New Results in Quantum Analysis of LED: Featuring One and Two Oracle Attacks

New Results with Optimizations and Consideration for Modes

Siyi Wang[1], Kyungbae Jang[2], Anubhab Baksi[1], Sumanta Chakraborty[3], Bryan Lee[1], Anupam Chattopadhyay[1] and Hwajeong Seo[2]

[1] Nanyang Technological University, Singapore
[2] Hansung University, Seoul, Republic of Korea
[3] Techno International New Town, Kolkata, India

Abstract. Quantum computing has attracted substantial attention from researchers across various fields. In case of the symmetric key cryptography, the main problem is posed by the application of Grover's search. In this work, we focus on quantum analysis of the lightweight block cipher LED.
This paper proposes an optimized quantum circuit for LED, minimizing the required number of qubits, quantum gates, and circuit depth. Furthermore, we conduct Grover's attack and Search with Two Oracles (STO) attack on the proposed LED cipher, estimating the quantum resources required for the corresponding attack oracles. The STO attack outperforms the usual Grover's search when the state size is less than the key size. Beyond analyzing the cipher itself (i.e., the ECB mode), this work also evaluates the effectiveness of quantum attacks on LED across different modes of operation.

Keywords: Grover's Search · Search with Two Oracles · LED Block Cipher · Modes of Operation · Quantum Circuits · Quantum Cryptography

1 Introduction

The Internet of Things (IoT) is revolutionizing the way devices communicate and operate through interconnected devices, resulting in a large number of applications across a variety of domains [CXL+14, LL15]. Nonetheless, IoT security remains a significant concern due to the large number of devices and their limited resources. Lightweight cryptography aims to address these limitations, ensuring secure communication while maintaining efficiency in terms of power, speed, and hardware resources. However, with the development of quantum computing, quantum attacks such as Grover's algorithm pose substantial risks to encryption systems, thereby significantly impacting the security of lightweight cryptography in the quantum computing era.

Therefore, it is crucial to explore and analyze lightweight ciphers in the context of quantum attacks. Several relevant studies have already been conducted [JSK+21, JCKS20], focusing on lightweight block ciphers such as PRESENT, GIFT, and SPECK. Moreover, we have seen recent improvements on the attack on AES, namely [LPZW23, SF24, JBS+22].

Among various lightweight block ciphers, the LED cipher stands out due to its compact hardware design and minimal silicon footprint. Given these properties, this paper focuses

E-mail: siyi002@e.ntu.edu.sg (Siyi Wang), starj1234@hansung.ac.kr (Kyungbae Jang), anubhab.baksi@ntu.edu.sg (Anubhab Baksi), csum1009@gmail.com (Sumanta Chakraborty), blee061@e.ntu.edu.sg (Bryan Lee), anupam@ntu.edu.sg (Anupam Chattopadhyay), hwajeong@hansung.ac.kr (Hwajeong Seo)

Date of this document: 2024-12-12.

on the quantum implementation and quantum attack of the LED cipher. Specifically, we propose an efficient quantum circuit for the LED cipher, minimizing the required qubits, quantum gates, and circuit depth. Furthermore, we perform Grover's attack and Search with Two Oracles (STO) attack on the proposed quantum LED cipher and estimate the quantum resources required for these attack oracles. This work also addresses several issues found in the quantum LED cipher proposed by Song et al. [SJS+23]. By comparing it with the corrected version of their implementation, it is evident that our design provides a more comprehensive and efficient solution. Additionally, this paper also addresses modes of operation in the context of quantum attack. In classical cryptography, modes of operation such as ECB, CBC, OFB, and CFB have been widely discussed and standardized since their introduction in FIPS 81 back in 1981. However, in quantum cryptography, while quantum attacks on block ciphers have been extensively explored, modes of operation have not been considered thus far. To fill this critical gap, this work proposes frameworks for ECB, CBC, and CFB modes based on the quantum LED cipher. Furthermore, we provide a thorough analysis of the quantum attack costs associated with these different modes of operation.

Contributions & Novelty

In brief, our contributions are detailed as follows[1]:

- **Efficient Quantum Circuit for LED Block Cipher and Resource Estimates.** We present a novel approach to implementing the LED block cipher as a quantum circuit. The proposed approach optimizes the quantum circuit of the LED cipher by exploring efficient subcircuit structures, thereby enhancing the overall efficiency.

  We evaluate the performance of our design using the ProjectQ framework [SHT16]. Our analysis includes a detailed comparison of the required quantum resources, such as qubits and circuit depth, with respect to other block cipher implementations in the literature. Compared to previous works, our implementation achieves a more efficient quantum circuit for the LED cipher.

- **Quantum Cost Estimation of One Oracle (Grover's) and Two Oracles (STO) Attack.** In this work, Grover's attack (that employs one quantum oracle) complexity is reported for LED. Moreover, as referenced in [KLL15, DP20], we also investigate the requirement and the impact of the STO attack, which is an improvement over Grover's attack when the state size is less than the key size for a block cipher (and more complex as it employs two quantum oracles). For both types of attacks, we estimate the quantum resource requirements and provide a comprehensive evaluation of their efficiency.

- **Modes of Operation for Block Ciphers in Quantum.** This work explores the modes of operation in quantum attacks, proposing frameworks for ECB, CBC and CFB modes based on the proposed quantum LED cipher. Besides, it also provides a thorough analysis of Grover's attack's effectiveness across these modes.

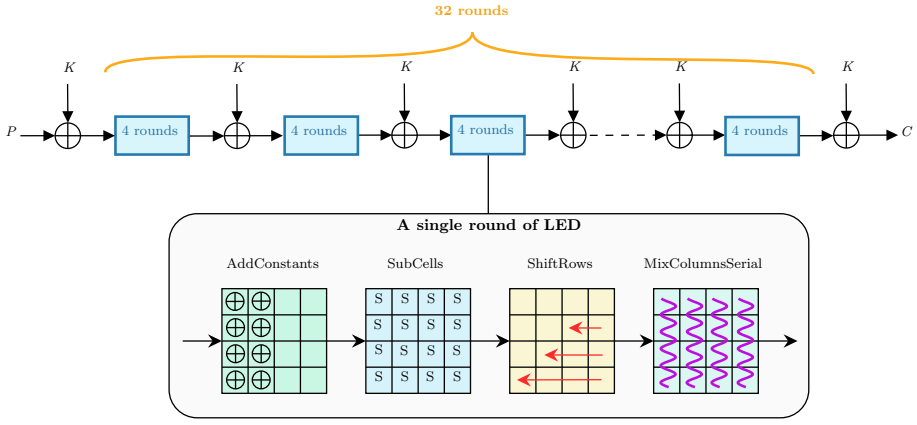
2 Background

2.1 LED Block Cipher

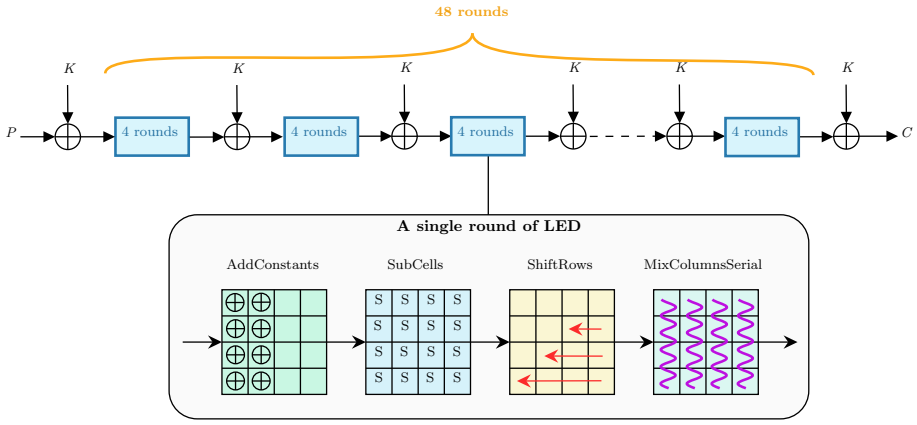
In 2011, Guo et al. [GPPR11] introduced the LED block cipher, well-suited for efficient encryption/decryption in resource-constrained environments. LED operates on 64-bit blocks and supports key sizes of 64 and 128 bits. The cipher employs 32 rounds for the

[1]The source-codes will be provided with a future version of this paper.

64-bit key version and 48 rounds for the 128-bit key version. Specifically, before the first round, the LED cipher performs an initial AddRoundKey operation. Subsequently, the AddRoundKey operation is executed every four rounds. Each round consists of four steps performed sequentially: AddConstants, SubCells, ShiftRows and MixColumns. Besides, it is necessary to highlight that performing one MixColumnsSerial requires executing four MixColumn operations. An overview of the LED encryption process is shown in Figure 1, with a detailed discussion of each component as follows.



(a) LED-64.



(b) LED-128.

Figure 1: Schematics for LED block cipher.

- **KeySchedule** and **AddRoundKey.** LED employs a straightforward key schedule. For LED-64, the same 64-bit user key K is used directly in each round. While for LED-128, the 128-bit user key is divided into two subkeys ($K = K0||K1$), where the key in each round is alternately set to equal the left part $K0$ and the right part $K1$ of K. Each 64-bit round key is Exclusive-OR-ed with 64-bit state.

- **AddConstants.** The round constants are detailed in Table 1, which presents the constants $(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$ encoded as byte values for each round. Paticularly, rc_0 represents the least significant bit.

- **SubCells.** The LED cipher reuses the S-box ($C56B90AD3EF84712$) from the PRESENT block cipher [BKL+07].

Table 1: Round constants used in LED.

Rounds	Constants
$1-24$	$01, 03, 07, 0F, 1F, 3E, 3D, 3B, 37, 2F, 1E, 3C, 39, 33, 27, 0E, 1D, 3A, 35, 2B, 16, 2C, 18, 30$
$25-48$	$21, 02, 05, 0B, 17, 2E, 1C, 38, 31, 23, 06, 0D, 1B, 36, 2D, 1A, 34, 29, 12, 24, 08, 11, 22, 04$

- **ShiftRows.** This operation involves cyclically left shifting the bytes in $i-th$ row of the state array by i cell positions. Specifically, the $0th$ row remains unchanged, while the $1st$ row is shifted left by 4 bits, the $2nd$ row is shifted left by 8 bits, and the $3rd$ row is shifted left by 12 bits.

- **MixColumns.** This operation processes each column of the state array as a column vector, which is then replaced by a new vector obtained by post-multiplying it by the matrix M (the MixColumns matrix). This matrix is actually obtained by another matrix, A (the MixColumnsSerial matrix), by raising it to the fourth power. Both are defined over $\mathrm{GF}(2^4)/x^4+x+1$ and are given by:

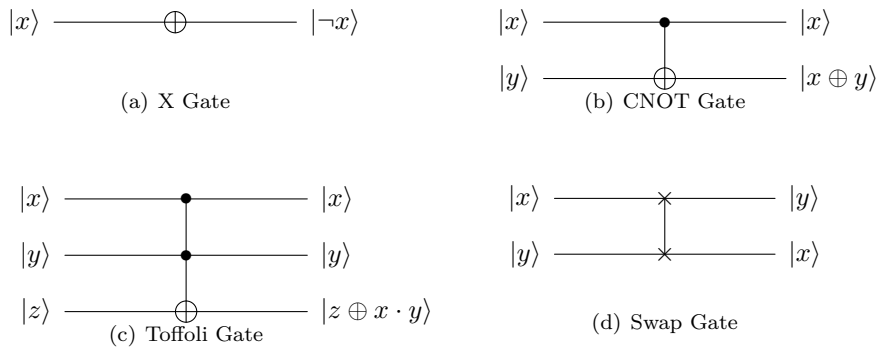
$$M = \begin{bmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{bmatrix}$$

  Note that $M = A^4$ and is MDS[2]. The authors recommended to implement M by successively applying the circuit of A four times for low resource utilization.

In brief, the prudent design of LED strikes a balance between security and resource efficiency, making this cipher particularly well-suited for IoT devices and other applications with strict resource constraints.

2.2 Quantum Gates

Shown in Figure 2, the 4 quantum gates play crucial roles in the implementation. That said, we may like to decompose the AND operations, see [CBC23, Section II] or [BJ24] for more details.



(a) X Gate (b) CNOT Gate (c) Toffoli Gate (d) Swap Gate

Figure 2: Common quantum gates.

- **X Gate.** This quantum gate performs a bit-flip operation, inverting the state of a qubit from $|x\rangle$ to $|\neg x\rangle$ and vice versa.

[2]Also, one may note that $A = M^{64}$.

- **CNOT Gate.** This gate is a two-qubit gate that flips the state of the target qubit if the control qubit is in the state $|1\rangle$.

- **Toffoli Gate.** This operation, also known as the controlled-controlled-NOT (CC-NOT) gate, flips the state of the target qubit only when both control qubits are in the state $|1\rangle$. It is noteworthy that this quantum gate can be decomposed into basic quantum gates such as X, CNOT, H, and T gates [AMM+13] in various ways. Consequently, the implementation cost depends on the specific decomposition method.

- **SWAP Gate.** This is a two-qubit gate that exchanges the states of two qubits. As illustrated in Figure 2(d), the operation SWAP(x, y) results in (y, x).

2.3 Grover's Attack

In quantum cryptography, Grover's attack can be utilized to perform a quantum key search, significantly reducing the time complexity of finding the correct encryption key.

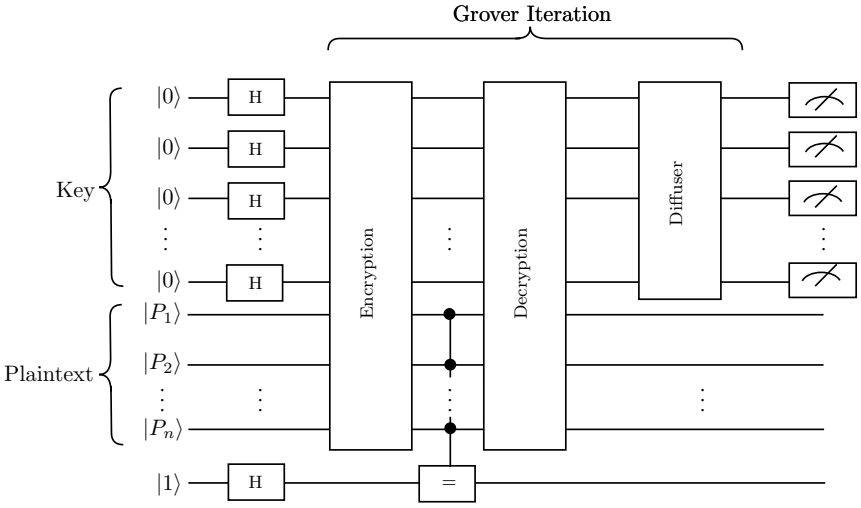


Figure 3: Quantum oracles for Grover's algorithm.

Before delving into Grover's attack, it is essential to first understand Grover's algorithm. This famous quantum algorithm provides a quadratic speedup over classical algorithms for unstructured search problem. Specifically, Grover's algorithm can locate the solution to a problem within an unsorted database of N items using approximately $O(\sqrt{N})$ queries, compared to the $O(N)$ queries required by classical brute-force method.

Grover's attack is based on the principles of Grover's algorithm. Figure 3 illustrates the quantum oracles used in this attack, demonstrating how it directly borrows and adapts the structure of Grover's algorithm to achieve its key search objective. Specifically, this attack begins by applying Hadamard gates to the key qubits, which creates superposition states. Next, the specific encryption quantum circuit uses these states to encrypt the plaintext. If the generated ciphertext matches the known ciphertext, this oracle then inverts the sign of the corresponding key state. Following this, the diffusion operator is applied to amplify the amplitude of the potential solution key. The combination of the oracle and diffusion operator is repeated multiple times to further enhance the amplitude of the correct key. Finally, measuring the key qubits reveals the most probable solution.

2.4 Improvement over Grover: Search with Two Oracles

In this subsection, we introduce the Search with Two Oracles (STO) attack as an enhancement of Grover's algorithm.

Initially proposed by Kimmel et al. [KLL15], the STO attack is able to reduce the overall cost of quantum search procedures compared to directly applying the Grover's algorithm. This attack utilizes two quantum oracles: the first oracle, O_γ, is relatively inexpensive and marks both the M target items and a number of false positives; the second, O_χ, is more expensive but accurately identifies the M target items. Particularly, O_χ is identical to the one used in Grover's algorithm. For STO attack, the number of queries required remains $O(\sqrt{2^k/M})$ as that for the Grover's attack, where k denotes the key size. In brief, this attack enhances the overall efficiency by balancing the costs of O_γ and O_χ with the number of false positives. By carefully designing these oracles and managing their associated costs, the search process can be significantly optimized.

The STO attack we implement in this paper follows a similar methodology to that described in Reference [DP20]. In particular, our construction of the oracle O_χ employs a serial-oracle design pattern, as suggested in the same reference.

In subsequent content, we provide a thorough analysis of the quantum resources required for the proposed STO attack on LED cipher.

3 Efficient Quantum Implementation of LED Cipher

In this section, we present an efficient quantum implementation of the LED cipher, focusing on minimizing the overall cost. This cipher consists of several submodules, which are arranged into multiple rounds of encryption. Accordingly, we will detail the efficient implementations for the five key submodules of quantum LED cipher in the subsequent subsections.

3.1 AddRoundKey

The quantum AddRoundKey operation is implemented using only CNOT gates, resulting in a circuit with a depth of one. This straightforward operation is simpler than other components like the S-box and MixColumn, as it follows a generic implementation method. Specifically, in the LED cipher, a 64-qubit round key is XORed with the intermediate state using 64 CNOT gates, as illustrated in ProjectQ Code 3.1.

Code 3.1: AddRoundKey of quantum LED implementation

```
1     def KeyAddition(eng, state, key):
2         for i in range(64):
3             CNOT | (key[i], state[i])
```

3.2 AddConstants

Similar to the AddRoundKey operation, the quantum AddConstants operation is also based on a generic method. Since the round constant in the LED cipher is a fixed value, as detailed in Table 1, this quantum implementation only involves applying X gates to the state qubits where the corresponding bits in the round constant are set to 1. The specific ProjectQ implementation details are provided in Code 3.2.

Code 3.2: AddConstants of quantum LED implementation.

```
def Round_constant_XOR(eng, state, round_constant):
    for i in range(64):
        if (round_constant >> i & 1):
            X | state[i]
```

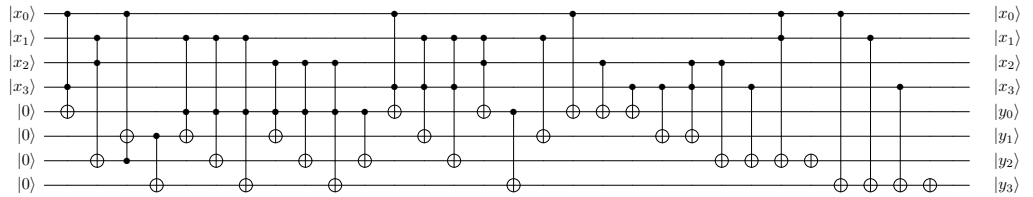
3.3 SubCells

The SubCells operation applies a non-linear substitution using Sbox. As mentioned in Section 2.1, the LED cipher reuses the same Sbox as PRESENT [BKL+07]. The coordinate functions of this Sbox are:

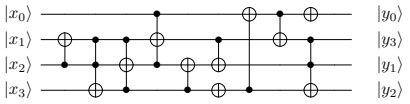
$$
\begin{aligned}
y_0 &= x_0 \oplus x_1x_2 \oplus x_2 \oplus x_3 \\
y_1 &= x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \oplus x_1x_3 \oplus x_1 \oplus x_2x_3 \oplus x_3 \\
y_2 &= x_0x_1x_3 \oplus x_0x_1 \oplus x_0x_2x_3 \oplus x_0x_3 \oplus x_1x_3 \oplus x_2 \oplus x_3 \oplus 1 \\
y_3 &= x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \oplus x_0 \oplus x_1x_2 \oplus x_1 \oplus x_3 \oplus 1
\end{aligned}
$$

For the quantum implementation, the design of an efficient quantum Sbox is crucial, as it directly impacts the performance of this operation. Given the inherent complexity and the numerous possible approaches for implementing a quantum LED Sbox, we have explored a range of strategies to identify the most efficient design. These strategies are outlined in detail below.

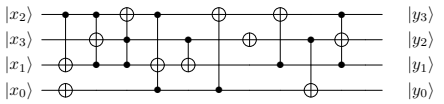
- **Naïve.** This approach directly utilizes the Sbox coordinate functions to design the quantum circuit. While straightforward, this method does not optimize for quantum resources, resulting in higher quantum costs.
- **LIGHTER-R.** LIGHTER-R [DBSC19] is a tool designed for implementing S-boxes using reversible logic libraries, specifically targeted for 4-bit S-boxes. Using this tool, we generated a quantum implementation of the LED Sbox, which is illustrated in Figure 4(b).
- **DORCIS.** DORCIS [CBC23] is a tool that optimizes quantum implementations for arbitrary 3- and 4-bit S-boxes, extending the capabilities of the LIGHTER-R. Unlike LIGHTER-R, which only handles 4-bit S-boxes and operates at a top level using Toffoli gates, DORCIS incorporates quantum decomposition with Clifford and T gates, optimizing both quantum depth and T-depth. In this paper, we employed DORCIS to generate a quantum implementation of the LED S-box, as shown in Figure 4(c).
- **Sbox α.** In the paper [CHM11], Courtois et al. optimized various small digital circuits, including LED S-box. It is shown in Figure 4(d).
- **Sbox β.** In 2024, Feng et al. [FWZ+24] proposed optimized implementations of lightweight cryptographic S-boxes using SAT solvers, which significantly enhance the overall cryptographic performance. Here, we implement their optimized LED S-box into quantum circuit, as shown in Figure 4(e).
- **Sbox γ.** In the paper by Cai et al. [CGL22], a quantum S-box for the LED cipher is proposed that utilizes 5 qubits (i.e., 1 ancilla/garbage qubit). This design is illustrated in Figure 4(f).



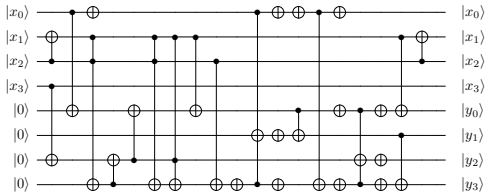
(a) Naïve.



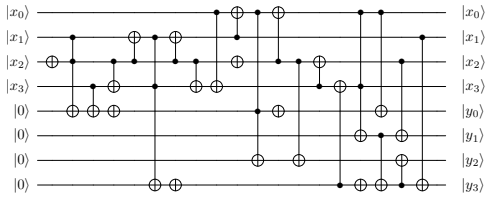
(b) LIGHTER-R [DBSC19].



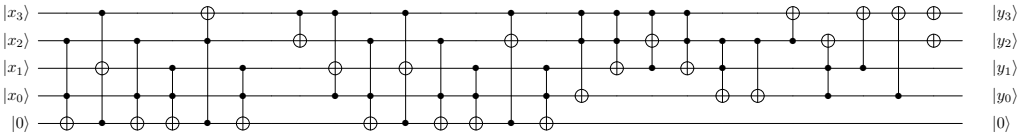
(c) DORCIS [CBC23].



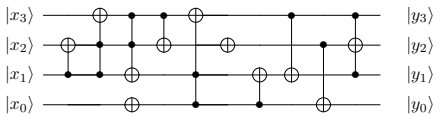
(d) Sbox α [CHM11].



(e) Sbox β [FWZ+24].



(f) Sbox γ [CGL22].



(g) Sbox δ [CLF+24]

Figure 4: Quantum implementations of LED Sbox.

- **Sbox δ.** In 2024, Chen et al. [CLF+24] proposed a SAT-based model that optimizes quantum circuits incorporating three metrics. Based on this model, we designed a more compact quantum circuit for LED Sbox, as illustrated in Figure 4(g).

In Table 2, the quantum costs associated with each approach are presented. Here "1qClifford" refers to the Clifford gate operating on a single qubit, such as the Hadamard, X or S gates. It can be seen that both the Sbox γ by Chen et al. and the DORCIS-generated Sbox exhibit the highest efficiency, with a circuit depth of 8 and a Toffoli depth of 4. Since the two designs are very similar, either one could be chosen. In this paper, we decided to use the DORCIS-generated implementation for our quantum S-box.

3.4 ShiftRows

The ShiftRows operation, which performs circular shifts, can be implemented using quantum swap gates, also known as "physical swap" approach. Alternatively, this operation can be achieved using the "logical swap" approach, which involves rearranging the indices of the

Table 2: Quantum resource requirement for LED SBox.

Method	#1qCliff	#CNOT	#Toffoli	#T	#Qubits	Circuit depth	Toffoli depth
Naïve (Figure 4(a))	2	13	15	105	8	27	14
LIGHTER-R [DBSC19] (Figure 4(b))	2	5	4	28	4	9	4
DORCIS [CBC23] (Figure 4(c))⋆	2	5	4	28	4	8	4
Sbox α [CHM11] (Figure 4(d))	12	12	5	35	8	16	5
Sbox β [FWZ+24] (Figure 4(e))	6	16	4	28	8	19	4
Sbox γ [CGL22] (Figure 4(f))	7	5	19	133	5	24	19
Sbox δ [CLF+24] (Figure 4(g))	2	5	4	28	4	8	4

⋆: Used in this work

qubits logically without employing additional quantum swap gates. As one can see, the second approach is more resource-efficient, as it avoids the requirement for extra quantum resources. Therefore, in this work, we adopt the "logical swap" approach. Details of the implementation are provided in Code 3.3.

Code 3.3: ShiftRows of quantum LED implementation.

```
def ShiftRows(eng, state):
    new_state = []
    # 1st row--left rotate 4 places
    new_state[0:4] = state[4:8]
    new_state[4:8] = state[8:12]
    new_state[8:12] = state[12:16]
    new_state[12:16] = state[0:4]
    # 2nd row--left rotate 8 places
    new_state[16:20] = state[24:28]
    new_state[20:24] = state[28:32]
    new_state[24:28] = state[16:20]
    new_state[28:32] = state[20:24]
    # 3rd row--left rotate 12 places
    new_state[32:36] = state[44:48]
    new_state[36:40] = state[32:36]
    new_state[40:44] = state[36:40]
    new_state[44:48] = state[40:44]
    # 0th row--left rotate 0 place
    new_state[48:64] = state[48:64]
    return new_state
```

3.5 MixColumn

Based on the specification of LED (see Section 2.1), the MixColumnsSerial matrix (A) and the MixColumns (M) matrix can be given in binary as follows:

$$M = \begin{bmatrix}
0&0&1&0&1&0&0&0&0&1&0&0&0&1&0&0\\
1&0&0&1&0&1&0&0&0&0&1&0&0&0&1&0\\
1&1&0&0&0&0&1&0&1&0&0&1&1&0&0&1\\
0&1&0&0&0&0&0&1&1&0&0&0&1&0&0&0\\
1&0&0&1&0&1&1&0&1&0&1&0&0&1&1&0\\
1&1&0&0&1&0&1&1&1&1&0&1&1&0&1&1\\
0&1&1&0&0&1&0&1&1&1&1&0&0&1&0&1\\
0&0&1&0&1&1&0&0&0&1&0&1&1&1&0&0\\
0&1&0&1&1&1&1&1&1&1&1&0&1&0&0&0&1\\
1&0&1&0&0&1&1&1&1&1&1&1&0&1&0&0&0\\
1&1&0&1&0&0&1&1&1&1&1&1&1&0&1&0&0\\
1&0&1&1&1&1&1&0&1&0&1&0&0&0&1&1\\
0&1&0&0&0&1&0&0&0&1&1&1&0&1&0&1\\
0&0&1&0&0&0&1&0&0&0&1&1&1&0&1&0\\
1&0&0&1&1&0&0&1&0&0&0&1&1&1&0&1\\
1&0&0&0&1&0&0&0&1&1&1&1&1&0&1&1
\end{bmatrix}$$

$$A = \begin{bmatrix}
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&1&0&1&0&0&0&0&1&0&0&0&1&0&0\\
1&0&0&1&0&1&0&0&0&0&1&0&0&0&1&0\\
1&1&0&0&0&0&1&0&1&0&0&1&1&0&0&1\\
0&1&0&0&0&0&0&1&1&0&0&0&1&0&0&0
\end{bmatrix}$$

Here, we explore several methods to determine the most efficient design, which are summarized as follows:

- **Naïve (Out-of-place).** The naïve approach involves using ancilla qubits to compute the new state of a qubit while preserving its previous state in a backup qubit, which is

essential for the calculation of new state of two other qubits. Following this approach, Song et al. [SJS+23] proposed the naïve implementation of the MixColumn operation in the quantum LED cipher. As shown in Table 3, this implementation requires 32 qubits and 108 CNOT gates, with a quantum depth of 10.

- **PLU Factorization.** PLU factorization decomposes a given binary matrix into a permutation matrix, a lower triangular matrix, and an upper triangular matrix. This factorization can also be applied to achieve an efficient in-place quantum implementation for MixColumn operation. By following the method detailed in [vH19], we use Sage[3] to obtain an in-place quantum implementation through PLU factorization.
- **Gauss-Jordan Elimination.** Gauss-Jordan elimination is used to factorize any binary matrix through elementary operations, which correspond to CNOT and SWAP gates in quantum circuits. By applying this method, we achieve an in-place quantum implementation that utilizes 16 qubits with 103 CNOT gates and 8 SWAP gates, with a quantum depth of 52.
- **XZLBZ.** The XZLBZ algorithm [XZL+20] offered an innovative approach for the in-place quantum implementation of binary matrices. At that time, it was the first tool to efficiently implement a given linear layer. The most notable result from this paper was to find an in-place implementation of the AES MixColumn with 92 CNOT gates and 30 quantum depth (although they did not optimize for quantum depth). Note that a revision of this algorithm was reported in [BCC+24].
- **YWSZZ.** This is a very recent paper [YWS+24] where the authors managed to find an in-place implementation of the AES MixColumn matrix with 91 CNOT gates and (13 classical depth, 35 quantum depth). This is where the record stands till date, to the best of our knowledge. In our case though, it turns out that XZLBZ (44) outperforms YWSZZ (47) in terms of CNOT gates.

Table 3: Quantum resource requirement for LED MixColumns.

Method	#CNOT	#SWAP	Circuit depth
Out-of-place (32 qubits)			
Naïve (used in [SJS+23])	108	0	10
In-place (16 qubits)			
PLU	103	8	62
Gauss-Jordan	103	8	52
XZLBZ [XZL+20]	45	16	19
XZLBZ [XZL+20]*	44	16	16
Modified XZLBZ [BCC+24]	50	14	17
Modified XZLBZ [BCC+24]	46	12	20
YWSZZ [YWS+24]	47	16	10

*: Used in this work

As shown in Table 3, we summarize the quantum costs for the various approaches discussed before. One may note that, the XZLBZ implementation demonstrates the highest efficiency, achieving 44 CNOT gates with 16 qubits and a circuit depth of 16. Hence, we chose this implementation as our quantum MixColumn.

The matrix A requires 14 CNOT gates in the optimal in-place implementation. This was found using the MILP tool of [BKD21]. Thus, it would require $4 \times 14 = 52$ CNOT gates to implement M by using this matrix, and hence were not considered here. Using

[3] https://doc.sagemath.org/html/en/reference/matrices/sage/matrix/matrix2.html

the same tool, we also found that the minimum number of CNOT gates that are required to implement M in-place is at least 15[4].

3.6 Architecture

Using the presented modules AddConstants, SubCells, ShiftRows, and MixColumns, we construct the architecture for the LED quantum circuit. Even at the architectural level, the in-place design is implemented, where the ciphertext is computed directly on the input qubits (i.e., plaintext) without allocating additional qubits.

Figure 5 shows the in-place architecture of the LED quantum circuit (here AC, SC, SR, and MC represent AddConstants, SubCells, ShiftRows, and MixColumns, respectively). The only differences between LED-64 and LED-128 are the number of steps (s) and the AddRoundKey operation, neither of which affects the overall architecture. Thus, the in-place architecture shown in Figure 5 applies equally to both the LED-64 and LED-128 quantum circuits ($s = 8$ and $s = 12$, respectively).

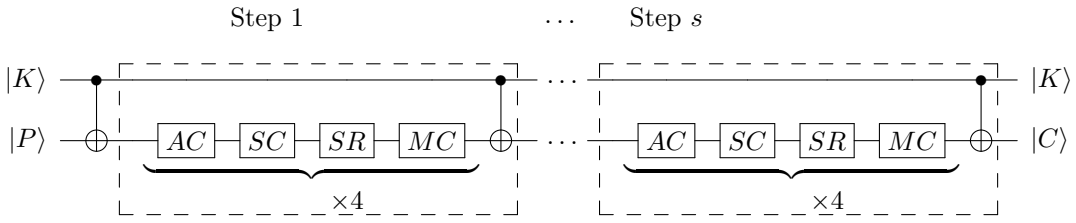


Figure 5: In-place architecture of LED quantum circuit.

4 Consideration for Modes of Operation

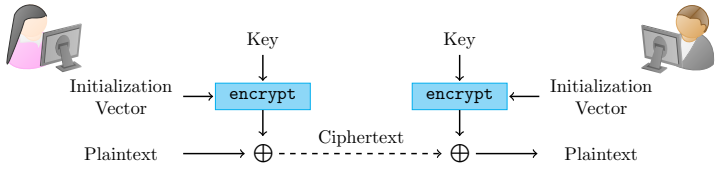
In classical cryptography, modes of operation were introduced to enhance security when using block ciphers to encrypt long messages. However, in the field of quantum cryptography, this topic has not received much attention. To address this gap, this work proposes several frameworks for modes of operation based on the quantum LED circuit discussed in Section 3. In this work, we focus on three widely utilized modes of operation: Electronic Codebook (ECB), Cipher Block Chaining (CBC), and Cipher Feedback (CFB). The detailed construction methods for these modes are illustrated in Figure 7.

One may note from [Bak21, Chapter 2.2.2] that the block cipher modes can be thought of as belonging from one of the two categories, non-symmetric and symmetric (see Figure 6). In the non-symmetric modes, both the sender and the recipient use two operations (one being the inverse of the other); whereas in symmetric modes, they use the same operation. In this paper, we take the representative modes; ECB and CBC from the non-symmetric category; and CFB from the symmetric category (analysis about the other modes, like OFB or CTR, will be similar to that of the CFB mode).

The following subsections will provide detailed implementations of the quantum frameworks for ECB, CBC, and CFB modes. For clarity and conciseness, the simplified representation of the proposed quantum LED implementation, as shown in Figure 8, will be utilized throughout the remaining part of this section.

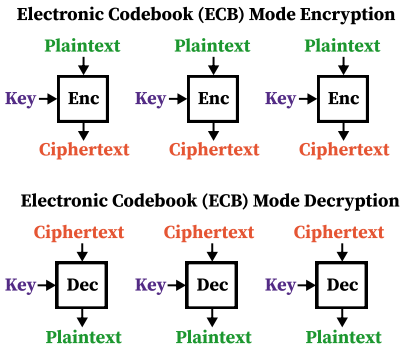
[4]The program took a long time since it searched for an implementation with 15 CNOT gates, yet there was no outcome.

(a) Non-symmetric.

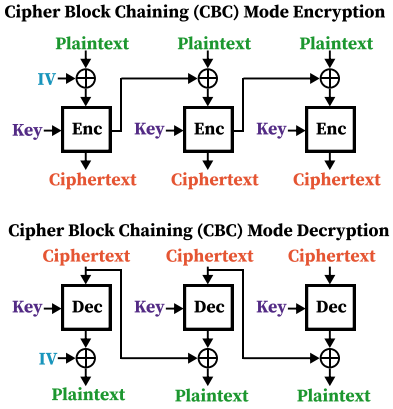


(b) Symmetric.

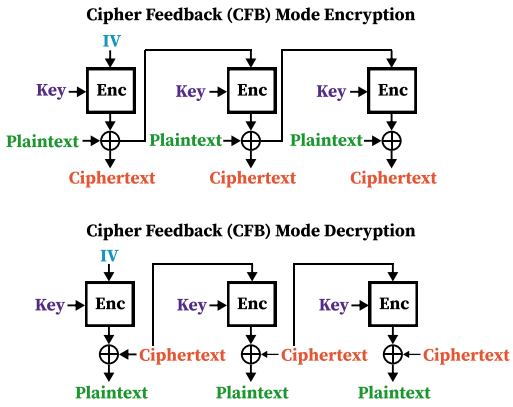
Figure 6: Two types of block cipher modes.



(a) ECB.



(b) CBC



(c) CFB

Figure 7: Typical modes of operations considered in this work

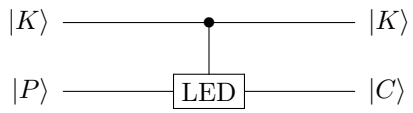


Figure 8: Simplified representation of LED in quantum

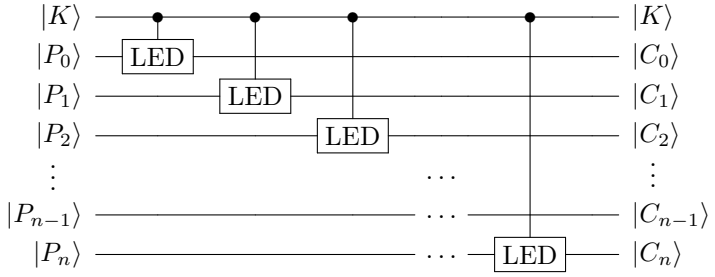


Figure 9: Framework for ECB Mode based on the proposed quantum LED.

4.1 Electronic Codebook Mode (ECB)

As shown in Figure 7(a), in ECB mode, each plaintext block is independently encrypted using the same block cipher, resulting in a corresponding ciphertext block. This mode is straightforward, as it directly applies the quantum cipher to each block without any interdependencies between them. However, the absence of inter-block dependencies in ECB can make it very vulnerable to various attacks.

The quantum framework for ECB mode is depicted in Figure 9, where each plaintext block is processed independently using the quantum LED circuit. In this diagram, K denotes the key, P denotes plaintext, and C denotes ciphertext.

4.2 Cipher Block Chaining Mode (CBC)

In CBC mode, interdependencies between plaintext blocks are established by XORing each plaintext block with the ciphertext of the previous block before encryption. For the first plaintext block, the XOR operation is performed with an initialization vector (IV), which is typically a random or pseudorandom value. This chaining mechanism ensures that identical plaintext blocks produce different ciphertexts, thereby enhancing security compared to ECB mode.

The quantum framework for CBC mode is shown in Figure 10, illustrating the process of XORing each plaintext block with the previous ciphertext block, followed by encryption using the quantum LED cipher. Here, IV denotes initialization vector.

4.3 Cipher Feedback Mode (CFB)

The CFB mode operates similarly to a stream cipher, converting a block cipher into a stream cipher. Particularly, this mode is suitable for real-time data encryption by allowing for the partial encryption of data streams. In CFB, the previous ciphertext block is encrypted using the block cipher, and the resulting output is then XORed with the current plaintext block to generate the new ciphertext block.

The quantum framework for implementing CFB mode is provided in Figure 11, where the feedback mechanism and sequential encryption process are illustrated. Although the

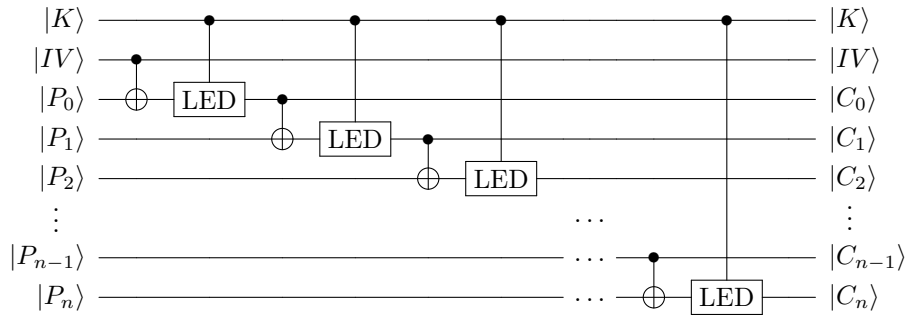


Figure 10: Framework for CBC Mode based on the proposed quantum LED.

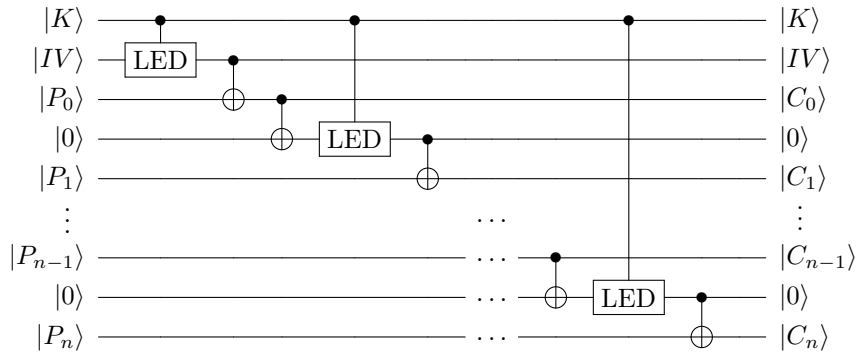


Figure 11: Framework for CFB Mode based on the proposed quantum LED.

proposed framework generates garbage qubits, all of them will be cleaned to $|0\rangle$ through the subsequent uncomputation step in Grover's attack.

In this section, we have proposed frameworks for ECB, CBC, and CFB modes based on our quantum LED implementations. Subsequently, Grover's attack estimation will be performed on the LED cipher under these proposed modes of operation frameworks to evaluate their effectiveness.

5 Results & Discussions

In this section, we conduct a thorough analysis of all the experimental results. To begin with, the issues in the previous quantum LED implementation proposed by Song et al. [SJS+23] are revised, thereby establishing a more accurate benchmark for subsequent comparisons. Following this, we compare the quantum resources required to implement the LED cipher with those needed for other block ciphers. Besides, this work provides the quantum attack resources based on the proposed quantum LED implementations, including Grover's attack oracles and STO attack oracles. At the end of this section, the effectiveness of Grover's attack oracles on various quantum modes of operation are carefully estimated, with the proposed quantum LED cipher serving as the underlying block cipher.

5.1 Revision of Previous Quantum LED Implementations

In the previous work [SJS+23], Song et al. also proposed a quantum implementation of the LED cipher. However, this work contains an issue in estimating the required number of qubits for the quantum implementation of LED cipher. In this subsection, we analyze the issue and provide a correction to ensure a more accurate comparison in subsequent content.

As shown in Table 3, the authors proposed an out-of-place naïve MixColumn. To perform one MixColumnsSerial, four MixColumn operations are required, with each MixColumn requiring 16 newly allocated output qubits, totaling 64 qubits for a single MixColumnsSerial. Since LED-64 consists of 32 rounds, each applying one MixColumnsSerial, 2048 qubits (32 rounds $\times$ 64) should be allocated for their implementation. However, their paper reports that only 142 qubits are required for the quantum LED-64 implementation without using any special techniques. This discrepancy arises from an issue in their MixColumnsSerial and naïve MixColumn implementation. To address this, we correctly allocate the output qubits for the naïve MixColumn and MixColumnsSerial in their implementation.

Table 4: Revised benchmark for the quantum implementation of LED-64

Quantum Circuit	#Qubits	#Toffoli	#CNOT	#X	Circuit depth
LED-64 [SJS+23]	142	2048	19008	1438	810
LED-64 [SJS+23] (Revised)	2176	2048	19008	1438	1147

In Table 4, we report the corrected quantum resource requirements for their LED-64 quantum circuit. In Song et al.'s paper [SJS+23], all submodules of the quantum LED cipher are implemented without ancilla qubits, except for MixColumnsSerial. Consequently, the revised benchmark includes an additional 2048 qubits from the corrected MixColumnsSerial, bringing the total to 2176 qubits when combined with the initial 128 qubits allocated for plaintext (64 qubits) and key (64 qubits). Furthermore, the circuit depth is revised from 810 to 1147.

5.2 Experiment 1: Cost Analysis for LED in Quantum

In this subsection, we present a detailed cost analysis of the proposed quantum implementations of LED cipher. Specifically, the comparison details between our designs and various implementations of quantum lightweight block ciphers from previous researches are provided in Table 5.

Table 5: Quantum rexesource requirements for LED and other lightweight block ciphers.

Quantum Circuit	#Qubits	#Toffoli	#CNOT	#X	Circuit depth
LED-64 (This work)	128	2048	8768	1158	753
LED-128 (This work)	192	3072	13120	1734	1127
LED-64 (Corrected) [SJS+23]	2176	2048	19008	1438	1147
PRESENT-64/80 [JSK+21]	144	2108	4683	1118	311
PRESENT-64/128 [JSK+21]	192	2232	4838	1164	311
GIFT-64/128 [JSK+21]	192	1792	1792	3261	308
GIFT-128/128 [JSK+21]	256	6144	6144	10,953	528
SIMON-64/128 [AMM20]	192	1408	7396	1216	2643
SIMON-128/128 [AMM20]	256	4352	17,152	4224	8427
SPECK-64/128 [JCK+20]	193	3286	9238	57	-
SPECK-128/128 [JCK+20]	257	7942	22,086	75	-
CHAM-64/128 [JCK+20]	196	2400	12,285	240	-
CHAM-128/128 [JCK+20]	268	4960	26,885	240	-

- **Our Work vs. Previous Quantum LED Design.** The proposed quantum LED-64 design in this work significantly outperforms the corrected LED-64 implementation discussed in Section 5.1, which represents the only previous quantum implementation of the LED cipher. Specifically, it requires fewer qubits, with a total of 128 compared to 2176 in the previous implementation, representing a reduction of approximately 94%. In terms of gate count, both designs use the same number of Toffoli gates, totaling 2048. However, our design shows clear advancements in other gate costs. Specifically, it utilizes fewer CNOT gates, with 8768 compared to 19008 in the previous implementation, resulting in a reduction of approximately 53.9%. Similarly, our design requires fewer X gates, with 1158 compared to 1438, which corresponds to a reduction of around 19.5%. Regarding circuit depth, our design has a depth of 753, whereas the previous implementation has a depth of 1147. This represents a reduction of approximately 34.4%, highlighting a significant improvement in time efficiency for our proposed design.
- **Our Work vs. Other Quantum Analysis Lightweight Block Ciphers.** Our quantum LED designs demonstrate significant advantages in efficiency when compared to other quantum lightweight block cipher implementations. For qubits, our quantum LED-64 and LED-128 designs utilize 128 and 192 qubits, respectively. Compared to all the other lightweight block ciphers mentioned in this section, our designs achieve the lowest qubit requirements. Compared to other quantum block ciphers, our proposed quantum LED designs exhibit competitive gate efficiency. For the most resource-intensive quantum gate, the Toffoli gate, both LED-64 and LED-128 maintain a relatively low count, comparable to quantum PRESENT implementations [JSK+21]. Additionally, our designs show comparable efficiency in CNOT and X gates, requiring fewer resources than ciphers such as SIMON [AMM20] and GIFT [JSK+21]. In terms of circuit depth, our designs exhibit a relatively high cost compared to other work, ranking just below the quantum Simon designs reported in [AMM20]. While our designs maintain competitive performance, there is still room for optimization in circuit depth.

In brief, our LED-64 and LED-128 quantum implementations demonstrate the lowest qubit requirements when compared to previous quantum block cipher implementations, while also achieving competitive gate cost and circuit depth.

5.3 Experiment 2: Grover's Attack on LED

This experiment focuses on estimating the quantum resources necessary for executing Grover's attack on the proposed LED ciphers. This process begins with a critical step—accurately estimating the quantum LED implementations. Specifically, the quantum LED-64 and LED-128 are implemented according to the methodology outlined in Section 3, followed by the decomposition of all involved Toffoli gates into Clifford and T gates to ensure precise resource estimation. Among various decomposition methods available [AMM+13, Sel13, HLZ+17], this paper adopts the approach introduced in Reference [AMM+13]. As shown in Figure 12, this method utilizes 8 Clifford and 7 T gates, resulting in a T-depth of 4 and a total depth of 8 for each Toffoli gate. Based on this specific decomposition strategy, we outlined the quantum resources required for the proposed LED implementations in Table 6.

In Tables 6 and 7 $TD/Td\text{-}M$ and $FD\text{-}M$ represent the trade-off performance of quantum circuits by being the product of the Toffoli/T depth and qubit count, and the full depth and qubit count, respectively. Additionally, we report the metrics $TD^2/Td^2\text{-}M$ and $FD^2\text{-}M$, which are major trade-off metrics for parallelization in Grover's search[5],

[5]Grover's key search demands extreme circuit depth due to the large number of iterations. Under the

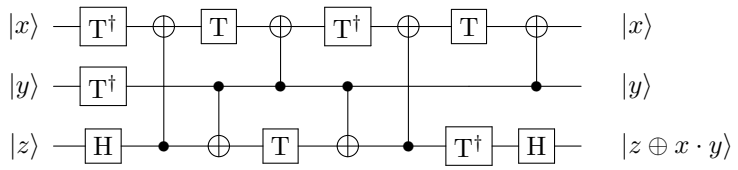


Figure 12: Decomposition for Toffoli gate (T-depth 4, full depth 8)

emphasizing the importance of depth.

Table 6: Decomposed resource requirements for proposed quantum LED implementations.

Circuit	#CNOT	#1qCliff	#T	Toffoli depth (TD)	#Qubit (M)	Full depth (FD)	TD-M	FD-M	TD^2-M	FD^2-M
LED-64	20032	5254	14336	128	128	1649	$1.00 \cdot 2^{14}$	$1.61 \cdot 2^{17}$	$1.00 \cdot 2^{21}$	$1.3 \cdot 2^{28}$
LED-128	30016	7878	21504	192	192	2471	$1.13 \cdot 2^{15}$	$1.81 \cdot 2^{18}$	$1.69 \cdot 2^{22}$	$1.09 \cdot 2^{30}$

Based on the quantum cost of the proposed LED implementations, the corresponding Grover attack cost can be directly estimated. As detailed in Section 2.3, recovering a k-bit key for a cipher using Grover's algorithm requires approximately $\sqrt{2^k}$ iterations of both the Grover oracle and the diffusion operator. Moreover, reference [BBHT98] provides a precise analysis, indicating that the optimal number of iterations is $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$. In this work, we adopt this estimate for our cost calculations.

Furthermore, we employed a widely used method [GLRS16, JNRV20] to simplify our analysis. Particularly, we disregard the diffusion operator from the cost estimation due to its minimal contribution. Consequently, the focus of Experiment 2 remains solely on the cost of the oracle, ensuring an accurate and efficient estimation process.

Next, we delve into the quantum resource analysis for Grover's attack oracle. This oracle consists of several components: the LED quantum circuit for encryption, an n-controlled NOT gate (where n represents the ciphertext size) for comparing the ciphertext with a known value, and the reverse operation of the previously executed LED quantum circuit for each subsequent iteration. It is crucial to emphasize the decomposition of the n-controlled NOT gate. Based on the approach outlined in Reference [WR14], this gate is estimated to require $(32 \cdot n - 64)$ T gates.

Overall, the estimated costs for Grover's attack oracles on LED are summarized in Table 7. The total cost required is approximated as $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor \times$(Table 6 $\times$ 2) + $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor \times$ $(32 \cdot n - 64)$ T gates. Since the iterations are executed sequentially, the number of qubits required remains consistent with the values presented in Table 6, with only one additional decision qubit required for ciphertext comparison.

Table 7: Quantum resource requirements for Grover's attack on LED.

Cipher	#Gate (G)	Full depth (FD)	T-depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
LED-64	$1.99 \cdot 2^{47}$	$1.27 \cdot 2^{43}$	$1.57 \cdot 2^{41}$	$1.01 \cdot 2^{7}$	$1.26 \cdot 2^{91}$	$1.28 \cdot 2^{50}$	$1.58 \cdot 2^{48}$	$1.61 \cdot 2^{93}$	$1.24 \cdot 2^{90}$
LED-128	$1.47 \cdot 2^{81}$	$1.90 \cdot 2^{75}$	$1.18 \cdot 2^{74}$	$1.00 \cdot 2^{8}$	$1.39 \cdot 2^{157}$	$1.90 \cdot 2^{83}$	$1.18 \cdot 2^{82}$	$1.80 \cdot 2^{159}$	$1.39 \cdot 2^{156}$

constraint of circuit depth (such as the MAXDEPTH parameter introduced by NIST [NIS16, NIS22]), parallelization of Grover's search is required to reduce the circuit depth, but its performance is poor. As a result, the product of the squared depth and qubit count serves as the trade-off metric in Grover's parallelization. See Section 5.3 for more details.

NIST Post-Quantum Security Levels

To assess the security of a cipher against quantum attacks, NIST [NIS16, NIS22] has defined security bounds for various levels:

- Level 1: The resource requirements for an attack are comparable to those for breaking AES-128: 2^{170} (stated) $\rightarrow$ $\mathbf{2^{157}}$ (state-of-the-art [JBS+22]).
- Level 3: The resource requirements for an attack are comparable to those for breaking AES-192: 2^{233} (stated) $\rightarrow$ $\mathbf{2^{221}}$ (state-of-the-art [JBS+22]).
- Level 5: The resource requirements for an attack are comparable to those for breaking AES-256: 2^{298} (stated) $\rightarrow$ $\mathbf{2^{285}}$ (state-of-the-art [JBS+22]).

Based on the cost estimates for Grover's key search against AES variants as presented by Grassl et al. [GLRS16], NIST determined the quantum attack complexities for Levels 1, 3 and 5 (corresponding to different AES variants) as 2^{170}, 2^{233} and 2^{298}, respectively (calculated as total gates multiplied by the depth of Grover's search). It is important to highlight that NIST's complexity estimates in [NIS16] are derived from research results published in PQCrypto'16 [GLRS16]. Since that time, quantum circuits for AES have undergone continuous optimization, resulting in a significant decrease in the cost of attacks in recent years [JNRV20, ZWS+20, JBS+22, HS22].

NIST also acknowledges that the attack complexities based on these levels are relative, given the ongoing optimizations in quantum circuits for AES (see [NIS16, Page 17]). Therefore, if a more efficient attack is proposed, the benchmarks may need to be updated.

Recently, NIST revised the security bounds for AES [NIS22] following the findings presented at Eurocrypt 2020 [JNRV20]. In [JNRV20], the quantum attack costs for AES-128, -192 and -256 were significantly reduced to 2^{157}, 2^{221} and 2^{285}, respectively; aligning with the updated values in [NIS22].

It is worth noting that the newly updated costs have an issue, as the estimation in [JNRV20] was found to be incorrect (the corrected estimation is in [JNRV19]). However, the updated costs presented in [NIS22] are also achievable in [JBS+22] through their optimized quantum circuits for AES (see Table 8). Thus, we still use the security bounds from [NIS22] to assess the post-quantum security of LED.

In Table 8, the evaluation of post-quantum security levels for LED-64 and LED-128 is presented. As expected, LED-64 cannot achieve the required security level due to its 64-bit key length. However, LED-128 offers the same level of difficulty as breaking AES-128 using Grover's key search algorithm, thereby achieving the Level-1 post-quantum security.

Table 8: Comparison of the Grover's key search costs.

Post-quantum Security	NIST'16 [NIS16] (based on [GLRS16])	NIST'22 [NIS22] (based on [JNRV20])	Jang et al. [JBS+22]	Table 7 (G-FD)	
				LED-64	LED-128
Level-1 (AES-128)	2^{170}	2^{157}	2^{156}	Not achieved (2^{91})	Level 1 (2^{157})
Level-3 (AES-192)	2^{233}	2^{221}	2^{221}		
Level-5 (AES-256)	2^{298}	2^{285}	2^{286}		

5.4 Experiment 3: STO Attack on Quantum LED

In this experiment, we present a detailed analysis of the STO attack on the proposed quantum LED design, with associated quantum costs outlined in Table 9.

- **STO Attack on Quantum LED-64.** The STO attack does not offer significant advantages over the Grover's oracle for attacking the LED-64 cipher. The parameter r is calculated as $r = \lceil k/n \rceil$, where k is the key size and n is the state size. A quantum

Table 9: Quantum resource requirements for one oracle (Grover) and two oracles (STO) attacks on LED.

Cipher	#Gate (G)	Full depth (FD)	T-depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
LED-64 (Grover)	$1.99 \cdot 2^{47}$	$1.27 \cdot 2^{43}$	$1.57 \cdot 2^{41}$	$1.01 \cdot 2^{7}$	$1.26 \cdot 2^{91}$	$1.28 \cdot 2^{50}$	$1.58 \cdot 2^{48}$	$1.61 \cdot 2^{93}$	$1.24 \cdot 2^{90}$
LED-64 (STO)	$1.99 \cdot 2^{47}$	$1.27 \cdot 2^{43}$	$1.57 \cdot 2^{41}$	$1.01 \cdot 2^{7}$	$1.26 \cdot 2^{91}$	$1.28 \cdot 2^{50}$	$1.58 \cdot 2^{48}$	$1.61 \cdot 2^{93}$	$1.24 \cdot 2^{90}$
LED-128 (Grover)	$1.47 \cdot 2^{81}$	$1.90 \cdot 2^{75}$	$1.18 \cdot 2^{74}$	$1.00 \cdot 2^{8}$	$1.39 \cdot 2^{157}$	$1.90 \cdot 2^{83}$	$1.18 \cdot 2^{82}$	$1.80 \cdot 2^{159}$	$1.39 \cdot 2^{156}$
LED-128 (STO)	$1.03 \cdot 2^{81}$	$1.33 \cdot 2^{76}$	$1.65 \cdot 2^{74}$	$1.51 \cdot 2^{7}$	$1.36 \cdot 2^{157}$	$1.99 \cdot 2^{83}$	$1.24 \cdot 2^{82}$	$1.32 \cdot 2^{160}$	$1.02 \cdot 2^{157}$

attack on a block cipher requires at least r pairs of plaintext and corresponding ciphertext, where k denotes the key size and n denotes the block size. For LED-64, with both the block size and key size set to 64 bits, the required r is only 1. Hence, the Grover's attack oracle remains equally efficient when compared to the cheap oracle O_γ in STO attack. In brief, the STO attack does not provide a substantial advantage over Grover's attack oracle for the LED-64 cipher.

- **STO Attack on Quantum LED-128.** The STO attack offers significant improvement over Grover's attack for the LED-128 cipher. For LED-128, with a key size of 128 bits and a block size of 64 bits, the required number of rounds r is 2. Hence, the STO attack benefits from the cheap oracle O_γ, which is much more efficent compared to the corresponding Grover's oracle. While the STO attack requires a higher depth, it compensates by reducing the number of gates and qubits involved. This efficiency in terms of gate count and qubit cost highlights the advantages of the STO approach over Grover's attack for LED-128.

Overall, we have successfully implemented the STO attack on the proposed quantum LED ciphers, resulting in improved efficiency compared to Grover's attack. Although the STO attack does not offer significant advantages for LED-64, it is clear that for LED-128, STO attack oracle requires fewer gates and qubits, though with an increase in circuit depth. Moreover, the product of Gate and Full depth (G-FD) decreases, further highlighting the benefits of the STO approach in attacking LED-128.

5.5 Experiment 4: Grover's Attack on LED across Various Modes of Operation

In the last experiment, we perform Grover's attack on the proposed quantum LED cipher across multiple modes of operation, namely ECB, CBC and CFB.

Before estimating the resources required for Grover's attack, we first analyze the quantum costs associated with the different modes of operation frameworks discussed in Section 4. According to these frameworks, Table 10 provides the detailed quantum resource requirements for η-block quantum LED encryption across different modes of operation, showing how the cost of each quantum framework varies with block size η.

Table 10: Quantum resource requirements for LED under modes of operation with η blocks.

Quantum Circuit	#Qubits	#Toffoli	#CNOT	#X	Circuit depth
LED-64 (ECB)	$64 + 64\eta$	2048η	8768η	1158η	753η
LED-128 (ECB)	$128 + 64\eta$	3072η	13120η	1734η	1127η
LED-64 (CBC)	$128 + 64\eta$	2048η	8832η	1158η	754η
LED-128 (CBC)	$192 + 64\eta$	3072η	13184η	1734η	1128η
LED-64 (CFB)	$64 + 128\eta$	2048η	$8896\eta - 64$	1158η	$755\eta - 1$
LED-128 (CFB)	$128 + 128\eta$	3072η	$13248\eta - 64$	1734η	$1129\eta - 1$

However, regardless of the mode of operation used, Grover's attack only needs to be applied to the first block of the cipher. This is because all blocks within a specific mode generally share the same key. Consequently, as shown in Table 11, the cost estimation for the quantum attack exhibits only minor differences across different modes of operation.

Table 11: Quantum Resource Requirements for Grover's Attack on LED across various modes of operations.

Cipher	#Gate (G)	Full depth (FD)	T-depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
LED-64 (ECB)	$1.99 \cdot 2^{47}$	$1.27 \cdot 2^{43}$	$1.57 \cdot 2^{41}$	$1.01 \cdot 2^{7}$	$1.26 \cdot 2^{91}$	$1.28 \cdot 2^{50}$	$1.58 \cdot 2^{48}$	$1.61 \cdot 2^{93}$	$1.24 \cdot 2^{90}$
LED-128 (ECB)	$1.47 \cdot 2^{81}$	$1.90 \cdot 2^{75}$	$1.18 \cdot 2^{74}$	$1.00 \cdot 2^{8}$	$1.39 \cdot 2^{157}$	$1.90 \cdot 2^{83}$	$1.18 \cdot 2^{82}$	$1.80 \cdot 2^{159}$	$1.39 \cdot 2^{156}$
LED-64 (CBC)	$1.99 \cdot 2^{47}$	$1.27 \cdot 2^{43}$	$1.57 \cdot 2^{41}$	$1.51 \cdot 2^{7}$	$1.26 \cdot 2^{91}$	$1.91 \cdot 2^{50}$	$1.18 \cdot 2^{49}$	$1.20 \cdot 2^{94}$	$1.85 \cdot 2^{90}$
LED-128 (CBC)	$1.47 \cdot 2^{81}$	$1.90 \cdot 2^{75}$	$1.18 \cdot 2^{74}$	$1.25 \cdot 2^{8}$	$1.39 \cdot 2^{157}$	$1.19 \cdot 2^{84}$	$1.475 \cdot 2^{82}$	$1.13 \cdot 2^{160}$	$1.74 \cdot 2^{156}$
LED-64 (CFB)	$1.99 \cdot 2^{47}$	$1.27 \cdot 2^{43}$	$1.57 \cdot 2^{41}$	$1.51 \cdot 2^{7}$	$1.26 \cdot 2^{91}$	$1.91 \cdot 2^{50}$	$1.18 \cdot 2^{49}$	$1.20 \cdot 2^{94}$	$1.85 \cdot 2^{90}$
LED-128 (CFB)	$1.47 \cdot 2^{81}$	$1.90 \cdot 2^{75}$	$1.18 \cdot 2^{74}$	$1.25 \cdot 2^{8}$	$1.39 \cdot 2^{157}$	$1.19 \cdot 2^{84}$	$1.475 \cdot 2^{82}$	$1.13 \cdot 2^{160}$	$1.74 \cdot 2^{156}$

In brief, the cost of Grover's attack depends solely on the first block, and the quantum resource requirements for this block vary only slightly between ECB, CBC and CFB. Therefore, there is minimal variation in the overall cost of Grover's attack across different modes of operation.

6 Conclusion

In this work, we have designed efficient quantum implementation of the variants of LED block cipher (LED-64 and LED-128). By optimizing the quantum circuits through minimization of qubits, quantum gates, and circuit depth, we were able to estimate the least quantum resources required for executing the Grover's attack and the STO attack (which is more difficult than Grover's, but offers quantum advantage for LED-128). Furthermore, we have successfully implemented ECB, CBC and CFB modes for quantum LED; this kind of analysis was not done in the past, to best of our knowledge.

A Per-Step Benchmarks

We present the per-step (4 rounds) resource benchmarks for our quantum circuits of LED-64 and LED-128 in Tables 12 and 13, respectively. Note that the initial key XOR is excluded from Tables 12 and 13.

Table 12: Quantum resources required per step (4 rounds) for LED-64.

Step	#CNOT	#NOT	#Toffoli	Toffoli depth (TD)	Circuit depth
1	1088	136	256	16	96
2	1088	148	256	16	96
3	1088	152	256	16	96
4	1088	146	256	16	96
5	1088	144	256	16	96
6	1088	148	256	16	95
7	1088	136	256	16	95
8	1088	148	256	16	96

Table 13: Quantum resources required per step (4 rounds) for LED-128.

Step	#CNOT	#NOT	#Toffoli	Toffoli depth (TD)	Circuit depth
1	1088	136	256	16	95
2	1088	148	256	16	96
3	1088	152	256	16	96
4	1088	146	256	16	96
5	1088	144	256	16	96
6	1088	148	256	16	95
7	1088	136	256	16	95
8	1088	148	256	16	96
9	1088	142	256	16	96
10	1088	146	256	16	96
11	1088	148	256	16	96
12	1088	140	256	16	95

References

[AMM+13] Matthew Amy, Dmitri Maslov, Michele Mosca, Martin Roetteler, and Martin Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6):818–830, Jun 2013. URL: http://dx.doi.org/10.1109/TCAD.2013.2244643. 5, 16

[AMM20] Ravi Anand, Arpita Maitra, and Sourav Mukhopadhyay. Grover on simon. *Quantum Information Processing*, 19(9), September 2020. URL: http://dx.doi.org/10.1007/s11128-020-02844-w, doi:10.1007/s11128-020-02844-w. 15, 16

[Bak21] Anubhab Baksi. *Classical and Physical Security of Symmetric Key Cryptographic Algorithms*. PhD thesis, School of Computer Science & Engineering, Nanyang Technological University, Singapore, 2021. https://dr.ntu.edu.sg/handle/10356/152003. 11

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, Jun 1998. URL: http://dx.doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P, doi:10.1002/(sici)1521-3978(199806)46:4/5<493::aid-prop493>3.0.co;2-p. 17

[BCC+24] Anubhab Baksi, Sumanta Chakraborty, Anupam Chattopadhyay, Matthew Chun, SK Hafizul Islam, Kyungbae Jang, Hyunji Kim, Yujin Oh, Soham Roy, Hwajeong Seo, and Siyi Wang. Quantum implementation of linear and non-linear layers. *IEEE International System-on-Chip Conference (SOCC)*, 2024. 10

[BJ24] Anubhab Baksi and Kyungbae Jang. *Quantum Computing Fundamental and Cryptographic Perspective*, pages 7–20. Springer Nature Singapore, Singapore, 2024. doi:10.1007/978-981-97-0025-7_2. 4

[BKD21] Anubhab Baksi, Banashri Karmakar, and Vishnu Asutosh Dasu. POSTER: optimizing device implementation of linear layers with automated tools. In *Applied Cryptography and Network Security Workshops - ACNS 2021 Satellite Workshops, Kamakura, Japan, June 21-24, 2021, Proceedings*, volume 12809

of *Lecture Notes in Computer Science*, pages 500–504. Springer, 2021. doi:10.1007/978-3-030-81645-2_30. 10

[BKL+07] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *CHES*, volume 4727, pages 450–466. Springer, 2007. 3, 7

[CBC23] Matthew Chun, Anubhab Baksi, and Anupam Chattopadhyay. Dorcis: Depth optimized quantum implementation of substitution boxes. Cryptology ePrint Archive, Paper 2023/286, 2023. https://eprint.iacr.org/2023/286. URL: https://eprint.iacr.org/2023/286. 4, 7, 8, 9

[CGL22] BinBin Cai, Fei Gao, and Gregor Leander. Quantum attacks on two-round even-mansour. *Frontiers in Physics*, 10:1028014, 2022. 7, 8, 9

[CHM11] Nicolas T. Courtois, Daniel Hulme, and Theodosis Mourouzis. Solving circuit optimisation problems in cryptography and cryptanalysis. Cryptology ePrint Archive, Paper 2011/475, 2011. https://eprint.iacr.org/2011/475. URL: https://eprint.iacr.org/2011/475. 7, 8, 9

[CLF+24] Jingwen Chen, Qun Liu, Yanhong Fan, Lixuan Wu, Boyun Li, and Meiqin Wang. New SAT-based model for quantum circuit decision problem: Searching for low-cost quantum implementation. *IACR Communications in Cryptology*, 1(1), 2024. doi:10.62056/anmmp-4c2h. 8, 9

[CXL+14] Shanzhi Chen, Hui Xu, Dake Liu, Bo Hu, and Hucheng Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, 1(4):349–359, 2014. 1

[DBSC19] Vishnu Asutosh Dasu, Anubhab Baksi, Sumanta Sarkar, and Anupam Chattopadhyay. Lighter-r: Optimized reversible circuit implementation for sboxes. *2019 32nd IEEE International System-on-Chip Conference (SOCC)*, pages 260–265, 2019. URL: https://api.semanticscholar.org/CorpusID:218564036. 7, 8, 9

[DP20] James H Davenport and Benjamin Pring. Improvements to quantum search techniques for block-ciphers, with applications to aes. In *International Conference on Selected Areas in Cryptography*, pages 360–384. Springer, 2020. 2, 6

[FWZ+24] Jingya Feng, Yongzhuang Wei, Fengrong Zhang, Enes Pasalic, and Yu Zhou. Novel optimized implementations of lightweight cryptographic s-boxes via sat solvers. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(1):334–347, 2024. doi:10.1109/TCSI.2023.3325559. 7, 8, 9

[GLRS16] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. Applying Grover's algorithm to AES: Quantum resource estimates. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography*, pages 29–43, Cham, 2016. Springer International Publishing. 17, 18

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In *Cryptographic Hardware and Embedded Systems–CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1, 2011. Proceedings 13*, pages 326–341. Springer, 2011. 2

[HLZ+17] Yong He, Ming-Xing Luo, E Zhang, Hong-Ke Wang, and Xiao-Feng Wang. Decompositions of n-qubit toffoli gates with linear circuit complexity. *International Journal of Theoretical Physics*, 56(7):2350–2361, 2017. 16

[HS22] Zhenyu Huang and Siwei Sun. Synthesizing quantum circuits of aes with lower t-depth and less qubits. Cryptology ePrint Archive, Report 2022/620, 2022. https://eprint.iacr.org/2022/620. 18

[JBS+22] Kyungbae Jang, Anubhab Baksi, Gyeongju Song, Hyunji Kim, Hwajeong Seo, and Anupam Chattopadhyay. Quantum analysis of aes. Cryptology ePrint Archive, Paper 2022/683, 2022. https://eprint.iacr.org/2022/683. 1, 18

[JCK+20] Kyoungbae Jang, Seungju Choi, Hyeokdong Kwon, Hyunji Kim, Jaehoon Park, and Hwajeong Seo. Grover on korean block ciphers. *Applied Sciences*, 10(18), 2020. URL: https://www.mdpi.com/2076-3417/10/18/6407, doi:10.3390/app10186407. 15

[JCKS20] Kyungbae Jang, Seungjoo Choi, Hyeokdong Kwon, and Hwajeong Seo. Grover on SPECK: Quantum resource estimates. Cryptology ePrint Archive, Report 2020/640, 2020. https://eprint.iacr.org/2020/640. 1

[JNRV19] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing grover oracles for quantum key search on aes and lowmc, 2019. URL: https://arxiv.org/abs/1910.01700, arXiv:1910.01700. 18

[JNRV20] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing grover oracles for quantum key search on AES and lowmc. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020, Zagreb, Croatia, May 10-14, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 280–310. Springer, 2020. doi:10.1007/978-3-030-45724-2_10. 17, 18

[JSK+21] Kyungbae Jang, Gyeongju Song, Hyunjun Kim, Hyeokdong Kwon, and Hwajeong Seo. Efficient implementation of present and gift on quantum computers. *Applied Sciences*, 11:4776, 05 2021. doi:10.3390/app11114776. 1, 15, 16

[KLL15] Shelby Kimmel, Cedric Yen-Yu Lin, and Han-Hsuan Lin. Oracles with costs. *Communication and Cryptography, TQC*, 2015. 2, 6

[LL15] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business horizons*, 58(4):431–440, 2015. 1

[LPZW23] Qun Liu, Bart Preneel, Zheng Zhao, and Meiqin Wang. Improved quantum circuits for AES: Reducing the depth and the number of qubits. Cryptology ePrint Archive, Paper 2023/1417, 2023. URL: https://eprint.iacr.org/2023/1417. 1

[NIS16] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf. 17, 18

[NIS22] NIST. Call for additional digital signature schemes for the post-quantum cryptography standardization process, 2022. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf. 17, 18

[Sel13] Peter Selinger. Quantum circuits of t-depth one. *Physical Review A*, 87(4):042302, 2013. 16

[SF24] Haotian Shi and Xiutao Feng. Quantum circuits of AES with a low-depth linear layer and a new structure. Cryptology ePrint Archive, Paper 2024/381, 2024. URL: https://eprint.iacr.org/2024/381. 1

[SHT16] Damian Steiger, Thomas Häner, and Matthias Troyer. Projectq: An open source software framework for quantum computing. *Quantum*, 2, 12 2016. doi:10.22331/q-2018-01-31-49. 2

[SJS+23] Min-ho Song, Kyung-bae Jang, Gyeong-ju Song, Won-woong Kim, and Hwa-Jeong Seo. Quantum circuit implementation of the led block cipher with compact qubit. *Journal of the Korea Institute of Information Security & Cryptology*, 33(3):383–389, 2023. 2, 10, 14, 15

[vH19] Iggy van Hoof. Space-efficient quantum multiplication of polynomials for binary finite fields with sub-quadratic toffoli gate count. *arXiv preprint arXiv:1910.02849*, 2019. 10

[WR14] Nathan Wiebe and Martin Roetteler. Quantum arithmetic and numerical analysis using repeat-until-success circuits. *arXiv preprint arXiv:1406.2040*, 2014. 17

[XZL+20] Zejun Xiang, Xiangyong Zeng, Da Lin, Zhenzhen Bao, and Shasha Zhang. Optimizing implementations of linear layers. *IACR Transactions on Symmetric Cryptology*, 2020. 10

[YWS+24] Yufei Yuan, Wenling Wu, Tairong Shi, Lei Zhang, and Yu Zhang. A framework to improve the implementations of linear layers. *IACR Transactions on Symmetric Cryptology*, 2024(2):322–347, June 2024. URL: https://tosc.iacr.org/index.php/ToSC/article/view/11633. 10

[ZWS+20] Jian Zou, Zihao Wei, Siwei Sun, Ximeng Liu, and Wenling Wu. Quantum circuit implementations of AES with fewer qubits. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 697–726, Cham, 2020. Springer International Publishing. 18