Depth-Optimized Implementation of ASCON Quantum Circuit

Yujin Oh[1][0009−0005−6693−277X],
Kyungbae Jang[1][0000−0001−5963−7127],
Anubhab Baksi[2][0000−0002−5639−7372], and
Hwajeong Seo[1][0000−0003−0069−9061]

[1] Hansung University, Seoul (02876), South Korea

[2] Nanyang Technological University, 50 Nanyang Ave (639798), Singapore

oyj0922@gmail.com, starj1023@gmail.com, anubhab.baksi@ntu.edu.sg, hwajeong84@gmail.com

Abstract. The development of quantum computers, which employ a different paradigm of computation, is posing a threat to the security of cryptography. Narrowing down the scope to symmetric-key cryptography, the Grover search algorithm is probably the most influential in terms of its impact on security. Recently, there have been efforts to estimate the complexity of the Grover's key search for symmetric key ciphers and evaluate their post-quantum security.

In this paper, we present a depth-optimized implementation of a quantum circuit for ASCON, which is a symmetric key cipher that has recently been standardized in the NIST (National Institute of Standards and Technology) Lightweight Cryptography standardization. As far as we know, this is the first implementation of a quantum circuit for the ASCON AEAD (Authenticated Encryption with Associated Data) scheme. To our understanding, reducing the depth of the quantum circuit for the target cipher is the most effective approach for Grover's key search. We demonstrate the optimal Grover's key search cost for ASCON, along with a proposed depth-optimized quantum circuit. Further, based on the estimated cost, we evaluate the post-quantum security strength of ASCON according to relevant evaluation criteria and state-of-the-art research.

Keywords: Grover's Algorithm · NIST · Lightweight Cryptography · ASCON · Post-Quantum Security.

1 Introduction

Due to the powerful/potential threat posed by quantum computers, researchers and organizations are reassessing the security of cryptographic algorithms in the field of cryptography. One of the noteworthy endeavors is the NIST Post-Quantum Cryptography (PQC) standardization process[3]. The need for quantum-

[3] https://csrc.nist.gov/projects/post-quantum-cryptography

safe cryptography arises from the fact that Shor's algorithm [1] can solve factorization and discrete logarithm problems (security foundation of RSA and Elliptic Cruve Cryptography) in polynomial time.

Another notable quantum algorithm used in cryptography is due to Grover [2]. It has the ability to accelerate data search, thereby reducing the complexity of exhaustive search in symmetric key cryptography. The Grover's algorithm indeed reduces the security strength, but the quantum circuit required for the attack is significantly large. Quantum attacks can be evaluated from two perspectives: problem-solving power and the size of the quantum circuits required to solve those problems. A different interpretation of this would be that the security of a cryptographic algorithm can be evaluated differently based on the size of the quantum circuit needed for a quantum attack. This is addressed in the NIST Post-Quantum Cryptography document, where the post-quantum security strength is determined by considering the quantum cost needed for quantum attacks (which will be explained in Section 2.3). NIST defines the post-quantum security strength based on the cost of Grover's attack against AES-128, -192, -256 (conceptually similar to how the security parameters of PQC algorithms are related to the AES family). The cost of the Grover attack is determined by the implementation efficiency of the quantum circuit for the targeted cryptographic algorithm.

This paper presents an optimized quantum circuit for the AEAD scheme of ASCON [3], which has been selected as part of the NIST Lightweight Cryptography standardization[4]. It is also used in the post-quantum signature, Ascon-Sign [4]. Our focus is to minimize the depth of the ASCON quantum circuit while maintaining a reasonable number of qubits, aligning with the concept of the Grover's algorithm. The depth of the quantum circuit is directly related to the execution time of the circuit [5]. While Grover's algorithm reduces the search complexity by the square root, it still requires a significant number of iterations in the quantum circuit. In other words, Grover's exhaustive key search is a time-consuming process, and NIST also takes this into account when evaluating security. As far as we understand, minimizing the depth is the optimal strategy for Grover's algorithm (more discussion in Section 2.3), and it has become the design philosophy of the ASCON quantum circuit. Finally, based on the proposed ASCON quantum circuit, we estimate the cost of the Grover attack and evaluate the post-quantum security strength of ASCON according to NIST's criteria.

Our Contribution

The contribution in this paper is manifold and can be summarized as follows:

1. **Quantum Circuit Implementation of ASCON AEAD.** We present the first implementation of a quantum circuit for ASCON AEAD.

[4] https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon

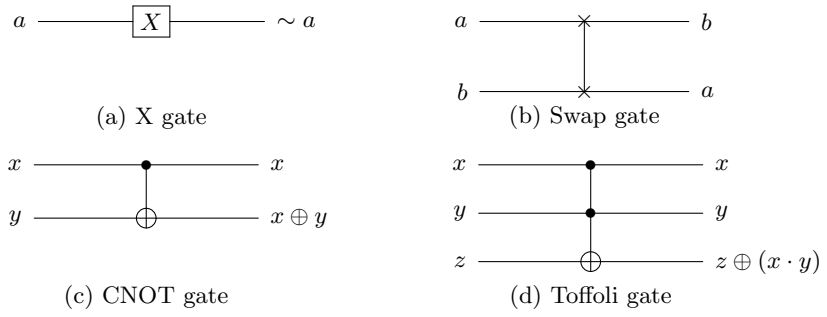
2. **Low-Depth Implementation of ASCON AEAD.** In our quantum circuit implementation of ASCON, we prioritize achieving a low Toffoli depth and full depth. We demonstrate the reduction of Toffoli depth and full depth through parallelization. Additionally, to maintain a reasonable qubit count, we utilize the method of reusing ancilla qubits.
3. **Post-quantum Security Assessment of ASCON AEAD.** We assess the quantum security of ASCON by estimating the cost of Grover's key search based on our implemented quantum circuit for ASCON. This evaluation involves comparing the estimated cost of Grover's key search for ASCON with the security levels provided by NIST.

2 Preliminaries

2.1 Quantum Gates

In this section, we explain commonly employed quantum gates for constructing quantum circuits used in block ciphers (this list does not encompass all possible gates that can be employed for this purpose).

Figure 1(a) shows the quantum X gate that can replace the classical NOT operation. The qubit state is reversed through the X gate. Figure 1(b) represents Swap gate that exchanges two qubit states. The quantum CNOT gate illustrated in Figure 1(c) serves as a replacement for the classical XOR operation. By using one control qubit, the CNOT gate determines the value of the target qubit. Figure 1(d) shows the quantum Toffoli gate, which acts as an alternative to the classical AND operation. The Toffoli gate employs two control qubits to determine the value of the target qubit . Note that the Toffoli gate is implemented using various quantum gates, such as the T, CNOT, X, and H gates, among others [6]. Therefore, it is important to minimize Toffoli-related metrics when optimizing quantum circuits.



(a) X gate (b) Swap gate

(c) CNOT gate (d) Toffoli gate

Fig. 1: Common (top level) quantum gates.

2.2 Key Search using Grover's Algorithm

In cryptography, for an encryption scheme that uses a k-bit key, a classical computer requires a search of $O(2^k)$ complexity for exhaustive key search. However, thanks to Grover's algorithm, a quantum computer can accomplish this search with a reduced complexity of only $O(\sqrt{2^k})$, which is decreased by a square root. We split the steps of the Grover's key search into three stages; and describe them as follows.

1. *Input Setting*: Hadamard gates are used to prepare a k-qubit key in a superposition state $|\psi\rangle$, resulting in equal amplitudes for all 2^k possible states.

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

2. In the *Oracle*, the target cipher is implemented as a quantum circuit that encrypts the known plaintext using the previously prepared key in a superposition state, generating ciphertexts for all possible key values. These ciphertexts (actually one ciphertext in a superposition state) are then compared with the known ciphertext, and if a match is found (i.e., if $f(x) = 1$ in Expression (1)), the sign of the key state to be recovered is negated (i.e., if $f(x) = -1$ in Expression (2)). Finally, the implemented quantum circuit is reversed, transforming the generated ciphertexts back into the known plaintext (for the next iteration).

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

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

3. The *Diffusion Operator* serves to enhance the amplitude of the target key state marked by the oracle, which is identified by changing the sign of the corresponding amplitude to negative. The diffusion circuit is often a standard, off-the-shelf design that can be easily implemented. As the overhead of the diffusion operator is negligible compared to that of the oracle, it is typically disregarded in the cost analysis of the Grover's search algorithm [7,8,9]. In practice, Grover's algorithm executes a sufficient number of iterations of the oracle and diffusion to boost the amplitude of the target key state, thereby enabling a high probability of measuring the solution key.

2.3 NIST Security Criteria

We need to take note of NIST's document [10,11] on current/potential quantum attacks related to ciphers. NIST establishes criteria for post-quantum security based on the complexity of quantum attacks on the AES and SHA-2/3 families.

For the sake of brevity, this paper will only mention the criteria for estimating quantum attack complexity on the AES family (corresponding to levels 1, 3 and 5). Levels 2 and 4 that correspond to the complexity of quantum attacks (collision search) on SHA-2/3 families. Unlike the hashes, the tag depends on the secret key; so one may consider further quantum security level to account for it.

- **Level 1**: To be considered secure, any attack that compromises the relevant security definition must require computational resources that are at least comparable to those required for a key search on a 128-bit key block cipher, such as AES-128 ($2^{170} \rightarrow 2^{157}$).
- **Level 3**: To be considered secure, any attack that compromises the relevant security definition must require computational resources that are at least comparable to those required for a key search on a 192-bit key block cipher, such as AES-192 ($2^{233} \rightarrow 2^{221}$).
- **Level 5**: To be considered secure, any attack that compromises the relevant security definition must require computational resources that are at least comparable to those required for a key search on a 256-bit key block cipher, such as AES-256 ($2^{298} \rightarrow 2^{285}$).

As is well known, Grover's search algorithm is one of the optimal quantum attacks on symmetric key ciphers, and NIST also takes this into account. The difficulty of attacks corresponding to Levels 1, 3, and 5 are determined by the cost of Grover's key search on AES-128, 192, and 256, respectively, which is calculated by the total gate count $\times$ depth of Grover's key search circuit. NIST estimates the costs for Levels 1, 3 and 5 to be $2^{170}, 2^{233}$, and 2^{298}, respectively; based on the AES quantum circuit implementation by Grassl et al [7]. Recently, the costs of Grover's key search on the AES family have been adjusted/decreased by NIST [11]. In recent years, there have been numerous efforts to optimize the quantum circuits of AES [12,13,14,15,9,8]. Among them, Jaques et al. presented depth-optimized quantum circuits for AES at Eurocrypt'20, reporting a decreased cost of Grover's key search on AES [8]. Currently, NIST has newly defined the quantum attack cost for AES-128, 192, and 256 based on the reported costs from [8] as 2^{157}, 2^{221} and 2^{285}; respectively[5]. To the best of our knowledge, the currently best-known results are reported in [9].

Furthermore, we must also take into account NIST's specified MAXDEPTH, which represents the maximum circuit depth that a quantum computer can execute. NIST classifies the depth limitations of quantum attacks (i.e., MAXDEPTH) into the following ranges: ($2^{40} < 2^{64} < 2^{96}$) because it considers that the extreme depth of Grover's key search (due to numerous sequential iterations) makes the attack practically difficult.

[5] Note that although there are some programming-related issues reported in their quantum circuit implementation, Jang et al. analyze those issues from [9] and demonstrate that the reported costs in [8] are achievable with their optimized AES quantum circuits.

2.4 ASCON

ASCON is a symmetric key cipher that has been standardized in the NIST Lightweight Cryptography standardization. ASCON includes an authenticated encryption with associated data (AEAD) mode, a hash function, and a variant called Ascon-80pq, which provides enhanced resistance against quantum key-search attacks. Within ASCON, there are two versions of the AEAD mode: ASCON-128 and ASCON-128a. The encryption process in ASCON involves several phases: *Initialization*, *Processing Associated Data*, *Processing Plaintext*, and *Finalization*.

The main components of all schemes in ASCON are two 320-bit permutations, denoted as p^a and p^b, with different numbers of rounds (a and b correspond to 12 and 6 rounds, respectively). These permutations are used in all phases of ASCON. For computational purposes, the 320-bit state S is divided into five 64-bit register words x_i ($S = x_0||x_1||x_2||x_3||x_4$, where x_0 is the most significant word and x_4 is the least significant word). The permutation functions include adding constants, a substitution layer with a 5-bit S-box, and a linear layer with 64-bit diffusion functions.

3 Quantum Implementation of ASCON

As noted earlier, ASCON has two schemes (AEAD and hash), and our primary focus lies on ASCON-128, which corresponds to the AEAD variant. With our design philosophy that emphasizes minimizing the depth for optimal performance in Grover's algorithm, we focus on optimizing the depth of the ASCON-128 quantum circuit while also maintaining a reasonable number of qubits.

3.1 Implementation (with Parallelization) of S-box

Due to the reversible nature of quantum computing, the implementation of S-boxes using look-up table based methods is not suitable. Therefore, it becomes apparent to implement S-box quantum circuits based on Boolean expression (of the coordinate functions) using quantum gates. The 5-bit ASCON S-box can be implemented by using the Boolean relationships that involve NOT ($\sim$), AND ($\cdot$) and XOR ($\oplus$) gates; and are shown in Expression (3), this is adopted from [3, Section 7.3]. Note that the current quantum implementation finding tools [16,17] do not work with 5-bit S-boxes.

$$
\begin{aligned}
&x_0 = x_0 \oplus x_4, \quad x_4 = x_4 \oplus x_3, \quad x_2 = x_2 \oplus x_1, \\
&t_0 = x_0, \quad t_1 = x_1, \quad t_2 = x_2, \quad t_3 = x_3, \quad t_4 = x_4, \\
&t_0 = \sim t_0, \quad t_1 = \sim t_1, \quad t_2 = \sim t_2, \quad t_3 = \sim t_3, \quad t_4 = \sim t_4, \\
&t_0 = t_0 \cdot x_1, \; t_1 = t_1 \cdot x_2, \; t_2 = t_2 \cdot x_3, \; t_3 = t_3 \cdot x_4, \; t_4 = t_4 \cdot x_0, \\
&x_0 = x_0 \oplus t_1, \quad x_1 = x_1 \oplus t_2, \quad x_2 = x_2 \oplus t_3, \quad x_3 = x_3 \oplus t_4, \quad x_4 = x_4 \oplus t_0, \\
&x_1 = x_1 \oplus x_0, \quad x_0 = x_0 \oplus x_4, \quad x_3 = x_3 \oplus x_2, \quad x_2 = \sim x_2.
\end{aligned}
\tag{3}
$$

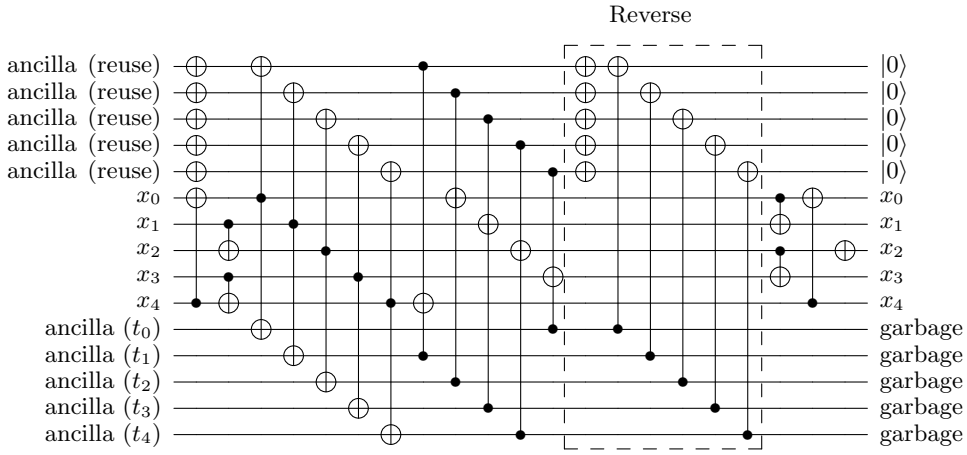


Fig. 2: ASCON S-box implementation in quantum (Toffoli depth 1).

It is evident that ancilla qubits $t_{0\sim4}$ are required for calculating intermediate values using combinations of AND and XOR operations. Therefore, we allocate 5 ancilla qubits to each S-box, and since there are 64 S-boxes operating in the substitution layer, a total of 320 (5×64) ancilla qubits are allocated for each execution of the substitution layer. However, in this case, Toffoli gates are executed sequentially for the AND operation, increasing the Toffoli depth. To address this, we propose a shallow version of the ASCON S-box quantum circuit optimized with Toffoli depth one.

Figure 2 illustrates our proposed quantum circuit for the ASCON S-box. Our approach involves allocating an additional set of ancilla qubits and continuously reusing them through reverse operations. By having an extra ancilla set, we can independently prepare operands for Toffoli gates. As depicted in Figure 2, all Toffoli gates operate in parallel, resulting in a Toffoli depth of one.

The Toffoli gate can be decomposed using various methods depending on specific objectives, such as minimizing T-depth or qubit count. In our implementation, we follow a method described in [6] where the Toffoli gate is decomposed into 8 Clifford gates followed by 7 T gates, resulting in a T-depth of 4 and a full depth of 8. As expected, due to the optimized Toffoli depth, the T-depth and full depth of our ASCON quantum circuit are also optimized.

To enable parallel operation of the S-boxes in the substitution layer, we need to allocate an equal number of qubits (i.e., 320 qubits) for the additional ancilla set, just as we allocated 320 qubits for $t_{0\sim4}$. The number of qubits is a crucial metric for optimizing quantum circuits. Taking this into account, we effectively address the increased overhead of qubit count by reusing the ancilla qubits used for parallel Toffoli gate operations through reverse operations. We will be further explained in the next section.

3.2 Reusing Ancilla Set with Reverse Operation

Thanks to the parallel implementation of Toffoli gates within the substitution layer, we can achieve a Toffoli depth of 1 (Section 3.1). However, this is accomplished by allocating an additional ancilla set of 320 ancilla qubits, resulting in a significant increase in qubit count. To address this issue, we allocate the ancilla set only once and reuse it throughout the process.

In this scenario, where we reuse the ancilla set, there is no need to allocate a new ancilla set for each execution of the substitution layer. Only the initial allocation of 320 ancilla qubits is required. To reuse the ancilla set, we perform reverse operations after the Toffoli gate operations (see Figure 2). During the reverse process, the number of X and CNOT gates increases. However, the depth does not increase because this reverse operation is performed simultaneously with the ongoing quantum gates from other operations.

In summary, by accepting the initial overhead of allocating an additional ancilla set and tolerating a slight increase in the number of quantum gates, we can achieve the benefits of reducing Toffoli and overall depth. Conceptually, our circuit architecture is similar to the regular version described in Jang et al.'s AES paper [9], which allocates sufficient ancilla qubits to parallelize S-box operations and reuse them with reverse operations. In their paper, Jang et al. propose a novel architecture called the shallow version, which offsets the depth overhead for reverse operations. However, in our case, there is no depth overhead for reverse operations, as explained earlier. Therefore, the new architecture for the shallow version is not suitable for the ASCON substitution layer.

3.3 Quantum Implementation of Linear Layer

The ASCON linear layer operates on the 320-bit state and is typically described with the 64-bit variables $x_{0\sim4}$ as given in Expression (4).

$$\begin{aligned}
x_0 &\leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28),\\
x_1 &\leftarrow \Sigma_1(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39),\\
x_2 &\leftarrow \Sigma_2(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6),\\
x_3 &\leftarrow \Sigma_3(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17),\\
x_4 &\leftarrow \Sigma_4(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41).
\end{aligned} \tag{4}$$

The ASCON linear layer can be considered as a series of operations over 32×32 binary matrices, where each of $x_{0\sim4}$ represents 64-qubit arrays, resulting in a 320×320 binary matrix. To implement a quantum circuit for linear operations, an in-place approach can be adopted. PLU-based quantum implementations of linear layers are presented in [7,8]. These implementations utilize a PLU factorization and do not require the allocation of ancilla or output qubits, resulting an in-place implementation. As a result, the number of qubits needed for the quantum circuit is reduced. However, due to the restricted computation space resulting from a small number of qubits, PLU-based implementations require sequential operations of CNOT gates, leading to an increase in the circuit

depth. This observation is reported in [18], where the PLU-based quantum circuit of the LowMC linear layer is compared with the naïve quantum implementation.

Additionally, a study in [19] explores different methods of implementing the ASCON linear layer and compares the results. It is observed that while the naïve implementation requires a higher qubit count, it provides a lower-depth quantum circuit. Therefore, based on our optimization goal, we choose to implement the quantum circuit of the linear layer with additional qubits.

To store the output of the linear layer, 320 ancilla qubits are allocated for each round (i.e., out-of-place) in our method. During the implementation process, we discovered that the depth of the quantum circuit can be affected by the order of the CNOT gates. Although quantum programming tools attempt to find the optimal path of quantum gates, they do not always provide the lowest depth. Considering this perspective, we arrange the order of CNOT gates to implement the quantum circuit of the ASCON linear layer with a quantum depth of 3. Table 1 provides a comparison of quantum resources for the ASCON linear layer, and our quantum implementation of linear layer has the lowest depth.

Table 1: Comparison of quantum resources required for ASCON linear layer.

Linear layer	Source	#CNOT	#Qubit	Depth
Out-of-place	This work	960	640	3
Naïve (binary matrix)	RBC'23 [19]	960	640	26
Gauss-Jordan	RBC'23 [19]	2,413	320	358
PLU	RBC'23 [19]	2,413	320	288
Modified [20]	RBC'23 [19]	1,595	320	119

3.4 Constructing ASCON AEAD Quantum Circuit

Algorithm 1 provides the implementation of the ASCON AEAD quantum circuit. The function $\mathsf{Permutation}^{\mathsf{a}}(S, ancilla)$ includes our quantum circuits for the substitution layer and linear layer. Note that a single set of ancilla qubits ($ancilla$ in Algorithm 1) is reused throughout the circuit until its completion, following the method described in Section 2.4. In the initialization, after the permutation operations, a bitwise XOR operation is performed between the 320-qubit S value and the 128-qubit key using CNOT gates (CNOT64 means CNOT gates operate on 64 qubits). To align the key qubits with S, padding with zeros is applied to the key value. At this point, since XOR with 0 is an identity operation, only the least significant 128 qubits (x_3 and x_4) need to be XORed. During both the associated data processing and plaintext processing, the input data are processed in blocks of 64 qubits. Therefore, by applying padding to the data, which involves adding a single 1 and the least number of 0s, the data can be divided into blocks of 64 qubits each. The XOR operation with 1 is equivalent to the NOT

Algorithm 1: Quantum circuit implementation of ASCON-128.

Input: $S = x_0||x_1||x_2||x_3||x_4$, pt, A , $ancilla$
Output: ct , T

1: $S \leftarrow \mathsf{Permutation}^{\mathsf{a}}(S, ancilla)$ ▷ Initialization
2: $x_3 \leftarrow \text{CNOT64}(key_0, x_3)$
3: $x_4 \leftarrow \text{CNOT64}(key_1, x_4)$

4: $x_0[32:64] \leftarrow \text{CNOT32}(A, x_0[32:64])$ ▷ Processing Associated Data

5: $x_0[31] \leftarrow \text{NOT}(x_0[31])$ ▷ $A||1||0^r - 1 - (|A| \pmod r)$ XORed with x_0

6: $S \leftarrow \mathsf{Permutation}^{\mathsf{b}}(S, ancilla)$

7: $x_4[0] \leftarrow \text{NOT}(x_4[0])$ ▷ Last bit of S XORed with 1

8: $x_0[32:64] \leftarrow \text{CNOT32}(pt, x_0[32:64])$ ▷ Processing Plaintext
9: $ct \leftarrow$ allocate new 32 qubits
10: $ct \leftarrow x_0[32:64]$

11: $x_0[31] \leftarrow \text{NOT}(x_0[31])$ ▷ $pt||1||0^{r-1-(|A| \pmod r)}$ XORed with x_0

12: $x_1 \leftarrow \text{CNOT64}(key_0, x_1)$ ▷ Finalization
13: $x_2 \leftarrow \text{CNOT64}(key_1, x_2)$

14: $S \leftarrow \mathsf{Permutation}^{\mathsf{a}}(S, ancilla)$

15: $x_3 \leftarrow \text{CNOT64}(key_0, x_3)$
16: $x_4 \leftarrow \text{CNOT64}(key_1, x_4)$

17: $T \leftarrow x_3||x_4$
18: **return** ct,T

operation, so we apply the NOT operation (i.e., X gate) to the corresponding qubit (corresponds to $x_0[31]$ in Algorithm 1).

4 Performance of Quantum Circuits

In this section, we present the performance analysis of our implemented ASCON-128 quantum circuit. We use the quantum programming tool ProjectQ to implement and simulate the quantum circuits. We verify the implementation using the `ClassicalSimulator` library in ProjectQ and analyze the quantum resources used with the `ResourceCounter`. Tables 2 and 3 show the resource requirements of our ASCON-128 quantum circuit implementation. The quantum resources presented in Table 2 are the result of an analysis conducted at the NCT (NOT, CNOT, Toffoli) level (enables intuitive comparison). In contrast, Table 3 represents the resource analysis conducted at the Clifford + T level. Note that the

sizes of the Associated Data (AD) and Plaintext (P) in the resource estimation are fixed at 32 bits, following the approach used in [21,22]. This implies that we also fix the sizes in the same manner for our paper.

Table 2: Required quantum resources for ASCON-128 quantum circuit implementation

Cipher	#X	#CNOT	#Toffoli	Toffoli depth	#Qubit	Depth	TD-M cost
ASCON-128	21,243	69,600	9,600	30	20,064	304	601,920

※: Associated data and plaintext are both of 32-bits.

Table 3: Required decomposed quantum resources for ASCON-128 quantum circuit implementation

Cipher	#Clifford	#T	T-depth	#Qubit	Full depth	FD-M cost
ASCON-128	167,643	67,200	120	20,064	513	10,292,832

※: Associated data and plaintext are both of 32-bits.

When comparing the results of Tables 2 and 3 with other quantum circuit implementations for ciphers [21,22,15,18], it can be observed that the proposed quantum circuit for ASCON-128 achieves a low Toffoli depth. However, our quantum circuit requires a high number of qubits, which is a result of the trade-off between qubit count and depth. In this trade-off, we report the TD-M and FD-M costs in Tables 2 and 3. The TD cost represents the Toffoli depth, FD represents the full depth, and M represents the qubit count. These metrics are used to evaluate the trade-off performance of quantum circuits. Although reducing depth is optimal under the MAXDEPTH constraint (as the parallelization of Grover instances has poor performance), we still find the TD-M or the FD-M cost useful for comparing the performance of quantum circuits themselves. Based on our proposed ASCON-128 quantum circuit, we estimate the cost of the Grover's key search and discuss the post-quantum security of ASCON.

5 Evaluation of Grover's Search Complexity

To estimate the cost of Grover's key search for ASCON-128, we follow the methodology outlined in Section 2.2. Grover's key search requires executing a large number of sequential iterations of the ASCON-128 quantum circuit. For each successive key recovery attempt for the cipher using a k-bit key, a set of oracle and diffusion operators should be iterated $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$ times. However, the overhead of the diffusion operator can be neglected compared to the oracle since

most of the quantum resources are used for implementing the target cipher in the quantum circuit. In many studies [9,8,7], the cost of iterations for the oracle is considered as the Grover's key search cost. Following this approach, we only count the quantum resources required for the iterations of the oracle to estimate the Grover's key search cost. The Grover oracle in our case consists of two sequential executions of the ASCON-128 quantum circuit, and the oracle is iterated $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$ times. To summarize, we estimate the Grover's key search cost for ASCON-128 as follows: Table 3 $\times 2 \times \lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$. Table 4 shows cost of the Grover's key search for ASCON-128.

Table 4: Cost of the Grover's key search for ASCON-128

Cipher	Total gates	Total depth	Cost (complexity)	#Qubit	TD-M cost	FD-M cost
ASCON-128	$1.180 \cdot 2^{83}$	$1.574 \cdot 2^{73}$	$1.857 \cdot 2^{156}$	20065	$1.799 \cdot 2^{83}$	$1.925 \cdot 2^{87}$

※: Associated data and plaintext are both of 32-bits.

6 Concluding Remarks (and Note on Quantum Security)

The cost of quantum attacks on ciphers can be used to assess the post-quantum security of a cipher. In this context, it is important to consider the post-quantum security criteria defined by NIST. In 2016, NIST established post-quantum security levels based on the estimated costs for AES-128, AES-192, and AES-256. However, as the costs of attacks against AES have decreased over time, NIST has recently adjusted the attack costs according to the security level (as outlined in Section 2.3).

According to Table 4, the quantum attack cost for ASCON is 1.857×2^{156}. Thus, ASCON can be evaluated as achieving post-quantum security Level 1, which corresponds to a cost equivalent to AES-128 (2^{157}) according to the recent standards (see Section 2.3 and [11]).

In conclusion, this paper presents the first optimized implementation of the ASCON-128 quantum circuit. We utilize multiple methodologies to minimize Toffoli and full depths while keeping the number of qubits at a reasonable level. By evaluating our depth-optimized ASCON-128 quantum circuit, we can confidently conclude that ASCON-128 achieves post-quantum security Level 1. Furthermore, the implementation techniques presented in this paper have the potential to be applied to other quantum circuit implementations of ciphers.

7 Acknowledgements

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2022R1A6A3A13062701, 25%) and this work was 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, 75%).

References

1. P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999. 2
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*, pp. 212–219, 1996. 2
3. C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1.2." Submission to NIST, 2019. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf. 2, 6
4. V. Srivastava, N. Gupta, A. Jati, A. Baksi, J. Breier, A. Chattopadhyay, S. K. Debnath, and X. Hou, "Ascon-sign." NIST PQC Additional Round 1 Candidates, 2023. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/Ascon-sign-spec-web.pdf. 2
5. D. Bhattacharjee and A. Chattopadhyay, "Depth-optimal quantum circuit placement for arbitrary topologies," *arXiv preprint arXiv:1703.08540*, 2017. 2
6. 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, p. 818–830, Jun 2013. 3, 7
7. M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying Grover's algorithm to AES: quantum resource estimates," in *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings 7*, pp. 29–43, Springer, 2016. 4, 5, 8, 12
8. S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing grover oracles for quantum key search on AES and LowMC," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 280–310, Springer, 2020. 4, 5, 8, 12
9. K. Jang, A. Baksi, H. Kim, G. Song, H. Seo, and A. Chattopadhyay, "Quantum analysis of AES," *Cryptology ePrint Archive*, 2022. 4, 5, 8, 12
10. 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. 4
11. NIST., "Call for additional digital signature schemes for the post-quantum cryptography standardization process," 2022. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf. 4, 5, 12
12. 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, 2020. 5

13. D. Lin, Z. Xiang, R. Xu, S. Zhang, and X. Zeng, "Optimized quantum implementation of AES," *Cryptology ePrint Archive*, 2023. 5
14. 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: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pp. 697–726, Springer, 2020. 5
15. Z. Huang and S. Sun, "Synthesizing quantum circuits of AES with lower t-depth and less qubits," in *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*, pp. 614–644, Springer, 2023. 5, 11
16. V. A. Dasu, A. Baksi, S. Sarkar, and A. Chattopadhyay, "LIGHTER-R: optimized reversible circuit implementation for sboxes," in *32nd IEEE International System-on-Chip Conference, SOCC 2019, Singapore, September 3-6, 2019*, pp. 260–265, 2019. 6
17. M. Chun, A. Baksi, and A. Chattopadhyay, "Dorcis: Depth optimized quantum implementation of substitution boxes." Cryptology ePrint Archive, Paper 2023/286, 2023. https://eprint.iacr.org/2023/286. 6
18. K. Jang, A. Baksi, H. Kim, H. Seo, and A. Chattopadhyay, "Improved quantum analysis of SPECK and LowMC," in *Progress in Cryptology–INDOCRYPT 2022: 23rd International Conference on Cryptology in India, Kolkata, India, December 11–14, 2022, Proceedings*, pp. 517–540, Springer, 2023. 9, 11
19. S. Roy, A. Baksi, and A. Chattopadhyay, "Quantum implementation of ASCON linear layer," *Cryptology ePrint Archive*, 2023. 9
20. Z. Xiang, X. Zeng, D. Lin, Z. Bao, and S. Zhang, "Optimizing implementations of linear layers," *IACR Transactions on Symmetric Cryptology*, pp. 120–145, 2020. 9
21. A. Baksi, K. Jang, G. Song, H. Seo, and Z. Xiang, "Quantum implementation and resource estimates for rectangle and knot," *Quantum Information Processing*, vol. 20, pp. 1–24, 2021. 11
22. R. Anand, S. Maitra, A. Maitra, C. S. Mukherjee, and S. Mukhopadhyay, "Resource estimation of Grovers-kind quantum cryptanalysis against FSR based symmetric ciphers," *Cryptology ePrint Archive*, 2020. 11