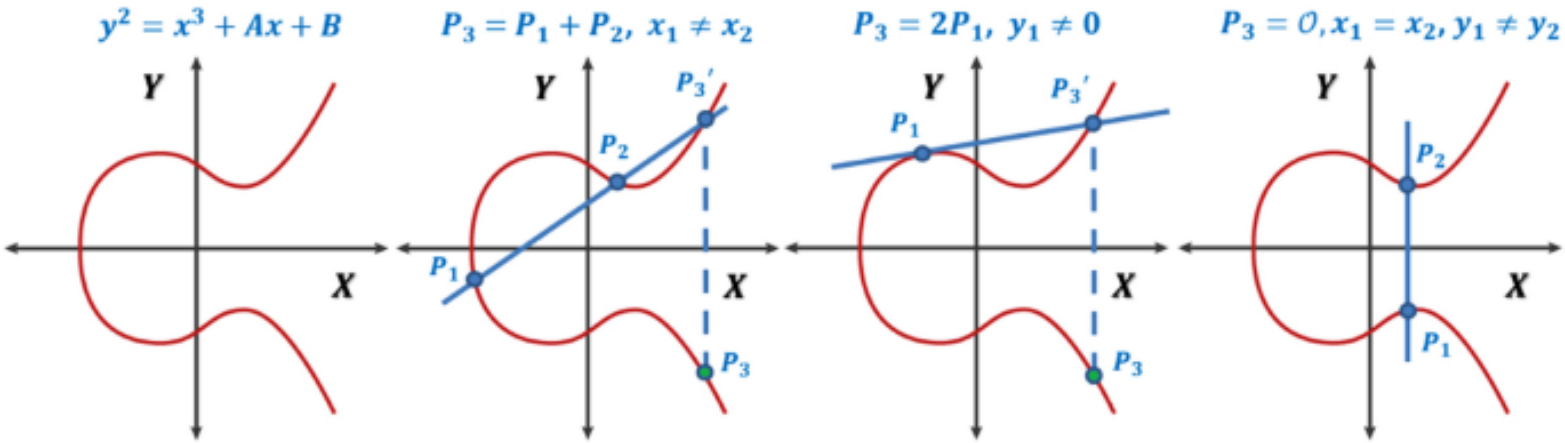
$$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$$

Overview

Elliptic Curve Cryptography (ECC)

- ECC is a form of public key cryptography based on the algebraic structure of elliptic curves over finite fields.
- Provides the same level of security as RSA but with much smaller key sizes – leading to faster computations and lower power consumption.
- Widely used for secure communication protocols, including SSL/TLS for Internet security and encryption schemes for cryptocurrency like Bitcoin.

Elliptic Curves and Their Properties



Elliptic Curve

Elliptic Curve Definition

An elliptic curve $E(\mathbb{F}_p)$ over a finite field $\mathbb{F}_p$ is the set of solutions $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ to the equation

$$y^2 \equiv x^3 + ax + b \mod p$$

including a point at infinity O, with p being a prime.

Elliptic Curve

Group Operation on $E(\mathbb{F}_p)$

The set of points on $E(\mathbb{F}_p)$, together with the point at infinity O, form a group under the addition operation:

- $P + O = O + P = P$
- $P + (-P) = O$
- For $P \neq -R$, $P + R$ is a third point S that also lies on the curve.

Associativity, commutativity, and identity and inverse elements properties are satisfied, making $(E(\mathbb{F}_p), +)$ an abelian group.

Note

In the context of ECDSA, these operations are crucial for key generation, signing, and verification processes.

Elliptic Curves

Elliptic Curve Point Operations

Let $P = (x_1, y_1)$ and $R = (x_2, y_2)$ be points on an elliptic curve E.

- The addition of a point to the identity point O is defined as:

$$P + O = O + P = P$$

- The addition of two distinct points P and R is defined as:

$$P + R = \begin{cases} O & \text{if } (x_1, y_1) = (x_2, -y_2) \\ (x_3, y_3) & \text{otherwise} \end{cases}$$

Elliptic Curves

Calculation of (x_3, y_3)

The coordinates of (x_3, y_3) are calculated as:

$$x_3 = \lambda^2 - (x_1 + x_2), \quad y_3 = \lambda(x_1 - x_3) - y_1$$

where λ is the slope, defined as:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq R \\ \frac{3x_1^2 + a}{2y_2} & \text{if } P = R \end{cases}$$

Note

The operations above are performed modulo a prime number in ECDSA to ensure closure under addition within the finite field.

Key Generation in ECC

- Choose a prime number p to define the finite field, select an elliptic curve E over that field, and choose a base point G on that curve.
- Generate a private key d which is a random number from $[1, n-1]$, where n is the order of the base point G.
- Compute the public key Q by multiplying the base point G by the private key d, $Q = dG$.

Key Generation in ECC

Cyclic Subgroup Generated by a Base Point P

In elliptic curve cryptography, we often focus on a cyclic subgroup of E generated by a base point P. This subgroup, denoted by $\langle P \rangle$, is defined as:

$$\langle P \rangle = \{nP \in E : n \in \mathbb{Z}\}$$

where nP represents the n-fold addition of P with itself.

Key Generation in ECC

Efficient Computation of Multiples of P

The point Q, which is a multiple of P, can be efficiently computed as:

$$Q = dP = \underbrace{P + P + \cdots + P}_{d \text{ times}}$$

This computation can be performed in $O(\log d)$ operations using the method of double-and-add, which is essential for large integers $d = d_1 d_2 \cdots d_n$.

Key Generation in ECC

Example of Scalar Multiplication

The scalar multiplication of P by d, which is represented in binary as $(d_n d_{n-1} \cdots d_1)_2$, is calculated as:

$$Q = \sum_{k=1}^{n} 2^{n-k} P$$

This exploits the binary representation of d, effectively doubling P and adding it to the sum if the corresponding binary digit d_k is 1.

Discrete Logarithm Problem (DLP) on Elliptic Curves

Definition of ECDLP

Given points P and Q on an elliptic curve G, the Discrete Logarithm Problem (DLP) is to find the discrete logarithm $d = \log_P Q$ such that:

$$dP = Q$$

Here, d is an integer and the operation dP denotes the scalar multiplication of point P by d.

Discrete Logarithm Problem on Elliptic Curves

The ECDLP is considered to be computationally infeasible for classical computers, making it the foundation for the security of elliptic curve cryptography (ECC). In the Elliptic Curve Digital

Signature Algorithm (ECDSA), the discrete logarithm d serves as the private key, while $Q = dP$ is used as the corresponding public key.

ECC for Encryption and Decryption

- To encrypt a message, a sender combines the recipient's public key with a randomly selected point on the curve to create a shared secret.
- The sender then uses this shared secret to encrypt the message using a symmetric cipher.
- The recipient can generate the same shared secret using their private key and decrypt the message.

ECC and Digital Signatures

- ECC is commonly used for digital signatures, which provide authenticity and non-repudiation.
- The ECC digital signature algorithm (ECDSA) is used to generate and verify signatures.
- ECDSA is more efficient than RSA and offers equivalent security with shorter key lengths, thus it's commonly used in constrained environments like smart cards and mobile devices.

ECDSA Parameters

Curve Parameters

The elliptic curve is defined by the equation $y^2 = x^3 + ax + b$

Key Parameters

- P: A base point on the curve that generates a subgroup of large prime order q.
- q: The order of the subgroup generated by P, which is a large prime number.
- d: The private key, a randomly selected integer from $[1, q - 1]$.
- Q: The public key, calculated as $Q = dP$, where d is the private key.
- m: The message to be signed.

ECDSA: Signature Generation and Verification

Signature Generation (d, m)

Given a private key d and message m:

1. Calculate z as the L_q leftmost bits of HASH(m), where L_q is the bit length of order q.
2. Select integer k randomly from $[1, q-1]$.
3. Compute $(x_1, y_1) = kP$.
4. Set $r = x_1 \mod q$.
5. The signature is (r, s), where $s = k^{-1}(z + rd) \mod q$.

ECDSA: Signature Generation and Verification

Signature Verification (Q, m, r, s)

Given a public key Q, message m, and signature (r, s):

1. Calculate $u_1 = zs^{-1} \mod q$ and $u_2 = rs^{-1} \mod q$.
2. Compute $(x_1, y_1) = u_1 P + u_2 Q$.
3. If $(x_1, y_1) = O$ (the point at infinity), then the signature is invalid.
4. The signature is valid if $r \equiv x_1 (\mod q)$, invalid otherwise.

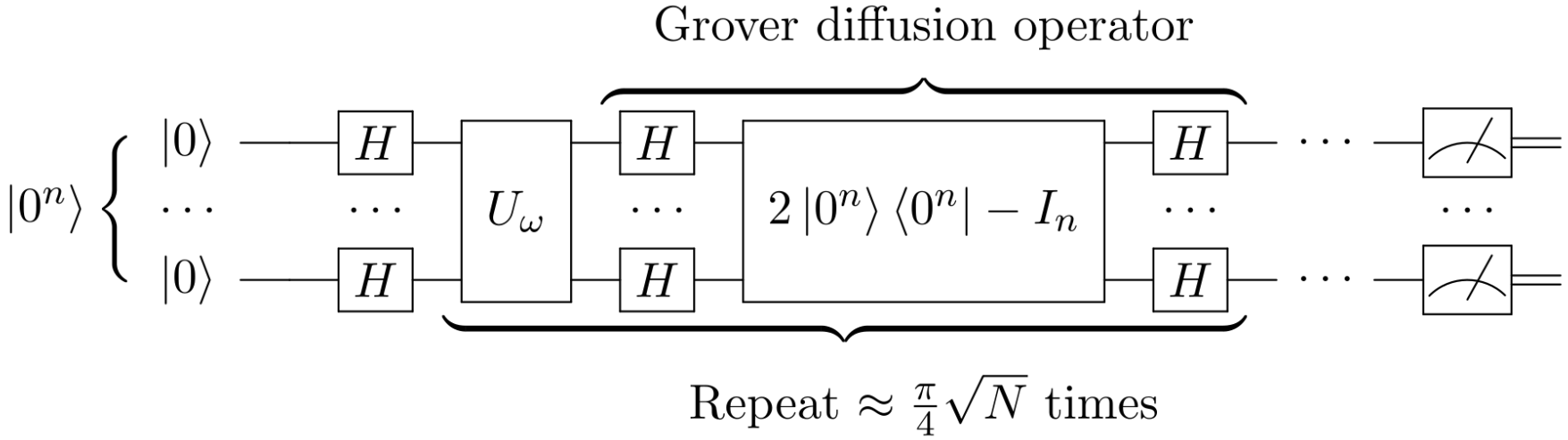
Security of ECC

- The difficulty of the discrete logarithm problem in the context of elliptic curves is the basis for ECC's security.
- No sub-exponential time algorithm is known for solving this problem on elliptic curves, making ECC very secure for a given key size.
- ECC with a 256-bit key provides comparable security to RSA with a 3072-bit key. **And this point is where quantum computer can bit classical one.**

Overview

Grover's Algorithm

- Grover's Algorithm is a quantum algorithm for searching an unsorted database with N entries in $O(\sqrt{N})$ time.



Grover's Algorithm

1. Initialize the system to the uniform superposition over all states:
$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$
2. Perform the Grover iteration $r(N)$ times, where each iteration consists of:
   - Applying the oracle operator U_ω.
   - Applying the Grover diffusion operator $U_s = 2|s\rangle\langle s| - I$.
3. Measure the resulting state in the computational basis.
   - With the correct choice of r, the output will be $|\omega\rangle$ with high probability for $N \gg 1$.
   - The number of iterations $r(N)$ is upper bounded by $\left\lceil \frac{\pi}{4}\sqrt{N} \right\rceil$.
   - The gate complexity of Grover's Algorithm is $O(\log(N)r(N))$

Grover's Algorithm

Function Inversion

Grover's algorithm can invert functions efficiently on a quantum computer.

$$y = f(x)$$

Given y, it finds x such that the function f maps x to y.

Grover's Algorithm

- Provides quadratic speed-up for brute-force attacks on symmetric-key ciphers.
- Enhances the efficiency of Collision attacks and Pre-image attacks.
- Implications:
  1. Key sizes may need to be doubled to maintain security against quantum attacks.
  2. Symmetric algorithms remain secure with larger key sizes.

Grover's Algorithm

However, we already know that Pollard's rho algorithm is capable of collision attacks at a faster rate than Grover's algorithm.

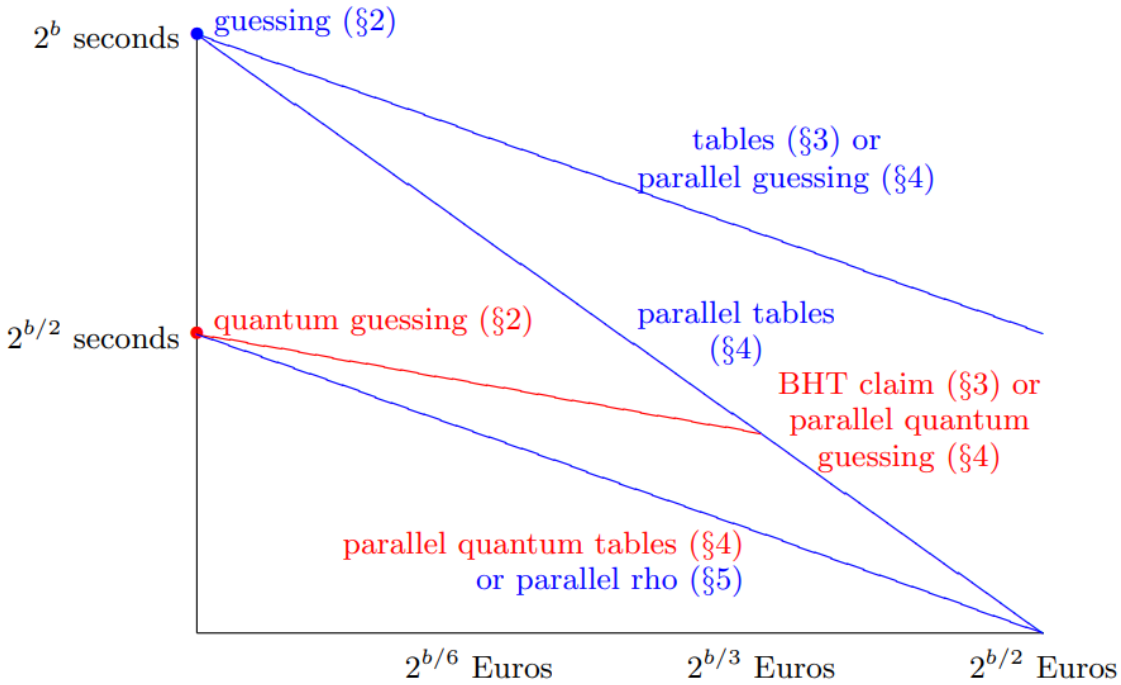
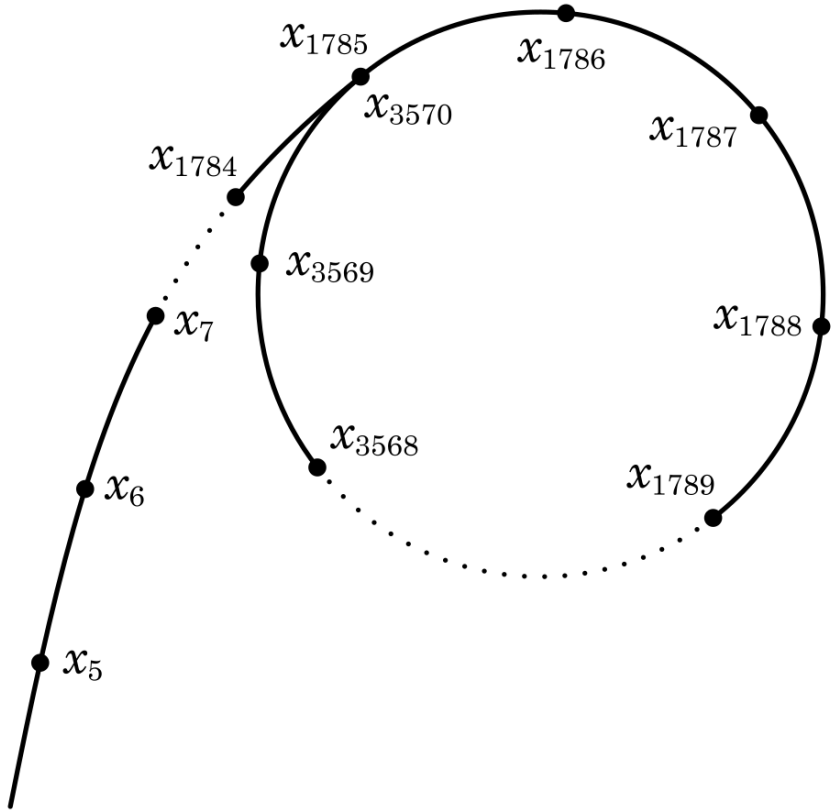


Fig. 1.2. Asymptotic collision-search time assuming free communication. Parallel rho is 1994 van Oorschot–Wiener [17]. "BHT claim" is 1998 Brassard–Høyer–Tapp [6]. Parallel quantum guessing is 2003 Grover–Rudolph [10].

Pollard rho Algorithm

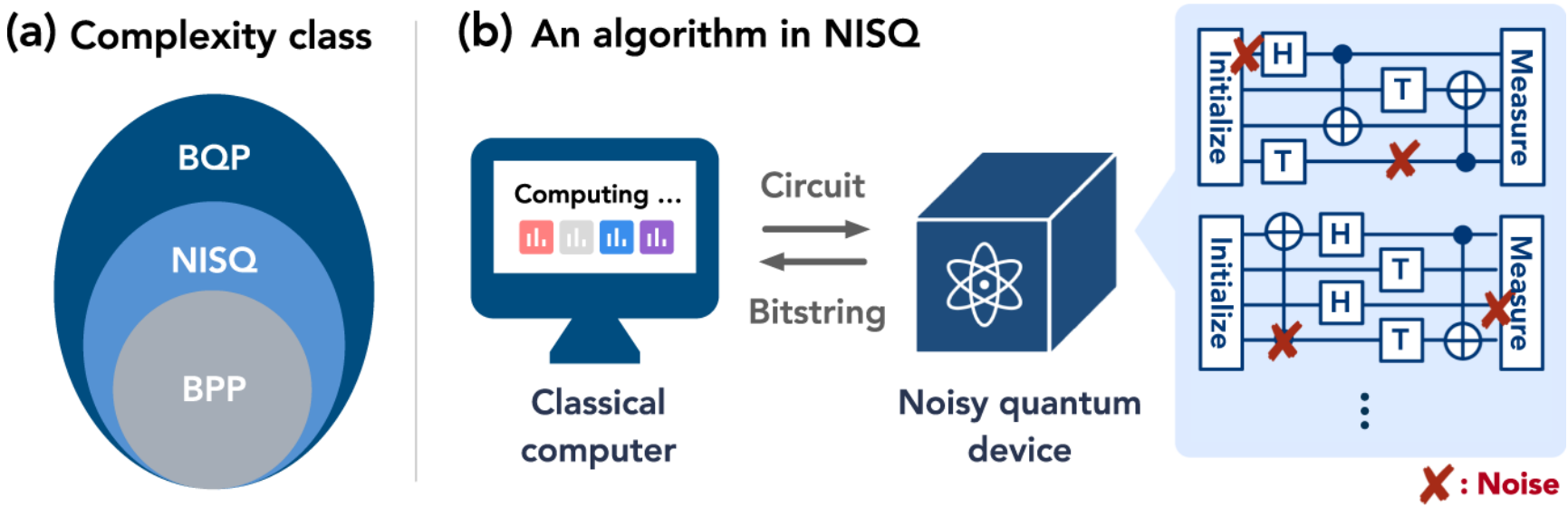


Optimality of Grover's Algorithm

Optimality

Bennett, Bernstein, Brassard, and Vazirani proved that any quantum solution to the search problem requires at least $\Omega(\sqrt{N})$ function evaluations, establishing that Grover's algorithm is asymptotically optimal.

Noisy intermediate-scale quantum era



BPP

BPP, bounded-error probabilistic polynomial time, is the class of decision problems solvable by a probabilistic Turing machine in polynomial time with an error probability bounded by 1/3 for all instances. A language L is in BPP if and only if there exists a probabilistic Turing Machine such that

- M runs for polynomial time on all inputs
- For all x in L, M outputs 1 with probability greater than or equal to 2/3
- For all x not in L, M outputs 1 with probability less than or equal to 1/3.

BQP

BQP

Let $A = (A_{yes}, A_{no})$ be a promise problem and let $a, b : \mathbf{N} \to [0, 1]$ be functions. Then $A \in BQP(a, b)$ if and only if there exists a polynomial-time generated family of quantum circuits $\mathbf{Q} = \{\mathbf{Q}_n : n \in \mathbf{N}\}$, where each circuit $\mathbf{Q}_n$ takes n input qubits and produces one output qubit, that satisfies the following properties.

- If $x \in A_{yes}$ then $Pr[\mathbf{Q}\ accepts\ x] \geq a(|x|)$
- If $x \in A_{no}$ then $Pr[\mathbf{Q}\ accepts\ x] \leq b(|x|)$

Also, the class BQP defined as BQP = BQP$(\frac{2}{3}, \frac{1}{3})$

NISQ

NISQ complexity class, informal

The NISQ complexity class contains all problems that can be solved by a polynomial-time probabilistic classical algorithm with access to a noisy quantum device. To solve a problem of size n, the classical algorithm can access a noisy quantum device capable of:

1. Preparing a noisy poly(n)-qubit all-zero state;
2. Applying arbitrarily many layers of noisy two-qubit gates;
3. Performing noisy computational basis measurements on all the qubits simultaneously.

All quantum operations are subject to a constant amount of depolarizing noise per qubit.

Noisy intermediate-scale quantum era

- In the current NISQ era, there is no quantum supremacy, not only for realistic problems like prime factorization, but also for simple Porter-Thomas Distribution computational problems.
- Until quantum computers show comparable performance to classical computers on problems that are not theoretically favorable, Grover's algorithm is unnecessary.

Overview

Shor's Algorithm for DLP

Periodic Function in DLP

Let P be a generator of a group $G = \langle P \rangle$ of prime order q and $Q = dP$ be an element of G. Then function $f : \mathbb{Z}_q \times \mathbb{Z}_q \to G$, defined by $f(x, y) = xP + yQ$, is periodic with period $(d, -1)$.

Quantum State for DLP

Shor's algorithm aims to find this period by generating the quantum state $|\psi\rangle$ with three registers, which can be represented as:

$$|\psi\rangle = \frac{1}{q} \sum_{x=0}^{q-1} \sum_{y=0}^{q-1} |x, y, f(x, y)\rangle$$

Shor's Algorithm for DLP

- The quantum state $|\psi\rangle$ is a superposition of all possible inputs to the function f and their corresponding outputs.
- By performing a quantum Fourier transform and measuring, we can obtain the period $(d, -1)$ which leads to the solution of the DLP.
- Finding the period efficiently is what gives Shor's algorithm its exponential speed-up over classical algorithms for DLP.

Shor's Algorithm for DLP

Quantum Superposition

The state $|\psi\rangle$ is a uniform superposition over all group elements generated by P and Q:

$$|\psi\rangle = \frac{1}{q}\sum_{x=0}^{q-1}\sum_{y=0}^{q-1}|x, y, xP + yQ\rangle$$

This can also be represented as a sum over cosets:

$$|\psi\rangle = \sum_{z'} c_{z'}|\phi_{z'}\rangle|z'P\rangle$$

Shor's Algorithm for DLP

Measurement and Collapse

Measuring the last register gives us a random group element zP. The first two registers collapse into a superposition of all x, y pairs such that $xP + yQ = (x + dy)P = zP$:

$$\frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} |z - dy \mod q, y\rangle$$

Each y has a corresponding x that solves $x = z - dy \mod q$.

- The superposition indicates the linear relationship between x and y with slope $-d$, the discrete logarithm we seek.
- Repeating this process and applying continued fractions to the measured values can reveal the value of d.

Shor's Algorithm for DLP

Application of QFT

After applying the Quantum Fourier Transform to the state:

$$\frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} |z - dy \mod q, y\rangle$$

We obtain:

$$\frac{1}{q^{3/2}} \sum_{x'=0}^{q-1} \sum_{y'=0}^{q-1} \sum_{y=0}^{q-1} e^{\frac{2\pi i}{q}((z-dy)x' + yy')} |x', y'\rangle$$

Shor's Algorithm for DLP

Interference Pattern and Condition

The coefficient of $|x', y'\rangle$ is significant if and only if $y' = dx' \mod q$:

$$\sum_{y=0}^{q-1} e^{\frac{2\pi i}{q}((z-dy)x'+yy')} = \begin{cases} qe^{\frac{2\pi i}{q}zx'} & \text{if } y' = dx' \mod q \\ 0 & \text{otherwise} \end{cases}$$

Shor's Algorithm for DLP

- The QFT amplifies the frequency components that correspond to the solution of the DLP.
- Measuring $|x', y'\rangle$ now has a high probability of yielding values related by $y' = dx' \mod q$.
- This relationship can be used to find the secret multiplier d by observing the output and applying the Extended Euclidean Algorithm.

Shor's Algorithm for DLP

Final Measurement and Finding d

Upon measuring the first two registers after applying the QFT, we obtain x' and y' such that:

$$y' = dx'$$

The discrete logarithm d can then be found as:

$$d = y' \cdot (x')^{-1} \mod q$$

- Measurement outputs x' and y' are such that y' is congruent to d times x' modulo q.
- We calculate the modular inverse of x' to solve for the private key d.
- This effectively "breaks" the DLP.

Shor's Algorithm for ECDLP

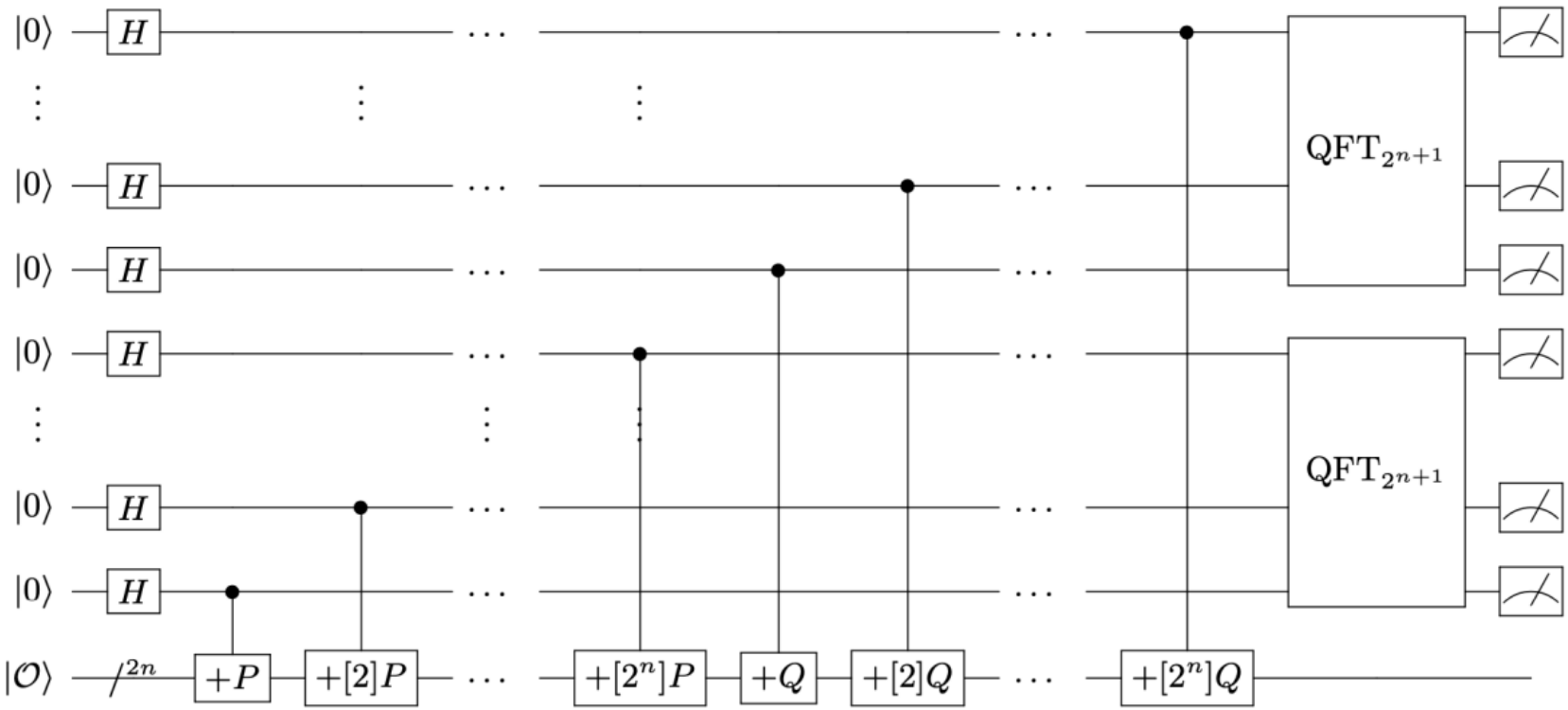
Generating $|\psi\rangle$

We can generate the state $|\psi\rangle = \sum_{x,y=0}^{N} |x, y, xP + yQ\rangle$ using the following quantum circuit steps:

1. Apply Hadamard gates to the first and second registers to create a superposition of all possible x and y.
2. Apply controlled-$U_{2^{n-k}P}$ operations to the third register, using the k-th qubit of the first register as the control qubit.
3. Repeat a similar process for the second register, using its qubits as control bits for controlled-$U_{2^{n-k}Q}$ operations on the third register.

Being able to create quantum states means we can solve this problem with QFT.

Shor's Algorithm for ECDLP



Shor's Algorithm for ECDSA vs RSA

ECDLP in $E(\mathbb{F}_p)$ simulation results					Factoring of RSA modulus N interpolation from [21]		
$\lceil \log_2(p) \rceil$ bits	#Qubits	#Toffoli gates	Toffoli depth	Sim time sec	$\lceil \log_2(N) \rceil$ bits	#Qubits	#Toffoli gates
110	1014	$9.44 \cdot 10^9$	$8.66 \cdot 10^9$	273	512	1026	$6.41 \cdot 10^{10}$
160	1466	$2.97 \cdot 10^{10}$	$2.73 \cdot 10^{10}$	711	1024	2050	$5.81 \cdot 10^{11}$
192	1754	$5.30 \cdot 10^{10}$	$4.86 \cdot 10^{10}$	1 149	–	–	–
224	2042	$8.43 \cdot 10^{10}$	$7.73 \cdot 10^{10}$	1 881	2048	4098	$5.20 \cdot 10^{12}$
256	2330	$1.26 \cdot 10^{11}$	$1.16 \cdot 10^{11}$	3 848	3072	6146	$1.86 \cdot 10^{13}$
384	3484	$4.52 \cdot 10^{11}$	$4.15 \cdot 10^{11}$	17 003	7680	15362	$3.30 \cdot 10^{14}$
521	4719	$1.14 \cdot 10^{12}$	$1.05 \cdot 10^{12}$	42 888	15360	30722	$2.87 \cdot 10^{15}$

Overview

Simon's Problem

Simon's Proble

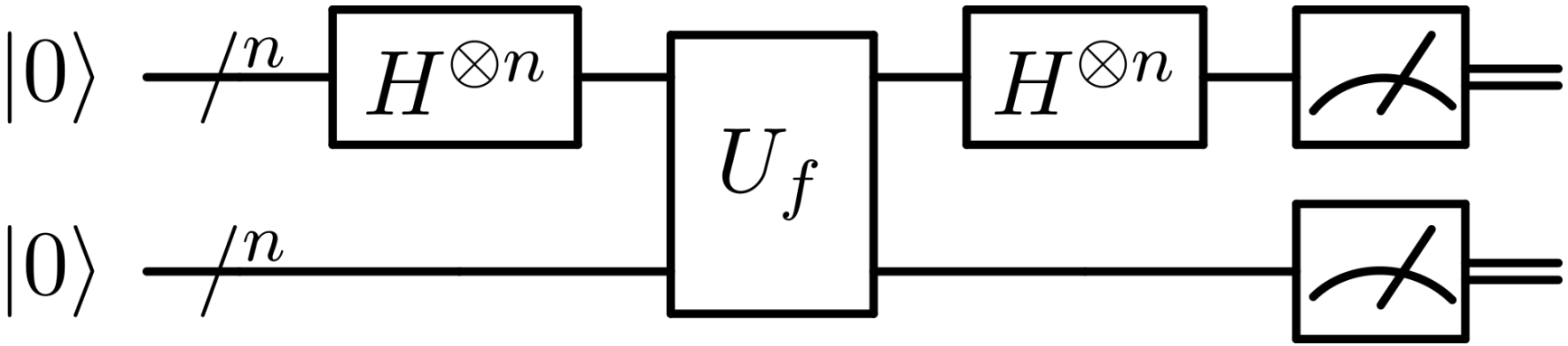
Given a black-box function $f : \{0,1\}^n \to \{0,1\}^n$ with the promise that there exists an unknown $s \in \{0,1\}^n$ such that for all $x, y \in \{0,1\}^n$,

$$f(x) = f(y) \quad \text{iff} \quad x \oplus y \in \{0^n, s\}$$

where $\oplus$ denotes bitwise XOR.

The goal is to identify s with as few queries to $f(x)$ as possible.

Simon's Algorithm



Simon's Algorithm

Linear Independence in Simon's Algorithm

We run the quantum part of the algorithm until we have a linearly independent set of bitstrings $\{y_1, \ldots, y_{n-1}\}$, where each y_k satisfies $y_k \cdot s = 0$. We can then classically solve this system of equations to find the hidden bitstring s.

Probability of Linear Independence

The probability that $y_1, y_2, \ldots, y_{n-1}$ are linearly independent is at least

$$\prod_{k=1}^{\infty} \left(1 - \frac{1}{2^k}\right) \approx 0.288788\ldots$$

Simon's Algorithm

Verification and Success Probability

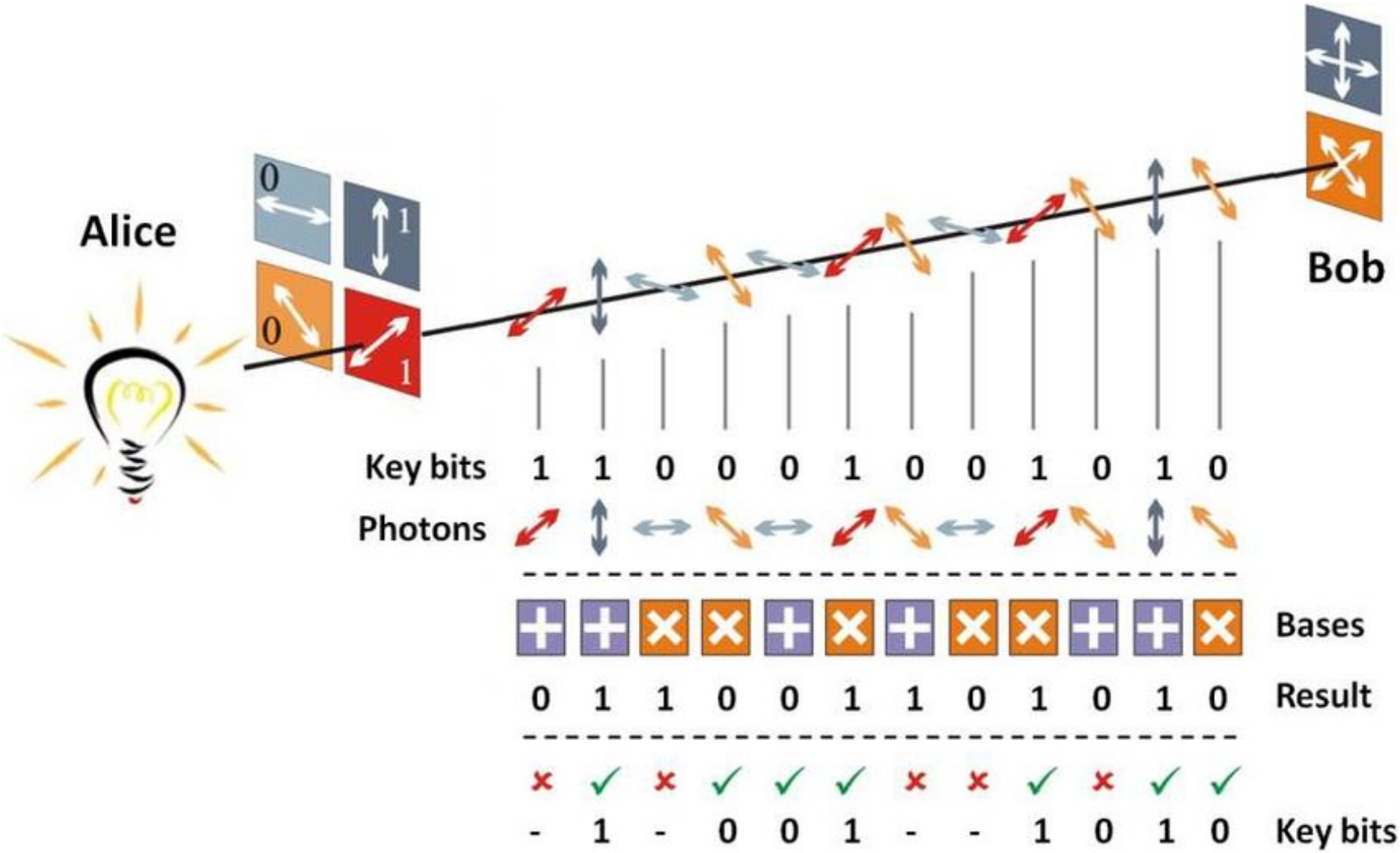
Once we obtain a candidate solution s', we verify if $f(0^n) = f(s')$. If they match, $s' = s$; otherwise, s may be 0^n. By repeating the algorithm a constant number of times, we can arbitrarily increase the success probability.

Efficiency of Simon's Algorithm

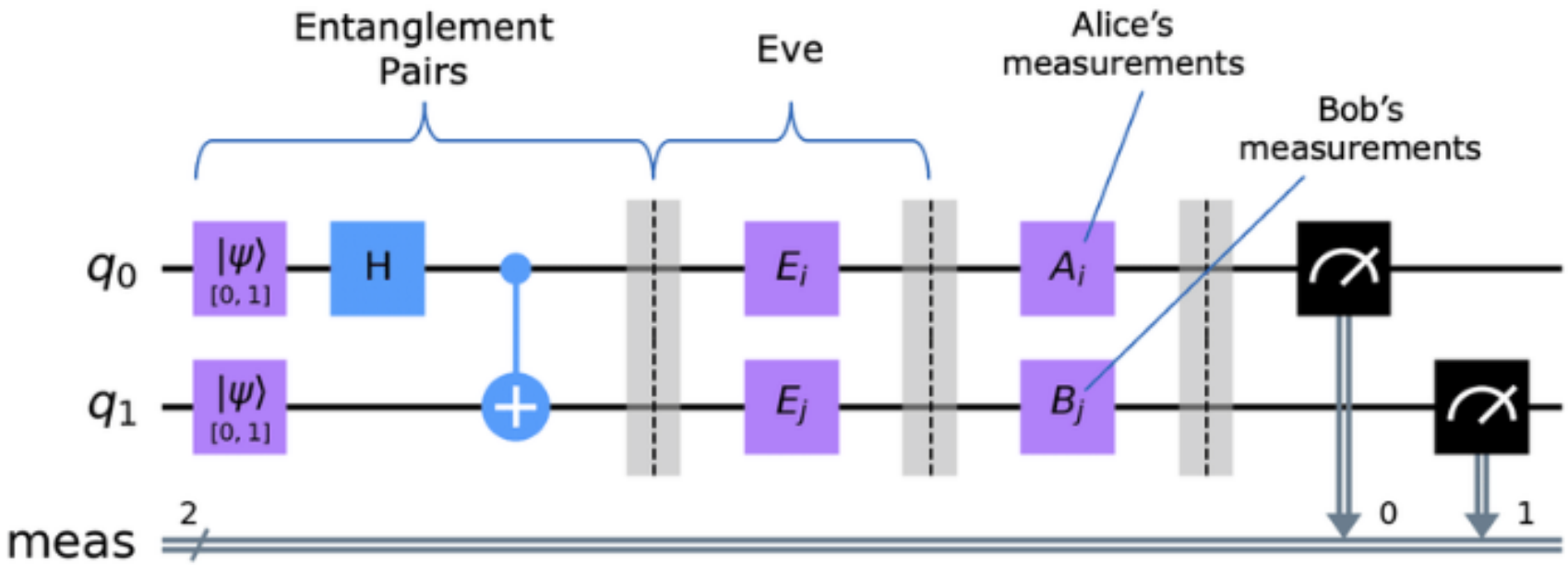
Simon's algorithm can identify the hidden bitstring with an exponentially lower number of queries compared to any classical algorithm, showcasing the potential of quantum computing.

Overview

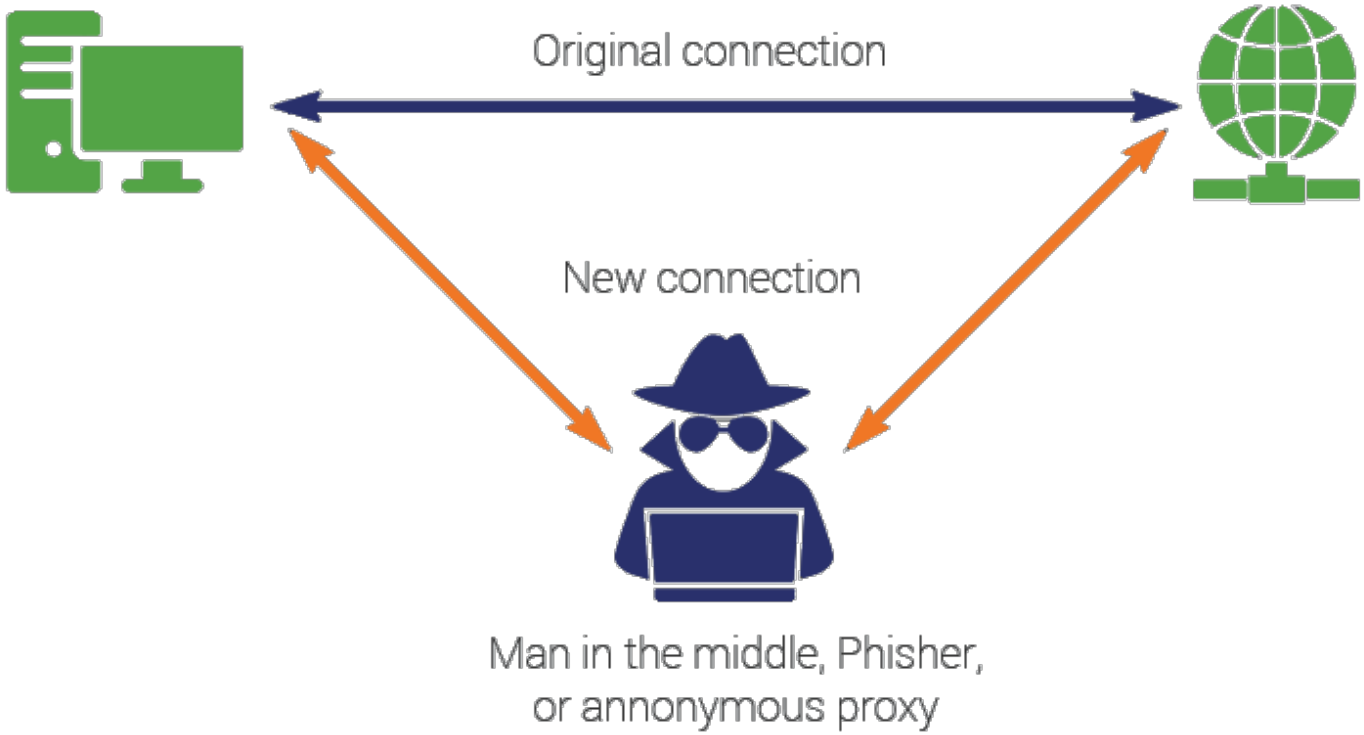
BB84 Protocol



E91 Protocol



Man in the Middle Attack



Overview

Aaronson's Theorem

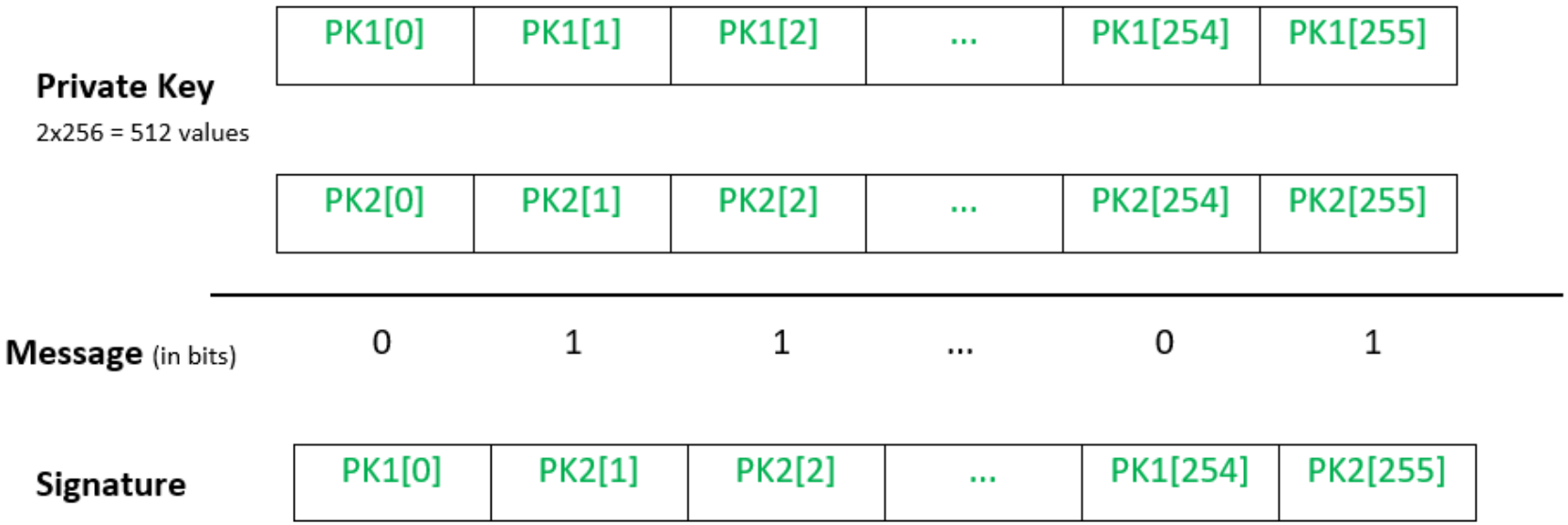
Hash Collision Lower Bound

Given a sequence $X = x_1, \ldots, x_n$ of n integers with n even, drawn from the set $\{1, \ldots, n\}$, the collision problem, denoted as Col_n, asks us to determine whether:

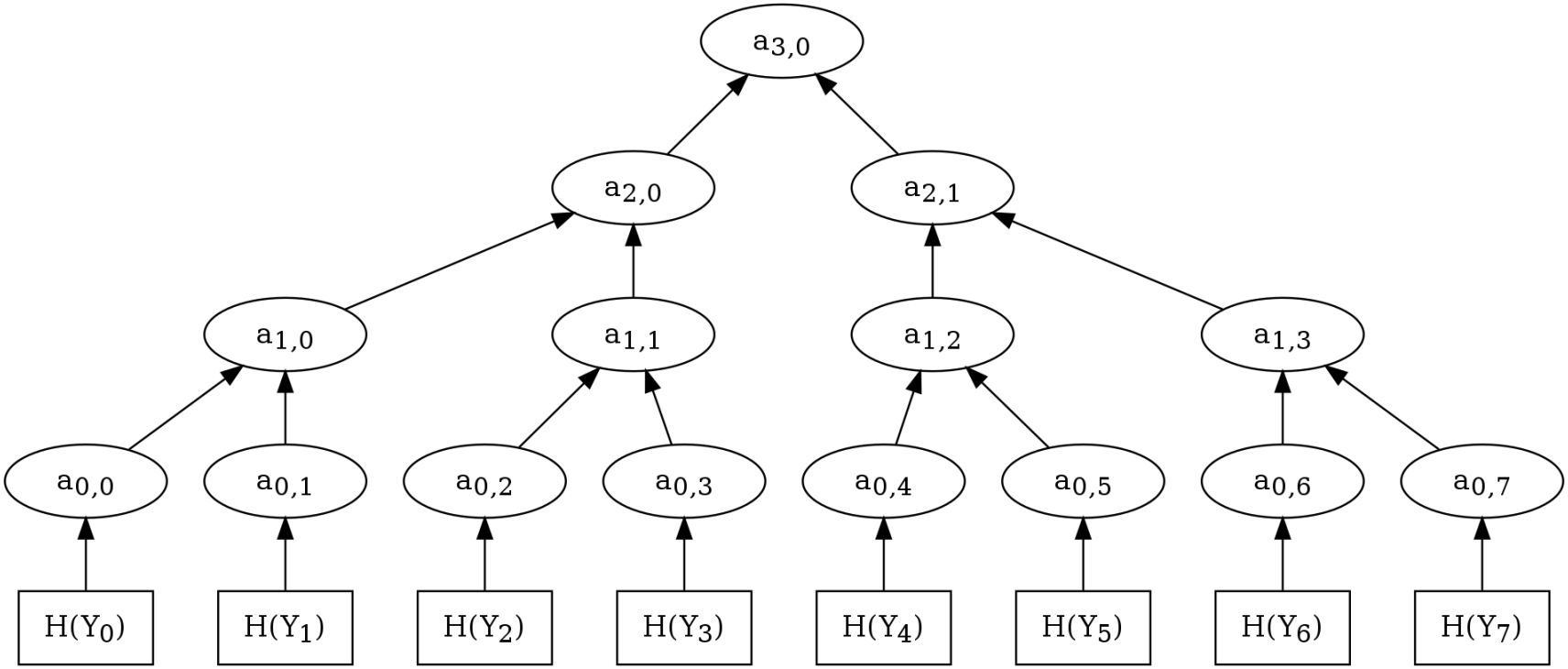
1. X is a one-to-one function (a permutation of $\{1, \ldots, n\}$), or
2. X is a two-to-one function (each element of $\{1, \ldots, n\}$ appears exactly twice or not at all in X).

The bounded-error quantum query complexity of the collision problem, $Q_2(\text{Col}_n)$, is $\Omega(n^{1/5})$, where Q_2 is bounded-error quantum query complexity.

One Time Pad : Lamport's Signautre



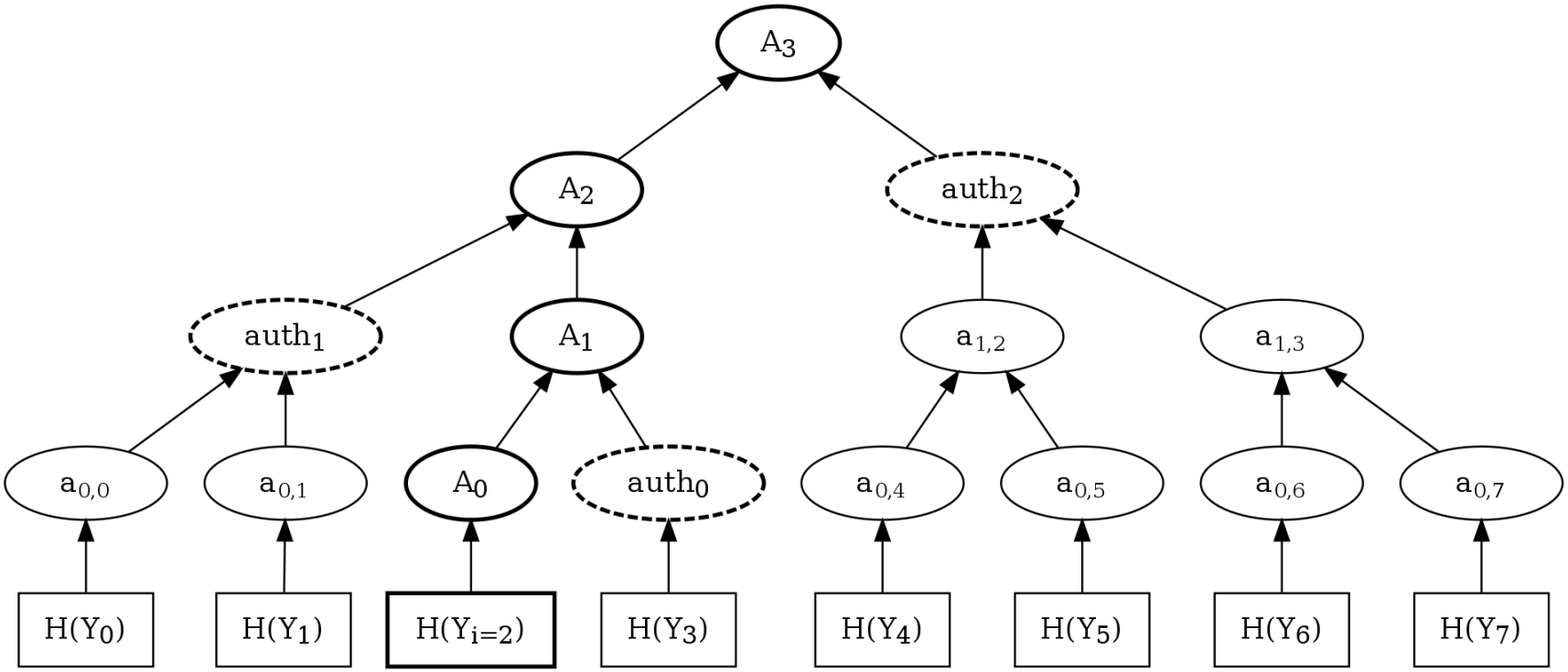
Merkle Signature Scheme : Key Generation



Merkle Signature Scheme : Key Generation

- Generate 2^n private/public key pairs (X_i, Y_i) for $N = 2^n$ messages.
- Employ a one-time signature scheme such as Lamport for each key pair.
- Compute hash $h_i = H(Y_i)$ for each public key Y_i.
- Construct a Merkle tree with these 2^n hash values as leaves.
- Define nodes at height i and position j as $a_{i,j}$.
- The tree root, $a_{n,0}$, serves as the public key 'pub'.
- Note: Private key size is proportional to the number of messages.

Merkle Signature Scheme : Signature Generation



Merkle Signature Scheme : Signature Generation

- Select an unused key pair (X_i, Y_i) for message M.
- Use the one-time signature scheme to sign M, producing sig' and Y_i.
- Include Merkle tree nodes to link h_i to the public key 'pub'.
- Authentication path: $(A_0, \ldots, A_n)$ from leaf to root.
- Authentication nodes $auth_i$ are provided for verification.
- Complete signature sig comprises sig', Y_i, and $auth_0, \ldots, auth_{n-1}$.

Merkle Signature Scheme : Signature Verification

- Verify the one-time signature sig' with public key Y_i.
- On sig' validation, hash Y_i to compute A_0.
- Sequentially compute nodes A_j using authentication nodes $auth_{j-1}$.
- If final node A_n matches the public key 'pub', the signature is valid.

Overview

Multivariate Cryptography

Multivariate cryptography is based on the challenging mathematical problem of solving a system of multivariate polynomials. Specifically, multivariate cryptosystems leverage the multivariate quadratic map.

Definition (Multivariate Quadratic Map)

A multivariate quadratic map P takes a sequence $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ and maps it to an output $y = (p_1(x), \ldots, p_m(x)) \in \mathbb{F}_q^m$, where each $p_i(x)$ is a multivariate quadratic polynomial, and the coefficients of the polynomials are in $\mathbb{F}_q$.

The MQ Problem

The MQ Problem is foundational for multivariate cryptography and is considered hard for quantum computers.

MQ Problem

Given a multivariate quadratic map $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and a target $t \in \mathbb{F}_q^m$, find a value s such that $P(s) = t$. Note that s is not unique since map P is not injective.

Methods like Gröbner basis, and algorithms such as F4/F5 and XL, are used to tackle the MQ Problem.

Oil and Vinegar Scheme

The Oil and Vinegar Scheme is a signature scheme based on the MQ problem, involving a mix of "oil" and "vinegar" variables in the polynomials.

Scheme Overview

- A random quadratic map $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is selected, with $n > m$.
- There are m oil variables and $n - m$ vinegar variables.
- Polynomials include terms with vinegar-vinegar and oil-vinegar variables, but no oil-oil terms.

The signature process for a document d involves computing a hash $y = H(d)$, solving for s' such that $P(s') = T^{-1}(y)$, and sharing $s = P^{-1}(s')$ as the signature.

Unbalanced Oil and Vinegar and Rainbow Schemes

- The Unbalanced Oil and Vinegar (UOV) scheme modifies the number of vinegar variables to increase security.
- The Rainbow digital signature scheme incorporates multiple layers of UOV to strengthen against MQ Problem attacks.

MinRank Problem

Given matrices M_i for $i = 1, 2, \ldots, k$ with n rows and m columns, and a target rank r, find a coefficient vector y such that the rank of the linear combination of M_i with y is at most r.

However, the complexity of the Rainbow scheme introduces susceptibility to the MinRank Problem, constraining the optimized design to 2 layers in the NIST competition.

Security Concerns and Recent Developments

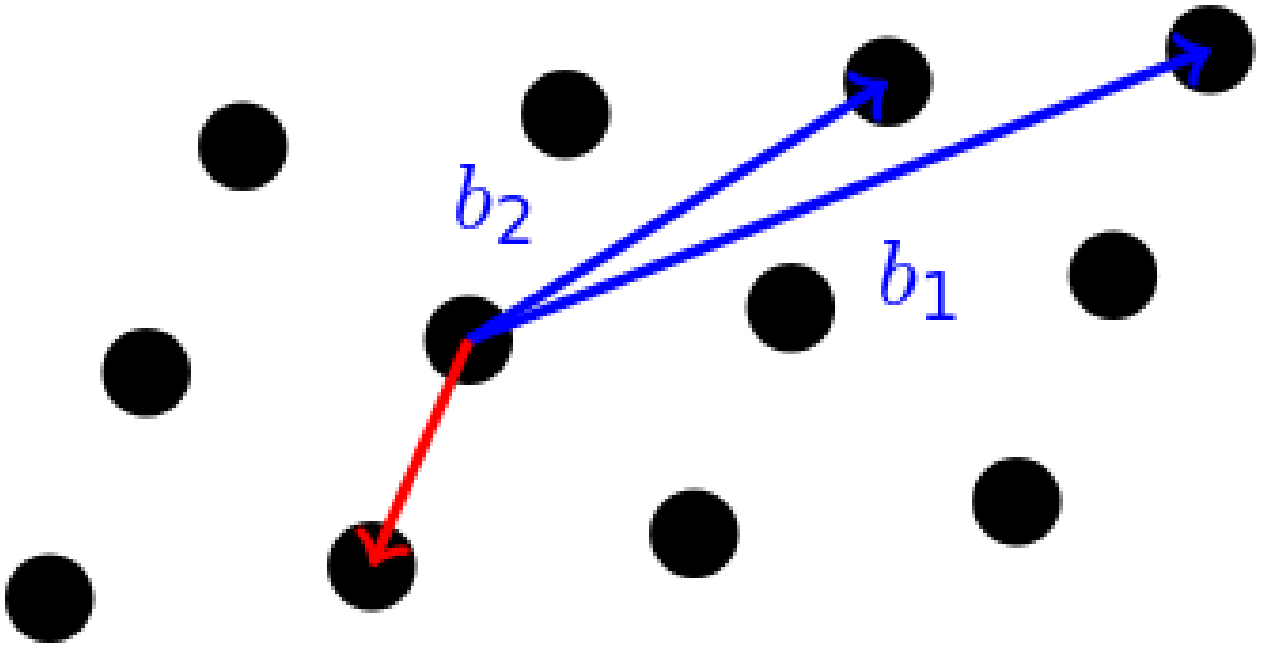
While the UOV and Rainbow schemes are robust against traditional attacks, they face new challenges:

- The balance of oil and vinegar variables is crucial for security.
- Recent intersection attacks have reduced the security of the Rainbow scheme by a few bits.

Proper parameter selection for n and m is critical in minimizing vulnerabilities to these attacks.

Overview

The Shortest Vector Problem (SVP)



The Shortest Vector Problem (SVP)

The Shortest Vector Problem (SVP) in a lattice is defined as finding the shortest nonzero vector in the lattice. This problem can be formally stated as follows:
Given a basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$ of a lattice L, find a nonzero vector $\mathbf{v} \in L$ such that:

$$||\mathbf{v}|| = \min_{\mathbf{0} \neq \mathbf{x} \in L} ||\mathbf{x}||$$

where $|| \cdot ||$ denotes the Euclidean norm, and L is the set of all integer linear combinations of the basis vectors:

The Shortest Vector Problem (SVP)

$$L = \left\{ \sum_{i=1}^{n} z_i \mathbf{b}_i \mid z_i \in \mathbb{Z} \right\}$$

- SVP is known to be NP-hard under randomized reductions.
- It is the basis for the security of many lattice-based cryptosystems.
- Approximation versions of SVP, where the goal is to find a vector within a factor γ of the shortest vector, are also widely used in cryptography.

Multivariate Lattices

Multivariate polynomials form the basis of multivariate lattice problems, which are represented as:

- A system of equations $F(x_1, x_2, ..., x_n) = 0$, where F is a set of multivariate polynomials over a finite field $\mathbb{F}_q$.

The security of multivariate lattice-based schemes often relies on the problem of solving systems of nonlinear equations, which is NP-hard:

$$F(\mathbf{x}) = \mathbf{y}$$

where $F : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is a system of m multivariate polynomials in n variables, and $\mathbf{y}$ is given.

Learning With Errors (LWE)

The Learning With Errors problem is described by the following equation:

$$\mathbf{b} = A\mathbf{s} + \mathbf{e} \mod q$$

where:

- A is a given uniformly random $m \times n$ matrix over $\mathbb{Z}_q$.
- $\mathbf{s}$ is a secret vector in $\mathbb{Z}_q^n$.
- $\mathbf{e}$ is a vector of errors, where each component is drawn from a small noise distribution.
- $\mathbf{b}$ is the observed vector in $\mathbb{Z}_q^m$.

The goal is to recover $\mathbf{s}$ given $(A, \mathbf{b})$, which is computationally hard.

Ring Variants

Ring variants of lattice problems use ring structures to improve efficiency. For example, Ring-LWE is based on the following problem:

- Given pairs $(\mathbf{a}_i, \mathbf{b}_i)$ where $\mathbf{b}_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \mod q$ for $i = 1, \ldots, m$.
- $\mathbf{a}_i \in R_q$ are chosen uniformly at random.
- $\mathbf{s} \in R_q$ is a secret element.
- e_i is a small error term from a noise distribution.

One must solve for $\mathbf{s}$, which is difficult due to the ring structure and the noise:

$$R_q = \frac{\mathbb{Z}[x]}{(f(x), q)}$$

where $f(x)$ is a monic polynomial that defines the ring and q is a modulus.