IACR Transactions
ISSN XXXX-XXXX, Vol. 0, No. 0, pp. 1–16. DOI:XXXXXXXX

Post-Quantum Security of the Sum of Even-Mansour

YanJin Tan[1], JunTao Gao[1] and XueLian Li[1]

Xidian University, Xi'an, China
24011211360@stu.xidian.edu.cn ,
jtgao@mail.xidian.edu.cn ,
xlli@mail.xidian.edu.cn

Abstract. The Sum of Even-Mansour (SoEM) construction was proposed by Chen et al. at Crypto 2019. This construction implements a pseudorandom permutation via the modular addition of two independent Even-Mansour structures and can spawn multiple variants by altering the number of permutations or keys. It has become the design basis for some symmetric schemes, such as the nonce-based encryption scheme CENCPP* and the nonce-based message authentication code scheme nEHTm. This paper provides a proof of the quantum security of the SoEM21 construction under the Q1 model: when an attacker has quantum access to the random permutations but only classical access to the keyed construction, the SoEM21 construction ensures security of up to $n/3$ bits. This exactly matches the complexity $O(2^{n/3})$ of the quantum key recovery attack in the Q1 model recently proposed by Li et al., thus establishing a tight bound.

Keywords: Post-Quantum Cryptography · Sum of Even-Mansour · Q1 Quantum Security · Tight Security Bound

1 Introduction

The development of quantum computers and quantum algorithms has profoundly impacted cryptography. Over the past decade, with the rapid advancement of quantum computing, proving the security of cryptographic algorithms or constructions in a quantum computing environment has become particularly important. Early foundational work was done by Boneh et al. [BDF+11], who first systematically pointed out that if a quantum adversary can make quantum superposition queries to a random oracle, the security proofs in the classical Random Oracle Model (ROM) would no longer hold. This insight led to the formal proposal of the Quantum Random Oracle Model (QROM), establishing a new foundational framework for post-quantum provable security. Zhandry [Zha13], by proposing the security model for "Quantum-secure Pseudorandom Functions (QPRF)", proved that classical constructions like GGM remain secure under this model, thereby extending the security of this core cryptographic primitive to the quantum query scenario and laying the theoretical foundation for building more complex post-quantum cryptographic protocols. Subsequently, Zhandry [Zha19], by introducing compressed oracle techniques, constructed a dynamically updatable quantum state simulation mechanism in the Fourier domain. This mechanism, for the first time, effectively recorded the adversary's quantum query history without being detected, thereby overcoming the "recording barrier" in the QROM and establishing the cornerstone for crucial proofs such as the quantum indistinguishability of hash functions. Meanwhile, the quantum extraction and rewinding techniques developed by Unruh in the study of quantum zero-knowledge proofs [Unr12], and the One-way to Hiding (O2H) Lemma he proposed [Unr15], provided key tools for quantifying the

differences introduced by randomized operations in quantum query scenarios and have become one of the core techniques in post-quantum security proofs. The measure-and-reprogram framework further formalized by Don et al. [DFM20] enabled the migration of many classical "reprogramming"-based security proofs to quantum settings. Grilo et al. [GHHM21] further achieved tight adaptive reprogramming in the QROM, providing more concise and tighter security proofs for several key constructions in post-quantum cryptography.

Based on the gradual refinement of these fundamental theories and technical tools, three mainstream analytical approaches for quantum security proofs have gradually formed:

1. **QPRF-based Substitution Method:** Replace the random oracle with an instance of a quantum-secure pseudorandom function and prove their indistinguishability under the adversary's quantum access capability within the reduction.

2. **Compressed Oracle-based Amplitude Analysis:** Quantify the adversary's query amplitude for specific points to safely resample or rewrite oracle responses in a controlled manner.

3. **Measure-and-Reprogram-based Adaptive Reprogramming Technique:** Safely reprogram points that the adversary queries with high probability, under the premise of ensuring controllable perturbation probability.

These methods have been continuously improved and integrated, evolving into a general proof framework capable of proving the security of the Fiat-Shamir transform, digital signature schemes, Even-Mansour-like constructions, and various symmetric cryptographic structures under the Q1 or Q2 models.

Pseudorandom functions based on public permutations are widely used in various cryptographic scheme designs due to their simple structure and clear security evaluation, making them important candidate directions for cryptographic constructions in the post-quantum era. The Sum of Even-Mansour (SoEM) construction was proposed by Chen et al. [CLM19] at Crypto 2019. As a typical class of unkeyed pseudorandom functions based on public permutations, it has become the core foundational construction for schemes like the nonce-based encryption scheme CENCPP* [BDLN22] and the nonce-based message authentication code scheme nEHTm [CLLL20]. The classical security of the SoEM21 construction has been thoroughly studied, with existing results indicating its classical security bound is at the birthday bound, requiring an attacker to have a query complexity on the order of $2^{n/2}$ to effectively distinguish the construction from an ideal random function.

With the development of quantum cryptanalysis, evaluating the security of cryptographic constructions under quantum models has become a new research focus. In quantum attack models, adversarial capabilities are typically divided into two categories: Q1 and Q2 [Zha12]. In the Q2 model, the attacker can make quantum queries simultaneously to the underlying public permutation and the keyed construction. This means the adversary can execute its own customized quantum superposition queries, such as applying the unitary operator $U_k : |x\rangle|y\rangle \to |x\rangle|y \oplus E_k(x)\rangle$. Many symmetric cryptosystems have already been proven insecure in the Q2 model, i.e., polynomial-complexity attacks can be achieved using quantum computers [KM10, KM12, KLLNP16, LM17]. In principle, one can always transform the circuit of any classical function f into a reversible quantum circuit U_f. Therefore, in a cryptographic context, it is reasonable and necessary to consider that an attacker can use U_f if they know the circuit of function f. However, this premise changes fundamentally in scenarios involving symmetric keys. If f is a key-dependent, keyed function E_k, the only realistic way for an attacker to gain quantum access to it is if there exists an explicit quantum interface provided by the honest party. Yet, in the vast majority of real-world applications, the honest party using the keyed function E_k (e.g., a

server or Hardware Security Module) implements it only via classical computers, accepting only classical inputs and returning classical outputs. Even if they were to run E_k on a quantum computer in the future, there would be no reason to expose a quantum interface beyond classical queries to external attackers.

Therefore, the **Q1 model** is more aligned with actual deployment scenarios. In this model, the attacker can make quantum queries to the public, keyless components, but can only make classical queries to the keyed construction E_k. This is because, in the absence of a dedicated quantum interface, although the general algorithm for E is public, the attacker cannot build an effective quantum query circuit U_{E_k} for a specific instance E_k with an unknown key k. Once large-scale fault-tolerant quantum computers become available, attacks in the Q1 model will pose the most immediate and realistic threat to today's deployed cryptographic systems that offer only classical interfaces. Therefore, systematically evaluating the security of cryptosystems under the Q1 model and exploring their security boundaries has become a core issue in post-quantum cryptography research in recent years. This concerns not only theoretical security but also directly determines how we should select, deploy, and parameterize real-world systems that rely on symmetric cryptographic primitives in the post-quantum era.

However, from the perspective of security proofs, the Q1 model presents unique theoretical challenges compared to the Q2 model. In the Q2 model, the homogeneity of the adversary's capabilities (both accessible via quantum queries) makes many classical secure constructions directly vulnerable to destructive attacks using quantum algorithms like Simon's or Grover's [Gro96, Sim97], often rendering their original security proofs invalid or requiring complete redesign. In contrast, in the more realistic Q1 model, although the adversary's capabilities are restricted, the core difficulty lies in the adversary possessing an **asymmetric, hybrid access capability**: it can make unlimited quantum superposition queries to the underlying public permutations while only obtaining sparse point-wise information from the classical interface of the keyed construction. This pattern prevents standard quantum algorithms from being directly applied and also forces security reduction proofs to handle the complex, adaptive interdependencies between the two types of queries. To constrain any inconsistencies the adversary might discover through interleaved queries, fine-grained analytical tools such as reprogramming lemmas and resampling lemmas must be developed and applied. Therefore, establishing tight security bounds for the Q1 model is not only crucial for assessing the post-quantum security of actually deployed cryptosystems but also advances the development of foundational proof techniques in post-quantum cryptography.

In this direction, the cryptographic community has made a series of important advances, laying a solid foundation for the Q1 security analysis of similar constructions. Jaeger et al. [JST21] first proved a Q1 security lower bound for the FX construction, showing it guarantees $(k+n)/3$-bit security under non-adaptive attacks. This matches the attack conclusion on the FX construction by Bonnetain et al. [BHNPS19], thus being tight. Alagic et al. [ABKM22a] subsequently proved that the Even-Mansour construction guarantees $n/3$-bit post-quantum Q1 security under adaptive attacks, while the offline Simon's algorithm attack results on the Even-Mansour construction by Bonnetain et al. [BHNPS19] show this bound is tight. Following this, Alagic et al. [ABKM22b] proved the security of the Tweakable Even-Mansour construction under the Q1 model, showing it also guarantees $n/3$-bit post-quantum Q1 security under adaptive attacks. Guo Chun et al. [GHY24] further extended the security proof of the FX construction to the adaptive chosen-plaintext-and-ciphertext attack scenario. Chen et al. [CEM25] systematically studied and proved the security of 2-round and t-round Key-Alternating Ciphers (t-KAC) under the Q1 model. However, for the SoEM construction proposed by Chen et al. [CLM19] in 2019, security analysis under the Q1 model remains a gap. Although Shinagawa et al. [SI22] achieved polynomial-time key recovery against SoEM21 in the strongly adversarial Q2 model using

Simon's algorithm, recent work by Li et al. [LFG+25] indicates that in the more realistic Q1 model, the complexity of the key recovery attack on SoEM21 based on the offline Simon's algorithm is $O(2^{n/3})$, suggesting its security lower bound might be $n/3$ bits, but a rigorous security proof has not yet been provided.

1.1 Contributions of This Paper

As mentioned above, a security proof for the SoEM construction under the more realistic and threatening Q1 model is still lacking, and the optimal attack complexity $O(2^{n/3})$ proposed by Li et al. [LFG+25] also requires a matching security lower bound to confirm its tightness. To fill this critical gap, this paper conducts a systematic study of the SoEM21 construction, aiming to establish a complete and tight Q1 security foundation for it. The main contributions of this paper are as follows:

1. **Established a complete and tight Q1 security bound for SoEM21, unifying security proof and attack reduction.** The core contribution of this paper is that we, for the first time, rigorously prove that the SoEM21 construction can provide $n/3$-bit security under the Q1 model, thereby filling the theoretical gap in the post-quantum Q1 security proof for this construction. Our proof systematically verifies the construction's exponential resistance to Simon's algorithm. Crucially, the proven security lower bound exactly matches the upper bound $O(2^{n/3})$ of the known optimal attack. This not only confirms the tightness of the security bound but also implies that in the Q1 model, no attacker can exceed this complexity limit, providing a reliable security conclusion for the practical deployment of SoEM21.

2. **Developed quantum lower-bound proof techniques for composite permutation constructions.** Compared to basic single-permutation constructions like Even-Mansour, SoEM21 involves the parallel invocation of two permutations and the superposition of keys, making its security proof more complex. By defining a combined function and comprehensively applying the **Reprogramming Lemma** and the **Resampling Lemma**, this paper successfully addresses the challenges posed by the adversary's interleaved quantum queries on multiple public permutations. This proof framework exhibits good generality and can serve as an important technical reference for the subsequent analysis of other similar "multi-branch" or "summation-type" cryptographic constructions.

3. **Clarified the security characteristics of the SoEM structure, providing guidance for post-quantum cryptographic design.** Our analysis clearly demonstrates that the security of SoEM21 does not rely on the publicity or internal structure of the underlying permutations but is entirely guaranteed by the key length and the logic of the construction itself. This "permutation-independent" security characteristic further strengthens the candidacy of such permutation-based constructions in the post-quantum cryptography standardization process and provides direct theoretical support for the security evaluation of schemes based on the SoEM idea, such as CENCPP* and nEHTm.

2 Preliminaries

2.1 Notation and Model

Let n be the security parameter. All operations in this paper are considered over $\{0,1\}^n$. Let $\mathcal{P}_n$ denote the set of all permutations from $\{0,1\}^n$ to $\{0,1\}^n$. We uniformly and independently sample public permutations P_1 and P_2 from $\mathcal{P}_n$. The key k is a bit string uniformly sampled from $\{0,1\}^n$.

SoEM21 Construction: Defined as a function $E_k : \{0,1\}^n \to \{0,1\}^n$, computed as follows:

$$E_k(x) = P_1(x \oplus k) \oplus P_2(x \oplus k) \oplus k.$$

We consider the adversarial capability of an adversary $\mathcal{A}$ in the **Q1 model**: $\mathcal{A}$ can make **quantum queries** to the public permutations P_1 and P_2 (and their inverses). This means $\mathcal{A}$ can apply the unitary operator $U_P : |x\rangle|y\rangle \to |x\rangle|y \oplus P(x)\rangle$ and its inverse. $\mathcal{A}$ can only make **classical queries** to the keyed construction E_k. That is, $\mathcal{A}$ provides a classical input x and receives the classical output $E_k(x)$.

We denote the number of classical queries made by adversary $\mathcal{A}$ as q_E, and the total number of quantum queries (the sum of queries to P_1 and P_2) as q_P.

2.2 Security Definition

Our goal is to prove that SoEM21 is a pseudorandom permutation (PRP). We formalize its security through a distinguishing game. When proving the security of a Pseudorandom Function (PRF), the reduction game is typically designed not to allow inverse queries. This is because the security definition of a PRF itself does not consider the computability of the inverse operation but focuses on the indistinguishability of the function's output from a random function. Inverse queries introduce capabilities that do not align with the PRF security goal and may break the validity of the reduction proof. Therefore, we assume the adversary only has the ability for forward queries $E_k(x) = P_1(x \oplus k) \oplus P_2(x \oplus k) \oplus k$.

Definition 1 (PRP Security Game for SoEM21 in the Q1 Model)**.** The game involves a challenger and a quantum adversary $\mathcal{A}$. The challenger flips a uniform random bit b.

- If $b = 0$ (**Real World**): The challenger randomly selects a key $k \leftarrow \{0,1\}^n$ and permutations $P_1, P_2 \leftarrow \mathcal{P}_n$. The adversary $\mathcal{A}$ is granted **quantum query access** to $P_1, P_2, P_1^{-1}, P_2^{-1}$ and **classical query access** to $E_k(\cdot) = P_1(\cdot \oplus k) \oplus P_2(\cdot \oplus k) \oplus k$.
- If $b = 1$ (**Ideal World**): The challenger randomly selects a random permutation $R \leftarrow \mathcal{P}_n$ with the same domain and range as E_k, as well as $P_1, P_2 \leftarrow \mathcal{P}_n$. The adversary $\mathcal{A}$ is granted **quantum query access** to $P_1, P_2, P_1^{-1}, P_2^{-1}$ and **classical query access** to $R(\cdot)$.

At the end of the game, the adversary $\mathcal{A}$ outputs a guess bit b'. The advantage of adversary $\mathcal{A}$ in this game is defined as:

$$\mathrm{Adv}^{\text{Q1-PRP}}_{\text{SoEM21}}(\mathcal{A}) = \left|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\right|.$$

More generally, for any adversary making at most q_E classical queries and q_P quantum queries, we define:

$$\mathrm{Adv}^{\text{Q1-PRP}}_{\text{SoEM21}}(q_E, q_P) = \max_{\mathcal{A}} \mathrm{Adv}^{\text{Q1-PRP}}_{\text{SoEM21}}(\mathcal{A}).$$

The core objective of this paper is to prove the following theorem:

Theorem 1 (Main Theorem)**.** *For any adversary $\mathcal{A}$ in the Q1 model making q_E classical queries and q_P quantum queries, it holds that:*

$$Adv^{Q1\text{-}PRP}_{SoEM21}(q_E, q_P) \leq 12 \cdot \frac{q_E\sqrt{q_P} + q_P\sqrt{q_E}}{2^{n/2}}.$$

In particular, when both q_E and q_P are far less than $2^{n/3}$, the adversary's advantage is negligible. This proves that SoEM21 has $n/3$-bit security under the Q1 model.

We now introduce Alagic et al.'s Reprogramming Lemma and Resampling Lemma [ABKM22a, ABKM22b].

2.3 Reprogramming Lemma

The Reprogramming Lemma applies to the following experiment: A distinguisher D chooses an arbitrary function F and a randomized procedure B that determines a set B of points where F may be reprogrammed to some known value. Then, D is given quantum access to either F or a reprogrammed version of F; after it finishes its oracle queries, D is given B. The lemma states that as long as no point is reprogrammed with high probability, D cannot determine whether it interacted with F or its reprogrammed version.

For a function $F : \{0,1\}^m \to \{0,1\}^n$ and a set $B \subset \{0,1\}^m \times \{0,1\}^n$ such that each $x \in \{0,1\}^m$ is the first element of at most one tuple in B, define

$$F^{(B)}(x) = \begin{cases} y & \text{if } (x,y) \in B \\ F(x) & \text{otherwise} \end{cases}.$$

Lemma 1 (Reprogramming Lemma)**.** *[ABKM22a] Let D be a quantum distinguisher in the following experiment:*

Phase 1: *D outputs a description of a function $F_0 = F : \{0,1\}^m \to \{0,1\}^n$ and a description of a randomized algorithm B whose output is a set $B \subset \{0,1\}^m \times \{0,1\}^n$, where each $x \in \{0,1\}^m$ is the first element of at most one tuple in B. Let $B_1 = \{x \mid \exists y : (x,y) \in B\}$ and*

$$\epsilon = \max_{x \in \{0,1\}^m} \{\Pr[x \in B_1]\}.$$

Phase 2: *Run B to obtain B. Let $F_1 = F^{(B)}$. Choose a uniform random bit b, and D is given quantum access to F_b.*

Phase 3: *D loses access to F_b and receives the randomness r used to invoke B in Phase 2. Then D outputs a guess b'.*

For any D that makes at most q queries to its oracle in expectation,

$$|\Pr[D \text{ outputs } 1 \mid b = 1] - \Pr[D \text{ outputs } 1 \mid b = 0]| \leq 2q \cdot \sqrt{\epsilon}.$$

2.4 Resampling Lemma

The Resampling Lemma applies to the following experiment: First, a distinguisher D is given quantum access to a random permutation P. Then, in a second phase, P may be reprogrammed so that its value at a single uniform point s is changed to an independent uniform value. Since the distribution of $P(s)$ is identical before and after any such reprogramming, we call it "resampling". D's goal is to determine whether its oracle was resampled. That is, D can tell only if it placed significant amplitude on s in a query during the first phase, even if given s and continued oracle access in the second phase.

Given a permutation $P : \{0,1\}^n \to \{0,1\}^n$ and $s, y \in \{0,1\}^n$, define the reprogrammed function $P_{s \to y} : \{0,1\}^n \to \{0,1\}^n$ as

$$P_{s \to y}(w) = \begin{cases} y & \text{if } w = s \\ P(w) & \text{otherwise} \end{cases}.$$

Lemma 2 (Resampling Lemma)**.** *[ABKM22a] Let D be a quantum distinguisher in the following two-phase experiment:*

Phase 1: *D is given quantum access to a uniform random permutation $P_0 = P$ and its inverse $P_0^{-1} = P^{-1}$.*

Phase 2: *Uniformly choose $s_0, s_1 \in \{0,1\}^n$. Define P_1: if $b = 0$, then $P_1 = P$; if $b = 1$, then $P_1 = P \circ swap_{s_0,s_1}$, where $swap_{s_0,s_1}$ swaps the values at s_0 and s_1. D receives s_0, s_1 and is given quantum access to P_1, P_1^{-1}. Finally, D outputs a guess bit b'.*

For D making at most q queries to P_0, P_0^{-1} in Phase 1, we have:

$$|\Pr[D \text{ outputs } b' = 0] - \Pr[D \text{ outputs } b' = 1]| \leq 4\sqrt{\frac{q}{2^n}}.$$

3 Post-Quantum Security of the SoEM21 Construction

We now establish the post-quantum security of the SoEM21 construction based on the lemmas from the previous section. Recall that the SoEM21 construction is defined as: $E_k(x) = P_1(x \oplus k) \oplus P_2(x \oplus k) \oplus k$, where $P_1, P_2 : \{0,1\}^n \to \{0,1\}^n$ are independent public random permutations, and $k \in \{0,1\}^n$ is a uniform key. Our proof considers an adversary $\mathcal{A}$ who can make classical queries to E_k (and its inverse, if applicable) and quantum queries to P_1, P_2 (and their inverses). $\mathcal{A}$'s goal is to distinguish the real world (interacting with $E_k[P_1, P_2]$ and P_1, P_2) from the ideal world (interacting with an independent random permutation R and P_1, P_2). In the following, we let $\mathcal{P}_n$ denote the set of all permutations over $\{0,1\}^n$. We write $E_k[P_1, P_2]$ to emphasize the dependence on P_1, P_2. Our main result is as follows:

Theorem 2 (Q1 Security of SoEM21)**.** *Let $\mathcal{A}$ be an adversary making q_E classical queries to its first oracle and q_P quantum queries to its second oracle (i.e., to P_1 and P_2). Then*

$$\begin{aligned} Adv^{Q1\text{-}PRP}_{SoEM21}(\mathcal{A}) &= \left| \Pr_{\substack{k \leftarrow \{0,1\}^n \\ P_1, P_2 \leftarrow \mathcal{P}_n}} \left[\mathcal{A}^{E_k[P_1,P_2], P_1, P_2}(1^n) = 1 \right] - \Pr_{\substack{R \leftarrow \mathcal{P}_n \\ P_1, P_2 \leftarrow \mathcal{P}_n}} \left[\mathcal{A}^{R, P_1, P_2}(1^n) = 1 \right] \right| \\ &\leq 12 \cdot 2^{-n/2} \left(q_E \sqrt{q_P} + q_P \sqrt{q_E} \right). \end{aligned}$$

Proof. We now use the lemmas from the Section 2 to prove the post-quantum security of the SoEM21 construction. We divide an execution of $\mathcal{A}$ into $q_E + 1$ phases, labeled $0, \ldots, q_E$, where phase j corresponds to the time interval between $\mathcal{A}$'s j-th and $(j+1)$-th classical queries. Specifically, phase 0 corresponds to the period before $\mathcal{A}$ makes its first classical query, and phase q_E corresponds to the period after $\mathcal{A}$ makes its last classical query. We allow $\mathcal{A}$ to adaptively distribute its q_P quantum queries among these phases arbitrarily. Let $q_{P,j}$ denote the expected number of queries $\mathcal{A}$ makes in phase j in the ideal world $\mathcal{A}^{R,P_1,P_2}$; note that $\sum_{j=0}^{q_E} q_{P,j} = q_P$.

Denote $\mathcal{A}$'s i-th classical query as (x_i, y_i), where y_i is the response. Let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ denote the ordered list describing $\mathcal{A}$'s first j classical queries. We use $\prod$ to denote sequential composition of operations, i.e., $\prod_{i=1}^{n} f_i = f_n \circ \cdots \circ f_1$. The operation $\mathrm{swap}_{a,b}$ swaps the values at points a and b. Without loss of generality, we assume algorithm $\mathcal{A}$ never makes redundant classical queries; that is, once $\mathcal{A}$ obtains an input-output pair (x, y) through a classical query, it does not submit query x to that oracle again. In the sequence of hybrid experiments we define, we use reprogramming techniques to ensure that in phase 2, the responses of the keyed classical oracle for points already queried in phase 1 are consistent with the random oracle responses from phase 1. Therefore, the adversary cannot distinguish the experiments by repeatedly querying points from phase 1.

Now we reprogram P_1, P_2. Define $a_i = x_i \oplus k$. We first reprogram P_1. For P_1: For each $i \in \{1, 2, \cdots, j\}$, sequentially choose random values $u_i \in \{0,1\}^n$. If $P_1(a_i) \neq u_i$, find a point z_i such that $P_1(z_i) = u_i$, and swap the values of P_1 at a_i and z_i. After swapping, $P_{1,T_j,k}(a_i) = u_i$. Take u_i as some random value, ensuring all u_i are distinct.

$$S_{T_j,P_1,k} = \prod_{i=1}^{j} \mathrm{swap}_{a_i,z_i}, \quad S_{T_j,P_1,k} \circ P_1 = P_{1,T_j,k}.$$

Next, reprogram P_2. For P_2: Set the target value $t_i = k \oplus u_i \oplus y_i$. If $P_2(a_i) \neq t_i$, find a point w_i such that $P_2(w_i) = t_i$, and swap the values of P_2 at a_i and w_i. After swapping, $P_{2,T_j,k}(a_i) = t_i$.

$$S_{T_j,P_2,k} = \prod_{i=1}^{j} \mathrm{swap}_{a_i,w_i}, \quad S_{T_j,P_2,k} \circ P_2 = P_{2,T_j,k}.$$

At this point we have:

$$P_{1,T_j,k}(x_i \oplus k) = u_i, \quad P_{2,T_j,k}(x_i \oplus k) = u_i \oplus k \oplus y_i, \quad P_{1,T_j,k}(x_i \oplus k) \oplus P_{2,T_j,k}(x_i \oplus k) = y_i \oplus k.$$

Thus, we have $P_{1,T_j,k}(a_i) \oplus P_{2,T_j,k}(a_i) \oplus k = y_i$, satisfying the requirement.

We now define hybrid experiments H_j for $j = 0, 1, \cdots, q_E$. □

Experiment H_j: Phase 1: The distinguisher $\mathcal{D}$ samples $P_1, P_2, R \leftarrow \mathcal{P}_n$. It then runs $\mathcal{A}$, using P_1, P_2 to answer its quantum queries and using R to answer classical queries, until responding to $\mathcal{A}$'s $(j+1)$-th classical query. Let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ be the list of classical query-response pairs.

Phase 2: For the remaining execution of $\mathcal{A}$, use $E_k[P_{1,T_j,k}, P_{2,T_j,k}]$ to answer its classical queries, and use $P_{1,T_j,k}, P_{2,T_j,k}$ to answer its quantum queries.

We now define hybrid experiments H'_j for $j = 0, 1, \cdots, q_E - 1$.

Experiment H'_j: Phase 1: $\mathcal{D}$ samples $P_1, P_2, R \leftarrow \mathcal{P}_n$. It then runs $\mathcal{A}$, using P_1, P_2 to answer its quantum queries and using R to answer classical queries, until the conclusion of $\mathcal{A}$'s $(j+1)$-th classical query. Let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ be the list of classical query-response pairs.

Phase 2: For the remaining execution of $\mathcal{A}$, use $E_k[P_{1,T_j,k}, P_{2,T_j,k}]$ to answer its classical queries, and use $P_{1,T_j,k}, P_{2,T_j,k}$ to answer its quantum queries.

In subsequent Lemma 3 and Lemma 4, we establish bounds on the distinguishability between H_j and H'_j and between H'_j and H_{j+1}. For $0 \le j < q_E$ we have:

$$\left|\Pr[\mathcal{A}(\mathrm{H}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}'_j) = 1]\right| \le 8\sqrt{\frac{q_P}{2^n}} + \frac{3j}{2^n} + \frac{2q_E}{2^n}, \tag{1}$$

$$\left|\Pr[\mathcal{A}(\mathrm{H}'_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}_{j+1}) = 1]\right| \le 2 \cdot q_{P,j+1}\sqrt{2(j+1)/2^n}. \tag{2}$$

Then we have:

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathrm{H}_0) = 1] - \Pr[\mathcal{A}(\mathrm{H}_{q_E}) = 1]| \\
\le\ & \sum_{j=0}^{q_E-1} \left[\left|\Pr[\mathcal{A}(\mathrm{H}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}'_j) = 1]\right| + \left|\Pr[\mathcal{A}(\mathrm{H}'_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}_{j+1}) = 1]\right|\right] \\
\le\ & \sum_{j=0}^{q_E-1} \left[8\sqrt{\frac{q_P}{2^n}} + \frac{3j}{2^n} + \frac{2q_E}{2^n} + 2 \cdot q_{P,j+1}\sqrt{\frac{2(j+1)}{2^n}}\right] \\
\le\ & \sum_{j=0}^{q_E-1} \left(8\sqrt{\frac{q_P}{2^n}} + \frac{3j}{2^n} + \frac{2q_E}{2^n} + 2 \cdot q_{P,j+1}\sqrt{\frac{2(j+1)}{2^n}}\right) \\
\le\ & 2 \cdot q_E \cdot (q_E - 1) \cdot 2^{-n} + \frac{3 \cdot q_E \cdot (q_E - 1)}{2} \cdot 2^{-n} + \sum_{j=0}^{q_E-1} \left(8\sqrt{\frac{q_P}{2^n}} + 2 \cdot q_{P,j+1}\sqrt{\frac{2q_E}{2^n}}\right)
\end{aligned}$$

Next, we proceed to further simplify the bound obtained. Note that if $q_P = 0$, the adversary makes no quantum queries, and hence E_k and R are perfectly indistinguishable; in this case, the theorem holds trivially. Therefore, without loss of generality, we assume $q_P \ge 1$ in the following. Therefore, we can assume $q_P \ge 1$. We can also assume $q_E < 2^{n/2}$, otherwise the bound would exceed 1. Under these assumptions, we have $2^{-n}q_E^2 \le 2^{-n/2}q_E$ and $2^{-n/2}q_E\sqrt{q_P} \le 2^{-n/2}q_P\sqrt{q_E}$. Thus,

$$\begin{aligned}
|\Pr[\mathcal{A}(\mathrm{H}_0) = 1] - \Pr[\mathcal{A}(\mathrm{H}_{q_E}) = 1]| &\le 4 \cdot q_E\sqrt{q_P} \cdot 2^{-n/2} + 2^{-n/2}\left(8q_E\sqrt{q_P} + 3q_P\sqrt{q_E}\right) \\
&\le 12 \cdot 2^{-n/2}(q_P\sqrt{q_E} + q_E\sqrt{q_P}).
\end{aligned}$$

A constant-level advantage is achieved when $2^{-n/2}(q_P\sqrt{q_E} + q_E\sqrt{q_E}) \approx 1$, implying $\Omega(2^{n/3})$ queries are required.

To complete the proof of Theorem 2, we now prove the indistinguishability of H'_j and H_{j+1}, and of H_j and H'_j.

Lemma 3. *Let $\mathcal{A}$ be an adversary making at most q_E classical queries and q_P quantum queries. Then for $j = 0, \cdots, q_E - 1$, we have:*

$$\left|\Pr[\mathcal{A}(H'_j) = 1] - \Pr[\mathcal{A}(H_{j+1}) = 1]\right| \leq 2 \cdot q_{P,j+1}\sqrt{2(j+1)/2^n}.$$

Proof. We can write the oracle sequences defined by H'_j and H_{j+1} as follows:

The oracle sequence for H_{j+1} :

$$\underbrace{P_1, P_2, R, P_1, P_2, \ldots, R}_{j+1 \text{ classical queries}}, \quad E_k[P_{1,T_{j+1},k}, P_{2,T_{j+1},k}], P_{1,T_{j+1},k}, P_{2,T_{j+1},k}, \ldots, E_k[P_{1,T_{j+1},k}, P_{2,T_{j+1},k}], P_{1,T_{j+1},k}, P_{2,T_{j+1},k}$$

The oracle sequence for H'_j :

$$\underbrace{P_1, P_2, R, P_1, P_2, \ldots, R}_{j+1 \text{ classical queries}}, \quad E_k[P_{1,T_j,k}, P_{2,T_j,k}], P_{1,T_j,k}, P_{2,T_j,k}, \ldots, E_k[P_{1,T_j,k}, P_{2,T_j,k}], P_{1,T_j,k}, P_{2,T_j,k}$$

We now define a combined function to facilitate the use of the Reprogramming Lemma. Let $F : \{1,2\} \times \{0,1\}^n \to \{0,1\}^n$, where $F(1,x) = P_1(x)$, corresponding to queries to P_1, and $F(2,x) = P_2(x)$, corresponding to queries to P_2. The adversary's quantum queries to P_1 and P_2 are now viewed as queries to F. Therefore, the adversary's total quantum queries to F is q_P (the sum of queries to P_1 and P_2).

In the hybrid experiment, when adding a new classical query point (x_{j+1}, y_{j+1}), we need to reprogram P_1 and P_2 at the point $a_{j+1} = x_{j+1} \oplus k$. The specific steps are: 1. Choose a random value $u_{j+1} \in \{0,1\}^n$. 2. Reprogram P_1 at a_{j+1} to u_{j+1}. 3. Reprogram P_2 at a_{j+1} to $k \oplus u_{j+1} \oplus y_{j+1}$. In the combined function F, this is equivalent to reprogramming two points: $(1, a_{j+1})$ and $(2, a_{j+1})$. The probability that any given point is reprogrammed is: For any fixed point $(s,x) \in \{1,2\} \times \{0,1\}^n$, the probability it is reprogrammed is $\Pr[(s,x) \text{ is reprogrammed}] \leq \frac{1}{2^n}$. This is because a_{j+1} is uniformly random (due to the randomness of k).

Let $\mathcal{A}$ be a distinguisher between H'_j and H_{j+1}. We construct a distinguisher $\mathcal{D}$ for the blinding experiment in Lemma 1:

Phase 1: $\mathcal{D}$ samples $P_1, P_2, R \leftarrow \mathcal{P}_n$. It then runs $\mathcal{A}$, using P_1, P_2 to answer its quantum queries and using R to answer classical queries, until responding to $\mathcal{A}$'s $(j+1)$-th classical query. Let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ be the list of classical query-response pairs. $\mathcal{D}$ defines the combined function $F : \{1,2\} \times \{0,1\}^n \to \{0,1\}^n$, where querying P_1 corresponds to querying $F(1,\cdot)$ and querying P_2 corresponds to querying $F(2,\cdot)$. The adversary's quantum queries to P_1 and P_2 are now viewed as queries to F. Therefore, the adversary's total quantum queries to F is q_P (the sum of queries to P_1 and P_2) such that for all a, x, we have $F^{(B)}(a,x) = P_{a,T_{j+1},k}(x)$.

Phase 2: $\mathcal{D}$ is allowed to generate B, and $\mathcal{D}$ is given quantum access to oracle F_b. $\mathcal{D}$ continues running $\mathcal{A}$, answering its quantum queries via $P_a = F_b(a,\cdot)$. When $\mathcal{A}$ issues the next (i.e., the $(j+2)$-th) classical query, Phase 2 ends.

Phase 3: $\mathcal{D}$ is allowed to use B to generate k. It continues running $\mathcal{A}$, using $E_k[P_{1,T_{j+1},k}, P_{2,T_{j+1},k}]$ to answer its classical queries and using $P_{1,T_{j+1},k}, P_{2,T_{j+1},k}$ to answer its quantum queries. Finally, it outputs whatever $\mathcal{A}$ outputs.

Note that $\mathcal{D}$ is a valid distinguisher for the reprogramming experiment in Lemma 1. Clearly, if $b = 0$ (i.e., the'oracle of $\mathcal{D}$ in Phase 2 is $F_0 = F$), then the' output of $\mathcal{A}$ is distributed identically to its output in H_{j+1}; if $b = 1$ (i.e., the'oracle of $\mathcal{D}$ in Phase 2 is $F_1 = F^{(B)}$), then the' output of $\mathcal{A}$ is distributed identically to its output in H'_j. Therefore, $\left|\Pr[\mathcal{A}(\mathrm{H}'_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}_{j+1}) = 1]\right|$ equals the distinguishing advantage of $\mathcal{D}$ in the reprogramming experiment. To limit this using Lemma 1, we need to limit the probability of reprogramming ϵ and the expected number of queries $\mathcal{D}$ in Phase 2 (when $F = F_0$).

The probability of reprogramming ϵ can be bounded by the definition of $P_{1,T_{j+1},k}$ and the fact that $F^{(B)}(a, x) = P_{a,T_{j+1},k}(x)$. Fixing P_1 and P_2, the probability (over k) that any given (a, x) is reprogrammed is at most the probability it belongs to the set $\{(1, a_i), (1, P_1^{-1}(u_i)), (2, a_i), (2, P_2^{-1}(u_i \oplus y_i))\}_{i=1}^{j+1}$ (where $a_i = x_i \oplus k$). Considering the impact of a single query, just the i-th classical query to P_1 ($1 \le i \le j + 1$) can specify at most 2 input points to be reprogrammed (e.g., a_i and $P_1^{-1}(u_i)$ for a forward query to P_1). So, Pr[point is reprogrammed by the i-th query] $\le$ Pr[point is the 1st target] + Pr[point is the 2nd target] $= \frac{1}{2^n} + \frac{1}{2^n} = \frac{2}{2^n}$. Therefore, Pr[point is reprogrammed] $\le \sum_{i=1}^{j+1} \frac{2}{2^n} = (j + 1) \cdot \frac{2}{2^n} = \frac{2(j+1)}{2^n}$.

When $F = F_0$, the expected number of queries $\mathcal{D}$ makes in Phase 2 equals the expected number of queries $\mathcal{A}$ makes in phase $(j + 1)$ of H_{j+1}. Since H_{j+1} and H'_j are identical up to the completion of phase $(j + 1)$, this is precisely $q_{P,j+1}$. Thus we have:

$$\left|\Pr[\mathcal{A}(\mathrm{H}'_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}_{j+1}) = 1]\right| \le 2 \cdot q_{P,j+1}\sqrt{2(j + 1)/2^n}.$$

□

Lemma 4. *Let $\mathcal{A}$ be an adversary making at most q_E classical queries and q_P quantum queries. Then for $j = 0, \cdots, q_E$, we have:*

$$\left|\Pr[\mathcal{A}(H_j) = 1] - \Pr[\mathcal{A}(H'_j) = 1]\right| \le 8\sqrt{\frac{q_P}{2^n}} + \frac{3j}{2^n} + \frac{2q_E}{2^n}.$$

Proof. To prove Lemma 4, we introduce auxiliary experiments: H^*_j, H^{**}_j, H^{***}_j.

Experiment H^*_j: 1. $\mathcal{D}$ uniformly samples $P_1, P_2, R \leftarrow \mathcal{P}_n$. 2. Run $\mathcal{A}$, using R to answer its classical queries and P_1, P_2 to answer its quantum queries, until $\mathcal{A}$ issues the $(j+1)$-th classical query x_{j+1}. 3. Uniformly sample $s_0, s_1 \in \{0, 1\}^n$, define $k = s_0 \oplus x_{j+1}$. Define $P_1^{(1)}$ as

$$P_1^{(b)}(x) = P_1 \circ \mathrm{swap}_{s_0,s_1}(x)$$

(i.e., apply swap of s_0 and s_1 to P_1). Then continue running $\mathcal{A}$, using $E_k[P^{(1)}_{1,T_j,k}, P_{2,T_j,k}]$ to answer its remaining classical queries (including the $(j + 1)$-th), and using $P^{(1)}_{1,T_j,k}, P_{2,T_j,k}$ to answer its quantum queries, where $P^{(1)}_{1,T_j,k} = S_{T_j,P_1,k} \circ P_1^{(1)}$.

Experiment H^{**}_j: 1. $\mathcal{D}$ uniformly samples $P_1, P_2, R \leftarrow \mathcal{P}_n$. 2. Run $\mathcal{A}$, using R to answer its classical queries and P_1, P_2 to answer its quantum queries, until $\mathcal{A}$ issues the $(j+1)$-th classical query x_{j+1}. 3. Uniformly sample $s_0, s_1 \in \{0, 1\}^n$, define $k = s_0 \oplus x_{j+1}$. Define $P_1^{(1)}$ as

$$P_1^{(b)}(x) = P_1 \circ \mathrm{swap}_{s_0,s_1}(x)$$

and define $P_2^{(1)}$ as

$$P_2^{(b)}(x) = P_2 \circ \mathrm{swap}_{s_0,s_1}(x)$$

.Then continue running $\mathcal{A}$, using $E_k[P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}]$ to answer its remaining classical queries (including the $(j + 1)$-th), and using $P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}$ to answer its quantum queries, where $P^{(1)}_{1,T_j,k} = S_{T_j,P_1,k} \circ P_1^{(1)}$ and similarly for $P^{(1)}_{2,T_j,k}$.

Experiment $\mathbf{H}_j^{***}$: 1.$\mathcal{D}$ uniformly samples $P_1, P_2, R \leftarrow \mathcal{P}_n$. 2. Run $\mathcal{A}$, using R to answer its classical queries and P_1, P_2 to answer its quantum queries, until the conclusion of $\mathcal{A}$'s $(j+1)$-th classical query. 3. Uniformly sample $s_0, s_1 \in \{0,1\}^n$, define $k = s_0 \oplus x_{j+1}$. Use $E_k[P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}]$ to answer its remaining classical queries (definitions of $P^{(1)}_{1,T_j,k}$ and $P^{(1)}_{2,T_j,k}$ as above). And use $P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}$ to answer its quantum queries, where $P^{(1)}_{1,T_j,k} = S_{T_j,P_1,k} \circ P^{(1)}_1$ and similarly for $P^{(1)}_{2,T_j,k}$.

We have the following inequalities which will be proven in subsequent lemmas:

$$\left|\Pr[\mathcal{A}(\mathrm{H}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}^*_j) = 1]\right| \leq 4\sqrt{\frac{q_P}{2^n}}, \tag{3}$$

$$\left|\Pr[\mathcal{A}(\mathrm{H}^*_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}^{**}_j) = 1]\right| \leq 4\sqrt{\frac{q_P}{2^n}}, \tag{4}$$

$$\left|\Pr[\mathcal{A}(\mathrm{H}^{**}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}^{***}_j) = 1]\right| \leq \frac{j}{2^n}, \tag{5}$$

$$\left|\Pr[\mathcal{A}(\mathrm{H}^{***}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}'_j) = 1]\right| \leq \frac{2(j+q_E)}{2^n}. \tag{6}$$

Applying the triangle inequality, we get:

$$\left|\Pr[\mathcal{A}(\mathrm{H}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}'_j) = 1]\right| \leq 8\sqrt{\frac{q_P}{2^n}} + \frac{3j}{2^n} + \frac{2q_E}{2^n}.$$

□

Lemma 5. *Let $\mathcal{A}$ be an adversary making at most q_E classical queries and q_P quantum queries. Then for $j = 0, \cdots, q_E$, we have:*

$$\left|\Pr[\mathcal{A}(H_j) = 1] - \Pr[\mathcal{A}(H^*_j) = 1]\right| \leq 4\sqrt{\frac{q_P}{2^n}}.$$

Proof. Construct a distinguisher $\mathcal{D}$ that participates in the resampling experiment for P_1. Its behavior is divided into two stages:

Phase 1: $\mathcal{D}$ is given quantum access to random permutations P_1, P_2 and samples a random permutation $R \leftarrow \mathcal{P}_n$ (used to answer classical queries). $\mathcal{D}$ runs the adversary $\mathcal{A}$ and answers its queries: quantum queries are answered with P_1, P_2; classical queries are answered with R. $\mathcal{D}$ continues running $\mathcal{A}$ until $\mathcal{A}$ submits the $(j+1)$-th classical query x_{j+1} (let the list of the first j classical queries be $T_j = ((x_1, y_1), ..., (x_j, y_j))$).

Phase 2: $\mathcal{D}$ receives $s_0, s_1 \in \{0,1\}^n$ from the resampling experiment (s_0, s_1 are uniformly random) and quantum access to $P^{(b)}_1$, where b is the random bit of the resampling experiment, and

$$P^{(b)}_1(x) = P_1 \circ \mathrm{swap}_{s_0,s_1}(x)$$

. Set the key component $k = s_0 \oplus x_{j+1}$. Answer $\mathcal{A}$'s $(j+1)$-th classical query:

$$y_{j+1} = E_k[P^{(b)}_{1,T_j,k}, P_2](x_{j+1}) = P^{(b)}_{1,T_j,k}(x_{j+1} \oplus k) \oplus P_2(x_{j+1} \oplus k) \oplus k.$$

Substituting $k = s_0 \oplus x_{j+1}$, this simplifies to $y_{j+1} = P^{(b)}_{1,T_j,k}(s_0) \oplus P_{2,T_j,k}(s_0) \oplus k$. Continue running $\mathcal{A}$, answering remaining queries: classical queries are answered with $E_k[P^{(b)}_{1,T_j,k}, P_{2,T_j,k}]$; quantum queries are answered with the modified permutations $P^{(b)}_1$

and P_2, where $P^{(b)}_{1,T_j,k} = S_{T_j,P_1,k} \circ P^{(b)}_1$ (the sequence $S_{T_j,P_1,k}$ ensures consistency of $E_k[P^{(b)}_{1,T_j,k}, P_{2,T_j,k}]$ with classical queries); $\mathcal{D}$ outputs $\mathcal{A}$'s output bit.

When $b = 0$, $P^{(0)}_1 = P_1$, and thus $\mathcal{D}$ simulates experiment H_j: $\Pr[\mathcal{D}$ outputs $1 \mid b = 0] = \Pr[\mathcal{A}(\mathrm{H}_j) = 1]$. When $b = 1$, $P^{(1)}_1 = P_1 \circ \mathrm{swap}_{s_0,s_1}$, and thus $\mathcal{D}$ simulates experiment H^*_j: $\Pr[\mathcal{D}$ outputs $1 \mid b = 1] = \Pr[\mathcal{A}(\mathrm{H}^*_j) = 1]$. Applying the Resampling Lemma (Lemma 2), the number of queries $\mathcal{D}$ makes to P_1 in Phase 1 is q_{P1}, i.e., the number of quantum queries $\mathcal{A}$ makes to P_1 in Phase 1. Since $\mathcal{A}$'s total quantum queries are q_P, distributed between P_1 and P_2, we have $q_{P1} \le q_P$. Combined with the conclusion of the Resampling Lemma, we get:

$$\begin{aligned}\left|\Pr[\mathcal{A}(\mathrm{H}_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}^*_j) = 1]\right| &= |\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1]| \\ &\le 4\sqrt{\frac{q_{P1}}{2^n}} \le 4\sqrt{\frac{q_P}{2^n}}.\end{aligned}$$

□

Lemma 6. *Let $\mathcal{A}$ be an adversary making at most q_E classical queries and q_P quantum queries. Then for $j = 0, \cdots, q_E$, we have:*

$$\left|\Pr[\mathcal{A}(H^*_j) = 1] - \Pr[\mathcal{A}(H^{**}_j) = 1]\right| \le 4\sqrt{\frac{q_P}{2^n}}.$$

Proof. Construct a distinguisher $\mathcal{D}$ that participates in the resampling experiment for P_2. Its behavior is divided into two stages:

Phase 1: $\mathcal{D}$ is given quantum access to random permutations P_1, P_2 and samples a random permutation $R \leftarrow \mathcal{P}_n$ (used to answer classical queries). $\mathcal{D}$ runs the adversary $\mathcal{A}$ and answers its queries: quantum queries are answered with P_1, P_2; classical queries are answered with R. $\mathcal{D}$ continues running $\mathcal{A}$ until $\mathcal{A}$ submits the $(j+1)$-th classical query x_{j+1}. Let $T_j = \{(x_1, y_1), ..., (x_j, y_j)\}$ be the list of the first j classical queries.

Phase 2: $\mathcal{D}$ receives $s_0, s_1 \in \{0,1\}^n$ from the resampling experiment (s_0, s_1 are uniformly random) and quantum access to $P^{(b)}_2$ (where b is the random bit of the resampling experiment). Set the key component $k = s_0 \oplus x_{j+1}$. $\mathcal{D}$ answers $\mathcal{A}$'s $(j+1)$-th classical query:

$$y_{j+1} = E_k[P^{(1)}_{1,T_j,k}, P^{(b)}_{2,T_j,k}] = P^{(1)}_{1,T_j,k}(x_{j+1} \oplus k) \oplus P^{(b)}_{2,T_j,k}(x_{j+1} \oplus k) \oplus k.$$

Substituting $k = s_0 \oplus x_{j+1}$, this simplifies to $y_{j+1} = P^{(1)}_{1,T_j,k}(s_0) \oplus P^{(b)}_{2,T_j,k}(s_0)$. $\mathcal{D}$ continues running $\mathcal{A}$: it answers remaining classical queries with $E_k[P^{(1)}_{1,T_j,k}, P^{(b)}_{2,T_j,k}]$ and quantum queries with the modified permutations $P^{(1)}_1$ and $P^{(b)}_2$. $\mathcal{D}$ outputs $\mathcal{A}$'s output bit.

When $b = 0$: $P^{(0)}_2 = P_2$. Then $\mathcal{D}$ simulates experiment H^*_j: $\Pr[\mathcal{D}$ outputs $1 \mid b = 0] = \Pr[\mathcal{A}(\mathrm{H}^*_j) = 1]$. When $b = 1$: $P^{(1)}_2 = P_2 \circ \mathrm{swap}_{s_0,s_1}$. Then $\mathcal{D}$ simulates experiment H^{**}_j: $\Pr[\mathcal{D}$ outputs $1 \mid b = 1] = \Pr[\mathcal{A}(\mathrm{H}^{**}_j) = 1]$. By the Resampling Lemma, the number of queries $\mathcal{D}$ makes to P_2 in Phase 1 is q_{P2}, i.e., the number of quantum queries $\mathcal{A}$ makes to P_2 in Phase 1. Since $\mathcal{A}$'s total quantum queries are q_P, distributed between P_1 and P_2, we have $q_{P2} \le q_P$. Therefore:

$$\begin{aligned}\left|\Pr[\mathcal{A}(\mathrm{H}^*_j) = 1] - \Pr[\mathcal{A}(\mathrm{H}^{**}_j) = 1]\right| &= |\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1]| \\ &\le 4\sqrt{\frac{q_{P2}}{2^n}} \le 4\sqrt{\frac{q_P}{2^n}}.\end{aligned}$$

□

Lemma 7. *Let $\mathcal{A}$ be an adversary making at most q_E classical queries and q_P quantum queries. Then for $j = 0, \cdots, q_E$, we have:*

$$\left|\Pr[\mathcal{A}(H_j^{**}) = 1] - \Pr[\mathcal{A}(H_j^{***}) = 1]\right| \leq \frac{j}{2^n}.$$

Proof. Construct a distinguisher $\mathcal{D}$ whose behavior is divided into two stages:

Phase 1: $\mathcal{D}$ is given quantum access to random permutations P_1, P_2 and samples a random permutation $R \leftarrow \mathcal{P}_n$ (used to answer classical queries). $\mathcal{D}$ runs the adversary $\mathcal{A}$ and answers its queries: quantum queries are answered with P_1, P_2; classical queries are answered with R. $\mathcal{D}$ continues running $\mathcal{A}$ until $\mathcal{A}$ submits the $(j+1)$-th classical query x_{j+1}. Let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ be the list of the first j classical queries.

Phase 2: $\mathcal{D}$ receives s_0, s_1 from the resampling experiment (uniformly random). And quantum access to $P^{(1)}_{1,T_j,k}$ and $P^{(1)}_{2,T_j,k}$ (b is the random bit of the resampling experiment). Set the key component $k = s_0 \oplus x_{j+1}$. $\mathcal{D}$ continues running $\mathcal{A}$: it answers the remaining queries: for the $(j+1)$-th classical query x_{j+1}, use $E_k[P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}]$ to answer; H_j^{***} uses the random function R to answer. The other queries are identical in H_j^{**} and H_j^{***}. Classical queries are answered with $E_k[P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}]$, and quantum queries are answered with the modified permutations $P^{(1)}_{1,T_j,k}$ and $P^{(1)}_{2,T_j,k}$.

We need to prove that in H_j^{**}, the response y_{j+1} for the $(j+1)$-th query is sufficiently uniformly random from the adversary's perspective, and is indistinguishable from the response using the random function in H_j^{***}.

In H_j^{**},

$$y_{j+1} = E_k[P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}](x_{j+1}) = P^{(1)}_{1,T_j,k}(x_{j+1} \oplus k) \oplus P^{(1)}_{2,T_j,k}(x_{j+1} \oplus k) \oplus k,$$

$P^{(1)}_{1,T_j,k}(x_{j+1} \oplus k) = P^{(1)}_{1,T_j,k}(s_0) = P_{1,T_j,k}(s_1), P^{(1)}_{2,T_j,k}(x_{j+1} \oplus k) = P^{(1)}_{2,T_j,k}(s_0) = P_{2,T_j,k}(s_1)$

Let $S = \{a_i = x_i \oplus k, 1 \leq i \leq j\}$,

$$P_{1,T_j,k}(s_1) = \begin{cases} u_i, & \text{if } s_1 \in S, s_1 = a_i \\ P_1(s_1), & \text{if } s_1 \notin S \end{cases},$$

$$P_{2,T_j,k}(s_1) = \begin{cases} y_i \oplus u_i \oplus k, & \text{if } s_1 \in S, s_1 = a_i \\ P_2(s_1), & \text{if } s_1 \notin S \end{cases},$$

where u_i is the random value set when reprogramming P_1.

In H_j^{***}, $y_{j+1} = R(x_{j+1})$ is a random value. Due to the randomness of s_1, we have $\Pr[s_1 \in S] \leq \frac{j}{2^n}$.

Now analyze the randomness of y_{j+1}.

When $s_1 \notin S$: $P_{1,T_j,k}(s_1) = P_1(s_1)$ (not reprogrammed); $P_{2,T_j,k}(s_1) = P_2(s_1)$ (not reprogrammed). Since P_1, P_2 are random permutations, $P_1(s_1)$ and $P_2(s_1)$ are uniformly random, and combined with the uniformly random k, y_{j+1} is uniformly random.

When $s_1 \in S$: Reprogramming sets $P_{1,T_j,k}$ to some random value u_i and $P_{2,T_j,k}$ to $y_i \oplus u_i \oplus k$. In this case, y_{j+1} in H_j^{**} is $P_{1,T_j,k}(s_1) \oplus P_{2,T_j,k}(s_1) \oplus k = u_i \oplus (y_i \oplus u_i \oplus k) \oplus k = y_i$, which is identical to a result obtained from a previous query, allowing the adversary to significantly distinguish H_j^{**} from H_j^{***}.

Therefore, we can give a specific upper bound:

$$\left|\Pr[\mathcal{A}(\mathrm{H}_j^{**}) = 1] - \Pr[\mathcal{A}(\mathrm{H}_j^{***}) = 1]\right| \leq \Pr[s_1 \in S] \leq \frac{j}{2^n}.$$

□

Lemma 8. *Let $\mathcal{A}$ be an adversary making at most q_E classical queries and q_P quantum queries. Then for $j = 0, ..., q_E - 1$, we have:*

$$\left|\Pr[\mathcal{A}(H_j^{***}) = 1] - \Pr[\mathcal{A}(H'_j) = 1]\right| \leq \frac{2(j + q_E)}{2^n}.$$

Proof. The key to distinguishing hybrid experiments Hj^{***} and Hj' is to prove that strict permutation equality holds when certain bad events do not occur. Specifically, we need to show $P1, T_j, k^{(1)} = P1, T_j, k$ and $P^{(1)}_{2,T_j,k} = P_{2,T_j,k}$, where $P^{(1)}_{1,T_j,k}$ and $P^{(1)}_{2,T_j,k}$ are permutations defined based on the first j queries and the resampling points s_0, s_1, while $P_{1,T_j,k}$ and $P_{2,T_j,k}$ are defined based on the first $j + 1$ queries.

Let $S = \{a_i = x_i \oplus k \mid 1 \leq i \leq j\}$.

Define the following bad events: * Bad$_0$: $s_1 \in S$. Suppose $s_1 = a_i = x_i \oplus k, 1 \leq i \leq j$, where s_1 is the random point used in the reprogramming of P_1 and P_2. In this case, $P_{1,T_j,k}(s_1) = u_i$, $P_{2,T_j,k}(s_1) = u_i \oplus y_i \oplus k$, and

$$E_k[P^{(1)}_{1,T_j,k}, P^{(1)}_{2,T_j,k}](x_{j+1}) = P_{1,T_j,k}(s_1) \oplus P_{2,T_j,k}(s_1) \oplus k = y_i,$$

which conflicts with the adversary's previous query result (x_i, y_i). * Bad$_1$: $s_0 \in S$ (since s_0 is randomly sampled, i.e., $x_{j+1} \oplus k \in S$, suppose $x_{j+1} = x_i$). In this case,

$$P_{1,T_j,k}(s_0) = P_{1,T_j,k}(x_{j+1} \oplus k) = P_{1,T_j,k}(x_i \oplus k) = u_i,$$

$$P_{2,T_j,k}(s_0) = P_{2,T_j,k}(x_{j+1} \oplus k) = P_{2,T_j,k}(x_i \oplus k) = u_i \oplus y_i \oplus k,$$

$$E_k[P_{1,T_j+1,k}, P_{2,T_j+1,k}](x_{j+1}) = y_i,$$

which also conflicts with the previous query result (x_i, y_i). * Bad$_2$: The adversary queries a point x in the second phase such that $x \oplus k \in \{s_0, s_1\}$. Since s_0 and s_1 are uniformly random and k is uniformly random, for any adversary query x, we have $\Pr[x \oplus k = s_0] \leq \frac{1}{2^n}$ and $\Pr[x \oplus k = s_1] \leq \frac{1}{2^n}$. Therefore, for q_E classical queries, $\Pr[\text{Bad}_2] \leq \frac{2q_E}{2^n}$.

The overall bad event is Bad = Bad$_0 \cup$ Bad$_1 \cup$ Bad$_2$, where:

$$\Pr[\text{Bad}_0] \leq \frac{j}{2^n} \quad (\text{from } s_1 \in S), \quad \Pr[\text{Bad}_1] \leq \frac{j}{2^n} \quad (\text{from } s_0 \in S), \quad \Pr[\text{Bad}_2] \leq \frac{2q_E}{2^n} \quad (\text{from query conflicts}).$$

Thus,

$$\left|\Pr[\mathcal{A}(\text{H}_j^{***}) = 1] - \Pr[\mathcal{A}(\text{H}'_j) = 1]\right| \leq \Pr[\text{Bad}] \leq \frac{2j}{2^n} + \frac{2q_E}{2^n} = \frac{2(j + q_E)}{2^n}.$$

□

4. Conclusion and Discussion

The development of quantum computing poses a serious threat to symmetric cryptography based on fixed key lengths. To address this, this paper studies the security of the SoEM21 pseudorandom function construction under the quantum Q1 model, which captures the realistic scenario where the attacker only has classical access to the encryption/decryption oracles. By comprehensively applying the Reprogramming and Resampling Lemmas, we have for the first time established a rigorous Q1 security proof for the SoEM21 construction, demonstrating that it has a tight security lower bound of $n/3$ bits. This result matches the known optimal attack complexity, thereby confirming that the SoEM21 construction can still provide a reliable security foundation in the quantum era.

Worthwhile future work includes: generalizing this proof framework to generalized SoEM constructions with a number of branches $t > 2$; investigating whether there exists a provable security bound for SoEM under the stronger Q2 model; or applying the analytical methods developed in this work to the security verification of other permutation-based "summation-type" cryptographic components.

References

[ABKM22a] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the Even-Mansour cipher. In *Advances in Cryptology – EUROCRYPT 2022*, volume 13277 of *Lecture Notes in Computer Science*, pages 458–487, Cham, 2022. Springer.

[ABKM22b] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of tweakable Even-Mansour, and applications. In *Advances in Cryptology – CRYPTO 2022*, volume 13509 of *Lecture Notes in Computer Science*, pages 403–432, Cham, 2022. Springer.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Berlin, Heidelberg, 2011. Springer.

[BDLN22] Arghya Bhattacharjee, Avijit Dutta, Eik List, and Mridul Nandi. CENCPP*: Beyond-birthday-secure encryption from public permutations. *Designs, Codes and Cryptography*, 90:1381–1425, 2022.

[BHNPS19] Xavier Bonnetain, Akinori Hosoyamada, Maria Naya-Plasencia, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In *Advances in Cryptology – ASIACRYPT 2019*, volume 11921 of *Lecture Notes in Computer Science*, pages 552–583, Cham, 2019. Springer.

[CEM25] Bai Chen, Mehdi Esmaili, and Atul Mantri. Quantum security analysis of the key-alternating ciphers. Cryptology eprint archive, Cryptology ePrint Archive, 2025. Report 2025/945, https://eprint.iacr.org/2025/945.

[CLLL20] Wonseok Choi, Byeonghak Lee, Yeongmin Lee, and Jooyoung Lee. Improved security analysis for nonce-based enhanced hash-then-mask MACs. In *Advances in Cryptology – ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 697–723, Cham, 2020. Springer.

[CLM19] Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In *Advances in Cryptology – CRYPTO 2019*, volume 11692 of *Lecture Notes in Computer Science*, pages 266–293, Cham, 2019. Springer.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round Fiat-Shamir and more. In *Advances in Cryptology – CRYPTO 2020*, volume 12172 of *Lecture Notes in Computer Science*, pages 602–631, Cham, 2020. Springer.

[GHHM21] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In *Advances in Cryptology – ASIACRYPT 2021*, volume 13093 of *Lecture Notes in Computer Science*, pages 599–628, Cham, 2021. Springer.

[GHY24] Chun Guo, Anjia Huang, and Yu Yu. Quantum Q1 security proof for FX key-length extension construction. *Journal of Cryptologic Research*, 11(5):1139–1151, 2024.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. *arXiv preprint*, 1996. arXiv:quant-ph/9605043.

[JST21] Jonathan Jaeger, Fang Song, and Stefano Tessaro. Quantum key-length extension. In *Theory of Cryptography – TCC 2021*, volume 13043 of *Lecture Notes in Computer Science*, pages 209–239, Cham, 2021. Springer.

[KLLNP16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology – CRYPTO 2016*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[KM10] Hiroyuki Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and a random permutation. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2010)*, pages 2682–2685. IEEE, 2010.

[KM12] Hiroyuki Kuwakado and Masakatu Morii. Security of the quantum-type Even-Mansour cipher. In *Proceedings of the International Symposium on Information Theory and Its Applications (ISITA 2012)*, pages 312–316. IEEE, 2012.

[LFG+25] Zhenqiang Li, Shuqin Fan, Fei Gao, Yonglin Hao, Hongwei Sun, Xichao Hu, and Dandan Li. Quantum attacks on sum of Even-Mansour construction utilizing online classical queries. *EPJ Quantum Technology*, 12, 2025.

[LM17] Gregor Leander and Alexander May. Grover meets Simon – quantumly attacking the FX-construction. In *Advances in Cryptology – ASIACRYPT 2017*, volume 10625 of *Lecture Notes in Computer Science*, pages 161–178. Springer, 2017.

[SI22] Kenta Shinagawa and Tetsu Iwata. Quantum attacks on sum of Even-Mansour pseudorandom functions. *Information Processing Letters*, 173:106172, 2022.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2012.

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *Journal of the ACM*, 62(6):1–49, 2015.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, pages 679–687. IEEE, 2012.

[Zha13] Mark Zhandry. A note on quantum-secure PRPs. In *Advances in Cryptology – ASIACRYPT 2013*, volume 8269 of *Lecture Notes in Computer Science*, pages 272–288, Berlin, Heidelberg, 2013. Springer.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology – CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268, Cham, 2019. Springer.