Quantum Implementation of AIM: Aiming for Low-Depth

Kyungbae Jang, Dukyoung Kim, Yujin Oh, Sejin Lim, Yujin Yang, Hyunji Kim, and Hwajeong Seo

Division of IT Convergence Engineering, Hansung University, Seoul, South Korea

starj1023@gmail.com, dudejrdl123@gmail.com, oyj0922@gmail.com, dlatpwls834@gmail.com, yujin.yang34@gmail.com, khj1594012@gmail.com, hwajeong84@gmail.com

Abstract. Security vulnerabilities in the symmetric-key primitives of a cipher can undermine the overall security claims of the cipher. With the rapid advancement of quantum computing in recent years, there is an increasing effort to evaluate the security of symmetric-key cryptography against potential quantum attacks.

This paper focuses on analyzing the quantum attack resistance of AIM, a symmetric-key primitive used in the AIMer digital signature scheme. We present the first quantum circuit implementation of AIM and estimate its complexity (such as qubit count, gate count, and circuit depth) with respect to Grover's search algorithm.

For Grover's key search, the most important optimization metric is the depth, especially when considering parallel search. Our implementation gathers multiple methods for a low-depth quantum circuit of AIM in order to reduce the Toffoli depth and full depth.

Keywords: Quantum computing · Grover's search · AIM · AIMer

1 Introduction

Quantum computing poses a serious threat to cryptography, particularly to public key algorithms which can be weakened by Shor's algorithm, reducing the attack complexity to a polynomial time. As a result, researchers have been studying the applicability of public key ciphers against a quantum adversary [20,9,2]. Generally speaking, symmetric key ciphers are more robust against quantum attacks than public key ciphers, with Grover's algorithm capable of recovering a k-bit key with $\sqrt{2^k}$ searches (i.e., reduced by the square root). That means, symmetric key ciphers should double their key size to achieve a reasonable level of security claim even on quantum computers. It is worth noting that quantum security is not properly analyzed during the design phase of symmetric key ciphers.

Although Grover's algorithm theoretically reduces the security of key search by the square root, practical quantum key recovery is still very difficult due to the extreme iterations required. And the current level of quantum computer development cannot handle the depth of extreme iterations. Thus, it is meaningful to implement and analyze newly proposed symmetric key ciphers with respect to adversaries that have quantum computing capabilities.

If the quantum resources required to attack a symmetric key cipher are extensive, the cipher can be considered safe from quantum attacks without increasing the key size. In this context, it is important to note the post-quantum security requirements of the National Institute of Standards and Technology (NIST) [17,18]. NIST has defined post-quantum security levels (Level-1 $\sim$ -5) to assess the resistance of ciphers against quantum attacks (this will be described in Section 2.2).

In this work, we perform quantum cryptanalysis on the symmetric key primitive AIM used in the new signature, AIMer [16]. AIMer is one of the digital signatures[1] of Korea Post-Quantum Cryptography (KPQC) competition[2]. Additionally, AIMer is a candidate for additional post-quantum digital signature standardization by NIST[3]. We present quantum circuits for AIM, and our design philosophy prioritizes minimizing depth rather than qubit count. This choice is driven by the necessity of parallelizing Grover's search instances, which is often an inevitable option due to the substantial depth of Grover's search (discussed in Section ??). Based on the depth-optimized quantum circuits of AIM, we estimate the cost of Grover's key search and assess the post-quantum security level of AIM according to NIST's evaluation criteria.

[1] AIMer is the only symmetric key primitive (AIM) based digital signature in the KPQC competition.

[2] https://www.kpqc.or.kr/competition.html

[3] https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures

As far as we know, this work is the first attempt to analyze the post-quantum security of AIM on quantum computers. While the compromise of the inner key primitive (i.e., AIM) may not directly invalidate the security claims of the algorithm (i.e., AIMer), it can introduce vulnerabilities that should not be overlooked. From this perspective, our work holds significant importance within the context of security analysis.

Contribution

In short, this work entails the following:

1. **Quantum Circuit Implementation of AIM**. We present the implementation of quantum circuit for the variants of AIM s(-I, -III, and -V). This marks the first quantum implementation of AIM, which is the symmetric key primitive of AIMer (a candidate algorithm in the KPQC and NIST PQC competitions).
2. **Low-Depth Implementation**. Our implementation of the quantum circuits for AIM focuses on low Toffoli depth and full depth. To minimize depth while allowing for a reasonable number of qubits, we gather multiple contributions, including a low-depth quantum circuit for the multiplication, optimized quantum circuits for inner operations (Mer and Linearlayer) of AIM, and the reuse of ancilla qubits (through reverse operation).
3. **Post-quantum Security Evaluation of AIM**. We evaluate the post-quantum security of AIM by estimating the cost of Grover's key search based on the implemented quantum circuits of AIM-I, -III, and -V. For this security evaluation, we compare the estimated cost of Grover's key search for AES, as estimated by NIST [17,18], with the findings[4] from recent related work [11].

2 Foundations

2.1 Grover's Key Search

Grover's search algorithm is a quantum algorithm that can reduce the search complexity of ciphers against classical computers by a square root. That is, ciphers that use a k-bit key have an exhaustive key search complexity of $O(2^k)$ against classical computers, but Grover's key search on a quantum computer reduces the complexity to $\sqrt{2^k}$. The Grover's key search process for recovering a k-bit key for a known plaintext-ciphertext pair can be summarized as follows; *prepare* $\rightarrow$ (*Grover oracle* and *diffusion operator*)$^{\sqrt{2^k}}$ $\rightarrow$ *measure.*

Firstly, to prepare the key in a superposition state, Hadamard (H) gates are applied to the k qubits, which causes the k-qubit key to be represented as a probability distribution over all possible key values (i.e., 2^k values, as in Equation 1).

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

For the known plaintext (P), qubits are allocated and X gates are applied according to the value of the known plaintext. The main component, the Grover oracle, contains the quantum circuit of the target cipher. The known plaintext is encrypted using the quantum circuit for the target cipher and the k-qubit key ($\psi(k)$). This generates a superposition state of the ciphertext encrypted with all possible key values. Then, the Grover oracle compares a superposition state of the ciphertext with the known ciphertext (C). If there is a match (Equation 2), the Grover oracle returns the solution by flipping the sign of the corresponding state of the key (Equation 3) as follows:

$$f(x) = \begin{cases} 1 \text{ if } Enc_{\psi(k)}(P) = C \\ 0 \text{ if } Enc_{\psi(k)}(P) \neq C \end{cases} \tag{2}$$

$$U_f(|\psi\rangle |-\rangle) = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle |-\rangle \tag{3}$$

Another module, the diffusion operator, amplifies the amplitude of the solution returned from the Grover oracle, increasing the probability of recovering the key. Grover's key search sequentially iterates the oracle and

[4] The authors of [11] reported/analyzed the issue of underestimated cost in [13], the result in [18].

the diffusion operator $\sqrt{2^k}$ times to increase the amplitude of the solution sufficiently, then recover (measure) the key with high probability.

From a cost perspective, optimizing the encryption quantum circuit within the Grover oracle is crucial for reducing the cost of Grover's key search.

2.2 NIST Post-quantum Security and MAXDEPTH

To evaluate the security of a cipher against quantum attacks, NIST specifies security bounds for the cipher [17,18]:

- Level 1: Resource requirements for the attack are similar to those for breaking AES-128 ($2^{170} \rightarrow \mathbf{2^{157}}$).
- Level 3: Resource requirements for the attack are similar to those for breaking AES-192 ($2^{233} \rightarrow \mathbf{2^{221}}$).
- Level 5: Resource requirements for the attack are similar to those for breaking AES-256 ($2^{298} \rightarrow \mathbf{2^{285}}$).

Based on the cost estimation of Grover's key search for AES variants in Grassl et al.'s work [8], NIST has calculated the quantum attack complexities for Levels 1, 3, and 5 (corresponding to AES variants) to 2^{170}, 2^{233}, and 2^{298}, respectively (total gates $\times$ depth of Grover's search). One important point to note is that the attack complexity estimated by NIST in [17] is based on research results from PQCrypto'16 [8], and since then, quantum circuits for AES have been steadily optimized, leading to significant reductions in the cost of attacks in recent years [13,24,11,10]. NIST acknowledges that the estimated attack complexity based on the Levels is relative, considering the ongoing optimization of quantum circuits for AES (page 17 on [17]). Therefore, if an attack with reduced cost is proposed, the benchmark should be reconsidered.

Recently, NIST adjusted the security bounds for AES [18,19] based on the results presented in Eurocrypt'20 [13]. In [13], the quantum attack costs for AES-128, -192, and -256 were significantly reduced to 2^{157}, 2^{221}, and 2^{285}, respectively (which align with the costs specified in [18,19]). However, there is an issue of underestimation of quantum resources for their implementations, particularly with respect to non-linear operations (like S-box).

In [11], the authors analyzed estimation issue in [13] and reported corrected results. In addition to the corrections, they also presented depth-optimized quantum circuits for AES. It is worth noting that [11] has not yet undergone peer review, but the costs derived from their implementation currently represent the lowest estimates.

Throughout this paper, we consider complexity estimates from NIST [18,19] and the reduced estimates from [11] to assess the post-quantum security of AIM.

2.3 Quantum Gates

There are various quantum gates that are commonly used to incorporate ciphers into quantum circuits, including the X (NOT), CNOT, and Toffoli (CCNOT) gates. The X gate flips the value of a qubit, which can be used instead of the classical NOT operation (i.e., $\mathrm{X}(x) = \sim x$). The CNOT gate works on two qubits, where the value of the target qubit depends on the value of the control qubit. If the control qubit is 1, the target qubit is flipped; if it is 0, the target qubit remains unchanged (i.e., $\mathrm{CNOT}(x, y) = (x, x \oplus y)$). As this is equivalent to XORing the control qubit's value to the target qubit, the CNOT gate can replace the classical XOR operation. The Toffoli gate operates on three qubits, with two control qubits and one target qubit. The target qubit's value is only flipped if both control qubits have a value of 1 (i.e., $\mathrm{Toffoli}(x, y, z) = (x, y, z \oplus xy)$). This operation can be described as XORing the result of the AND operation between the control qubits with the value of the target qubit. Therefore, the Toffoli gate can replace the classical AND operation. By using these quantum gates, we can implement cipher encryption in quantum computing, replacing classical NOT, XOR, and AND operations.

For optimizing quantum circuits, it is crucial to reduce the number of Toffoli gates. Toffoli gates are expensive to implement as they require a combination of T gates (which affect T-depth) and Clifford gates. Various methods for decomposing Toffoli gates exist, and the full depth indicates the depth when Toffoli gates are decomposed. In this study, we estimate decomposed resources using a decomposition method involving 7 T gates and 8 Clifford gates, with a T-depth of 4 and a full depth of 8 for one Toffoli gate, as introduced in [1].

2.4 AIM

AIMer [16] is a signature scheme that employs the symmetric primitive AIM and the BN++ proof system [3]. AIM is a one-way function designed to withstand algebraic attacks and to be compatible with secure multi-party computation in hardware. Before presenting the quantum circuit implementation of AIM, we describe the symmetric-key primitive AIM in this section.

Figure 1 shows the encryption process of AIM. AIM has three variants (-I, -III, -V), and in this paper, we propose the quantum circuits for all variants of AIM. AIM is designed with Mers, which are S-boxes that compute exponentiation by Mersenne numbers over a large field, and a Linear layer that performs binary matrix multiplications.

The parameters e for variants of AIM are given by AIM-I: $e_1(3), e_2(27), e_*(5)$, AIM-III: $e_1(5), e_2(29), e_*(7)$, and AIM-V: $e_1(3), e_2(53), e_3(7), e_*(5)$. Note that only AIM-V consists of three Mers before the LinearLayer. For more details about AIM (or AIMer), refer to [16].

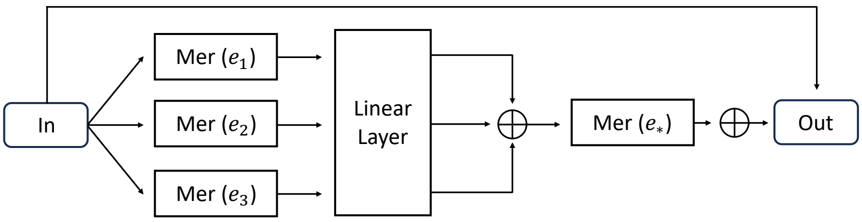

Fig. 1: Encryption process of AIM

Table 1: Quantum resources required for multiplication of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$, $\mathbb{F}_{2^{192}}/(x^{192}+x^7+x^2+x+1)$, and $\mathbb{F}_{2^{256}}/(x^{256}+x^{10}+x^5+x^2+1)$.

Field size 2^n	#CNOT	#1qCliff	$\#T$	T-depth※	#Qubit	Full depth
$n = 128$	29867	4374	15309	4	6561	78
$n = 192$	85577	10206	35721	4	15309	94
$n = 256$	115558	13122	45927	4	19683	180

※: Toffoli depth one has a T-depth of four.

3 Quantum Circuit Implementation of AIM

Our optimization goal in the quantum circuit implementation of AIM is to minimize the depth while allowing a reasonable number of additional qubits. Note that, we weigh on describing the quantum circuit implementation of AIM-I, and our design philosophy and optimization methods are also applicable to other variants (i.e., AIM-III and -V).

3.1 Binary Field Multiplication and Squaring

The component that requires the most quantum resources in the quantum circuit implementation of AIM is Mer(e). Mer(e) computes the exponentiation x^{2^e-1} on a binary field, so quantum circuits of binary multiplication and squaring are required for quantum implementation. We adopt the method in [12] for implementing quantum binary multiplication in Mer(e). In short, the authors of [12] employ the Karatsuba algorithm to implement quantum multiplication. They apply the Karatsuba algorithm recursively until the size of the divided multiplications is 1×1. The key approach they use is to prepare all operands for multiplications in advance by allocating additional ancilla qubits. As a result, all multiplications are executed simultaneously. This multiplication method has a Toffoli depth of 1 for arbitrary field sizes by generating all products in parallel and has the lowest full depth compared to other binary multiplication techniques. A disadvantage of the quantum binary multiplication in [12] is that it requires many ancilla qubits. However, we note that the burden of qubit allocation can be reduced by reusing the ancilla qubits used in the previous multiplication when the multiplication is not stand-alone (Section 3.3 in [12]).

Since $\mathsf{Mer}(e)$ is composed of multiple multiplications (i.e., not stand-alone), we can effectively reduce the burden on qubit count while reducing the depth of the multiplication. Table 2 shows the quantum resources required for quantum multiplication of binary fields $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$, $\mathbb{F}_{2^{192}}/(x^{192}+x^7+x^2+x+1)$, and $\mathbb{F}_{2^{256}}/(x^{256}+x^10+x^5+x^2+1)$ (defined in AIM-I, -III, and -V, respectively) using the method in [12].

As mentioned earlier, thanks to the reuse of ancilla qubits in the method presented in [12], we effectively reduce the number of qubits required for multiplications. Of the 6561, 15309, and 19683 qubits in Table 2, only 2443, 5387, and 7073 qubits are required for multiplications excluding the first multiplication in Mer, as 4118, 9922, and 12610 of the ancilla qubits can be reused.

Table 2: Quantum resources required for multiplication of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$, $\mathbb{F}_{2^{192}}/(x^{192}+x^7+x^2+x+1)$, and $\mathbb{F}_{2^{256}}/(x^{256}+x^{10}+x^5+x^2+1)$.

Field size 2^n	#CNOT	#1qCliff	#T	T-depth※	#Qubit	Full depth
$n=128$	29867	4374	15309	4	6561	78
$n=192$	85577	10206	35721	4	15309	94
$n=256$	115558	13122	45927	4	19683	180

※: Toffoli depth one has a T-depth of four.

While various methods to implement multiplication in quantum (and classical as well) are actively being investigated [12,14,22,7], squaring has no room for optimization or improvement due to its inherent simplicity. The quantum implementation of squaring is simpler and less costly than that of multiplication, as in classical implementation. Squaring only requires modular reduction of the input to the squared result, without the need to generate product terms, so it can be implemented with only CNOT gates.

Since modular reduction is a linear operation, an in-place implementation using PLU decomposition is possible. However, for simplicity, we implement modular reduction naively rather than using PLU decomposition. As a result, in addition to the input qubits, our squaring quantum circuit requires a few ancilla qubits (3 or 5) for temp values. Actually, this increase in the number of qubits is negligible in our implementation, which already uses many ancllia qubits. The required quantum resources for the squaring quantum circuit of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$, $\mathbb{F}_{2^{192}}/(x^{192}+x^7+x^2+x+1)$, and $\mathbb{F}_{2^{256}}/(x^{256}+x^{10}+x^5+x^2+1)$ are shown in Table 3.

Table 3: Quantum resources required for squaring of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$, $\mathbb{F}_{2^{192}}/(x^{192}+x^7+x^2+x+1)$, and $\mathbb{F}_{2^{256}}/(x^{256}+x^{10}+x^5+x^2+1)$.

Field size 2^n	#CNOT	#Qubit	Full depth
$n=128$	205	131	127
$n=192$	301	195	196
$n=256$	401	261	253

3.2 Mer

Now that we have the necessary building blocks (multiplication and squaring), we can proceed with implementing the quantum circuit of Mer. As noted earlier, for the purpose of describing the quantum circuit of Mer, we primarily focus on the variant AIM-I, which consists of $\mathsf{Mer}(3)$, (27) (before $\mathsf{LinearLayer}$), and (5) (after $\mathsf{LinearLayer}$).

Algorithm 1 describes the quantum circuit implementation of $\mathsf{Mer}(3)$. The notation CNOT128 of Algorithm 1 means the operation of CNOT gates for 128-qubit arrays. In the quantum implementation of $\mathsf{Mer}(3)$, multiple

(Mul, Reduction), and Squaring operations are performed. Additionally, CleanAncilla initializes the ancilla qubits used in multiplication (4118) without significant overhead (overhead related to reuse is discussed in Section 3.3 of [12]). These initialized ancilla qubits are reused in subsequent multiplications.

To optimize the quantum circuit of Mer, we combine Mer(3) and Mer(27) (before LinearLayer. Mer(3) and Mer(27) use the same input, and as there is a duplicated intermediate value between them, we utilize it. Algorithm 1 (Mer(3)) copies the output right before finishing (lines 12 and 13) because the same value is used in Mer(27). That is, instead of using multiplication and squaring to generate the same value in Mer(27), we use a copy of the output from Mer(3) as the input for Mer(27) and continue with subsequent operations. As we can observe in Algorithm 2, the output of Mer(3) is used as an input for Mer(27). Thanks to this efficient sharing, we can conserve the quantum resources required for certain multiplications and squarings.

Unsurprisingly, this method of efficient sharing is also applicable to AIM-III and AIM-V. Note that Mer(5) (after LinearLayer) is implemented using the same mechanism as Mer(3) and Mer(27), but sharing method is impossible due to input is different. We omit the explanation of implementation for other variants (AIM-III and AIM-V) in this paper, but report the required quantum resources. Table 4 shows the quantum resources required for the quantum circuit implementations of Mers.

Algorithm 1: Quantum circuit implementation of Mer(3).

Input: x
Output: x^{2^3-1}, x^{2^3-1}(copy), $ancilla$
//Allocate ancilla qubits for Mul
1: $ancilla \leftarrow$ allocate 4118 qubits

//Compute Mer(3)
//Copy x to $x1$
2: $x1 \leftarrow$ allocate new 128 qubits
3: CNOT128(x, $x1$)

//x^{2^2-1}
4: $x1 \leftarrow$ Squaring($x1$)

5: $x2 \leftarrow$ Mul(x, $x1$, $ancilla$)
6: $x2 \leftarrow$ Reduction($x2$)
7: $ancilla \leftarrow$ CleanAncilla(x, $x1$, $ancilla$)

//x^{2^3-1}
8: $x2 \leftarrow$ Squaring($x2$)

9: $out \leftarrow$ Mul(x, $x2$, $ancilla$)
10: $out \leftarrow$ Reduction(out)
11: $ancilla \leftarrow$ CleanAncilla(x, $x2$, $ancilla$)

//Copy out to $x3$ for Mer (27)
12: $x3 \leftarrow$ allocate new 128-qubit
13: CNOT128(out, $x3$)
14: **return** out, $x3$, $ancilla$

3.3 LinearLayer

In the LinearLayer operation, either four (for AIM-I and -III) or six (for AIM-V) matrix-vector multiplications are performed. The binary matrices of sizes 128×128, 192×192, and 256×256 for AIM-I, -III, and -V, respectively, are generated using the hash values (SHAKE-128 for AIM-I, and SHAKE-256 for AIM-III and -V). The output of Mer is used as the input vector, and it is multiplied by the binary matrices.

Since the initial vector (input of hash) is public, these binary matrices are constant. Therefore, the matrix-vector multiplication corresponds to a classical-quantum (matrix-vector) implementation that does not require

Algorithm 2: Quantum circuit implementation of Mer(27).

Input: $x^{2^3-1}(x3)$
Output: $x^{2^{27}-1}$, *ancilla*

//Compute Mer(27)
//Copy $x3$ to $x4$
$x4 \leftarrow$ allocate new 128 qubits
CNOT128($x3$, $x4$)

//x^{2^6-1}
for $i = 0$ to 2 **do**
    $x4 \leftarrow \mathsf{Squaring}(x4)$
end for

$x5 \leftarrow \mathsf{Mul}(x3, x4, ancilla)$
$x5 \leftarrow \mathsf{Reduction}(x5)$
$ancilla \leftarrow \mathsf{CleanAncilla}(x3, x4, ancilla)$

//Copy $x5$ to $x6$
$x6 \leftarrow$ allocate new 128 qubits
CNOT128($x5$, $x6$)

//$x^{2^{12}-1}$
for $i = 0$ to 5 **do**
    $x6 \leftarrow \mathsf{Squaring}(x6)$
end for

$x7 \leftarrow \mathsf{Mul}(x5, x6, ancilla)$
$x7 \leftarrow \mathsf{Reduction}(x7)$
$ancilla \leftarrow \mathsf{CleanAncilla}(x5, x6, ancilla)$

//Copy $x7$ to $x8$
$x8 \leftarrow$ allocate new 128 qubits
CNOT128($x7$, $x8$)

//$x^{2^{24}-1}$
for $i = 0$ to 11 **do**
    $x8 \leftarrow \mathsf{Squaring}(x8)$
end for

$x9 \leftarrow \mathsf{Mul}(x7, x8, ancilla)$
$x9 \leftarrow \mathsf{Reduction}(x9)$
$ancilla \leftarrow \mathsf{CleanAncilla}(x7, x8, ancilla)$

//$x^{2^{27}-1}$
for $i = 0$ to 2 **do**
    $x9 \leftarrow \mathsf{Squaring}(x9)$
end for

$out \leftarrow \mathsf{Mul}(x3, x9, ancilla)$
$out \leftarrow \mathsf{Reduction}(out)$
$ancilla \leftarrow \mathsf{CleanAncilla}(x3, x9, ancilla)$
return out, *ancilla*

Table 4: Quantum resources required for the Mer of AIM-I.

Component	#CNOT	#1qCliff	$\#T$	T-depth※	#Qubit	Full depth
Mer(3)	68636	8748	30618	8	8882	411
Mer(27)	226224	26244	91854	16	13840	2488
Mer(5)	115385	13122	45927	12	6957	678
Mer(29)	780375	91854	321489	32	67005	6547
Mer(53)	1037702	118098	413343	32	86833	14482

※: Toffoli depth one has a T-depth of four.

a SHAKE-128 or SHAKE-256 quantum circuit. In other words, a quantum circuit for this matrix-vector multiplication can be efficiently designed using the classical values of the generated matrices.

We adopt a naive approach for the quantum circuit implementation of matrix-vector multiplication, rather than the PLU decomposition. An in-place implementation based on PLU decomposition increases the depth due to the execution of CNOT gates in limited space (fewer qubits) without using additional qubits. On the other hand, we allocate a new 128, 192, or 256-qubit output vector (AIM-I, -III, and -V, respectively) and perform CNOT gates between the input vector and output vector where the value of the matrix is 1. This method (out-of-place) requires additional qubits, but the depth decreases due to the execution of CNOT gates in the increased space. Actually, opting for the out-of-place method, which allocates output qubits rather than the in-place method where the input is transformed into the output, represents our design philosophy.

One more thing to note is that for the last matrix-vector multiplication, instead of allocating a new output vector, we use the final output vector from the previous matrix-vector multiplication. This implementation is possible due to the XOR operations between the resulting vectors after the LinearLayer is performed. As a result, we can save 128, 192, and 256 qubits (CNOT gates also) for AIM-I, -III, and -V, respectively. Table 5 shows the quantum resources required for Linearlyer of AIM-I, -III, and -V.

Table 5: Quantum resources required for the LinearLayer of AIM.

LinearLayer	#CNOT	#Qubit	Full depth
AIM-I ($n = 128$)	16889	640	426
AIM-III ($n = 192$)	37657	960	632
AIM-V ($n = 256$)	99352	1792	1015

Although we did not describe it earlier, the vector extracted from the hash value is XORed with the result vector of the matrix-vector multiplication. This step corresponds to a classical-quantum implementation. Thus, we only apply X gates to the input vector depending on the bit values of the public vector, instead of using CNOT gates which is a more efficient approach. Lastly, $\mathsf{Mer}(\mathsf{e}_*)$ and FeedForward, which XOR the input with the output, are performed. Note that FeedForward is a quantum-quantum implementation, so CNOT gates are used.

Finally, Table 6 shows the quantum resources required for the AIM quantum circuits. Our proposed AIM quantum circuits require a significant number of ancilla qubits, which is due to the Karatsuba multiplication method [12] we adopted. Most of the ancilla qubits are used for the multiplication operations inside Mer. While we allow for a large number of qubits, we provide low T-depth and full depth. In particular, since a single multiplication has a T-depth of only 4, the T-depth of our proposed quantum circuit is very low. In the trade-off between qubit count and depth, we use metrics such as Toffoli depth × qubit count ($TD \times M$) and full depth × qubit count ($FD \times M$), which are common metrics for quantifying the performance of quantum circuits.

4 Post-Quantum Security Evaluation of AIM

In this section, we discuss the post-quantum security of AIM. In a nutshell, we estimate the cost of Grover's key search for AIM and compare the cost with the costs of Grover's key search for AES variants. The costs for

Table 6: Quantum resources required for the AIM quantum circuits.

Cipher	#CNOT	#1qCliff	$\#T$	T-depth※	#Qubit	Full depth	$TD \times M$	$FD \times M$
AIM-I	358754	39430	137781	36	25299	3499	227691	88521201
AIM-III	1144536	132785	464373	48	88395	8583	1060740	758694285
AIM-V	1486100	157588	551124	44	108072	16857	1188792	1821769704

※: Toffoli depth one has a T-depth of four.

the AES variants we use to evaluate post-quantum security are NIST estimates [17,18]. These estimates draw from the work of Grassl et al. [8] and Jaques et al [13]. Additionally, we also consider the work of Jang et al. [11], which currently provides the lowest cost for evaluation (since the estimates of [18] based on [13] are lower bound).

As explained in Section 2.1, Grover's key search for a cipher using a k-bit key involves approximately $\sqrt{2^k}$ iterations of Grover oracle and diffusion operator. Tight analysis of the Grover search algorithm [5] suggests that the optimal number of iterations is $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$, and we estimate the cost based on this. When estimating the cost of Grover's key search, we ignore the diffusion operator, as its overhead can be considered negligible (as is done in most related studies [11,13]). Therefore, we estimate the cost based only on the oracle. The Grover oracle consists of the AIM quantum circuit for encryption, an n-controlled NOT gate (n is the ciphertext size) for comparing the ciphertext (with known ciphertext), and the reverse operation of the previously executed AIM quantum circuit for the next iteration. The n-controlled NOT gate is estimated to be $(32 \cdot n - 64)$ T gates using the decomposition method in [23]. Thus, the cost of Grover's key search for AIM is estimated as $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor \times (32 \cdot 128 - 64)$ T gates $+ \lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor \times$(Table 6 $\times$ 2). Since the iterations are sequential, the number of qubits does not increase from Table 6, but only one decision qubit to check the ciphertext (with known ciphertext) is added. Table 7 shows the costs of Grover's key search for AIM.

In addition, we include the metrics $TD^2 \times M$ and $FD^2 \times M$ in Table 7. Grover's key search suffers from extreme depth, making it difficult to execute. Therefore, performing parallel search to reduce the depth is more practical. However, the efficiency of parallelizing Grover's search is very poor (see Section ??). The reason is that to reduce the depth by S, S^2 Grover instances must be executed in parallel [15,13], resulting in a qubit count increase of S. Therefore, when considering parallel search, the metrics that need to be optimized are $TD^2 \times M$ and $FD^2 \times M$. This is why minimizing the depth is clearly advantageous for quantum circuits of target ciphers for Grover's key search.

Table 7: Costs of the Grover's key search for AIM

Cipher	Total gates	Total depth	Cost (complexity)	#Qubit	$TD \times M$	$FD \times M$	For parallel search	
							$TD^2 \times M$	$FD^2 \times M$
AIM-I	$1.612 \cdot 2^{83}$	$1.342 \cdot 2^{76}$	$1.082 \cdot 2^{160}$	25300	$1.351 \cdot 2^{82}$	$1.036 \cdot 2^{91}$	$1.182 \cdot 2^{150}$	$1.39 \cdot 2^{167}$
AIM-III	$1.306 \cdot 2^{117}$	$1.646 \cdot 2^{109}$	$1.075 \cdot 2^{227}$	88396	$1.517 \cdot 2^{116}$	$1.110 \cdot 2^{126}$	$1.707 \cdot 2^{216}$	$1.827 \cdot 2^{235}$
AIM-V	$1.645 \cdot 2^{149}$	$1.616 \cdot 2^{142}$	$1.33 \cdot 2^{292}$	108073	$1.752 \cdot 2^{148}$	$1.332 \cdot 2^{159}$	$1.862 \cdot 2^{280}$	$1.076 \cdot 2^{302}$

We compare the Grover's key search cost for AES variants to evaluate the post-quantum security of AIM. Table 8 provides a birds-eye view of our discussion of post-quantum security for AIM. While it may not be unfair, when compared to NIST's estimates [17] based on Grassl et al.'s quantum circuit implementation of AES [8], AIM-I, -III, and -V cannot achieve Level-1, -3, and -5, respectively. This is because the cost of implementing AES quantum circuits in [8] is high, which in turn makes NIST's estimated costs for each level overly conservative [17].

Recently, NIST adjusted the Grover's key search costs for AES variants [18] based on Jaques et al's work [13]. Therefore, we use these adjusted costs for our evaluation[5]. Additionally, we also consider the work of Jang et al. [11] for our evaluation. They presented the lowest Grover's key search costs for AES (with their

[5]Note that these costs have an issue of underestimation, as discussed in [11].

depth-optimized circuits); AES-128: 2^{156}, AES-192: 2^{222}, AES-256: 2^{286}. As of now, since these costs are the lowest and do not have any issues of underestimation, we incorporate these costs into our evaluation as well.

As a result of comparing the Grover's key search costs from [18,13] and [11] with the Grover's key search costs of AIM, AIM-I, -III, and -V reliably achieve Level-1, -3, and -5 (post-quantum security), as 2^{160}, 2^{227}, and 2^{292} are greater than 2^{157}, $2^{221 \text{ or } 222}$, and $2^{285 \text{ or } 286}$, respectively (see Table 8).

Additionally, we can consider the required number of qubits for Grover's key search. Although NIST does not consider qubit counts as a key metric for estimating attack complexity (considering the limit of depth rather than the limit of qubit counts), the qubit count is certainly a significant metric. The estimated costs of Grover's key search for AIM require a large number of qubits (more than AES) and have a higher attack complexity than AES (for all variants). In this context, we can evaluate that AIM can achieve the appropriate post-quantum security for the key sizes (i.e., Level-1, -3, and -5 for 128, 192, and 256-bit keys), and the required number of qubits for an attack is also high.

Table 8: Comparison of the Grover's key search costs

Post-quantum Security	NIST'16 [17] (based on [8])	NIST'22 [18] (based on [13])	J++ [11]	AIM		
				-I	-III	-V
Level-1 (AES-128)	2^{170}	2^{157}	2^{157}	2^{160}		
Level-3 (AES-192)	2^{233}	2^{221}	2^{222}		2^{227}	
Level-5 (AES-256)	2^{298}	2^{285}	2^{286}			2^{292}

5 Conclusion

This paper presents the first quantum circuit implementation of the symmetric-key primitive AIM used in AIMer. To reduce the cost of Grover's key search, an effective quantum circuit implementation of AIM is essential, and our effort reduces the depth while allowing a reasonable number of qubits. Specifically, various techniques are applied to optimize the quantum implementation of the components of AIM, such as binary field multiplication, Mer, and LinearLayer.

With our depth-optimized quantum circuits for AIM, we have estimated the Grover's key search costs for AIM variants. When evaluating AIM against several recent benchmarks [17,18,11,13], we found that AIM-I, -III, and -V reliably achieve post-quantum security Levels-1, -3, and -5, respectively.

Assessing the post-quantum security of cryptographic systems against potential attacks on quantum computers, which pose significant threats, contributes to the establishment of a secure post-quantum system. In this context, our future plans involve evaluating post-quantum security in various ways for cryptographic algorithms. The Simon algorithm [21] for period finding in ciphers or the application of Grover's algorithm to other PQC algorithms, such as quantum sieving [6] for lattice-based ciphers and Quantum Information Set Decoding (QISD) [4] for code-based ciphers, may be prominent choices.

References

1. Amy, M., Maslov, D., Mosca, M., Roetteler, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**(6), 818–830 (Jun 2013). https://doi.org/10.1109/tcad.2013.2244643, `http://dx.doi.org/10.1109/TCAD.2013.2244643` 3
2. Banegas, G., Bernstein, D.J., Van Hoof, I., Lange, T.: Concrete quantum cryptanalysis of binary elliptic curves. Cryptology ePrint Archive (2020) 1
3. Baum, C., Nof, A.: Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In: Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part I. pp. 495–526. Springer (2020) 4
4. Bernstein, D.J.: Grover vs. mceliece. In: Post-Quantum Cryptography: Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings 3. pp. 73–80. Springer (2010) 10

5. Boyer, M., Brassard, G., Høyer, P., Tapp, A.: Tight bounds on quantum searching. Fortschritte der Physik **46**(4-5), 493–505 (Jun 1998). https://doi.org/10.1002/(sici)1521-3978(199806)46:4/5¡493::aid-prop493¿3.0.co;2-p, http://dx.doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P 9
6. Chailloux, A., Loyer, J.: Lattice sieving via quantum random walks. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27. pp. 63–91. Springer (2021) 10
7. Cheung, D., Maslov, D., Mathew, J., Pradhan, D.K.: On the design and optimization of a quantum polynomial-time attack on elliptic curve cryptography. In: Workshop on Quantum Computation, Communication, and Cryptography. pp. 96–104. Springer (2008) 5
8. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying Grover's algorithm to AES: Quantum resource estimates. In: Takagi, T. (ed.) Post-Quantum Cryptography. pp. 29–43. Springer International Publishing, Cham (2016) 3, 9, 10
9. Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Improved quantum circuits for elliptic curve discrete logarithms. In: Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11. pp. 425–444. Springer (2020) 1
10. Huang, Z., Sun, S.: Synthesizing quantum circuits of AES with lower T-depth and less qubits. Cryptology ePrint Archive, Report 2022/620 (2022), https://eprint.iacr.org/2022/620 3
11. Jang, K., Baksi, A., Kim, H., Song, G., Seo, H., Chattopadhyay, A.: Quantum analysis of aes. Cryptology ePrint Archive, Paper 2022/683 (2022), https://eprint.iacr.org/2022/683, https://eprint.iacr.org/2022/683 2, 3, 9, 10
12. Jang, K., Kim, W., Lim, S., Kang, Y., Yang, Y., Seo, H.: Optimized implementation of quantum binary field multiplication with toffoli depth one. In: Information Security Applications: 23rd International Conference, WISA 2022, Jeju Island, South Korea, August 24–26, 2022, Revised Selected Papers. pp. 251–264. Springer (2023) 4, 5, 6, 8
13. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing grover oracles for quantum key search on AES and lowmc. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 280–310. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_10, https://doi.org/10.1007/978-3-030-45724-2_10 2, 3, 9, 10
14. Kepley, S., Steinwandt, R.: Quantum circuits for $\mathbb{F}_{2^n}$-multiplication with subquadratic gate count. Quantum Information Processing **14**(7), 2373–2386 (2015) 5
15. Kim, P., Han, D., Jeong, K.C.: Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to aes and sha-2. Quantum Information Processing **17**, 1–39 (2018) 9
16. Kim, S., Ha, J., Son, M., Lee, B., Moon, D., Lee, J., Lee, S., Kwon, J., Cho, J., Yoon, H., et al.: The AIMer signature scheme 1, 4
17. NIST.: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf 1, 2, 3, 8, 9, 10
18. NIST.: Call for additional digital signature schemes for the post-quantum cryptography standardization process (2022), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf 1, 2, 3, 8, 9, 10
19. NIST.: Stateless hash-based digital signature standar (2023), https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.ipd.pdf 3
20. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.: Quantum resource estimates for computing elliptic curve discrete logarithms. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23. pp. 241–270. Springer (2017) 1
21. Simon, D.R.: On the power of quantum computation. SIAM journal on computing **26**(5), 1474–1483 (1997) 10
22. Van Hoof, I.: Space-efficient quantum multiplication of polynomials for binary finite fields with sub-quadratic Toffoli gate count. arXiv preprint arXiv:1910.02849 (2019) 5
23. Wiebe, N., Roetteler, M.: Quantum arithmetic and numerical analysis using repeat-until-success circuits. arXiv preprint arXiv:1406.2040 (2014) 9
24. Zou, J., Wei, Z., Sun, S., Liu, X., Wu, W.: Quantum circuit implementations of AES with fewer qubits. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 697–726. Springer International Publishing, Cham (2020) 3