Physica Scripta

OPEN ACCESS

RECEIVED
16 September 2025

REVISED
9 October 2025

ACCEPTED FOR PUBLICATION
14 October 2025

PUBLISHED
24 October 2025

PAPER

Quantum implementation of SLIM and its Grover cryptanalysis

H O Cildiroglu[1,2,3,*] **and O Yayla**[2]

[1] Department of Physics Engineering, Ankara University, Dögol st, 06100, Ankara, Türkiye
[2] Institute of Applied Mathematics, Middle East Technical University, Universiteler, Ankara, 06800, Ankara, Türkiye
[3] Physics Department, Boston University, Commonwealth Ave, Boston, 02100, MA, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: hoc@physics.bu.edu **and** oguz@metu.edu.tr

Keywords: SLIM, block ciphers, quantum implementation, grover analysis

Abstract

We present a novel quantum implementation of SLIM, a lightweight 32/80-bit Feistel cipher. By inverting its Key addition-Substitution-Permutation (KSP) layers without ancillary qubits, we construct the full quantum circuit using 112 qubits with quantum cost 30,404 and depth 4,066. To assess quantum resilience, we implement Grover's algorithm with three plaintext-ciphertext pairs ($r = 3$; 337 qubits), yielding an attack cost of 2^{111} under NIST's MAXDEPTH constraints—below the Level-1 threshold 2^{170}. While SLIM falls short of NIST's security threshold for near-term quantum attacks, among other lightweight BCs, its efficient architecture retains practical advantages for constrained applications.

1. Introduction

Classical cryptography relies on computational hardness of mathematical problems such as prime factorization and discrete logarithm to ensure encrypted data security. The potential of quantum computers to efficiently solve these problems via quantum algorithms fundamentally threatens encryption protocols [1–6]. This challenge also extends to emerging cryptographic paradigms, including chaos-based and neuromorphic encryption systems [7–11]. Therefore, research on quantum-resistant algorithms has intensified to maintain secure communications and data protection [12–17]. Designing quantum implementations of classical ciphers is thus critical both for enhancing quantum threat resilience and for comprehensive security analysis against quantum attacks [18–29].

In the post-quantum era, despite Grover's algorithm halving the effective key length for brute-force attacks, block ciphers (BCs) are considered the most prominent classical algorithms for encrypting data blocks or fixed-length bit groups [2]. Most BCs process predefined 64- or 128-bit data blocks [30–45]. The Feistel network constitutes a core design paradigm for symmetric BCs, enabling block splitting, subkey usage, round-based processing, and iterative structure to perform both encryption and decryption processes with the same key [30, 31]. Designed to enhance encryption security, it provides a flexible framework that can be employed in many encryption algorithms. SLIM, a novel variant of Feistel-based BCs with 32-bit block size, is designed for lightweight applications from RFID technologies to the Internet of Things (IoTs), striking a balance between security and efficiency [46]. By combining a compact structure with non-linear operations, SLIM achieves robust encryption despite its low bit-width constraints.

In this paper, we study the quantum implementation of BCs, which aims to analyze classical encryption algorithms within quantum computing frameworks, assess their security against quantum adversaries, and develop quantum-resistant cryptographic methods. Specifically, we present a novel quantum implementation of SLIM, a lightweight BC (LBC), that shows a higher quantum gate cost and circuit depth, which are key measures to evaluate resistance to quantum attack, compared to ARX-based designs SIMON and SPECK, as well as SPN-Feistel architectures PRINT and RECTANGLE. While the gate cost of SLIM in our implementation remains lower than that of the SPN cipher LBLOCK, its greater circuit depth leads to a higher overall quantum cost. SLIM's optimized quantum cost of 30,404 is achieved with a minimal qubit count. A comprehensive

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Figure 1. S-box of SLIM.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
P(x)	7	D	1	8	B	E	2	5	4	A	F	0	3	6	9	C

Figure 2. P-box of SLIM.

quantum analysis of its Key addition–Substitution–Permutation (KSP) layers enabled implementation with 112 total qubits through inversion of the KSP without ancillary qubits. This approach establishes a foundation for practical implementation with current technology and facilitates the evaluation of resistance against quantum attacks.

Complementing this, we quantitatively assess SLIM's resilience against quantum threats by implementing Grover's search algorithm. Resource estimates (gate depth, T-count, total cost) for Grover-based attacks against SLIM in the 80-bit key space are computed for scenarios using two and three ciphertext-plaintext pairs, requiring 225 and 337 qubits, respectively. The total qubit count is calculated as the sum of the key register allocation, combined with the qubits required for parallel execution of multiple SLIM cipher instances and an additional target qubit for the Grover oracle. We see that SLIM's projected total quantum cost 2^{111} falls below NIST's Level-1 security threshold 2^{170}, indicating SLIM may exhibit insufficient resistance to near-term quantum threats [47]. However, its low qubit requirement, combined with a relatively strong security profile, provides advantages over comparable LBCs.

This paper is organized as follows. Section 2 presents the KSP structure of SLIM along with the notation adopted for quantum circuit design. In section 3, the quantum gate-level implementation of SLIM is detailed. Section 4 focuses on the quantum resource estimation of the SLIM implementation and provides a comparison with other BC realizations. Lastly, section 5 analyzes SLIM's resistance to quantum attacks through Grover's search algorithm, offering resource estimates under `MAXDEPTH` constraint, which defines the maximum allowed quantum circuit depth for an attack and is used to estimate the necessary circuit size for parallel computation.

2. The structure of SLIM

The development of quantum technologies poses a significant threat to the security of existing cryptographic algorithms, encouraging the exploration of quantum-resilient (or post-quantum) cryptography. In this regard, block ciphers (BC) stand out with its simple but resilient design. However, some BC algorithms may be solved by quantum computers due weakness in their design, e.g. Grover search attack on their differential characteristics [48].

SLIM, a recently introduced variant of LBC, promises to be a quantum-resistant algorithm with its lightweight Feistel structure, functioning as a symmetric encryption algorithm that uses the same key for both encryption and decryption and offering a well-balanced design in terms of security and efficiency [46]. The only difference between these processes is the use of decryption sub-keys in reverse order. SLIM integrates both confusion and diffusion principles. Confusion is efficiently handled through a compact 4-bit S-box (See figure 1), using the same structure of PRESENT [37]. Operating with an 80-bit key, SLIM is designed to encrypt and decrypt 32-bit plaintext and ciphertext blocks, respectively.

The rearrangement phase of SLIM utilizes a crucial permutation layer, generating a 16-bit output from 16-bit inputs through a meticulously selected rule specified in figure 2, which addresses the need to provide resistance against linear and differential cryptanalysis [46]. In linear cryptanalysis, the permutation rule is designed to resist patterns, creating substantial confusion to hinder adversaries from identifying patterns within the cipher. Simultaneously, the rule takes into account differential cryptanalysis, disrupting systematic relationships between input and output differentials to fortify the cipher's resilience against such attacks. Hence, these

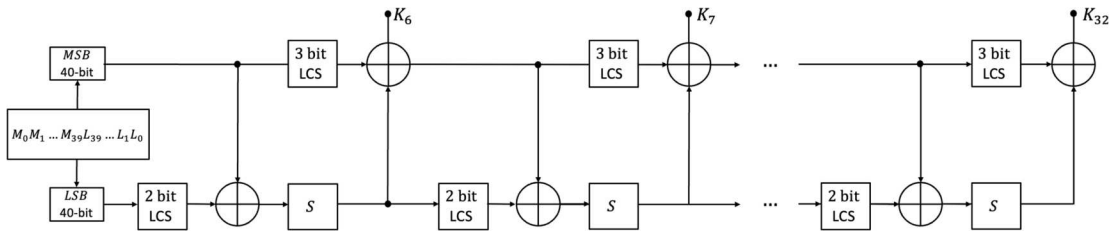

Figure 3. Key Generation of SLIM ($i > 5$).

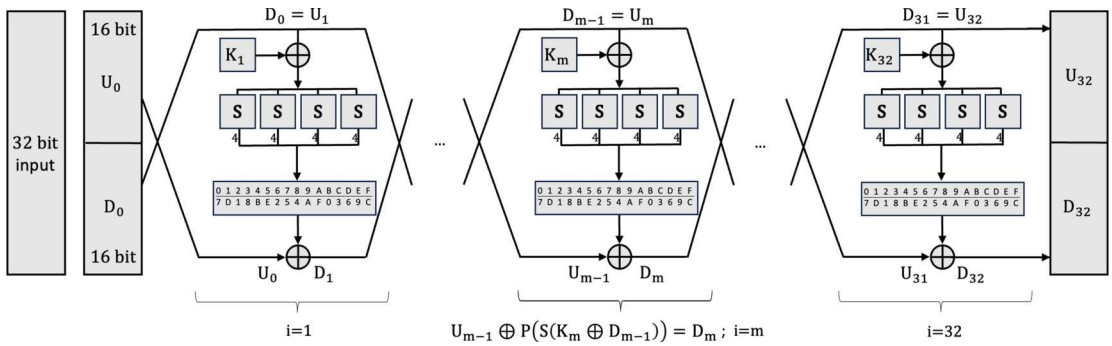

Figure 4. The interior structure of SLIM.

techniques play a role in thwarting various cryptanalytic approaches, illustrating the algorithm's effectiveness and suitability for secure information transmission.

NIST guidelines recommend a key length of at least 112 bits to apply cryptographic protection to sensitive information (especially for federal government information, e.g., encrypting data or generating a digital signature) [49]. However, some flexibility for 80 bits is allowed for processing information already protected at those security strengths (e.g. decrypting encrypted data or verifying digital signatures). Furthermore, there is an increasing need for LBC architectures with smaller key lengths (<112) in resource-constrained environments, such as IoT or RFID technologies [46]. In this context, SLIM is designed to meet this requirement, where 32 sub-keys (16-bit each) are generated from an 80-bit encryption key. The initial five sub-keys $\{K_1, \ldots, K_5\}$ are directly derived from the original key ($K = M_0M_1 \ldots M_{39}L_{39} \ldots L_1L_0$). Specifically, K_1 corresponds to the first least significant 16 bits, K_2 to the subsequent 16 bits, and so forth (up to the fifth round). The 80-bit key is then divided by a splitter, yielding two 40-bit values as the most and least significant bits (MSB and LSB). Each half is subsequently processed individually (See figure 3). In each round, the LSB undergoes a left cyclical shift (LCS) of two bits, followed by an XOR operation with the MSB. The output of this XOR process is then passed to a substitution layer. The round sub-key is created by manipulating the output of the S-boxes and the rotated MSB, achieved through an LCS of 3 bits using the XOR technique.

In the SLIM's encryption algorithm, the input splits into up and down parts, which go through 32 rounds of processing along with the generated sub-keys. The interior structure of SLIM is given in figure 4. Accordingly, the 32-bit input is split into two equal sixteen-bit halves as the lower half (D_i) and the upper half (U_i), where $D_{i-1} = U_i$ and $U_{i-1} \oplus P(S(K_i \oplus D_{i-1})) = D_i$.

3. Quantum implementation of SLIM

In quantum mechanics, the state of a system is described by wave functions ($|\psi\rangle$), defined in the Hilbert space ($\mathcal{H}$)-a vectorial complex inner product space. These wave functions represent the probability amplitude of a particle that exhibits a specific property (e.g. position or spin) upon measurement. Thus, a quantum state is a complex linear superposition of possible substates within this space. Single-particle quantum systems with two subparts, such as horizontal $|H\rangle$ and vertical $|V\rangle$ polarization, spin-up $|\uparrow\rangle$ and spin-down $|\downarrow\rangle$, or more broadly $|1\rangle$ and $|0\rangle$, are referred to as quantum bits (or simply qubits). Qubits are the quantum counterparts of classical bits, the fundamental units of classical computation. The general state of a single qubit ψ is expressed as:

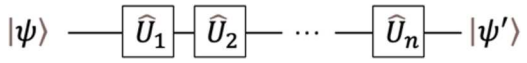

Figure 5. An equivalent representation of the sequential action of the operators on the state.

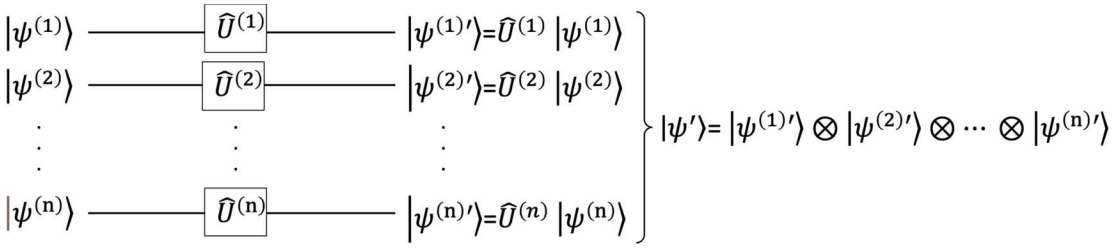

Figure 6. Representation of the operators acting on an n-qubit system.

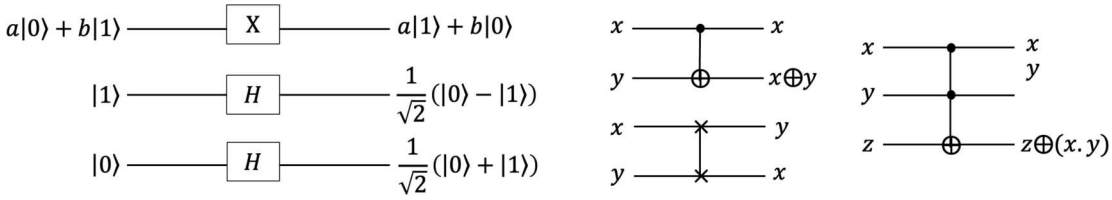

Figure 7. Quantum gates: (a) X: NOT, H: Hadamard; (b-top) CNOT: Controlled-NOT, (b-bottom) SWAP; (c) CCNOT. Here $a,\ b \in \mathbb{C}, x, y, z \in \{0, 1\}$, and $\oplus$ is the bitwise XOR.

$$|\psi\rangle = a|1\rangle + b|0\rangle; \quad |\psi\rangle \in \mathcal{H}_2; \quad a, b \in \mathbb{C}; \quad a^2 + b^2 = 1. \tag{1}$$

Composite systems are formed by combining two or more discrete and separately prepared qubits ($|\psi^{(i)}\rangle \in \mathcal{H}^i;\ i = 1,\ldots,n$), which reside in the tensor product space $\mathcal{H}^{\otimes n}$, can be given as:

$$|\Psi\rangle = |\psi^{(1)}\rangle \otimes \cdots \otimes |\psi^{(n)}\rangle \quad (\text{or simply } |\psi^{(1)} \ldots \psi^{(n)}\rangle), \tag{2}$$

where $|\Psi\rangle \in \mathcal{H} = \mathcal{H}^{(1)} \otimes \cdots \otimes \mathcal{H}^{(n)}$. Besides, due to the causality of the theory, the evolution of the system can be controlled by unitary operators such as $|\Psi'\rangle = \hat{U}|\Psi\rangle$. If the system undergoes a stepwise evolution, its state is transformed sequentially by a series of unitary operators. The final output state of the system is $|\Psi'\rangle = \hat{U}_n \ldots \hat{U}_2 \hat{U}_1 |\Psi\rangle$ (See figure 5). On the other hand, the transformed state of the n-qubit system is $|\Psi'\rangle$=Uˆ(1)⊗⋯⊗Uˆ(n)|ψ(1)⟩⊗⋯⊗|ψ(n)⟩ (See figure 6). Here, the evolution operator $\hat{U}^{(i)}$ acts on the ket $|\psi^{(i)}\rangle$. These representations are suitable for constructing quantum implementations of the KSP-layers for the encryption algorithm given in the previous section. To achieve this, some quantum gates such as the X (or simply NOT), Hadamard (H) gate, CNOT, CCNOT (or Toffoli), and SWAP gates need to be introduced (See figure 7).

To implement SLIM as a quantum circuit, the 32-qubit input is first divided into two equal ($U$and$D$) halves. The 16 qubits (D_0) from the D-box are combined with the first round key (K_1), derived by dividing the 80-bit key into five equal parts. Next, the S-box is executed four times in parallel, starting from the least significant qubits (LSQ). The resulting outputs are then passed through the P-box. Following the KSP process, the results are manipulated using CNOT gates with the 16 qubits (U_0) from the U-box, and the first round is completed by obtaining D_1.

At this stage, it is essential to ensure the feasibility of implementing the second and subsequent rounds of SLIM. When KSP is applied on D_0, it transforms into U_1. While this transition is straightforward from a classical perspective, it introduces challenges in quantum paradigm, as it necessitates duplicating the D_0 packet, which is a task that inherently requires additional qubits. This duplication involves adding and applying CNOT gates to a set of ancilla qubits, which are initialized to the $|0\rangle$ state, matching the size of the packet (16 qubits). Moreover, to complete all rounds of SLIM, new ancilla qubits must be introduced before each round, in addition to the initial set. In experimental setups where an increase in qubit count is negligible, such as idealized scenarios, this approach proves advantageous, as it significantly reduces the number of quantum gates

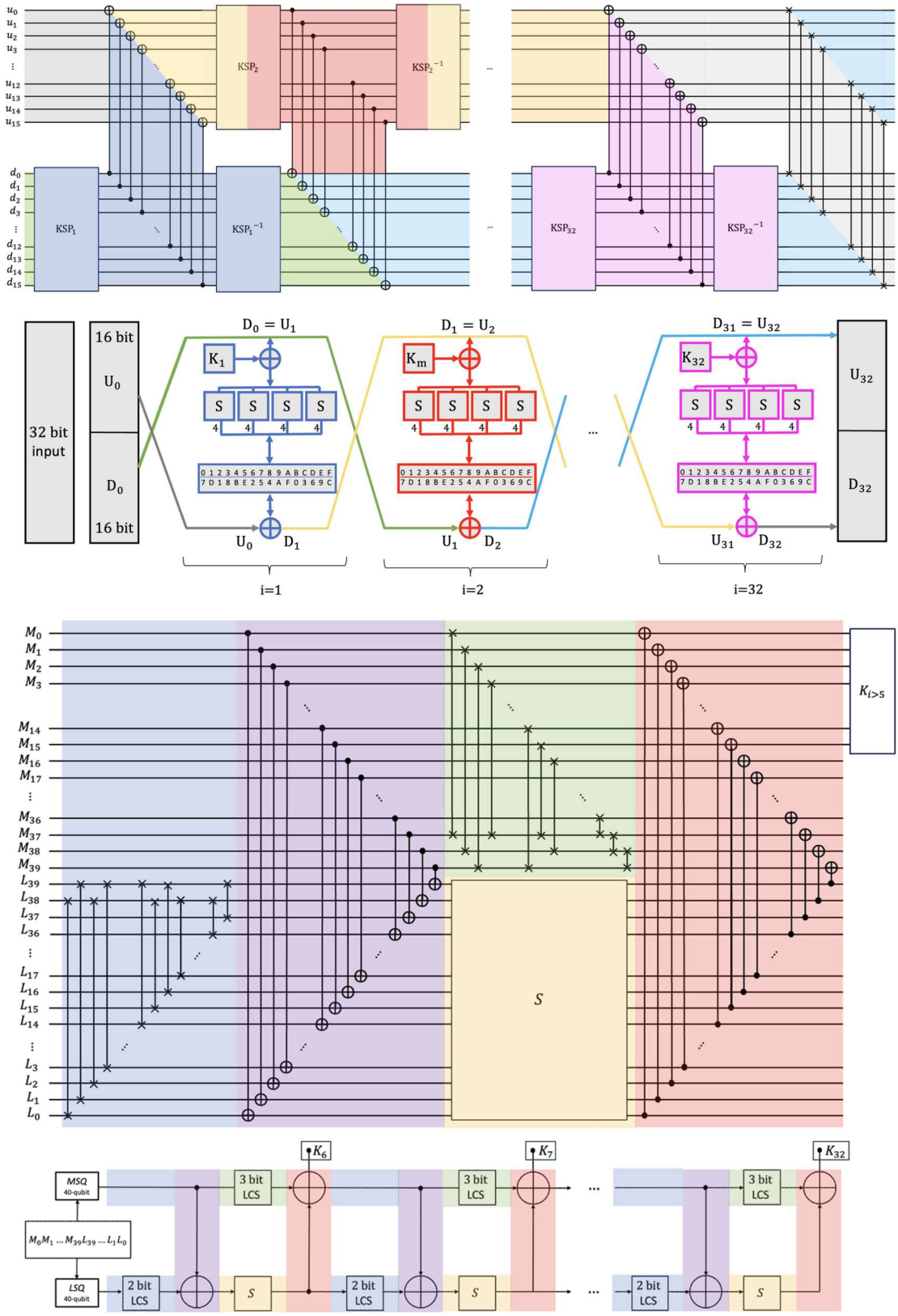

Figure 8. (top) Quantum implementation of 32 rounds of SLIM and (bottom) the key schedule for the rounds $i > 5$.

required. However, given the limitations of current quantum computing technology, such a method is impractical due to the cost of qubit resources. Instead, we employ a more efficient and a novel strategy that avoids excessive qubit usage. By leveraging the Feistel structure of SLIM, we retrieve the original qubits for subsequent rounds through the reverse application of the KSP process (KSP^{-1}), which consists of P^{-1}, S^{-1}, and $K^{-1} = K$, applied in reverse order from the lower to upper branches. This approach effectively prepares the system for the

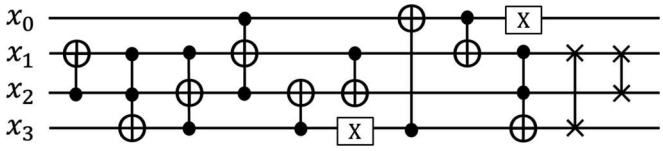

Figure 9. Quantum implementation of S-box.

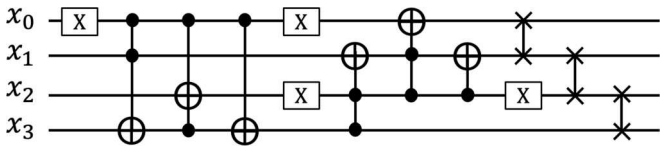

Figure 10. Quantum implementation of S^{-1}-box.

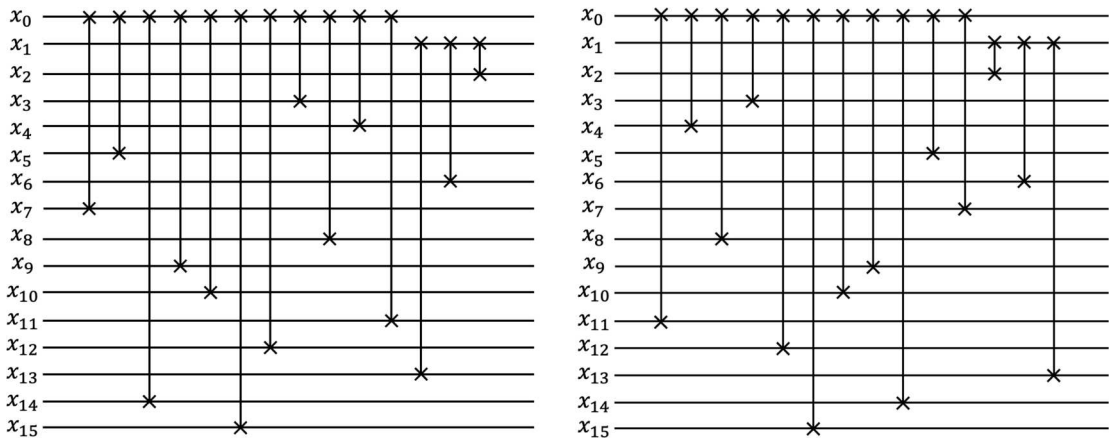

Figure 11. Quantum implementations of (left) P-box and (right) P^{-1}-box.

second round and allows U_i to be reused as D_{i-1} in each subsequent round. Now, let us realise the quantum implementations of the K, S and P layers and their inverses respectively.

K-Layer: SLIM has 80 qubit keys and runs 32 rounds in total. In this context, for the first 5 rounds, the 80-qubit key is directly divided by five according to the LSQ and added as 16-qubit keys (See figure 8). Then, the 80-qubit key goes through a divider to create two 40-qubit keys, labeled MSQ and LSQ. The resulting 40-qubit keys are then treated separately. Accordingly, in each round, the LSQ undergoes a two-qubit circular left shift operation (blue colored) and then the output produced is CNOTed with the MSQ (purple colored). The output of the CNOT process is forwarded to a substitution layer (yellow colored). The result of the S-boxes, which undergoes a three-qubit circular left-shift operation (green colored), is manipulated using the CNOT operations to generate the subkey (red colored). The first 16 qubits are then taken and used as the key. This cycle continues until the 32nd round. The decryption process leverages the Feistel structure of SLIM for the implementation.

S-box: The S-box used in SLIM, given in figure 1, is identical to the one designed for the PRESENT algorithm [37]. Its quantum implementation has been optimized using the LIGHTER-R framework, a specialized tool for reversible circuit synthesis [19]. LIGHTER-R is particularly well-suited for this task because it eliminates the need for ancillary qubits and minimizes garbage output while efficiently optimizing quantum gates. This tool provides an end-to-end solution for reversible S-box construction, offering significant advantages in terms of gate cost and resource efficiency. As such, the quantum circuits for both the S-box and its inverse, S^{-1}-box, are carefully designed in this work to align with these optimizations . To reconstruct the S-box as a quantum circuit with variable output, we use the quantum gates described in the preceding section (NOT, CNOT, and CCNOT gates). The Boolean functions that define the logical operations necessary for constructing the S-box transformations are as follows.

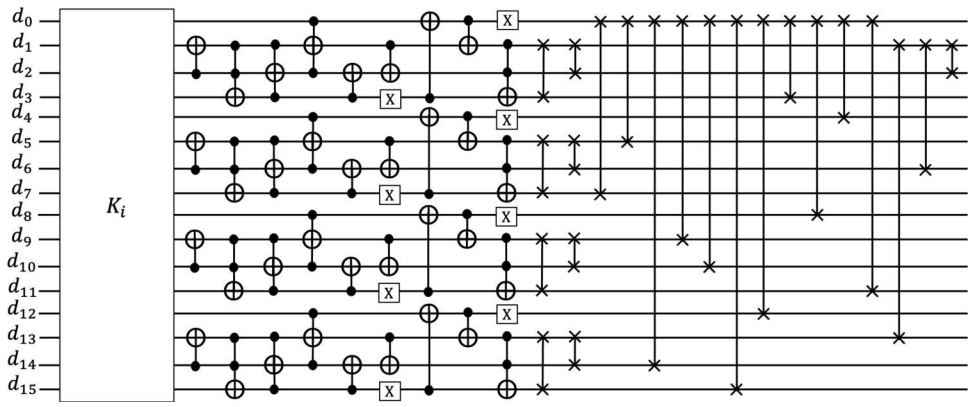

Figure 12. Illustration of one round of implementation of KSP.

$$
\begin{aligned}
x_0 \rightarrow S_0 &= x_3'x_2'x_0 + x_3'x_1x_0 + x_3x_2'x_0' + x_3x_1x_0' + x_3x_2x_1'x_0 + x_3'x_2x_1'x_0' \\
x_1 \rightarrow S_1 &= x_3'x_2'x_1 + x_3x_2x_0 + x_3'x_1x_0' + x_3x_2'x_1' + x_3x_2'x_0' \\
x_2 \rightarrow S_2 &= x_3x_2x_1' + x_2'x_1x_0' + x_3'x_2x_1x_0 + x_3'x_2'x_1' + x_2'x_1'x_0 \\
x_3 \rightarrow S_3 &= x_3x_2'x_1 + x_3x_2'x_0 + x_3'x_1'x_0' + x_3'x_2x_1 + x_3'x_1x_0
\end{aligned} \tag{3}
$$

Here, the notation, e.g., $[x_3'x_2'x_0] + [x_3'x_1x_0]$, indicates [(NOT$x_3$) AND (NOT$x_2$) AND ($x_0$)] OR [(NOT$x_3$) AND ($x_1$)AND($x_0$)], and the relevant input needs to be placed from x_3 to x_0. Using these functions and the LIGHTER-R framework (NCT-gc), the optimized quantum implementation of the SLIM S-box is constructed as shown in figure 9.

Similarly, the Boolean functions for the S^{-1}-box are as follows [50]:

$$
\begin{aligned}
x_0 &= S_3'S_2'S_0' + S_2'S_1'S_0' + S_2S_1'S_0 + S_3'S_2S_0 + S_3S_2S_1S_0' + S_3S_2'S_1S_0 \\
x_1 &= S_3'S_2'S_1'S_0 + S_3'S_1S_0' + S_3S_2S_0 + S_3S_1S_0 + S_3S_2'S_0' \\
x_2 &= S_3'S_2'S_1' + S_3'S_1'S_0' + S_3S_1'S_0 + S_2'S_1S_0' + S_3'S_2S_1S_0 \\
x_3 &= S_3'S_2'S_0 + S_3'S_2'S_1 + S_2S_1S_0 + S_3S_2S_1 + S_3S_2'S_1'S_0' + S_3'S_2S_1'S_0'
\end{aligned} \tag{4}
$$

These functions, when implemented using the LIGHTER-R framework, allow for an efficient reversible construction of the S^{-1}-box. The resulting circuit (figure 10) is optimized to balance quantum gate costs and qubit usage, ensuring compatibility with the overall KSP^{-1} structure.

After the key addition layer processes the 16 bits of the input block U_i, the S-box operates in parallel for four 4-qubit segments, starting from the LSB. These transformed outputs are then routed through the P-layer, where the qubits are permuted according to a predefined mapping.

P-box: The permutation layer given in figure 2 can also be implemented in a quantum circuit. For this, it is sufficient to use the SWAP gates with no quantum cost. In this context, the permutation operation (0 7 5 E 9 A F C 3 8 4 B) (1 D 6 2) can be directly implemented in a 16-qubit quantum circuit (See figure 11). The operation of the P^{-1}-box is (0 B 4 8 3 C F A 9 E 5 7)(1 2 6 D).

KSP: We combine the key addition, substitution, and permutation layers into a single cohesive unit for constructing the KSP structure, which plays a critical role in the implementation of the SLIM cipher. For one encryption round, the KSP acts on the 16 qubits coming from the D-box (D_i) is given in figure 12.

First, the 16 qubits from D_i are CNOTed with the round key K_i, derived from the key schedule (figure 8). Following this, the transformed qubits are divided into four 4-qubit blocks, each of which is processed in parallel through the quantum S-box circuits. These S-boxes, optimized for quantum implementation, introduce the required non-linear transformations (figure 9). The outputs of the four parallel S-box operations are then routed through the P-layer, which performs a predefined permutation of the qubits (figure 11). Upon completing the P-layer, the KSP structure for the current round is finalized, preparing the qubits for interaction with the U-box in the subsequent step.

The KSP's modular design ensures efficiency and straightforward reversibility by leveraging SLIM's Feistel structure. A similar process can be constructed for the inverse operation, KSP^{-1}, which includes P^{-1}-layer (from figure 11-(b)), the S^{-1}-box (from figure 10), and the $K^{-1}(=K)$-layer (from figure 8-bottom). These components work together to return the qubits to their initial state, ensuring seamless integration into the overall quantum implementation of SLIM.

Table 1. Quantum resources requirement for the proposed quantum implementation of SLIM. The column labeled TOTAL represents the total number of gates required for the implementation. The cost calculation employs the standard quantum gate costs metric, where NOT and CNOT gates are assigned a cost of 1 unit, and the CCNOT (Toffoli) gate has a cost of 6 units, based on its gate decomposition [51].

LAYER	NOT	CNOT	CCNOT	TOTAL	COST
S	2	5	4	11	31
S^{-1}	4	2	4	10	30
$K_{i>5}$	20	130	40	5130	9450
$KSP_{i\leqslant 5}$	8	52	16	380	780
$KSP^{-1}_{i\leqslant 5}$	16	24	16	280	680
$KSP_{i>5}$	28	182	56	7182	14742
$KSP^{-1}_{i>5}$	36	154	56	6642	14202
SLIM	1848	9452	3184	14484	30404

Table 2. Cost and qubit comparison of quantum implementations with other LBCs.

CIPHER	QUBIT	NOT	CNOT	CCNOT	TOTAL	DEPTH	COST	REF
SIMON 32/64	96	448	2816	512	3776	946	6336	[20]
SIMON 48/72	120	792	3312	864	4968	1062	9288	[20]
SPECK 32/64	96	42	4222	1290	5554	1694	12004	[21]
SPECK 48/72	120	42	6462	1978	8482	2574	18372	[21]
RECTANGLE 64/80	144	567	4964	2000	7531	226	17531	[22]
LBLOCK 64/80	144	877	16747	14280	31904	1740	103304	[24]
PRINT 48/80	128	154	3840	2304	6298	336	17818	[25]
SLIM 32/80	112	1848	9452	3184	14484	4066	30404	This work

4. Quantum resources for SLIM

The total quantum cost of our design of the SLIM algorithm is evaluated by calculating the number of quantum gates required to construct its circuits. Specifically, the cost is determined using standard gate metrics: a NOT gate has a cost of 1 unit, a CNOT gate also costs 1 unit, and a CCNOT gate (Toffoli gate) has a cost of 6 units [51]. The SLIM algorithm operates over 32 rounds in total. As described in section 3, the first 5 rounds involve the direct use of 80 qubits in the K layer without requiring additional quantum gate operations for key scheduling. Consequently, these rounds incur no extra gate cost. However, starting from the 6th round, the K layer introduces a significant computational overhead due to the quantum gates required for the key scheduling process. Therefore, to calculate the total quantum resource requirements for SLIM, the first 5 rounds and the subsequent rounds should be considered separately (See table 1).

In the quantum circuit of SLIM (figure 8), each round consists of three main steps: the first half of the qubits undergo KSP, followed by CNOT operations with the second half of the qubits, and then the qubits pass through KSP^{-1}. For simplicity, the cost of the CNOT operations between KSP and KSP^{-1} is included in the cost of KSP. The total cost in the first 5 rounds is 1460, which is the sum of the costs of KSP and KSP^{-1} as given in table 1. In subsequent rounds, each K-layer includes 20 NOT, 130 CNOT, and 40 CCNOT. However, the cost of KSP and KSP^{-1} for the remaining 27 rounds is 28944. As a result, the total cost required for our quantum implementation of SLIM is 30404.

Besides, the depth of a quantum circuit (simply quantum-depth), measures to the number of sequential steps required to execute all operations in the circuit. Unlike classical depth, which refers to the maximum number of gates along a single left-to-right path through the circuit, quantum depth accounts for the layering of quantum gates, where parallelizable gates within a layer are considered as one step. This distinction highlights the unique structure of quantum circuits and their reliance on parallelism to optimize execution. Moreover, certain gates, such as the Toffoli gate, play a critical role in determining quantum depth. While a single Toffoli gate has a T-depth of 1, its full decomposition requires seven layers and four additional ancilla qubits [18]. This metric reflects both the circuit's computational complexity and the time required for execution. Deeper circuits often indicate more intricate computations but may also introduce challenges in maintaining coherence.

A detailed analysis of SLIM requires an evaluation of its total quantum cost, circuit depth, and overall qubit utilization compared to other BCs (see table 2). In the quantum implementation of SLIM, the depth of the

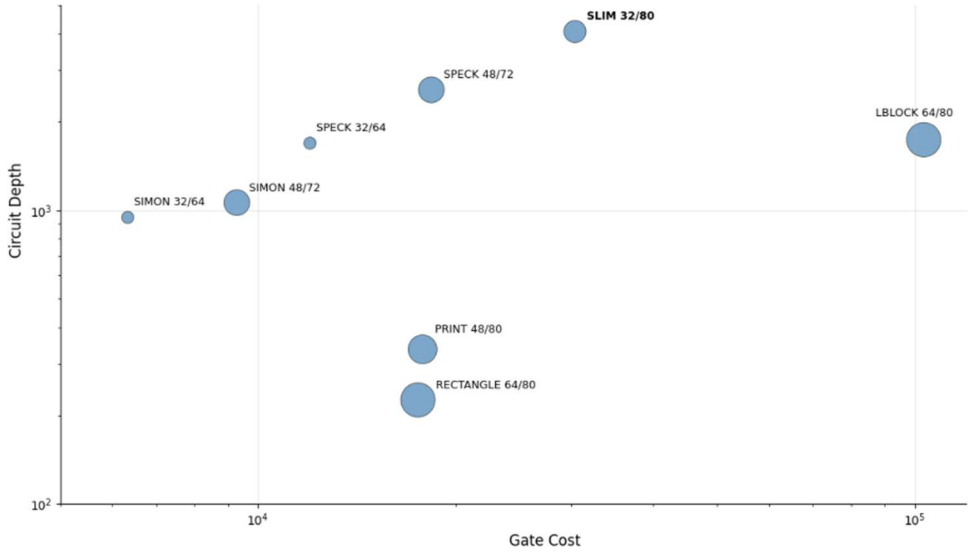

Figure 13. Trade-off analysis of resource metrics. The figure illustrates the relationship between gate cost and circuit depth, where the size of each bubble is scaled to represent the corresponding qubit count.

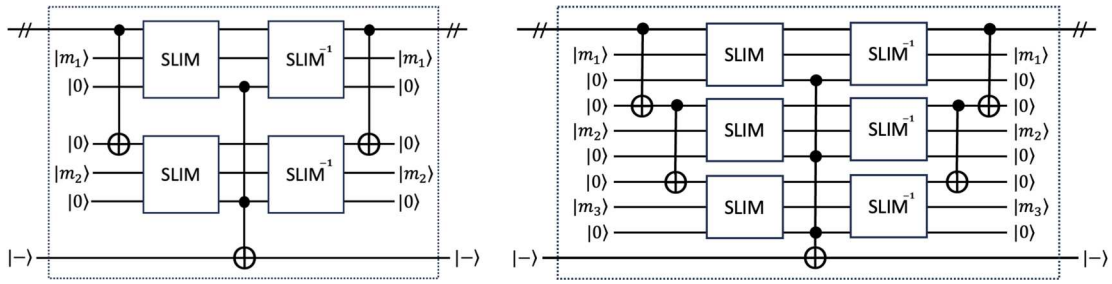

Figure 14. Quantum circuits of Grover oracles U_f for SLIM with inner parallelization: (left) r = 2 two plaintext-ciphertext pairs, and (right) r = 3 three plaintext-ciphertext pairs.

circuit is determined by the different components used in each round. The depth of the S-box is 33, while its inverse, the S^{-1}-box, has a depth of 32. For the K-layer in rounds $i > 5$, the depth is 35. For the first 5 rounds ($i \leqslant 5$), the combined depth of KSP and KSP^{-1} is relatively small, totaling 340. However, in the subsequent 27 rounds ($i > 5$), the depth increases as KSP and KSP^{-1} together contribute to a total depth of 3,726. The total depth across all 32 rounds is 4,066. Hence, the quantum circuit of SLIM achieves an optimal balance, maximizing circuit depth while minimizing qubit count and maintaining a favorable quantum cost.

On the other hand, figure 13 presents a trade-off analysis between gate cost, depth, and qubit count. The current design, which avoids ancillae, results in a higher quantum cost. If ancillary qubits are employed instead, the KSP^{-1} operations become unnecessary. In this scenario, the total quantum cost would be reduced to 16,018 units, composed of 15,522 units from the KSP operations and an additional 496 from the CNOT gates required for ancilla management. In this context, when compared to other BC implementations, SLIM demonstrates an optimal depth-to-cost ratio. Note also that SPN-based SLIM aligns along an approximate linear trend with the ARX-based SIMON and SPECK families on the exponential graph. Despite its low qubit usage, we see that SLIM achieves higher depth values compared to RECTANGLE, PRINT, and LBLOCK, underscoring the impact of different architectures on quantum resource requirements for quantum cryptanalysis.

5. Grover on SLIM

In this section, we quantitatively assess the quantum attack resistance by applying Grover's search algorithm on SLIM [2]. Grover's algorithm is a fundamental quantum protocol providing quadratic speedup over classical approaches when searching unstructured databases that locate specific data within a k-bit key space ($N = 2^k$ elements), reducing brute-force attack complexity from $O(2^k)$ to $O(2^{k/2})$. The algorithm comprises a quantum oracle (U_f) based on a Boolean function f: $\{0,1\}^k \rightarrow \{0,1\}$, and a diffusion operator D. Here, $f(x)$ is defined by

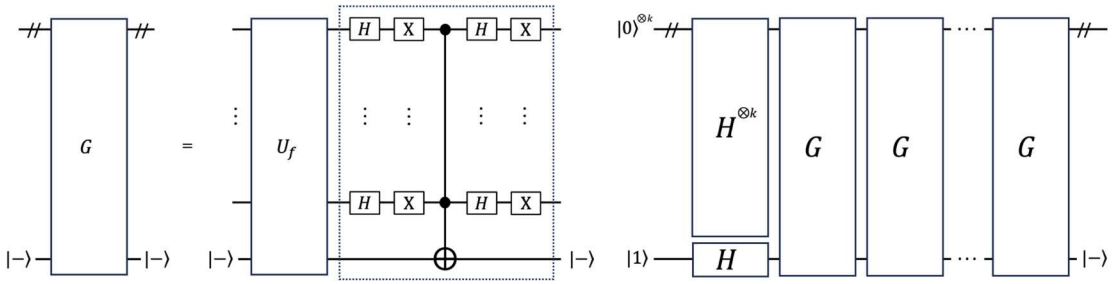

Figure 15. (Left) Grover operator with the general diffusion operator D for n-qubit systems. (Right) Quantum circuits for Grover's algorithm under $\frac{\pi}{4}2^{40}$ iterations.

Table 3. Main resource counts for Grover's oracle in SLIM.

CIPHER	r	QUBIT	NOT	CNOT	CCNOT	TOTAL	COST	DEPTH
SLIM 32/80	2	225	7392	38128	12736	58256	121936	8129
SLIM 32/80	3	337	11088	57272	19104	87464	182984	8129

whether plaintext m encrypted with key x yields (c) ciphertext: $f(x) = 1$if $\mathrm{Enc}_k(m) = c$, and $f(x) = 0$ otherwise. The main idea is to locate the element x_0 satisfying $f(x_0) = 1$. The oracle encrypts the known plaintext with all candidate keys ($x \in \{0, 1\}^k$) in superposition. The resulting ciphertexts are compared against the target ciphertext. For potential matches ($f(x) = 1$), the phase sign of the corresponding key is inverted: $\sum_x a_x|x\rangle|y\rangle \rightarrow \sum_x a_x|x\rangle|y \oplus f(x)\rangle$. At last, the quantum circuit is reversed to uncompute the ciphertexts, leaving only the phase-altered key states.

In this context, the Hadamard operator $H^{\otimes k}$ is applied to the initial state $|0\rangle^{\otimes k}$, yielding a quantum linear complex superposition $|\psi\rangle = \frac{1}{\sqrt{2^k}}\sum_{x\in\{0,1\}^k}|x\rangle$ of all possible states, leveraging quantum supremacy. An ancillary qubit $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ is then introduced, preparing the joint state $|\psi\rangle \otimes |-\rangle$. Subsequently, the oracle operator and generic diffusion operator are applied sequentially on the joint state.

$$U_f(|\psi\rangle \otimes |-\rangle) = 2^{-\frac{k}{2}}\sum_{x=0}^{2^k-1}(-1)^{f(x)}|x\rangle \otimes |-\rangle$$
$$D = 2|\psi\rangle\langle\psi| - I \tag{5}$$

At this stage, the number of plaintext-ciphertext pairs (r) required for successful key recovery attacks must be determined in quantum implementations. The validity of a candidate key can be verified by encrypting r plaintexts and comparing them with the corresponding ciphertexts in the Grover oracle. Only when all pairs match, the correct key is marked by flipping the target qubit. In this circumstance, candidate keys matching only a single pair may contain false positives. Hence, r is expected to exceed the key size/block size ratio ($r > k/n$) [16]. Considering SLIM's lightweight structure, $r = 3$ ciphertext-message pairs are standard for its 80-bit key and 32-bit block size (See figure 14). Following the encryption of the given $|m_1\rangle$, $|m_2\rangle$ and $|m_3\rangle$ using SLIM, the outputs are compared through the ancilla qubits. If the outputs are identical, the target is flipped. Conversely, the quantum implementation of diffusion operator D given in equation (5) is generic for n-qubit systems (figure 15-left). Here, gates between two Hadamard layers invert the phase of the $|0\rangle$ basis state in the top k-bit. Consequently, iterative application of the Grover operator $G = DU_f$ amplifies amplitudes of solution keys satisfying $f(x) = 1$. This process ensures high-probability key measurement when repeated for optimal number (j) of iterations.

Geometrically, each iteration rotates the state vector by a fixed angle θ within the plane spanned by two orthogonal vectors where $\sin^2\theta = M/N$. After j iterations, Grover's algorithm yields the correct state with probability $p(j) = \sin^2((2j + 1)\theta)$. Thus, the optimal iteration count $j \approx \frac{\pi}{4\theta} = \frac{\pi}{4}\sqrt{\frac{N}{M}}$ produces a success probability of at least $p(j) \geqslant 1 - \frac{M}{N}$. This process gradually increases the measurement probability of the solution keys [52]. However, the effectiveness of the algorithm is highly dependent on the efficient implementation of the oracle circuit U_f. This requires an accurate estimation of quantum resources, including the number of qubits and the overall depth and quantum cost. In this regard, since the quantum cost of D is negligible compared to U_f, the resource requirements for G and U_f are approximately equivalent.

Table 4. Resource estimation for Grover's Oracle on SLIM.

CIPHER	r	GATE COST	#GATES	FULL DEPTH	TOTAL COST
SIMON 32/64	3	38016[a]	1.91×2^{45}	1.29×2^{46}	1.23×2^{92}
SIMON 48/72	2	37152[a]	1.59×2^{49}	1.36×2^{50}	1.08×2^{100}
SPECK 32/64	3	72024[a]	1.78×2^{46}	1.43×2^{45}	1.27×2^{92}
SPECK 48/72	2	73488[a]	1.37×2^{51}	1.11×2^{50}	1.52×2^{101}
RECTANGLE 64/80	2	62284	1.79×2^{58}	1.40×2^{49}	1.25×2^{108}
LBLOCK 32/80	2	111296[a]	1.04×2^{56}	1.34×2^{52}	1.39×2^{108}
PRINT 48/80	2	62216	1.97×2^{58}	1.06×2^{50}	1.04×2^{108}
SLIM 32/80	2	121936	1.76×2^{56}	1.56×2^{53}	1.37×2^{110}
SLIM 32/80	3	182984	1.33×2^{57}	1.56×2^{53}	1.04×2^{111}

[a] Estimated from cited gate counts for completeness.

Table 5. `MAXDEPTH analysis of Grover's algorithm`.

CIPHER	r	2^{40}	2^{64}	2^{96}	approximation
SLIM 32/80	2	70	46	14	2^{111}/MAXDEPTH
SLIM 32/80	3	71	47	15	2^{110}/MAXDEPTH

Here, we now focus on inner parallelization to optimize total depth to achieve NIST's MAXDEPTH parameters [47]. We clear the channels using $SLIM^{-1}$, consistent with the discussion in section 3 for exhaustive key search. Given SLIM's Feistel structure, encryption and decryption processes exhibit identical quantum costs. Thus, for r = 2 and r = 3, we use 4 and 6 SLIM instances, respectively. Leveraging the optimized qubit count (q) for SLIM in table 2, total qubit requirement is $r \cdot q + 1$. Accordingly, U_f employs 113 ancilla qubits for r = 2 and 225 for r = 3. Comparing the 32-bit outputs of SLIM instances with r ciphertexts requires $80 \cdot r$ CNOT gates. Additionally, $2(r-1)k$ CNOT gates are required to execute input keys across r parallel SLIM instances. NOT gates arising from these optimizations are negligible. Hence, total additional CNOT gates are 320 (160+160) for r = 2 and 560 (240+320) for r = 3. Quantum resource estimates for Grover key search are given in table 3.

NIST has standardized Grover's algorithm as a reference benchmark for assessing BC resistance against quantum threats, defining the MAXDEPTH parameter to impose an upper bound on total circuit depth. To establish MAXDEPTH for SLIM, decomposition of the CCNOT gates listed in table 3 is required. The number of T-gate for a t-fold controlled NOT gate ($t \geqslant 5$) is calculated as $32t84$. Toffoli gates require 7 T-gates + 8 Clifford gates with T-depth 4 and total depth 8 [53]. The total number of Clifford gates includes those used in SLIM instances and $2(r1)k$ CNOT gates in U_f, which arise from parallel pair processing. Under $(\frac{\pi}{4}2^{40})$ iterations, the estimated total cost of a Grover-based attack against SLIM yields; for r = 2 the T-gate count of 1.13×2^{56}, the total Clifford gates of 1.76×2^{56}, and the full depth of 1.56×2^{53}, for r = 3 the T-gate count of 1.82×2^{56}, the total Clifford gates of 1.33×2^{57} with the same full depth.

A detailed analysis of the resource estimates for Grover's oracle given in table 4 confirms that SLIM demonstrates stronger resilience (2^{111}) against quantum adversaries compared to other LBCs. Furthermore, while LBLOCK exhibits a high gate cost in its quantum realization, its relatively lower circuit depth reduces its quantum security. This underscores the importance of the cipher design in determining the quantum resistance. The observed trade-off in SLIM between low-qubit usage and higher total cost and depth points to enhanced quantum resilience, reflecting a balanced design relevant to post-quantum security among its lightweight counterparts.

NIST characterizes post-quantum security in symmetric cryptography by the total cost of Grover's key search. This is quantified as the product of total gate count and circuit depth, derived from resource estimates for attacks against AES-128, AES-192, and AES-256, reflecting the algorithm's iterative nature. The estimated costs for AES correspond to 2^{170} (Level 1), 2^{233} (Level 3), and 2^{298}(Level 5) [54]. To ensure sufficient resilience against anticipated quantum threats, NIST sets forth a recommendation to adopt security parameters achieving at least Level 1 or 3 protection in the near term.

The total quantum cost of SLIM is approximately 2^{111} for $r = 3$, which is below NIST's Level 1 threshold of 2^{170} for symmetric cryptography (See table 5). Under the MAXDEPTH constraint, which bounds the maximum feasible circuit depth, SLIM's depth of about 2^{70} remains well below the recommended limit 2^{130} with the maximum number of gates expected to be implemented in one year in foreseeable quantum computer architectures 2^{40}. These results suggest that, while SLIM does not reach the NIST levels required for short-term quantum resistance, it compares favorably to other LBCs. Consequently, SLIM and the other evaluated LBCs

are unlikely to meet the security requirements for NIST Level 3 (2^{233}; the decade-long serial gate count for current architectures) or Level 5 (2^{298}; the atomic-scale computational limit imposed by the speed of light over a millennium) [47]. These findings highlight the urgent need to further improve the designs of LBCs to maintain their security in a post-quantum era.

6. Conclusion

In this study, we present a qubit optimized quantum implementation of SLIM, featuring a compact quantum architecture. Our gate-level analysis demonstrates efficient resource distribution across KSP/KSP^{-1} layers, with computational demands managed without auxiliary resources. This approach achieves a total of 112 qubits, a balanced quantum cost of 30404, and a depth of 4066—positioning SLIM as a resource-efficient solution among comparable BCs (table 2). This places SLIM as a viable candidate for lightweight cryptographic applications in quantum-constrained environments.

We then assess the quantum resilience of SLIM against Grover's search algorithm, leveraging a quadratic speedup for key recovery. The quantum oracle U_f encodes SLIM encryption under superposition, marking candidate keys x via phase inversion when $\mathrm{Enc}_x(m_i) = c_i$ for r plaintext-ciphertext pairs. To mitigate false positives from SLIM's 32-bit block and 80-bit key, $r = 3$ pairs are necessary, exceeding the k / n ratio. As the plaintext is encrypted via SLIM and is subsequently retrieved for channel cleaning, two SLIM instances (SLIM and SLIM^{1}) are required for each plaintext-ciphertext pair. Hence, the oracle integrates parallel six SLIM instances, ancilla qubits (337 total), and controlled comparisons, followed by clearing operations. The diffusion operator D amplifies solution states via amplitude amplification. Resource estimates (table 3) confirm U_f dominates the Grover operator $G = DU_f$, with depth-invariant diffusion.

Inner parallelization minimizes depth to comply with NIST's MAXDEPTH constraints. Total quantum cost —factoring the total counts of T and Clifford gates, and the total depth (table 4)—yields 1.33×2^{57} total gates and 1.56×2^{53} depth for r = 3, culminating in a total attack cost of 2^{111}. This falls below NIST's Level 1 security threshold (2^{170}) and MAXDEPTH limit (table 5), indicating vulnerability to near-term quantum attacks. Nevertheless, SLIM's low qubit footprint offers comparative advantages over other LBCs in quantum resource efficiency, albeit insufficient for post-quantum security certification. Despite other LBCs exhibiting lower quantum circuit costs (gate counts), SLIM's higher depth enables a superior total cost metric under the NIST framework. This demonstrates that quantum depth can be leveraged as a critical security parameter in the design with the total gate count.

We propose to adopt as a standard the implementation of depth-optimized Grover analyses with qubit-optimized BC implementations. This framework enables systematic evaluation of quantum attack resilience under NIST's MAXDEPTH guidelines. The proposed ancilla-free method, which relies on the efficient clearing of quantum circuits by leveraging the reversible nature of quantum logic gates, is not limited to SLIM. Its applicability extends to any Feistel-based cipher, regardless of block size, key length, or number of rounds, utilizing the reversibility of quantum logic gates. Future work will therefore focus on applying this framework to a wider range of Feistel-based ciphers. Such designs could enhance quantum security margins while preserving lightweight efficiency and would be instrumental in further improving the trade-off between quantum security and resource requirements.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

References

[1] Shor P W 1997 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM J Comput* **26** 1484–509
[2] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc of the 28th Annua ACM Symposium on Theory of Computing (STOC), ACM* 212–9
[3] Yamamura A and Ishizuka H 2000 Quantum cryptanalysis of block ciphers (Algebraic systems, formal languages and computations) *RIMS Kokyuroku* **1166** 235–43
[4] Simon D 1994 On the power of quantum computation *Proceedings of the 35th IEEE Symposium on the Foundations of Computer Science (FOCS)* 116–23
[5] Dowling J P and Gerard J M 2003 Quantum technology: the second quantum revolution *Phil. Trans. R. Soc. A* **361** 1655–74
[6] Harrow A W and Montanaro A 2017 Quantum computational supremacy *Nature* **549** 203–9
[7] Gao S, Ho-Ching Iu H, Erkan U, Simsek C, Toktas A, Cao Y, Wu R, Mou J, Li Q and Wang C 2025 A 3d memristive cubic map with dual discrete memristors: Design, implementation, and application in image encryption *IEEE Trans. Circuits Syst. Video Technol.* **35** 7706–18

[8] Gao S, Ding S, Iu H H-C, Erkan U, Toktas A, Simsek C, Wu R, Xu X, Cao Y and Mou J 2025 A three-dimensional memristor-based hyperchaotic map for pseudorandom number generation and multi-image encryption *Chaos* **35** 073105
[9] Gao S, Zhang Z, Iu H H-C, Ding S, Mou J, Erkan U, Toktas A, Li Q, Wang C and Cao Y 2025 A parallel color image encryption algorithm based on a 2-d logistic-rulkov neuron map *IEEE Internet of Things Journal* **12** 18115–24
[10] Gao S, Zhang Z, Li Q, Ding S, Iu H H-C, Cao Y, Xu X, Wang C and Mou J 2025 Encrypt a story: A video segment encryption method based on the discrete sinusoidal memristive rulkov neuron *IEEE Trans. Dependable Secure Comput.* **1545-5971** 1–15
[11] Gao S, Wu R, Iu H H, Erkan U, Cao Y, Li Q, Toktas A and Mou J 2025 Chaos-based video encryption techniques: A review *Computer Science Review* **58** 100816
[12] Daemen J and Rijmen V 2001 *Specification for.: the Advanced Encryption Standard (AES). FIPS 197* **197** upd1
[13] Li Z, Cai B, Sun H, Liu H, Wan L, Qin S, Wen Q and Gao F 2022 Novel quantum circuit implementation of advanced encryption standard with low costs *Science China Physics, Mechanics and Astronomy* **65** 290–311
[14] Huang Z and Sun S 2022 Synthesizing Quantum Circuits of AES with Lower T-depth and Less Qubits *Advances in Cryptology ASIACRYPT 2022* ed S Agrawal and D Lin (Springer) 614–44
[15] Almazrooie M, Samsudin A, Abdullah R and Mutter K N 2018 Quantum reversible circuit of AES-128 *Quantum Inf. Process.* **17** 1–30
[16] Langenberg B, Pham H and Steinwandt R 2020 Reducing the cost of implementing the advanced encryption standard as a quantum circuit *IEEE Transactions on Quantum Engineering* **1** 1–12
[17] Luo Q-B, Yang G-W, Li X-Y and Li Q 2022 Quantum reversible circuits for multiplicative inverse *EPJ Quantum Technology* **9** 24
[18] Selinger P 2013 Quantum circuits of T-depth one *Physical Review A'Atomic, Molecular, and Optical Physics* **87** 042302
[19] Dasu V A, Baksi A, Sarkar S and Chattopadhyay A 2019 LIGHTER-R: optimized reversible circuit implementation for SBoxes *2019 32nd IEEE International System-on-Chip Conference (SOCC)* (IEEE) 260–5
[20] Anand R, Maitra A and Mukhopadhyay S 2020 Grover on SIMON *Quantum Inf. Process.* **19** 1–17
[21] Anand R, Maitra A and Mukhopadhyay S 2020 Evaluation of quantum cryptanalysis on SPECK *Cryptology -INDOCRYPT 2020.* **12578** 395–413
[22] Saravanan P, Jenitha J, Aasish S and Sanjana S 2021 Quantum circuit design of RECTANGLE lightweight cipher *2021 25th International Symposium on VLSI Design and Test (VDAT)* (IEEE) 1–4
[23] Lee W-K, Jang K, Song G, Kim H, Hwang S O and Seo H 2022 Efficient implementation of lightweight hash functions on GPU and quantum computers for IoT applications *IEEE Access* **10** 59661–74
[24] Jing X, Li Y, Zhao G, Xie H and Wang Q 2023 Quantum circuit implementation and resource analysis of LBlock and LiCi *Quantum Inf. Process.* **22** 347
[25] Paramasivam S, Jenitha J, Sanjana S and Haghparast M 2023 Compact quantum circuit design of PUFFIN and PRINT lightweight ciphers for quantum key recovery attack *IEEE Access* **11** 66767–76
[26] Luo Q B, Li Q, Li X Y, Yang G W, Shen J and Zheng M 2024 Quantum circuit implementations of SM4 block cipher optimizing the number of qubits *Quantum Inf. Process.* **23** 177
[27] Zheng Y, Luo Q and Li Q 2024 Quantum circuit implementations of lightweight authenticated encryption ascon *J Supercomput* 11322–37
[28] Chun M, Baksi A and Chattopadhyay A 2023 DORCIS: Depth optimized quantum implementation of substitution boxes *Cryptology ePrint Archive*
[29] Ding L, Luo Q, Lv Y, Zheng Y, Liao H and Chen Z 2025 Quantum circuit implementation and security analysis of ivlbc *Class. Quantum Grav.* **42** 085006
[30] Feistel H 1973 Cryptography and computer privacy *Sci. Am.* **228** 15–23
[31] (FIPS) 1999 *FIPS: Data Encryption Standard NBS* 46–3
[32] (FIPS) 2001 FIPS: Advanced encryption standard *NBS* **197**
[33] Jaques S, Naehrig M, Roetteler M and Virdia F 2020 Implementing grover oracles for quantum key search on AES and LowMC *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques* 30 (Springer) 280–310 May 10–14, 2020, Proceedings, Part II
[34] Schlieper L 2020 In-place implementation of quantum-gimli arXiv:2007.06319 2 Nov 2021
[35] Pal O, Jain M, Murthy B and Thakur V 2022 Quantum and Post-Quantum Cryptography *Cyber Security and Digital Forensics: Challenges and Future Trends* **1** 45–58
[36] Chauhan A K and Sanadhya S K 2020 Quantum resource estimates of Grover's key search on ARIA *In: Proc. Int. Conf. Secur. Privacy Appl. Cryptogr. Eng.* **12586** 238–58
[37] Bogdanov A, Knudsen L R, Leander G, Paar C, Poschmann A, Robshaw M J, Seurin Y and Vikkelsoe C 2007 PRESENT: An ultra-lightweight block cipher *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings 9* 4727 (Springer) 450–66
[38] Hong D *et al* 2006 HIGHT: A new block cipher suitable for low-resource device *Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop* **8** 46–59
[39] Knudsen L R *et al* 2010 PRINT-cipher: A block cipher for ic-printing *In: Proc. 12th Int. Workshop in Lecture Notes in Computer Science* **6225** 16–32 USA
[40] Guo J *et al* 2011 The PHOTON family of lightweight hash functions ed P Rogaway *Proc. 31st Annu. Int. Cryptol. Conf. (CRYPTO), in Lecture Notes in Computer Science* 6841, 222239(Springer)
[41] Beaulieu R *et al* 2015 The SIMON and SPECK lightweight block ciphers *In: Proc. 52nd Annu. Design Autom. Conf.* 1–6
[42] Zhang W *et al* 2014 RECTANGLE: A bit-slice ultra-lightweight block cipher suitable for multiple platforms *In: Proc. IACR* **58** 1–15
[43] Cheng H *et al* 2008 PUFFIN: A novel compact block cipher targeted to embedded digital systems *In: Proc. 11th EUROMICRO Conf. Digit. Syst. Des. Archit. Methods Tools* **34** 383–90
[44] Wu W and Zhang L 2011 LBlock: a lightweight block cipher *In: App. Cryp. Net. Sec. 9th Int. Conf. ACNS 2011, Nerja, Spain* 327–44
[45] Patil J, Bansod G and Kant K S 2017 LiCi: a new ultra-lightweight block cipher *In: Int. Conf. on Emerging Trends and Innov ICT* 40–5
[46] Aboushosha B, Ramadan R, Dwivedi A, El-Sayed A and Dessouky M 2020 SLIM: A lightweight block cipher for internet of health things *IEEE Access* **8** 203747–57
[47] (NIST) 2016 *Project for Post-Quantum Cryptography Standardization. Call for Proposals: Post-quantum Cryptography (NIST IR 8105). U.S. Department of Commerce*
[48] Yadav T, Kumar M, Kumar A and Pal S 2023 A practical-quantum differential attack on block ciphers *Cryptography and Communications* **17** 337–357
[49] Elaine B 2020 Recommendation for Key Management: Part 1 - General, NIST Special Publication 800-57 Part 1 Revision 5

[50] Mohanapriya R and Kumar N 2023 Optimized Implementation of S-box and Inverse S-box for PRESENT Lightweight Block Cipher *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)* (IEEE) 1–5
[51] Shende V and Markov I 2009 On the CNOT-cost of TOFFOLI gates *Quantum Inf. Comput.* **9** 461–86
[52] Jaques S, Naehrig M, Roetteler M and Virdia F 2020 Implementing Grover oracles for quantum key search on AES and LowMC *In Advances in Cryptology-EUROCRYPT 2020* **12106** 280–310
[53] Nathan W and Roetteler M 2016 Quantum arithmetic and numerical analysis using repeat-until-success circuits *Quantum Inf. Comput.* **16** 134–78
[54] Grassl M, Langenberg B, Roetteler M and Steinwandt R 2016 Applying Grover's algorithm to AES: quantum resource estimates *PQCRYPTO 16* **9606** 29–43