IACR Communications in Cryptology
ISSN 3006-5496, Vol. 1, No. 1, 24 pages.

https://doi.org/10.62056/anmmp-4c2h

New SAT-based Model for Quantum Circuit Decision Problem: Searching for Low-Cost Quantum Implementation

Jingwen Chen[1,2], Qun Liu[1,2], Yanhong Fan[1,2], Lixuan Wu[1,2], Boyun Li[1,2] and Meiqin Wang[a, 3,1,2]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, China
[2] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[3] Quan Cheng Shandong Laboratory, Jinan, China

Abstract. In recent years, quantum technology has been rapidly developed. As security analyses for symmetric ciphers continue to emerge, many require an evaluation of the resources needed for the quantum circuit implementation of the encryption algorithm. In this regard, we propose the *quantum circuit decision problem*, which requires us to determine whether there exists a quantum circuit for a given permutation f using M ancilla qubits and no more than K quantum gates within the circuit depth D. Firstly, we investigate heuristic algorithms and classical SAT-based models in previous works, revealing their limitations in solving the problem. Hence, we innovatively propose an improved SAT-based model incorporating three metrics of quantum circuits. The model enables us to find the optimal quantum circuit of an arbitrary 3 or 4-bit S-box under a given optimization goal based on SAT solvers, which has proved the optimality of circuits constructed by the tool, `LIGHTER-R`. Then, by combining different criteria in the model, we find more compact quantum circuit implementations of S-boxes such as `RECTANGLE` and `GIFT`. For `GIFT` S-box, our model provides the optimal quantum circuit that only requires 8 gates with a depth of 31. Furthermore, our model can be generalized to linear layers and improve the previous SAT-based model proposed by Huang et al. in ASIACRYPT 2022 by adding the criteria on the number of qubits and the circuit depth.

Keywords: Quantum circuit · Decision problem · SAT · S-box · Linear layer

1 Introduction

Recently, the rapid development of quantum information technology has had a profound impact on the security of data and communication in cyberspace. To address the emerging challenges in the post-quantum era, the National Institute of Standards and Technology (NIST) initiated a public solicitation for post-quantum cryptography algorithms in 2016 and used the complexity of the quantum key search circuit of AES as the benchmark for the classification of post-quantum public key cryptography algorithms [CJL+16]. Regarding symmetric ciphers, Grover's algorithm [Gro96] offers a quadratic speedup in searching for the correct key given a pair of plaintext and ciphertext. Subsequently, the security analysis for symmetric ciphers has been actively explored, with many proposals requiring quantum

E-mail: jingwenchen@mail.sdu.edu.cn (Jingwen Chen), qunliu@mail.sdu.edu.cn (Qun Liu), yanhongfan@sdu.edu.cn (Yanhong Fan), lixuanwu@mail.sdu.edu.cn (Lixuan Wu), 202316987@mail.sdu.edu.cn (Boyun Li), mqwang@sdu.edu.cn (Meiqin Wang)

[a]This is the corresponding author.

Received: 2024-01-09 Accepted: 2024-03-05

circuit implementation for encryption algorithms. So the quantum circuit implementation of a cipher has attracted significant attention within the cryptography community.

The fundamental unit in optimizing the circuits of ciphers is the circuitry tailored to optimize cryptographic components, such as S-boxes and linear layers. In classical circuit design, we typically employ various optimization techniques to reduce metrics. The most popular one should be the gate equivalents (GE) required by the chip-level implementation of the ciphers. GE effectively approximates the complexity of digital electronic circuits. Generally, two components are relevant to the cost. On the other hand, as an important criterion, latency has been attracting more and more attention. Many of the applications require low latency, including automobiles, robots, or mission-critical computation applications. It impacts the throughput of encryption/decryption and plays an important role in the low-energy consideration of ciphers [BBI+15]. The S-box is one of the most popular confusion components of symmetric-key ciphers. Many tools are proposed to optimize the primitive, such as `LIGHTER` [JPST17] and `PEIGEN` [BGLS19]. In addition, the diffusion components are essential matrices and are the most well-known diffusion components. Although it has been shown to be an NP-hard problem [BMP08], there is still a growing body of work solely concentrating on decreasing the GE. More and more concerns for heuristics searching for sub-optimal solutions have arisen (see [BMP13, KLSW17, TP20, XZL+20, BDK+21, LWF+22, LWS+22] for an incomplete list).

When implementing a quantum circuit for a certain boolean function, the gates under the classical circuit are usually converted into reversible quantum gates, as the quantum operation is reversible. However, it is not straightforward to directly translate optimized classical circuits into quantum circuits. Due to the unique characteristics of quantum circuits, employing such a direct conversion method may result in increased resource consumption. There are three common quantum gates called the NCT gate set, the Pauli-X gate, the CNOT gate, and the Toffoli gate, which can respectively replace the NOT gate, XOR gate, and AND gate under the classical circuit. Several factors influence the complexity of quantum circuits, including the number of quantum gates, qubits, circuit depth, and so on. Currently, the quantum circuit of the symmetric ciphers mainly takes AES as the research object [GLRS15, LPS20, ASAM18, JNRV20, ZWS+20, HS22, JBS+22, LPZW23]. In 2016, Grassl et al. [GLRS15] presented a comprehensive quantum implementation scheme that utilized the Grover algorithm to exhaustively search for AES keys. They analyzed the quantum resource overhead, including the size of the quantum circuit, the number of qubits needed, and the depth of the quantum circuit. Subsequently, researchers have continuously proposed and optimized AES quantum circuits using different circuit structures and considering various quantum circuit metrics.

For the optimization of quantum circuits of symmetric primitives, it is very important to optimize the cryptographic components. As the only nonlinear part in block ciphers, the S-box has always been the main content of the research. Based on the public tool, `LIGHTER` [JPST17], which was proposed under the background of classical computing and generated in-place implementation, Dasu et al. proposed `LIGHTER-R` [DBSC19] which can give a reversible circuit implementation for a specific 4-bit S-box, along with the optimization for gate cost. Subsequently, Chun et al. proposed the `DORCIS` tool [CBC23] to find depth-optimized quantum circuit implementations for arbitrary 3-/4-bit S-boxes. For linear transformations, the heuristic algorithm proposed in [XZL+20] can be used to find efficient implementations of binary matrices of size up to 32 under s-Xor metric, which is equivalent to the CNOT gate in the quantum circuit. In 2022, Huang et al. [HS22] presented a SAT-based method to generate the most compact CNOT circuit for invertible linear transformations over $\mathbb{F}_2^n$.

1.1 Our Contributions

In this paper, we investigate three metrics, the number of qubits, the number of gates, and the circuit depth, which are essential for quantum circuits. To assess the optimality of a quantum circuit of a permutation f, we present the *quantum circuit decision problem*, which requires us to determine whether there exists a quantum circuit for the permutation f using M ancilla qubits and no more than K quantum gates within the circuit depth D.

Analysis of previous related methods. In order to answer the quantum circuit decision problem, we investigate and analyze a series of methods both in the classical and quantum circuits in the literature. They predominantly fall into two categories: the heuristic algorithm employed to search for reversible quantum circuits, such as LIGHTER-R [DBSC19] and DORCIS [CBC23], and the SAT-based classical circuit optimization model, such as Lu et al.'s model [LWH$^+$21] and Huang et al.'s model [HS22]. However, we observe that these tools are unable to comprehensively answer this question, namely, to prove the optimal quantum circuit for a permutation. DORCIS is a highly practical heuristic tool, effectively reducing the depth of quantum circuits. If we require whether a circuit is optimal, we will not receive an answer as it is a heuristic algorithm. Ancilla qubits are not allowed in the tool, which also renders it incapable of addressing the problem regarding the number of qubits. On the other hand, the SAT-based model can prove the optimality in the classical setting. However, in the quantum setting, the properties of non-fanout and reversibility render this model incapable of addressing the question.

Proposal of our new SAT model. Building upon the above discussion, we introduce the improved SAT-based quantum circuit optimization model based on the model in [Sto16, LWH$^+$21] and present the detailed coding schemes. For the convenience of using the model, we provide fully automated code generation. Our model incorporates three metrics M, K, and D, related to the number of qubits, the number of gates, and the circuit depth of the quantum circuit decision problem.

Applications on permutations. Firstly, we discuss the quantum circuit decision problem about S-box. By controlling the number of qubits, we can apply the model to find the smallest gate-count implementation of the S-box, proving the results in LIGHTER-R. We can also provide circuit implementations of S-box with odd permutation by adding qubits, which solves the situations that the original model couldn't handle. Finally, by combining three metrics, we have achieved the currently optimal circuit for quantum S-boxes. Table 1 and Table 2 show the comparison results with LIGHTER-R and DORCIS, respectively. All of our experiments are running on AMD EPYC 7302 CPU 3.0Hz with 8-core. For example, for RECTANGLE S-box, our model proves that the optimal quantum circuit only requires 10 gates with depth 32, which represents the current optimal circuit, surpassing both LIGHTER-R and DORCIS. Notably, in Table 2, for optimizing GIFT S-box, DORCIS uses 13 gates with depth 31. We can find a new quantum circuit with the same depth using only 8 gates. Furthermore, if we allow an ancilla qubit in the circuit, the depth can be reduced to 30. Next, we consider a specific type of permutation, the linear permutation, which is commonly used in the linear layers. For the in-place circuit (no ancilla qubits), in ASIACRYPT 2022, Huang et al. [HS22] proposed a SAT-based model to find the lower bound of the number of quantum CNOT gates. Apart from the number of CNOT gates, our model can control the circuit depth and the number of qubits. All the source codes and results of this paper are available at https://github.com/Chenjingwen-cyber/Sample_implementation.

Table 1: Result summary of `LIGHTER-R` and our model.

S-box	LUT	`LIGHTER-R`		**Our model**		
		Gate	Depth	Gate	Depth	Time
`RECTANGLE` [ZBL+15]	65CA1E79B03D8F42	10	34	10	**32**	70 s
`GIFT` [BPP+17]	1A4C6F392DB7508E	8	32	8	**31**	18 s
`PRESENT` [SOT+21]	C56B90AD3EF84712	11	33	11	**32**	28 s
`SKINNY` [BJK+16]	C6901A2B385D4E7F	10	33	10	**31**	42 s
`MIDORI` [BBI+15]	CAD3EBF789150246	10	33	10	**31**	210 s
`Multiplicative Inverse in` $\mathbb{F}_2^4$ [LXX+23]	062493D51EC78ABF	/	/	**10***	**44**	365 s

* These circuits use an ancilla qubit.

Table 2: Result summary of `DORCIS` and our model.

S-box	LUT	`DORCIS`		**Our model**		
		Gate	Depth	Gate	Depth	Time
`RECTANGLE` [ZBL+15]	65CA1E79B03D8F42	11	32	**10**	32	70 s
`GIFT` [BPP+17]	1A4C6F392DB7508E	13	31	**8**	31	18 s
`GIFT` [BPP+17]	1A4C6F392DB7508E	13	31	**9***	**30**	190 s
`ELEPHANT` [BCDM21]	EDB0214F7A859C36	13	33	**12**	33	225 s
`LBLOCK` [WZ11]	E9F0D4AB128376C5	10	31	**9**	31	47 s
`UBLOCK` [WL21]	749CBAD8FE160325	9	31	**8**	31	18 s
`Multiplicative Inverse in` $\mathbb{F}_2^4$ [LXX+23]	062493D51EC78ABF	/	/	**10***	**44**	365 s

* These circuits use an ancilla qubit.

1.2 Organization

In Section 2, notations used in this paper are defined, then we introduce the quantum circuit and SAT/SMT problem. In Section 3, we propose the new SAT-based model for the optimization of a given permutation. The application of our model is shown in Section 4. Finally, we conclude and propose future research directions in Section 5.

2 Preliminaries

2.1 Notations

Let $\mathbb{F}_2$ be the finite field with two elements 0 and 1, and $\mathbb{F}_2^k$ be the finite field with 2^k elements. An n-bit permutation f can be denoted by $\mathbb{F}_2^n \mapsto \mathbb{F}_2^n$. Consider $a, b \in \mathbb{F}_2$, we then use $\overline{a} = a \oplus 1$ to represent the inversion of a, and $a \oplus b$, $a \cdot b$ and $a|b$ denote the XOR, AND and OR operations of a and b.

2.2 Quantum Circuits

Similar to classical circuits, the quantum circuit is a model that manipulates qubits through several quantum gates, where qubits can exist in superposition states and become entangled with one another. We denote a qubit state by $|u\rangle$, where the classical bit values 0 and 1 are denoted by $|0\rangle$ and $|1\rangle$, respectively. Leveraging the inherent reversibility of quantum computation, quantum gates are implemented as simple unitary transformations, which can be succinctly represented by matrices as follows.

Usually, a quantum circuit is synthesized with the commonly used universal fault-tolerant gate set Clifford+T [Sel13, AMMR12]:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, \quad \mathrm{CNOT} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}, \quad T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}.$$

In addition, the Pauli-X gate $X = HS^2H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and the Toffoli gate

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

are employed.

To reduce the quantum resources, quantum circuits can be optimized according to various criteria. In this paper, We focus on the optimization criteria addressed in [DBSC19] and [CBC23]. The definitions are as follows.

Quantum Gate Cost(GC). The quantum gate cost of the target circuit is defined as the number of quantum gates in $\{\text{Pauli-}X, \text{CNOT}, \text{Toffoli}\}$. A Pauli-$X$ gate maps $|a\rangle$ to $|a \oplus 1\rangle$, and a CNOT gate can be regarded as a transformation that maps $|a\rangle|b\rangle$ to $|a\rangle|b \oplus a\rangle$, where only the operand b is updated. A Toffoli gate maps $|a\rangle|b\rangle|c\rangle$ to $|a\rangle|b\rangle|c \oplus a \cdot b\rangle$, which can be seen as the classical AND gate when the operand c is 0. These gates are depicted in Figure 1.

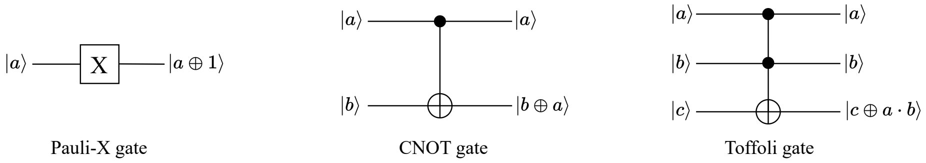


Figure 1: The description of the quantum gates.

Quantum Bit Cost(BC). The qubit cost is defined as the circuit width. It reflects the limited number of qubits available in contemporary quantum computers. We can categorize the qubits in a quantum circuit into three distinct types.

1. *Data qubit* is the input variable of a quantum circuit.
2. *Ancilla qubit* is the variable initialized to $|0\rangle$, used in the process of generating output values. Note that we shall clean up the ancilla qubits at the end of the quantum circuit.
3. *Output qubit* contains the information about output value.

Specifically, a quantum circuit in which the values of output qubits are directly stored within the data qubits is termed an "*in-place*" circuit. In this paper, we construct in-place circuits to conserve qubit resources.

Quantum Full Depth(FD). Quantum full depth cost [SM13] is defined as the largest number of elementary gates on any path from inputs to outputs in a circuit. In particular, as a non-Clifford gate, the Toffoli gate is required to be decomposed with Clifford+T gate set in different ways. We present the cost metrics in Table 3 for the NCT gate set.

Table 3: Cost metrics for the NCT gate set.

Gate	$\#FD$
Pauli-X	1
CNOT	1
Toffoli	7

To reduce the T-depth, we adopt the decomposition method of the Toffoli gate mentioned in [NC01] with $\#FD = 7$.

2.3 Cryptographic Permutation

In this paper, we mainly focus on optimizing quantum circuits for the n-bit S-boxes and matrices. Both can be regarded as permutations and play an important role in symmetric cipher. To facilitate understanding of the encoding scheme in our model, we present three representation methods of these two permutations, namely the LUT, ANF, and bit-sliced representation.

S-box. S-box is a function f: $\{0,1\}^m \mapsto \{0,1\}^n$, which ensures the property of confusion for a symmetric cipher as the only non-linear component. In the quantum circuit, we must have f be bijective since information loss is irreversible in the reversible computing paradigm. Thus, for this paper, we refer to an n-bit bijective S-box: $\{0,1\}^n \mapsto \{0,1\}^n$, which is a permutation: $[0, 2^n - 1] \mapsto [0, 2^n - 1]$.

Example 1. We take the S-box of `GIFT` [BPP+17] as an example.

The LUT representation of the S-box of `GIFT` is shown in Table 4.

Table 4: Look-up table of `GIFT` S-box.

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

The ANF representation of the S-box of `GIFT` is given by the set of equations 1, the variables x_i and y_i denote the inputs and outputs of S-box.

$$\begin{aligned}
y_0 &= 1 \oplus x_0x_1 \oplus x_0 \oplus x_1 \oplus x_2 \oplus x_3, \\
y_1 &= x_0x_1 \oplus x_0x_2 \oplus x_0 \oplus x_2 \oplus x_3, \\
y_2 &= x_0x_3 \oplus x_1x_2x_3 \oplus x_1x_3 \oplus x_1 \oplus x_2, \\
y_3 &= x_0x_2x_3 \oplus x_0 \oplus x_1x_3.
\end{aligned} \tag{1}$$

The bit-sliced representation of the S-box of `GIFT` is shown in Table 5.

Table 5: The bit-sliced representation of `GIFT` S-box.

LUT →	1	A	4	C	6	F	3	9	2	D	B	7	5	0	8	E	bit slice ↓
z_3	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1	1	5563
z_2	0	0	1	1	1	1	0	0	0	1	0	1	1	0	0	1	3C59
z_1	0	1	0	0	1	1	1	0	1	0	1	1	0	0	0	1	4EB1
z_0	1	0	0	0	0	1	1	1	0	1	1	1	1	0	0	0	8778

Linear matrix. As the linear layer of a symmetric-key primitive can be represented as a binary matrix, the implementation of a linear layer is a sequence of XOR gates. In this paper, due to the reversibility of quantum computing, we focus on the n-bit invertible matrix which also can be regarded as a permutation: $[0, 2^n - 1] \mapsto [0, 2^n - 1]$.

Example 2. We take a toy 4-bit invertible matrix M as an example.

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix},$$

The LUT representation of M is shown in Table 6.

Table 6: Look-up table of M.

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

The ANF representation of M is given by the set of equations 2, the variables x_i and y_i denote the inputs and outputs of the invertible matrix.

$$\begin{aligned} y_0 &= x_0 \oplus x_1 \oplus x_3, \\ y_1 &= x_1 \oplus x_2, \\ y_2 &= x_2 \oplus x_3, \\ y_3 &= x_0 \oplus x_3. \end{aligned} \tag{2}$$

The bit-sliced representation is shown in Table 7.

Table 7: The bit-sliced representation of a given 4-bit invertible matrix.

LUT →	0	9	3	A	6	F	5	C	D	4	E	7	B	2	8	1	bit slice ↓
z_3	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	55AA
z_2	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0FF0
z_1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	3C3C
z_0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	6699

2.4 SAT/SMT Problem

In recent years, the application of automated search tools in cryptography has become more and more extensive. The SAT problem belongs to the deterministic problem, and it is also the first problem to be proved to be NP-complete. To solve it, boolean expressions are usually encoded in Conjunctive Normal Form (CNF) as the inputs of a SAT solver. Extending SAT to satisfy the modulus theory (satisfiability modulo theories, abbreviated as SMT) can enrich the form of CNF expression, which includes linear constraints, arrays, and so on. Compared with the method based on SAT problems, the method based on SMT is more flexible and applicable to a wider range, which is very suitable for application in the field of cryptography.

2.4.1 A Constraint Solver: STP

This paper mainly uses the solver of the SMT problem, STP[1], to automatically solve our new proposed quantum circuit decision problems. When STP successfully finds a circuit for some value k but outputs `UNSAT` for $k-1$, it is proven that k is the minimum value. CVC formats are one of the commonly used file-based input languages in STP. We list some CVC language references and two examples as follows.

[1] http://http://stp.github.io/

Table 8: Usage of the STP solver.

Name	Symbol	Example
Concatenation	@	$t_1 @ t_2 @ \ldots @ t_n$
Extraction	$i : j$	$x[31 : 26]$
Bitwise XOR	BVXOR	BVXOR (t_1, t_2)
Bitvector AND	BVPLUS	BVPLUS $(n, t_1, t_2, \ldots, t_n)$
Less Than Or Equal To	BVLE	BVLE (t_1, t_2)
Greater Than or Equal To	BVGE	BVGE (t_1, t_2)
Not Equal to	$\backslash =$	$t_1 \backslash = t_2$

Example 3. We list statements based on CVC language to describe `GIFT` S-box.

```
ARRAY BITVECTOR(4) OF BITVECTOR(4);
```

//The size of the S-box is 2^4 and each element is a 4-bit boolean variable.

```
ASSERT( S[0bin0000] = 0bin0001 );
ASSERT( S[0bin0001] = 0bin1010 );
...
```

//Assignment: $S[0] = 1; S[1] = 10; \ldots$

Example 4. The description of the condition "if $a = b$ then $c = 1$" based on CVC language.

```
a, b, c:  BITVECTOR(1);
ASSERT( ( a = b ) => ( c = 1 ) );
```

2.4.2 Stoffelen's Model Based on SAT Solvers

To find more compact implementations of small S-box circuits, Stoffelen proposed a search model [Sto16] based on SAT solvers to achieve multiple optimization criteria, including nonlinear gate count, gate count, and circuit depth, etc. The main idea is to convert the problem of solving the target circuit into a satisfiability problem and use the off-the-shelf SAT solvers to solve it. For example, when optimizing the gate number of an n-bit S-box, Stoffelen proposed a binary model to solve the following decision problem:

Is there a circuit implementing $\mathbb{F}_2^n \to \mathbb{F}_2^n$ and that uses at most K logic operations?

This model encodes each gate as an ANF equation and can judge the existence of solutions when given the number of gates. To get the smallest number of gates, it should exhaust K until it finds the smallest one that there exists an implementation of the target S-box.

3 New SAT Model for Quantum Circuit Decision Problem

In this section, we discuss the optimized implementations of quantum circuits for a given permutation f. To begin with, we propose the quantum circuit decision problem for the optimization goals. Within this problem, these criteria mentioned in Section 2.2 for quantum circuits are taken into account. Then we analyze the limitations of the existing optimization techniques for circuits in effectively solving this problem. Hence, we introduce a novel quantum circuit model based on SAT solvers, followed by a comprehensive exposition of encoding schemes.

3.1 Quantum Circuit Decision Problem

In the research of quantum circuits, many related works have proposed various optimization methods for the criteria mentioned in Section 2.2. For example, Huang et al. [HS22] in ASIACRYPT 2022 proposed a model that aims to search for an in-place quantum circuit with the minimum number of CNOT gates of a linear permutation, which can be regarded as a decision problem solved by SAT solvers. To implement a more compact quantum circuit covering the three optimization goals of quantum gate cost(GC), quantum bit cost(BC), and quantum full depth cost(FD), we present the quantum circuit decision problem (cf. Definition 1).

Definition 1 (Quantum circuit decision problem)**.** Given an n-bit permutation $f\colon [0, 2^n - 1] \mapsto [0, 2^n - 1]$, the quantum circuit decision problem requires us to determine whether there exists a quantum circuit implementation of f that uses no more than M ancilla qubits, K quantum gates with the full depth at most D.

3.2 Limitations in the Previous Methods

Currently, there exist two types of potential methods for solving the quantum circuit decision problem.

- **Local optimization method.** This approach focuses on the comprehensive consideration of multiple heuristic algorithms [CBC23, DBSC19] to directly optimize different criteria of quantum circuits.
- **Exhaustive search method.** This approach can achieve an exhaustive search [Sto16, LWH+21] for small-scale classical circuits based on SAT solvers, obtaining the optimal circuit under the target goal.

However, these methods cannot directly solve the quantum circuit decision problem. We clarify the limitations inherent in these methods, thereby presenting our proposed solution, which entails a novel SAT model.

3.2.1 Analysis of Chun et al. 's Quantum Heuristic Algorithm

Chun et al. proposed a tool, `DORCIS`, that found depth-optimized quantum circuit implementations for arbitrary 3- and 4-bit S-boxes in 2023. However, `DORCIS` has two limitations in solving the quantum circuit decision problem.

Optimality of the quantum circuit cannot be proven. `DORCIS` is an algorithm with highly effective optimization results, constructing circuits of very low depth for multiple permutations. However, it relies on some heuristics and is infeasible to prove that their results are optimal. If we inquire `DORCIS` whether this circuit is optimal or if there exist circuits with lower depth, we won't obtain an answer.

The issue of unresolved ancilla qubits remains unaddressed. `DORCIS` does not provide ancilla qubits. It means that if we try to generate a circuit with an ancilla qubit, the tool will not return a result. Note that not all S-boxes can be implemented with in-place circuits without ancilla qubits. In [LXX+23], Lin et al. studied the permutation and proposed a definition of the odd permutation.

Definition 2 (Odd permutation)**.** A permutation is called odd if it can be written as the product of an odd number of transpositions.

No quantum circuit can be given for S-boxes with odd permutations without ancilla qubits. In this case, `DORCIS` will be unable to provide the optimal circuit or even a solution.

3.2.2 Analysis of Stoffelen and Lu et al. 's SAT Model

In 2016, Stoffelen [Sto16] presented a SAT-based model for optimizing S-box circuits. Subsequently, Lu et al. [LWH+21] improved Stoffelen's model by adding the criteria for optimizing area to find the circuit implementation of the minimum area of the S-box.

In classical computing, the SAT-based model effectively addresses the optimization challenge of circuit hardware area and latency. However, there exist some problems if the model is employed directly for the optimization of quantum circuits due to the properties of quantum computing. We briefly outline the issues encountered by the model.

Reversible operations can not be guaranteed. We summarized the SAT model used in [LWH+21]. It is shown in Figure 2. The variables x_i and y_i denote the inputs and outputs of a circuit.

The set of equations (3) is encoding the decision of choosing a type of gate.

$$\begin{aligned} t_0 &= b_0 \cdot q_0 \cdot q_1 + b_1 \cdot q_0 + b_1 \cdot q_1 + b_2 \\ t_1 &= b_2 \cdot q_2 \cdot q_3 + b_3 \cdot q_2 + b_3 \cdot q_3 + b_4 \end{aligned} \tag{3}$$

The variable q_i denotes the input of a gate, and b_i determines what kind of gate the t_i will represent. When the value of the pattern $b_{3i}||b_{3i+1}||b_{3i+2}$ is different, t_i represents different kinds of gate. The above method can encode different classical gates, such as when $b_0||b_1||b_2 = 010$, it represents an XOR gate, which can be denoted by $t_0 = q_0 \oplus q_1$. In the quantum circuit, to achieve the operation of XOR, it requires three qubits and needs to be split into two operations $t_0 = t_0 \oplus q_0$ and $t_0 = t_0 \oplus q_1$, which contradicts the principle of optimal gate count.

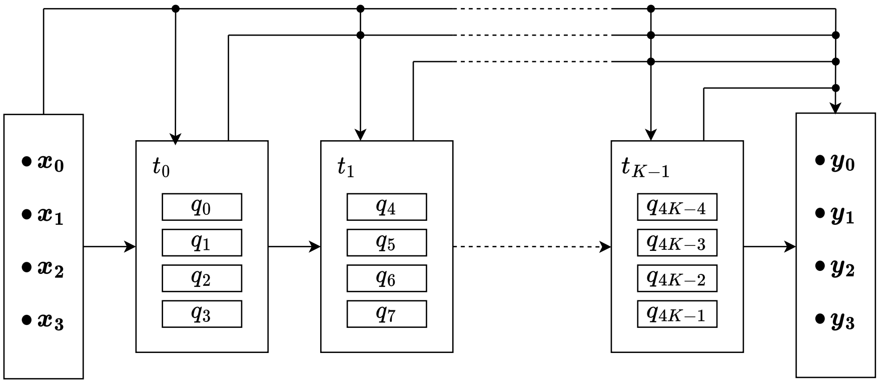


Figure 2: The SAT model framework in [LWH+21].

No fan-out property in the quantum setting. In Lu et al. 's SAT-based model, the fan-out operation is allowed because the model considers that the output of the previous gates can be used as the input of any subsequent gates, described by Figure 2. But quantum gates do not allow fan-out, which means that the output of the previous gate can only be used as the input of the next gate. Besides, if a variable x_i is used twice in the same depth, the quantum depth has to be increased, different from classical circuits.

3.3 Main Idea of Our Model

As discussed above, in light of the issues, there currently exists no model capable of solving the quantum circuit decision problem. Since the SAT problem can be solved by existing solvers, we employ an enhanced SAT model to solve it. For the permutation f, if our model successfully finds a circuit for the metric k and the solver proves that there is no solution for $k-1$, it is proven that k is the minimum value. Thus, our model is proper for the decision problem.

From a higher-level perspective, the new SAT model needs to answer the following three questions.

- Question 1: How to incorporate the parameter of qubit number(BC) into the model. This question can enable the model to find quantum circuits that implement odd permutations, which need ancilla qubits. Besides, by introducing extra variables into the model beforehand, there is a potential to discover quantum circuits with lower full depth.

- Question 2: How to represent reversible gates in the enhanced model. For the Toffoli gate, the model only allows the operation like $c = c \oplus a \cdot b$. For the CNOT gate, the model only allows the operation like $b = b \oplus a$. Given these constraints, the circuits we obtain always adhere to the quantum setting.

- Question 3: How to constrain quantum depth without assuming fan-out. This requires us to constrain a variable to appear only once at the same depth, which is attributed to the novel model scheme that we have introduced.

As an outline, we introduce our new gate-level structure search model (cf. Figure 3). The variables used in the encoding scheme are shown in Table 9. We first encode the problem in logical formulas in Conjunctive Normal Form (CNF). Then, the constraints are used to describe quantum circuits in SAT problems.

Table 9: Variables used in the encoding process.

Notations	Definitions
$x_i[j]$	The i-th data wire in front of the j-th gate
$anc_i[j]$	The i-th ancilla wire in front of the j-th gate
$depth_i[j]$	The depth of the i-th wire in front of the j-th gate
y_i	Permutation outputs
q_i	Gate inputs
t_i	Gate outputs
a_i	Wiring between gates
b_i	Wiring ‘inside’ gates

The structure contains the input and output of the circuit, consisting of gates as internal subcircuits. The reason for using this serial framework is that quantum gates cannot fan out, the output of each gate can only be used as the input of the next gate. In Figure 3, the circuit of the 0-th gate can be denoted by $(x_0[0], x_1[0], x_2[0], x_3[0] \mapsto x_0[1], x_1[1], x_2[1], x_3[1])$. If an ancilla qubit is required, we just set it as the input variable.

3.4 Encoding Scheme

In this section, we use the SAT solver to solve this decision problem, which requires the problem to be encoded. Hence, we present the encoding process of the model to a set of equations in ANF, and finally use STP to return the target circuit.

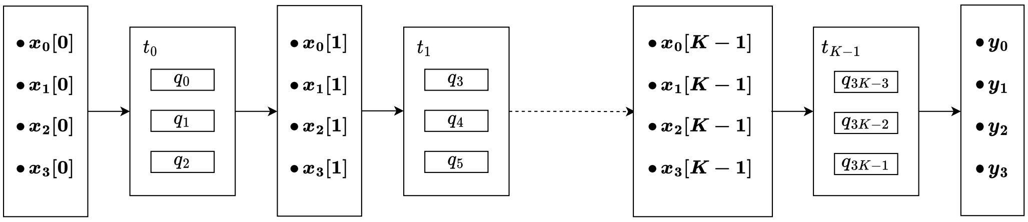

$\bullet\, \boldsymbol{x_0[0]}$
$\bullet\, \boldsymbol{x_1[0]}$
$\bullet\, \boldsymbol{x_2[0]}$
$\bullet\, \boldsymbol{x_3[0]}$
t_0
q_0
q_1
q_2
$\bullet\, \boldsymbol{x_0[1]}$
$\bullet\, \boldsymbol{x_1[1]}$
$\bullet\, \boldsymbol{x_2[1]}$
$\bullet\, \boldsymbol{x_3[1]}$
t_1
q_3
q_4
q_5
$\bullet\, \boldsymbol{x_0[K-1]}$
$\bullet\, \boldsymbol{x_1[K-1]}$
$\bullet\, \boldsymbol{x_2[K-1]}$
$\bullet\, \boldsymbol{x_3[K-1]}$
t_{K-1}
q_{3K-3}
q_{3K-2}
q_{3K-1}
$\bullet\, \boldsymbol{y_0}$
$\bullet\, \boldsymbol{y_1}$
$\bullet\, \boldsymbol{y_2}$
$\bullet\, \boldsymbol{y_3}$

Figure 3: A new gate-level structure search model for a 4-bit S-box.

To better understand the encoding scheme, we give a model of the quantum decision problem whether there is a quantum circuit that implements GIFT S-box with 1 ancilla qubit, and 9 quantum gates with $\#FD$ at most 31. The bit-sliced representation of GIFT S-box has been given in Table 5.

3.4.1 Encode Input, Ancilla Qubits, and Output

For an $\mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ permutation f, due to the bit-sliced technique, the input variable $x_i[0]$ $(0 \leq i \leq n-1)$ is a 2^n-dimensional array over $\mathbb{F}_2^n$, and the output variable y_i is a 2^n-dimensional variable over $\mathbb{F}_2^n$. Note that distinct from the conventional encoding schemes, we innovatively employ arrays to represent input variables, avoiding the generation of intermediary variables and facilitating the construction of in-place circuits. Note that the size of arrays can be adjusted according to the gate count.

For GIFT S-box, assume that the number of gates is 0bin001001 (9 gates) and the number of ancilla qubits is 1. The input X_i, the ancilla qubit Anc_k, and the output Y_j can be encoded as follows.

```
X_0, X_1, X_2, X_3:  ARRAY BITVECTOR(6) OF BITVECTOR(16);
Y_0, Y_1, Y_2, Y_3:  BITVECTOR(16);
```

Next, we encode the look-up table of the permutation.

```
ASSERT( X_0[0bin000000] = 0bin0000000011111111 );
ASSERT( X_1[0bin000000] = 0bin0000111100001111 );
ASSERT( X_2[0bin000000] = 0bin0011001100110011 );
ASSERT( X_3[0bin000000] = 0bin0101010101010101 );
ASSERT( Y_0 = 0bin0101010101100011 );
ASSERT( Y_1 = 0bin0011110001011001 );
ASSERT( Y_2 = 0bin0100111010110001 );
ASSERT( Y_3 = 0bin1000011101111000 );
```

The ancilla qubit is equivalent to the wire with an input of 0, which is used to store the intermediate value in the circuit, so the encoding method is consistent with the input variable, denoted by the array with a determined value of 0. The ancilla qubit in the S-box of GIFT is given by the following statement.

```
Anc_0:  ARRAY BITVECTOR(6) OF BITVECTOR(16);
ASSERT( Anc_0[0bin000000] = 0bin0000000000000000 );
```

Considering that we construct in-place circuits, it is necessary to store the output value within the data qubit. Consequently, we establish a direct correspondence between the input and output of the S-box at the end of this model. That is to say, the output variable Y_j corresponds to the output wire X_i according to Figure 3. For the decision problem of GIFT S-box, upon traversing a sequence of nine gates from input to output, the data qubits can be succinctly represented by $x_0[9], x_1[9], x_2[9], x_3[9]$.

```
ASSERT((Y_0=X_0[0bin001001])OR(Y_0=X_1[0bin001001])OR(Y_0=X_2
[0bin001001])OR(Y_0=X_3[0bin001001]));
ASSERT((Y_1=X_0[0bin001001])OR(Y_0=X_1[0bin001001])OR(Y_0=X_2
[0bin001001])OR(Y_0=X_3[0bin001001]));
ASSERT((Y_2=X_0[0bin001001])OR(Y_0=X_1[0bin001001])OR(Y_0=X_2
[0bin001001])OR(Y_0=X_3[0bin001001]));
ASSERT((Y_3=X_0[0bin001001])OR(Y_0=X_1[0bin001001])OR(Y_0=X_2
[0bin001001])OR(Y_0=X_3[0bin001001]));
```

3.4.2 Encode K Quantum Gates

Encode the decision of choosing inputs of a gate. This paper uses the NCT gate set, including the single-input NOT gate, the two-input CNOT gate, and the three-input Toffoli gate, so at least three-input gates need to be encoded. For $\forall i \in \{0, \dots, K-1\}$, $q_{3i}, q_{3i+1}, q_{3i+2}$ are the three inputs of the i-th gate t_i. We have simplified the statements in CVC format and expressed them by the set of equations in ANF for convenience.

$$\begin{aligned}
q_0 &= a_0 \cdot x_0[0] + a_1 \cdot x_1[0] + a_2 \cdot x_2[0] + a_3 \cdot x_3[0] + a_4 \cdot anc_0[0] \\
q_1 &= a_5 \cdot x_0[0] + a_6 \cdot x_1[0] + a_7 \cdot x_2[0] + a_8 \cdot x_3[0] + a_9 \cdot anc_0[0] \\
q_2 &= a_{10} \cdot x_0[0] + a_{11} \cdot x_1[0] + a_{12} \cdot x_2[0] + a_{13} \cdot x_3[0] + a_{14} \cdot anc_0[0]
\end{aligned} \tag{4}$$

We then can encode the input of the second gate in the same form as follows. This enables us to describe the inputs of all gates.

$$\begin{aligned}
q_3 &= a_{15} \cdot x_0[1] + a_{16} \cdot x_1[1] + a_{17} \cdot x_0[1] + a_{18} \cdot x_1[1] + a_{19} \cdot anc_0[1] \\
q_4 &= a_{20} \cdot x_0[1] + a_{21} \cdot x_1[1] + a_{22} \cdot x_0[1] + a_{23} \cdot x_1[1] + a_{24} \cdot anc_0[1] \\
q_5 &= a_{25} \cdot x_0[1] + a_{26} \cdot x_1[1] + a_{27} \cdot x_0[1] + a_{28} \cdot x_1[1] + a_{29} \cdot anc_0[1]
\end{aligned} \tag{5}$$

Encode the decision of choosing a type of gate. The variable b_i determines what kind of gate the circuit will choose, the encoding of different types of gates is shown in Table 10. The choices of the first two gates are presented as follows.

$$\begin{aligned}
t_0 &= q_0 + q_1 + b_0 \cdot q_1 + b_0 \cdot q_1 \cdot q_2 + b_1 \cdot q_1 + b_1 \\
t_1 &= q_3 + q_4 + b_2 \cdot q_4 + b_2 \cdot q_4 \cdot q_5 + b_3 \cdot q_4 + b_3
\end{aligned} \tag{6}$$

Table 10: Encoding of different types of gates.

$b_{2i} \| b_{2i+1}$		Operations	Gate function
0	0	CNOT	$q_{3i} \oplus q_{3i+1}$
0	1	NOT	$q_{3i} \oplus 1$
1	0	Toffoli	$q_{3i} \oplus q_{3i+1} \cdot q_{3i+2}$

More constraints in the quantum setting. In order to ensure that each gate conforms to be reversible, we use the three-step operation in Figure 4 to complete the self-renewal of the input wire.

- Step 1: The model chooses one, two, or three inputs of a quantum gate. Then, we add two constraints. Firstly, for the first gate in Equation (4), $\forall i, j \in \{0, \dots, 4\}, i \neq j, \forall m, n \in \{5, \dots, 9\}, m \neq n, \forall u, v \in \{10, \dots, 14\}, u \neq v$, we add the constraint

  $$0 = a_i \cdot a_j, \ 0 = a_m \cdot a_n, \ 0 = a_u \cdot a_v \tag{7}$$

  to encode that only one of the variables a_t in an equation can be equal to 1.

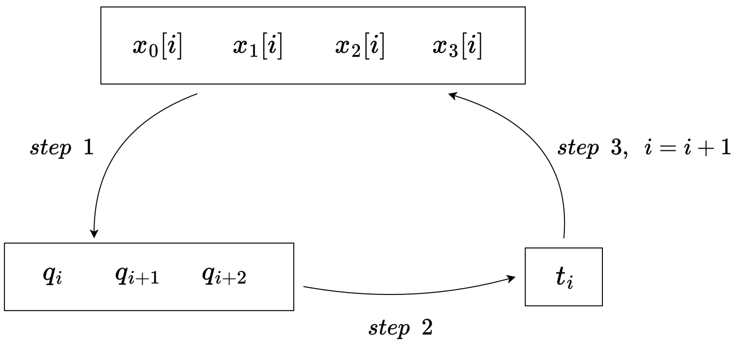


Figure 4: The updating process of the i-th gate for a 4-bit S-box.

Secondly, we add the following constraints on the gate inputs to ensure that each input of a gate is different.

$$\begin{array}{c}(a_0 \cdot a_5 = 0) \mathbin{\&} (a_0 \cdot a_{10} = 0) \mathbin{\&} (a_5 \cdot a_{10} = 0)\\ (a_1 \cdot a_6 = 0) \mathbin{\&} (a_1 \cdot a_{11} = 0) \mathbin{\&} (a_6 \cdot a_{11} = 0)\\ (a_2 \cdot a_7 = 0) \mathbin{\&} (a_2 \cdot a_{12} = 0) \mathbin{\&} (a_7 \cdot a_{12} = 0)\\ (a_3 \cdot a_8 = 0) \mathbin{\&} (a_3 \cdot a_{13} = 0) \mathbin{\&} (a_8 \cdot a_{13} = 0)\\ (a_4 \cdot a_9 = 0) \mathbin{\&} (a_4 \cdot a_{14} = 0) \mathbin{\&} (a_9 \cdot a_{14} = 0)\end{array} \tag{8}$$

- Step 2: Our model limits that the output of a gate must be one of the input wires. In Equation (6), t_0 must be one of q_0, q_1, and q_2. We list four possible values of q_{3i}: $x_0[i], x_1[i], x_2[i], x_3[i]$ for the i-th gate, and constrain the value to be equal to t_i. For example, if $q_{3i} = x_0[i]$, then the solver will constraint $x_0[i + 1] = t_i$.
- Step 3: For the wires that have not been updated, we simply constrain the input to be equal to the output for the next gate operation. For example, $x_1[i + 1] = x_1[i]$, $x_2[i + 1] = x_2[i]$, $x_3[i + 1] = x_3[i]$. The following simplified CVC language can be used to control this conditional statement.

$$\begin{array}{r}(q_{3i} = x_0[i]) => ((x_0[i+1] = t_i) \mathbin{\&} (x_1[i+1] = x_1[i]) \mathbin{\&}\\ (x_2[i+1] = x_2[i]) \mathbin{\&} (x_3[i+1] = x_3[i]))\end{array} \tag{9}$$

3.4.3 Encode D Quantum Depth

The search model in Figure 3 is layered according to gates. We can consider it a serial search structure, so it is better to combine circuit depth with gates. Different from the classical depth, due to the non-replication of qubits, gates acting on the same qubit will increase the quantum depth. Besides, we have the following observations.

Observation 1. *The depth of a quantum circuit is determined by the wire (or path) with the highest depth.*

Observation 2. *After applying the same quantum gate, the output wires have the same depth. If the input wires have different depths, the depth is determined by the higher one.*

Based on the above observations, in order to encode the circuit depth to D, we can control the depth of each wire at the same time and use the SAT solver to search the circuit. If there is a circuit implementation when the depth is less than $D + 1$, and no solution when the depth is less than D, it means that the minimum circuit depth is D. The encoding and constraint process are as follows.

- Firstly, for an n-bit S-box, similar to the input variables, we define n array variables for the input wires to denote depth: $depth_0[]$, $depth_1[]$, $\dots$, $depth_{n-1}[]$. Take the PRINCE S-box as an example, we have the following constraints:

```
depth_0,depth_1,depth_2,depth_3:ARRAY BITVECTOR(6) OF BITVECTOR(16);
```

- Then, it is necessary to restrict the depth relationship between the input and output circuits of a quantum gate. We need to enumerate all possibilities of quantum gates by constraint encoding, and we take a subcircuit composed of CNOT gates as an example in Figure 5. Use the CVC language to constrain the following if-condition statement:

$$\begin{aligned}(depth_0[j] > depth_2[j]) => &((depth_0[j+1] = depth_0[j+1]+1)\ \& \\ &(depth_2[j+1] = depth_0[j+1]+1)\ \& \\ &(depth_1[j+1] = depth_1[j])\ \& \\ &(depth_3[j+1] = depth_3[j]))\end{aligned} \tag{10}$$

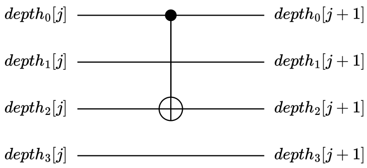


Figure 5: The subcircuit of the j-th gate.

- Finally, our model constrains the circuit depth of each wire to be no more than D. For GIFT S-box, the encoding is as follows.

```
ASSERT( BVLT(depth_0[0bin001001], 0bin0000000000011111) );
ASSERT( BVLT(depth_1[0bin001001], 0bin0000000000011111) );
ASSERT( BVLT(depth_2[0bin001001], 0bin0000000000011111) );
ASSERT( BVLT(depth_3[0bin001001], 0bin0000000000011111) );
```

4 Applications

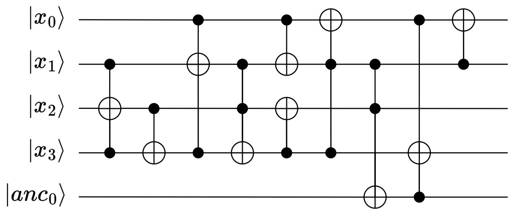
In this section, we apply the new model to some permutations. Firstly, for 3-bit and 4-bit S-boxes, we first prove that the gate count from LIGHTER-R is optimal due to the exhaustive search by using the SAT solver. Then, by combining different criteria in the model, we find more compact quantum circuit implementations of S-boxes such as RECTANGLE and GIFT than using LIGHTER-R and DORCIS. Furthermore, we can streamline the model to optimize the quantum circuit for reversible linear permutations, which offers enhanced efficiency compared to the optimization of S-boxes.

4.1 Optimizing the Quantum Implementation of S-box

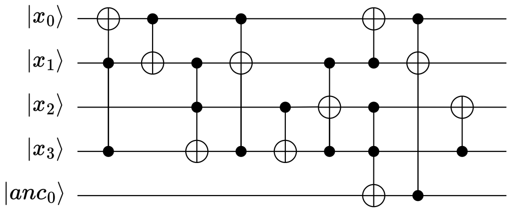
Using our model, we can see an improvement in quantum gate cost(GC), quantum bit cost(BC), and quantum full depth(FC) on the currently known best quantum circuits for 3-bit and 4-bit S-boxes. We have undertaken efforts in three aspects.

- Firstly, we apply our model to validate the circuits produced by current heuristic methodologies, addressing the critical inquiry of whether these methods achieve optimality in quantum circuit optimization.
- Secondly, we answer the question of optimizing quantum circuits for odd permutations that require ancilla qubits by introducing additional variables.
- Ultimately, we refine the quantum circuits for S-boxes by leveraging Algorithm 1 and Algorithm 2, enhancing their performance through targeted optimization strategies.

Optimize for GC. Setting M to 0 in the model and D to a larger value (the depth limit can be ignored temporarily), the quantum circuit decision problem can be transformed into a sub-problem: whether there exists a quantum circuit implementing using no more than K quantum gates. We apply the model to solve this problem for 3-bit and 4-bit S-boxes and obtain the same gate count as LIGHTER-R. When we reduce the constraint of the number of gates M, the solver returns "NO SOLUTION", which can prove that the quantum s-box given by the meet-in-the-middle algorithm LIGHTER-R regarding quantum gate complexity is optimal.



(a) Method in [LXX+23] (#GC 10, #FD 45)



(b) **Our model** (#GC 10, #FD 44)

Figure 6: Different implementations of 4-bit quantum S-box in AES.

Add ancilla qubits to implement quantum S-boxes. By controlling the number of ancilla qubits, our model can provide the quantum circuit of some specific S-boxes with odd permutations. It is worth noting that in the construction of quantum circuits for AES, Lin et al. [LXX+23] implemented a low-qubit S-box circuit based on Tower Field, where the S-box is decomposed into three layers. The middle layer can be viewed as a 4-bit odd permutation (062493D51EC78ABF). We apply LIGHTER-R to this permutation. However, no circuit can be returned as they require more qubits.

By adding more constraints and using two strategies, Lin et al. obtained the quantum circuit for this permutation using 10 gates with a full depth of 45. Directly utilizing our model, we obtained the optimized circuit with depth 44, which is more compact. The comparison is illustrated in Figure 6. This novel quantum circuit for the permutation can be seamlessly integrated into the circuits detailed in [LXX+23], subsequently improving the quantum circuit of AES.

Combine criteria to solve the quantum decision problem. Three parameters M, K, and D are considered in the model. To better weigh the corresponding optimization goals, we decompose the optimization process into the following two steps.

- Firstly, solve the sub-problem: whether a circuit can implement a permutation using at most K logic operations and M ancilla qubits with the circuit depth less than D, which is a given larger value (DC can be ignored).
- Secondly, after finding the smallest gate count, further optimize for the full depth of circuit.

We propose Algorithm 1 to solve the first sub-problem for a given n-bit permutation f. By setting the depth D larger, Algorithm 1 finds the circuit implementation with the smallest number of gates and qubits.

Algorithm 1 Solve the first sub-problem

```
Input: Number of gates $K$, ancilla qubits $M$, Depth $D$
Output: If the sub-problem has a solution, the solver returns "1" and the implementation
  of this S-box or other case returns "0"
  for $m$ from 0 to $M-1$ do
      for $k$ from 0 to $K-1$ do
          Input to the model for encoding, using STP to solve it
          if STP returns the implementations of $f$ then
              Finish and return $m$ and $k$
          end if
      end for
  end for
```

We then further optimize the quantum depth. The Algorithm 1 can be used to get the circuit C for f and the returned results K_{low}, M_{low}, denote the circuit depth of C by D_{up} as the upper bound, and then determine the lower bound of the circuit depth. We have the following observation.

Observation 3. *The lower bound on the depth of an n-bit quantum circuit with k gates composed of NCT gate sets is $\lceil \frac{k}{n} \rceil$.*

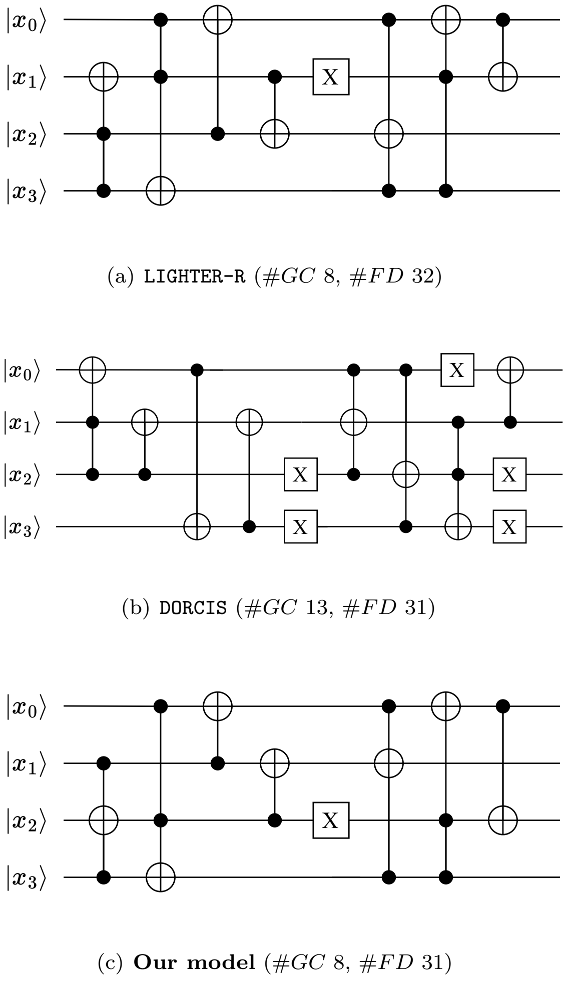
Obviously, the highest degree of parallelism can only be achieved when all gates are NOT gates, resulting in the lowest circuit depth. So the lower depth of S is $\lceil \frac{K_{low}}{n+M_{low}} \rceil$. Finally, use Algorithm 2 to get the depth-optimal circuit.

Running Algorithm 1 and Algorithm 2, we show the improvement of our new model. Table 1 and Table 2 show the comparison results with LIGHTER-R and DORCIS respectively. Notably, in Table 2, for optimizing GIFT S-box, DORCIS uses 13 gates with depth 31. Our model can find a new quantum circuit with the same depth using only 8 gates. Furthermore, if we allow an ancilla qubit in the circuit, our model can reduce the depth to 30. The three different quantum circuits (proposed by LIGHTER-R, DORCIS and our model) of GIFT S-box are shown in Figure 7.

It is worth noting that if the number of ancilla qubits is increased, it is likely to continue to reduce the circuit depth, but due to the difficulty of solving large scale SAT models in practice, we only give examples of a single ancilla qubit in this paper.

Algorithm 2 Solve the quantum circuit decision problem of the circuit C

```
Input: The gate number $K_{low}$, the ancilla qubits number $M_{low}$, the circuit depth $D_{up}$
Output: The circuit implementation of optimal depth
  for $d$ from $\lceil \frac{k_{low}}{n+M_{low}} \rceil$ to $D_{up}$ do
      Call the Algorithm 1 with the input $(k_{low}, m_{low}, d)$
      if Algorithm 1 returns “NO SOLUTION" then
          $d = d + 1$
      end if
      Finish and return $d$
  end for
```



(a) LIGHTER-R (#GC 8, #FD 32)

(b) DORCIS (#GC 13, #FD 31)

(c) **Our model** (#GC 8, #FD 31)

Figure 7: GIFT S-box implementation with different methods.

4.2 Optimizing the Implementation of Quantum Linear Matrix.

In Section 3.4, we provide a comprehensive description of the model's encoding process for S-boxes, which can be extended to the linear matrix with the constraint of the coding type of the gate. In Table 10, we provide the encoding of each gate. If we set $b_i = 0$ for each b_i, then the SAT-based model can solve the decision problem for the linear permutation.

We also use the model to optimize the `AES` S-box proposed by Lin et al. [LXX+23]. In the first layer of the S-box, it contains two 8-bit linear matrices. We apply our model to one of the matrix A

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

to show the optimization results. It can be seen that the quantum gate number and depth have been greatly improved after using our model in Figure 8. In [LXX+23], their circuit requires 16 CNOT gates with $\#FD$ 9. However, our model just needs 13 CNOT gates with $\#FD$ 6. The different quantum circuits for A are presented in Appendix A.

5 Conclusion

In this paper, we introduced the quantum circuit decision problem for optimizing the quantum circuits of permutations. In order to solve it, we investigate existing tools while highlighting the challenges encountered. Subsequently, we innovatively proposed a new SAT model based on STP to solve this problem and algorithms to employ the model to solve many sub-problems. By applying this model, we can prove the optimality for certain original quantum circuits constructed by `LIGHTER-R` and further optimize the quantum circuits of some permutations from the demonstrated examples.

This is a general model that can implement quantum circuits of S-box and linear matrix. However, because the search space is far too large, the model is challenging for circuit realization of permutation of large states. We look forward to follow-up work to solve such quantum circuit decision problems.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This research is supported by the National Key Research and Development Program of China (Grant No.2018YFA0704702), the National Natural Science Foundation of China (Grant No.62032014, U2336207), Department of Science & Technology of Shandong Province (No.SYS202201), National Key R&D Program of China (2023YFA1009500), Quan Cheng Laboratory (Grant No.QCLZD202301, QCLZD202306).

References

[AMMR12] Matthew Amy, Dmitrii L. Maslov, Michele Mosca, and Martin Rötteler. 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*, 32:818–830, 2012. URL: https://api.semanticscholar.org/CorpusID:6879679.

[ASAM18] Mishal Almazrooie, Azman Samsudin, Rosni Abdullah, and Kussay N. Mutter. Quantum reversible circuit of AES-128. *Quantum Inf. Process.*, 17(5):112, 2018. doi:10.1007/s11128-018-1864-3.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015. doi:10.1007/978-3-662-48800-3_17.

[BCDM21] Tim Beyne, Yu Long Chen, Christoph Dobraunig, and Bart Mennink. Multi-user security of the elephant v2 authenticated encryption mode. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers*, volume 13203 of *Lecture Notes in Computer Science*, pages 155–178. Springer, 2021. doi:10.1007/978-3-030-99277-4_8.

[BDK+21] Anubhab Baksi, Vishnu Asutosh Dasu, Banashri Karmakar, Anupam Chattopadhyay, and Takanori Isobe. Three input exclusive-or gate support for boyar-peralta's algorithm. In Avishek Adhikari, Ralf Küsters, and Bart Preneel, editors, *Progress in Cryptology - INDOCRYPT 2021 - 22nd International Conference on Cryptology in India, Jaipur, India, December 12-15, 2021, Proceedings*, volume 13143 of *Lecture Notes in Computer Science*, pages 141–158. Springer, 2021. URL: https://doi.org/10.1007/978-3-030-92518-5_7.

[BGLS19] Zhenzhen Bao, Jian Guo, San Ling, and Yu Sasaki. PEIGEN - a platform for evaluation, implementation, and generation of s-boxes. *IACR Trans. Symmetric Cryptol.*, 2019(1):330–394, 2019. URL: https://doi.org/10.13154/tosc.v2019.i1.330-394.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016. doi:10.1007/978-3-662-53008-5_5.

[BMP08] Joan Boyar, Philip Matthews, and René Peralta. On the shortest linear straight-line program for computing linear forms. In Edward Ochmanski and Jerzy Tyszkiewicz, editors, *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, 2008, Proceedings*, volume 5162 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-85238-4_13.

[BMP13] Joan Boyar, Philip Matthews, and René Peralta. Logic minimization techniques with applications to cryptology. *J. Cryptol.*, 26(2):280–312, 2013. URL: https://doi.org/10.1007/s00145-012-9124-7.

[BPP$^+$17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017. doi:10.1007/978-3-319-66787-4_16.

[CBC23] Matthew Chun, Anubhab Baksi, and Anupam Chattopadhyay. DORCIS: depth optimized quantum implementation of substitution boxes. *IACR Cryptol. ePrint Arch.*, page 286, 2023. URL: https://eprint.iacr.org/2023/286.

[CJL$^+$16] Lidong Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography, 2016-04-28 2016. doi:10.6028/NIST.IR.8105.

[DBSC19] Vishnu Asutosh Dasu, Anubhab Baksi, Sumanta Sarkar, and Anupam Chattopadhyay. LIGHTER-R: optimized reversible circuit implementation for sboxes. In *32nd IEEE International System-on-Chip Conference, SOCC 2019, Singapore, September 3-6, 2019*, pages 260–265. IEEE, 2019. doi:10.1109/SOCC46988.2019.1570548320.

[GLRS15] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. Applying grover's algorithm to aes: quantum resource estimates, 2015. arXiv:1512.04965.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996. doi:10.1145/237814.237866.

[HS22] Zhenyu Huang and Siwei Sun. Synthesizing quantum circuits of AES with lower t-depth and less qubits. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 614–644. Springer, 2022. doi:10.1007/978-3-031-22969-5_21.

[JBS$^+$22] Kyungbae Jang, Anubhab Baksi, Gyeongju Song, Hyunji Kim, Hwajeong Seo, and Anupam Chattopadhyay. Quantum analysis of AES. *IACR Cryptol. ePrint Arch.*, page 683, 2022.

[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 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 280–310. Springer, 2020. doi:10.1007/978-3-030-45724-2_10.

[JPST17] Jérémy Jean, Thomas Peyrin, Siang Meng Sim, and Jade Tourteaux. Optimizing implementations of lightweight building blocks. *IACR Trans. Symmetric Cryptol.*, 2017:130–168, 2017. URL: https://api.semanticscholar.org/CorpusID:24785069.

[KLSW17] Thorsten Kranz, Gregor Leander, Ko Stoffelen, and Friedrich Wiemer. Shorter linear straight-line programs for MDS matrices. *IACR Trans. Symmetric Cryptol.*, 2017(4):188–211, 2017. URL: https://doi.org/10.13154/tosc.v2017.i4.188-211.

[LPS20] Brandon Langenberg, Hai Pham, and Rainer Steinwandt. Reducing the cost of implementing the advanced encryption standard as a quantum circuit. *IEEE Transactions on Quantum Engineering*, 1:1–12, 2020. doi:10.1109/TQE.2020.2965697.

[LPZW23] Qun Liu, Bart Preneel, Zheng Zhao, and Meiqin Wang. Improved quantum circuits for AES: reducing the depth and the number of qubits. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 67–98. Springer, 2023. doi:10.1007/978-981-99-8727-6_3.

[LWF$^+$22] Qun Liu, Weijia Wang, Yanhong Fan, Lixuan Wu, Ling Sun, and Meiqin Wang. Towards low-latency implementation of linear layers. *IACR Trans. Symmetric Cryptol.*, 2022(1):158–182, 2022. URL: https://doi.org/10.46586/tosc.v2022.i1.158-182.

[LWH$^+$21] Zhenyu Lu, Weijia Wang, Kai Hu, Yanhong Fan, Lixuan Wu, and Meiqin Wang. Pushing the limits: Searching for implementations with the smallest area for lightweight s-boxes. In Avishek Adhikari, Ralf Küsters, and Bart Preneel, editors, *Progress in Cryptology - INDOCRYPT 2021 - 22nd International Conference on Cryptology in India, Jaipur, India, December 12-15, 2021, Proceedings*, volume 13143 of *Lecture Notes in Computer Science*, pages 159–178. Springer, 2021. doi:10.1007/978-3-030-92518-5_8.

[LWS$^+$22] Qun Liu, Weijia Wang, Ling Sun, Yanhong Fan, Lixuan Wu, and Meiqin Wang. More inputs makes difference: Implementations of linear layers using gates with more than two inputs. *IACR Transactions on Symmetric Cryptology*, 2022(2):351–378, Jun. 2022. URL: https://tosc.iacr.org/index.php/ToSC/article/view/9724.

[LXX$^+$23] Da Lin, Zejun Xiang, Runqing Xu, Shasha Zhang, and Xiangyong Zeng. Optimized quantum implementation of aes. Cryptology ePrint Archive, Paper 2023/146, 2023. https://eprint.iacr.org/2023/146. URL: https://eprint.iacr.org/2023/146.

[NC01] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*, volume 2. Cambridge university press Cambridge, 2001.

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

[SM13] Mehdi Saeedi and Igor L. Markov. Synthesis and optimization of reversible circuits—a survey. *ACM Computing Surveys*, 45(2):1–34, February 2013. URL: http://dx.doi.org/10.1145/2431211.2431220, doi:10.1145/2431211.2431220.

[SOT$^+$21] Layth Sliman, Tasnime Omrani, Zahir Tari, Abed Ellatif Samhat, and Rhouma Rhouma. Towards an ultra lightweight block ciphers for internet of things. *J. Inf. Secur. Appl.*, 61:102897, 2021. doi:10.1016/j.jisa.2021.102897.

[Sto16] Ko Stoffelen. Optimizing s-box implementations for several criteria using SAT solvers. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 140–160. Springer, 2016. doi:10.1007/978-3-662-52993-5_8.

[TP20] Quan Quan Tan and Thomas Peyrin. Improved heuristics for short linear programs. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):203–230, 2020. URL: https://doi.org/10.13154/tches.v2020.i1.203-230.

[WL21] Qinglin Wang and Jiqiang Lu. Fault analysis of the ARIA and ublock block ciphers. In Ian McLoughlin, editor, *2021 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Singapore, December 11-12, 2021*, pages 1–6. IEEE, 2021. doi:10.1109/SOLI54607.2021.9672378.

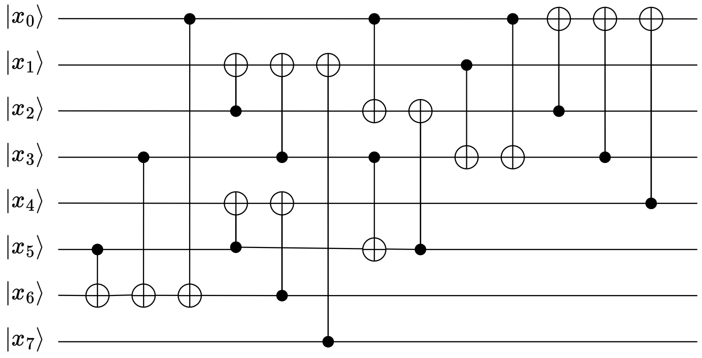
[WZ11] Wenling Wu and Lei Zhang. Lblock: A lightweight block cipher. In Javier López and Gene Tsudik, editors, *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, volume 6715 of *Lecture Notes in Computer Science*, pages 327–344, 2011. doi:10.1007/978-3-642-21554-4_19.

[XZL$^+$20] Zejun Xiang, Xiangyong Zeng, Da Lin, Zhenzhen Bao, and Shasha Zhang. Optimizing implementations of linear layers. *IACR Trans. Symmetric Cryptol.*, 2020(2):120–145, 2020. doi:10.13154/tosc.v2020.i2.120-145.

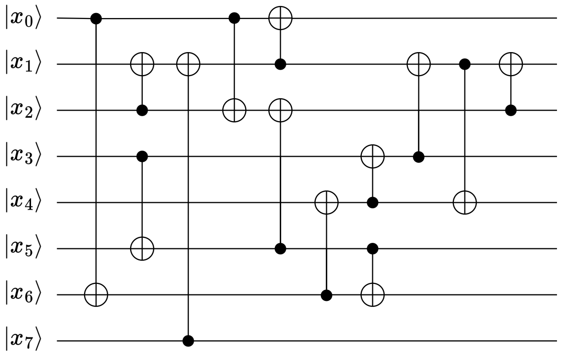
[ZBL$^+$15] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. *Sci. China Inf. Sci.*, 58(12):1–15, 2015. doi:10.1007/s11432-015-5459-7.

[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 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 697–726. Springer, 2020. doi:10.1007/978-3-030-64834-3_24.

A Different quantum implementations of A



(a) Method in [LXX+23] (#GC 16, #FD 9)



(b) **Our model** (#GC 13, #FD 6)

Figure 8: Different quantum implementation of A.