Physica Scripta

INCLUSIVE PUBLISHING TRUSTED SCIENCE

PURPOSE-LED PUBLISHING™

PAPER • **OPEN ACCESS**

Quantum circuit realization and Grover cryptanalysis of the hybrid ARX-SPN cipher GFSPX

To cite this article: Ibrahim Ulgen *et al* 2026 *Phys. Scr.* **101** 285103

View the article online for updates and enhancements.

You may also like

- Quantum implementation of SLIM and its Grover cryptanalysis
H O Cildiroglu and O Yayla
- Quantum circuit implementation and security analysis of IVLBC
Lang Ding, Qingbin Luo, Yi Lv et al.
- Efficient Implementation for PRINCE Algorithm in FPGA Based on the BB84 Protocol
Alharith A. Abdullah and Noor R. Obeid

This content was downloaded from IP address 144.122.201.61 on 30/07/2026 at 13:18

Physica Scripta

Check for updates

PAPER

OPEN ACCESS

RECEIVED
13 May 2026

REVISED
2 July 2026

ACCEPTED FOR PUBLICATION
6 July 2026

PUBLISHED
15 July 2026

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Quantum circuit realization and Grover cryptanalysis of the hybrid ARX-SPN cipher GFSPX

Ibrahim Ulgen[1,3,*], **Hasan Ozgur Cildiroglu**[2] and **Oğuz Yayla**[1]

[1] Institute of Applied Mathematics, Middle East Technical University, Universiteler, Çankaya, 06800 Ankara, Türkiye
[2] Physics Department, Ankara University, Dogol st, 06100 Ankara, Türkiye
[3] Department of Mathematics, Siirt University, Universite st, 56100 Siirt, Türkiye
[*] Author to whom any correspondence should be addressed.

E-mail: ibrahim.ulgen@siirt.edu.tr, cildiroglu@ankara.edu.tr and oguz@metu.edu.tr

Keywords: GFSPX, lightweight cryptography, quantum circuit implementation, Grover's algorithm, ARX-SPN, post-quantum security

Abstract

The security of classical symmetric-key primitives is fundamentally challenged by the emergence of quantum computing, necessitating a rigorous evaluation of their post-quantum resilience. This paper presents a comprehensive quantum circuit realization and Grover cryptanalysis of GFSPX, a lightweight block cipher featuring a 64-bit data block and a 128-bit secret key. GFSPX utilizes a unique hybrid architecture that integrates a 4-branch generalized Feistel structure with both Addition-Rotation-XOR (ARX) and Substitution-Permutation Network (SPN) components. Our quantum implementation optimizes resource distribution by exploiting the inherent reversibility of the Feistel network and employing a compact ripple-carry adder for the ARX layers. The proposed architecture achieves a qubit-optimized footprint of 209 qubits with a baseline quantum cost of 32 498 and a circuit depth of 7617. To evaluate the cipher's resistance against quantum adversaries, we construct a parallelized Grover oracle using three plaintext–ciphertext pairs to eliminate spurious matches. Our analysis reveals that the total quantum cost of a key-recovery attack on GFSPX is 1.12×2^{159} quantum gates. Although this cost falls below the NIST Level 1 security threshold of 2^{170}, the hybrid ARX-SPN design demonstrates a higher quantum attack resistance among other lightweight designs. These findings provide critical insights into the balance between classical efficiency and quantum resilience in next-generation cryptographic designs for resource-constrained environments.

1. Introduction

Cryptography provides a formal framework to ensure secure communication, authentication, and data integrity. Conventionally, the security of cryptographic primitives is grounded in the computational intractability of specific mathematical problems. The emergence of quantum information processing challenges classical security models, as the coherent manipulation of quantum states provides physical mechanisms to overcome established limits [1–8]. Shor's algorithm [9] compromises widely deployed public-key schemes by efficiently solving integer factorization and discrete logarithms [10–13]. Grover's algorithm [14] yields a quadratic speedup for unstructured search, effectively halving the security margin of symmetric-key constructions [15]. Consequently, the resilience of existing cryptographic primitives necessitates rigorous re-evaluation within frameworks that explicitly exploit quantum resources [16–18].

In this post-quantum landscape, symmetric block ciphers (BCs) remain fundamentally viable. Their resilience relies on iterative, key-dependent substitutions and permutations that establish rigorous confusion and diffusion [19]. The Feistel network constitutes a core design paradigm for symmetric BCs, enabling block splitting, subkey injection, and iterative structures to perform both encryption and decryption with the same key [20, 21]. For resource-constrained environments, such as the Internet of Things (IoT) and RFID systems, lightweight BCs are specifically designed to strike a balance between robust security and hardware efficiency [22–26]. Among the core design paradigms,

© 2026 The Author(s). Published by IOP Publishing Ltd

Addition-Rotation-XOR (ARX) and Substitution-Permutation Networks (SPNs) are particularly prominent. ARX designs employ simple operations to achieve non-linearity without relying on pre-defined S-boxes, reducing storage overhead, while SPN architectures provide rapid diffusion [27, 28].

GFSPX is a lightweight cipher that synthesizes a 4-branch generalized Feistel structure (GFS) with a hybrid ARX-SPN paradigm [29]. GFSPX utilizes two distinct functional blocks: an ARX-based function (F_1), designed to achieve non-linearity, and a 32-bit SPN structure (F_2), which incorporates the PRESENT's S-box and bit-level permutations to facilitate rapid and comprehensive diffusion across all data paths [30]. This hybrid approach enables GFSPX to achieve a full avalanche effect in six rounds, addressing the slow diffusion properties of pure ARX ciphers while maintaining a hardware footprint comparable to standardized lightweight primitives [31]. The inclusion of modular addition in this hybrid design further motivates an investigation into its impact on Grover's attack complexity.

Quantifying the post-quantum resilience of a cryptographic primitive fundamentally requires a detailed mapping into a quantum circuit [32, 33]. Such a realization enables the construction of the Grover oracle and facilitates a precise estimation of the quantum resources used to mount a key-recovery attack. In this work, we present a comprehensive quantum circuit realization of GFSPX. Our implementation focuses on minimizing the qubit count and circuit depth by exploiting the reversibility of the Feistel structure and optimizing the gate-level architecture of both the modular addition and substitution layers. We demonstrate a resource-efficient design that utilizes 209 qubits and achieves an optimized gate-to-depth ratio of 4.27. While the full-scale circuit exceeds the capabilities of current NISQ devices, the proposed architecture is designed to be compatible with fault-tolerant quantum computing frameworks and provides a rigorous basis for quantum resource estimation under the NIST `MAXDEPTH` constraints.

Furthermore, we quantitatively evaluate the quantum security of GFSPX against Grover's algorithm within the NIST `MAXDEPTH` framework [32]. To eliminate spurious matches in the 128-bit key space, we construct a parallelized Grover oracle utilizing three plaintext–ciphertext pairs. The resulting attack requires 628 qubits and exhibits a total quantum cost of approximately 1.12×2^{159} with its comparison with respect to other lightweight ciphers. Although this estimated cost falls below NIST's Level 1 security threshold of 2^{170} quantum gates, our analysis indicates that GFSPX does not meet the strict post-quantum criteria for security against a sufficiently advanced quantum adversary. However, its optimized quantum circuit implementation and relatively high temporal depth offer a competitive quantum cost compared to other lightweight BCs, providing valuable insights for future quantum-resistant lightweight designs. Moreover, our resource estimation confirms that the modular addition-based ARX primitive relatively increases the Grover attack complexity. These findings provide insights for the development of future quantum-resistant lightweight cryptographic designs.

The remainder of this paper is organized as follows: section 2 details the classical structure of GFSPX, including its hybrid ARX-SPN round functions and key schedule. Section 3 describes the proposed quantum circuit implementation of the cipher. Section 4 provides a comprehensive quantum resource estimation and comparison with other lightweight BCs. Finally, section 5 analyzes the security of GFSPX against Grover's algorithm and concludes with resource estimates under the `MAXDEPTH` constraint.

2. The structure of GFSPX

GFSPX is a symmetric-key lightweight BC designed for resource-constrained IoT environments, operating on 64-bit data blocks with a 128-bit secret key [29] (See figure 1). The encryption process consists of 20 rounds, using a hybrid architecture that integrates a 4-branch GFS with ARX-SPN components. 64-bit plaintext is divided into four 16-bit branches, $\{D_0, D_1, U_0, U_1\}$, ordered from the most significant bits (MSBs) to the least significant bits (LSBs). These segments are subsequently processed through two distinct nonlinear functional blocks. To ensure adequate diffusion across all data paths, a branch permutation is applied at the end of each cycle, except for the final round.

The round function: The GFSPX round function is characterized by the integration of the F_1 and F_2 functional blocks. The function $F_1 : \{0,1\}^{16} \rightarrow \{0,1\}^{16}$ is defined as $F_1(X) = \big(X \boxplus (X \lll a)\big) \oplus (X \lll b), \text{with} a = 5,\ b = 1$. The evaluation of F_1 proceeds in three sequential stages. First, two linear shifts are performed, yielding $R_1 = X \lll 5$ and $R_2 = X \lll 1$. Second, a modular addition is carried out, $S = X \boxplus R_1 = (X + R_1) \bmod 2^{16}$, which constitutes the sole source of non-linearity within F_1. Third, the output is obtained as $Y = S \oplus R_2$. Here, $\oplus$ denotes bitwise XOR, $\boxplus$ stands for modular addition over $\mathbb{Z}/2^n\mathbb{Z}$, and $X \lll t$ represents the center circular (cyclic) shift of X by t bit positions. Complementing

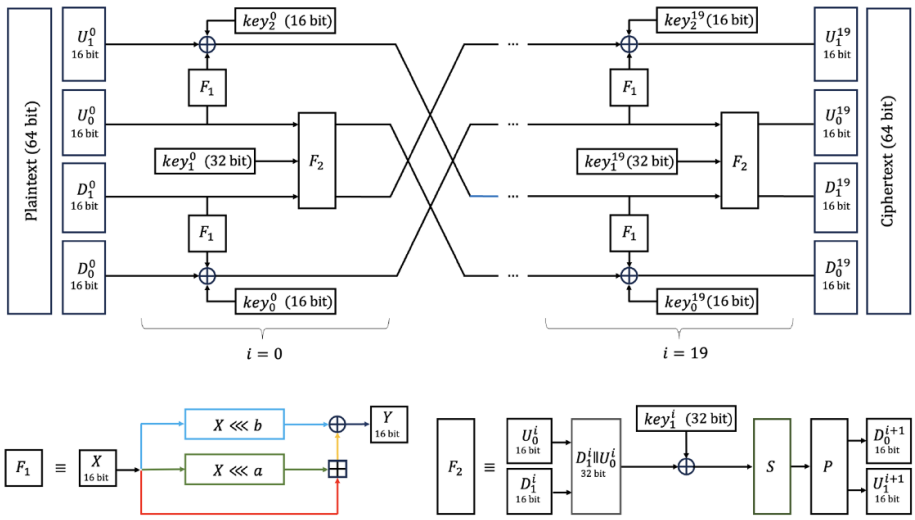

Figure 1. Core round functions of GFSPX [29]: $D_1^{i+1} = F_1(U_0^i) \oplus U_1^i \oplus \text{key}_2^i$, $U_0^{i+1} = F_1(D_1^i) \oplus D_0^i \oplus \text{key}_0^i$, and $(U_1^{i+1}, D_0^{i+1}) = F_2(D_1^i \| U_0^i, \text{key}_1^i)$.

Table 1. *S*-box (Substitution-box) of GFSPX.

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

Table 2. Permutation table of GFSPX.

i	$P(i)$	i	$P(i)$	i	$P(i)$	i	$P(i)$
0	0	8	2	16	4	24	6
1	8	9	10	17	12	25	14
2	16	10	18	18	20	26	22
3	24	11	26	19	28	27	30
4	1	12	3	20	5	28	7
5	9	13	11	21	13	29	15
6	17	14	19	22	21	30	23
7	25	15	27	23	29	31	31

the ARX operations, the F_2 function is designed as a 32-bit SPN structure. The process begins with the bitwise XOR of the input and a 32-bit subkey (key_1^i), followed by a substitution layer comprising eight parallel 4-bit *S*-boxes consistent with the *S*-box definitions specified in the PRESENT [30], see table 1. Last, a bit-level permutation layer (*P*-layer) is applied to facilitate effective diffusion throughout the internal state (table 2). By integrating this hybrid design, GFSPX achieves a full avalanche effect in just six rounds [29].

Key schedule: The key scheduling algorithm operates on a 128-bit master key register $K^0 = (k_{127}, \ldots, k_0)$ from MSB to LSB (see figure 2). The initial round subkeys key_0^i, key_1^i, and key_2^i are directly extracted as contiguous segments of K^0: $\text{key}_0^0 = K^0[127:112]$, $\text{key}_1^0 = K^0[111:80]$, and $\text{key}_2^0 = K^0[79:64]$. Following subkey extraction, the register is transformed for the subsequent iteration through a series of operations. First, the register undergoes a center cyclic shift of 113 bits ($\text{temp}_0 = K^{i-1} \lll 113$ for $i = 1, 2, \ldots, 19$). Next, the 8 MSBs are processed through the *S*-box in two 4-bit nibbles: $\text{temp}_1 = \mathcal{S}(\text{temp}_0[127:124])$ and $\text{temp}_2 = \mathcal{S}(\text{temp}_0[123:120])$. Subsequently, a round counter (RC) is XORed with a specific bit-slice, targeting $\text{temp}_0[14:10]$, to produce the updated segment $\text{temp}_3 = \text{temp}_0[14:10] \oplus RC$. The succeeding key state, K^i, is constructed by concatenating these processed segments: $K^i = \text{temp}_1 \,||\, \text{temp}_2 \,||\, [119:15] \,||\, \text{temp}_3 \,||\, [9:0]$ for $i = 1, 2, \ldots, 19$.

The decryption process relies on the inherent reversibility of the GFS. While the ARX-based function F_1 is applied directly, the SPN-based F_2 function requires the application of its inverse, F_2^{-1}. This inversion incorporates S^{-1}-box and P^{-1}-layer, which are derived by reversing their respective mappings

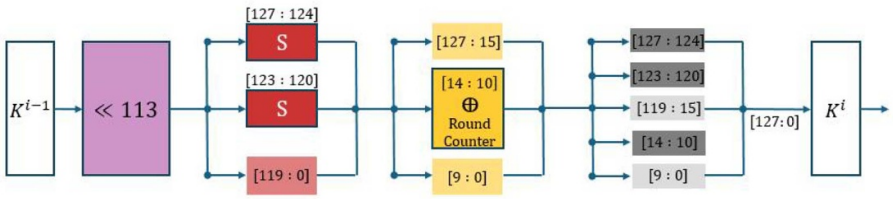

Figure 2. Key scheduling mechanism of GFSPX.

as defined in tables 1 and 2. Furthermore, the subkeys derived from the key schedule are introduced in reverse order to recover the original plaintext.

3. Quantum realization of the GFSPX

Evaluating the security of the GFSPX against Grover attacks requires an estimation of the quantum resources. Since the complexity of such attacks is fundamentally governed by the quantum gate cost and circuit depth of the target primitive, constructing an optimized quantum implementation is a prerequisite for rigorous security analysis. Therefore, we present the quantum realization of GFSPX by mapping its classical structure into a reversible quantum circuit. To minimize key quantum resources, we exploit the inherent reversibility of the Feistel network and optimize the gate-level architecture of the ARX and SPN components.

Grover's algorithm [14] is a quantum search procedure that provides a provable quadratic speedup over classical exhaustive search. For an unstructured search space of size $N = 2^k$, the algorithm identifies a unique target state using only $O(2^{k/2})$ evaluations of a quantum oracle, compared to $O(2^k)$ evaluations required classically. This speedup has a direct implication for symmetric-key cryptography: a cipher secured by a k-bit key, which classically requires $O(2^k)$ operations to break by brute force, can in principle be attacked in $O(2^{k/2})$ quantum operations, effectively halving the security margin [15].

The algorithm operates on a k-qubit register initialized in a uniform superposition $|\Psi\rangle = 2^{-k/2}\sum_{x\in\{0,1\}^k}|x\rangle$, which encodes all 2^k candidate keys simultaneously. A single Grover iteration consists of two operations: (i) the quantum oracle U_f, which encodes the target condition by applying a conditional phase inversion that transforms $|x\rangle \to -|x\rangle$ if x is the correct key while leaving all other states unchanged, and (ii) the diffusion operator $D = 2|\Psi\rangle\langle\Psi| - I$, which performs an inversion about the mean amplitude, thereby amplifying the probability of measuring the target state. After approximately $j \approx (\pi/4)\sqrt{N}$ iterations of $G = DU_f$, the correct key can be recovered with near-unity probability.

In the context of BC cryptanalysis, the oracle U_f is realized as a reversible quantum circuit implementing the target cipher. Given a known plaintext m, the oracle executes the encryption $E_x(m)$ for each candidate key x in superposition and applies a phase flip conditioned on the resulting ciphertext matching a known reference value c. Since the key space (2^k) generally exceeds the ciphertext space (2^n), a single plaintext–ciphertext pair is insufficient in practice: multiple distinct keys may map m to the same ciphertext c, producing false positives known as spurious key candidates. Eliminating these requires evaluating $r > k/n$ plaintext–ciphertext pairs simultaneously within the oracle, where k is the key size and n is the block size [34]. For GFSPX, with $k = 128$ and $n = 64$, this condition yields $r = 3$, meaning the oracle must verify three independent pairs concurrently before triggering the phase inversion.

The quantum circuit realization of GFSPX presented in this section serves directly as the computational core of this oracle: each candidate key is loaded into the key register, the cipher is evaluated reversibly, and the output is compared against the known ciphertexts. Following the comparison, the intermediate computation is uncomputed (exploiting the inherent reversibility of the Feistel structure) to restore the ancilla qubits to their initial state $|0\rangle$, ensuring that the superposition over the key space remains coherent throughout all Grover iterations. Building on this foundation, the remainder of this section presents the gate-level construction of the GFSPX quantum circuit.

The proposed quantum architecture (figure 3) operates on a 64-qubit input, which is initially divided into four 16-qubit subsystems, $\{D_0^0, D_1^0, U_0^0, U_1^0\}$, ordered from the most significant qubits (MSQs) to the least significant qubits. It incorporates an auxiliary 16-qubit register to store intermediate states during the F_1 operations. Within a given round i, the F_1 operator first acts on the U_0^i subsystem using the

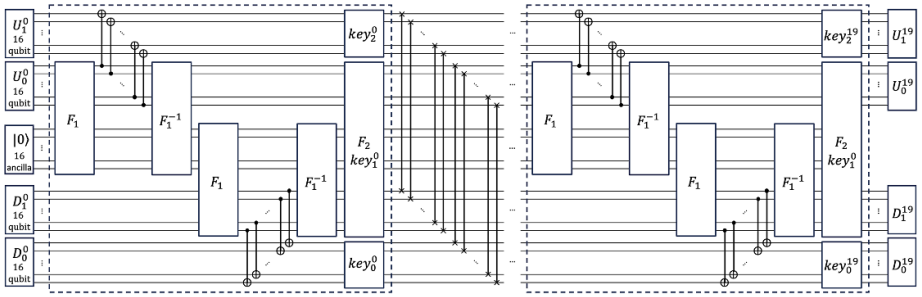

Figure 3. The complete quantum circuit for the GFSPX.

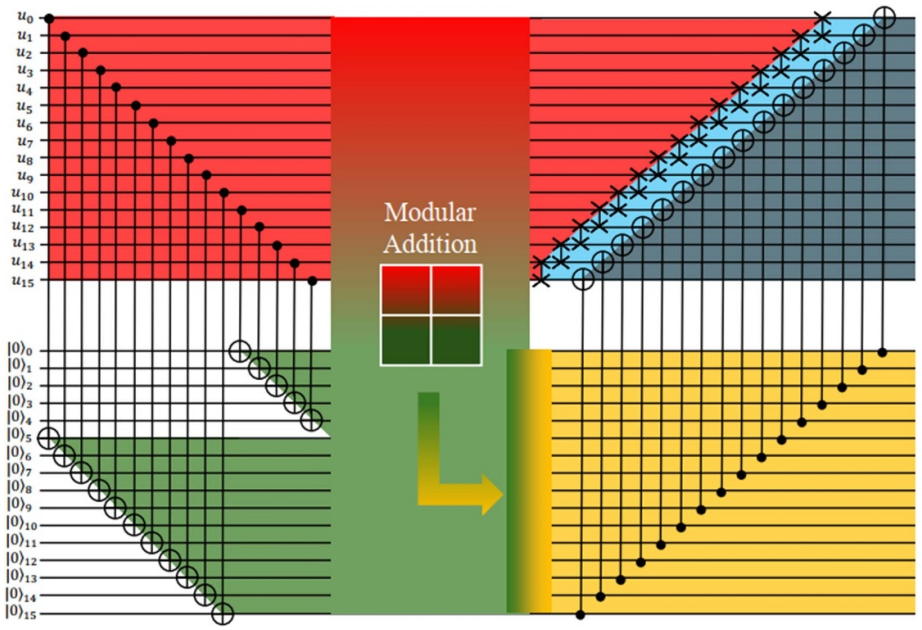

Figure 4. Quantum realization of the ARX-based F_1 function.

ancilla qubits, and the outcome is CNOTed into the U_1^i register. To recover the initial U_0^i and ancillaries $|0\rangle^{\otimes 16}$, a subsequent uncomputation F_1^{-1} is performed. This sequence is then applied to the D_1^i subsystem. Following these operations, the F_2 operator is applied to both U_0^i and D_1^i subsystems, while the round-specific subkeys derived from the key schedule ($\text{key}_2^i, \text{key}_1^i, \text{and}\, \text{key}_0^i$) are simultaneously CNOTed into the U_1^i, F_2, and D_0^i layers. Thus, the i-th round is completed. To ensure diffusion before the subsequent round commences, a permutation of the subsystems is executed, where the U_1^i and U_0^i registers are mapped to D_1^{i+1} and D_0^{i+1}, respectively. This protocol is applied in all rounds except for the final round, which produces the ciphertext.

***F_1* function (16-qubit ARX)**: The quantum circuit for the F_1 function is designed to implement $F_1(X) = (X \boxplus (X \lll 5)) \oplus (X \lll 1)$ while prioritizing resource efficiency. As illustrated in figure 4, the process initiates with the preparation of operands for 16-bit modular addition. Specifically, the primary register X serves as the control to map the 5-bit center-cyclically shifted state, $X \lll 5$, onto a 16-qubit ancilla target initialized to $|0\rangle^{\otimes 16}$ via bitwise CNOT operations.

The n-bit modular addition (The quantum ripple-carry adder (RCA)): The overall computational complexity of the F_1 operator is primarily driven by its arithmetic layer, we employ the quantum RCA architecture proposed by Cuccaro *et al* [35]. This approach offers a linear-depth solution that minimizes qubit overhead, making its minimal-ancilla property particularly well-suited for our proposed architecture.

The efficiency of the quantum RCA stems from a symmetric, two-phase computational structure facilitated by two compact subroutines. The first is the in-place majority (MAJ) gate, which computes $c_{i+1} = \text{MAJ}(a_i, b_i, c_i) = a_i b_i \oplus a_i c_i \oplus b_i c_i$ using two CNOT gates followed by a single Toffoli gate,

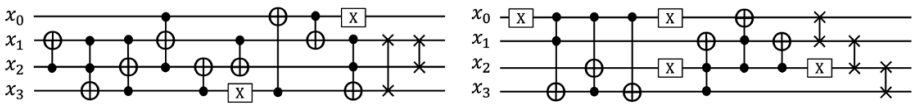

Figure 5. Quantum implementation of S-box (center) and S^{-1}-box (right).

where a_i and b_i denote the i-th bits of the n-bit input operands a and b, and c_i represents the carry bit propagated from the preceding bit position; this construction advances the carry chain in the forward pass without consuming additional ancillae. The second is the UnMajority-and-Add gate, which performs three operations within a single compact structure: it reverses the action of the MAJ gate to restore the input bit a_i to its original location, returns the carry c_i to the preceding register, and simultaneously writes the sum bit $s_i = a_i \oplus b_i \oplus c_i$ to the corresponding output position. The forward pass thus propagates the carry chain from the least significant to the MSB, while the backward pass uncomputes the intermediate carries and produces the sum bits in a reverse-order ripple, ensuring that the entire computation remains reversible.

In its standard form, the Cuccaro RCA achieves a depth of $2n+4$ using $2n-1$ Toffoli and $5n-3$ CNOT gates. For the present application, we adopt the modulo-2^n variant, where the most significant carry bit c_n is omitted as it is irrelevant for modular addition. This modification reduces the resource requirements to $2n-3$ CCNOT gates, $5n-7$ CNOT gates, and $2n-6$ NOT gates, with a reduced depth of $2n+2$.

In our 16-qubit implementation, this design choice yields a concrete resource budget of 29 CCNOT gates, 73 CNOT gates, 26 NOT gates, and a circuit depth of 34 for the arithmetic layer. Beyond reducing the qubit count, omitting the terminal carry bit eliminates the final Toffoli operation, directly enhancing the temporal efficiency of the F_1 operator. The remaining linear components—circular shifts and bitwise XORs—are integrated via SWAP and CNOT gates, respectively (see figure 4). Accordingly, the total gate complexity for the F_1 operator is characterized by 26 NOT, 105 CNOT, 29 CCNOT, and 15 SWAP gates.

F_1^{-1} function (16-qubit ARX): The inverse of F_1 uncomputes the function to restore the initial state, thereby inherently preserving the gate configuration and architectural complexity of the forward operation. This uncomputation process requires 26 NOT, 105 CNOT, 29 CCNOT, and 15 SWAP gates, mirroring the resource requirements of the F_1 operator as detailed in [35].

F_2 function (32-bit SPN): The F_2 function executes a sequential SPN transformation on the recovered D_1^i and U_0^i subsystems. This transformation is realized through three sequential quantum layers:

Round Subkey Addition: The 32-qubit round subkey (key_1^i) is integrated into the 32-qubit concatenated ($D_1^i \parallel U_0^i$) via 32 parallel CNOT gates in figure 6.

S-box: The S-box layer used in GFSPX in table 1, is adopted from the PRESENT algorithm [30] and its quantum implementation has been synthesized by the LIGHTER-R framework [36–38]. As illustrated in figure 5, the quantum implementation of the S-box features a specific optimization that decomposes the non-linear permutation into a minimum set of CNOT and Toffoli gates without requiring additional ancilla qubits. The inverse transformation of the S-box is illustrated in figure 5. The quantum resource requirements for the S-box implementation comprise 2 NOT gates, 5 CNOT gates, 4 CCNOT (Toffoli) gates, and 2 SWAP gates. Conversely, the inverse S-box (S^{-1}) is realized through a configuration of 4 NOT, 2 CNOT, 4 CCNOT, and 3 SWAP gates.

P-layer: Both P and P^{-1} layers, detailed in table 2, are implemented by systematically reordering the 32-qubit state using SWAP gates (figure 7). This 32-bit layer is realized using 24 SWAP gates organized into six parallelizable quartets. Since these quartets are logically independent, they facilitate simultaneous execution across the register. This parallelization strategy is critical for resource optimization as it allows the entire permutation to be executed without increasing the overall circuit depth.

Dynamic key-schedule: The GFSPX key schedule derives round-specific subkeys by unitarily evolving a 128-qubit master register. It initiates with a 113-bit center circular rotation (LCR) of the 128-qubit register. To minimize circuit depth, this cyclic shift is synthesized using 127 SWAP gates arranged into nine parallel execution layers (see figure 8). Following the cyclic shift, the eight MSQ are partitioned into two 4-bit segments ($\text{temp}_1, \text{temp}_2$) and routed through parallel S-boxes to induce non-linearity. Concurrently, a round-counter layer of NOT gates evaluates the $[k_{14} : k_{10}]$ subsystem (table 3). The

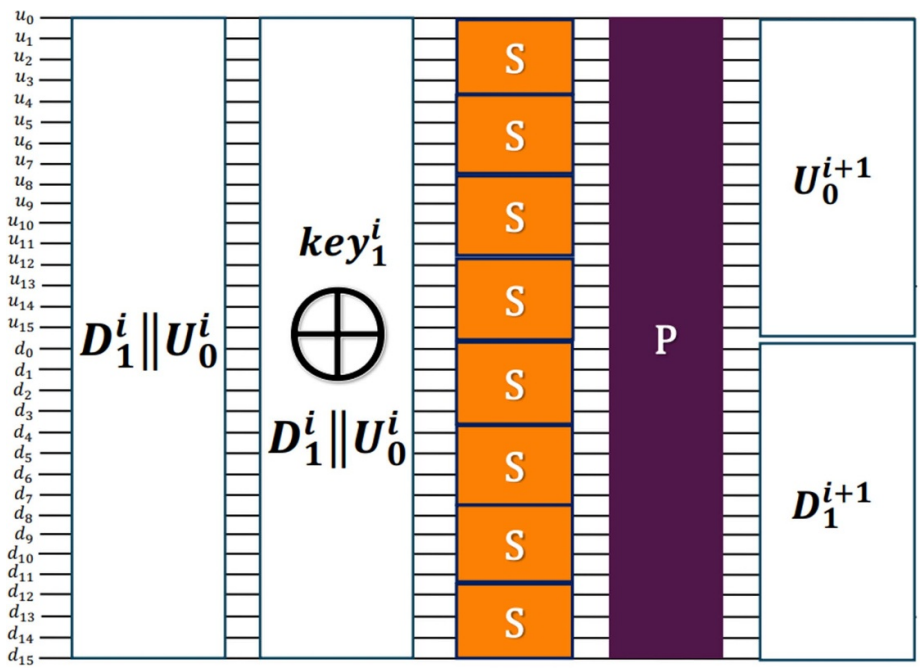

Figure 6. Quantum realization of the SPN-based F_2 function. It yields a total resource cost of 16 NOT, 72 CNOT, 32 CCNOT, and 40 SWAP gates.

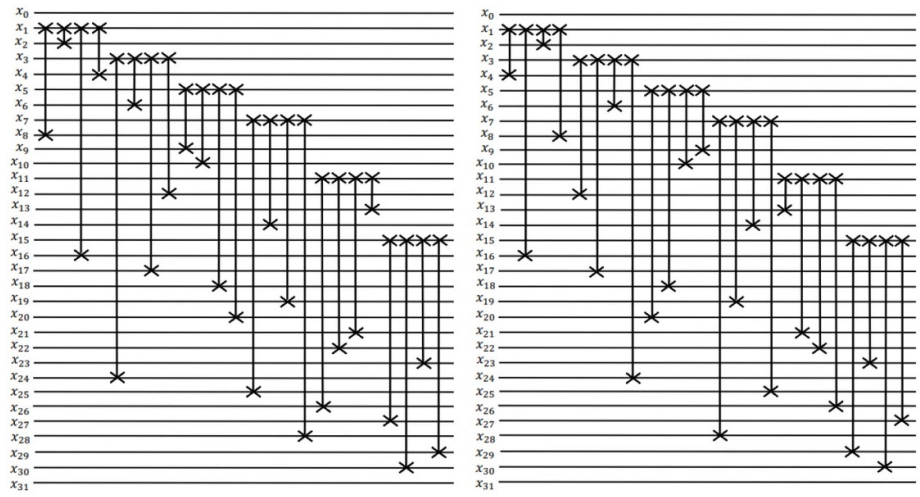

Figure 7. Quantum implementations of (a) P-box and (b) P^{-1}-box.

resulting 128-qubit configuration inherently provides the required 64-qubit subkeys, $\text{key}_0^i, \text{key}_1^i, \text{key}_2^i$, for rounds $i \in \{1, 2, \ldots, 19\}$ without additional gate overhead (figure 8). The inherent reversibility of the Feistel structure facilitates Grover's oracle, permitting the uncomputation of the schedule to recover the initial key state without auxiliary qubit overhead. Thus, the subkeys are provided in reverse for the decryption process.

The round function: As highlighted by the dashed boundaries in figure 3, each round function is a composition of two F_1 operators, their corresponding uncomputation counterparts ($2 \times F_1^{-1}$), a single F_2 primitive, and 32 CNOT gates. These components are integrated within a round-dependent sub-key layer that utilizes a dual-path injection strategy. While key_0^i and key_2^i are applied to the state via 32 CNOT gates, and key_1^i is embedded within the F_2 functional block, all subkeys are integrated into the circuit simultaneously. To facilitate the iterative execution, each round function, except for the final iteration, is linked with the succeeding round through 32 SWAP gates, as indicated in the circuit diagram, see figure 3.

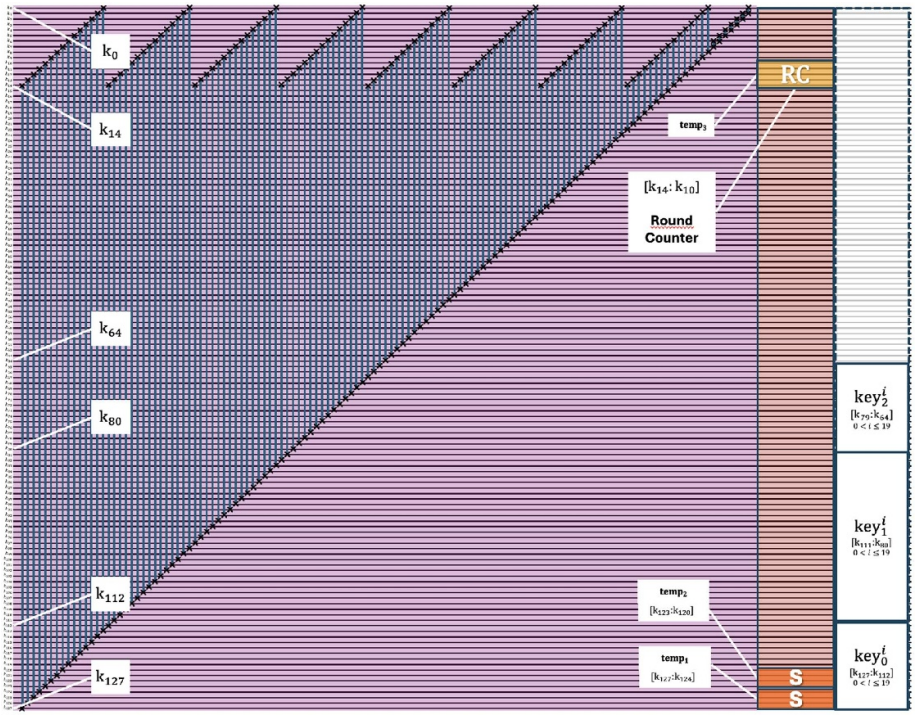

Figure 8. Quantum implementation of the GFSPX key schedule.

Table 3. Round counter algorithm for GFSPX.

$k_Q \setminus$ RC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
k_{14}																	X	X	X	X
k_{13}									X	X	X	X	X	X	X	X				
k_{12}					X	X	X	X					X	X	X	X				
k_{11}			X	X			X	X			X	X			X	X			X	X
k_{10}		X		X		X		X		X		X		X		X		X		X

4. Quantum resources for the GFSPX

To evaluate the resistance of GFSPX against Grover attacks, we quantify the quantum resource overhead, specifically gate costs and circuit depth, of its oracle. This evaluation decomposes the architecture into its fundamental operators: the S-box (S, S^{-1}), permutation layers (P, P^{-1}), round-key additions (K^i), and modular arithmetic $(\boxplus, \boxplus^{-1})$. Because the initial round differs in its key integration and swap-layer configuration, its resource cost is evaluated independently from the subsequent iterations. Aggregating these modular costs yields the cumulative quantum footprint for the full 20-round execution.

We calculate the quantum cost using the standard decomposition metric [39], assigning a unit cost to NOT and CNOT gates, and the CNOT cost of 6 to the Toffoli gate to account for its ancilla-free CNOT equivalent. Because SWAP implementation costs strictly depend on the target hardware's connectivity topology, we evaluate the overall complexity under two bounding paradigms (table 4). Excluding routing overhead, the fundamental gate sequence (2516 NOT, 11 310 CNOT, and 3112 CCNOT) yields a baseline cost of 32 498. Conversely, strictly penalizing spatial routing by decomposing the 5097 SWAP gates into three CNOTs each [40] establishes an upper bound of 47 789.

The quantum circuit depth is defined as the minimum number of sequential time steps required to execute the circuit, where gates acting on disjoint qubit sets are considered independent and may be executed simultaneously within a single time step. This metric is computed by compressing all logically independent gates into parallel layers and counting the resulting number of layers along the critical path of the circuit, which represents the longest sequential chain of dependent gate operations from input to output [18].

In our implementation, a SWAP gate is decomposed into three sequential CNOT gates acting on the same qubit pair [40]. For the Toffoli gate, a direct physical realization requires decomposition into

Table 4. Quantum resource requirements for implementation of GFSPX.

LAYER	NOT	CNOT	CCNOT	SWAP	TOTAL	COST (COST♠)	DEPTH
S	2	5	4	2	13	31 (37)	33
S^{-1}	4	2	4	3	13	30 (39)	32
P & P^{-1}	0	0	0	24	24	0 (72)	12
* K^i	4+RC	10	8	131	153+RC	62+RC (455+RC)	36
$\boxplus$ & $\boxplus^{-1}$	26	73	29	0	128	273 (273)	34
F_1 & F_1^{-1}	26	105	29	15	175	305 (350)	81
F_2	16	72	32	40	160	280 (400)	52
F_2^{-1}	32	48	32	48	160	272 (416)	54
Round Function ($i = 0$)	120	556	148	100	924	1564 (1864)	378
Round Function ($1 \leqslant i \leqslant 19$)	124+RC	566	156	231	1077+RC	1626+RC (2319+RC)	378
GFSPX	2516	11 310	3112	5097	22 035	32 498 (47 789)	7617

* This represents the computational cost of the 128-bit key schedule for all iterations except the first round. Following the key is generated, subkeys (key_0^i, key_1^i, key_2^i) are systematically sent to the relevant functional units within the round transformation.
* The gate complexity of RC is round-dependent, varying throughout the execution. The total complexity for the RC component is established as 40 NOT gates over all iterations.
♠ Values enclosed in parentheses represent the circuit analysis scenarios where SWAP gates are explicitly included. To incorporate SWAP gates into the overall cost, each gate is decomposed into three CNOT operations.

a fault-tolerant Clifford+T sequence. Each Toffoli gate is assigned a physical depth of seven and requires four ancilla qubits [18].

Applying this methodology to the fundamental primitives of GFSPX, the S-box ($\mathcal{S}$) and its inverse ($\mathcal{S}^{-1}$) achieve depths of 33 and 32, respectively, while the permutation layer (P, P^{-1}) contributes a depth of 12 due to the parallel organization of its SWAP-based implementation into six independent quartets. The round-key addition (K^i) and the modular addition layers ($\boxplus$, $\boxplus^{-1}$) require 36 and 34 depth layers, respectively. The composite F_1 and F_1^{-1} transformations yield a total depth of 81, while F_2 and F_2^{-1} establish depths of 52 and 54. Aggregating these contributions across a single round function yields an execution depth of 378, and scaling over the complete 20-round encryption process produces a global circuit depth of 7 617, as reported in table 4. To contextualize these metrics, table 5 benchmarks the GFSPX implementation against other lightweight BCs based on fundamental constraints. This comparative evaluation highlights that the GFSPX architecture achieves a high circuit depth while offering a competitive quantum cost, attaining the lowest cost-to-depth ratio (4.27) among all compared implementations.

5. Grover analysis of GFSPX

The quantum cost of a Grover key-recovery attack on GFSPX is fundamentally dictated by the architectural complexity of its quantum oracle, U_f. To evaluate the cipher's resilience under the NIST `MAXDEPTH` framework [32], the oracle is constructed using an inner parallelization strategy. Since eliminating spurious key candidates requires r = 3 plaintext–ciphertext pairs [34], the integrated oracle incorporates six distinct cipher instances (three forward (GFSPX) and three inverse (GFSPX^{-1})) to ensure that data channels are cleanly reversed via uncomputation.

Scaling the base 209-qubit footprint, the composite U_f structure demands $r \cdot 209 + 1 = 628$ qubits (See figure 9). Furthermore, integrating these parallel branches introduces a spatial routing overhead of 704 supplementary CNOT gates, partitioned into $2(r-1)k = 512$ gates to distribute the 128-bit master key and $n \cdot r = 192$ gates to evaluate the 64-bit ciphertext comparisons. The cumulative resource requirements for this integrated oracle are detailed in table 6.

Table 5. Comparison of quantum implementations with other lightweight BCs.

CIPHER [ARCH.]	QUBIT	NOT	CNOT	CCNOT	TOTAL	DEPTH(D)	COST(C)	C/D	REF
PIPO-64/128 [♣]	192	1477	2248	1248	4973	248	11 213†	45.21	[41]
DEFAULT-128/128 [♣]	256	1131	13 824	8192	23 147	644	64 107†	99.55	[41]
RECTANGLE-64/128 [♣]	192	667	6264	2400	9331	226	18 931	83.77	[42]
GIFT-64/128 [♣]	192	3261	1792	1792	6845	308	14 013	45.50	[43]
GIFT-128/128 [♣]	256	10 953	6144	6144	23 241	528	47 817	90.56	[43]
PRESENT-64/128 [♣]	192	1164	4838	2232	8234	311	17 162	55.18	[44]
PUFFIN-64/128 [♣]	192	620	3136	3584	7340	353	21 676	61.41	[44]
LiCi-64/128 [▲ + ♣]	192	379	12 900	1232	14 511	1210	20 671†	17.08	[45]
CHAM-64/128 [■]	204	2320	13 200	2320	17 840	2615	29 440†	11.26	[46]
CHAM-128/128 [■]	292	4880	28 760	4880	38 520	5307	62 920†	11.86	[46]
HIGHT-64/128 [■]	228	4496	22 614	5824	32 934	2479	62 054†	25.03	[46]
LEA-128/128 [■]	388	11 152	32 616	10 248	54 016	6505	105 256†	16.18	[46]
SIMON-64/128 [▲+■]	192	1216	7396	1408	10 020	2643	17 060†	6.45	[47]
SIMON-128/128 [▲+■]	256	4224	17 152	4352	25 728	8427	47 488†	5.64	[47]
SPECK-64/128 [■]	194	3131	10 669	3233	12 581	1863	33 228†	17.84	[48]
SPECK-128/128 [■]	256	7761	25 799	7875	41 435	4256	80 810†	18.99	[48]
GFSPX-64/128 [♣+▲+■]	209	2516	11 310	3112	22 035	7617	32 498 47 789*	4.27 6.27*	This work

♣: SPN, ▲: FEISTEL, ■: ARX

* SWAP gates are included in the cost using a 3-CNOT decomposition.

† Estimated from referenced gate counts, assigning a cost of 1 to NOT/CNOT and 6 to ancilla-free CCNOT decompositions excluded the SWAP overhead of 3-CNOT.

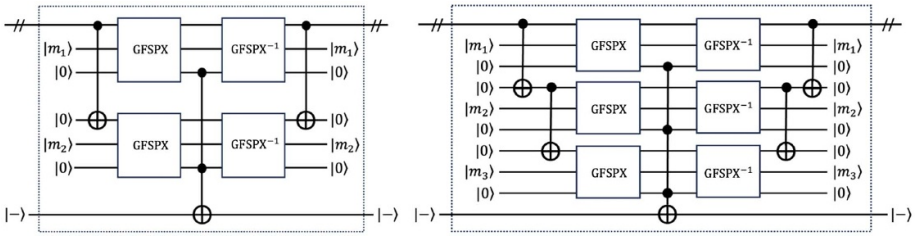

Figure 9. Quantum circuits of Grover oracles U_f for GFSPX with inner parallelization: (center) $r = 2$, and (right) $r = 3$ plaintext–ciphertext pairs.

Table 6. Main resource counts for Grover's oracle in the GFSPX.

CIPHER	r	QUBIT	NOT	CNOT	CCNOT	TOTAL	COST (COST*)	DEPTH
GFSPX	2	419	10 704	45 704	12 448	89 261	131 096 (192 308)	15 277
GFSPX	3	628	16 056	68 684	18 672	123 817	196 772 (257 984)	15 279

* SWAP gates are accounted for in the total cost via a 3-CNOT decomposition.

The ciphertext comparison triggering the phase inversion is implemented as a two-stage reversible quantum circuit. Initially, the temporary quantum register storing the candidate encryption results is evaluated against the known reference ciphertext via n parallel CNOT gates. This register reduces to the computational basis state $|0^n\rangle$ if the candidate perfectly matches the reference. Thus, evaluating r pairs simultaneously requires $n \cdot r = 192$ qubits at unit depth and zero T-gate cost. Second, a t-fold multi-controlled NOT gate is applied to the phase qubit $|-\rangle$ (figure 9). To evaluate resource efficiency within a universal elementary gate set, we decompose all multi-qubit operations into the standard fault-tolerant Clifford+T basis. To execute the phase inversion for the all-zero state, the control qubits are conjugated with $2 \cdot n \cdot r = 384$ NOT gates. Using the standard fault-tolerant decomposition [49], a t-fold MCX gate ($t \geqslant 5$) requires $32t - 84$ T-gates. For the target marking operation within the oracle ($t = 192$), this yields

6060 T-gates. Furthermore, the amplitude amplification is driven by the standard diffusion operator (D). Structurally, D consists of H and X layers bounding a single multi-controlled inversion over the 128-bit key space. Because H and X are Clifford operations, they contribute zero T-gates, localizing the entire non-Clifford cost of the diffusion operator to a 128-controlled MCX gate. Using the same decomposition, this adds 4012 T-gates. Inner-parallelization requires 2r cipher instances per iteration. Accordingly, the 20-round Toffoli gate count decomposes into $2r \cdot 18\,672 \cdot 7 \approx 7.84 \times 10^5$ T-gates. The combined overhead per Grover iteration amounts to 10,072 T-gates. This overhead contributes $\approx 1.28\%$ to the total T-gate requirement. Therefore, the total quantum cost of the Grover iteration is dominated by the architectural complexity of the cipher instances.

Since evaluating pairs $r = 3$ eliminates false positives, the search space ($N = 2^{128}$) contains exactly one valid key state ($M = 1$). The Grover operator must be applied iteratively for $j \approx \frac{\pi}{4}\sqrt{N/M} = \frac{\pi}{4}2^{64}$ iterations to amplify the target measurement probability to near-unity [50]. To determine the total attack complexity, the integrated oracle resource metrics are scaled by this iteration bound. Scaling the $r = 2$ oracle results in an estimated T-gate count of 1.09×2^{80}, a total Clifford count of 1.06×2^{81} (1.30×2^{81} including SWAP overhead), and an execution depth of 1.46×2^{77} (table 7). For $r = 3$, these estimates increase to a T-gate count of 1.64×2^{80} and a total Clifford count of 1.53×2^{81} (1.77×2^{81} including SWAP overhead), while the execution critical depth remains bound at 1.46×2^{78}.

Under the NIST post-quantum cryptography framework, the security of symmetric primitives is evaluated by comparing their quantum attack complexity to that of AES [32, 33]. This framework quantifies resilience using a total cost metric, defined as the product of the quantum circuit's total gate count and its total depth. To model realistic physical limitations of quantum hardware, NIST imposes a `MAXDEPTH` constraint that bounds the maximum allowable circuit depth over specific operational timeframes: 2^{40} models the maximum sequential operations a quantum computer could execute in approximately one year, 2^{64} corresponds to ten years, and 2^{96} models a millennium. Based on these constraints, security benchmarks are established, with Level 1 (AES-128 equivalent) requiring a total attack cost of at least 2^{170}, while Level 3 and Level 5 correspond to AES-192 (2^{233}) and AES-256 (2^{298}) equivalents, respectively (Levels 2 and 4 are reserved for hash function collision resistance, such as SHA-256 and SHA-384) [51]. The total quantum cost of the Grover attack on GFSPX is estimated at 1.55×2^{158} for $r = 2$ and 1.12×2^{159} for $r = 3$, see the last column of table 7. Because these costs strictly fall below the 2^{170} threshold within the allowable `MAXDEPTH` limits (table 8), GFSPX fails to meet NIST's Level 1 post-quantum security criteria. On the other hand, GFSPX shows a comparatively high Grover attack cost while it achieves the least quantum implementation cost, as we show it in table 5.

6. Results and discussion

The proposed quantum circuit realization of GFSPX requires 209 qubits, yielding a fundamental quantum cost of 32 498 (47 789 when accounting for SWAP decomposition) at a circuit depth of 7617. Contextualizing these results within the broader landscape of lightweight BCs (table 5) reveals several key structural insights.

GFSPX exhibits the lowest cost-to-depth ratio (4.27) in the comparison set, reflecting a high degree of parallel execution that distributes the gate operations across the temporal depth of the circuit. Such a configuration is disadvantageous under NIST `MAXDEPTH` constraints. Since the total attack cost is defined as the product of gate count and circuit depth, a higher depth directly amplifies the overall attack complexity estimate. Categorizing the evaluated ciphers identifies two distinct cost-to-depth clusters. Specifically, SPN designs (e.g. PRESENT, GIFT, RECTANGLE, PUFFIN) typically range from 45 to 100, whereas ARX constructions (e.g. SIMON, SPECK, CHAM, HIGHT, LEA) occupy the 5 to 25 range. GFSPX falls below both, indicating that its hybrid nature accumulates the depth contributions of both architectures. However, its Toffoli-to-total-gate ratio ($\approx 14\%$) closely aligns with ARX designs such as SIMON, indicating that the modular addition primitive accounts for the nonlinearity cost without introducing Toffoli overhead to the SPN branch. Furthermore, the requirement of 209 qubits is offset by a greater computational density per qubit, maintaining overall architectural efficiency.

Regarding the Grover attack analysis, the total quantum attack cost of GFSPX is estimated at 1.55×2^{158} for $r = 2$ and 1.12×2^{159} for $r = 3$ plaintext–ciphertext pairs (table 7). Comparing these values across the cipher landscape reveals the quantitative signature of the hybrid design. ARX-based ciphers such as SPECK-64/128 and SIMON-64/128 exhibit Grover costs in the range of $1.0–1.5 \times 2^{159}$, while SPN ciphers such as GIFT-64/128 and PRESENT-64/128 cluster around $1.1–1.8 \times 2^{152}$, separated by six orders of magnitude in base two. Placing within the ARX cluster despite the presence of an SPN

Table 7. Resource estimation for Grover's Oracle on GFSPX.

CIPHER [ARCH.]	r	GATE COST	#GATES	FULL DEPTH	TOTAL COST
PIPO-64/128 [♣]	2	* 45 364	$^{*}1.33 \times 2^{79}$	$^{*}1.53 \times 2^{72}$	$^{*}1.02 \times 2^{152}$
RECTANGLE-64/128 [♣]	2	* 75 980	$^{*}1.26 \times 2^{80}$	$^{*}1.40 \times 2^{72}$	$^{*}1.76 \times 2^{152}$
GIFT-64/128 [♣]	2	* 56 308	$^{*}1.87 \times 2^{79}$	$^{*}1.90 \times 2^{72}$	$^{*}1.78 \times 2^{152}$
PRESENT-64/128 [♣]	2	* 69 160	$^{*}1.15 \times 2^{80}$	$^{*}1.92 \times 2^{72}$	$^{*}1.11 \times 2^{153}$
PUFFIN-64/128 [♣]	2	* 86 960	$^{*}1.56 \times 2^{80}$	$^{*}1.09 \times 2^{73}$	$^{*}1.70 \times 2^{153}$
GIFT-128/128 [♣]	1	* 95 762	$^{*}1.59 \times 2^{80}$	$^{*}1.62 \times 2^{73}$	$^{*}1.29 \times 2^{154}$
GIFT-128/128 [♣]	2	* 191 780	$^{*}1.59 \times 2^{81}$	$^{*}1.62 \times 2^{73}$	$^{*}1.29 \times 2^{155}$
LiCi-64/128 [▲ + ♣]	2	* 83 196	$^{*}1.12 \times 2^{80}$	$^{*}1.86 \times 2^{74}$	$^{*}1.04 \times 2^{155}$
DEFAULT-128/128 [♣]	1	* 128 342	1.59×2^{81}	1.75×2^{75}	1.39×2^{157}
CHAM-64/128 [■]	2	* 117 504	1.23×2^{81}	1.00×2^{76}	1.23×2^{157}
HIGHT-64/128 [■]	2	* 247 960	1.38×2^{82}	1.90×2^{75}	1.31×2^{158}
CHAM-128/128 [■]	1	* 125 968	1.30×2^{81}	1.01×2^{77}	1.32×2^{158}
LEA-128/128 [■]	1	* 210 512	1.19×2^{82}	1.24×2^{77}	1.49×2^{159}
SIMON-64/128 [▲ + ■]	3	$*$ 34 776	1.03×2^{79}	1.03×2^{80}	1.06×2^{159}
SPECK-64/128 [■]	2	* 133 424	1.13×2^{81}	1.28×2^{78}	1.45×2^{159}
SIMON-128/128 [■]	2	* 50 688	1.03×2^{80}	1.12×2^{80}	1.15×2^{160}
SPECK-128/128 [■]	1	* 161 748	1.81×2^{81}	1.54×2^{79}	1.40×2^{161}
GFSPX (This work)	2	131 096	1.06×2^{81}	1.46×2^{77}	1.55×2^{158}
[♣+▲+■]		192 308‡	$1.30 \times 2^{81\ddagger}$		$1.91 \times 2^{158\ddagger}$
GFSPX (This work)	3	196 772	1.53×2^{81}	1.46×2^{77}	1.12×2^{159}
		257 984‡	$1.77 \times 2^{81\ddagger}$		$1.30 \times 2^{159\ddagger}$

♣: SPN, ▲: FEISTEL, ■: ARX

* Estimated using the values from table 5 for completeness.

‡ SWAP gates are accounted for in the total cost via a 3-CNOT decomposition.

Table 8. `MAXDEPTH analysis of Grover's algorithm.`

CIPHER	r	2^{40}	2^{64}	2^{96}	approximation
GFSPX-64/128	2	118	94	62	2^{158}/`MAXDEPTH`
GFSPX-64/128	3	119	95	63	2^{159}/`MAXDEPTH`

branch demonstrates that the quantum overhead introduced by the modular addition primitive governs the Grover attack complexity of this hybrid construction.

The obtained attack cost falls short of the NIST Level 1 threshold of 2^{170} by a factor of approximately 2^{11}, indicating that achieving Level 1 post-quantum security at a 128-bit key length requires structural modifications beyond incremental adjustments to the round function. Moreover, the relative overhead introduced by SWAP decomposition increases this total cost by $\approx 16\%$ (reaching 1.30×2^{159}), which is comparatively modest and attributable to the parallelized bit-level permutation in F_2. Transitioning from $r = 2$ to $r = 3$ scales the attack cost by a factor of 1.44 while reducing the expected number of false key candidates to approximately 2^{-64}, quantitatively confirming that $r = 3$ pairs are both necessary and optimal to eliminate spurious candidates for GFSPX. Finally, the combined overhead accounts for $\approx 1.28\%$ of the total T-gate cost ($10072/7.84 \times 10^5$), indicating that the cipher instances strictly dominate the attack complexity with minimal margin for external optimization.

7. Conclusion

This study presents a qubit-optimized quantum circuit realization and a comprehensive Grover cryptanalysis of GFSPX. By mapping its hybrid 4-branch generalized Feistel architecture, integrating both ARX and SPN components into a reversible quantum framework, we systematically evaluated the cipher's post-quantum resilience under NIST security criteria.

Our analysis demonstrates that while GFSPX achieves a highly efficient spatial implementation with a competitive cost-to-depth profile, it does not satisfy the NIST's post-quantum security requirements for a 128-bit key. Crucially, however, the cipher exhibits a higher Grover attack resistance than pure SPN equivalents, a property directly attributable to the quantum overhead inherent in its modular addition primitive. This highlights a fundamental trade-off in hybrid designs as the architectural choices that drive classical resource efficiency can simultaneously amplify post-quantum security margins.

These findings suggest two main research directions. First, the framework can be extended to evaluate GFSPX variants with expanded key lengths (e.g. 192 or 256 bits) or intensified ARX layers. Second, further circuit optimization remains possible by refining the reversible modular-addition primitive through depth-reduced or T-count-minimized adder variants. The methodology developed in this study provides a rigorous framework for the quantum-aware evaluation of other hybrid ARX-SPN designs in resource-constrained environments.

Data availability statement

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

Author contributions

Ibrahim Ulgen 0000-0001-7656-074X
Conceptualization (equal), Data curation (equal), Formal analysis (equal), Investigation (equal), Methodology (equal), Resources (equal), Supervision (equal), Validation (equal), Visualization (equal), Writing – original draft (equal), Writing – review & editing (equal)

Hasan Ozgur Cildiroglu 0000-0003-0246-1102
Conceptualization (equal), Data curation (equal), Formal analysis (equal), Investigation (equal), Methodology (equal), Resources (equal), Supervision (equal), Validation (equal), Visualization (equal), Writing – original draft (equal), Writing – review & editing (equal)

Oğuz Yayla 0000-0001-8945-2780
Investigation (supporting), Visualization (supporting), Writing – original draft (equal), Writing – review & editing (equal)

References

[1] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
[2] Google Quantum AI and Collaborators 2025 Observation of constructive interference at the edge of quantum ergodicity *Nature* **646** 825–30
[3] Gidney C 2025 How to factor 2048 bit RSA integers with less than a million noisy qubits (arXiv:2505.15917v1)
[4] Cain M *et al* 2026 Shor's algorithm is possible with as few as 10,000 reconfigurable atomic qubits (arXiv:2603.28627)
[5] Babbush R *et al* 2026 Securing elliptic curve cryptocurrencies against quantum vulnerabilities: resource estimates and mitigations (arXiv:2603.28846)
[6] Gu A, Ataides J P B, Lukin M D and Yelin S F 2026 Scalable neural decoders for practical fault-tolerant quantum computation (arXiv:2604.08358)
[7] Webster P *et al* 2026 The pinnacle architecture: reducing the cost of breaking RSA-2048 to 100 000 physical qubits using quantum LDPC codes (arXiv:2602.11457)
[8] Tripier F *et al* 2026 Fault-tolerant quantum computing with trapped Ions: the walking cat architecture (arXiv:2604.19481)
[9] Shor P W 1997 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM J. Comput.* **26** 1484–509
[10] Rivest R L, Shamir A and Adleman L 1978 A method for obtaining digital signatures and public-key cryptosystems *Commun. ACM* **21** 120–6
[11] Diffie W and Hellman M E 1976 New directions in cryptography *IEEE Trans. Inf. Theory.* **22** 644–54
[12] Miller V S 1985 Use of elliptic curves in cryptography *Conf. Theory and Application of Cryptographic Techniques* pp 417–26
[13] Koblitz N 1987 Elliptic curve cryptosystems *Math. Comput.* **48** 203–9
[14] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. 28th Annual ACM Symp. on Theory of Computing (Stoc)* (ACM)
[15] Yamamura A and Ishizuka H 2000 Quantum cryptanalysis of block ciphers (Algebraic systems, formal languages and computations) *RIMS Kokyuroku* **1166** 235–43
[16] Daemen J and Rijmen V 2001 Specification for the advanced encryption standard (AES) *FIPS* **197** 1–51
[17] 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 *Sci. China Phys. Mech. Astron.* **65** 290–311
[18] Selinger P 2013 Quantum circuits of T-depth one *Phys. Rev. A* **87** 042302
[19] Shannon C E 1949 Communication theory of secrecy systems *Bell Syst. Tech. J.* **28** 656–715
[20] Feistel H 1973 Cryptography and Computer Privacy *Sci. Am.* **228** 15–23
[21] FIPS 1999 Data encryption standard *NBS* **46-3** (availible at: https://csrc.nist.gov/files/pubs/fips/46-3/final/docs/fips46-3.pdf)
[22] Aboushosha B *et al* 2020 SLIM: a lightweight block cipher for internet of health things *IEEE Access* **8** 203747–57
[23] Khan S, Ferreira Lopes martins P A, Sousa B and Pereira V 2025 A comprehensive review on lightweight cryptographic mechanisms for industrial internet of things systems *ACM Comput. Surv.* **58** 1–37

[24] Seo H, Jeong I, Lee J and Kim W-H 2018 Compact implementations of ARX-based block ciphers on IoT processors *ACM Trans. Embed. Comput. Syst. (TECS)* **17** 1–16
[25] Cildiroglu H O and Yayla O 2025 Quantum implementation of SLIM and its Grover cryptanalysis *Phys. Scr.* **100** 105023
[26] Cildiroglu H O, Basmaci H and Yayla O 2026 Quantum Grover attack on MIBS *Open Phys.* **24**
[27] Huang M and Wang L 2020 Automatic search for the linear (hull) characteristics of ARX ciphers: applied to SPECK, SPARX, Chaskey and CHAM-64 *Secur. Commun. Netw.* **2020** 4898612
[28] Qin H and Wu B 2022 Towards non-independence of modular additions in searching differential trails of ARX ciphers: new automatic methods with application to SPECK and Chaskey (arXiv:2203.09741)
[29] Zhang X, Shao C, Li T, Yuan Y and Wang C 2024 GFSPX: an efficient lightweight block cipher for resource-constrained IoT nodes *J. Supercomput.* **80** 25256–82
[30] 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 Int. Workshop, Vienna, Austria, September 10-13, 2007. Proc. 9* pp 450–66
[31] Dobraunig C, Eichlseder M, Mendel F and Schläffer M 2021 Ascon v1. 2: lightweight authenticated encryption and hashing *J. Cryptol.* **34** 33
[32] NIST 2016 *Project for Post-Quantum Cryptography Standardization. Call for Proposals: Post-Quantum Cryptography (Nist IR 8105)* (U.S. Department of Commerce)
[33] Barker E and Barker W 2020 Sp 800-57 part 1 rev. 5 recommendation for key management: part 1 – general *Technical Report* (National Institute of Standards and Technology)
[34] Langenberg B, Pham H and Steinwandt R 2020 Reducing the cost of implementing the advanced encryption standard as a quantum circuit *IEEE Trans. Quantum Eng.* **1** 1–12
[35] Cuccaro S A, Draper T G, Kutin S and Moulton D P 2004 A new quantum ripple-carry addition circuit (available at: https://api.semanticscholar.org/CorpusID:118445805)
[36] Dasu V A, Baksi A, Sarkar S and Chattopadhyay A 2019 LIGHTER-R: optimized reversible circuit implementation for Sboxes *2019 32nd IEEE Int. System-on-Chip Conf. (SOCC)* pp 260–5
[37] Chun M, Baksi A and Chattopadhyay A 2023 DORCIS: depth optimized quantum implementation of substitution boxes *Cryptol. Eprint Arch.* 2023/286 (available at: https://eprint.iacr.org/2023/286)
[38] Tilki O, Cildiroglu H O, Yayla O and Saran A N 2025 Quantum implementation of S-boxes: a memory optimized approach *Proc. 18th Int. Conf. on Information Security and Cryptography (ISC2025)* (IEEE Access)
[39] Shende V and Markov I 2009 On the CNOT-cost of TOFFOLI gates *Quantum Inf. Comput.* **9** 461–86
[40] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* 10th Anniversary edn (Cambridge University Press)
[41] Jang K *et al* 2025 Quantum implementation and analysis of default *Cryptogr. Commun.* **17** 359–75
[42] Saravanan P, Jenitha J, Aasish S and Sanjana S 2021 Quantum circuit design of RECTANGLE lightweight cipher *2021 25th Int. Symp. on VLSI Design and Test (VDAT)* pp 1–4
[43] Jang K, Kim H, Eum S and Seo H 2020 Grover on GIFT Cryptology ePrint Archive
[44] 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
[45] 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
[46] Jing X *et al* 2022 Parallel quantum addition for Korean block ciphers *Quantum Inf. Process.* **21** 373
[47] Anand R, Maitra A and Mukhopadhyay S 2020 Grover on SIMON *Quantum Inf. Process.* **19** 1–17
[48] Jang K, Baksi A, Kim H and Seo H 2022 Improved quantum analysis of SPECK and LowMC *Progress in Cryptology – INDOCRYPT 2022* pp 517–40
[49] Nathan W and Roetteler M 2016 Quantum arithmetic and numerical analysis using repeat-until-success circuits *Quantum Inf. Comput.* **16** 134–78
[50] Jaques S, Naehrig M, Roetteler M and Virdia F 2020 Implementing Grover oracles for quantum key search on AES and LowMC *Adv. Cryptol.* **12106** 280–310
[51] Grassl M, Langenberg B, Roetteler M and Steinwandt R 2016 Grover's algorithm to AES: quantum resource estimates *PQCRYPTO* **9606** 29–43