Distributed Simon's Algorithm with Less Per-Node Qubit Overhead and Its Application to Cryptanalysis

Zhenqiang Li[1], Xiaofan Zhen[2], Shuqin Fan[1⋆], Yonglin Hao[1], Fei Gao[2]

[1] State Key Laboratory of Cryptology, Beijing 100878, China
fansq@sklc.org
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract. Distributed quantum computing (DQC) enables multi-device collaboration to reduce per-node circuit depth and solve larger-scale problems beyond the processing capability of a single quantum device. In 2022, Tan et al. proposed a distributed Simon's algorithm via a concatenation-type periodic function. In comparison with the standard version, the distributed Simon's algorithm has a lower per-node quantum query complexity resulting in not only a reduced per-node circuit depth but a higher per-node qubit requirement as well. This paper proposes a new distributed Simon's algorithm by constructing an XOR-type periodic function, which can reduce both the per-node quantum query complexity and the per-node qubit requirement. Specifically, the per-node quantum query complexity is reduced to $2c(n-t)$ $(c>3)$, matching that of Tan et al.'s scheme; furthermore, the per-node qubit requirement is diminished significantly from Tan et al.'s $2^{t+1}m$ to $m+n-t$, which is an exponential reduction with respect to t. Here, n and m respectively denote the input and output lengths of the periodic function while t is an integer satisfying $n/2 < t < n$. Given the scale limitations of current quantum hardware, our distributed algorithm makes it feasible to tackle larger-scale problems that cannot be solved on a single quantum device. Based on this new algorithm, we propose distributed quantum key-recovery attacks on the SoEM22 construction. Compared with state-of-the-art non-distributed quantum attacks based on the standard Simon's algorithm, our attack requires notably lower per-node qubit overhead, while retaining comparable time complexity in both the classical and quantum query models.

Keywords: Distributed quantum computing, Simon's algorithm, per-node qubits, SoEM construction, quantum attack.

1 Introduction

Quantum computing has demonstrated remarkable speedups over classical computing in solving specific problems, such as large integer factorization [20], linear systems solving [10, 17, 28], dimensionality reduction [18, 19, 33], and others [16, 22, 23, 29]. However, limited by contemporary physical devices, long-depth

quantum circuits suffer from severe noise, and the number of available qubits is highly limited. As a result, large-scale quantum computation remains unattainable. Whether it is possible to accomplish larger quantum tasks with limited qubits is a critical research direction.

Distributed quantum computing (DQC) has emerged as a promising paradigm, enabling collaboration across multiple quantum nodes [1]. Unlike single-device quantum computing, DQC decomposes large-scale tasks into smaller sub-tasks to achieve two core advantages for overcoming the limitations of single quantum devices: reducing the number of qubits per node and decreasing per-node circuit depth. The significance of DQC is that even if the total number of qubits increases, reducing the per-node qubit requirement and shortening the per-node circuit depth can each turn impossible tasks for single quantum devices into feasible ones. For instance, a task requiring 100 qubits can be accomplished using multiple 80-qubit quantum processors connected via quantum communication; a task requiring circuit depth 2^{20} can be accomplished through collaboration among multiple quantum processors with circuit depth 2^{19}.

In recent years, some efforts have been devoted to developing distributed quantum algorithms, including distributed Shor's algorithms [11, 30, 32], distributed Grover's algorithm [36] etc. Simon's algorithm [21], a quantum algorithm for period finding, has attracted particular attention since it has been widely applied in the cryptanalysis of numerous symmetric ciphers, including 3-round Feistel [14], Even-Mansour constructions [15], CBC-MAC [12], and PMAC variants [24] etc.

In 2022, Tan et al. [26] proposed a distributed Simon's algorithm by constructing a concatenation-type periodic function derived from the target periodic function. It achieves a lower per-node quantum query complexity than the standard Simon's algorithm, meaning that the per-node circuit depth can be reduced. However, the implementation of the concatenation-type function incurs significantly higher per-node qubit overhead, which poses severe challenges to practical hardware implementation. Tan et al.'s scheme reveals a trade-off between the per-node qubit overhead and the per-node circuit depth. Thus, exploring a more favorable trade-off represents an interesting and meaningful research direction.

Contributions. Considering that the number of available qubits in quantum computing devices is limited, we propose a distributed Simon's algorithm that reduces the per-node qubit requirement and apply it to quantum attacks against SoEM cipher constructions. Our contributions are as follows:

– We propose a distributed Simon's algorithm by constructing an XOR-type periodic function, which can simultaneously achieve a reduction in per-node qubit requirement and quantum query complexity. Compared with the standard Simon's algorithm, our scheme reduces the per-node qubit requirement and quantum query complexity from $m+n$ and cn to $m+n-t$ and $2c(n-t)$, respectively. Compared with Tan et al.'s work [26] requiring $2^{t+1}m$ per-node qubits, our scheme achieves an exponential reduction with respect to t, while maintaining the same per-node quantum query complexity. Here, n and m

respectively denote the input and output lengths of the periodic function, and t is an integer satisfying $n/2 < t < n$.
– We formalize the distributed offline Simon's problem and present a quantum algorithm for its special case with (multi-)XOR-type conditional periodic functions, enabling distributed key-recovery attacks on the SoEM22 construction. Compared with the existing non-distributed quantum attack [25], our method exhibits notably lower per-node qubit overhead with comparable time complexity in both the classical and quantum query models.

Outline. The remainder of this paper is structured as follows. In Section 2, we briefly review relevant preliminaries. Section 3 proposes our distributed Simon's algorithm. Section 4 introduces the distributed offline Simon's problem, and presents a quantum algorithm for its special case with (multi-)XOR-type conditional periodic functions. Section 5 demonstrates the application of our proposed distributed algorithms in attacking the SoEM22 construction. Finally, Section 6 presents a succinct conclusion.

2 Preliminaries

2.1 Notations

Let $\{0,1\}^n$ be the set of all n-bit strings, and $\mathrm{Perm}(n)$ the set of all permutations on $\{0,1\}^n$. For two bit strings a and b, $a\|b$ represents their concatenation. When a and b have the same length, their bitwise XOR is $a \oplus b$, and their inner product over $\mathbb{F}_2$ is $a \cdot b$. Besides, let 0^u and 1^u denote u consecutive zeros and ones, respectively. $\tilde{O}$ denotes the soft asymptotic notation, hiding lower-order terms to focus on the dominant complexity order. I_n is the identity operator acting on the n-qubit quantum system.

Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function implemented by a quantum circuit $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, where $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$. In this paper, D_f, W_f and G_f denote the depth, number of auxiliary qubits, and number of gates for implementing f.

For quantum cryptanalysis on symmetric cryptosystems, two quantum attack models Q1 and Q2, are defined based on attacker capabilities [34]:

– Q1 model: the classical query model, where attackers execute any offline quantum computation and only online queries in a classical manner to an encryption oracle;
– Q2 model: the quantum query model, where attackers execute any offline quantum computation and superposition queries to an encryption oracle.

The Q2 model relies on the strong assumption of online superposition queries, rendering its practical impact unclear. In contrast, quantum attacks in the Q1 model are more practical: once a large-scale fault-tolerant quantum computer becomes available, Q1 attacks pose real threats to symmetric-key primitives.

2.2 Some Typical Quantum Algorithms

This part reviews the quantum algorithms mentioned in the paper.
Grover's Algorithm [8], also known as Grover search, is designed to solve the following Problem 1.

Problem 1. **(Grover's problem)** Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Given oracle access to f, find a good element x such that $f(x) = 1$.

Let M, $1 \leq M \leq 2^n$, be the number of solutions. Grover's algorithm requires $O(\sqrt{2^n/M})$ quantum queries to f with probability at least $1 - \frac{M}{2^n}$. In particular, if $M = 1$, the number of quantum queries is $O(2^{n/2})$, and the number of qubits is $n + 1$. The formal description of Grover's algorithm is shown in Algorithm 1.

Algorithm 1 Grover's algorithm [8].

Require: $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, where $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function and $y \in \{0,1\}$.
Ensure: x such that $f(x) = 1$.
1: Initialize a quantum state $|0^n\rangle|1\rangle$.
2: Apply $(n+1)$-fold Hadamard gate $H^{\otimes n+1}$ to obtain:

$$H^{\otimes n+1}|0^n\rangle|1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|-\rangle = |\varphi\rangle|-\rangle.$$

Here, $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
3: Perform Grover iteration operation $R \approx \frac{\pi}{4}\sqrt{2^n/M}$ times:

$$[(H^{\otimes n}(2|0^n\rangle\langle 0^n| - I_n)H^{\otimes n} \otimes I_1)O_f]^R|\varphi\rangle|-\rangle.$$

4: Measure x.

Simon's Algorithm [21] is designed to solve the following Problem 2.

Problem 2. **(Simon's problem)** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a Boolean function ($m \geq n-1$). Given oracle access to f, find the non-trivial period $s \neq 0$ such that $f(x) = f(x \oplus s)$ for any x.

From Algorithm 2, it can be seen that each run of Simon's routine outputs a vector y satisfying $y \cdot s = 0$. Simon's algorithm consists of c Simon's routines. After obtaining c such vectors, we can determine s in $O(n^3)$ time by using the classical Gaussian elimination method. It has been proved in [12] that, when $c > 3n$, the success probability of recovering the correct s becomes exponentially close to 1. Thus, Simon's algorithm requires c quantum queries to f and $n + m$ qubits.
Offline Simon's Algorithm [3] combines Simon's algorithm and Grover's algorithm to solve the following Problem 3.

Problem 3. **(Offline Simon's problem)** Let $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ and $g : \{0,1\}^n \to \{0,1\}^m$ be two functions, where F is interpreted as a family of functions indexed by $\{0,1\}^\kappa$. For each index k, define $F(k,\cdot) = f_k(\cdot)$. Suppose quantum oracle access to F and g are provided, with the following conditions:

- There exists a unique $k_0 \in \{0,1\}^\kappa$ such that $f_{k_0}(x) \oplus g(x)$ has a period $s \in \{0,1\}^n \backslash \{0^n\}$.
- For all $k \in \{0,1\}^\kappa \backslash \{k_0\}$ and $t \in \{0,1\}^n \backslash \{0^n\}$, the maximum probability over x such that $(f_k \oplus g)(x \oplus t) = (f_k \oplus g)(x)$ satisfies:

$$\max_{\substack{k \in \{0,1\}^\kappa \backslash \{k_0\} \\ t \in \{0,1\}^n \backslash \{0^n\}}} \Pr_{x \leftarrow \{0,1\}^n} [(f_k \oplus g)(x \oplus t) = (f_k \oplus g)(x)] \leq \frac{1}{2}. \tag{1}$$

The goal is to find k_0 and s.

In cryptographic applications, $g(x)$ is a key-dependent function in general evaluated via online queries, while $f_k(x)$ is a key-independent function computable offline.

Algorithm 2 Simon's routine [21]

Require: $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$ with function $f : \{0,1\}^n \to \{0,1\}^m$ and $y \in \{0,1\}^m$.

Ensure: y with $y \cdot s = 0$.

1: Initialize a quantum state $|0\rangle^{\otimes n}|0\rangle^{\otimes m}$.

2: Apply the quantum gate $H^{\otimes n} \otimes I_m$ to obtain:

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|0\rangle.$$

3: Perform O_f to obtain:

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle.$$

4: Reapply the quantum gate $H^{\otimes n} \otimes I_m$ to get:

$$\frac{1}{2^n} \sum_{x,y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle|f(x)\rangle =$$
$$\frac{1}{2^n} \sum_{\substack{x \in \{0,1\}^n / \langle s \rangle, \\ y \in \{0,1\}^n}} (-1)^{x \cdot y} (1 + (-1)^{s \cdot y}) |y\rangle|f(x)\rangle.$$

5: Measure y.

The offline Simon's algorithm consists of an online phase and offline computation, iteratively reusing the quantum state $|\psi_g\rangle = \bigotimes_{j=1}^{c} \sum_{x_j \in \{0,1\}^n} |x_j\rangle|g(x_j)\rangle$:

1. **Online phase**: prepare the quantum state $|\psi_g\rangle$;
2. **Offline phase**: perform Grover search over $k \in \{0,1\}^\kappa$. In each iteration:
   - For fixed k, apply $|\psi_g\rangle$ and $O_F : |k\rangle|x\rangle|y\rangle \to |k\rangle|x\rangle|y \oplus f_k(x)\rangle$ to compute $g(x) \oplus f_k(x)$;
   - Check if the rank of c parallel Simon routine outputs is at most $n-1$;
   - Uncompute to recover $|\psi_g\rangle$ for reuse.

The period s is recovered via Simon's algorithm. Besides, Bonnetain [2] established the following result:

Theorem 1. *Let τ be a positive integer, $m \geq \log_2(4e(n+\kappa+\tau+1))$ and $\kappa \geq 7$. The offline Simon's algorithm solves Problem 3 with $\frac{\pi}{4\arcsin\sqrt{2^{-\kappa}}}$ iterations ($c = n+\kappa+\tau+1$ queries per iteration). The probability that the success rate is less than*

$$1 - 2^{-\tau} - (2^{-\tau/2-1} + 2^{-\tau} + 2^{-\kappa/2+1})^2$$

is bounded above by $2^{n+\kappa-\frac{2^n}{4(n+\kappa+\tau+1)}}$.

Computing the rank of vector systems is a core step in both Simon's algorithm and offline Simon's algorithm. Bonnetain and Jaques [4] proposed a reversible quantum circuit for this task, with quantum resources as follows:

Lemma 1. *Given c n-bit vectors, the quantum circuit for computing the rank of their linear span has:*

- *Depth $D_{test}[c,n] = (c+n)\lg(n)$;*
- *Ancilla qubits $W_{test}[c,n] \geq c + \frac{3}{2}n^2 - \frac{n}{2}$;*
- *Total quantum gates $G_{test}[c,n] = cn^2 + cn$ (for $c > n$ and $c = \Theta(n)$).*

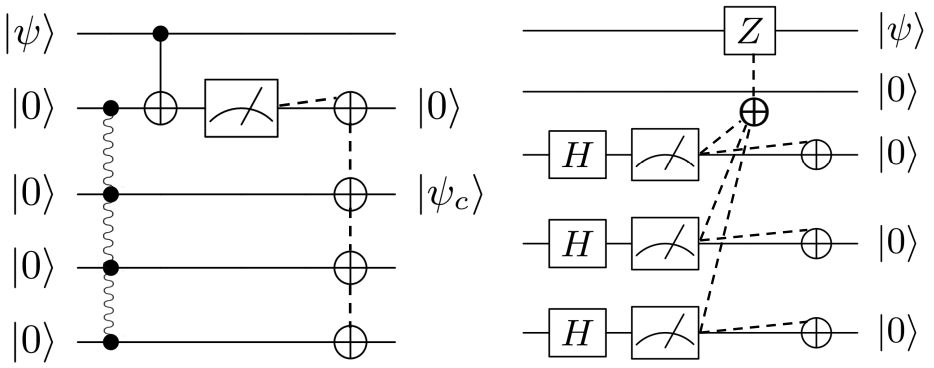
Theorem 2. *Suppose $\kappa = O(n)$ and $c = \kappa + n + \tau + 1$. Let $k_0 \in \{0,1\}^\kappa$ be the good element such that $f_{k_0} \oplus g$ is a periodic function, and assume that Eq. (1) holds. The offline Simon's algorithm recovers k_0 and s with success probability $\geq 99\%$ using $2c2^{\kappa/2}$ quantum queries to F:*

- *Q2 model: c online quantum queries to g;*
- *Q1 model: 2^n classical queries to g.*

The offline computation time (excluding $|\psi_g\rangle$ preparation) is $O((n^3 + nT_f)2^{\kappa/2})$, where T_f is the time to evaluate F once. The total qubit requirement is $\kappa + c(n+m) + W_{test}[c,n] + 1$.

2.3 Distributed Quantum Computing

Distributed quantum computing (DQC) is a paradigm that leverages a network of quantum nodes, each having limited qubit capacity, interconnected via classical and quantum channels. The core goal of DQC is to overcome the qubit scalability bottleneck of single quantum devices, enabling large-scale quantum computations beyond the qubit limit of individual nodes. Each node is equipped with two types of qubits:



(a) Cat-entangler (b) Cat-disentangler

Fig. 1: Quantum circuits for *cat-entangler* and *cat-disentangler* operations (5-qubit system). Dotted lines denote classical measurement results used to control NOT gates. (a) *Cat-entangler* circuit: $|\psi\rangle = (\alpha|0\rangle + \beta|1\rangle)_1$, wavy lines generate a GHZ state $\frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)_{2345}$, and $|\psi_c\rangle = (\alpha|0000\rangle + \beta|1111\rangle)_{1345}$. (b) *Cat-disentangler* circuit: The Z gate is controlled by the XOR of three measurement bits, i.e., performing Z gate if and only if the XOR is 1.

- **Register qubits**: Qubits that can freely interact with each other and with the channel qubits of the same node, serving as the local computing resource for node-specific operations.
- **Channel qubits**: Qubits dedicated to inter-node communication, which can be transmitted back and forth across the network to facilitate interactions between distributed nodes.

Interactions between qubits from different nodes can be achieved via two approaches: (1) non-local quantum operations; (2) qubit teleportation or physical transport to enable local interactions at remote nodes.

All quantum computations correspond to unitary operators on a Hilbert space and are thus reversible. Any unitary operation can be implemented with a universal gate set $\{\mathrm{CNOT}, \mathrm{H}, \mathrm{S}, \mathrm{NOT}, \mathrm{T}\}$. Single-qubit quantum gates $\mathrm{H}, \mathrm{S}, \mathrm{NOT}$ and T can be performed on register qubits within a node, while the two-qubit quantum gate CNOT between nodes is implemented using entanglement and quantum teleportation protocols. Thus, the distributed implementation of any unitary transformation can be reduced to the implementation of non-local CNOT gates.

To implement the non-local CNOT gate, *cat-entangler* and *cat-disentangler* operations serve as two primitive operations. For instance, the *cat-entangler* operation transforms a 5-qubit system from $(\alpha|0\rangle+\beta|1\rangle)_1|0\rangle_2|000\rangle_{345}$ to $|0\rangle_2(\alpha|0000\rangle+\beta|1111\rangle)_{1345}$, where qubits 1 and 2 are collocated in the same node, and its inverse operation corresponds to the *cat-disentangler* operation. For details, refer to Fig. 1.

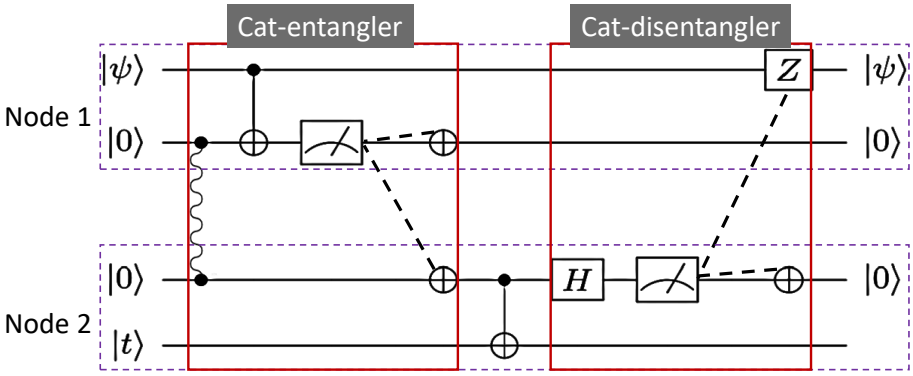


Fig. 2: Non-local CNOT circuit (two-node distributed system). $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ denotes the control qubit on Node 1. In Node 1, the first/second lines are register/channel qubits; in Node 2, the first/second lines are channel/register qubits. Wavy line generates an EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Recall that a local CNOT gate transforms the state as $\text{CNOT}|a\rangle|b\rangle \to |a\rangle|b \oplus a\rangle$, where $|a\rangle$ (control qubit) and $|b\rangle$ (target qubit) reside on the same node. The implementation of the *non-local* CNOT gate requires two nodes, called Node 1 and Node 2. Node 1 has one control qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and one channel qubit $|0\rangle$, while Node 2 has one target qubit $|t\rangle$ and one channel qubit $|0\rangle$. The implementation process is as follows (see Fig. 2):

1. Apply the *cat-entangler* operation to the 3-qubit system consisting of $(\alpha|0\rangle + \beta|1\rangle)|0\rangle$ in Node 1 and $|0\rangle$ in Node 2. This generates the shared state $\alpha|00\rangle + \beta|11\rangle$ between the control qubit in Node 1 and the channel qubit in Node 2, allowing each node to employ its locally paired qubit as a control qubit.
2. Perform the target local CNOT operation at Node 2. Then apply the *cat-disentangler* operation to retrieve the original control qubit and reset all channel qubits to $|0\rangle$ for subsequent reuse.

Communication complexity in DQC is quantified by the number of inter-node EPR pairs and bidirectional classical bits: each non-local CNOT operation consumes 1 EPR pair (for cross-node entanglement) and 2 classical bits (for measurement result feedback). Although multi-qubit entangled states such as GHZ may be generated during non-local operations, they can all be constructed from EPR pairs under local operations and classical communication (LOCC).

Notably, while entanglement generation is a critical factor in practical networks, this paper focuses on optimizing the *algorithmic logic* of non-local operations, which is the core bottleneck in our cryptographic setting. This aligns with the standard paradigm in the field, which separates quantum network implementation from distributed quantum algorithm design.

3 Distributed Simon's algorithm

In a distributed computing scenario, Simon's problem is generalized as follows:

Problem 4. **(Distributed Simon's Problem)** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a periodic function with non-trivial period $s \neq 0$, where $m \geq n-1$. The function f can be partitioned into 2^t subfunctions $f_w : \{0,1\}^{n-t} \to \{0,1\}^m$ parameterized by $w \in \{0,1\}^t$, such that $f_w(u) = f(u\|w)$ for all $u \in \{0,1\}^{n-t}$. Here, $0 < t < n$. The goal is to recover the period s by querying the subfunctions $\{f_w\}_{w\in\{0,1\}^t}$.

Remark 1. The subfunctions $\{f_w\}_{w\in\{0,1\}^t}$ are not restricted to decompositions over the last t bits of f's domain; arbitrary t bits of the input $x \in \{0,1\}^n$ are supported. Via the core technique of *logical relabeling of input bits*, the problem can be equivalently reduced to the canonical form with decomposition over the last t bits.

3.1 Our Distributed Simon's Algorithm

Based on the function $f(x)$ in Problem 4, we introduce an XOR-type periodic function $F(u)$, and present its periodicity property.

Property 1. For any $u \in \{0,1\}^{n-t}$, define a XOR-type periodic function as follows:

$$F(u) = \bigoplus_{w\in\{0,1\}^t} f_w(u). \tag{2}$$

Then $F(u)$ has a period s_1. Here, $s = s_1\|s_2$, where s_1 is of $n-t$ bits and s_2 is of t bits.

Proof. From the periodicity of $f(x)$, we have $f(x \oplus s) = f(x)$ for all $x \in \{0,1\}^n$. Since w traverses all elements of $\{0,1\}^t$, $w \oplus s_2$ also traverses $\{0,1\}^t$. Thus:

$$\begin{aligned} F(u) &= \bigoplus_{w\in\{0,1\}^t} f_w(u) = \bigoplus_{w\in\{0,1\}^t} f(u\|w) \\ &= \bigoplus_{w\in\{0,1\}^t} f((u\oplus s_1)\|(w\oplus s_2)) \\ &= \bigoplus_{w'\in\{0,1\}^t} f((u\oplus s_1)\|w') \\ &= \bigoplus_{w'\in\{0,1\}^t} f_{w'}(u\oplus s_1) = F(u\oplus s_1). \end{aligned}$$

This implies s_1 is a period of $F(u)$. □

Based on this property, we present a distributed quantum algorithm with 2^t+1 nodes to recover an element z satisfying $z\cdot s_1 = 0$, as shown in Algorithm 3. The quantum circuit in Fig. 3 corresponds to Algorithm 3 in the case $t = 1$. After $c(n-t)$ $(c > 3)$ repetitions, we can obtain $(n-t-1)$ linearly independent vectors

orthogonal to s_1 with high probability. Then, by using the classical Gaussian elimination method, s_1 can be recovered.

Algorithm 3 Distributed quantum algorithm for finding an element orthogonal to s_1.

Require: 2^t+1 nodes; For Node i ($2 \le i \le 2^t+1$), each maps to a unique t-bit string w_i = binary$(i-2)$, and has access to $O_{f_{w_i}} : |u\rangle|y\rangle \to |u\rangle|y \oplus f(u\|w_i)\rangle$, where $u \in \{0,1\}^{n-t}$ and $y \in \{0,1\}^m$.

Ensure: z satisfying $z \cdot s_1 = 0$.

1: Prepare the joint state of all 2^t+1 nodes as:

$$|0^{n-t}\rangle_{\text{Node 1}} \bigotimes_{i=2}^{2^t+1} |0^m\rangle_{\text{Node } i} = |0^{n-t}\rangle|0^{2^t m}\rangle.$$

2: Perform the quantum gate $H^{\otimes n-t}$ in Node 1:

$$\frac{1}{\sqrt{2^{n-t}}} \sum_{u\in\{0,1\}^{n-t}} |u\rangle|0^{2^t m}\rangle.$$

3: Under the control of Node 1, Node i ($2 \le i \le 2^t+1$) queries its respective $O_{f_{w_i}}$ in parallel:

$$\frac{1}{\sqrt{2^{n-t}}} \sum_{u\in\{0,1\}^{n-t}} |u\rangle \bigotimes_{w\in\{0,1\}^t} |f_w(u)\rangle.$$

4: Perform non-local CNOT gates to compute the XOR sum of these 2^t f_w values and store the result in Node 2:

$$\frac{1}{\sqrt{2^{n-t}}} \sum_{u\in\{0,1\}^{n-t}} |u\rangle|F(u)\rangle \bigotimes_{w\in\{0,1\}^t\setminus\{0^t\}} |f_w(u)\rangle.$$

5: Under the control of Node 1, Node i ($3 \le i \le 2^t+1$) queries its respective $O_{f_{w_i}}$ in parallel to reset their registers to $|0^m\rangle$:

$$\frac{1}{\sqrt{2^{n-t}}} \sum_{u\in\{0,1\}^{n-t}} |u\rangle|F(u)\rangle \bigotimes_{w\in\{0,1\}^t\setminus\{0^t\}} |0^m\rangle.$$

6: Perform the quantum gate $H^{\otimes n-t}$ in Node 1:

$$\begin{aligned}&\frac{1}{2^{n-t}} \sum_{u,z\in\{0,1\}^{n-t}} (-1)^{z\cdot u}|z\rangle|F(u)\rangle \\ &= \frac{1}{2^{n-t}} \sum_{\substack{u\in\{0,1\}^{n-t}/\langle s_1\rangle \\ z\in\{0,1\}^{n-t}}} (-1)^{z\cdot u}(1+(-1)^{z\cdot s_1})|z\rangle|F(u)\rangle.\end{aligned}$$

7: Measure the register qubits z of Node 1.

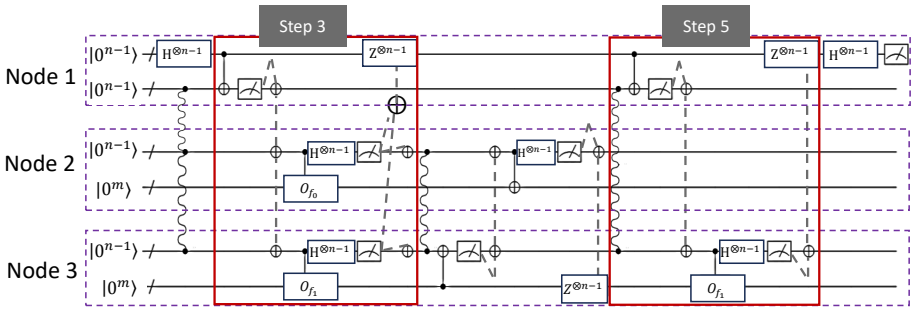


Fig. 3: Quantum circuit diagram for Algorithm 3 ($t = 1$). The operations between the two red boxes correspond to Step 4.

After recovering s_1, we use the distributed algorithm (see Algorithm 4) proposed by Tan et al. [26] to recover s_2.

3.2 Complexity Analysis

In distributed scenarios, query complexity refers to the maximum number of queries executed by any single node. Algorithm 3 performs at most 2 quantum queries for a single node, and is run $c(n-t)$ times to obtain $(n-t-1)$ vectors orthogonal to s_1 with high probability. Thus, recovering s_1 requires $2c(n-t)$ query complexity.

Algorithm 4 Distributed algorithm for recovering s_2 [26]

Require: 2^t nodes; Node i ($1 \le i \le 2^t$) maps to a unique t-bit string $w_i = \text{binary}(i-1)$, and can query $f_{w_i}(u) = f(u\|w_i)$; the previously recovered s_1.
Ensure: s_2
1: Each node makes a classical query to f_w once in parallel to obtain $f(0^{n-t}\|w)$, where $w \in \{0,1\}^t$.
2: Node 1 queries f_{0^t} once to obtain $f(s_1\|0^t)$.
3: Via classical communication, Node 1 collects all $f(0^{n-t}\|w)$, and then finds a vector $v \in \{0,1\}^t$ such that $f(0^{n-t}\|v) = f(s_1\|0^t)$.
4: Obtain $s_2 = v$.

Each node requires $(n-t)$ channel qubits for inter-node communication. Node 1 uses $(n-t)$ register qubits, while Node i ($2 \le i \le 2^t+1$) each uses m register qubits. Since $m \ge n-1$, it follows that $\max\{m+n-t, 2(n-t)\} = m+n-t$. Thus, the maximum number of qubits for a single node is $m+n-t$.

Step 4 computes the XOR sum of 2^t values $\{f_w\}_{w\in\{0,1\}^t}$ via non-local CNOT gates in parallel. The implementation follows a recursive pairwise aggregation strategy:

– Nodes are grouped into pairs, with each pair requiring $\lceil \frac{m}{n-t} \rceil$ non-local levels to complete the XOR operation (due to the bandwidth constraint imposed by $(n-t)$ channel qubits, which each non-local round can transmit up to $(n-t)$ qubits in parallel, so transmitting m register qubits requires $\lceil \frac{m}{n-t} \rceil$ levels);
– This pairwise aggregation is recursed for t levels (since 2^t nodes can be fully aggregated in t levels of a binary tree).

We estimate the communication complexity of Algorithm 3:

– Step 3: Nodes 2 to 2^t+1 execute respective $O_{f_{w_i}}$ in parallel under the control of Node 1. Using non-local operations, the $(n-t)$ channel qubits in each of these 2^t nodes are transformed into $\frac{1}{\sqrt{2^{n-t}}} \sum_{u \in \{0,1\}^{n-t}} |u\rangle$. Each node establishes $(n-t)$ EPR pairs with Node 1, leading to a communication overhead of $O(2^t(n-t))$.
– Step 4: Computing the XOR sum of these 2^t elements requires $O((2^t-1)m)$ non-local CNOT gates, leading to a communication overhead of $O((2^t-1)m)$.
– Step 5: Similar to Step 3, each of the last 2^t-1 nodes executes the oracle O_{f_w} under the control of Node 1, contributing a communication overhead of $O((2^t-1)(n-t))$.

Repeating Algorithm 3 $O(n-t)$ times to recover s_1 yields the total communication complexity $O((n-t)(2^t(n-t+m)-m))$.

Besides, s_2 is recovered in Algorithm 4 with 2^t nodes. The maximum number of queries for a single node is 2, and the communication complexity is $O(2^t)$.

Based on the analysis above, we obtain the following Theorem 3.

Theorem 3. *Consider the setting of Problem 4: Let $s = s_1 \| s_2$, where s_1 is of $n-t$ bits and s_2 is of t bits. There exists a distributed quantum algorithm for recovering s using 2^t+1 nodes. The maximum per-node quantum queries and qubit requirement are $2c(n-t)$ $(c>3)$ and $m+n-t$, respectively. The total communication complexity is $O((n-t)(2^t(n-t+m)-m))$, where the lower-order term $O(2^t)$ for recovering s_2 is negligible.*

Comparison. A complexity comparison of state-of-the-art distributed Simon's algorithms is summarized in Table 1. Specifically,

– The standard Simon's algorithm is structurally incompatible with the distributed oracle model defined in Problem 4. It relies on a single quantum computer with global access to the full function $f(x)$, which is impossible when f is partitioned into 2^t subfunctions and each node can only query its local f_w. By contrast, our algorithm is tailored to this constraint, which leverages quantum communication to aggregate distributed information. Even in terms of resource efficiency, our distributed variant outperforms the standard Simon's algorithm: while the standard Simon's algorithm requires $n+m$ qubits and cn quantum queries on a single node, our distributed scheme reduces the per-node quantum query complexity to $2c(n-t)$, which can reduce the per-node circuit depth, and the per-node qubit requirement to $m+n-t$. Here, $n/2 < t < n$ ensures $2c(n-t) < cn$

– Different from our distributed Simon's algorithm using XOR-type periodic function $F(u)$, the distributed Simon's algorithm proposed by Tan et al. [26] is designed based on a concatenation-type periodic function

$$S(u) = f(u\|w_0)\|f(u\|w_1)\|\cdots\|f(u\|w_{2^t-1}).$$

The implementation of $S(u)$ needs to gather all the sub-function values f_{w_i} at a node afterwards, which requires the node to employ exponentially many qubits with respect to t. In contrast, the implementation of $F(u)$ does not have such a requirement. Thus, compared with Tan et al.'s algorithm, our proposed algorithm matches the per-node quantum query complexity, while achieving the following two advantages:

- The per-node qubit requirement is far lower than the exponential $2^{t+1}m$ of Tan et al.'s scheme. This avoids the resource bottleneck of quantum hardware caused by excessive qubit consumption;
- Our communication complexity is lower than that of Tan et al.'s scheme.

Table 1: Comparisons of quantum algorithms for distributed Simon's problem. #Nodes: number of computing nodes; #Qubits: maximum number of qubits per node; #Queries: maximum number of queries per node. Here, $n/2 < t < n$ and $c > 3$.

Source	#Nodes	#Queries	#Qubits	Communication Complexity
[21]	1	cn	$n+m$	0
[26]	2^t+2	$2c(n-t)$	$2^{t+1}m$	$O((n-t)(2^t(m+n-t)))$
Ours	2^t+1	$2c(n-t)$	$m+n-t$	$O((n-t)(2^t(m+n-t)-m))$

4 Distributed Offline Simon's Algorithm

In distributed scenarios, the offline Simon's problem is formulated as follows:

Problem 5. **(Distributed Offline Simon's Problem)** Consider a function $F(k,x) \oplus g(x)$ that satisfies the offline Simon's problem criteria (see Problem 3), where there exists a unique k_0 such that $F(k_0,x) \oplus g(x)$ admits a non-trivial period $s \in \{0,1\}^n \setminus \{0^n\}$. Suppose $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ and $g : \{0,1\}^n \to \{0,1\}^m$ can be partitioned into $2^{t+t'}$ and 2^t subfunctions $F_{w',w}$ and g_w, respectively, with the following structures:

– $g_w : \{0,1\}^{n-t} \to \{0,1\}^m$, defined by $g_w(u) = g(u\|w)$, where $x = u\|w$, $u \in \{0,1\}^{n-t}$ and $w \in \{0,1\}^t$;
– $F_{w',w} : \{0,1\}^{\kappa-t'} \times \{0,1\}^{n-t} \to \{0,1\}^m$, defined by $F_{w',w}(k^l,u) = F(k^l\|w',u\|w)$, where $k = k^l\|w'$, $k^l \in \{0,1\}^{\kappa-t'}$ and $w' \in \{0,1\}^{t'}$.

Here, $1 \le t' < \kappa$ and $1 \le t < n$. The goal is to recover both k_0 and s by querying these subfunctions.

4.1 Quantum Algorithm for Problem 5 with XOR-Type Conditional Periodic Functions

For the function $F(k,x)\oplus g(x)$, we consider a special case: let α_k be the n-bit secret state determined by k, such that $F(k,x)\oplus g(x)$ reduces to the XOR-type conditional periodic function:

$$F(x\oplus\alpha_k)\oplus g(x). \tag{3}$$

When $\alpha_k=\alpha_{k_0}$, this function has a non-trivial period s. Since $\alpha_k\in\{0,1\}^n$ inherits x's decomposition ($\alpha_k=\alpha_k^l\|\alpha_k^r$, $\alpha_k^l\in\{0,1\}^{n-t}$ and $\alpha_k^r\in\{0,1\}^t$), we deduce that $t=t'$ holds for Problem 5.

Given $x=u\|w$ ($u\in\{0,1\}^{n-t}$, $w\in\{0,1\}^t$), the segment-wise XOR yields $x\oplus\alpha_k=(u\oplus\alpha_k^l)\|(w\oplus\alpha_k^r)$, which leads to

$$F(x\oplus\alpha_k)=F\left((u\oplus\alpha_k^l)\|(w\oplus\alpha_k^r)\right).$$

Suppose that $w''=w\oplus\alpha_k^r$ for reindexing. For fixed α_k^r, $w\mapsto w''$ is a bijection over $\{0,1\}^t$, so we can replace w'' with w without loss of generality. And then the expression simplifies to:

$$F\left((u\oplus\alpha_k^l)\|w\right)=F_w(u\oplus\alpha_k^l).$$

This eliminates the dependence on w', reducing $F_{w',w}$ to F_w.

To simplify the discussion, we assume that k is directly XORed with x, thus the XOR-type conditional periodic function $F(x\oplus k)\oplus g(x)$ is considered. We set $\kappa=n$ for simplicity, while retaining both symbols to distinguish the bit lengths of k and x. Based on this simplification, we derive the following key property of $F(x\oplus k)\oplus g(x)$.

Property 2. Suppose that the XOR-type conditional periodic function $F(x\oplus k)\oplus g(x)$ has a non-trivial period s when $k=k_0$. For any $u\in\{0,1\}^{n-t}$ and $k^l\in\{0,1\}^{\kappa-t}$, define a new function as follows:

$$F^{\mathcal{L}}(k^l,u)=G_1(u)\oplus G_2(k^l,u) \tag{4}$$

where $G_1(u)=\bigoplus_{w\in\{0,1\}^t}g(u\|w)$ and $G_2(k^l,u)=\bigoplus_{w\in\{0,1\}^t}F((u\oplus k^l)\|w)$. When $k^l=k_0^l$, $F^{\mathcal{L}}$ admits an $(n-t)$-bit period s_1. Here, $s=s_1\|s_2$, where $s_1\in\{0,1\}^{n-t}$ and $s_2\in\{0,1\}^t$. Similarly, $k=k^l\|k^r$ and $k_0=k_0^l\|k_0^r$.

Proof. First, $G_1(u)$ only depends on u, and $G_2(k^l, u)$ only depends on k^l and u. Then when $k^l = k_0^l$, we have

$$
\begin{aligned}
F^{\mathcal{L}}(k_0^l, u) &= \bigoplus_{w\in\{0,1\}^t} F_w(u \oplus k_0^l) \oplus g_w(u) \\
&= \bigoplus_{w\in\{0,1\}^t} F((u \oplus k_0^l)\|(w \oplus k_0^r)) \oplus g(u\|w) \\
&= \bigoplus_{w\in\{0,1\}^t} F((u \oplus k_0^l \oplus s_1)\|(w \oplus k_0^r \oplus s_2)) \oplus \\
g((u \oplus s_1)\|(w \oplus s_2)) &= \bigoplus_{w\in\{0,1\}^t} F((u \oplus k_0^l \oplus s_1)\|w) \\
\oplus\, g((u \oplus s_1)\|w) &= \bigoplus_{w\in\{0,1\}^t} F_w(u \oplus k_0^l \oplus s_1) \oplus g_w(u \oplus s_1) \\
&= F^{\mathcal{L}}(k_0^l, u \oplus s_1).
\end{aligned}
$$

Thus, $F^{\mathcal{L}}(k_0^l, u)$ has a period s_1. Note that $w \mapsto w \oplus k_0^r \oplus s_2$ is a bijection over $\{0,1\}^t$, so the summation range is preserved. □

Based on Property 2, we present a distributed quantum algorithm (see Algorithm 5) for finding k_0^l and s_1, which combines the original offline Simon's algorithm with the method in Section 3.1. Fig. 4 illustrates the overview of this algorithm. The idea is to decompose the process into three stages.

1. **Online phase:** prepare $|\psi_{G_1}\rangle = \bigotimes_{j=1}^{c'} \sum_{u_j\in\{0,1\}^{n-t}} |u_j\rangle|G_1(u_j)\rangle$ with distributed $2^t + 1$ nodes by querying respective $O_{g_{w_i}}$.
2. **Offline computation phase:** scale the number of nodes to $2^t + 2$, and perform Grover search over $k^l \in \{0,1\}^{\kappa-t}$.
   - For each fixed k^l, verify whether $G_1(u) \oplus G_2(k^l, u)$ admits a period by using $|\psi_{G_1}\rangle$ and querying respective $O_{F_{w_i}}$;
   - Appropriate uncomputation operations are executed to recover $|\psi_{G_1}\rangle$ for iteratively reusing.
3. **Period recovery:** after recovering k_0^l, Algorithm 3 is performed to recover s_1.

After finding k_0^l and s_1, we present Algorithm 6 to recover k_0^r and s_2 based on $F(k_0^l\|(k_0^r \oplus s_2)) \oplus g(0^{n-t}\|s_2) = F((s_1 \oplus k_0^l)\|k_0^r) \oplus g(s_1\|0^t)$.

4.2 Complexity Analysis

As the expected number of correct solutions is small and the search space is $\{0,1\}^{\kappa-t}$, we perform $O(2^{(\kappa-t)/2})$ Grover iterations. Based on Theorem 1, setting $c' = \kappa - t + n - t + \tau + 1$ suffices to recover k_0^l with high probability.

In the Q2 model, Node i ($3 \le i \le 2^t + 2$) is assigned to make quantum queries on the subfunction g_{w_i} ($w_i = \text{binary}(i-3)$). The quantum state $|\psi_{G_1}\rangle$

can be generated by performing the process from Steps 1 to 5 from Algorithm 3, which implements $O_{G_1} : |u\rangle|y\rangle \to |u\rangle|y \oplus G_1(u)\rangle$, c' times in parallel. Since $|\psi_{G_1}\rangle$ is reused, the maximum number of per-node quantum queries to $O_{g_{w_i}}$ $(w_i \in \{0,1\}^t)$ is $2c'$.

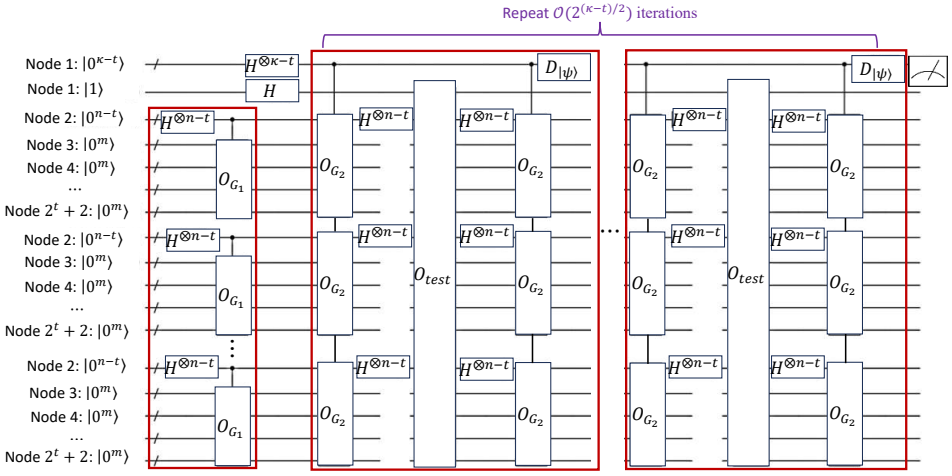


Fig. 4: Quantum circuit diagram corresponding to Algorithm 5. Here, each node only shows register qubits, and channel qubits are omitted for simplicity. The operations in the first red box from the left are used to prepare $|\psi_{G_1}\rangle$ (correspond to Step 1). Both the operations in the second and third red boxes correspond to Grover iteration. $D_{|\psi\rangle} = H^{\otimes\kappa-t}(2|0^{\kappa-t}\rangle\langle 0^{\kappa-t}| - I_{\kappa-t})H^{\otimes\kappa-t}$.

In the Q1 model, Node i $(3 \le i \le 2^t + 2)$ has to perform 2^{n-t} classical queries to g_{w_i} to emulate the quantum superposition of $|u\rangle|G_1(u)\rangle$. Since classical computation cannot directly generate the state $|\psi_{G_1}\rangle$, all inputs $u \in \{0,1\}^{n-t}$ have to be traversed. This results in a maximum of 2^{n-t} classical queries per node to $\{g_{w_i}\}_{w_i\in\{0,1\}^t}$.

To implement O_{G_2}, for each fixed k^l, we can add it to u_j $(j = 1, 2, \cdots, c')$ by performing non-local CNOT gates, and the rest is similar to prepare $|\psi_{G_1}\rangle$. In offline quantum computing, Node i $(3 \le i \le 2^t + 2)$ can perform quantum queries to F_{w_i} in both the Q2 model and Q1 model. $O(2^{(\kappa-t)/2})$ iterations result in a maximum of $4c'2^{(\kappa-t)/2}$ quantum queries per-node to $O_{F_{w_i}}$ $(w_i \in \{0,1\}^t)$.

Based on Table 1, both the maximum number of qubits required for the node implementing O_{G_1} and O_{G_2} is $m + n - t$. Thus, the number of qubits required for Nodes 3 to $2^t + 2$ is $A = c'(m + n - t)$. O_{test} is used to compute the rank of c' $(n - t)$-bit vectors, which is completed with the circuit in [4], and XOR r to b. Thus, the number of qubits for Node 2 is $B = 2c'(n - t) + W_{test}[c', n - t]$. Besides, Node 1 requires $C = \kappa - t + c'(n - t) + 1$ qubits. Note that $c'(n - t)$ channel qubits are required with each node.

Algorithm 5 Distributed quantum algorithm for recovering k_0^l and s_1

Require: 2^t+2 nodes; For Node i ($3 \leq i \leq 2^t+2$), each maps to a unique t-bit string $w_i = \text{binary}(i-3)$, has access to the oracle $O_{F_{w_i}} : |k^l\rangle|u\rangle|y\rangle \to |k^l\rangle|u\rangle|y \oplus F((u \oplus k^l)\|w_i)\rangle$ and $O_{g_{w_i}} : |u\rangle|y\rangle \to |u\rangle|y \oplus g(u\|w_i)\rangle$

Ensure: k_0^l, s_1

1: For Nodes 2 to 2^t+2, perform the process from Steps 1 to 5 in Algorithm 3 to prepare

$$|\psi_{G_1}\rangle = \bigotimes_{j=1}^{c'} \sum_{u_j \in \{0,1\}^{n-t}} |u_j\rangle|G_1(u_j)\rangle.$$

2: For Node 1, apply the quantum gate $H^{\otimes(\kappa-t+1)}$ on $|0^{\kappa-t}\rangle|1\rangle$ to create

$$\sum_{k^l \in \{0,1\}^{\kappa-t}} |k^l\rangle|b\rangle,$$

where $|b\rangle = |-\rangle$.

3: Under the control of Node 1 and Node 2, Node i ($3 \leq i \leq 2^t+2$) performs the oracle $O_{F_{w_i}}$ in parallel via non-local operations to generate

$$\sum_{k^l \in \{0,1\}^{\kappa-t}} |k^l\rangle \bigotimes_{j=1}^{c'} \sum_{u_j \in \{0,1\}^{n-t}} |u_j\rangle|F^{\mathcal{L}}(k^l, u_j)\rangle.$$

4: For Node 2, execute quantum gate $H^{\otimes n-t}$ on u_j to obtain

$$\sum_{k^l \in \{0,1\}^{\kappa-t}} |k^l\rangle \bigotimes_{j=1}^{c'} \sum_{u_j, v_j \in \{0,1\}^{n-t}} (-1)^{u_j \cdot v_j} |v_j\rangle|F^{\mathcal{L}}(k^l, u_j)\rangle.$$

5: For Node 2, compute the dimension $d = dim(Span(v_1, v_2, \cdots, v_{c'}))$. If $d = n-t$, set $r = 0$; If $d < n-t$, set $r = 1$. XOR r to b by performing a non-local CNOT gate, and then uncompute d and r to obtain

$$\sum_{k^l \in \{0,1\}^{\kappa-t}} |k^l\rangle \bigotimes_{j=1}^{c'} \sum_{u_j, v_j \in \{0,1\}^{n-t}} (-1)^{u_j \cdot v_j} |v_j\rangle|F^{\mathcal{L}}(k^l, u_j)\rangle \otimes |b \oplus r\rangle.$$

6: Uncompute Steps 3-4 to revert the state

$$\sum_{k^l \in \{0,1\}^{\kappa-t}} |k^l\rangle|b \oplus r\rangle \otimes |\psi_{G_1}\rangle$$

7: Apply $O(2^{(\kappa-t)/2})$ Grover iterations, measure k^l.

8: After obtaining k_0^l, apply Algorithm 3 to obtain s_1.

Based on Table 1, the quantum communication complexity of implementing O_{G_1} or O_{G_2} is $O(2^t(m+n-t)-m)$. Thus, both the quantum communication complexity of step 1 and step 3 is $O(c'(2^t(m+n-t)-m))$. Using $O(2^{(\kappa-t)/2})$ iterations, the quantum communication complexity of Algorithm 5 is $O(2^{(\kappa-t)/2}[c'(2^t(m+n-t)-m)])$.

Algorithm 6 recovers k_0^r and s_2 with $O(2^{2t})$ classical communication. The maximum number of classical queries to $\{g_{w_i}\}_{w_i\in\{0,1\}^t}$ for a single node is $O(1)$.

Algorithm 6 Distributed algorithm for recovering s_2 and k_0^r

Require: 2^t nodes, where:

- Node i $(1 \le i \le 2^t)$ maps to a unique t-bit string $w_i = \text{binary}(i-1)$;
- Node i can query $F_{w_i}(u) = F(u\|w_i)$ and $g_{w_i}(u) = g(u\|w_i)$ for all $u \in \{0,1\}^{n-t}$.

Previously recovered s_1, k_0^l.

Ensure: s_2, k_0^r.

1: Node 1 queries $g_{0^t}(s_1)$ to compute $C = g(s_1\|0^t)$, then broadcasts C to all nodes.

2: For each Node j $(1 \le j \le 2^t)$ in parallel:

- query $g_{w_j}(0^{n-t})$ once to get $G(w_j) = g(0^{n-t}\|w_j)$;
- Broadcast $G(w_j)$ to all nodes.

3: For each Node i $(1 \le i \le 2^t)$ in parallel:

- Query $F_{w_i}(k_0^l \oplus s_1)$ once to get $F_2(w_i) = F((k_0^l \oplus s_1)\|w_i)$;
- For each Node j $(1 \le j \le 2^t)$:
  - Compute $w_m = w_i \oplus w_j$ (Node m maps to w_m);
  - Node m queries $F_{w_m}(k_0^l)$ once to get $F_1(w_m) = F(k_0^l\|w_m)$, then sends to Node i;
  - Node i computes LHS $= F_2(w_i) \oplus F_1(w_m)$ and RHS $= C \oplus G(w_j)$;
  - If LHS = RHS:
    - Node i broadcasts $(s_2 = w_j, k_0^r = w_i)$ to all nodes;
    - All nodes terminate immediately.

4: Output s_2 and k_0^r.

When $t \le n/3$, we have $2^{(\kappa+t)/2} \ge 2^{2t}$ and $B > A > C$. In summary, we give the complexity of solving Problem 5 as follows:

Theorem 4. *Suppose that $0 < t \le n/3$ and $c' = \kappa - t + n - t + \tau + 1$. Under the setting of Problem 5, let $k_0 \in \{0,1\}^\kappa$ be the good element such that $g(x) \oplus F(x \oplus k_0)$ has a non-trivial period s. Combining Algorithm 5 with Algorithm 6, k_0 and s can be recovered with $2^t + 2$ nodes.*

The maximum per-node quantum queries to $O_{F_{w_i}}$ $(w_i \in \{0,1\}^t)$ are $O((n-t)2^{(\kappa-t)/2})$. For $\{g_{w_i}\}_{w_i\in\{0,1\}^t}$, the maximum per-node quantum queries are $O(n-$

t) in the Q2 model, and classical queries are $O(2^{n-t})$ in the Q1 model. The maximum per-node qubit requirement is $2c'(n-t)+W_{test}[c', n-t]$, and the communication complexity is $\tilde{O}(2^{(\kappa+t)/2})$.

Excluding the preparation of $|\psi_{G_1}\rangle$, the offline quantum computation in both models runs in time

$$\tilde{O}(2^{(\kappa+t)/2}T_c + ((n-t)^3 + (n-t)T_{F_w})2^{(\kappa-t)/2}).$$

Here, T_{F_w} denotes the time to evaluate F once, and T_c is the inter-node communication latency.

Comparison. By comparing Theorem 4 (our distributed algorithm) with Theorem 2 (the standard offline Simon's algorithm), we can observe the advantages of the former over the latter, as summarized below:

- The maximum per-node qubit requirement is lower than the $\kappa + c(n+m) + W_{\text{test}}[c, n]$ required by the standard offline Simon's algorithm.
- The maximum per-node quantum queries to g is $2c'$ in the Q2 model instead of $2c$, while the classical query complexity is 2^{n-t} in the Q1 model instead of 2^n.

4.3 Generalization to Multi-XOR-Type Conditional Periodic Functions

In this section, we consider a more general case: $F(k, x) \oplus g(x)$ is expressed as a multi-XOR-type conditional periodic function

$$F_1(x \oplus k_1) \oplus F_2(x \oplus k_2) \oplus \cdots \oplus F_a(x \oplus k_a) \oplus g(x), \tag{5}$$

where $a \geq 1$. When $k_1 = h_1, k_2 = h_2, \cdots, k_a = h_a$, the function has a period s. Similarly, we define a function $F^{\mathcal{L}} : \{0,1\}^{a(\kappa-t)} \times \{0,1\}^{n-t} \to \{0,1\}^m$ as

$$F^{\mathcal{L}}(k_1^l \| k_2^l \| \cdots \| k_a^l, u) = G_1(u) \oplus G_2(k_1^l \| k_2^l \| \cdots \| k_a^l, u), \tag{6}$$

where

$$G_1(u) = \bigoplus_{w \in \{0,1\}^t} g(u \| w),$$

$$G_2(k_1^l \| k_2^l \| \cdots \| k_a^l, u) = \bigoplus_{w \in \{0,1\}^t} F_1((u \oplus k_1^l) \| w) \oplus$$

$$F_2((u \oplus k_2^l) \| w) \oplus \cdots \oplus F_a((u \oplus k_a^l) \| w).$$

Aggregating over all $w \in \{0,1\}^t$ yields

$$F^{\mathcal{L}}(h_1^l \| h_2^l \| \cdots \| h_a^l, u) = F^{\mathcal{L}}(h_1^l \| h_2^l \| \cdots \| h_a^l, u \oplus s_1). \tag{7}$$

Thus, $F^{\mathcal{L}}(h_1^l \| h_2^l \| \cdots \| h_a^l, u)$ has an $(n-t)$-bit period s_1. Here, $s = s_1 \| s_2$ where $s_1 \in \{0,1\}^{n-t}$ and $s_2 \in \{0,1\}^t$. Similarly, $h_j = h_j^l \| h_j^r$ and $k_j = k_j^l \| k_j^r$ for $j = 1, 2, \cdots, a$.

In distributed scenarios, we recover $\{h_j^l\}_{j=1}^a$ and s_1 by modifying Algorithm 5. Specifically:

– Instead of $\sum_{k^l\in\{0,1\}^{\kappa-t}}|k^l\rangle|b\rangle$, in Step 2 we prepare the quantum state $\sum_{k_1^l,k_2^l,\cdots,k_a^l\in\{0,1\}^{\kappa-t}}|k_1^l,k_2^l,\cdots,k_a^l\rangle|b\rangle$.
– Instead of the single oracle $O_{F_{w_i}}$ in Step 3, we use oracles $\{O_{F_{j,w_i}}\}_{j=1}^a$ to compute $F^{\mathcal{L}}(k_1^l\|k_2^l\|\cdots\|k_a^l,u)$.
– We perform $O(2^{a(\kappa-t)/2})$ Grover iterations, instead of $O(2^{(\kappa-t)/2})$.
– $c''=a(\kappa-t)+(n-t)+\tau+1$ instead of $(\kappa-t)+(n-t)+\tau+1$, and the communication complexity becomes $\tilde{O}(2^{a(\kappa-t)/2+t})$.
– Node 1 requires $a(\kappa-t)+c''(n-t)+1$ qubits instead of $\kappa-t+c'(n-t)+1$.

Then, based on the equation (derived from Eq. (5)):

$$\bigoplus_{j=1}^{a}F_j(h_j^l\|(h_j^r\oplus s_2))\oplus g(0^{n-t}\|s_2)=\bigoplus_{j=1}^{a}F_j((s_1\oplus h_j^l)\|h_j^r)\oplus g(s_1\|0^t),\tag{8}$$

we extend Algorithm 6 to recover $\{h_j^r\}_{j=1}^a$ and s_2. The key differences from Algorithm 6 are summarized as follows:

– Whereas Algorithm 6 employs single-node traversal (each node i maps uniquely to the single key suffix candidate $k_0^r\leftrightarrow w_i$), the generalized algorithm adopts *a-node group traversal* with $\mathcal{I}=(i_1,i_2,\ldots,i_a)$. Each node i_m in group $\mathcal{I}$ maps to a distinct key suffix candidate $h_m^r\leftrightarrow w_{i_m}$, enabling the joint traversal of all independent key suffix candidates $\{h_1^r,h_2^r,\ldots,h_a^r\}$.
– The single F-term XOR operation (i.e., LHS $=F_2(w_i)\oplus F_1(w_m)$ and RHS $=C\oplus G(w_j)$ in Algorithm 6) is replaced with the aggregated XOR of a F_m-terms (one per F_j) for both sides of the validation equation:

$$\begin{aligned}\mathrm{LHS}_{\mathrm{total}}&=\bigoplus_{m=1}^{a}F_m(h_m^l\|(w_{i_m}\oplus w_j))\oplus g(0^{n-t}\|w_j),\\ \mathrm{RHS}_{\mathrm{total}}&=\bigoplus_{m=1}^{a}F_m((s_1\oplus h_m^l)\|w_{i_m})\oplus g(s_1\|0^t),\end{aligned}$$

where w_j denotes the candidate value of s_2 (consistent with Algorithm 6), and h_m^l is the pre-recovered left half of the m-th key.
– A dedicated coordinator node (e.g., the first node i_1 in group $\mathcal{I}$) is introduced to: (1) aggregate F_m-term query results from all a nodes in the group; (2) compute $\mathrm{LHS}_{\mathrm{total}}$ and $\mathrm{RHS}_{\mathrm{total}}$; (3) validate the core periodic equation; (4) broadcast the valid candidates $\{h_m^r=w_{i_m}\}_{m=1}^a$ and $s_2=w_j$ upon a match. This replaces the peer-to-peer communication model (where each node i independently validates the equation locally) in Algorithm 6, reducing communication overhead and ensuring consistency in multi-parameter validation.
– The communication complexity becomes $O(2^{t(a+1)})$ instead of $O(2^{2t})$.

Combining the above analysis, we obtain Theorem 5.

Theorem 5. *Suppose that* $a \geq 1$, $0 < t \leq n/3$, *and* $c'' = a(\kappa - t) + n - t + \tau + 1$. *Let* $h_1, h_2, \cdots, h_a \in \{0,1\}^\kappa$ *be the good elements such that* $F_1(x \oplus h_1) \oplus F_2(x \oplus h_2) \oplus \cdots \oplus F_a(x \oplus h_a) \oplus g(x)$ *admits a period* s. *In the distributed setting,* $h_1, h_2, \cdots, h_a$ *and* s *can be recovered with* $2^t + 2$ *nodes.*

The maximum number of quantum queries per node to $\{F_{j,w_i}\}_{j=1,w_i\in\{0,1\}^t}^a$ *is* $O\left(a(n-t)2^{a(\kappa-t)/2}\right)$. *For* $\{g_{w_i}\}_{w_i\in\{0,1\}^t}$, *the maximum per-node quantum queries in the Q2 model are* $O(a(n-t))$, *while the classical query complexity is* $O(2^{n-t})$ *in the Q1 model. The maximum per-node qubit requirement is* $2c''(n-t) + W_{test}[c'', n-t]$, *and the communication complexity is* $\tilde{O}\left(2^{a(\kappa-t)/2+t}\right)$.

Excluding the preparation of $|\psi_{G_1}\rangle$, *the offline quantum computation in both models runs in time*

$$\tilde{O}(2^{a(\kappa-t)/2+t}T_c + ((n-t)^3 + (n-t)\sum_{j=1}^{a} T_{F_j})2^{a(\kappa-t)/2}).$$

Here, T_{F_i} $(i = 1, \cdots, a)$ *denotes the time to evaluate* F_i *once, and* T_c *is the inter-node communication latency.*

4.4 Summary

This section formalizes the distributed offline Simon's problem and designs a quantum algorithm for its special case with XOR-type conditional periodic function based on the result in Section 3. Complexity analysis under the Q1 and Q2 models proves that it reduces per-node qubit requirements compared with the standard offline Simon's algorithm. We further extend the framework to multi-XOR-type conditional periodic functions and verify its scalability.

5 Quantum Attacks on Permutation-Based Pseudorandom Functions

In this section, we apply the distributed offline Simon's algorithm from Section 4 to analyze a pseudorandom function (PRF) utilizing two parallel permutations, and evaluate the security of its concrete instantiation SoEM22.

For simplicity, we assume that one evaluation of a primitive (e.g., a block cipher or public permutation) and one inter-node communication can both be done in time $O(1)$.

5.1 Attacks on a PRF Using Two Parallel Permutations

A pseudorandom function using two public parallel permutations $P_1, P_2 \in \mathrm{Perm}(n)$ is defined as follows [9]:

$$\begin{aligned} &f : \{0,1\}^n \to \{0,1\}^n, \\ &x \mapsto l_{00}P_1(l_{01}x \oplus l_{02}k_1) \oplus l_{10}P_2(l_{11}x \oplus l_{12}k_2) \oplus h(x) \oplus e, \end{aligned} \tag{9}$$

where

- $h(x) = (l_{20}l_{21} \oplus l_{22}l_{23})x$;
- $e = l_{30}l_{31}k_1 \oplus l_{32}l_{33}k_2 \oplus l_{34}k_3 \oplus l_{35}k_4$;
- All coefficients $l_{ij} \in \{0,1\}^n$ are public, with $l_{00}, l_{01}, l_{02}, l_{10}, l_{11}, l_{12} \neq 0$;
- $k_1, k_2, k_3, k_4 \in \{0,1\}^n$ are secret keys.

Attacks in the Q2 model. Define $g : \{0,1\}^n \to \{0,1\}^n$ and $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ as

$$\begin{aligned} g(x) &= f(x) \oplus l_{00}P_1(l_{01}x) \oplus h(x), \\ F(x \oplus \alpha_k) &= l_{10}P_2(l_{11}(x \oplus \alpha_k)). \end{aligned} \tag{10}$$

One can verify that $F(x \oplus \alpha_k) \oplus g(x)$ admits a period $s = l_{01}^{-1}l_{02}k_1$ when

$$\alpha_k = l_{11}^{-1}l_{12}k_2 \quad \text{or} \quad \alpha_k = l_{11}^{-1}l_{12}k_2 \oplus l_{01}^{-1}l_{02}k_1.$$

This exactly matches the XOR-type conditional periodic function studied in Section 4.1.

We partition $g(x)$ and $F(x \oplus k)$ into 2^t subfunctions

$$\begin{aligned} g_w &: \{0,1\}^{n-t} \to \{0,1\}^n, \\ F_w &: \{0,1\}^{n-t} \times \{0,1\}^{n-t} \to \{0,1\}^n, \end{aligned} \tag{11}$$

such that

$$g_w(u) = g(u\|w), \quad F_w(\alpha_k^l \oplus u) = F((u \oplus \alpha_k^l)\|w),$$

where $w \in \{0,1\}^t$. We further define the following function

$$F^{\mathcal{L}}(\alpha_k^l, u) = \bigoplus_{w \in \{0,1\}^t} F_w(u \oplus \alpha_k^l) \oplus g_w(u). \tag{12}$$

One can verify that $F^{\mathcal{L}}(\alpha_k^l, u)$ has period $s_1 = \left(l_{01}^{-1}l_{02}k_1\right)^l$ when

$$\alpha_k^l = \left(l_{11}^{-1}l_{12}k_2\right)^l \quad \text{or} \quad \alpha_k^l = \left(l_{11}^{-1}l_{12}k_2\right)^l \oplus \left(l_{01}^{-1}l_{02}k_1\right)^l.$$

Here $s = s_1\|s_2$ and $\alpha_k = \alpha_k^l\|\alpha_k^r$, which is consistent with Section 4.

Our key-recovery attack proceeds in three steps:

1. Apply Algorithm 5 to $F^{\mathcal{L}}(\alpha_k^l, u)$ in Eq. (12) to recover s_1 and $\left(l_{11}^{-1}l_{12}k_2\right)^l$.
2. Run Algorithm 6 to recover $s_2 = \left(l_{01}^{-1}l_{02}k_1\right)^r$ and $\left(l_{11}^{-1}l_{12}k_2\right)^r$. Since l_{01}, l_{02} are publicly known, we can combine s_1 and s_2 to get s, and then solve $k_1 = l_{02}^{-1}l_{01}s$. Similarly, α_k^l and α_k^r can be combined to recover k_2.
3. Derive the linear combination of the remaining keys:

$$l_{34}k_3 \oplus l_{35}k_4 = f(0^n) \oplus l_{00}P_1(l_{02}k_1) \oplus l_{10}P_2(l_{12}k_2) \oplus l_{30}l_{31}k_1 \oplus l_{32}l_{33}k_2.$$

For $0 < t \leq n/3$, Theorem 4 guarantees that Algorithm 5 and Algorithm 6 can recover k_1 and k_2 with $2^t + 2$ nodes in the Q2 model. The maximum number of per-node superposition queries to f is $O(n-t)$, and to public permutations is $O((n-t)2^{(n-t)/2})$. The maximum per-node qubit requirement is $O((n-t)^2)$, and

the communication complexity is $\tilde{O}(2^{(n+t)/2})$. Step 3 only requires $O(1)$ queries to f and public permutations.

Attacks in the Q1 model. We introduce a new parameter $0 < t' < n$ for hierarchical splitting, and define the functions $g : \{0,1\}^{n-t'} \to \{0,1\}^n$ and $F : \{0,1\}^{t'+n} \times \{0,1\}^{n-t'} \to \{0,1\}^n$ as follows:

$$\begin{aligned} &g(x) = f(x\|0^{t'}) \oplus h(x\|0^{t'}), \\ &F(i_1\|i_2, x) = l_{00}P_1(l_{01}(x\|i_1)) \oplus l_{10}P_2(l_{11}(x\|0^{t'} \oplus i_2)). \end{aligned}$$

Notably, the function $F(i_1\|i_2, x) \oplus g(x)$ has a period $s = (l_{01}^{-1}l_{02}k_1)^{l'}$ when $i_1 = (l_{01}^{-1}l_{02}k_1)^{r'}$, $i_2 = l_{11}^{-1}l_{12}k_2$ or $i_2 = (l_{11}^{-1}l_{12}k_2 \oplus l_{01}^{-1}l_{02}k_1)^{l'}\|(l_{11}^{-1}l_{12}k_2)^{r'}$. Here, $l_{01}^{-1}l_{02}k_1 = (l_{01}^{-1}l_{02}k_1)^{l'}\|(l_{01}^{-1}l_{02}k_1)^{r'}$, where $(l_{01}^{-1}l_{02}k_1)^{l'} \in \{0,1\}^{n-t'}$ and $(l_{01}^{-1}l_{02}k_1)^{r'} \in \{0,1\}^{t'}$.

Define the following functions:

$$\begin{aligned} &F_1((x\|0^{t'}) \oplus (0^{n-t'}\|i_1)) = l_{00}P_1(l_{01}((x\|0^{t'}) \oplus (0^{n-t'}\|i_1))), \\ &F_2((x\|0^{t'}) \oplus i_2) = l_{10}P_2(l_{11}((x\|0^{t'}) \oplus i_2)), \end{aligned}$$

then $F(i_1\|i_2, x)$ can be rewritten as $F_1(x\|0^{t'} \oplus 0^{n-t'}\|i_1) \oplus F_2(x\|0^{t'} \oplus i_2)$. Similarly, $g(x)$ can be rewritten as $g'(x\|0^{t'}) = g(x)$. It follows that $F_1((x\|0^{t'}) \oplus (0^{n-t'}\|i_1)) \oplus F_2((x\|0^{t'}) \oplus i_2) \oplus g'(x\|0^{t'})$ is a multi-XOR-type conditional periodic function.

We define the following function $F^{\mathcal{L}} : \{0,1\}^{2n-2t-p} \times \{0,1\}^p \to \{0,1\}^n$ as

$$\begin{aligned} F^{\mathcal{L}}(i\|j, u) = &\bigoplus_{w \in \{0,1\}^t} F_1((u\|0^{n-t-p} \oplus 0^p\|i)\|w) \\ &\oplus F_2((u\|0^{n-t-p} \oplus j)\|w) \oplus g'(u\|0^{n-t-p}\|w). \end{aligned} \tag{13}$$

It can be verified that $F^{\mathcal{L}}(i\|j, u)$ has a p-bit ($0 < p < n - t$) period s_{11}, when $i = s_{12}$, $j = (l_{11}^{-1}l_{12}k_2)^l$ or $j = (s_{11} \oplus (l_{11}^{-1}l_{12}k_2)^{l_1})\|(l_{11}^{-1}l_{12}k_2)^{l_2}$. Here, $s_1 = (l_{01}^{-1}l_{02}k_1)^l = s_{11}\|s_{12}$ ($s_{11} \in \{0,1\}^p$, $s_{12} \in \{0,1\}^{n-t-p}$). Similarly, $(l_{11}^{-1}l_{12}k_2)^l = (l_{11}^{-1}l_{12}k_2)^{l_1}\|(l_{11}^{-1}l_{12}k_2)^{l_2}$ ($(l_{11}^{-1}l_{12}k_2)^{l_1} \in \{0,1\}^p$, and $(l_{11}^{-1}l_{12}k_2)^{l_2} \in \{0,1\}^{n-t-p}$).

As described in Section 4.3, we can recover $s_{11}\|s_{12} = (l_{01}^{-1}l_{02}k_1)^l$, $(l_{11}^{-1}l_{12}k_2)^l$, $(l_{01}^{-1}l_{02}k_1)^r$ and $(l_{11}^{-1}l_{12}k_2)^r$ by modifying Algorithm 5 and Algorithm 6. After obtaining k_1 and k_2, we can compute $l_{34}k_3 \oplus l_{35}k_4$ using the same formula in Step 3 of the Q2 attack.

Suppose that $0 < t \leq 2n/9$ and $0 < p < n - t$. By Theorem 5, our attack recovers the secret keys with $2^t + 2$ nodes in the Q1 model by making a maximum of $D = O(2^p)$ classical queries per node to f and $O((n-t)2^{(2n-2t-p)/2})$ quantum queries to public permutations per node. The communication complexity is $\tilde{O}(2^{(2n-p)/2})$. The offline quantum computation runs in time $T = \tilde{O}((n-t)2^{(2n-2t-p)/2} + 2^{(2n-p)/2}) = \tilde{O}(2^{(2n-p)/2})$. This gives a time-query tradeoff of $DT^2 = 2^{2n}$, which is balanced at $p \approx 2n/3$, yielding $T = D = \tilde{O}(2^{2n/3})$. Besides, the maximum per-node qubit requirement is $O((n-t)^2)$.

5.2 Instantiation: Attacking SoEM22

The Sum of Even-Mansour (SoEM) construction [5], formed by XORing two Even-Mansour cipher instances [7], is an instantiation of PRF with two parallel permutations. As an SoEM variant, the SoEM22 construction is defined as:

$$\text{SoEM22}(x) = P_1(x \oplus k_1) \oplus P_2(x \oplus k_2) \oplus k_1 \oplus k_2, \tag{14}$$

where P_1 and P_2 are two independent public permutations in $\text{Perm}(n)$, k_1 and k_2 are two n-bit keys. It corresponds to Eq. (9) with: $l_{00} = l_{01} = l_{02} = l_{10} = l_{11} = l_{12} = 1$, $h(x) = 0$, and $e = k_1 \oplus k_2$.

Our distributed quantum attack framework simplifies when instantiated to SoEM22:

- In the Q2 model, the period $s = l_{01}^{-1} l_{02} k_1$ reduces to $s = k_1$, and the functions in Eq. (10) simplify to $g(x) = \text{SoEM22}(x) \oplus P_1(x)$ and $F(x \oplus k) = P_2(x \oplus k)$.
- Step 3 of the attack procedure is omitted, and k_1, k_2 are directly recovered via periodicity analysis.
- In the Q1 model, the period $s = (l_{01}^{-1} l_{02} k_1)^{l'}$ reduces to $k_1^{l'}$, and the functions simplify to $g(x) = \text{SoEM22}(x \| 0^{t'})$ and $F(i_1 \| i_2, x) = P_1(x \| i_1) \oplus P_2((x \| 0^{t'}) \oplus i_2)$.

Comparison. We compare our distributed quantum attacks on the SoEM22 construction with existing non-distributed quantum attacks. Key metrics of each attack are summarized in Table 2.

Table 2: Comparisons of key recovery attacks on SoEM22. #Queries: maximum number of queries to SoEM22 per node; #Qubits: maximum number of qubits per node. $0 < t \leq n/3$ in the Q2 model, and $0 < t \leq 2n/9$ in the Q1 model.

Source	Setting	#Nodes	#Queries	Time	#Qubits
[9]	Q2	1	$O(n2^{n/2})$	$\tilde{O}(2^{n/2})$	$O(n^2)$
[25]	Q2	1	$O(n)$	$\tilde{O}(2^{n/2})$	$O(n^2)$
	Q1	1	$O(2^{2n/3})$	$\tilde{O}(2^{2n/3})$	$O(n^2)$
Ours	Q2	$2^t + 2$	$O(n-t)$	$\tilde{O}(2^{(n+t)/2})$	$O((n-t)^2)$
	Q1	$2^t + 2$	$O(2^{2n/3})$	$\tilde{O}(2^{2n/3})$	$O((n-t)^2)$

As shown in Table 2, the Q2 attack in [9] achieves $\tilde{O}(2^{n/2})$ time but requires $O(n2^{n/2})$ superposition queries. The Q2 attack in [25] optimizes superposition queries to $O(n)$ while maintaining the same $\tilde{O}(2^{n/2})$ time. However, all existing quantum attacks suffer from a $O(n^2)$ qubit cost.

Our distributed attacks excel in resource scalability via the parameter t. In both the Q1 and Q2 models, the per-node qubit requirement is lower than that

of previous non-distributed quantum attacks [9, 25]. The Q2 attack reduces the per-node quantum queries from $O(n)$ in [25] to $O(n-t)$, with comparable time complexity.

Notably, our time complexity is fully comparable in the Q1 model, and remains competitive with existing attacks in the Q2 model for practical choices of t, making our approach favorable for distributed quantum implementations with limited per-node resources.

6 Conclusion

This paper presents a distributed Simon's algorithm with $2^t + 1$ nodes by constructing an XOR-type periodic function, which outperforms the standard version in both the per-node qubit requirement and the per-node quantum query complexity. Compared with previous distributed works with $2^t + 2$ nodes by constructing concatenation-type periodic functions, we reduce the per-node qubit requirement from exponential to polynomial with respect to t, while maintaining the same per-node quantum query complexity. Given the limited number of available qubits in current quantum hardware, our distributed algorithm makes it feasible to tackle larger-scale problems that cannot be solved on a single quantum device. Based on this result, we formalize the distributed offline Simon's problem and propose a dedicated quantum algorithm for its special case with (multi-)XOR-type conditional periodic functions, which can be applied to key-recovery attacks on the SoEM22 construction (an instantiation of PRF using two public parallel permutations). Compared with existing non-distributed quantum attacks, our method requires notably fewer qubits per node, while achieving comparable time complexity in both the Q1 and Q2 models.

In the future, two directions are worthy of further investigation. First, designing other distributed quantum algorithms (e.g., distributed BV algorithm [35], distributed Kuperberg's algorithm [31]) with lower per-node qubit overhead based on our methods, to further address the qubit resource constraint of current quantum devices. Second, extending our framework to symmetric-key constructions (e.g., Feistel [6], FX [13], and Lai-Massey [27] constructions) to analyze their quantum vulnerabilities, providing comprehensive support for post-quantum security evaluation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. U25B2014, 62372048, 62272056).

References

1. Beals, R., Brierley, S., Gray, O., Harrow, A.W., Kutin, S., Linden, N., Shepherd, D., Stather, M.: Efficient distributed quantum computing. Proceedings of the Royal

Society A: Mathematical, Physical and Engineering Sciences **469**(2153), 20120686 (2013)
2. Bonnetain, X.: Tight bounds for simon's algorithm. In: Progress in Cryptology–LATINCRYPT 2021: 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6–8, 2021, Proceedings 7. pp. 3–23. Springer (2021)
3. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: the offline simons algorithm. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 552–583. Springer (2019)
4. Bonnetain, X., Jaques, S.: Quantum period finding against symmetric primitives in practice. IACR Cryptol. ePrint Arch. **2020**, 1418 (2020), https://api.semanticscholar.org/CorpusID:226955710
5. Chen, Y.L., Lambooij, E., Mennink, B.: How to build pseudorandom functions from public random permutations. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I 39. pp. 266–293. Springer (2019)
6. Dong, X., Wang, X.: Quantum key-recovery attack on feistel structures. Science China Information Sciences **61**(10), 102501 (2018)
7. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. Journal of cryptology **10**, 151–161 (1997)
8. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
9. Guo, T., Wang, P., Hu, L., Ye, D.: Quantum attacks on prfs based on public random permutations. In: International Conference on Cryptology in India. pp. 566–591. Springer (2022)
10. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Physical review letters **103**(15), 150502 (2009)
11. Jiang, J.R., Wang, T.Y., Huang, W.H., Zhang, J.Z.: Distributed shors algorithm with sequential quantum teleportation. In: 2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE). pp. 479–484. IEEE (2023)
12. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36. pp. 207–237. Springer (2016)
13. Kilian, J., Rogaway, P.: How to protect des against exhaustive key search. In: Advances in Cryptology-CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16. pp. 252–267. Springer (1996)
14. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: 2010 IEEE international symposium on information theory. pp. 2682–2685. IEEE (2010)
15. Kuwakado, H., Morii, M.: Security on the quantum-type even-mansour cipher. In: 2012 international symposium on information theory and its applications. pp. 312–316. IEEE (2012)
16. Li, L., Li, J., Song, Y., Qin, S., Wen, Q., Gao, F.: An efficient quantum proactive incremental learning algorithm. Science China Physics, Mechanics & Astronomy **68**(1), 1–9 (2025)

17. Liu, H.L., Wu, Y.S., Wan, L.C., Pan, S.J., Qin, S.J., Gao, F., Wen, Q.Y.: Variational quantum algorithm for the poisson equation. Physical Review A **104**(2), 022418 (2021)
18. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nature physics **10**(9), 631–633 (2014)
19. Pan, S.J., Wan, L.C., Liu, H.L., Wang, Q.L., Qin, S.J., Wen, Q.Y., Gao, F.: Improved quantum algorithm for a-optimal projection. Physical Review A **102**(5), 052402 (2020)
20. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
21. Simon, D.R.: On the power of quantum computation. SIAM journal on computing **26**(5), 1474–1483 (1997)
22. Song, Y., Wu, Y., Wu, S., Li, D., Wen, Q., Qin, S., Gao, F.: A quantum federated learning framework for classical clients. Science China Physics, Mechanics & Astronomy **67**(5), 250311 (2024)
23. Su, J., Fan, J., Wu, S., Li, G., Qin, S., Gao, F.: Topology-driven quantum architecture search framework. Science China Information Sciences **68**(8), 180507 (2025)
24. Sun, H.W., Cai, B.B., Qin, S.J., Wen, Q.Y., Gao, F.: Quantum attacks on beyond-birthday-bound macs. Physica A: Statistical Mechanics and its Applications **625**, 129047 (2023)
25. Sun, H.W., Gao, F., Xu, R.X., Li, D.D., Li, Z.Q., Zhang, K.J.: Quantum key-recovery attacks on permutation-based pseudorandom functions. IEEE Internet of Things Journal (2025)
26. Tan, J., Xiao, L., Qiu, D., Luo, L., Mateus, P.: Distributed quantum algorithm for simon's problem. Physical Review A **106**(3), 032417 (2022)
27. Vaudenay, S.: On the lai-massey scheme. In: Advances in Cryptology-ASIACRYPT'99: International Conference on the Theory and Application of Cryptology and Information Security, Singapore, November 14-18, 1999. Proceedings. pp. 8–19. Springer (1999)
28. Wan, L.C., Yu, C.H., Pan, S.J., Gao, F., Wen, Q.Y., Qin, S.J.: Asymptotic quantum algorithm for the toeplitz systems. Physical Review A **97**(6), 062322 (2018)
29. Wu, S., Li, R., Song, Y., Qin, S., Wen, Q., Gao, F.: Quantum assisted hierarchical fuzzy neural network for image classification. IEEE Transactions on Fuzzy Systems (2024)
30. Xiao, L., Qiu, D., Luo, L., Mateus, P.: Distributed shor's algorithm. arXiv preprint arXiv:2207.05976 (2022)
31. Yang, P., Zhang, X., Lin, S.: Distributed quantum algorithm for the dihedral hidden subgroup problem. arXiv preprint arXiv:2503.06478 (2025)
32. Yimsiriwattana, A., Lomonaco Jr, S.J.: Distributed quantum computing: A distributed shor algorithm. In: Quantum information and computation II. vol. 5436, pp. 360–372. SPIE (2004)
33. Yu, C.H., Gao, F., Lin, S., Wang, J.: Quantum data compression by principal component analysis. Quantum Information Processing **18**(8), 249 (2019)
34. Zhandry, M.: How to construct quantum random functions. Journal of the ACM (JACM) **68**(5), 1–43 (2021)
35. Zhou, X., Qiu, D., Lou, L.: Distributed exact quantum algorithms for bernstein-vazirani and search problems. arXiv preprint arXiv:2303.10670 (2023)
36. Zhou, X., Xu, X., Zheng, S., Luo, L.: Distributed exact generalized grovers algorithm. Frontiers of Computer Science **20**(7), 2007905 (2026)