mathematics

IMPACT FACTOR 2.4

CITESCORE 3.5

Article

Depth-Optimized Quantum Circuits for ASCON: AEAD and HASH

Yujin Oh, Kyungbae Jang, Anubhab Baksi and Hwajeong Seo

Special Issue

Quantum Cryptography and Applications

Edited by

Dr. Chun-Wei Yang, Dr. Chia-Wei Tsai and Dr. Jason Lin

MDPI

https://doi.org/10.3390/math12091337

mathematics

MDPI

Article

Depth-Optimized Quantum Circuits for ASCON: AEAD and HASH †

Yujin Oh [1], Kyungbae Jang [1], Anubhab Baksi [2] and Hwajeong Seo [1,*]

[1] Division of IT Convergence Engineering, Hansung University, Seoul 02876, Republic of Korea; oyj0922@hansung.ac.kr (Y.O.); starj1234@hansung.ac.kr (K.J.)
[2] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 639539, Singapore; anubhab.baksi@ntu.edu.sg
* Correspondence: hwajeong@hansung.ac.kr; Tel.: +82-760-8033

† This paper is an extended version of our paper published in the 24th World Conference on Information Security Applications (WISA'23), Jeju Island, Republic of Korea, 23–25 August 2023; pp. 312–325.

Abstract: Quantum computing advancements pose security challenges for cryptography. Specifically, Grover's search algorithm affects the reduction in the search complexity of symmetric-key encryption and hash functions. Recent efforts have been made to estimate the complexity of Grover's search and evaluate post-quantum security. In this paper, we propose a depth-optimized quantum circuit implementation for ASCON, including both symmetric-key encryption and hashing algorithms, as a part of the lightweight cryptography standardization by NIST (National Institute of Standards and Technology). As far as we know, this is the first implementation of a quantum circuit for the ASCON AEAD (Authenticated Encryption with Associated Data) scheme, which is a symmetric-key algorithm. Also, our quantum circuit implementation of the ASCON-HASH achieves a reduction of more than 88.9% in the Toffoli depth and more than 80.5% in the full depth compared to the previous work. As per our understanding, the most effective strategy against Grover's search involves minimizing the depth of the quantum circuit for the target cipher. We showcase the optimal Grover's search cost for ASCON and introduce a proposed quantum circuit optimized for depth. Furthermore, we utilize the estimated cost to evaluate post-quantum security strength of ASCON, employing the relevant evaluation criteria and the latest advancements in research.

Keywords: quantum computer; ASCON; Grover's search algorithm; post-quantum security

MSC: 94A60; 81P94

Citation: Oh, Y.; Jang, K.; Baksi, A.; Seo, H. Depth-Optimized Quantum Circuits for ASCON: AEAD and HASH. *Mathematics* **2024**, *12*, 1337. https://doi.org/10.3390/math12091337

Academic Editor: Jonathan Blackledge

Received: 26 March 2024
Revised: 20 April 2024
Accepted: 26 April 2024
Published: 27 April 2024

1. Introduction

The emergence of quantum computers offers rapid computational speed and enhanced processing capabilities. These advantages open up possibilities for effectively addressing cryptographic problems by leveraging the properties of quantum states. However, simultaneously, the advent of quantum computers presents a potential threat to existing security systems, necessitating a reevaluation of security in the field of cryptography. One prominent initiative in this regard is the NIST Post-Quantum Cryptography (PQC) standardization process (https://csrc.nist.gov/projects/post-quantum-cryptography, accessed on 25 March 2024), which aims to address the need for quantum-resistant cryptography. This need arises from the aspect that the Shor's algorithm [1] can efficiently solve factorization and discrete logarithm problems.

Another significant quantum algorithm pertinent to cryptography is Grover's algorithm [2]. It offers the capability to accelerate data searches, thereby reducing the complexity associated with exhaustive searches in symmetric-key cryptography. However, while Grover's algorithm undeniably diminishes security strength, implementing the quantum circuit required for the attack entails a significant size.

Quantum attacks can be assessed from two primary perspectives, the computational power to solve cryptographic problems and the scale of the quantum circuits needed to execute those solutions. An alternative viewpoint suggests that the security of a cryptographic algorithm can be appraised differently depending on the size of the quantum circuit necessary for a quantum attack. This aspect is addressed in the NIST Post-Quantum Cryptography documentation, where post-quantum security strength is evaluated by considering the quantum cost required for potential quantum attacks (Section 2.3). NIST establishes post-quantum security strength by estimating the cost of Grover's attack against AES-128, -192, and -256 (similar in concept to how the security parameters of PQC algorithms are associated with the AES family). The cost of a Grover attack depends on the efficiency of implementing the quantum circuit for the targeted cryptographic algorithm.

This paper introduces an optimized quantum circuit for the AEAD and hash function scheme of ASCON [3], which has been chosen as the winner of the NIST Lightweight Cryptography standardization (https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon, accessed on 25 March 2024). Our primary focus is on reducing the depth of the ASCON quantum circuit while maintaining a reasonable number of qubits, which is in line with the principles of Grover's algorithm.

The depth of the quantum circuit directly affects the execution time of the circuits [4]. Although Grover's algorithm accelerates search speed by the square root, it still demands a considerable number of iterations within the quantum circuit. In essence, Grover's exhaustive key search is a time-intensive process, and NIST considers this aspect in security evaluations. To our understanding, minimizing the depth for symmetric-key ciphers represents the optimal strategy for Grover's algorithm (further elaborated in Section 2.3), which has consequently become the guiding principle for implementing the ASCON quantum circuit. Based on the proposed ASCON quantum circuit, we estimate the cost of a Grover attack and assess the post-quantum security strength of ASCON in accordance with the documentation by NIST.

1.1. Our Contribution

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

1. **Quantum Circuit Implementation of ASCON.** We demonstrate the first implementation of a quantum circuit for ASCON AEAD. Additionally, we improve an optimized quantum circuit for the hash function of ASCON and compare it with previous work;
2. **Depth Optimization of ASCON.** In our implementation of the ASCON quantum circuit, our primary focus is on achieving a low Toffoli depth and full depth. We demonstrate how to decrease these depths using various methods (parallelization, AND gate). Moreover, to ensure a reasonable qubit count, we adopt the method of reusing ancilla qubits;
3. **Post-quantum Security Assessment of ASCON.** We evaluate the quantum security of ASCON by estimating the cost of Grover's key search using our implemented quantum circuit for ASCON. This assessment includes comparing the estimated cost of Grover's search for ASCON with the security levels defined by NIST.

The relevant source codes for our work can publicly accessed (https://github.com/yudini/ASCON_quantum, accessed on 25 March 2024).

1.2. Extension from WISA'23 (Oh et al.)

This current work is indeed a substantially enlarged version of our previous paper presented at WISA 2023. For reference, the WISA paper can be found at [5].

In the current version of the paper, we explore a more optimized quantum circuit than the one presented in [5]. Additionally, this time, we extend our research to include the quantum analysis of ASCON AEAD and ASCON-HASH, thereby expanding the scope.

1.3. Paper Organization

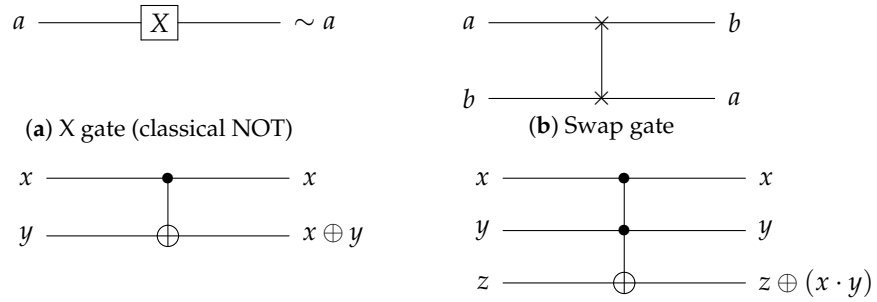
The overall structure of this paper is as follows: First, we provide the background knowledge necessary for the research in Section 2. Next, in Section 3, we describe the main method for our proposed ASCON quantum circuit. We explain the depth-optimized implementation of substitution (S-boxes) and the linear layers, which are the main components of ASCON. Then, we describe the overall implementation of ASCON AEAD and HASH. Sections 4 and 5 then present the quantum resources required for the proposed quantum circuit implementation, and use them to estimate the cost of Grover's attack. We also evaluate the post-quantum security of ASCON. Finally, in Section 6, we conclude our study.

2. Preliminaries

2.1. Quantum Gates

Figure 1 shows the gates most commonly utilized for implementing cryptography in quantum circuits (note that this list is not exhaustive and does not encompass all potential gates applicable for this purpose). Figure 1a illustrates the quantum X gate, which replaces the classical NOT operation by reversing the qubit state. Figure 1b showcases the Swap gate, exchanging the states of two qubits. The CNOT gate depicted in Figure 1c, replaces similarly to the classical XOR operation. It utilizes one control qubit to determine the value of the target qubit. Figure 1d depicts the quantum Toffoli gate, which serves as an alternative to the classical AND operation, and utilizes two control qubits to determine the target qubit's value.

In short, the X, CNOT, and Toffoli gates correspond to classical NOT, XOR, and AND operations, respectively. It is important to note that the Toffoli gate is implemented using various (decomposition-level) quantum gates, including the T, CNOT, X, and H gates, among others. Hence, it is essential to minimize the cost metrics associated with the constituents of the Toffoli gate when optimizing quantum circuits.



(**a**) X gate (classical NOT)

(**b**) Swap gate

(**c**) CNOT gate (classical XOR)

(**d**) Toffoli gate (classical AND)

Figure 1. Common (top level) quantum gates.

2.2. Search Using Grover's Algorithm

The significance of the Grover algorithm lies in its ability to efficiently tackle cryptographic decryption and search problems. By leveraging this algorithm, issues such as key search and database search can be efficiently addressed. For an encryption algorithm that uses a k-bit key, a classical computer has a search complexity of $O(2^k)$. On the other hand, in Grover's key search, a quantum computer has a reduced (by square root) complexity of $O(\sqrt{2^k})$. We describe the process of Grover's key search in three steps as follows. In the case of collision search for hash functions, only the search target is changed from a key to a collision pair.

1. Hadamard gates are applied to a k-qubit key, which result in a superposition state $|\psi\rangle$. The key has equal amplitudes for 2^k 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 encryption algorithm is implemented using quantum gates. The implemented quantum circuit performs encryption with a prepared key that represents 2^k states. The generated ciphertext is also in a superposition state, representing 2^k ciphertexts. This ciphertext in a superposition state is checked against the known ciphertext. If a match is found, the sign of the key in a superposition state is changed to negative. This is how the solution is returned in the Oracle. Note that the implemented quantum circuit operates in reverse by transforming the generated ciphertext (in a superposition state) back into the known plaintext;

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

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

3. The diffusion operator amplifies the amplitude (i.e., probability) of the solution key returned by the Oracle. Recall that the Oracle returns the solution key by changing the sign. The implementation method for the diffusion operator is standard and can be easily implemented. As the complexity of the diffusion operator is trivial compared to that of the Oracle, it is typically neglected in the cost estimation for Grover's key search [6–8]. Grover's algorithm iterates a numerous number of operations of the Oracle and diffusion (i.e., around $\sqrt{2^k}$ times) to amplify the amplitude of the solution key.

2.3. NIST Security Criteria

NIST provides security levels for post-quantum security against quantum attacks [9,10], and we refer to these for evaluating our implementation. NIST estimates the complexity of Grover's key search and collision search on the AES and SHA-2/3 families, respectively. Levels 1, 3, and 5 correspond to the complexity of Grover's key search for AES, while levels 2 and 4 correspond to the complexity of collision search for SHA-2/3.

- **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 2**: 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 collision search on a 256-bit hash function, such as SHA-256/SHA3-256;
- **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 4**: 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 collision search on a 384-bit hash function, such as SHA-384/SHA3-384;
- **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 ($2^{298} \rightarrow 2^{285}$).

Unsurprisingly, the Grover algorithm is one of the prominent quantum attacks on symmetric-key ciphers, and NIST also considers this aspect. The difficulty of attacks at Levels 1, 3, and 5 depends on the cost of Grover's key search applied to AES-128, 192, and 256, respectively. This cost is determined by multiplying the total gate count by the depth of Grover's key search circuit. NIST provides estimates for Levels 1, 3, and 5 as 2^{170}, 2^{233}, and 2^{298}, respectively, based on the quantum circuit implementation of AES by Grassl et al. Recently, NIST adjusted the costs of Grover's key search on the AES family, as reported in [10]. Over the past few years, various efforts have been made to optimize the quantum circuits of AES. For instance, Jaques et al. introduced depth-optimized quantum circuits for

AES at Eurocrypt'20, resulting in a decreased cost of Grover's key search on AES [7]. NIST has now defined new quantum attack costs for AES-128, 192, and 256 based on the findings from [7], yielding costs of 2^{157}, 2^{221}, and 2^{285}, respectively. It is worth noting that despite the reported programming-related issues in their quantum circuit implementation, Jang et al. address these concerns in [8], and demonstrates how to implemented optimized AES quantum circuits. As of now, the most up-to-date on AES results are documented in [8], to the best of our finding.

Moreover, we need to consider NIST-defined MAXDEPTH, which denotes the maximum circuit depth feasible for execution on a quantum computer. NIST categorizes the depth constraints of quantum attacks, represented by MAXDEPTH, into the following intervals: $[2^{40}, 2^{64}]$ and $[2^{64}, 2^{96}]$, as it acknowledges that the considerable depth of Grover's key search, resulting from numerous sequential iterations, renders the attack practically challenging.

Given this consideration, one would anticipate that the depth of the quantum circuit for Grover's search does not exceed 2^{96} (the highest estimated bound for MAXDEPTH (since Grover's search increases the circuit depth beyond $2^{k/2}$ for a k-bit key (where the quantum depth for cipher implementation $\times \lfloor \frac{\pi}{4} 2^{k/2} \rfloor$ is required for Grover's iteration), the quantum depth is necessarily greater than the two smaller MAXDEPTH values for AES variants)). If it turns out that the depth restriction exceeds the specified limit, the parallelization of Grover's search can be considered [11].

For the parallelization of Grover's algorithm, the trade-off metrics for quantum circuits change by multiplying circuit depth. In short, the product of qubit count and circuit depth is replaced with the product of qubit count and squared depth. Throughout this paper, we denote qubit count, full depth, Toffoli depth, and T-depth as M, FD, TD, and Td, respectively. For the evaluation of quantum circuits, we also estimate the changed trade-off metrics for Grover's parallelization as FD^2-M, TD^2-M, and Td^2-M.

2.4. ASCON

ASCON is a lightweight cryptographic algorithm standardized in the NIST Lightweight Cryptography standardization. ASCON comprises an authenticated encryption with associated data (AEAD) mode, a hash function, and a variant known as Ascon-80pq, designed to offer improved resistance against quantum key-search attacks. ASCON offers the following two AEAD modes: ASCON-128 and ASCON-128a. The encryption process in ASCON AEAD consists of *Initialization*, *Processing Associated Data*, *Processing Plaintext*, and *Finalization*. For a hash function, ASCON offers the following two modes: ASCON-HASH and ASCON-XoF. The encryption process in the hash function of ASCON consists of *Initialization*, *Absorb Message*, and *Squeeze Tag*.

The main components common to all ASCON schemes consist of two 320-bit permutations, each configured with different round counts (p^a and p^b). 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 the addition of constants, a substitution layer using a 5-bit S-box, and a linear layer using 64-bit diffusion functions.

3. Quantum Implementation of ASCON

In this section, we describe the quantum implementation of ASCON-128, which is a variant of AEAD, and ASCON-HASH, which is a hash function. Due to the same rate in ASCON-128 and ASCON-HASH (it means that they have the same data block size), when used together for both authenticated encryption and hashing, the two schemes can be efficiently combined. Aligning with our design philosophy, which prioritizes minimizing depth for optimal performance in Grover's algorithm, we focus on optimizing the depth of the ASCON-128 and ASCON-HASH quantum circuits while also ensuring a reasonable number of qubits.

3.1. Implementation (with Parallelization) of S-Box

While the lookup table method is a common choice for S-boxes implementations in classical computing (i.e., hardware and software implementations), the reversible nature of operations in quantum computing renders the use of lookup tables infeasible. Hence, it becomes evident that the implementation of S-box quantum circuits should rely on Boolean expressions (specifically, the coordinate functions) using quantum gates. In the quantum circuit of ASCON, the implementation of the S-box is notably resource-intensive. The 5-bit ASCON S-box can be realized by utilizing Boolean operations that involve NOT ($\sim$), AND ($\cdot$), and XOR ($\oplus$) gates, as follows (adopted from [3]).

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

In [12], the authors used the ancilla qubits allocated from the linear layer in order to reduce the number of qubits. In short, the substitution and linear layers share ancilla qubits. They partly use the result of the substitution layer in the linear layer. After utilizing the result, they reverse the operations previously performed in the substitution layer (to reuse the qubits). While this architecture can reduce the number of qubits, it increases the circuit depth due to the multiple reverse operations of high complexity. Unlike the previous approach, we implement the substitution and linear layers independently. The benefit achieved from this architecture is that it avoids performing reverse operations that involve intricate operations in the quantum circuit. Despite using more qubits than [12], we achieve the best trade-off performance in terms of time–space complexity by significantly reducing the circuit depth.

As shown in Expression (3), we can see that x_i words are intertwined to perform AND and XOR operations, which are referred to as Toffoli operations in quantum computing. Therefore, we need additional ancilla qubits to store these resulting values. By using an 64×5 ancilla qubits, we can perform an S-box that uses the fewest qubits. However, in this scenario, Toffoli gates are executed in a sequential manner for the AND operation, resulting in an increased Toffoli depth. In response, we propose a shallow version of the ASCON S-box quantum circuit with a Toffoli depth of one.

Figure 2 depicts the proposed quantum circuit for the ASCON S-box. Through reverse operations, we can reuse one ancilla qubit set. We optimize the Toffoli depth to one by processing all Toffoli operations in parallel. In order to facilitate the parallel operations of S-boxes in the substitution layer, it is essential to allocate the same quantity of qubits (i.e., 320 qubits) to an additional ancilla set as we did for $t_{0\sim4}$. Having an extra ancilla set allows the operands for all Toffoli gates to be independently prepared. Thus, as depicted in Figure 2, all the Toffoli gates operate in parallel, leading to a Toffoli depth of one. However, our main focus is on minimizing the depth while simultaneously aiming to decrease the number of qubits. We address the increased overhead of the qubit count effectively in the next section.

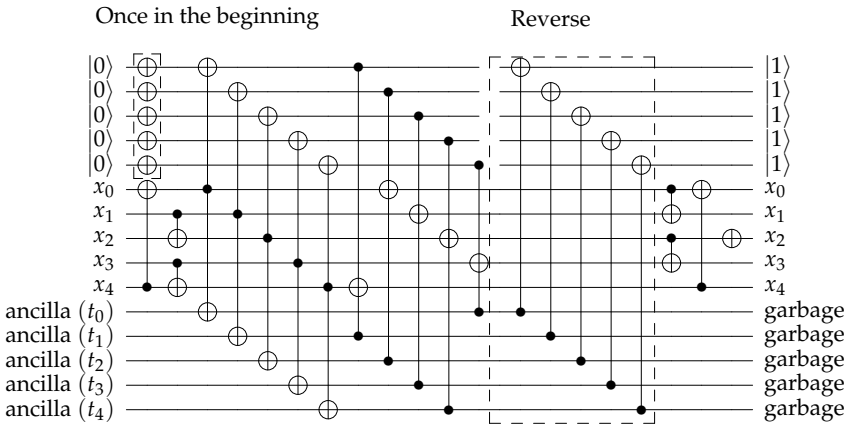


Figure 2. ASCON S-box quantum circuit with Toffoli depth of one using two sets of ancilla and reverse operations.

3.2. Reusing Ancilla Set with Reverse Operation

As a result of parallelizing the Toffoli gates within the substitution layer, we achieve a Toffoli depth of one (Section 3.1). However, allocating an ancilla qubit set for each round leads to a significant overhead in terms of the number of qubits. The number of qubits, in addition to depth, is also a crucial metric for optimizing a quantum circuit. For this purpose, we allocate the ancilla set once initially and subsequently reuse it throughout the entire process.

In this instance, because we reuse the ancilla set, there is no need to allocate a new ancilla set for each round of the substitution layer. Only the initial allocation of 320 ancilla qubits is required. To reuse the ancilla set, we perform the reverse operations after the Toffoli gate operations (see Figure 2). Throughout the reverse process, there is an increase in the number of CNOT gates. Nevertheless, the depth remains unaffected since this reverse operation is conducted simultaneously with the ongoing quantum gates from other operations. Additionally, we omit the X gate operation from the reverse operation. Instead of initializing the ancilla qubits to $|0\rangle$, we leave the ancilla qubits in the flipped state (i.e., $|1\rangle$) by skipping the X gate operation. This approach avoids the need for an X gate operation in the next round, resulting in fewer gates. Specifically, by applying the NOT operation only once in the initial round and omitting it in the reverse operations, the subsequent rounds no longer require the NOT operation. As a result, we utilize 640 (=320 $\times$ 2) ancilla qubits in a single substitution layer, and 320 ancilla qubits are reused without an additional X gate. Figure 2 shows our quantum circuit for the ASCON S-box.

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 the Toffoli and the overall depth. Table 1 shows the comparison of the quantum resources required for ASCON with a previous work [12]. As mentioned earlier, unlike our separated non-linear (i.e., substitution) and linear layer implementation, the work by Lee et al. [12] includes the resources for performing both substitution and linear operations combined for comparison, as it adopts an interconnected structure (the description of the linear implementation continues in Section 3.4).

In Table 1, we also investigate the ASCON implementations by Stoffelen [13] and Lu et al. [14] in the context of classical computing, and port to quantum circuits. The corresponding results can be seen from Table 1. Furthermore, we could not verify the ASCON implementations given in [15] (also in the context of classical computing), so those are not included here.

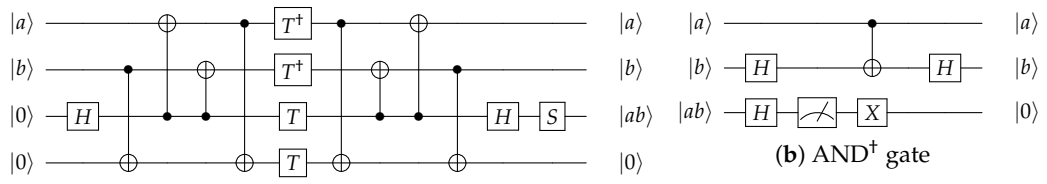
Table 1. Comparison of quantum resources required for ASCON permutation (one round).

Operation	Source	#CNOT	#1qCliff	#*T*	Toffoli Depth	#Qubit (Reuse)	Full Depth
Substitution	Stoffelen [13]	4608	918	2240	2	1600	24
	(ASCON) [3]	3264	1174	2240	1	960 (320)	15
	Luo$^+$ [14]	16,640	7808	15,680	7	5760	94
Substitution + Linear	Lee$^+$ [12]	4544	2070	3584	8	640	80
	Ours	4224	1174	2240	1	1280 (320)	18
Linear layer	Ours	960	0	0	0	640	3

Now, there are quantum-specific tools that find the implementation of a given, such as [16,17]; however, neither of these work with 5-bit. Recently, two new quantum implementations of a related paper have appeared online, namely [18,19] (we were informed by the authors of [19] that their paper was under submission when [18] first came online). Unfortunately, Ref. [19] does not scale-up for a 5-bit (the authors of [19] stated that the solver did not return a solution when the number of logic gates ≥ 13 within a feasible time). It is not mentioned if the method of [18] works for the ASCON (it is not mentioned in their paper, also their publicly available source code does not seem to produce any usable result).

3.3. Optimized Implementation of T-Depth One

Various approaches exist for decomposing the Toffoli gate, depending on specific goals like minimizing the T-depth or qubit count. In our implementation, we adopt a technique outlined in [20], breaking down the Toffoli gate into eight Clifford gates followed by seven T gates. This results in a T-depth of four and an overall depth of eight. Additionally, we apply the AND gate method described in [7]. This approach functions similarly to the Toffoli gate, but requires the target qubit to be in a clean state. The AND gate is comprised of 11 Clifford gates, 4 T gates, and 1 ancilla qubit, resulting in a T-depth of 1 and a full depth of 8 (Figure 3a). The AND† gate, being the reverse of the AND gate and based on the Measurement gate, is constructed with five Clifford gates and one Measurement gate. It achieves a full depth of four, with a T-depth of zero (Figure 3b). The ancilla qubit used within the AND gates can be initialized and reused, requiring only a single allocation at the beginning. Consequently, to process all the AND gates in parallel, an initial allocation of 320 ($=5 \times 64$) ancilla qubits for use in the AND gates is necessary. However, we opt not to allocate an additional 320, but rather declare the ancilla qubits in advance for use in the linear layer. As a result, we do not require additional ancilla qubits for the implementation of the AND gates. Furthermore, since the reverse operation of the Toffoli gate is not employed in our implementation, we do not benefit from the resource efficiency offered by the AND† gate. However, we utilize the AND† gate in the Grover oracle (as detailed in Section 5).



(**a**) AND gate (**b**) AND† gate

Figure 3. Quantum AND and AND† gates.

3.4. Quantum Implementation of Linear Layer

The linear layer of ASCON is composed of 320-bits. It is split into five blocks (each of 64-bits) as, $x_0, \ldots, x_4$. The update is given by:

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

The ASCON linear layer can indeed be perceived as a sequence of operations on five 32×32 binary matrices (where each $x_0, \ldots, x_4$ represents a 64-qubit array), ultimately resulting in a 320×320 binary matrix. When considering the implementation of a quantum circuit for these linear operations, both out-of-place and in-place strategies are feasible (see [21]). We explore various methodologies, including the PLU factorization-based implementations detailed in [6,7], aimed at minimizing the depth for the quantum circuit. Although these approaches allow for an in-place implementation and reduce the requirement for ancilla or output qubits, they involve consecutive CNOT gate operations, thereby leading to an increase in the circuit depth.

Leveraging insights from prior studies, particularly [21], we assess the different implementation methodologies for the ASCON linear layer to evaluate their trade-offs in terms of qubit count and circuit depth. Our primary optimization objective focuses on achieving a low depth of circuit. To this end, we choose to enhance the quantum circuit of the linear layer with ancilla qubits, allocating 320 ancilla qubits for each round to facilitate out-of-place operations and store the output of the linear layer. Throughout the implementation process, we recognize the significant impact of CNOT gate sequencing on circuit depth and strategically arrange the sequence to achieve a quantum depth of three for the ASCON linear layer. A comprehensive comparison of quantum resources for the ASCON linear layer is presented in Table 2, with our quantum implementation of the linear layer utilizing 640 qubits (320 qubits are used to store the output), 960 CNOT gates, and achieving a quantum depth of three, which stands as the lowest depth attainable.

Table 2. Comparison of quantum resources required for ASCON linear layer.

Linear Layer	Source	#CNOT	#Qubit	Depth
Out-of-place	Ours	960	640	3
Naïve (binary matrix)	Roy et al. [21]	960	640	26
Gauss-Jordan	Roy et al. [21]	2,413	320	358
PLU	Roy et al. [21]	2,413	320	288
Modified [22]	Roy et al. [21]	1,595	320	119

3.5. Constructing ASCON AEAD Quantum Circuit

Our implementation of the ASCON AEAD quantum circuit is outlined in Algorithm 1. The $\mathsf{Permutation}^{\mathsf{a}}(S, ancilla)$ function encompasses constant addition, the substitution layer, and the implementation of the linear layer circuit, as described earlier. Across the entire circuit, an ancilla set (referred to as Ancilla in Algorithm 1) is reused in accordance with the approach detailed in Section 2.4.

During the initialization, the permutation operation and a bit-wise XOR operation between the 320-qubit S value and the 128-qubit key are performed. For these XOR operations, CNOT gates are used (CNOT64 indicates that the CNOT gates operate on 64 qubits). To align the key qubits with S (320-bit), padding the key value with zeros is performed. As XORing with 0 has no effect, only the least significant 128 qubits (x_3 and x_4) require XORed.

During both the processing of associated data and plaintext, input data are processed in blocks of 64 bits each, requiring padding to divide it into 64-bit qubit blocks. Padding includes adding a single `1` and the minimum number of `0`s. Performing an XOR operation with one results in the same outcome as applying the NOT operation. Hence, we execute the NOT operation, represented by the X gate, on the qubit identified as $x_0[31]$ in Algorithm 1.

Algorithm 1 Quantum circuit implementation of ASCON-128.

Input: $S = x_0||x_1||x_2||x_3||x_4$, pt, A , $key = key_0||key_1$ $ancilla$
Output: ct , T

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

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

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

$S \leftarrow \text{Permutation}^{b}(S, ancilla)$

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

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

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

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

$S \leftarrow \text{Permutation}^{a}(S, ancilla)$

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

$T \leftarrow x_3||x_4$
return ct,T

3.6. Constructing ASCON-HASH Quantum Circuit

Algorithm 2 shows the implementation of the ASCON-HASH quantum circuit. In ASCON-HASH, unlike ASCON AEAD, only the permutation p^a is used, not p^b. For efficiency, we implement the initialization of ASCON-HASH using only X gates, as the initial 320-bit state S can be pre-computed for each instance. Simply speaking, we set the quantum state of S using only X gates depending on the classical value of S. Conceptually, it is the same to XOR the round constant to the intermediate state using X gates.

Algorithm 2 Quantum circuit implementation of ASCON-HASH.

Input: $S = x_0||x_1||x_2||x_3||x_4$, $Message$, $ancilla$
Output: $Hash$

Initialization(S) ▷ Only X gates are used
$m\_len = \lceil Message\ length/64 \rceil$
$h\_len = \lceil Hash\ length/64 \rceil$

for $0 \leq i \leq m\_len$ **do** ▷ Absorbing
    $x_0 \leftarrow \text{CNOT64}(Message[256 - (64 * (i+1)], x_0)$
    $S \leftarrow \text{Permutation}^{a}(S, ancilla)$

for $0 \leq i \leq h\_len - 1$ **do** ▷ Squeezing
    $Hash[64 \cdot i : 64 \cdot i + 63] \leftarrow \text{CNOT64}(x_0, Hash[64 \cdot i : 64 \cdot i + 63])$
    $S \leftarrow \text{Permutation}^{a}(S, ancilla)$
return $Hash$

In Absorbing, ASCON-HASH processes the message in blocks of 64 bits. To ensure that the length of the message becomes a multiple of 64 bits, padding is applied to the message by appending a single `1` and the minimum number of `0`s. Each message block of 64 bits is processed by XORing it with the first 64-bit block of the state S (i.e., x_0), followed by the application of the Permutation^a to the state S.

The processing of squeezing generates the 256-bit hash value. The hash value (*Hash*) is copied from the state of the 64-bit block x_0 until it reaches a total length of 256 bits (line 8 of Algorithm 2). After each extraction, the internal state S is transformed by Permutation^a.

4. Performance of Quantum Circuits

In this section, we provide a summary to estimate the resource of our implemented ASCON-128 and ASCON-HASH quantum circuits. We employ the quantum programming tool ProjectQ for both implementing and simulating the quantum circuits. The correctness of the implementation is confirmed by validating it with the `ClassicalSimulator` library in ProjectQ, while the usage of the quantum resources is assessed through analysis with the `ResourceCounter`.

Table 3 (in [23], a quantum circuit implementation for ASCON is presented but is not included in this table, as it was difficult to compare their implementation approach and the required quantum resources) shows the resource requirements for our ASCON quantum circuit. The resource requirements for the implementation of the ASCON-HASH quantum circuit are compared to the previous work [12] (the reported depths and costs from [12] are estimated before the decomposition of the Toffoli gates. For a fair comparison, we decompose the Toffoli gates used in their implementation/code and re-estimate the full depth and costs). The quantum resources presented in Table 3 are analyzed based on the decomposition of the Toffoli gate into Clifford + T levels (8 Clifford + 7 T gates, T-depth 4, and full depth 8).

The resource estimation maintains a fixed size of 32 bits for both the associated data (AD) and plaintext (P), aligning with the methodology described in [24,25]. As such, our paper also follows the same approach by keeping the size consistent. In the same context, the input message length for the resource estimation of ASCON-HASH is fixed at 256 bits, as in the previous work [12].

Table 3. Quantum resources required for implementations of ASCON.

Cipher	Source	#CNOT	#1qCliff	#T	Toffoli Depth (*TD*)	#Qubit (*M*)	Full Depth (*FD*)	*TD*-*M*	*FD*-*M*	TD^2-*M*	FD^2-*M*
ASCON-128	Ours	127,200	40,443	67,220	30	20,064	513	1.15×2^{19}	1.23×2^{23}	1.08×2^{24}	1.23×2^{32}
ASCON-HASH	L^+ [12]	491,008	208,018	387,072	864	35,136	8427	1.81×2^{24}	1.10×2^{28}	1.53×2^{34}	1.13×2^{41}
	Ours	406,016	68,435	215,040	96	62,592	1641	1.43×2^{22}	1.53×2^{26}	1.07×2^{29}	1.23×2^{37}

Comparing the results of ASCON AEAD presented in Table 3 with other quantum circuit implementations for ciphers [24–26], it becomes apparent that the devised quantum circuit for ASCON-128 demonstrates a significantly reduced Toffoli depth. Also, our implementation of ASCON-HASH provide improved results in terms of the Toffoli depth and the full depth compared to the previous work [12].

However, our implementation achieves a low depth at the cost of requiring a high number of qubits (there is a trade-off between the two). For this trade-off, we report the *TD*-*M*, *FD*-*M*, TD^2-*M*, and FD^2-*M* costs in Table 3. The *TD* cost denotes the Toffoli depth, *FD* denotes the full depth, and *M* denotes the qubit count. These metrics are commonly utilized to assess the trade-off performance of quantum circuits [7,8,26,27]. When using these metrics to compare our implementation with [12], our implementation provides a better performance. Using these estimated quantum resources, we approximate the cost of Grover's key/collision search for ASCON and explore the post-quantum security of ASCON.

5. Evaluation of Grover's Search Complexity

To estimate the cost of Grover's search for ASCON, we follow the methodology outlined in Section 2.2. In the Grover oracle, it is composed of the sequential execution of both the ASCON quantum circuit and its reverse circuit. In this scenario, the AND† gate can be utilized in the reverse circuit. According to the estimated costs of the Grover's oracle outlined in Table 4, it is evident that utilizing the AND gate results in lower resource costs across all aspects (without increasing the qubit count).

Grover's search requires executing a large number of sequential iterations of the ASCON quantum circuit. For each successive key recovery attempt for the cipher using a k-bit key/input, a set of oracle and diffusion operators should be iterated $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$ times. For a hash function with an ℓ-bit output, the BHT algorithm in [28] indicates that the complexity associated with finding collisions using a Grover's circuit is assumed to be $2^{\ell/3}$. Thus, a series of oracle and diffusion operators for a hash function should be iterated $\lfloor \frac{\pi}{4}2^{\ell/3} \rfloor$ times. However, the diffusion operator's overhead can be disregarded compared to the oracle, as the majority of the quantum resources are allocated for implementing the target cipher within the quantum circuit. For this reason, Grover's search cost is often considered as the cost of iteration for the oracle in many studies [6–8]. Taking this approach, we exclusively focus on the quantum resources essential for the iterations of the oracle to estimate the cost of Grover's search algorithm. In summary, we estimate the cost of Grover's search for ASCON-128 and ASCON-HASH as follows: Table 4 $\times$ $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$ and Table 4 $\times$ $\lfloor \frac{\pi}{4}2^{\ell/3} \rfloor$, respectively. Tables 5 and 6 show the costs for ASCON-128 and ASCON-HASH using Grover's search algorithm. According to the documents of NIST, we report the G-FD cost and also provide FD-M, Td-M, FD^2-M, and Td^2-M metrics for a trade-off between the number of qubits and the circuit depth. According to the NIST about the MAXDEPTH constraint, the metrics related to circuit depth, including the Toffoli depth, the T-depth, and the full depth, are highlighted as crucial factors. In this regard, our depth-optimized implementation provide the optimal performance in these metrics.

Table 4. Decomposed quantum resources for Grover's oracle on ASCON.

Cipher	Source	#CNOT	#1qCliff	#T	#Measure	T-Depth (Td)	#Qubit (M)	Full Depth (FD)	Td-M	FD-M	Td^2-M	FD^2-M
ASCON-128	Ours	254,400	80,886	134,440	0	240	20,065	1,026	1.15×2^{22}	1.23×2^{24}	1.08×2^{30}	1.23×2^{34}
	Ours-AND	225,600	71,926	38,400	9600	30	20,065	816	1.15×2^{19}	1.95×2^{23}	1.08×2^{24}	1.56×2^{33}
ASCON-HASH	L$^+$ [12]	982,016	416,036	774,144	0	6,912	35,137	16,854	1.81×2^{25}	1.10×2^{29}	1.53×2^{36}	1.13×2^{43}
	Ours	812,032	136,870	430,080	0	768	62,593	3,282	1.43×2^{25}	1.53×2^{27}	1.07×2^{35}	1.23×2^{39}
	Ours-AND	719,872	229,030	122,880	30,720	96	62,593	2,608	1.43×2^{22}	1.22×2^{27}	1.07×2^{29}	1.55×2^{38}

Note that for the entire key space, Grover's search is optimal, but in practical scenarios, cryptanalysis techniques are often employed to reduce or partition the search space. However, Grover's search can also be applied to these reduced search spaces, as they partly require exhaustive search.

Table 5. Cost of Grover's key search for ASCON-128 (ours).

Cipher	Source	#Gate (G)	Full Depth (FD)	T-Depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
ASCON-128	Ours	1.42×2^{82}	1.57×2^{73}	1.47×2^{71}	1.22×2^{14}	1.12×2^{156}	1.92×2^{87}	1.79×2^{85}	1.50×2^{161}	1.32×2^{157}
	Ours-AND	1.01×2^{82}	1.25×2^{73}	1.44×2^{68}	1.22×2^{14}	1.26×2^{155}	1.53×2^{87}	1.76×2^{82}	1.90×2^{160}	1.27×2^{151}

Table 6. Costs of Grover's collision search for ASCON-HASH.

Cipher	Source	#Gate (G)	Full Depth (FD)	T-Depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
ASCON-HASH	L$^+$ [12]	1.02×2^{106}	1.01×2^{99}	1.66×2^{97}	1.07×2^{15}	1.04×2^{205}	1.09×2^{114}	1.79×2^{112}	1.11×2^{213}	1.49×2^{210}
	Ours	1.30×2^{105}	1.58×2^{96}	1.48×2^{94}	1.91×2^{15}	1.03×2^{202}	1.51×2^{112}	1.41×2^{110}	1.20×2^{209}	1.05×2^{205}
	Ours-AND	1.04×2^{105}	1.25×2^{96}	1.47×2^{91}	1.91×2^{15}	1.31×2^{201}	1.20×2^{112}	1.41×2^{107}	1.51×2^{208}	1.04×2^{199}

6. Conclusions

The analysis of a quantum attack's costs on ciphers provides a means to evaluate the post-quantum security of a cipher. In this context, it becomes crucial to take into account the post-quantum security criteria established by NIST. In 2016, NIST introduced the post-quantum security levels (level 1, 3, and 5) determined by the anticipated costs for breaking AES. However, with the diminishing costs of attacks against AES, NIST has revised the attack cost metrics to align with the security levels, as discussed in Section 2.3. Based on the information provided in Table 5, the most optimized quantum attack cost for ASCON-128 is 1.26×2^{155}. Therefore, based on current standards, ASCON-128 falls short of achieving post-quantum security level 1, equivalent to the cost of breaking AES-128 (2^{157}). On the other hand, NIST does not assign specific costs for Levels 2 and 4, which relate to the costs associated with SHA2/3-256 and SHA2/3-384. Therefore, we focus on the comparison of costs in the previous paper [12]. Ultimately, our implementation demonstrates a higher level of optimization in terms of the NIST-provided metric, *G-FD*, compared to [12]. Additionally, both ASCON-128 and ASCON-HASH show more optimized metrics for the AND gate version.

In summary, this paper presents the first implementation of the ASCON-128 quantum circuit and the optimized implementation of the ASCON-HASH compared it with previous work. We employ various techniques to minimize the Toffoli and full depths while ensuring a reasonable qubit count. Our depth-optimized ASCON-128 quantum circuit fails to achieve post-quantum security level 1. Furthermore, our quantum circuit implementation of the ASCON-HASH achieves the full depth improvement of over 80.5% and the Toffoli depth by more than 88.9% compared to the implementation proposed in [12].

In [29], the authors of ASCON anticipated that ASCON, being designed for lightweight applications, does not claim resistance against all possible quantum attacks. Therefore, an extended version called ASCON-80pq with a 160-bit longer key length was proposed. Building on this, we will implement quantum circuits for ASCON-80pq and evaluate its security level in future work. Lastly, the implementation techniques proposed in this paper are expected to be applicable to quantum circuit implementations of other cipher systems.

Author Contributions: Software, Y.O., K.J., and A.B.; Writing—original draft, Y.O.; Writing—review & editing, K.J. and A.B.; Supervision, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by Hansung University for Hwajeong Seo. This project is partially supported by the Wallenberg-NTU Presidential Post-doctorate Fellowship for Anubhab Baksi.

Acknowledgments: This is a significantly improved version (with new results and discussions) of the paper accepted in WISA 2023 (https://wisa.or.kr/accepted, accessed on 25 March 2024) (also available in [5]).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
2. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
3. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. Ascon v1.2. Submission to NIST. 2019. Available online: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf (accessed on 25 March 2024).
4. Bhattacharjee, D.; Chattopadhyay, A. Depth-optimal quantum circuit placement for arbitrary topologies. *arXiv* **2017**, arXiv:1703.08540.
5. Oh, Y.; Jang, K.; Baksi, A.; Seo, H. Depth-Optimized Implementation of ASCON Quantum Circuit. Cryptology ePrint Archive, Paper 2023/1030. 2023. Available online: https://eprint.iacr.org/2023/1030 (accessed on 25 March 2024).

6. Grassl, M.; Langenberg, B.; Roetteler, M.; Steinwandt, R. Applying Grover's Algorithm to AES: Quantum Resource Estimates. In *Post-Quantum Cryptography*; Takagi, T., Ed.; Springer: Cham, Switzerland, 2016; pp. 29–43.
7. Jaques, S.; Naehrig, M.; Roetteler, M.; Virdia, F. Implementing Grover Oracles for Quantum Key Search on AES and LowMC. In *Advances in Cryptology—EUROCRYPT 2020, Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 10–14 May 2020*; Canteaut, A., Ishai, Y., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Part II, Volume 12106, pp. 280–310. [CrossRef]
8. Jang, K.; Baksi, A.; Kim, H.; Song, G.; Seo, H.; Chattopadhyay, A. Quantum Analysis of AES. Cryptology ePrint Archive, Paper 2022/683. 2022. Available online: https://eprint.iacr.org/2022/683 (accessed on 25 March 2024).
9. NIST. Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. 2016. Available online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf (accessed on 25 March 2024).
10. NIST. Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process. 2022. Available online: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf (accessed on 25 March 2024).
11. Kim, P.; Han, D.; Jeong, K.C. Time–space complexity of quantum search algorithms in symmetric cryptanalysis: Applying to AES and SHA-2. *Quantum Inf. Process.* **2018**, *17*, 339. [CrossRef]
12. Lee, W.K.; Jang, K.; Song, G.; Kim, H.; Hwang, S.O.; Seo, H. Efficient implementation of lightweight hash functions on gpu and quantum computers for iot applications. *IEEE Access* **2022**, *10*, 59661–59674. [CrossRef]
13. Stoffelen, K. Optimizing s-box implementations for several criteria using SAT solvers. In Proceedings of the International Conference on Fast Software Encryption, Bochum, Germany, 20–23 March 2016; pp. 140–160.
14. Lu, Z.; Wang, W.; Hu, K.; Fan, Y.; Wu, L.; Wang, M. Pushing the limits: Searching for implementations with the smallest area for lightweight s-boxes. In *Progress in Cryptology–INDOCRYPT 2021, Proceedings of the 22nd International Conference on Cryptology in India, Jaipur, India, 12–15 December 2021*; Proceedings 22; Springer: Berlin/Heidelberg, Germany, 2021; pp. 159–178.
15. Feng, J.; Wei, Y.; Zhang, F.; Pasalic, E.; Zhou, Y. Novel Optimized Implementations of Lightweight Cryptographic S-Boxes via SAT Solvers. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *71*, 334–347. [CrossRef]
16. Dasu, V.A.; Baksi, A.; Sarkar, S.; Chattopadhyay, A. LIGHTER-R: Optimized Reversible Circuit Implementation For SBoxes. In Proceedings of the 32nd IEEE International System-on-Chip Conference, SOCC 2019, Singapore, 3–6 September 2019; pp. 260–265. [CrossRef]
17. Chun, M.; Baksi, A.; Chattopadhyay, A. DORCIS: Depth Optimized Quantum Implementation of Substitution Boxes. Cryptology ePrint Archive, Paper 2023/286. 2023. Available online: https://eprint.iacr.org/2023/286 (accessed on 25 March 2024).
18. Chen, J.; Liu, Q.; Fan, Y.; Wu, L.; Li, B.; Wang, M. New SAT-based Model for Quantum Circuit Decision Problem: Searching for Low-Cost Quantum Implementation. *IACR Commun. Cryptol.* **2024**, *1*. [CrossRef]
19. Lin, D.; Yang, C.; Xu, S.; Tian, S.; Sun, B. On the Construction of Quantum Circuits for S-Boxes with Different Criteria Based on the SAT Solver. Cryptology ePrint Archive, Paper 2024/565. 2024. Available online: https://eprint.iacr.org/2024/565 (accessed on 25 March 2024).
20. Amy, M.; Maslov, D.; Mosca, M.; Roetteler, M.; Roetteler, M. A Meet-in-the-Middle Algorithm for Fast Synthesis of Depth-Optimal Quantum Circuits. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2013**, *32*, 818–830. [CrossRef]
21. Roy, S.; Baksi, A.; Chattopadhyay, A. Quantum Implementation of ASCON Linear Layer. Cryptology ePrint Archive, Paper 2023/617. 2023. Available online: https://eprint.iacr.org/2023/617 (accessed on 25 March 2024).
22. Xiang, Z.; Zeng, X.; Lin, D.; Bao, Z.; Zhang, S. Optimizing Implementations of Linear Layers. *IACR Trans. Symmetric Cryptol.* **2020**, *2020*, 120–145. [CrossRef]
23. Zheng, Y.; Luo, Q.; Li, Q.; Lv, Y. Quantum circuit implementations of lightweight authenticated encryption ASCON. *J. Supercomput.* **2024**, 1–16. [CrossRef]
24. Baksi, A.; Jang, K.; Song, G.; Seo, H.; Xiang, Z. Quantum Implementation and Resource Estimates for Rectangle and Knot. *Quantum Inf. Process.* **2021**, *20*, 395. [CrossRef]
25. Anand, R.; Maitra, A.; Maitra, S.; Mukherjee, C.S.; Mukhopadhyay, S. Quantum Resource Estimation for FSR Based Symmetric Ciphers and Related Grover's Attacks. In *Progress in Cryptology—INDOCRYPT 2021, Proceedings of the 22nd International Conference on Cryptology in India, Jaipur, India, 12–15 December 2021*; Lecture Notes in Computer Science; Adhikari, A., Küsters, R., Preneel, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 13143, pp. 179–198. [CrossRef]
26. Huang, Z.; Sun, S. Synthesizing Quantum Circuits of AES with Lower T-depth and Less Qubits. In *Advances in Cryptology—ASIACRYPT 2022, Proceedings of the 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 5–9 December 2022*; Lecture Notes in Computer Science; Agrawal, S., Lin, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; Part III, Volume 13793, pp. 614–644. [CrossRef]
27. Zou, J.; Wei, Z.; Sun, S.; Liu, X.; Wu, W. Quantum Circuit Implementations of AES with Fewer Qubits. In *Advances in Cryptology—ASIACRYPT 2020, Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, Republic of Korea, 7–11 December 2020*; Moriai, S., Wang, H., Eds.; Springer: Cham, Switzerland, 2020; pp. 697–726.

28. Brassard, G.; Hoyer, P.; Tapp, A. Quantum algorithm for the collision problem. *arXiv* **1997**, arXiv:quant-ph/9705002.
29. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. Ascon v1. 2: Lightweight authenticated encryption and hashing. *J. Cryptol.* **2021**, *34*, 33. [CrossRef]