applied sciences

MDPI

Article

Improved Low-Depth SHA3 Quantum Circuit for Fault-Tolerant Quantum Computers

Gyeongju Song, Kyungbae Jang and Hwajeong Seo *

Division of IT Convergence Engineering, Hansung University, Seoul 02876, Republic of Korea
* Correspondence: hwajeong@hansung.ac.kr; Tel.: +82-760-8033

Abstract: To build a secure cryptography system in the post-quantum era, one must find the minimum security parameters against quantum attacks by estimating the quantum resources of a fault-tolerant quantum computer. In a fault-tolerant quantum computer, errors must reach an acceptable level for practical uses according to error detection and error correction processes. However, these processes utilize additional quantum resources. As the depth of the quantum circuit increases, the computation time per qubit increases together with the processing errors. Therefore, in terms of errors in quantum circuits, it is a fundamental requirement to reduce the depth by trading off the number of qubits. This paper proposes novel low-depth SHA3 quantum circuit implementations for fault-tolerant quantum computers to reduce errors. The proposed SHA3 quantum circuit was implemented with the aim of optimizing the quantum circuit depth through a trade-off between the number of qubits, the quantum gate, and the quantum depth in each function. Compared to other state-of-art techniques, the proposed method achieved *T*-depth and full-depth reductions of 30.3% and 80.05%, respectively. We believe that this work will contribute to the establishment of minimum security parameters for SHA3 in the quantum era.

Keywords: quantum implementation; Grover algorithm; SHA3; circuit depth; qubit

Citation: Song, G.; Jang, K.; Seo, H. Improved Low-Depth SHA3 Quantum Circuit for Fault-Tolerant Quantum Computers. *Appl. Sci.* **2023**, *13*, 3558. https://doi.org/10.3390/app13063558

Academic Editors: Konstantinos Rantos, Konstantinos Demertzis and George Drosatos

Received: 17 February 2023
Revised: 7 March 2023
Accepted: 8 March 2023
Published: 10 March 2023

1. Introduction

As the world evolves into the information age, data encryption is essential to protect digital content. Currently, digital content is protected through symmetric-key cryptography (e.g., the Advanced Encryption Standard (AES)) and public-key cryptography (e.g., Riverst–Shamir–Adleman (RSA), Elliptic Curve Cryptography (ECC)). With the unexpected rapid development in quantum computers, the security of modern cryptography against quantum algorithms has become unclear. Grover's algorithm, proposed by Lov Grover in 1996 [1], is known to accelerate brute-force attacks and pre-image attacks on symmetric-key cryptography and hash functions. Shor's algorithm, proposed by Peter Shor in 1994 [2], is known to be able to solve the basic problems of existing public-key cryptography systems, such as factorization and discrete logarithms, in polynomial time.

In the past, quantum computers were an abstract concept, but many companies have recently established fault-tolerant quantum computers, showing the possibility of their use in practical applications. To operate a quantum computer, it is necessary to provide certain quantum resources (i.e., a certain number of qubits, quantum gates) that are required for operation. When quantum resources meet the requirements of quantum algorithms for a target cryptographic attack, the cryptography is considered to be no longer effective in data protection. In order to resolve this issue, symmetric-key cryptography increases the key size, and the hash function increases the output length. In the case of public-key cryptography, the current system should be replaced with other post-quantum cryptography standards. It is also important to establish an efficient security system for symmetric-key cryptography and hash functions by finding the minimum security parameters that ensure a certain level of security for fault-tolerant quantum computers. This can be achieved by estimating

the resources required for an attack through the implementation of quantum circuits for symmetric-key cryptography and hash functions. Since a quantum computer operates using the quantum mechanical phenomena of qubits, it is important to preserve the integrity of the qubit state in the operation. Qubit data can be corrupted when the value of an unstable qubit fluctuates during calculation due to the nature of high noise rates. To be operational, a fault-tolerant quantum computer should tolerate appropriate errors, error correction, and error detection for unstable qubits. Various studies have been conducted regarding these tasks [3–6].

The number of qubits and the quantum circuit depth are generally inversely proportional to each other in a quantum circuit implementation. When implementing a quantum circuit, two methods can be considered. One can reduce the depth of a quantum circuit by increasing the number of qubits, or one can reduce the number of qubits by increasing the depth of the quantum circuit. In the era of noisy intermediate-scale quantum (NISQ), quantum computers are not yet a stable technology [7]. For this reason, it is difficult to determine which is the more efficient and optimized method for implementing a quantum circuit. However, researchers need to conduct research in all aspects, and this factor is likely to be discussed again when quantum computers become more realistic. Approaches that increase the number of qubits and decrease the quantum circuit depth are more suitable in terms of optimizing the noise of the quantum circuit. As the depth of the quantum circuit increases, the operation time also increases, which affects the increase in the error rate of the quantum circuit.

With this research aim, this paper presents an improved low-depth SHA3 quantum circuit for fault-tolerant quantum computers to reduce errors. Error detection and correction are essential for fault-tolerant quantum computers, and errors increase proportionally with depth. We also present a corresponding technique to research how to reduce errors by reducing the depth of the overall operation through the proposed method. The method used in [8], which returned the qubit state to its previous state through an inverse operation and reused it in the next operation, could reduce the number of qubits. This approach increased the number of errors as the inverse of the function increased the depth. Many works have explored error detection and error correction. In [9], the Q# development tool was used to reduce the total number of qubits compared to the results presented in [8]. The number of qubits was reduced by 1280 compared to the first work, and 1920 qubits were used in total. The current study, taking a different direction to the previous research by aiming to reduce the number of qubits, implemented a quantum circuit by increasing the number of qubits and reducing the inverse operation process, and a separate attempt to reduce the number of qubits was also made. The overall quantum circuit depth was reduced by changing the internal operating structure rather than simply optimizing the depth through the increase in the use of qubits. Therefore, the proposed quantum circuit was efficient in terms of its depth. As a result, compared to previous works [8], the T depth and full depth were reduced by 30.3% and 80.05%, respectively. We could not compare the results of previously proposed methods with our own. The efficient implementation method pursued by each research group was different (i.e., the number of qubits), and we focused on reducing the depth of the quantum circuit. Therefore, the differences between the proposed quantum circuit and previous research results were as follows: (1) The reduced depth of the quantum circuit reduced the time and errors required for quantum operations. (2) Our work required more quantum qubits for computation than previous works. However, the setting with more qubits was suitable for real-world quantum computers.

1.1. Contributions

This paper proposes an improved low-depth SHA3 quantum circuit for fault-tolerant quantum computers. We worked to reduce the depth inside the quantum circuit. As a result, we reduced the T depth and full depth compared to previous works by 30.3% and 80.05%, respectively. The contributions made to the proposed SHA3 quantum circuit are summarized below.

1.1.1. Constructive Contributions

In this paper, the structure of SHA3 was identified, and the quantum circuit was designed in a way that reduced its depth. In the NISQ era, it is difficult to conclude which implementation produces a more efficient quantum circuit. However, researchers need to explore all directions, and to meet this requirement, we pioneered the direction of reducing errors by significantly reducing the quantum circuit depth.

1.1.2. Contributions in Terms of Quantum Cost (Resource Trade-Off)

In a fault-tolerant quantum computer, error detection and error correction are essential to control the errors that accumulate through noise, calculation errors, and incorrect operations. For this operation, additional quantum resources are required. In fault-tolerant circuits, the error increases as the length of the operation increases, so the error can be decreased by reducing the depth. Therefore, in terms of errors, it is more effective to reduce the depth of quantum circuits. The proposed SHA3 reversible quantum circuit showed the results of drastically reducing the depth through a quantum resource trade-off.

1.1.3. Preparing for the Post-Quantum Era

To prepare for the post-quantum era, it is necessary to find parameters that provide a satisfactory level of security by estimating the resources required for an attack through the implementation of a quantum circuit for the target cryptography. Finding the minimum security parameters is effective for constructing a secure cryptography system. The proposed new direction for SHA3 quantum circuit implementation could contribute to the study of minimum security parameters in terms of depth.

2. Preliminaries

2.1. Quantum Computing

Quantum computers can solve specific problems faster than classic computers by using the quantum mechanical properties of qubits for operation. In a quantum computer, data are expressed in qubits, and operations are performed by manipulating the state of qubits through a reversible circuit. The qubit exists in a superposition state that is probabilistically 0 and 1 at the same time throughout the operation until the final measurement. Thus, the qubit exists in multiple states at the same time. Due to this property, the operation for all cases of 0 and 1 can be probabilistically calculated at once, so the calculation speed is fast, and a probabilistic result is output at the last measurement. The measurement probability for a qubit in a particular state can be described by the probability amplitude associated with the state.

The superposition of qubits via the Hadamard gate is expressed as:

$$|\psi\rangle = \alpha|1\rangle + \beta|0\rangle, \quad |\alpha|^2 + |\beta|^2 = 1$$

A qubit in superposition produces an eigenvalue of either 0 or 1 after measurement, and the value is unknown before measurement [10]. In the above expression, α and β refer to the probability amplitude, and if $|\alpha|^2$ is 0, $|\beta|^2$ is 1, and vice versa.

Since all quantum gates used to control the state of qubits are reversible, inverse operations are possible. The placement of quantum gates is directly related to the depth of the quantum circuit, and many previous studies have been conducted to reduce the total number and depth of quantum gates by reducing the number or optimizing the placement of the quantum gates used in an operation [11–26]. Representative quantum gates include the H gate, X gate, CNOT gate, $Toffoli$ gate, and T gate, which are presented below:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad X = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

As a non-Clifford gate, the $Toffoli$ gate can be decomposed into lower-level gates. In the proposed quantum circuit, the $Toffoli$ gate decomposed as shown in Figure 1 was used for T-depth estimation. The Toffoli gate includes a non-Clifford T gate, and the steps of the T gate lead to the T depth. Minimizing the number of T gates is still important, as non-Clifford T gates have a long latency, and their implementation cost far exceeds that of Clifford gates in fault-tolerant implementation [27,28].

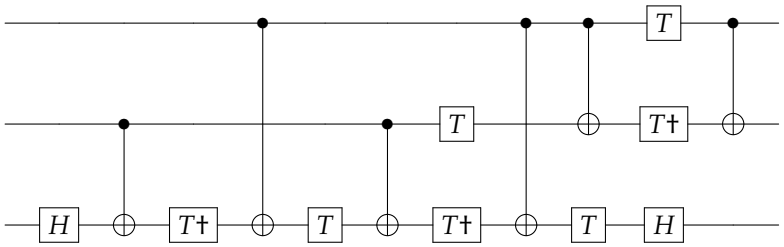


Figure 1. Decomposed Toffoli gate. † indicates reversible gates.

2.2. Secure Hash Algorithm (SHA) 3

In 2015, the National Institute of Standards and Technology (NIST) released Secure Hash Algorithm 3 (SHA3) [29] to replace SHA1 [30] and SHA2 [31]. The SHA3 hash function family consists of four hash functions (SHA3-224, SHA3-256, SHA3-384, and SHA3-512) and two extendable-output functions (XOFs) (SHAKE128 and SHAKE256). The input data output hash results through 'absorbing' and 'squeezing' steps using the sponge structure, as shown in Figure 2. SHA3 has a sponge structure, so it outputs a hash value of a constant length regardless of the input length. When absorbing, a message block is transformed through XOR and permutation functions, and the transformed message block is updated by repeating the function f (i.e., Keccak-f[1600, 24]) composed of five steps : θ, ρ, π, χ, and ι. The inner operation of function f is explained in detail in Section 3 for implementation in the SHA3 quantum circuit.

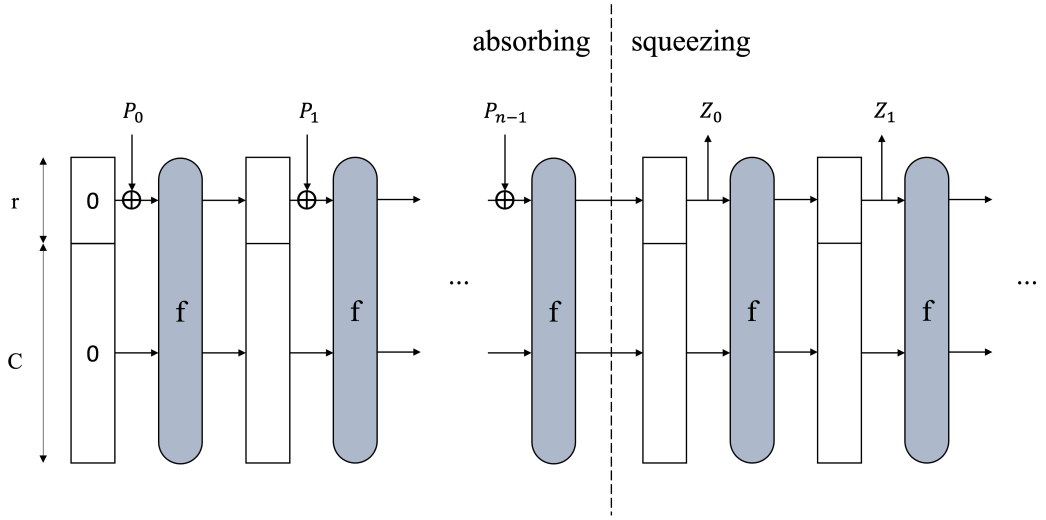


Figure 2. Operation process for SHA3 hash function. (r: block size, c: capacity, f: Keccak-f[1600]).

2.2.1. Pre-Image Attack

A hash function maps data of an arbitrary length onto a hash value of a fixed length. This feature increases the speed of the data search, as long and diverse data can be arranged in a certain length. A pre-image is a way to find the original message when a hash value is provided: find M given H for $H = hash(M)$. A pre-image attack is an attempt by an attacker to find the original message through a hash value. The pre-image resistance of the hash function, which increases as the hash length increases, has n-bit resistance for an n-bit hash length. A hash function that is difficult to find using a pre-image is defined as a better hash function. The collision resistance of Secure Hash Algorithm (SHA) 3 is $2^{n/2}$, the pre-image resistance is 2^n, and the output length is $n = 224, 256, 384, 512$. Table 1 shows the parameters and pre-image resistance for the SHA3 hash function family.

Table 1. Parameters and pre-image resistance for SHA3 hash function family.

Algorithm	Hash Length	Block Size	Capacity	Pre-Image
SHA3-224	224	1152	448	224
SHA3-256	256	1088	512	256
SHA3-384	384	832	768	384
SHA3-512	512	576	1024	512
SHAKE128	d	1344	256	$\geq$min(d,128)
SHAKE256	d	1088	512	$\geq$min(d,256)

2.2.2. Quantum Pre-Image Attack

In the worst case, it will take N searches to find specific data in N unsorted datasets. On quantum computers, Grover's algorithm allows one to find specific data in $\sqrt{N}$ searches. Grover's algorithm speeds up pre-image attacks on hash functions as it can quickly search N data fields to find an input that outputs a specific hash value in the hash function. Therefore, the computational complexity $O(N)$ for a brute-force attack in a classic computer is reduced to the computational complexity $O(\sqrt{N})$ in a quantum computer. Grover's algorithm for a pre-image attack is divided into the Oracle and Diffusion operators, as shown in Figure 3. This attack is a known plaintext attack (KPA) that proceeds when the plaintext–ciphertext pairs of the block cipher are known. The Oracle function includes the hash function $f_g(x) = y$ and the inverse operation $f_g^\dagger(x) = y$. If the result of $f_g(x)$ is y, then $x = 1$ in Oracle, and the measurement probability for the state is increased through the diffusion operator $U_s = 2|s\rangle\langle s| - I$. It is known that the state of the correct qubit can be found in about $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor$ iterations of Grover's algorithm.

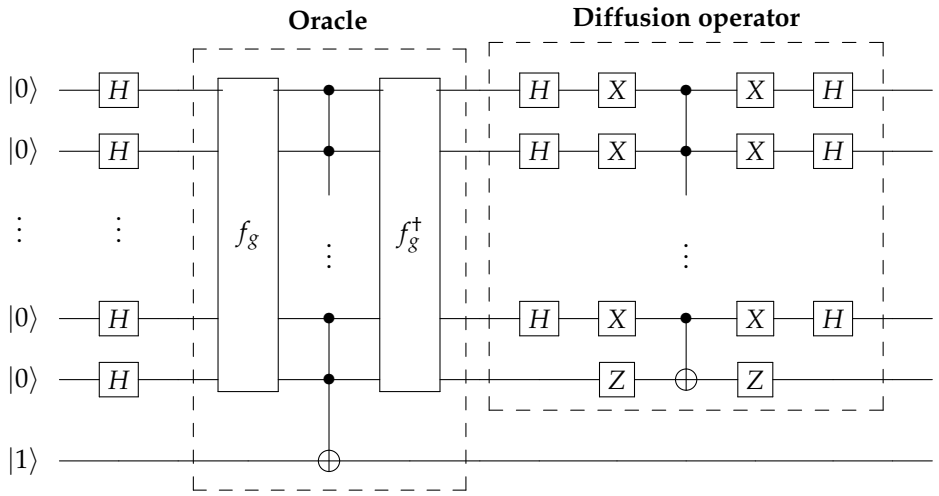


Figure 3. Grover's algorithm with $f_g : \{0,1\}^n \leftarrow \{0,1\}^n$ in oracle. † indicates reversible gates.

3. SHA3 Quantum Circuit

The data input to the qubit output the hash function through the process of absorbing and squeezing by the sponge structure. In the absorbing process, the message block is converted through XOR and permutation functions, and the converted message block is updated by repeating the function f (i.e., Keccak-f[1600, 24]). The final hash value is output through the squeezing process. The proposed improved low-depth SHA3 quantum circuit for fault-tolerant quantum computers was implemented for all Keccak-f phases. This section describes the implementation of quantum circuits for each function. The SHA3 internal function f consists of five steps, presented below, and operates as many as $12 + 2l$ rounds, depending on the b bits (SHA3: b = 1600). The power-of-two word size w is defined as $w = 2^l$ bit, and SHA3 uses a 64-bit word (i.e., l = 6):

$$f = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

In the SHA3 operation, each step of Keccak-f proceeds with a multi-dimensional bit array structure of data. In the same way, the quantum circuit was constructed assuming that the qubits were arranged in a multi-dimension bit array along the structure of each step. Figure 4 shows each part of the three-dimensional bit array matrix state in Keccak-f[1600].

3.1. Theta (θ)

Theta (θ) is one of the five phases of the SHA3 (i.e., Keccak-k) hash function. In the θ phase, the data are processed in the three-dimensional state array structure shown in Figure 4. The result of $\Sigma((x-1),z) \oplus \Sigma((x+1),(z-1))$ saves to (x,y,z) bits. The final result value is stored in (x,y,z).

$$\begin{aligned} &C[x,z] = A[x,0,z] \oplus A[x,1,z] \oplus A[x,2,z] \oplus A[x,3,z] \oplus A[x,4,z], \ \ \forall x,y \\ &D[x,z] = C[(x-1)\%4,z] \oplus C[(x+1)\%4,(z-1)\%w], \ \forall x \ and \ 0 \le z \le w \\ &R[x,y,z] = A[x,y,z] \oplus D[x,z] \end{aligned} \tag{1}$$

Equation (1) is the theta (θ) operation in a classic computer. In classic computer operation, temporary registers (hereafter referred to as *temp*) of C, D, and R are allocated to store intermediate calculation values. Therefore, four 1600-bit *temp* are used. Quantum circuits reduce the use of 4800 qubits by allocating one 1600-bit *temp* of one state size. The proposed quantum circuit avoids an increase in depth by not initializing *temp* qubits through reverse operation in each round. This scheme allocates one temp qubit to replace the inverse operation per round.

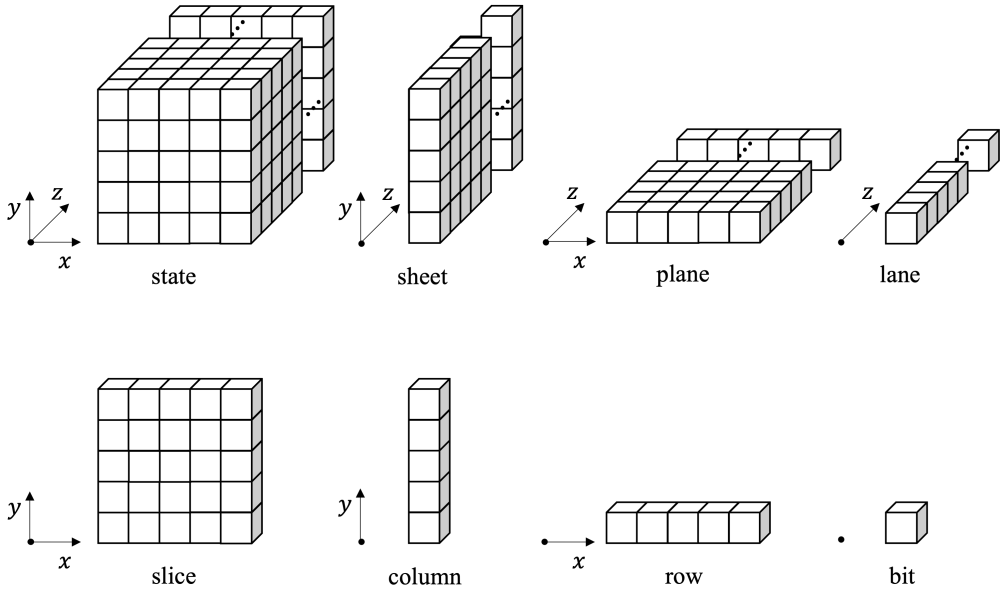
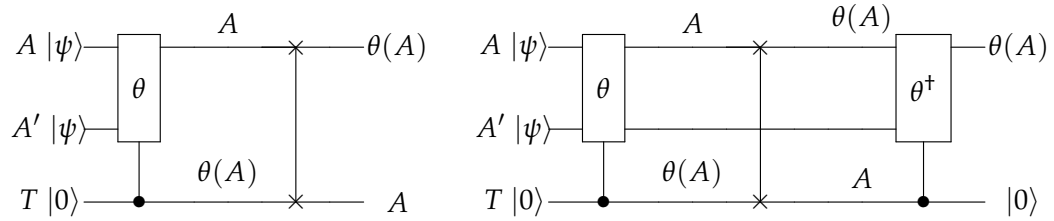


Figure 4. Each part of the three-dimensional bit array matrix state in Keccak-f[1600].

Figure 5 shows (a) the exclusion of the inverse operation and (b) the inclusion of the inverse operation in theta (θ). T represents the *temp* qubit. The scheme excluding the inverse operation includes only one θ function in the quantum circuit per round. The scheme including the inverse operation involves the θ function and the $\theta^{\dagger}$ function to return the temporary qubit T to its original state in the quantum circuit each round. Including the inverse operation can reduce T qubits but increases the number of quantum gates and the depth. In an attempt to reduce the depth, the scheme excluding inverse operations was selected in this paper. Compared to [8], the operation process proposed in this paper increased the CNOT gate by about 36.36% and reduced the depth by about 71.27% in θ (trade-off between quantum gates and depth). In the implementation of [8], $\theta^{\dagger}$ used the most CNOT gates and increased the depth. Considering this, the proposed quantum circuit replaced $\theta^{\dagger}$ with the use of T qubits. As a result, a trade-off occurred between 1,360,000 CNOT gates + 25 depth and 1600 qubits at $\theta^{\dagger}$ per round (increase: qubit, decrease: CNOT gate + depth).



(a) Excluding inverse operation (b) Including inverse operation

Figure 5. Quantum circuit for θ. (T: temporary qubits).

Algorithm 1 shows the operation of our quantum circuit for theta (θ). In the input, X and T denote the input qubit and the *temp* qubit. All operations performed by the CNOT gate updated T, and T was returned at the end. Compared to previous research results [8], the proposed algorithm increased the CNOT gate but reduced the full depth in theta (θ).

Algorithm 1 Quantum algorithm for theta (θ)

Input: X , T

1: **for** (i = 0 **to** 5) : **for** (j = 0 **to** 5) : **for** (k = 0 **to** 64) :
2: **for** $s = 0$ to 5 **do**
3: $T[i][j][k] \leftarrow \mathbf{CNOT}(X[(i-1)\%5][s][k], T[i][j][k])$
4: $T[i][j][k] \leftarrow \mathbf{CNOT}(X[(i+1)\%5][s][(k-1)\%64], T[i][j][k])$
5: $T[i][j][k] \leftarrow \mathbf{CNOT}(X[i][j][k], T[i][j][k])$
6: **end for**

Return T

3.2. Rho (ρ)

In the rho (ρ) phase, the operation proceeds with the lane structure of the state shown in Figure 4. The rotation of the index operates according to the set offset inside each lane. The index rotation operation in rho (ρ) is as follows:

$$Rho(\rho) : X[x][y][z] \leftarrow X[x][y][z-(t+1)(t+2)/2]$$

where

$$0 \leq t \leq 23, \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix}^t \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

If this is simply connected to a quantum circuit, a separate reversible gate is not required, and it is implemented in a way that only changes the physical position of the qubit without using a SWAP gate for rotation. As a result, no reversible quantum gate is used in rho (ρ).

3.3. Pi (π)

The pi (π) phase is used to permute the values of lanes within the state $x[3x+2y][x] \leftarrow x[x][y]$. Similar to rho ($\rho$), no reversible quantum gate is used in the pi (π) phase, because it only changes the physical position of the qubit.

3.4. Chi (χ)

Chi (χ) is the only non-linear part of Keccak-f. Considering the results of the quantum circuit implementation, the Toffoli gate was only used in this step. Since it was the only internal step to use the T gate, it presented the T depth. Chi (χ) is the process of XOR operation with the result of multiplying the correct two bits in the row, and the operation is as follows:

$$X'[x,y,z] = X[x,y,z] \oplus ((X[x+1]mod\ 5, y, z] \oplus 1) \cdot X[(x+2)mod\ 5, y, z])$$

This shows the classic chi (χ) operation. In the proposed quantum circuit, the operation results are reflected directly on the target qubit without intermediate *temp* qubits to reduce the depth and additional *temp* qubits. Using the Toffoli gate, the result of $(X[x+1]mod\ 5][y][z] \oplus 1) \cdot X[(x+2)mod\ 5][y][z]$ is directly reflected in the target qubit. The values required to update $X[x][y][z]$ in $0 \leq x \leq 4$ are shown in Table 2.

Table 2. The qubit values required for the step-by-step update of the input of chi (χ); $\circledast$ indicates a change from the preceding calculation.

Order	Required Qubit		Update Target
$x = 0$	$X[1][y][z]$	$X[2][y][z]$	$X[0][y][z]$
$x = 1$	$X[2][y][z]$	$X[3][y][z]$	$X[1][y][z]$
$x = 2$	$X[3][y][z]$	$X[4][y][z]$	$X[2][y][z]$
$x = 3$	$X[4][y][z]$	$\circledast$ $X[0][y][z]$	$X[3][y][z]$
$x = 4$	$X[0][y][z]$	$\circledast$ $X[1][y][z]$	$X[4][y][z]$

In the order $x = 0, 1, 2$, there is no problem in updating X, but for $x = 3, 4$, a problem arises because the state of the qubits of X' and X required for the operation has changed from the preceding operation (marked with $\circledast$ in the Table 2). A method involving inverse operation can be considered, but this greatly increases the depth of the quantum circuit. The proposed quantum circuit allocated qubits to store the values of $X[0][y][z]$ and $X[1][y][z]$ before operation and maintain the values. For each round in chi (χ), this method reduced the CNOT gate by about 98.08%, the T depth by about 30.3%, and the full depth by about 90.08% through the use of an additional 640 *temp* qubits. Algorithm 2 shows the quantum circuit operation for chi (χ).

Algorithm 2 Quantum algorithm for chi (χ)

Input: x , T_0, T_1

$x[0] \leftarrow$ **CNOT**$(X[0], T_0)$
$x[1] \leftarrow$ **CNOT**$(X[1], T_1)$

for (i = 0 **to** 5) : **for** (j = 0 **to** 5) : **for** (k = 0 **to** 64) :
if i == 0, 1, 2:
  $x[(i+1)\%5][j][k] \leftarrow$ **X**$|(x[(i+1)\%5][j][k])$
  $x[(i+1)\%5][j][k] \leftarrow$ **Toffoli**$(x[(i+1)\%5][j][k], (x[(i+2)\%5][j][k], x[i][j][k])$
  $x[(i+1)\%5][j][k] \leftarrow$ **X**$|(x[(i+1)\%5][j][k])$
if i == 3:
  $x[(i+1)\%5][j][k] \leftarrow$ **X**$|(x[(i+1)\%5][j][k])$
  $x[(i+1)\%5][j][k] \leftarrow$ **Toffoli**$(x[(i+1)\%5\ 5][j][k], T_0[j][k], x[i][j][k])$
  $x[(i+1)\%5][j][k] \leftarrow$ **X**$|(x[(i+1)\%5][j][k])$
if i == 4:
  $T_0[j][k] \leftarrow$ **X**$|(T_0[j][k])$
  $x[i][j][k] \leftarrow$ **Toffoli**$(T_0[j][k], T_1[j][k], x[i][j][k])$
  $T_0[j][k] \leftarrow$ **X**$|(T_0[j][k])$

Return x

3.5. Iota (ι)

Iota (ι) is the process of XOR operation between lane(0,0) and the round constant $T[x][0][0] = T[x][0][0] \oplus RC[x]$. Since RC is a constant, it proceeds as a classic calculation rather than a quantum circuit. In the CNOT operation on RC and T, since RC represents classic data and the input is quantum data, the X gate is applied to T according to the RC value. In this way, the quantum resources required for RC calculation can be reduced, and

the CNOT gate can be replaced with the X gate (the CNOT gate is regarded as a higher-cost quantum resource than the X gate). The quantum resource (X gate) used in iota (ι) depends on the constant RC.

3.6. Quantum Cost Analysis for SHA3

Table 3 shows the proposed quantum resources for the Keccak-f function in SHA3, and Table 4 shows the quantum resources for our method, with the results of Amy et al. [8] included for comparison. As shown in Table 3, in the proposed quantum circuit, θ and χ used the most quantum resources, and 1600 qubits in θ and 640 qubits in χ were used for each round.

As shown in Table 4, the proposed quantum circuit increased the number of qubits to reduce the depth of the quantum circuit, resulting in a reduction in the depth of each function compared to previous implementations. In the theta (θ) operation, we increased the CNOT gate by about 36.36% and reduced the depth by about 71.27% per round. To omit the θ^{-1} process, 1600 additional qubits were used. The result of this quantum trade-off was a reduction of 1,360,000 CNOT gates + 25 depth for θ^{-1} and an increase of 1600 qubits. In total, the full depth was reduced by about 73.67% in theta(θ) and theta inverse θ^{-1}.

In chi (χ), 640 qubits were used additionally to reduce the CNOT gate by about 98.08%, the T depth by about 30.3%, and the full depth by about 90.08% per round (trade-off between qubits and gate + depth). In [8], the number of Toffoli gates used was not shown, but through the T depth, it can be inferred that Toffoli gates were used more than in our method. Our method used more 1qClifford gates but reduced the number of CNOT and Toffoli gates, which are more expensive quantum gates than 1qClifford gates, so we saw this as an appropriate quantum resource trade-off.

For the operation of iota (ι), the classic-to-quantum method reduced the quantum resources required for RC calculation and replaced the use of CNOT gates with X gates.

As a result of these efforts, the proposed improved low-depth SHA3 quantum circuit for fault-tolerant quantum computers decreased the depth of all functions, reducing the overall quantum circuit depth by about 80.01%.

Table 3. Quantum resource estimation results for each phase of Keccak-f in SHA3 (Round: quantum resources per round, Total: quantum resources for full round). # incicates the number of gates.

Function	#1qClifford	#CNOT	#Toffoli	#T Depth	#Full Depth
θ	0	24,000	0	0	79
ρ	0	0	0	0	0
π	0	0	0	0	0
χ	3200	640	1600	23	12
ι	2	0	0	0	1
Round	3202	24,640	1600	23	88
Total	76,886	591,360	38,400	552	2020

Table 4. Comparison of quantum resources for the-state-of-art SHA3 quantum circuit implementation and proposed SHA3 quantum circuit implementation.

Function		#1qClifford	#CNOT	#Toffoli	#T Depth	#Full Depth
θ	Our method	0	24,000	0	0	79
	[8]	0	17,600	0	0	275
θ^{-1}	Our method	0	0	0	0	0
	[8]	0	1,360,000	0	0	25
ρ	Our method			(Not used)		
	[8]					
π	Our method			(Not used)		
	[8]					
χ	Our method	3200	640	1600	23	12
	[8]	0	14,400	(Not shown)	15	55
χ^{-1}	Our method	0	0	0	0	0
	[8]	0	18,880	(Not shown)	18	66
ι	Our method	2	0	0	0	1
	[8]	85	0	0	0	24
Total		76,886	591,360	38,400	552	2020
Total [8]		85	33,269,760	-	792	10,128

We optimized the depth of each function in the SHA3 quantum circuit for low quantum computing error rates. Our approaches reduced the depth of the quantum circuits. In [32], the additional parts for Korean block ciphers LEA, HIGHT, and CHAM were implemented in parallel to reduce the overall depth compared to the first proposed quantum circuit [33]. The quantum resource estimation results of the preceding quantum circuit [32] presented in Table 5 were efficiently reduced in terms of depth through quantum resource trade-off, and the results are shown in Table 6. The method presented in [33] showed performance enhancements in terms of quantum circuit depth for LEA, HIGHT, and CHAM of 78%, 85%, and 70%, respectively.

In [14], to reduce the depth of the Korean standard hash function LSH, a part that made parallel operation possible was identified in previous research [34] and designed to perform parallel operation inside the quantum circuit. As a result, compared to the initial research results presented in Table 7, the full depth of the quantum circuit was reduced by about 96%, and the results are shown in Table 8. The previous study reduced the internal calculation time and error by reducing the depth of the quantum circuit. The depth of the quantum circuit was reduced by about 70% to 96% compared to the initial work.

Table 5. Quantum resources required for the Korean block cipher proposed in [32].

Cipher		#Toffoli	#CNOT	#X	#Full Depth
LEA [32]	128/128	10,416	28,080	68	26,328
	128/192	15,624	39,816	100	39,452
	128/256	17,856	45,504	130	45,057
HIGHT [32]	64/128	6272	20,523	4	16,447
CHAM [32]	64/128	2400	12,285	240	7807
	128/128	4960	26,885	240	19,880
	128/256	5952	32,277	304	23,856

Table 6. Quantum resources required for the Korean block cipher proposed in [33]. (Results for depth reduction compared to Table 5 through quantum resource trade-off).

Cipher		#Toffoli	#CNOT	#X	#Full Depth
LEA [33]	128/128	10,248	32,616	11,152	6505
	128/192	15,372	46,620	17,004	7589
	128/256	17,568	53,280	19,494	8580
HIGHT [33]	64/128	5824	22,614	4496	2479
CHAM [33]	64/128	2320	13,200	2320	2615
	128/128	4880	28,760	4880	5307
	128/256	5856	34,944	5872	6594

Table 7. Quantum resources required for LSH hash function proposed in [34].

Cipher		#Toffoli	#CNOT	#X	#Full Depth
LSH [34]	256/224	63,488	145,152	1536	210,051
	256/256	63,488	145,152	3492	210,049
	512/224	139,104	312,832	7663	421,851
	512/256	139,104	312,832	7696	421,851
	512/384	139,104	312,832	7668	421,850
	512/512	139,104	312,832	7680	421,852

Table 8. Quantum resources required for LSH hash function proposed in [14]. (Results for depth reduction compared to Table 7 through quantum resource trade-off).

Cipher		#Toffoli	#CNOT	#X	#Full Depth
LSH [14]	256/224	62,464	170,752	59,392	6879
	256/256	62,464	170,752	59,392	6879
	512/224	138,000	375,760	134,688	14,517
	512/256	138,000	375,760	134,688	14,517
	512/384	138,000	375,760	134,688	14,517
	512/512	138,000	375,760	134,688	14,517

Table 9 shows the quantum resources needed for the SHA-256 [8] and SM3 [15] hash functions. The results of comparing our SHA3 quantum circuit with the quantum circuits for other hash functions (LSH-256 [14], SHA-256 [8], and SM3 [15]) were as follows:

The SHA3 quantum circuit proposed in this paper used more X gates and CNOT gates than the parallel quantum circuit of LSH-256 [14] and had fewer Toffoli gates and a smaller T depth and full depth. Compared to the SHA-256 [8] quantum circuit, our SHA3 quantum circuit used more X gates and CNOT gates but fewer Toffoli gates, and the T depth and full depth were smaller. Compared to the Chinese National Standard hash function SM3 [15], the SHA3 hash function used X gates and CNOT gates more and Toffoli gates less, and the T depth and full depth were smaller. Compared to the LSH, SHA256, and SM3 hash function quantum circuits, the proposed SHA3 required the more X and CNOT gates, but the required number of Toffoli gates, T depth, and full depth were lower. In future quantum computing, it is expected that SHA3 will be the most vulnerable hash function according to its depth. On the other hand, in terms of qubits, it is expected that the time to reach the number of qubits required for SHA3 quantum circuit operation will be the longest.

Table 9. Quantum resources required for SHA-256 [8] and SM3 [15] hash functions.

Cipher	#X	#CNOT	#Toffoli	#T Depth	#Full Depth
SHA-256 [8]	0	534,272	(Not shown)	171,552	528,768
SM3 [15]	2638	134,144	43,328	(Not shown)	121,242

4. Future Work

In this paper, we proposed an implementation of a SHA3 quantum circuit focusing on reducing the depth by using more qubits. Error detection and correction are essential for operation in a fault-tolerant quantum computer. Since the error is proportional to the depth, more qubits are needed for error correction as the depth increases. Therefore, we proposed a new method to reduce errors by reducing the depth of the overall operation. This is expected to show excellent results in terms of errors if large-scale fault-tolerant quantum computers appear in the future. However, in the NISQ era, neither the method of reducing qubits nor that of reducing the depth can be labeled as more efficient. Therefore, as quantum computers develop in the future, it is expected that research on a hybrid of the two methods will be necessary.

5. Conclusions

This paper proposed highly optimized low-depth SHA3 quantum circuits for fault-tolerant quantum computers. To operate a quantum circuit in a fault-tolerant quantum computer, it must be corrected to an acceptable level of error through proper error detection and correction. Certain quantum resources (e.g., qubits) are additionally used for this task. In a classic quantum circuit implementation, the number of qubits and the quantum circuit depth are inversely proportional. Quantum circuits can be implemented considering two methods. Since quantum computers are currently an uncertain technology, it is difficult to emphasize which is most efficient. As the quantum circuit depth increases, the computation time for each qubit increases, which increases the error. From a quantum noise perspective, it makes more sense to increase the number of qubits, reducing the quantum circuit depth to decrease errors. In this paper, we worked to reduce the quantum circuit depth to reduce the errors occurring in cryptography operations. Quantum circuits were implemented with the aim of reducing the depth through a trade-off between the number of qubits and the quantum gates + depth for each SHA3 function. As a result, the T depth was reduced by about 30.3% and the full depth by about 80.05% compared to the results of previous research. We expect that our work will contribute to the establishment of minimum security parameters for SHA3 in the post-quantum era.

Author Contributions: Software, G.S.; Writing—original draft, G.S.; Writing—review & editing, K.J.; 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.

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

References

1. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 22–24 May 1996; pp. 212–219.
2. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
3. Aharonov, D.; Ben-Or, M. Fault-tolerant quantum computation with constant error. In Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, El Paso, TX, USA, 4–6 May 1997; pp. 176–188.
4. Shor, P.W. Fault-tolerant quantum computation. In Proceedings of the 37th Conference on Foundations of Computer Science, Burlington, VT, USA, 14–16 October 1996; pp. 56–65.
5. Steane, A.M. Efficient fault-tolerant quantum computing. *Nature* **1999**, *399*, 124–126. [CrossRef]

6. Ofek, N.; Petrenko, A.; Heeres, R.; Reinhold, P.; Leghtas, Z.; Vlastakis, B.; Liu, Y.; Frunzio, L.; Girvin, S.; Jiang, L.; et al. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **2016**, *536*, 441–445. [CrossRef] [PubMed]
7. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
8. Amy, M.; Di Matteo, O.; Gheorghiu, V.; Mosca, M.; Parent, A.; Schanck, J. Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3. In Proceedings of the Selected Areas in Cryptography–SAC 2016: 23rd International Conference, St. John's, NL, Canada, 10–12 August 2016; Revised Selected Papers; Springer: Cham, Switzerland, 2017; pp. 317–337.
9. Preston, R.H. Applying Grover's Algorithm to Hash Functions: A Software Perspective. *IEEE Trans. Quantum Eng.* **2022**, *3*, 1–10. [CrossRef]
10. Hey, T. Quantum computing: An introduction. *Comput. Control Eng. J.* **1999**, *10*, 105–112. [CrossRef]
11. Grassl, M.; Langenberg, B.; Roetteler, M.; Steinwandt, R. Applying Grover's algorithm to AES: Quantum resource estimates. In Proceedings of the Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, 24–26 February 2016; Springer: Cham, Switzerland, 2016; pp. 29–43.
12. Huang, Z.; Sun, S. Synthesizing Quantum Circuits of AES with Lower T-depth and Less Qubits. In Proceedings of the Advances in Cryptology—ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 5–9 December 2022; Proceedings, Part III; Springer: Cham, Switzerland, 2023.
13. Jang, K.; Baksi, A.; Kim, H.; Song, G.; Seo, H.; Chattopadhyay, A. Quantum analysis of AES. *Cryptol. Eprint Arch.* **2022**, 2022/683.
14. Song, G.; Jang, K.; Kim, H.; Seo, H. A Parallel Quantum Circuit Implementations of LSH Hash Function for Use with Grover's Algorithm. *Appl. Sci.* **2022**, *12*, 10891. [CrossRef]
15. Song, G.; Jang, K.; Kim, H.; Lee, W.K.; Hu, Z.; Seo, H. Grover on SM3. In Proceedings of the Information Security and Cryptology—ICISC 2021: 24th International Conference, Seoul, Republic of Korea, 1–3 December 2021; Revised Selected Papers; Springer: Cham, Switzerland, 2022; pp. 421–433.
16. Zou, J.; Li, L.; Wei, Z.; Luo, Y.; Liu, Q.; Wu, W. New quantum circuit implementations of SM4 and SM3. *Quantum Inf. Process.* **2022**, *21*, 181. [CrossRef]
17. Song, G.; Jang, K.; Kim, H.; Eum, S.; Sim, M.; Kim, H.; Lee, W.K.; Seo, H. SPEEDY Quantum Circuit for Grover's Algorithm. *Appl. Sci.* **2022**, *12*, 6870. [CrossRef]
18. Jang, K.; Choi, S.; Kwon, H.; Seo, H. Grover on SPECK: Quantum resource estimates. *Cryptol. Eprint Arch.* **2020**, 2020/640.
19. Almazrooie, M.; Samsudin, A.; Abdullah, R.; Mutter, K.N. Quantum reversible circuit of AES-128. *Quantum Inf. Process.* **2018**, *17*, 1–30. [CrossRef]
20. Jang, K.; Song, G.; Kim, H.; Kwon, H.; Kim, H.; Seo, H. Efficient implementation of PRESENT and GIFT on quantum computers. *Appl. Sci.* **2021**, *11*, 4776. [CrossRef]
21. Baksi, A.; Jang, K.; Song, G.; Seo, H.; Xiang, Z. Quantum implementation and resource estimates for rectangle and knot. *Quantum Inf. Process.* **2021**, *20*, 1–24. [CrossRef]
22. Anand, R.; Maitra, A.; Mukhopadhyay, S. Grover on SIMON. *Quantum Inf. Process.* **2020**, *19*, 340. [CrossRef]
23. Jang, K.; Baksi, A.; Breier, J.; Seo, H.; Chattopadhyay, A. Quantum implementation and analysis of Default. *Cryptol. Eprint Arch.* **2022**, 2022/647.
24. Chauhan, A.K.; Sanadhya, S.K. Quantum resource estimates of grover's key search on aria. In Proceedings of the Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, 17–21 December 2020; Springer: Cham, Switzerland, 2020; pp. 238–258.
25. Jang, K.; Song, G.; Kwon, H.; Uhm, S.; Kim, H.; Lee, W.K.; Seo, H. Grover on PIPO. *Electronics* **2021**, *10*, 1194. [CrossRef]
26. Rahman, M.; Paul, G. Grover on KATAN: Quantum resource estimation. *IEEE Trans. Quantum Eng.* **2022**, *3*, 1–9. [CrossRef]
27. Amy, M.; Maslov, D.; Mosca, M. Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2014**, *33*, 1476–1489. [CrossRef]
28. Devitt, S.J.; Stephens, A.M.; Munro, W.J.; Nemoto, K. Requirements for fault-tolerant factoring on an atom-optics quantum computer. *Nat. Commun.* **2013**, *4*, 2524. [CrossRef]
29. Dworkin, M.J. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
30. Dang, Q.H. *Secure Hash Standard*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
31. Penard, W.; van Werkhoven, T. On the secure hash algorithm family. *Cryptogr. Context* **2008**, 1–18.
32. Jang, K.; Choi, S.; Kwon, H.; Kim, H.; Park, J.; Seo, H. Grover on Korean block ciphers. *Appl. Sci.* **2020**, *10*, 6407. [CrossRef]
33. Jang, K.; Song, G.; Kim, H.; Kwon, H.; Kim, H.; Seo, H. Parallel quantum addition for Korean block ciphers. *Quantum Inf. Process.* **2022**, *21*, 373. [CrossRef]
34. Song, G.j.; Jang, K.b.; Seo, H.j. Resource Eestimation of Grover Algorithm through Hash Function LSH Quantum Circuit Optimization. *J. Korea Inst. Inf. Secur. Cryptol.* **2021**, *31*, 323–330.