Improved Low-depth SHA3 Quantum Circuit for Fault-tolerant Quantum Computers

Gyeongju Song[1], Kyungbae Jang[1], and Hwajeong Seo[1]

IT Department, Hansung University, Seoul (02876), South Korea
thdrudwn98@gmail.com,starj1023@gmail.com,hwajeong84@gmail.com

Abstract. To build an efficient security system in the post-quantum era, it is possible to find the minimum security parameters for defending a fault-tolerant quantum computer by estimating the quantum resources required for an quantum attack. In a fault-tolerant quantum computer, errors must reach an acceptable level through error detection and error correction, which additionally uses quantum resources. As the depth of the quantum circuit increases, the computation time per qubit increases, and errors in quantum computers increases. Therefore, in terms of errors in quantum circuits, it is appropriate to reduce the depth by increasing the number of qubits. This paper proposes an low-depth quantum circuit implementations of SHA3 for fault-tolerant quantum computers to reduce errors. The proposed SHA3 quantum circuit is implemented in the direction of reducing the quantum circuit depth through a trade-off between the number of qubits, quantum gate, and quantum depth in each function. Compared to the-state-of-art works, proposed method decreased T-depth and Full-depth by 30.3% and 80.05%, respectively. We expect that this work will contribute to the establishment of minimum security parameters for SHA3 in the quantum era.

Keywords: Quantum implementation · Grover algorithm · SHA3

1 Introduce

As the world evolves into the information age, data encryption is essential to protect digital data. Currently, digital data is protected through symmetric key cryptography (e.g. Advanced Encryption Standard (AES)) and public key cryptography (e.g. Riverst-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC)). With the unexpected rapid development of quantum computers, the safety of existing cryptography is unclear. Grover's algorithm, proposed by Lov Grover in 1996 [6] is known to accelerate brute-force attack and pre-image attack on symmetric key cryptography and hash functions. Shor's algorithm proposed by Peter Shor in 1994 [12] is known to be able to solve basic problems of existing public key cryptography, such as factorization and discrete logarithms, in polynomial time.

In the past, quantum computers were an abstract concept, but many companies have actually realized fault-tolerant quantum computers, showing the possibility that they can exist and operate rather than being abstract. To operate

a quantum computer, it is necessary to achieve certain quantum resources (i.e. number of qubits, quantum gates) required for operation. When valid quantum resources meet the requirements for a target cryptographic attack, the cryptography is considered to be no longer effective in protecting information. In order to cope with this, symmetric key cryptography and hash function can maintain security strength by increasing key and hash length, but it is expected that public key cryptography should be replaced with other post-quantum cryptography. It is also important to establish an efficient security system for symmetric key cryptography and hash functions by finding the minimum security parameters safe for fault-tolerant quantum computers. This can be found 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 quantum mechanical phenomena of qubits, it is important to preserve the integrity of the qubit state in the operation. Data can be corrupted when the value of an unstable qubit fluctuates during calculation due to noise. To operate a fault-tolerant quantum computer that can tolerate appropriate errors, error correction and error detection for unstable qubits are essential, and various research is being conducted for these tasks [1, 13, 16, 10].

The number of qubits and the quantum circuit depth are generally inversely proportional to each other in a quantum circuit implementation. In implementing a quantum circuit, two methods can be considered: reducing the depth of a quantum circuit by increasing the number of qubits and reducing the number of qubits by increasing the depth of a quantum circuit. In the era of noisy intermediate-scale quantum (NISQ), quantum computers are not yet a solid technology [11], so it's difficult to name which is the more efficient and optimized in the 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 of the quantum circuit increases, which affects the increase in the error rate of the quantum circuit.

With this research motivation, this paper presents an improved low-depth SHA3 quantum circuit for fault-tolerant quantum computers to reduce errors. The method of returning the qubit state to its previous state through an inverse operation and reusing it in the next operation can reduce the number of qubits. This approach increases errors as the inverse of the function increases depth, so more resources are used for error detection and error correction. In this paper, a quantum circuit was implemented by increasing the number of qubits and reducing the inverse operation process, also a separate attempt was made to reduce the number of qubits. 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 is a very efficient quantum circuit in terms of depth of quantum circuit.

As a result, compared to previous works [2], T-depth and Full-depth are reduced by 30.3% and 80.05%, respectively.

1.1 Contribution

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, and as a result, we reduced the T-depth and full depth than previous works by 30.3% and 80.05%, respectively. The contributions to the proposed SHA3 quantum circuit are summarized as follows:

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

In terms of quantum cost (resource trade-off) In a fault-tolerant quantum computer, error detection and error correction are essential to control errors that accumulate through the 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 shows the result of drastically reducing the depth through quantum resource trade-off.

Preparing for the Post-quantum era To prepare for the post-quantum era, it is necessary to find parameters that satisfy security by estimating the resources required for an attack through the implementation of a quantum circuit for the target cryptography. Here, finding the minimum security parameters is effective for an efficient security system. The proposed new direction of SHA3 quantum circuit implementation can 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 is 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. That means being 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 one of the eigenvalues of 0 or 1 after measurement, and it is not known which one it will be before measurement[7]. α and β mean the probability amplitude, and if $|\alpha|^2$ is 0, $|\beta|^2$ is 1, and vice versa holds.

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 quantum circuits, 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 quantum gates used in operation[5, 8, 9, 15, 14, 17]. Representative quantum gates include H gate, X gate, CNOT gate, Toffoli gate and T gate as follows:

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

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

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 is used for T-depth estimation. The Toffoli gate includes a non-Clifford T gate, and the steps of the T gate lead to T-depth. Minimizing the number of T gates is still important, as non-Clifford T gates have long latency and implementation cost far exceeds that of Clifford gates in fault-tolerant implementation [3, 4].

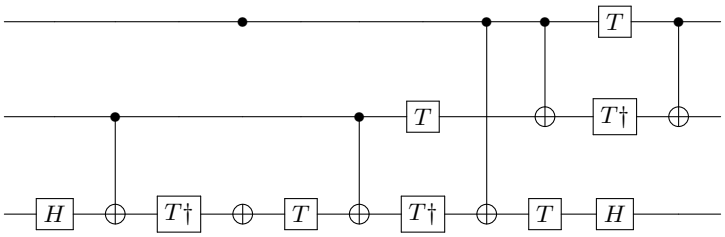


Fig. 1: Decomposed Toffoli gate

2.2 Secure Hash Algorithm(SHA)-3

In 2015, the National Institute of Standards and Technology (NIST) released Secure Hash Algorithm (SHA3) to replace SHA1 and SHA2. The SHA3 hash function family consists of four hash functions: SHA3-224, SHA3-256, SHA3-384, SHA3-512, and two extendable-output functions (XOF): SHAKE128, SHAKE256. The input data outputs hash results through 'absorbing' and 'squeezing' steps by the sponge structure. SHA3 has a sponge structure, so it outputs a hash value of a constant length regardless of the input length. In 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 with the implementation for SHA3 quantum circuit.

Pre-image attack A hash function maps data of an arbitrary length to a hash value of a fixed length. This feature increases the speed of data search as long and diverse data can be arranged in a certain length. Pre-image is a way to find the original message when a hash value is given: find M given H for $H = hash(M)$. A preimage 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 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. The following Table 1 shows the parameters and pre-image resistance for SHA3 hash function family.

Algorithm	H	r	c	Pre-image
SHA3-224	224	1,152	448	224
SHA3-256	256	1,088	512	256
SHA3-384	384	832	768	384
SHA3-512	512	576	1,024	512
SHAKE128	d	1,344	256	$\geq \min(d,128)$
SHAKE256	d	1,088	512	$\geq \min(d,256)$

Table 1: Parameters and pre-image resistance for SHA3 hash function family. (H : hash length, r : block size, c : capacity)

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 it 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 Oracle and Diffusion operators as shown in Figure 2. This attack is a known plaintext attack (KPA) that proceeds when the plaintext-ciphertext pairs of the block cipher are known. Inside the Oracle function, including 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.

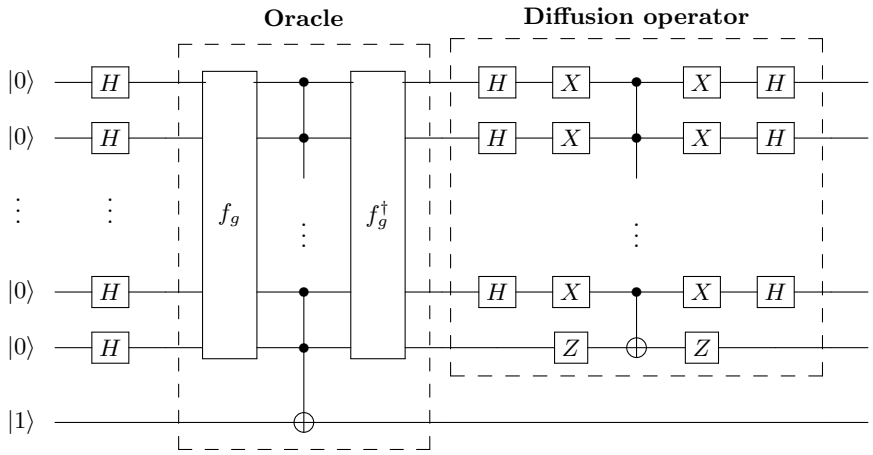


Fig. 2: Grover algorithm with $f_g : \{0,1\}^n \leftarrow \{0,1\}^n$ in oracle.

3 SHA3 Quantum Circuit

The data input to the qubit outputs 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 is implemented for all Keccak-f phases. This section describes the implementation of quantum circuits for each function. The SHA3 internal function f consists of 5 steps as follows 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 SHA-3 uses a 64-bit word (i.e. l=6) :

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

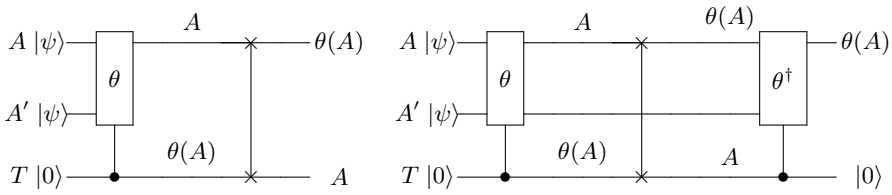
In the SHA3 operation, each step of Keccak-f proceeds with a multi-dimensional bit array structure of data. In the same way, the quantum circuit was constructed assuming that the qubits were arranged in a multi-dimension bit array along the structure of each step.

Theta($\boldsymbol{\theta}$) Theta(θ) is one of the five phases of the SHA-3 (Keccak-k) hash function. In the θ phase, the data is processed in the 3-dimensional state array

structure.The result of $\Sigma((x-1),z) \oplus \Sigma((x+1),(z-1))$ saves to (x,y,z) bits. That is, the final result value is stored in (x,y,z).

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

Equation 1 is 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 1,600-bit *temp* are used. Quantum circuits reduce the use of 4,800 qubits by allocating one 1,600-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 a temp qubit to replace the inverse operation per round.



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

Fig. 3: Quantum circuit for θ.(T: temporary qubits)

Figure 1 shows (a)Excluding inverse operation and (b)Including inverse operation in theta(θ). T represents the *temp* qubit. Excluding inverse operation: includes only one θ function in the quantum circuit per round. Including inverse operation: include the θ function and the $\theta^\dagger$ function to return the temporary qubit T to its original state in quantum circuit per round. Including inverse operation can reduce T qubits, but increases the number of quantum gates and depth. In an attempt to reduce the depth, Excluding inverse operations was selected in this paper. Compared to [2], the operation process proposed in this paper increases the CNOT gate by about 36.36% and reduces the depth by about 71.27% in the θ (Trade-off between quantum gates and depth). In the implementation of [2], $\theta^\dagger$ uses the most CNOT gates and increases the depth. About this, the proposed quantum circuit replaces $\theta^\dagger$ with the use of T qubits. As a result, a trade-off occurs between 1,360,000 CNOT gates+25 depth and 1,600 qubits at $\theta^\dagger$ per round(increase: qubit, decrease: CNOT gate+depth).

Algorithm 1 shows the operation of our quantum circuit for the theta(θ). In the input, X and T denote the input qubit and the *temp* qubit. All operations

performed by the CNOT gate update T, and T is returned at the end. Compared to the previous research result[2], the proposed algorithm increases the CNOT gate but reduces 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

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

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

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

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, there is no reversible quantum gate used in the Rho(ρ).

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($\rho$), there is no reversible quantum gate used in the Pi(π) phase because it only changes the physical position of the qubit.

Chi(χ) Chi(χ) is the only non-linear part in Keccak-f. Looking at the results of quantum circuit implementation, the Toffoli gate was only used in this step. Therefore, since it is the only internal step to use the T gate, it has the T depth. Chi(χ) is the process of XOR operation with the result of multiplying the right two bits in 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 proposed quantum circuits, 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.

Order	**Required qubit**		**Update target**
$x = 0$	$X[1][y][z]$	$X[2][y][z]$	$X[0][y][z]$
$x = 1$	$X[2][y][z]$	$X[3][y][z]$	$X[1][y][z]$
$x = 2$	$X[3][y][z]$	$X[4][y][z]$	$X[2][y][z]$
$x = 3$	$X[4][y][z]$	$\circledast$ $X[0][y][z]$	$X[3][y][z]$
$x = 4$	$X[0][y][z]$	$\circledast$ $X[1][y][z]$	$X[4][y][z]$

Table 2: The qubit values required to step-by-step update the input of Chi(χ), $\circledast$ means that it has changed in the preceding calculation.

In order $x = 0, 1, 2$, there is no problem in updating X, but in $x = 3, 4$, a problem arises because the state of the qubits of X' and X required for the operation is changed in the preceding operation. (marked with $\circledast$ in the Table 2). A method using inverse operation can be considered, but this greatly increases the depth of the quantum circuit. The proposed quantum circuit allocates qubits to store the values of $X[0][y][z]$ and $X[1][y][z]$ before operation and maintains the values. For each round in Chi(χ), this method reduces CNOT gate: about 98.08%, T-depth: about 30.3%, and Full-depth: about 90.08% by using 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

$x[0] \leftarrow \mathbf{CNOT}(X[0], T_0)$
$x[1] \leftarrow \mathbf{CNOT}(X[1], T_1)$

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

  return x

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 is classic data and the input is quantum data, X gate is performed to T according to the RC value. In this way, the quantum resources required for RC calculation can be reduced and the use of the CNOT gate can be replaced with the X gate. (CNOT gate is regarded as a higher-cost quantum resource than X gate). The quantum resource (X gate) used in Iota(ι) depends on the constant RC.

3.1 Quantum cost analysis for SHA3

Table 3 shows our quantum resources for Keccak-f function in SHA3, and Table 4 shows quantum resources for our result and Amy et al[2] result for comparison with previous research.

As shown in Table 3, in the proposed quantum circuit, θ and χ use the most quantum resources, and 1,600 qubits in θ and 640 qubits in χ are used for each round. The proposed quantum circuit increases the number of qubits to reduce the depth of the quantum circuit, resulting in a reduced depth of each function compared to previous implementations. In the theta operation, we increased per round the CNOT gate by 36.36% and reduced the depth by 71.27%. In addition, to omit the θ^{-1} process, 1,600 additional qubits were used and a trade-off was made to reduce 1,360,000 CNOT gate + 25 depth. A total of Full-depth is reduced by about 73.67% in theta(θ). In chi(χ), 640 qubits were

used to reduce CNOT gate: about 98.08%, T-depth: about 30.3%, and Full-depth: about 90.08% per round(trade-off between qubit and gate+depth). For the operation of the iota(ι), the classic-to-quantum method reduces the quantum resources required for RC calculation and replaces the use of CNOT gate with X gate.

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

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	3,200	640	1,600	23	12
ι	2	0	0	0	1
Round	3,202	24,640	1,600	23	88
Total	76,886	591,360	38,400	552	2,020

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)

4 Conclusion

This paper proposed an improved low-depth SHA3 quantum circuit 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, and quantum resources 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 both sides, but since quantum computers are currently an unclear technology, it is difficult to name which one is more efficient. As the quantum circuit depth increases, the computation time for each qubit increases, which increases the error. From a noise perspective, it makes more sense to increase the number of qubits and reduce the quantum circuit depth to reduce errors. In this paper, we worked to reduce the quantum circuit depth to reduce errors occurring in cryptography operations. Quantum circuits

Function		#1qClifford	#CNOT	#Toffoli	#T-depth	#Full Depth
θ	Our	0	24,000	0	0	79
	In [2]	0	17,600	0	0	275
θ^{-1}	Our	0	0	0	0	0
	In [2]	0	1,360,000	0	0	25
ρ	Our	(Not used)				
	In [2]					
π	Our	(Not used)				
	In [2]					
χ	Our	3,200	640	1,600	23	12
	In [2]	0	14,400	(Not shown)	15	55
χ^{-1}	Our	0	0	0	0	0
	In [2]	0	18,880	(Not shown)	18	66
ι	Our	2	0	0	0	1
	In [2]	85	0	0	0	24
Total		76,886	591,360	38,400	552	2,020
Total [2]		85	33,269,760	-	792	10,128

Table 4: Comparison of quantum resources for the proposed SHA3 quantum circuit and the SHA3 quantum circuit in [2].

were implemented in the direction of reducing the depth through a trade-off between the number of qubits and 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 attempts will contribute to the establishment of minimum security parameters for SHA3 in the post-quantum era.

References

1. 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. pp. 176–188 (1997)
2. 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: Selected Areas in Cryptography–SAC 2016: 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers. pp. 317–337. Springer (2017)
3. Amy, M., Maslov, D., Mosca, M.: Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **33**(10), 1476–1489 (2014)
4. Devitt, S.J., Stephens, A.M., Munro, W.J., Nemoto, K.: Requirements for fault-tolerant factoring on an atom-optics quantum computer. Nature communications **4**(1), 2524 (2013)
5. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying Grover's algorithm to AES: quantum resource estimates. In: Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings 7. pp. 29–43. Springer (2016)
6. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
7. Hey, T.: Quantum computing: an introduction. Computing & Control Engineering Journal **10**(3), 105–112 (1999)
8. Huang, Z., Sun, S.: Synthesizing quantum circuits of AES with lower t-depth and less qubits. Cryptology ePrint Archive (2022)
9. Jang, K., Baksi, A., Kim, H., Song, G., Seo, H., Chattopadhyay, A.: Quantum analysis of AES. Cryptology ePrint Archive (2022)
10. 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 **536**(7617), 441–445 (2016)
11. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018)
12. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp. 124–134. Ieee (1994)
13. Shor, P.W.: Fault-tolerant quantum computation. In: Proceedings of 37th conference on foundations of computer science. pp. 56–65. IEEE (1996)

14. Song, G., Jang, K., Kim, H., Lee, W.K., Hu, Z., Seo, H.: Grover on SM3. In: Information Security and Cryptology–ICISC 2021: 24th International Conference, Seoul, South Korea, December 1–3, 2021, Revised Selected Papers. pp. 421–433. Springer (2022)
15. Song, G., Jang, K., Kim, H., Seo, H.: A parallel quantum circuit implementations of LSH hash function for use with Grover's algorithm. Applied Sciences **12**(21), 10891 (2022)
16. Steane, A.M.: Efficient fault-tolerant quantum computing. Nature **399**(6732), 124–126 (1999)
17. Zou, J., Li, L., Wei, Z., Luo, Y., Liu, Q., Wu, W.: New quantum circuit implementations of SM4 and SM3. Quantum Information Processing **21**(5), 181 (2022)