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 presents the first quantum circuit implementation of AIM and estimates 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 [16,8,4]. 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. That means, symmetric key ciphers can 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. Therefore, 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. This reflects NIST's post-quantum security requirements [15], which consider this trend. In this work, we performs quantum cryptanalysis on the symmetric key primitive AIM used in the new signature, AIMer [14]. We estimate the cost of Grover's key search for AIM and evaluate the post-quantum security level according to NIST's evaluation criteria.

Contribution

In short, this work entails the following:

1. **Quantum Circuit Implementation of AIM**. We present for the first time an implementation of a quantum circuit for the symmetric key primitive AIM, focusing solely on the implementation for AIM-I, although the techniques used can be extended to other variations (i.e., AIM-III, -V).
2. **Low-Depth Implementation**. Our implementation of the quantum circuit for AIM-I focuses on low Toffoli depth and full depth. To minimize depth while allowing for a reasonable number of qubits, we gather multiple contributions.
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 circuit of AIM-I. For this security evaluation, we compare the estimated cost of Grover's key search for AES, as estimated by NIST [15], with recent work [10] that has reduced the cost of Grover's key search for AES.

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* $\rightarrow$ $(\textit{Grover oracle}$ and $\textit{diffusion operator})^{\sqrt{2^k}}$ $\rightarrow$ *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). For the known plaintext, 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. This generates a superposition state of the ciphertext encrypted with all possible key values (i.e., known-plaintext $\rightarrow \psi$(ciphertext)). Then, the Grover oracle compares a superposition state of the ciphertext with the known ciphertext. If there is a match, the Grover oracle returns the solution by flipping the sign of the corresponding state of the key. 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 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 [15].

- Level 1: Resource requirements for the attack are similar to those for breaking AES-128.
- Level 2: Resource requirements for the attack are similar to those for breaking SHA-256/SHA3-256.
- Level 3: Resource requirements for the attack are similar to those for breaking AES-192.
- Level 4: Resource requirements for the attack are similar to those for breaking SHA-384/SHA3-384.
- Level 5: Resource requirements for the attack are similar to those for breaking AES-256.

Based on the cost estimation of Grover's key search for AES variants in Grassl et al.'s work [7], NIST has calculated the quantum attack complexities for Levels 1, 3, and 5 (corresponding to AES variants). We can use these quoted attack complexities as a benchmark to estimate the post-quantum security of various ciphers.

One important point to note is that the attack complexity estimated by NIST is based on research results from PQCrypto'16 [7], and since then, quantum circuits for AES have been steadily optimized, leading to significant reductions in the cost of attacks in recent years [12,18,10,9]. 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 [15]). Therefore, if an attack with reduced cost is proposed, the benchmark should be reconsidered. Currently, the most cost-efficient attack is the work of Jang et al [10] . We consider both the complexity estimated by NIST and the reduced complexity estimated in [10] to evaluate the post-quantum security of AIM.

Grover's key search is much farther ahead than the current state of quantum computing. While it is true that Grover's key search theoretically reduces the security by the square root, succeeding in the attack requires handling an extreme circuit depth. For this reason, NSIT defines a limit on the required depth for quantum attacks, called MAXDEPTH; $2^{40} \leq 2^{64} \leq 2^{96}$. Thus, if the attacker reaches the MAXDEPTH limit, they will need to use a parallel search approach [13] for the Grover's key search algorithm. Parallel searches can be classified into outer and inner methods (for more details, refer to [13]).

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 [14] is a signature scheme that employs the symmetric primitive AIM and the BN++ proof system [5]. 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 first describe the symmetric-key primitive AIM in this section.

AIM has three variants (-I, -III, -V), but in this paper, we only focus on AIM-I (for more details about the other variants, see [14]). AIM is designed with Mer, which are S-boxes that compute exponentiation by Mersenne numbers over a large field, and a Linear layer that performs binary matrix multiplications. Figure 1 shows the encryption process of AIM-I.

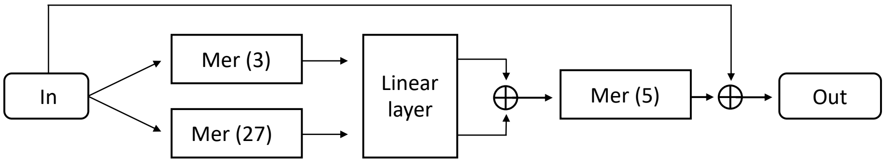

Fig. 1: Encryption process of AIM-I

3 Quantum Circuit Implementation of AIM

Our optimization goal in the quantum circuit implementation of AIM-I is to minimize the depth while allowing a reasonable number of additional qubits.

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 approach in [11] for implementing quantum binary multiplication in Mer(e). 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 [11] is that it requires many ancilla qubits. However, as described in [11] (Section 3.3), 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. Since 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 1 shows the quantum resources required for quantum multiplication of binary field $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$ (defined in AIM-I) using the method in [11]. Of the 6561 qubits in Table 1, only 2443 qubits are used for multiplications excluding the first multiplication in Mer, as 4118 of the ancilla qubits can be reused.

Table 1: Quantum resources required for multiplication of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$

Field size 2^n	#CNOT	#1qCliff	$\#T$	T-depth※	#Qubit	Full depth
$n=128$	29867	4374	15309	4	6561	78

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

The quantum implementation of squaring is simpler and less costly than that of multiplication. Squaring only requires modular reduction of the input to the squared result, without the need to generate product terms, and 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 perform modular reduction naively in our implementation. In addition to the input qubits, our squaring quantum circuit requires three ancilla qubits for temp values. The required quantum resources for the squaring quantum circuit of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$ are shown in Table 2.

Table 2: Quantum resources required for squaring of $\mathbb{F}_{2^{128}}/(x^{128}+x^7+x^2+x+1)$

Field size 2^n	#CNOT	#Qubit	Full depth
$n=128$	205	131	127

Now that we have the necessary building blocks (multiplication and squaring), we can proceed with implementing the quantum circuit of Mer. Algorithm 1 describes the quantum circuit implementation of Mer(3). The notation CNOT128 means the operation of CNOT gates for 128-qubit arrays. As mentioned earlier, the 4188 ancilla qubits used in multiplication are initialized (CleanAncilla) without significant overhead (see Section 3.3 in [11]) and are reused in subsequent multiplications. 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. Algorithm 2 describes the quantum circuit implementation of Mer(27).

In LinearLayer, a total of four matrix-vector multiplications are performed. 128 × 128 binary matrices are generated using the hash value (SHAKE-128) of the initial vector, and the output of Mer becomes a vector. Since the initial vector is public, these binary matrices are constant. Therefore, the matrix-vector multiplication corresponds to a classical (matrix)-quantum (vector) implementation that does not require a SHAKE-128 quantum circuit. 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-qubit output vector and perform CNOT gates between the input vector and output vector where the value of the matrix is 1. This requires additional qubits, but the depth decreases due to the execution of CNOT gates in the increased space. For the last matrix-vector multiplication, instead of allocating a new output vector, we use the final output vector of the previous matrix-vector multiplication. This implementation is possible when the XOR operation between the resulting vectors after the LinearLayer is precomputed. As a result, we can save 128 qubits and 128 CNOT gates.

Table 3 shows the quantum resources required for the quantum circuit implementations of Mer and LinearLayer. Since Mer(5) is implemented using the same mechanism as Mer(3) and (27), we omit the explanation of implementation in this paper and only report the required quantum resources.

Although not described in detail in this paper, XOR operations between vectors (128-qubit) can be implemented with only 128 CNOT gates. However, the XOR operation between the public vector and the input vector performed before Mer(5) (not illustrated in Figure 1) corresponds to a classical-quantum implementation since the public vector is constant (public). Thus, only X gates are applied depending on the bit values of the public vector.

Finally, Table 4 shows the quantum resources required for the AIM-I quantum circuit. Our proposed AIM quantum circuit requires a significant number of ancilla qubits, which is due to the Karatsuba multiplication method [11] 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

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
$ancilla$ ← allocate 4118 qubits

//Compute Mer(3)
//Copy $x$ to $x1$
$x1$ ← allocate new 128 qubits
CNOT128($x$, $x1$)

//$x^{2^2-1}$
$x1$ ← Squaring($x1$)

$x2$ ← Mul($x$, $x1$, $ancilla$)
$x2$ ← Reduction($x2$)
$ancilla$ ← CleanAncilla($x$, $x1$, $ancilla$)

//$x^{2^3-1}$
$x2$ ← Squaring($x2$)

$out$ ← Mul($x$, $x2$, $ancilla$)
$out$ ← Reduction($out$)
$ancilla$ ← CleanAncilla($x$, $x2$, $ancilla$)

//Copy $out$ to $x3$ for Mer (27)
$x3$ ← allocate new 128-qubit
CNOT128($out$, $x3$)
return $out$, $x3$, $ancilla$
```

Table 3: Quantum resources required for the components 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
LinearLayer	16889	·	·	·	640	426

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 ($TD * M$), which are common metrics for quantifying the performance of quantum circuits.

Table 4: Quantum resources required for the AIM-I quantum circuit.

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

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-I and compare the cost with the costs of Grover's key search for AES variants. The costs

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
    $x4 \leftarrow$ Squaring($x4$)
end for

$x5 \leftarrow$ Mul($x3$, $x4$, $ancilla$)
$x5 \leftarrow$ Reduction($x5$)
$ancilla \leftarrow$ 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
    $x6 \leftarrow$ Squaring($x6$)
end for

$x7 \leftarrow$ Mul($x5$, $x6$, $ancilla$)
$x7 \leftarrow$ Reduction($x7$)
$ancilla \leftarrow$ 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
    $x8 \leftarrow$ Squaring($x8$)
end for

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

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

$out \leftarrow$ Mul($x3$, $x9$, $ancilla$)
$out \leftarrow$ Reduction($out$)
$ancilla \leftarrow$ CleanAncilla($x3$, $x9$, $ancilla$)
return $out$, $ancilla$
```

for the AES variants we use to evaluate post-quantum security are NIST estimates [15] based on Grassl et al.'s work [7] and Jang et al.'s work [10] which is currently the lowest cost.

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 [6] suggests that the optimal number of iterations is $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$, and we estimate the cost based on this. However, 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 [10,12]). Therefore, we estimate the cost based only on the oracle. The Grover oracle consists of the AIM-I 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-I 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 [17]. Thus, the cost of Grover's key search for AIM-I 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 4 $\times$ 2). Since the iterations are sequential, the number of qubits does not increase from Table 4, but only one decision qubit to check the ciphertext (with known ciphertext) is added. Table 5 shows the cost of Grover's key search for AIM-I.

In addition, we include the metrics $TD^2 \times M$ and $FD^2 \times M$ in Table 5. 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. The reason is that to reduce the depth by $\sqrt{S}$, S Grover instances must be executed in parallel [13,12]. For example, to reduce the depth by a factor of 4, 16 Grover instances must be executed in parallel, and the gate count and qubit count increase by a factor of the depth. 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 5: Cost of the Grover's key search for AIM-I

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	$1.351 \cdot 2^{82}$	$1.036 \cdot 2^{91}$	$1.182 \cdot 2^{150}$	$1.39 \cdot 2^{167}$

We compare the Grover's key search cost for AES variants to evaluate the post-quantum security of AIM. When compared to NIST's estimates [15] based on Grassl et al.'s quantum circuit implementation of AES [7], AIM-I cannot achieve Level-1. However, this is because the cost of implementing AES's quantum circuit in [7] is high, and NIST's estimated costs for each level are too conservative [15]. The key search cost for Grover's algorithm has recently decreased for various ciphers [3,2,18,9]. We can see that achieving an appropriate post-quantum security level for key size is very difficult when compared to NIST's estimated costs. Therefore, it is more realistic to approach the Grover's key search cost for AES variants. Recently, Jang et al. significantly reduced the Grover's key search cost for AES [10] (from 2^{170} to 2^{157}). Comparing the Grover's key search cost for AIM to 2^{157}, 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 cost of Grover's key search for AIM-I requires a large number of qubits and has a higher attack complexity than AES-128 ($2^{160} >= 2^{157}$). Finally, we can evaluate that AIM-I can achieve the appropriate post-quantum security for the key size (i.e., Level-1 for a 128-bit key). We do not estimate the costs for AIM-3 and -5 in this work, but since the complexity of cipher variants usually increases similarly according to the key size, if correct, AIM-III and -V can achieve Level 3 and 5, respectively. Table 6 provides a birds-eye view of our discussion of post-quantum security for AIM.

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

Table 6: Comparison of the Grover's key search costs

Post-quantum Security	NIST [15] (based on [7])	J++ [10]	AIM		
			-I	-III	-V
Level-1 (AES-128)	2^{170}	2^{157}	2^{160}		
Level-3 (AES-192)	2^{233}	2^{222}		.	
Level-5 (AES-256)	2^{298}	2^{286}			.

multiplication, Mer, and LinearLayer. We estimate the cost of Grover's key search for AIM-I and demonstrate achieving Level-1 post-quantum security.

Our work does not involve the implementation and evaluation of AIM-III and AIM-V. Our future plan is to extend our work to all variations and add more optimization contributions.

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. Anand, R., Maitra, A., Mukhopadhyay, S.: Evaluation of quantum cryptanalysis on SPECK. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds.) Progress in Cryptology – INDOCRYPT 2020. pp. 395–413. Springer International Publishing, Cham (2020) 7
3. Anand, R., Maitra, A., Mukhopadhyay, S.: Grover on SIMON. Quantum Information Processing **19**(9) (Sep 2020). https://doi.org/10.1007/s11128-020-02844-w, http://dx.doi.org/10.1007/s11128-020-02844-w 7
4. Banegas, G., Bernstein, D.J., Van Hoof, I., Lange, T.: Concrete quantum cryptanalysis of binary elliptic curves. Cryptology ePrint Archive (2020) 1
5. 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) 3
6. 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 7
7. 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) 2, 7, 8
8. 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
9. 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 2, 7
10. 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 1, 2, 7, 8
11. 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) 3, 4
12. 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, 7
13. 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) 2, 7
14. 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, 3
15. 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, 7, 8

16. 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
17. Wiebe, N., Roetteler, M.: Quantum arithmetic and numerical analysis using repeat-until-success circuits. arXiv preprint arXiv:1406.2040 (2014) 7
18. 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) 2, 7