Quantum Implementation and Analysis of ARIA

1st Yujin Oh
IT department
Hansung University
Seoul, South Korea
oyj0922@gmail.com

2nd Kyungbae Jang
IT department
Hansung University
Seoul, South Korea
starj1023@gmail.com

3rd Yujin Yang
IT department
Hansung University
Seoul, South Korea
yujin.yang34@gmail.com

4th Hwajeong Seo
IT department
Hansung University
Seoul, South Korea
hwajeong84@gmail.com

***Abstract*—The progression of quantum computing is considered a potential threat to traditional cryptography system, highlighting the significance of post-quantum security in cryptographic systems. Regarding symmetric key encryption, the Grover algorithm can approximately halve the search complexity. Despite the absence of fully operational quantum computers at present, the necessity of assessing the security of symmetric key encryption against quantum computing continues to grow.**

In this paper, we implement the ARIA block cipher in a quantum circuit and compare it with previous research. Our implementation of the ARIA quantum circuit achieves over 92.5% improvement in full depth and over 98.7% improvement in Toffoli depth compared to the implementation proposed in Chauhan et al. Compared to Yang et al.'s implementation, our implementation is improved the full depth by 36.7% and the number of qubits by 8%. Additionally, we analyze the complexity of Grover's search attack and compare it with NIST criteria. We confirm that ARIA achieves quantum security level 1, 3, and 5 (ARIA-128, 192, and 256, respectively).

***Index Terms*—quantum circuit, Grover algorithm, Post-Quantum security, ARIA**

I. Introduction

The computational power of quantum computers poses a potential threat to existing cryptographic systems, emphasizing the importance of developing quantum-resistant cryptographic systems. Several quantum algorithms are being used to address cryptographic problems on quantum computers, with Shor's algorithm [1], in particular, known for its ability to break classical cryptographic systems like RSA. Additionally, Grover's algorithm [2] can reduce the search complexity of symmetric key cryptography by approximately the square root. Consequently, recent cryptographic research has been actively conducted in the field of quantum computing.

The National Institute of Standards and Technology (NIST) in the United States is actively organizing a competition in the field of Post-Quantum Cryptography (PQC) with the aim of standardizing algorithms resilient to potential quantum attacks. Additionally, NIST defines the quantum security strength based on the cost of Grover's attack against AES-128, AES-192, and AES-256. Based on this, there is a significant amount of research underway to implement quantum circuits and estimate the search complexity of Grover attacks to ascertain whether they achieve the criteria provided by NIST.

Our research implements quantum circuits for ARIA (ARIA-128, 192 and 256), one of the KCMVP ciphers. Furthermore, we analyzes the complexity of Grover's search attack and ensure conformity with the criteria established by NIST. During this process, we incorporate various relevant technologies and compare them with previously reported studies.

A. Our Contribution

Contributions of this paper are:

1) **Depth optimized quantum implementation**
We focus on optimizing the ARIA quantum circuit in terms of depth. As a result, it exhibits the lowest depth compared to previous studies.
2) **Applying various techniques for each part**
We apply various techniques in each part. Additionally, we compare the estimated resources to highlight the most efficient techniques for each part.
3) **Post-Quantum Security Analysis of ARIA**
Our examination of the quantum security of ARIA involves estimating the cost of Grover's key search using the quantum circuit we implemented for ARIA. In this evaluation, we compare the estimated cost of Grover's key search for ARIA with the security levels established by NIST.

II. Related Work

A. ARIA

ARIA is a Korean symmetric key cipher included in the validation subjects of the KCMVP(Korean Cryptographic Module Validation Program). ARIA adopts an SPN (Substitution-Permutation Network) structure and shares similarities with the AES (Advanced Encryption Standard) due to the consideration of AES design principles during its development. The input and output size of ARIA is 128 bits, and it supports key sizes of 128, 192, and 256 bits (the number of rounds 12, 14, 16, respectively). The encryption process of ARIA is shown in Fig 1. The main components of ARIA are the substitution layer, diffusion layer, and key schedule.

1) Substitution layer: ARIA has two types of substitution layers, (LS, LS, LS, LS), $(LS^{-1}, LS^{-1}, LS^{-1}, LS^{-1})$. And each LS has two S-boxes and their inversion $(S_1, S_2, S_1^{-1}, S_2^{-1})$. These S-boxes in ARIA are constructed by applying an affine transformation to the function x^{-1}, and x^{247} over the Galois Field GF(2^8). The equation defining the S-box transformation is as follows:

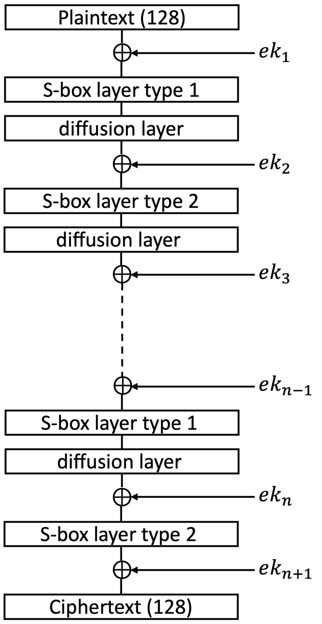


Fig. 1: encryption process of ARIA

$$S_1(x) = \mathbf{A} \cdot x^{-1} \oplus a,$$

$$\text{where} \quad \mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \tag{1}$$

$$\mathrm{S}_2(x) = \mathbf{B} \cdot x^{247} \oplus b = \mathbf{B} \cdot \mathbf{C} \cdot (x^{-1})^8 \oplus b = \mathbf{D} \cdot x^{-1} \oplus b$$

$$\text{where} \quad \mathbf{D} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

2) Diffusion Layer: The diffusion layer is defined by an invertible map A : GF$(2^8)^{16} \to$ GF$(2^8)^{16}$ which is given by $(x_0, x_1, ..., x_{15}) \to (y_0, y_1, ..., y_{15})$. It can be represented as a series of operations executed through a 16 × 16 binary matrix multiplication as follows.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&1&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix} \tag{3}$$

3) KeySchedule: The key scheduling of ARIA comprises two parts: initialization and round key generation. During the initialization part, four 128-bit values W_0, W_1, W_2, W_3 are derived from the master key MK.

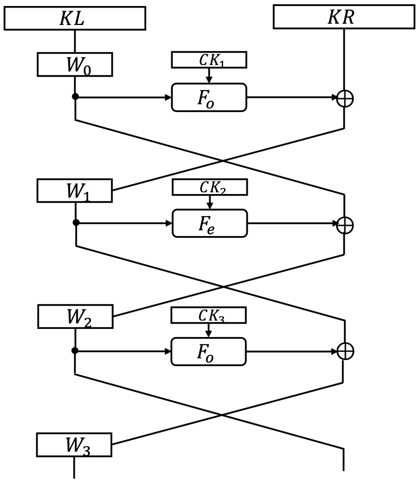


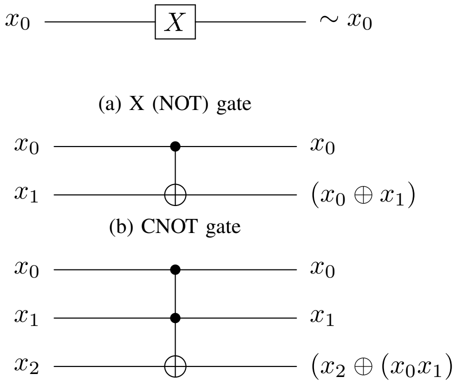
Fig. 2: Key Initialization of ARIA

During the round key generation part, the four values W_0, W_1, W_2, W_3 are used to derive the encryption round keys ek_i (each of 128 bits). ARIA employs 12, 14, and 16 rounds, corresponding to master key sizes of 128, 192, and 256 bits, respectively. As an additional key is required for the final round key addition, the total number of round keys needed is 13, 15, or 17, respectively. The equation of generate the round key is as follow:

$$\begin{array}{ll} ek_1 = (W_0) \oplus (W_1 \ggg 19), & ek_2 = (W_1) \oplus (W_2 \ggg 19) \\ ek_3 = (W_2) \oplus (W_3 \ggg 19), & ek_4 = (W_0 \ggg 19) \oplus (W_3) \\ ek_5 = (W_0) \oplus (W_1 \ggg 31), & ek_6 = (W_1) \oplus (W_2 \ggg 31) \\ ek_7 = (W_2) \oplus (W_3 \ggg 31), & ek_8 = (W_0 \ggg 31) \oplus (W_3) \\ ek_9 = (W_0) \oplus (W_1 \lll 61), & ek_{10} = (W_1) \oplus (W_2 \lll 61) \\ ek_{11} = (W_2) \oplus (W_3 \lll 61), & ek_{12} = (W_0 \lll 61) \oplus (W_3) \\ ek_{13} = (W_0) \oplus (W_1 \lll 31), & ek_{14} = (W_1) \oplus (W_2 \lll 31) \\ ek_{15} = (W_2) \oplus (W_3 \lll 31), & ek_{16} = (W_0 \lll 31) \oplus (W_3) \\ ek_{17} = (W_0) \oplus (W_1 \lll 19) & \end{array} \tag{4}$$

B. Quantum gates

Figure 3 shows some representative quantum gates. First, Figure 3a depicts the X gate. The X gate flips the state of a qubit and is equivalent to the NOT operation in classical computing. Figure 3b represents the CNOT gate. The CNOT gate uses two qubits (1 control qubit, 1 target qubit), and when the control qubit is 1, it flips the value of the target qubit. Figure 3c represents the Toffoli gate. The Toffoli gate requires two control qubits and one target qubit. When both control qubits are set to 1, the target qubit is flipped, similar to an AND operation. The Toffoli gate is composed of a combination of various other gates. There have been numerous studies on decomposing the Toffoli gate [3]–[6]. In this study, we utilize a method outlined in [3] to decompose the Toffoli gate, requiring 7 T gates, 8 Clifford gates, with a total depth of 8 (where the T-depth is 4). Optimization of Toffoli gates and depth is crucial due to the significant quantum resources it demands compared to other gates.



(a) X (NOT) gate

(b) CNOT gate

(c) Toffoli (CCNOT) gate

Fig. 3: Quantum gates.

C. Grover's key search

The Grover's key search consists of three main steps, as follows.

1) *Input Setting*: By applying Hadamard gates, we prepare a k-qubit key into a superposition state $|\psi\rangle$, where all 2^k possible states are equally probable.

$$H^{\otimes k}|0\rangle^{\otimes k} = |\psi\rangle = \Big(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\Big) = \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} |x\rangle$$

2) The *Oracle* is constructed as a quantum circuit that encrypts the known plaintext(p) using the key in the prepared superposition state in the input setting. In this process, ciphertexts for all possible key values are generated. These ciphertexts, effectively comprising a single ciphertext in a superposition state, undergoes comparison with the known ciphertext($f(x)$). When a match is found (i.e., if $f(x) = 1$ in Expression (5), the sign of the key state to be recovered is flipped (i.e., if $(-1)^{f(x)} = -1$ according to Expression (6). Finally, to prepare for the next iteration, the quantum circuit that has been implemented is reversed thereby converting the generated ciphertext back into the known plaintext. Ultimately, a single oracle encompasses two quantum circuits.

$$f(x) = \begin{cases} 1 \text{ if } Enc_{key}(p) = c \\ 0 \text{ if } Enc_{key}(p) \neq c \end{cases} \tag{5}$$

$$U_f(|\psi\rangle|-\rangle) = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle|-\rangle \tag{6}$$

3) The *Diffusion Operator* amplifies the amplitude of states with a negative sign that have been inverted by the oracle. It can be easily implemented using H, X, and k-qubit controlled Z gates. The overhead of the diffusion operator is so insignificant compared to that of the oracle that it is often overlooked in the cost analysis of the Grover search algorithm [7]–[9].

III. Quantum Circuit Implementation..

In this section, we describe depth-optimized quantum circuits for ARIA components and compare them with previous works.

Our purpose for implementing the quantum circuit prioritizes minimizing the depth of the quantum circuit over reducing the number of qubits. Nevertheless, it is crucial to emphasize that the number of qubits remains a fundamental resource in quantum circuit. In this regard, we implement ARIA quantum circuit with a focus on decreasing circuit depth while balancing the trade-off between the number of qubits and the depth.

A. Implementation of S-box

$$x^{-1} = x^{254} = ((x \cdot x^2) \cdot (x \cdot x^2)^4 \cdot (x \cdot x^2)^{16} \cdot x^{64})^2 \tag{7}$$

In SPN (Substitution-Permutation Network) structures such as ARIA, the S-box performs a nonlinear transformation and is primarily implemented based on lookup tables. On the contrary, in quantum computing, the inherent properties of superposition and entanglement poses difficulties for the direct utilization of lookup tables. Therefore, S-boxes should be implemented using Boolean expressions through quantum gates.

The expressions of S-boxes for ARIA are provided in Equations (1) and (2). To implement S-boxes, we first need to compute the inversion, x^{-1}. To calculate the inversion in a Galois field GF(2^8), the Itoh-Tsujii algorithm [10] can be employed. This algorithm efficiently computes the inversion using multiplication and squarings.

Chauhan et al. [11] used schoolbook multiplications and employed PLU decomposition [12] for squarings. In order to save on the number of qubits, they employed for inverse squarings and inverse multiplications. This resulted in a total of 7 multiplications, 33 squarings and utilized only 40 qubits. However, this strategy led to an increase in the depth while keeping qubit count low.

In the previous work [13], they used a optimized multiplication [14] to reduce the depth. This multiplication utilizes the Karatsuba algorithm recursively and allocates ancilla qubits. Through this, all Toffoli gates can operate in parallel, leading to a Toffoli depth of 1. Additionally, the reuse of ancilla qubits allows for efficiency in the implementation of inversion, which involves multiple multiplications.

In our implementation, we apply different methods to each S-box (S_1 and S_2). S_1 in ARIA is identical to the AES S-box. Recently, there has been considerable research on AES. In particular, based on Boyar-Peralta algorithm [15], [16], there are numerous studies focused on optimizing AES quantum circuits [8], [9], [17]–[19]. We apply the implementation by Jang et al. [9], which achieved the best depth reduction (while using a reasonable number of qubits), to the ARIA S-box. By applying this method, we can significantly reduce both the depth and the number of qubits and gates compared to previous research. This reduction is particularly impactful due to the significant decrease in Toffoli gate operations. This For S_1^{-1}, we combine the implementation of Jang et al. [9] and Huang et al. [19]. According to [19], implementing the inverse of S_1 (S_1^{-1}) requires the S_1 circuit. Therefore, we implement the inverse circuit of S_1 (S_1^{-1}) by replacing only the S_1 circuit part with the circuit in [9], which features the most depth optimization, in inverse circuit from [19].

However, S_2 can not be implemented by Boyar-Peralta algorithm, so we also use Itoh-Tsujii algorithm same as previous works. Similar to [13], we use the optimized Karatsuba algorithm, which is Toffoli depth one. In Squaring, we use XZLBZ [20] method. This is an in-place operation, similar to PLU decomposition, but it can reduce the number of CNOT gates and depth compared to PLU decomposition. Figure 4 shows the quantum circuit for the squaring operation using XZLBZ. Additionally, we implement the matrix-vector multiplication by allocating 8 ancilla qubits for each S-box (i.e., out-of-place).

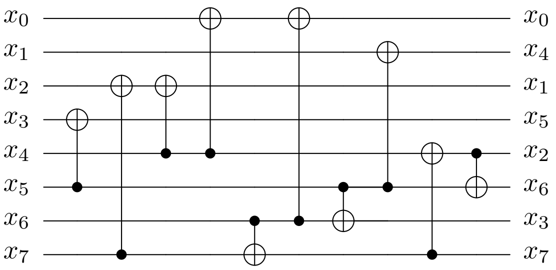


Fig. 4: Squaring in $\mathbb{F}_{2^8}/(x^8 + x^4 + x^3 + x + 1)$ using XZLBZ

Table I illustrates the quantum resources required for S-box implementation. Our approach using Itoh-Tsujii algorithm is applied only to S_2. However, for comparison, note that quantum resources applied to S_1 are presented.

In result, we can reduce the qubits and gate counts but can not affect the full depth for the overall circuit. It describes in Section III-B.

B. Implementation of Substitution Layer

ARIA employs two types of S-boxes, namely S_1 and S_2 , along with two distinct substitution layers denoted as (LS, LS, LS, LS) and $(LS^{-1}, LS^{-1}, LS^{-1}, LS^{-1})$, where LS = $(S_1, S_2, S_1^{-1}, S_2^{-1})$. Thus, each substitution layer utilizes 16 S-boxes. Similar to previous papers, we reduce the depth by parallelizing the processing of all S-boxes in each substitution layer. In [13], 608 (38 × 16) reusable ancilla qubits were initially allocated to process all S-boxes in parallel. In our implementation, as described in section III-A, different techniques are applied to S_1 and S_2, requiring ancilla qubits only for S_2. Therefore, we initially allocate a total of 304 (38 ×8) ancilla qubits to process them in parallel.

TABLE I: Quantum resources required for implementations of a S-box(S1).

Method	Source	#CNOT	#X	#Toffoli	Toffoli depth	#Qubit	depth
Itoh-Tsujii	[11]	569	4	448	196	40	-
	[13]	1114	4	108	4	162	151
	Ours	1106	4	108	4	170	137
Boyar-Peralta	**Ours**	162	4	34	4	84	33

Parallel processing allows for a significant reduction in depth compared to sequential operations, but it comes with one drawback in our implementation. Due to parallel processing, the technique applied to S_1 has been beneficial in reducing the number of qubits, but there is no corresponding gain in terms of depth. This is because the depth cost of S_2 is higher than that of S_1, resulting in the depth of a substitution layer being measured by S_2.

C. Implementation of Diffusion Layer

The diffusion layer can be expressed as a set of operations performed through a 16 × 16 binary matrix multiplication. To implement linear operations like matrix multiplication, various methods can be adopted. Firstly, there is the option of in-place operations, where no additional qubits are allocated. In [11], [13], the PLU decomposition technique was chosen. Additionally, for in-place operations, the XZLBZ technique can also be utilized. While these implementations lead to a reduction in the number of qubits required for the quantum circuit, the limited computational space resulting from the small number of qubits necessitates the sequential operations of CNOT gates, leading to an increase in circuit depth. Therefore, we adopt an out-of-place approach of allocating ancilla qubits to store the results, reducing the depth.

In our approach, 128 ancilla qubits are allocated for each round (i.e., out-of-place) to store the output of the diffusion layer. Algorithm 1 demonstrates the implementation of the out-of-place method on the diffusion layer. During this process, we aim to minimize the depth by reordering CNOT gates to maximize parallel processing wherever possible.

As shown in Table II, we observe that XZLBZ achieves a smaller depth. Furthermore, the out-of-place method allocates more qubits but demonstrates superior depth efficiency.

IV. Performance

In this section, we provide an estimated quantum resources of our ARIA-128, 192, 256 quantum circuit implementationm comparing the previous works. We utilize the ProjectQ quantum

Algorithm 1: Quantum circuit implementation of ARIA Diffusion Layer using out-of-place.

```
Input: $x$, $M$
Output: $result$
Allocate result qubit $\rightarrow result[16][8]$
for $0 \leq i \leq 16$ do
    for $0 \leq j \leq 16$ do
        if $M[16+j]$==1 then
            CNOT8bit($x$, $j$, $result$, $i$)
return result =0
```

TABLE II: Quantum resources required for implementations of a Diffusion layer.

Method	#CNOT	#Qubit	depth
PLU	768	128	31
XZLBZ	376	128	17
Out-of-place	896	256	**7**

programming tool for both implementation and simulation of the quantum circuits. The correctness of the implementation is validated using the ClassicalSimulator library within ProjectQ, and we scrutinize the quantum resources utilized with the assistance of the ResourceCounter.

Table III and IV present resource estimates for the ARIA quantum circuits, providing estimations for the NCT level and Clifford+T level, respectively. As mentioned in Section II-B, since the Toffoli gate can be decomposed into Clifford and T gates, we also provide resource estimates for this decomposition in Table IV.

Additionally, we compare the resource costs of our implementation with previous works in both Table III and IV to assess the efficiency of our implementation. However, since [11] did not provide the decomposed quantum resource costs, we rely on the estimation provided by [13], which used the information from [11] to estimate the quantum resource costs. As shown in Table III and IV, our implementation achieves the most optimized depth and the number of gates.

V. Evauation

In this section, we estimate the cost of Grover's key search for ARIA. According to various studies ([7], [8], [21] and others), the optimal number of iterations for Grover's key search in a cipher using a k-bit key is approximately $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$. Additionally, the quantum circuit for Grover's key search on block ciphers consists of repeated oracle and diffusion operators. The cost estimation for Grover's key search only calculates the quantum resources for the oracle, as the overhead of the diffusion operator is negligible. In the oracle, two quantum circuits are executed sequentially. Therefore, the cost of Grover's attack is calculated as $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor \times 2 \times$ quantum resources (Table IV).

However, there is an additional point to note. We should focus on the number of plaintext-ciphertext pairs (r) required to discover the unique key. In [8], [17], it was proposed that obtaining $r = \lceil$key size/block size$\rceil$ pairs of plaintext-ciphertext is adequate to identify a unique key. As a result, the conclusive cost of Grover's search is $2 \times r \times \lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor \times$ Table IV.

NIST has defined criteria (Level 1, 3, and 5) for quantum security based on the complexity of Grover's attack on AES (AES-128, 192, 256) to estimate the quantum resistance strength for symmetric-key cryptography [22], [23]. The metric used for attack complexity cost is Total depth × Total gates. This metric considers gate count and depth more than the number of qubits since gates and depth continue to increase while the qubit count remains fixed during the operation of the Grover algorithm. Thus, gates and depth were prioritized over qubit count in this metric

NIST initially established Level 1, 3, and 5 based on AES Grover attack costs by Grassl et al. (2^{170}, 2^{233}, 2^{298}, respectively) [22]. In this case, it can be noted that ARIA can not achieve those Levels. However, these estimates were considered too high. With recent advancements in AES research leading to reduced attack costs, NIST has introduced new post-quantum security standards [23]. Based on the work by Jaques et al. [8], the attack costs for Level 1, 3, and 5 have been revised to 2^{157}, 2^{221}, and 2^{285}, respectively.

In Table V, our work demonstrates the lowest cost in terms of total depth, total gates, and complexity (total depth × total gates) considered by NIST, compared to previous studies. Additionally, we can confirm that ARIA-128, 192, and 256 achieve levels 1, 3, and 5 respectively.

VI. Conclusion

In this paper, we emphasized optimizing the quantum circuit depth for ARIA block cipher. We provided a detailed of the implementation of ARIA quantum circuit focusing on fundamental components, such as the S-box and diffusion layer. We endeavored to minimize the depth of the quantum circuit by employing various novel techniques aimed at achieving the lowest quantum circuit depth. Subsequently, we estimated the required quantum resources and the cost of Grover's search attack on ARIA.

Based on this, we can conclude that ARIA-128, 192, and 256 achieve quantum security level 1, 3 and 5, respectively. Our implementation of the ARIA quantum circuit achieves over 92.5% improvement in full depth and over 98.7% improvement in Toffoli depth compared to the implementation proposed in [11]. Compared to [13], our implementation is improved the full depth by 36.7% and the number of qubits by 8%.

Lastly, our implementation has significantly optimized the depth for the S-boxes. As mentioned in Section III-B, this optimization cannot impact the reduction of the overall circuit depth. In future work, we plan to explore the Boyar-Peralta technique for all S-boxes and integrate it. We anticipate achieving more innovative optimization in terms of depth and reducing the number of qubits through this approach.

VII. Acknowledgement

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea govern-

TABLE III: Required quantum resources for ARIA quantum circuit implementation

Cipher	Source	#X	#CNOT	#Toffoli	Toffoli depth	#Qubit	Depth
ARIA-128	[11]	1,595	231,124	157,696	4,312	1,560	9,260
	[13]	1,408	285,784	25,920	60	29,216	3,500
	This work	1,408	173,652	17,040	60	26,864	**2,187**
ARIA-192	[11]	1,851	273,264	183,368	5,096	1,560	10,948
	[13]	1,624	324,136	29,376	68	32,928	3,978
	This work	1,624	197,036	19,312	68	30,320	**2,480**
ARIA-256	[11]	2,171	325,352	222,208	6,076	1,688	13,054
	[13]	1,856	362,488	32,832	76	36,640	4,455
	This work	1,856	220,420	21,584	76	33,776	**2,772**

TABLE IV: Required decomposed quantum resources for ARIA quantum circuit implementation

Cipher	Source	#Clifford	# T	T-depth	#Qubit	Full depth
ARIA-128	[11]	1,494,287	1,103,872	17,248	1,560	37,882
	[13]	494,552	181,440	240	29,216	4,650
	This work	311,380	119,280	240	26,864	**2,952**
ARIA-192	[11]	1,742,059	1,283,576	20,376	1,560	44,774
	[13]	560,768	205,632	272	32,928	5,285
	This work	353,156	135,184	272	30,320	**3,347**
ARIA-256	[11]	2,105,187	1,555,456	24,304	1,688	51,666
	[13]	627,000	229,824	304	36,640	5,919
	This work	394,948	151,088	304	33,776	**3,741**

TABLE V: Cost of the Grover's key search for ARIA

Cipher	Source	Total gates	Total depth	Cost (complexity)	#Qubit	NIST security
ARIA-128	[11]	$1.998 \cdot 2^{85}$	$1.816 \cdot 2^{79}$	$1.814 \cdot 2^{165}$	1,561	Level 1
	[13]	$1.117 \cdot 2^{84}$	$1.783 \cdot 2^{76}$	$1.991 \cdot 2^{160}$	29,217	
	This work	$\mathbf{1.296 \cdot 2^{83}}$	$\mathbf{1.132 \cdot 2^{76}}$	$\mathbf{1.468 \cdot 2^{159}}$	**26,865**	
ARIA-192	[11]	$1.146 \cdot 2^{119}$	$1.073 \cdot 2^{112}$	$1.23 \cdot 2^{231}$	3,121	Level 3
	[13]	$1.2 \cdot 2^{117}$	$1.013 \cdot 2^{109}$	$1.216 \cdot 2^{226}$	65,857	
	This work	$\mathbf{1.469 \cdot 2^{116}}$	$\mathbf{1.284 \cdot 2^{108}}$	$\mathbf{1.886 \cdot 2^{224}}$	**60,449**	
ARIA-256	[11]	$1.384 \cdot 2^{151}$	$1.238 \cdot 2^{144}$	$1.714 \cdot 2^{295}$	3,377	Level 5
	[13]	$1.336 \cdot 2^{149}$	$1.135 \cdot 2^{141}$	$1.516 \cdot 2^{290}$	72,081	
	This work	$\mathbf{1.642 \cdot 2^{148}}$	$\mathbf{1.435 \cdot 2^{140}}$	$\mathbf{1.178 \cdot 2^{289}}$	**67,553**	

ment(MSIT).(No. RS-2023-00277994, Quantum Circuit Depth Optimization for ARIA, SEED, LEA, HIGHT, and LSH of KCMVP Domestic Cryptographic Algorithms, 80%) and this work was partly supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (<Q|Crypton>, No.2019-0-00033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 20%).

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. IEEE, 1994, pp. 124–134.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[3] M. Amy, D. Maslov, M. Mosca, M. Roetteler, and M. 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*, vol. 32, no. 6, p. 818–830, Jun 2013. [Online]. Available: http://dx.doi.org/10.1109/TCAD.2013.2244643

[4] Y. He, M.-X. Luo, E. Zhang, H.-K. Wang, and X.-F. Wang, "Decompositions of n-qubit Toffoli gates with linear circuit complexity," *International Journal of Theoretical Physics*, vol. 56, no. 7, pp. 2350–2361, 2017.

[5] P. Selinger, “Quantum circuits of T-depth one,” *Physical Review A*, vol. 87, no. 4, p. 042302, 2013.

[6] P. Niemann, A. Gupta, and R. Drechsler, “T-depth optimization for fault-tolerant quantum circuits,” in *2019 IEEE 49th International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE, 2019, pp. 108–113.

[7] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, “Applying Grover’s algorithm to AES: Quantum resource estimates,” in *Post-Quantum Cryptography*, T. Takagi, Ed. Cham: Springer International Publishing, 2016, pp. 29–43.

[8] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, “Implementing Grover Oracles for quantum key search on AES and LowMC,” in *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*, ser. Lecture Notes in Computer Science, A. Canteaut and Y. Ishai, Eds., vol. 12106. Springer, 2020, pp. 280–310. [Online]. Available: https://doi.org/10.1007/978-3-030-45724-2_10

[9] K. Jang, A. Baksi, H. Kim, G. Song, H. Seo, and A. Chattopadhyay, “Quantum analysis of AES,” Cryptology ePrint Archive, Paper 2022/683, 2022, https://eprint.iacr.org/2022/683. [Online]. Available: https://eprint.iacr.org/2022/683

[10] T. Itoh and S. Tsujii, “A fast algorithm for computing multiplicative inverses in gf (2m) using normal bases,” *Information and computation*, vol. 78, no. 3, pp. 171–177, 1988.

[11] A. K. Chauhan and S. K. Sanadhya, “Quantum resource estimates of grover’s key search on aria,” in *Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, December 17–21, 2020, Proceedings 10*. Springer, 2020, pp. 238–258.

[12] I. Van Hoof, “Space-efficient quantum multiplication of polynomials for binary finite fields with sub-quadratic Toffoli gate count,” *arXiv preprint arXiv:1910.02849*, 2019.

[13] Y. Yang, K. Jang, Y. Oh, and H. Seo, “Depth-optimized quantum implementation of aria,” *Cryptology ePrint Archive*, 2023.

[14] K. Jang, W. Kim, S. Lim, Y. Kang, Y. Yang, and H. Seo, “Optimized implementation of quantum binary field multiplication with Toffoli depth one,” in *Information Security Applications: 23rd International Conference, WISA 2022, Jeju Island, South Korea, August 24–26, 2022, Revised Selected Papers*. Springer, 2023, pp. 251–264.

[15] J. Boyar and R. Peralta, “A new combinational logic minimization technique with applications to cryptology,” in *Experimental Algorithms*, P. Festa, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 178–189.

[16] ——, “A depth-16 circuit for the AES S-box,” Cryptology ePrint Archive, Report 2011/332, 2011, https://eprint.iacr.org/2011/332.

[17] B. Langenberg, H. Pham, and R. Steinwandt, “Reducing the cost of implementing the advanced encryption standard as a quantum circuit,” *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–12, 01 2020.

[18] J. Zou, Z. Wei, S. Sun, X. Liu, and W. Wu, “Quantum circuit implementations of aes with fewer qubits,” in *Advances in Cryptology – ASIACRYPT 2020*, S. Moriai and H. Wang, Eds. Cham: Springer International Publishing, 2020, pp. 697–726.

[19] Z. Huang and S. 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.

[20] Z. Xiang, X. Zeng, D. Lin, Z. Bao, and S. Zhang, “Optimizing implementations of linear layers,” *IACR Trans. Symmetric Cryptol.*, vol. 2020, no. 2, pp. 120–145, 2020. [Online]. Available: https://doi.org/10.13154/tosc.v2020.i2.120-145

[21] K. Jang, G. Song, H. Kim, H. Kwon, H. Kim, and H. Seo, “Parallel quantum addition for Korean block cipher,” *IACR Cryptol. ePrint Arch.*, p. 1507, 2021. [Online]. Available: https://eprint.iacr.org/2021/1507

[22] 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.

[23] ——, “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.