Optimized Quantum Circuit for Quantum Security Strength Analysis of Argon2

Gyeongju Song[1], Siwoo Eum[1], Hyeokdong Kwon[1], Minjoo Sim[1], Minwoo Lee[2], and Hwajeong Seo[2]

[1]Department of Information Computer Engineering,
Hansung University, Seoul (02876), South Korea,
[2]Department of Convergence Security,
Hansung University, Seoul (02876), South Korea,
{ thdrudwn98, shuraatum, korlethean, minjoos9797, minunejip, hwajeong84}@gmail.com

Abstract. This paper explores the optimization of quantum circuits for Argon2, a memory-hard function used for password hashing and other applications. With the rise of quantum computers, the security of classical cryptographic systems is at risk. It emphasizes the need to accurately measure the quantum security strength of cryptographic schemes using optimized quantum circuits. The proposed method focuses on two perspectives: qubit reduction (qubit optimization) and depth reduction (depth optimization). The qubit-optimized quantum circuit was designed to find a point where an appropriate inverse is possible and reuses the qubit through the inverse to minimize the number of qubits. The start point and end point of the inverse are set by finding a point where qubits can be reused with minimal computation. The depth-optimized quantum circuit reduces the depth by using the minimum number of qubits as necessary without performing an inverse operation. The trade-off between qubit and depth is confirmed by modifying the internal structure of the circuits and the quantum adders. Qubit optimization achieved up to a 12,229 qubit reduction, while depth optimization resulted in approximately 196,741 (approximately 69.02%) depth reduction. In conclusion, this research demonstrates the importance of implementing and analyzing quantum circuits from various optimization perspectives. The results contribute to the post-quantum strength analysis of Argon2 and provide valuable insights for future research on quantum circuit design, considering the appropriate trade-offs of quantum resources in response to advancements in quantum computing technology.

Keywords: Quantum Implementation · Quantum Computing · Quantum Circuit Optimization · Argon2

1 Introduction

Quantum computers have gained attention for their ability to solve specific problems faster than classical computers due to the properties of qubits. The emergence of large-scale quantum computers is anticipated to pose a threat to existing

cryptographic systems. In 1994, Peter Shor proposed an algorithm[16] capable of efficiently solving fundamental problems in public-key cryptography, such as integer factorization and discrete logarithms, thereby compromising the security of public-key cryptography. Consequently, the security of target public-key cryptography is no longer guaranteed when large-scale quantum computers capable of performing specific cryptographic attacks appear. In 1996, Lov Grover introduced an algorithm[8]. This algorithm can accelerate brute-force attacks and pre-image attacks on symmetric-key cryptography and hash functions. As a result, it achieves a computational complexity of $O(\sqrt{2^n})$ for finding specific data in unsorted n-bit data. To counter this, the length of the encryption key (hash output length) can be doubled to maintain resistance. However, classical computers and quantum computers differ in their operation, required resources, and feasible computations, making the security strength of classical computers not directly correspond to the quantum security strength of quantum computers. Accurately measuring the quantum security strength in the context of quantum computers requires optimizing the necessary operations of the specific cryptographic scheme using quantum circuits and accurately verifying the utilized quantum gates and circuit depth. In previous studies, ciphers were implemented as quantum circuits, and required quantum resources were estimated[7, 1, 2, 5, 17, 12, 4, 14, 10, 15, 13, 9, 11, 20, 19, 22, 18, 21].

Optimization of quantum circuits can be pursued from two perspectives: reducing the number of qubits and minimizing the circuit depth, with qubit and depth being inversely proportional in each implementation. While the number of physically implemented qubits is important in quantum circuit operation, in the Noisy Intermediate-Scale Quantum (NISQ) era, reducing depth, which helps mitigate errors, is crucial to obtaining desired quantum computing results. As the depth of quantum circuits increases, the computation time also increases, which is influence the error rate of qubits. In other words, as the depth of quantum circuits increases, the error rate of qubits also increases.

With this research motivation, this paper proposes two perspectives of optimized quantum circuits for Argon2 and presents estimations of the required quantum resources. The optimization perspectives in quantum circuit implementation involve attempts to reduce the number of qubits and the circuit depth. Each quantum circuit is divided into qubit optimization implementation and depth optimization implementation for specific operations. To further analyze this, we modify the internal addition to examine the trade-off between qubits and depth and make efforts to find the most optimized quantum circuit for qubits and depth, respectively. In evaluation confirm and analyzes the estimations of the required quantum resources for qubit optimization and depth optimization quantum circuits in relation to this. In a qubit-optimized implementation, we find and set points where inverse operations are possible, and continue to use reusable qubits. In the depth optimization implementation, the minimum number of qubits required for computation is allocated and used without including inverse computation. In addition, an attempt was made to further reduce the depth by changing the adder structure to parallel operation.

As a result of optimizing the quantum circuit from both perspectives, the qubit-optimized quantum circuit reduced up to 12,229 qubits, and the depth-optimized quantum circuit showed a maximum 196,741 (approximately 69.02%) depth reduction.

The structure of this paper is as follows: In Section 2, related research on quantum computers, Grover algorithm, and Argon2 was written to help understand the paper, and Section 3 describes the implementation of the proposed Argon2 quantum circuit. Section 4 estimates and analyzes the resources required for the proposed quantum circuit. Finally, Section 5 concludes the paper with a conclusion.

2 Background

2.1 Quantum computer

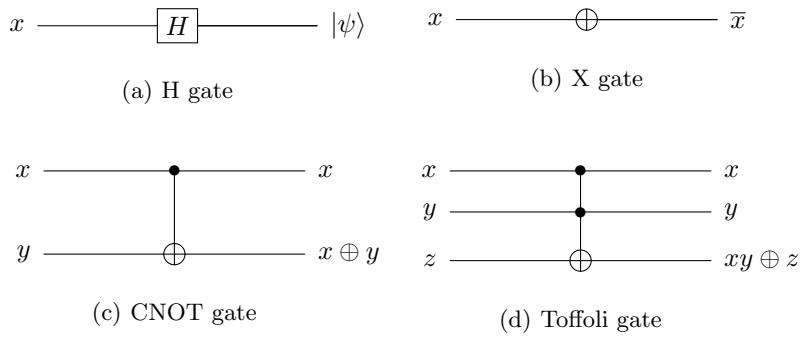
Quantum computers process data using quantum mechanical phenomena of qubits. These quantum computers can express and process 2^n data at once with n qubits due to the superposition and entanglement properties of qubits, enabling faster calculations than classic computers. Qubits are controlled through quantum gates, and because of the reversible nature of quantum gates, inverse operations are possible. The following shows H, X, CNOT, and Toffoli matrices among representative quantum gates that control qubits:

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

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The quantum gate operation of each gate is shown in Figure 1.

(a) **H gate**: The H gate works with a single qubit and makes the input a superposition.
(b) **X gate**: The X gate works with a single qubit and reversed the input.
(c) **CNOT gate**: The CNOT gate works with two qubits: control qubit and target qubit. The state of the target qubit y is reversed when the control qubit x is one.
(d) **Toffoli gate**: The Toffoli gate works with three qubits: two control qubits and one target qubit. The state of the target qubit z is reversed when the control qubits x and y are both one.



(a) H gate (b) X gate

(c) CNOT gate (d) Toffoli gate

Fig. 1: Quantum gates

2.2 Grover algorithm

Quantum computers have the potential to significantly improve the efficiency of certain computational tasks compared to classical computers. One such task is searching for specific n-bit data in an unsorted list, where classical algorithms typically require $O(2^n)$ operations. However, by employing the Grover algorithm on a quantum computer, the search complexity can be reduced to $O(\sqrt{2^n})$. The Grover algorithm for pre-image attacks consists of two main components: an Oracle and a Diffusion operator shown in Figure 2. This is designed for known-plaintext attacks (KPA) in block ciphers(hash functions), where both the plaintext-ciphertext pairs are known. The Oracle function includes both the hash function $f_g(x) = y$ and its inverse operation $f_g^\dagger(x) = y$. When the result of $f_g(x)$ matches the target hash value y, the Oracle sets $x = 1$. The Diffusion operator $U_s = 2|s\rangle\langle s| - I$ is then applied to enhance the probability of observing this state. Through approximately $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor$ iterations of the Grover algorithm, the probability of measuring the correct solution qubit can be significantly increased.

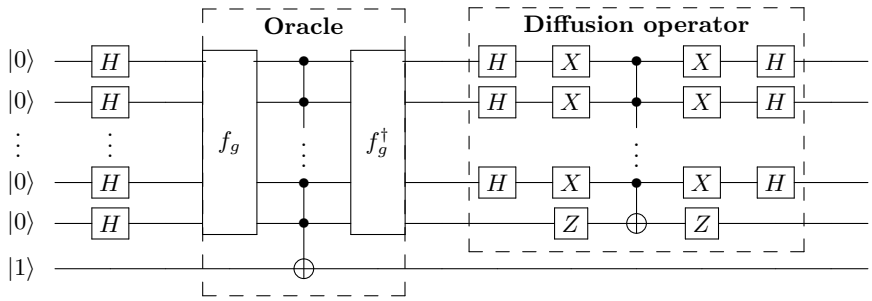


Fig. 2: Grover algorithm with $f_g : \{0,1\}^n \to \{0,1\}^n$

2.3 Argon2: a memory-hard function for password hashing and other applications

Argon2 is a key derivation function that won the 2015 Password Hashing Competition. Argon2, a memory-hard function for password hashing and other applications, can be used to hash for credential storage, key derivation, or other applications. It has a simple design that targets fast fill rates of memory and effective use of multiple computing devices while providing defense against trade-off attacks. Argon2 offers three variants: Argon2d, Argon2i and Argon2id, each variant has the following characteristics:

1. **Argon2d**: Argon2d uses fast, data-dependent memory accesses, making it highly resistant to GPU cracking attacks and suitable for applications where side-channel timing attacks are not the threat.
2. **Argon2i**: Argon2i uses data-independent memory access, but is slower as it uses more memory to protect against trade-off attacks (suitable for password hashing and cipher-based key derivation)
3. **Argon2id**: Argon2id is a hybrid of Argon2i and Argon2d, using a combination of data-dependent and data-independent memory accesses, giving Argon2i some resistance to side-channel cache timing attacks and most of Argon2d's resistance to GPU cracking attacks.

Figure 3 shows the operation of Argon2. Argon2 has two types of inputs: Primary inputs and Secondary inputs or parameters. The Primary inputs are: message P and nonce S, Secondary inputs are: Degree of parallelism p (integer value from 1 to $2^{24}-1$), Tag length τ (integer number of bytes from 4 to $2^{32}-1$), Memory size m (integer number of kilobytes from $8p$ to $2^{32}-1$), Number of iterations t (integer number from 1 to $2^{32}-1$), Version number v (one byte 0x13), Secret value K (length from 0 to $2^{32}-1$ bytes.), Associated data X (length from 0 to $2^{32}-1$ bytes.), Type y of Argon2 (Argon2d: 0, Argon2i: 1, Argon2id: 2)

2.4 Compression function G

The compression function G used in Argon2 is based on the round function P of Blake2b[3]. P operates on eight 16-byte registers (128-bit) inputs. The compression function $G(X, Y)$ works with two 1024-byte blocks X and Y. After first calculating $R = X \oplus Y$, R is defined as 16-byte registers R_0 to R_{63}. Then, apply to P in row-wise and column-wise order to obtain Z.

$$
\begin{gathered}
(Q_0, Q_1, \cdots, Q_7) \leftarrow P(R_0, R_1, \cdots, R_7) \\
(Q_8, Q_9, \cdots, Q_{15}) \leftarrow P(R_8, R_9, \cdots, R_{15}) \\
\cdots \\
(Q_{56}, Q_{57}, \cdots, Q_{63}) \leftarrow P(R_{56}, R_{57}, \cdots, R_{63})
\end{gathered}
$$

$$
(Z_0, Z_8, Z_{16}, \cdots, Z_{56}) \leftarrow P(Q_0, Q_8, Q_{16}, \cdots, Q_{56})
$$

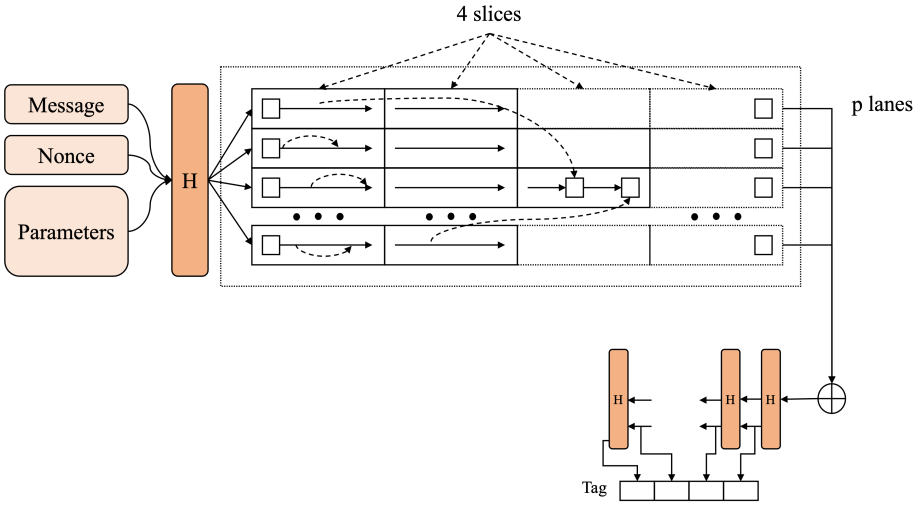


Fig. 3: Operation process of Argon2

$$(Z_1, Z_9, Z_{17}, \cdots, Z_{57}) \leftarrow P(Q_1, Q_9, Q_{17}, \cdots, Q_{57})$$
$$\cdots$$
$$(Z_7, Z_{15}, Z_{23}, \cdots, Z_{63}) \leftarrow P(Q_7, Q_{15}, Q_{23}, \cdots, Q_{63})$$

The operation of the compression function G is shown in Figure 4.

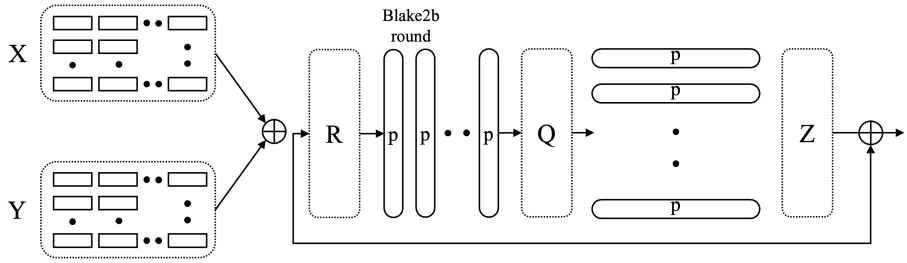


Fig. 4: Operation process of compression function G

Finally, G outputs the result of $Z \oplus R$.

$$G : (X, Y) \rightarrow R = X \oplus Y \rightarrow Q \rightarrow Z \rightarrow Z \oplus R$$

3 Propose Method

This paper proposes quantum circuits for Argon2 and estimates the required quantum resources for their operation.

This section provides a detailed explanation of the operation of the proposed quantum circuits. For the implementation of quantum circuits for Argon2, we divided the approach into two perspectives: qubit reduction (qubit optimization) and depth reduction (depth optimization). The qubit-optimized quantum circuit adopts a strategy of reusing qubits through inverse operations, while the depth-optimized quantum circuit increases the number of qubits without using inverse operations to maintain the same level of computations. Furthermore, the depth-optimized quantum circuit reduces the depth by employing a parallel design of addition with additional temporary qubits. We set the inverse point in a suitable location for this task. The operating point and starting point of the inverse are set by finding a point where qubits can be reused with minimal computation.

Regarding the quantum adders used within the compression function, we apply both the $(6n-2)$-depth adder (referred to as Ripple in this paper) and the $(2n+3)$-depth adder (referred to as Simple in this paper) proposed in [6]. Each adder is applied to both qubit and depth-optimized quantum circuits, allowing for the examination of the trade-off between qubit and depth quantum resources. In both perspectives, we adopt a common operation called Classic to Quantum, where the X-gate operation is applied to quantum data based on the positions where the corresponding classic data has an index of 1. This operation is employed to reduce the number of qubits. To reduce quantum circuit depth, the Shift operation is performed by changing the indices of the array rather than using Swap gates.

Figure 5 shows (1)Qubit-optimized quantum circuit and (2)Depth-optimized quantum circuit. For the two circuits in the figure, input m is pre-determined classic data, *sigma* is pre-determined quantum data, and $|a\rangle$ to $|d\rangle$ is the quantum data input to G. Detailed explanations of the two quantum circuits are provided in Section 3.1 and Section 3.2. The order in which $|a\rangle$ to $|d\rangle$ is input to G is as follows:

$$
\begin{aligned}
G(a,b,c,d) &= G(v_0, v_4, v_8, v_{12}) \\
G(a,b,c,d) &= G(v_1, v_5, v_9, v_{13}) \\
G(a,b,c,d) &= G(v_2, v_6, v_{10}, v_{14}) \\
G(a,b,c,d) &= G(v_3, v_7, v_{11}, v_{15}) \\
G(a,b,c,d) &= G(v_0, v_5, v_{10}, v_{15}) \\
G(a,b,c,d) &= G(v_1, v_6, v_{11}, v_{12}) \\
G(a,b,c,d) &= G(v_2, v_7, v_8, v_{13}) \\
G(a,b,c,d) &= G(v_3, v_4, v_9, v_{14})
\end{aligned}
$$

3.1 Qubit-optimized quantum circuit

Qubit-optimized quantum circuit reuses qubits through reverse operation, increasing the depth at the cost of reducing the number of qubits. This method

reuses the used 64-qubit sigma through inverse operation so that all compression functions operate as a single 64-qubit sigma. The quantum circuit for this can be seen in (1) of Figure 5. In this circuit, there are two reverse points to reduce the number of qubits, and the *sigma* is reset to $|0\rangle$ at both points. Allocated qubits for *sigma* are not only reused in functions but are still available in all rounds. Including the inverse operation, the quantum data $|a\rangle$ to $|d\rangle$ are updated according to the order.

Algorithm 1 shows the operation of the Qubit-optimized quantum circuit for the compression function G. Lines 3 and 19 are Reverse Points, and lines 6 and 22 indicate the timing of the reverse operation of each Reverse Point. In lines 2, 5, 12, 18, 21, and 26, ADD is implemented using two adders: depth $6n-2$ adder and depth $2n+3$ adder, and the difference between each adder is shown in Section 4. The depth was not increased by adjusting the operation index order instead of shift, and the depth was reduced by adjusting the physical location of qubits using a logical array instead of a SWAP gate. The Classic_to_Quantum function in lines 4 and 20 is designed so that m and Sigma are not quantum-to-quantum operations between qubits, but classic-to-quantum operations according to the state of classic constant values. This approach allows for reducing the number of qubits and quantum gates used for m and sigma updates. Since constant m is a known constant, m is stored in the pre-computation table, and the X gate is operated at the same *sigma* index as the part where the index bit value of m is one in each round. These operations are also very efficient in terms of quantum resources as they can be replaced with the use of a low-cost X gate than the CNOT gate.

3.2 Depth-optimized quantum circuit

Depth-optimized quantum circuits increase the use of temp qubits but decrease the depth. The 64-qubit *sigma* used is not reused, it is allocated and used whenever *sigma* is used in any function. The quantum circuit for this can be seen in (2) of Figure 5. Since there are no reverse actions, there is no reverse point. The qubits assigned to *sigma* are non-reusable, so it continues to be assigned in all rounds, not just in the function. Without including the inverse operation, the quantum data $|a\rangle$ to $|d\rangle$ are updated according to the order.

Algorithm 2 shows the pseudo-code for a depth-optimized quantum circuit for compression function G. In lines 2, 4, 10, 16, 18, and 22, ADD is implemented using two adders: depth $6n-2$ adder and depth $2n+3$ adder, and the difference between each adder is shown in Section 4. The depth was not increased by adjusting the operation index order instead of shift, and the depth was reduced by adjusting the physical location of qubits using a logical array instead of a SWAP gate. The **Classic_to_Quantum** function in lines 3 and 17 is designed so that m and Sigma are not quantum-to-quantum operations between qubits, but classic-to-quantum operations according to the state of classic constant values. This method does not involve an inverse operation, allowing the total depth to be reduced. As with qubit-optimized quantum circuits, the known constant m is stored in the pre-computation table, and the X gate is operated at the same

Sigma index as the part where the index bit value of m is one in each round. These operations are also very efficient in terms of quantum resources as they can be replaced with the use of a low-cost X gate than the CNOT gate.

4 Evaluation

This paper proposed two perspectives of quantum circuits for Argon2, focusing on qubit reduction and depth reduction. The qubit-optimized quantum circuit reduces the number of qubits by reusing them through inverse operations, while the depth-optimized quantum circuit increases the number of qubits without using inverse operations to maintain the same level of computations. Additionally, the internal structure of the quantum circuits is modified to explore the optimal design for qubit and depth, analyzing the trade-off between the two. Four distinct quantum circuits are presented, representing the two optimization perspectives and two variations of adder circuits.

The estimated results are shown in Tables 1, 2, 3, 4, and 5. Table 1 provides estimation results of quantum resources for each optimized function in Argon2. Qubit Opt and Depth Opt represent the qubit-optimized and depth-optimized quantum circuits, while Ripple and Simple refer to the adders proposed in [6] with $(2n+3)$-depth and $(6n-2)$-depth, respectively. The results show that Qubit-optimized G operates with 1,089 qubits, while Depth-optimized G operates with 13,318 qubits, demonstrating a reduction of up to 12,229 qubits through qubit optimization. The depth per round for Qubit-optimized is 74,713 and 220,033, depending on the adder used, while the depth per round for Depth-optimized G is 23,401 (approximately 68.68% reduction) and 68,163 (approximately 69.02% reduction) depending on the adder. This indicates a potential reduction of up to approximately 69.02% in depth through depth optimization.

Operation	Adder	#Qubit	#1qClifford	#CNOT	#Toffoli	#Full Depth
G (Qubit Opt)	Ripple	1,089	70,836	204,864	72,000	74,713
G (Qubit Opt)	Simple	1,089	22	172,032	72,576	220,033
G (Depth Opt)	Ripple	13,318	70,284	204,864	72,000	23,401
G (Depth Opt)	Simple	13,318	12	172,032	72,576	68,163
$Z \oplus R$	-	1,536	-	1,024	-	2

Table 1: Estimation results of quantum resources for each optimized function in Argon2. The result is a measure of the amount of resources per round. (Qubit Opt: Qubit-optimized quantum circuit, Depth Opt: Depth-optimized quantum circuit)

Tables 2, 3, 4, and 5 present the estimation of quantum resources for each step of Argon2. Among the steps, blake2b utilizes the highest amount of resources.

In summary, selecting the qubit-optimized quantum circuit can reduce the number of qubits by up to 12,740, with the flexibility to choose the adder based on optimization needs. Choosing the depth-optimized quantum circuit can reduce the depth by up to approximately 89.59%, with the possibility to select the adder based on the optimization perspective. The results of this paper indicate that selecting the Ripple adder in the depth-optimized quantum circuit minimizes the depth.

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	#Full Depth
Initial	1,090	(None)			
Update		(None)			
Final		1.62×2^{17}	1.17×2^{19}	1.64×2^{17}	1.71×2^{17}
blake2b		1.51×2^{22}	1.1×2^{24}	1.54×2^{22}	1.6×2^{22}
Total		1.51×2^{22}	1.14×2^{24}	1.59×2^{22}	1.65×2^{22}

Table 2: Quantum resource estimation results for steps in Argon2. (Optimization: Qubit, Adder: Ripple)

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	#Full Depth
Initial	1,090	(None)			
Update		(None)			
Final		1.03×2^{6}	1.98×2^{18}	1.66×2^{17}	1.25×2^{19}
blake2b		1.93×2^{10}	1.85×2^{23}	1.55×2^{22}	1.18×2^{24}
Total		1.99×2^{10}	1.91×2^{23}	1.6×2^{22}	1.21×2^{24}

Table 3: Quantum resource estimation results for steps in Argon2. (Optimization: Qubit, Adder: Simple)

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	#Full Depth
Initial	13,830	(None)			
Update		(None)			
Final		1.6×2^{17}	1.17×2^{19}	1.64×2^{17}	1.42×2^{14}
blake2b		1.5×2^{22}	1.1×2^{24}	1.54×2^{22}	1×2^{21}
Total		1.55×2^{22}	1.14×2^{24}	1.59×2^{22}	1.01×2^{21}

Table 4: Quantum resource estimation results for steps in Argon2. (Optimization: Depth, Adder: Ripple)

Function	#Qubit	#1qClifford	#CNOT	#Toffoli	#Full Depth
Initial	13,830	(None)			
Update		(None)			
Final		1.12×2^{5}	1.98×2^{18}	1.66×2^{17}	1.56×2^{17}
blake2b		1.05×2^{10}	1.85×2^{23}	1.55×2^{22}	1.46×2^{22}
Total		1.08×2^{10}	1.91×2^{23}	1.6×2^{22}	1.51×2^{22}

Table 5: Quantum resource estimation results for steps in Argon2. (Optimization: Depth, Adder: Simple)

5 Conclusion

This paper presents quantum circuits from two perspectives for Argon2. In the qubit-optimized quantum circuit, the number of qubits is reduced by reusing previously used qubits through inverse operations, but the depth increases due to the computations required for the inverse. In contrast, the depth-optimized quantum circuit increases the number of qubits by utilizing temp qubits and parallel adder structures without performing inverse operations, resulting in a significant reduction in depth. The analysis of quantum resources reveals a difference of up to 12,740 qubits and 196,741 depth between the four variations of qubit-optimized quantum circuits and depth-optimized quantum circuits. Given the current limitations of imperfect fault-tolerant quantum computers, it is necessary to analyze the post-quantum resistance strength through the implementation of quantum circuits from various perspectives. By appropriately adjusting the trade-off between qubits and depth, the most suitable quantum circuit can be identified. Therefore, the implementation and analysis of quantum circuits from various optimization perspectives are crucial research areas. The results of this paper contribute to the post-quantum strength analysis of Argon2 and

provide insights for future research on quantum circuit design with appropriate trade-offs of quantum resources in response to advancements in quantum computing technology.

6 Appendix

Algorithm 1 Qubit-optimized quantum circuit for the compression function(G)

Input: a , b, c, d, $sigma$

$a \leftarrow \mathbf{ADD}(b, a)$

@$Reverse\ Point$
  $sigma \leftarrow \mathbf{Classic\_to\_Quantum}(m[sigma[r][2 * i + 0]])$
$a \leftarrow \mathbf{ADD}(sigma, a)$
$\#Reverse$

for (k=0 **to** length(d)) :
  $d[k] \leftarrow \mathbf{CNOT}(a[k], d[k])$
for (k=0 **to** 64) :
  d_{box1}.append($d[(k + 32) mod\ 64]$)
$d = d_{box1}$

$c \leftarrow \mathbf{ADD}(d, c)$

for (k=0 **to** length(b)) :
  $b[k] \leftarrow \mathbf{CNOT}(c[k], b[k])$

for (k=0 **to** 64) :
  b_{box1}.append($b[(k + 24) mod\ 64]$)
$b = b_{box1}$

$a \leftarrow \mathbf{ADD}(b, a)$

@$Reverse\ Point$
  $sigma \leftarrow \mathbf{Classic\_to\_Quantum}(m[sigma[r][2 * i + 1]])$
$a \leftarrow \mathbf{ADD}(sigma, a)$
$\#Reverse$

for (k=0 **to** 64) :
  d_{box2}.append($d[(k + 16) mod\ 64]$)
$d = d_{box2}$

$c \leftarrow \mathbf{ADD}(d, c)$

for (k=0 **to** 64) :
  b_{box2}.append($b[(k + 64) mod\ 64]$)
$b = b_{box2}$

Algorithm 2 Depth-optimized quantum circuit for the compression function(G)

Input: a , b, c, d, $sigma_1$, $sigma_2$

$a \leftarrow \mathbf{ADD}(b, a)$

$sigma_1 \leftarrow \mathbf{Classic\_to\_Quantum}(m[sigma_1[r][2 * i + 0]])$
$a \leftarrow \mathbf{ADD}(sigma_1, a)$

for (k=0 **to** length(d)) :
    $d[k] \leftarrow \mathbf{CNOT}(a[k], d[k])$
for (k=0 **to** 64) :
    d_{box1}.append($d[(k + 32) mod\ 64]$)
$d = d_{box1}$

$c \leftarrow \mathbf{ADD}(d, c)$

for (k=0 **to** length(b)) :
    $b[k] \leftarrow \mathbf{CNOT}(c[k], b[k])$

for (k=0 **to** 64) :
    b_{box1}.append($b[(k + 24) mod\ 64]$)
$b = b_{box1}$

$a \leftarrow \mathbf{ADD}(b, a)$

$sigma_2 \leftarrow \mathbf{Classic\_to\_Quantum}(m[sigma_2[r][2 * i + 1]])$
$a \leftarrow \mathbf{ADD}(sigma_2, a)$

for (k=0 **to** 64) :
    d_{box2}.append($d[(k + 16) mod\ 64]$)
$d = d_{box2}$

$c \leftarrow \mathbf{ADD}(d, c)$

for (k=0 **to** 64) :
    b_{box2}.append($b[(k + 64) mod\ 64]$)
$b = b_{box2}$

References

1. Almazrooie, M., Samsudin, A., Abdullah, R., Mutter, K.N.: Quantum reversible circuit of aes-128. Quantum Information Processing **17**, 1–30 (2018)
2. Anand, R., Maitra, A., Mukhopadhyay, S.: Grover on simon. Quantum Information Processing **19**(9), 340 (2020)
3. Aumasson, J.P., Neves, S., Wilcox-O'Hearn, Z., Winnerlein, C.: Blake2: simpler, smaller, fast as md5. In: Applied Cryptography and Network Security: 11th Inter-

national Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings 11. pp. 119–135. Springer (2013)
4. Baksi, A., Jang, K., Song, G., Seo, H., Xiang, Z.: Quantum implementation and resource estimates for rectangle and knot. Quantum Information Processing **20**, 1–24 (2021)
5. Chauhan, A.K., Sanadhya, S.K.: Quantum resource estimates of grover's key search on aria. In: Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, December 17–21, 2020, Proceedings 10. pp. 238–258. Springer (2020)
6. Cuccaro, S.A., Draper, T.G., Kutin, S.A., Moulton, D.P.: A new quantum ripple-carry addition circuit. arXiv preprint quant-ph/0410184 (2004)
7. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying Grover's algorithm to AES: quantum resource estimates. In: Post-Quantum Cryptography. pp. 29–43. Springer (2016)
8. 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)
9. Huang, Z., Sun, S.: Synthesizing quantum circuits of aes with lower t-depth and less qubits. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 614–644. Springer (2022)
10. Jang, K., Baksi, A., Breier, J., Seo, H., Chattopadhyay, A.: Quantum implementation and analysis of default. Cryptology ePrint Archive (2022)
11. Jang, K., Baksi, A., Song, G., Kim, H., Seo, H., Chattopadhyay, A.: Quantum analysis of aes. Cryptology ePrint Archive (2022)
12. Jang, K., Song, G., Kim, H., Kwon, H., Kim, H., Seo, H.: Efficient implementation of present and gift on quantum computers. Applied Sciences **11**(11), 4776 (2021)
13. Jang, K., Song, G., Kim, H., Kwon, H., Kim, H., Seo, H.: Parallel quantum addition for korean block ciphers. Quantum Information Processing **21**(11), 373 (2022)
14. Jang, K., Song, G., Kwon, H., Uhm, S., Kim, H., Lee, W.K., Seo, H.: Grover on pipo. Electronics **10**(10), 1194 (2021)
15. Rahman, M., Paul, G.: Grover on katan: Quantum resource estimation. IEEE Transactions on Quantum Engineering **3**, 1–9 (2022)
16. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
17. Song, G.j., Jang, K.b., Seo, H.j.: Resource eestimation of grover algorithm through hash function lsh quantum circuit optimization. Journal of the Korea Institute of Information Security & Cryptology **31**(3), 323–330 (2021)
18. Song, G., Jang, K., Kim, H., Eum, S., Sim, M., Kim, H., Lee, W., Seo, H.: Speedy quantum circuit for grover's algorithm. Applied Sciences **12**(14), 6870 (2022)
19. 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)
20. 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)
21. Song, G., Jang, K., Seo, H.: Improved low-depth sha3 quantum circuit for fault-tolerant quantum computers. Applied Sciences **13**(6), 3558 (2023)
22. 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)

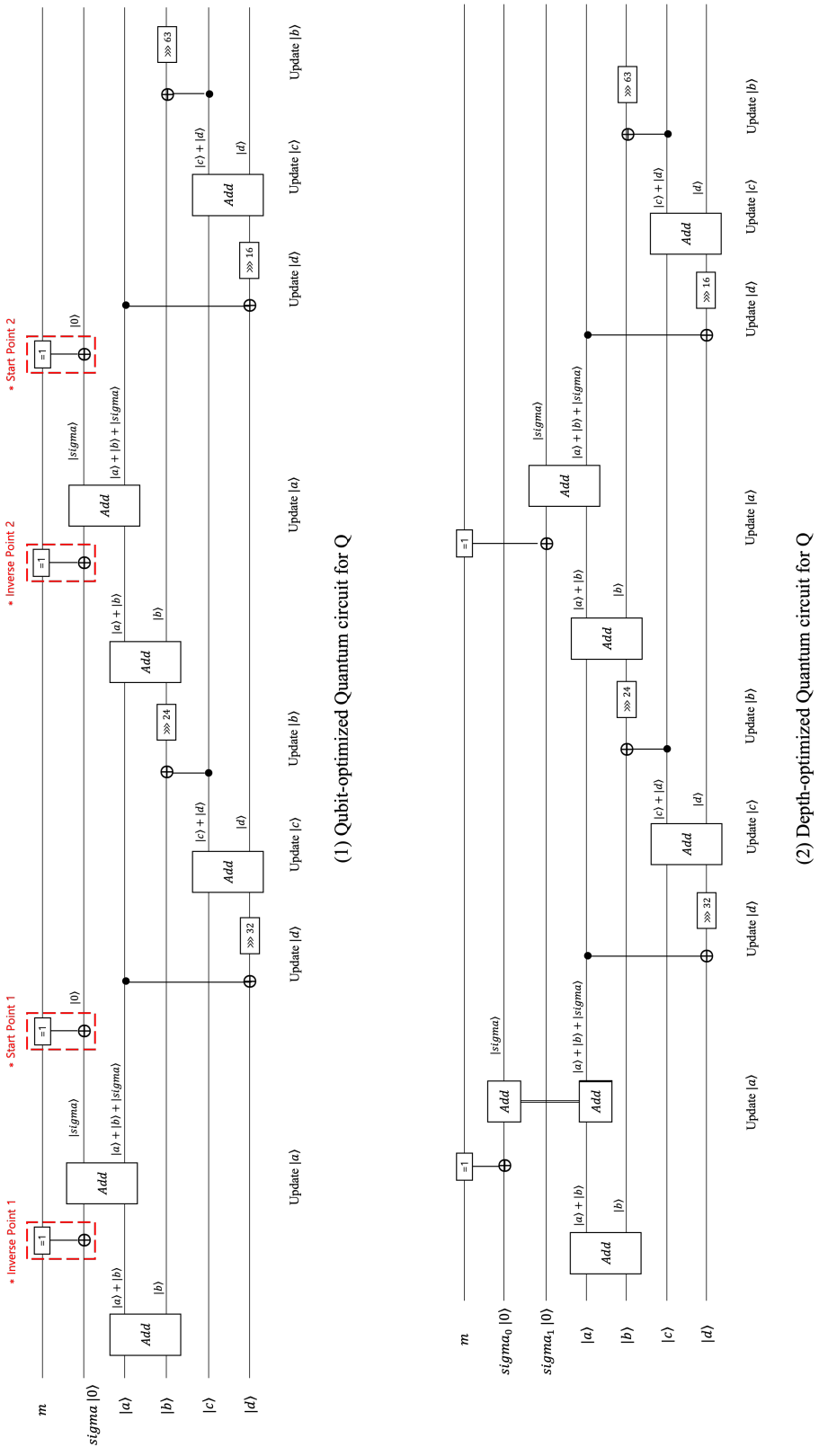


Fig. 5: Optimized Quantum circuit for G : (1)Qubit-optimized quantum circuit (2)Depth-optimized quantum circuit