Depth-Optimized Quantum Implementation of CHAM

Kyungbae Jang, Yujin Oh, and Hwajeong Seo

Hansung University, Seoul, South Korea
{starj1023, oyj0922, hwajeong84}@gmail.com

Abstract. Security weaknesses in the symmetric-key components of a cipher can compromise its overall security assurances. With the rapid progress in quantum computing in recent years, there is a growing focus on assessing the resilience of symmetric-key cryptography against possible quantum attacks.
This paper is dedicated to examining the quantum attack resistance of CHAM, a family of lightweight block ciphers. We provide an optimized quantum circuit implementation of CHAM and evaluate its complexity metrics, such as the number of qubits, gate count, and circuit depth, within the context of Grover's search algorithm.
For Grover's key search, minimizing the circuit depth is the key optimization goal, particularly when parallel search capabilities are taken into account. Our approach enhances parallelism for a low-depth quantum circuit of CHAM, aiming to reduce both the Toffoli depth and overall circuit depth.

Keywords: Quantum computing · Grover's search · CHAM.

1 Introduction

Quantum computing poses a substantial threat to cryptography, especially to public key algorithms. Shor's algorithm, for instance, can reduce the complexity of attacking these algorithms to polynomial time. As a result, there has been significant research into the vulnerability of public key ciphers against quantum adversaries [17,7,2]. Generally, symmetric key ciphers are considered more resilient to quantum attacks than public key ciphers, as Grover's algorithm can find a k-bit key with approximately $\sqrt{2^k}$ operations, effectively reducing the attack complexity by a square root factor. This implies that to retain a comparable level of security in the face of quantum threats, symmetric key ciphers should double their key size.

It is important to note that quantum security considerations are often overlooked during the design phase of symmetric key ciphers. Although Grover's algorithm theoretically weakens key search security by a square root, in practice, quantum key recovery remains highly challenging due to the vast number of iterations required. Additionally, current quantum computers lack the capability to handle such complex, deep iterations.

Given these limitations, it is crucial to implement and evaluate new symmetric key ciphers in the context of adversaries equipped with quantum computing capabilities. If the quantum resources required to break a symmetric key cipher are substantial, the cipher may be considered quantum-resistant without needing to increase the key size.

In this context, the post-quantum security standards set by the National Institute of Standards and Technology (NIST) are essential [14,15]. NIST has established post-quantum security levels (from Level 1 to Level 5) to assess the strength of ciphers against quantum attacks, as further detailed in Section 2.2.

In this work, we implement quantum circuits for the lightweight block cipher CHAM [13]. We provide quantum implementations of CHAM that are optimized for circuit depth, with our design approach focusing on reducing depth rather than the number of qubits. This focus is motivated by the need to parallelize instances of Grover's search, which often becomes necessary due to the significant circuit depth involved in Grover's algorithm (as discussed in Section 2.3). We compare our results with previous works [10,21], which represent the most efficient quantum circuits for CHAM to date.

Using these depth-optimized quantum circuits for CHAM, we estimate the resource cost of Grover's key search and evaluate the post-quantum security level of CHAM according to the criteria set by NIST.

Contribution

1. **Depth-Optimized Quantum Circuits for CHAM**. We present the implementation of quantum circuits for different variants of CHAM. Our approach emphasizes minimizing circuit depth by employing additional ancilla qubits, enabling the inner quantum additions within the round functions to be executed in parallel.
2. **Post-Quantum Security Evaluation of CHAM**. We assess the post-quantum security of CHAM by estimating the cost of Grover's key search using the implemented quantum circuits. To conduct this security evaluation, we compare the estimated cost of Grover's key search for CHAM with that of AES, as determined by NIST [14,15].

2 Foundations

2.1 Grover's Key Search

Grover's search algorithm is a quantum algorithm that reduces the search complexity of ciphers compared to classical computers by a square root factor. For instance, a cipher using a k-bit key has an exhaustive key search complexity of $O(2^k)$ on a classical computer, but Grover's key search on a quantum computer reduces this complexity to $\sqrt{2^k}$. The Grover key search process for recovering a k-bit key, given a known plaintext-ciphertext pair, can be summarized as follows: *prepare* $\rightarrow$ (*Grover oracle* and *diffusion operator*)$^{\sqrt{2^k}}$ $\rightarrow$ *measure*.

To begin, Hadamard (H) gates are applied to the k qubits to prepare the key in a superposition state. This operation creates a probability distribution over all possible key values (i.e., 2^k values), as represented in Equation 1.

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

Next, qubits are initialized according to the known plaintext (P) by applying X gates based on its value. The core component, the Grover oracle, incorporates the quantum circuit of the target cipher. The known plaintext is encrypted using this quantum circuit along with the k-qubit key ($\psi(k)$), generating a superposition state of ciphertexts for all possible key values. The Grover oracle then compares this superposition of ciphertexts with the known ciphertext (C). If a match is found (Equation 2), the oracle indicates the solution by flipping the sign of the corresponding key state (Equation 3) as follows:

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

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

The diffusion operator, another critical module, amplifies the amplitude of the solution identified by the Grover oracle, thereby increasing the likelihood of recovering the correct key. Grover's key search involves iteratively applying the oracle and the diffusion operator $\sqrt{2^k}$ times to sufficiently amplify the amplitude of the solution, allowing the key to be recovered (measured) with high probability.

From a cost optimization perspective, minimizing the resources required for the encryption quantum circuit within the Grover oracle is essential to reducing the overall cost of Grover's key search.

2.2 NIST Post-quantum Security

To assess the security of a cipher against quantum attacks, NIST has defined security bounds for various levels [14,15]:

- Level 1: The resource requirements for an attack are comparable to those needed to break AES-128 ($2^{170} \rightarrow \mathbf{2^{157}}$).
- Level 2: Resource requirements for the attack are similar to those for breaking SHA-256/SHA3-256.
- Level 3: The resource requirements for an attack are similar to those for breaking AES-192 ($2^{233} \rightarrow \mathbf{2^{221}}$).
- Level 4: Resource requirements for the attack are similar to those for breaking SHA-384/SHA3-384.

– Level 5: The resource requirements for an attack are equivalent to those for breaking AES-256 ($2^{298} \rightarrow \mathbf{2^{285}}$).

Based on the cost estimates for Grover's key search against AES variants as presented by Grassl et al. [6], NIST determined the quantum attack complexities for Levels 1, 3, and 5 (corresponding to different AES variants) as 2^{170}, 2^{233}, and 2^{298}, respectively (calculated as total gates multiplied by the depth of Grover's search). It is important to highlight that NIST's complexity estimates in [14] are derived from research results published in PQCrypto'16 [6]. Since that time, quantum circuits for AES have undergone continuous optimization, resulting in a significant decrease in the cost of attacks in recent years [11,22,9,8].

NIST also acknowledges that the attack complexities based on these levels are relative, given the ongoing optimizations in quantum circuits for AES (see page 17 in [14]). Therefore, if a more efficient attack is proposed, the benchmarks may need to be updated.

Recently, NIST revised the security bounds for AES [15,16] following the findings presented at Eurocrypt 2020 [11]. In [11], the quantum attack costs for AES-128, -192, and -256 were significantly reduced to 2^{157}, 2^{221}, and 2^{285}, respectively, aligning with the updated values in [15,16].

2.3 NIST MAXDEPTH

Exhaustive key search using Grover's algorithm is still far beyond the current capabilities of quantum computing. Although Grover's key search theoretically reduces security by a square root factor, executing a successful attack requires managing an extremely deep quantum circuit. In practical scenarios, Grover's search may be performed in parallel by breaking it into smaller instances to reduce the lengthy sequential computations (as mentioned on page 46 of [16]).

To address this, NIST defines a maximum allowable depth for quantum attacks, known as MAXDEPTH, which is set between 2^{40} and 2^{96}.

Thus, once the attacker reaches the MAXDEPTH limit, they must employ a parallel strategy for Grover's key search, as discussed in [12]. Parallel searches can be classified into outer and inner methods (for further details, see [12]).

It is crucial to understand that the efficiency of parallel searches is generally low due to the disproportionate trade-off between the reduction in depth and the success probability of key recovery. Usually, the depth-qubit count product is a primary metric for evaluating the performance of a quantum circuit. However, to reduce the depth of Grover's search by a factor of S, S^2 instances must be executed in parallel [12,11].

This means that the depth-qubit count product should be redefined as the depth^2-qubit count product when designing Grover's search for parallel execution to achieve successful key recovery. This is why minimizing the depth is more advantageous when considering the parallelization of Grover's search.

2.4 Quantum Gates

Various quantum gates are commonly used to implement ciphers within quantum circuits, including the X (NOT), CNOT, and Toffoli (CCNOT) gates. The X gate, equivalent to the classical NOT operation, flips the value of a qubit (i.e., $\mathrm{X}(x) = \sim x$). The CNOT gate operates on two qubits, where the target qubit's value is modified based on the control qubit's value. Specifically, if the control qubit is 1, the target qubit is flipped; if it is 0, the target qubit remains unchanged (i.e., $\mathrm{CNOT}(x, y) = (x, x \oplus y)$). Since this operation is equivalent to XORing the control qubit's value with that of the target qubit, the CNOT gate can serve as a quantum substitute for the classical XOR operation. The Toffoli gate, on the other hand, involves three qubits: two control qubits and one target qubit. The target qubit is flipped only when both control qubits are set to 1 (i.e., $\mathrm{Toffoli}(x, y, z) = (x, y, z \oplus xy)$). This can be viewed as XORing the result of the AND operation between the control qubits with the target qubit's value, making the Toffoli gate a quantum analogue for the classical AND operation.

By leveraging these quantum gates, we can implement encryption algorithms in quantum computing, replacing classical NOT, XOR, and AND operations.

Optimizing quantum circuits requires minimizing the number of Toffoli gates, as they are particularly costly to implement. Toffoli gates require a combination of T gates (which impact the T-depth) and Clifford gates. Various decomposition techniques for Toffoli gates have been developed, where the "full depth" refers to the circuit depth after these gates have been decomposed. In this study, we estimate the resources needed for decomposition using a method that involves 7 T gates and 8 Clifford gates, resulting in a T-depth of 4 and a full depth of 8 for a single Toffoli gate, as detailed in [1].

2.5 CHAM

CHAM operates with either a 64-bit plaintext (16×4) or a 128-bit plaintext (32×4), denoted as X, and utilizes either a 128-bit or 256-bit key, denoted as K. In the key schedule of CHAM, the input key K is divided into segments as $K = K[0]||K[1]||\dots||K[w]$, where $w = 8$ for CHAM-64/128 and CHAM-128/256, and $w = 4$ for CHAM-128/128. The key schedule is defined as follows:

$$RK[i] = K[i] \oplus (K[i] \lll 1) \oplus (K[i] \lll 8),$$
$$RK[i+w] = K[i] \oplus (K[i] \lll 1) \oplus (K[i] \lll 8)$$

The round function of CHAM is dependent on the round number i. If the round number i is odd, the round function is defined by:

$$X_{i+1}[3] = ((X_i[0] \oplus i) \boxplus (X_i[1] \lll 1) \oplus \ (RK[i \bmod w)) \lll 8$$
$$X_{i+1}[j] = X_i[j+1], (0 \le j \le 2)$$

Otherwise, if $i = 0$ or is an even number, the round function is given by:

$$X_{i+1}[3] = ((X_i[0] \oplus i) \boxplus (X_i[1] \lll 8) \oplus \ (RK[i \bmod w)) \lll 1$$
$$X_{i+1}[j] = X_i[j+1], (0 \le j \le 2)$$

3 Quantum Circuit Implementation of CHAM

In this section, we describe how we reduce the circuit depth of CHAM compared to that in previous works [10,21]. Our goal in implementing the quantum circuit for CHAM is to minimize the circuit depth (suitable strategy for Grover's parallelization, see Section 2.3) while using additional ancilla qubits. It is worth noting that our approach is applicable to all variants of CHAM (i.e., CHAM-64, CHAM-128, and CHAM-256).

In previous quantum implementation of CHAM [10,21], full parallelism was not achieved. The difference between [10] and [21] lies in the quantum implementation of the linear layer in CHAM, which consists of XOR and rotation operations. In [10], an incomplete in-place implementation was presented for the linear layer. In contrast, [21] proposed complete in-place implementations using PLU factorization, along with an optimized method for the linear layer based on the approach in [20]. In their approach, the rounds are executed sequentially due to the reverse operation of the round key used in the round function. They initialize the round key (i.e., $|0\rangle$, using the reverse operation) to reduce the number of qubits required. However, the intermediate round state is required to initialize the round key, and this value is updated in subsequent rounds. In other words, the subsequent rounds cannot be performed until the reverse operation (to initialize the round key) is completed. As a result, although the number of qubits is reduced, the circuit depth increases.

In contrast, we allocate additional ancilla qubits to copy the round keys, and we do not reverse the operations to reverse them. This allows us to perform up to three rounds in parallel. After the use of the copied round keys, they are treated as garbage qubits since the intermediate round state is updated by the parallelization of three rounds. Note that these garbage qubits cannot be initialized. As a result, although additional ancilla qubits must be allocated for each round key, the circuit depth is significantly reduced, making this optimization well-suited for Grover's parallelization (see Section 2.3).

Algorithms 1 and 2 describe the quantum circuit implementations for CHAM-64/128 and CHAM-128/128, -128/256, respectively. We allocate additional ancilla qubits (RK_{temp} in Algorithms 1 and 2) for each round, and the round key is copied to these qubits using CNOT gates (16 CNOT gates for CHAM-64/128 and 32 CNOT gates for CHAM-128/128 and CHAM-128/256). Note that RK_{temp} is discarded after use (i.e., garbage qubits).

Table 1 shows the required quantum resources for the quantum circuits of the CHAM variants, along with a comparison to previous works [10,21]. We estimate the detailed quantum resources after decomposing the Toffoli gates. Among various decomposition methods, we adopt the one presented in [1], which uses 8 Clifford + 7 T gates, a T-depth of 4, and a full depth of 8. Note that this decomposition is the same as the one adopted in [10,21].

Algorithm 1: Quantum circuit for round function of CHAM-64/128 (i is *odd*).

Input: $X[0] \sim [3], RK, c$ (an ancilla qubit for quantum addition)
Output: $X[0] \sim [3]$
//Round constant addition
for $j = 0$ to 7 **do**
  if $(i \gg j)$ & 1 **then**
    $X[0] \leftarrow$X($X[0][j]$)
  end if
end for

//Store the round key in the ancilla qubits
$X[1] \leftarrow X[1] \lll 8$
$RK_{temp} \leftarrow$ Allocate 16 ancilla qubits for RK_{temp}
$RK_{temp} \leftarrow$ CNOT16($X[1], RK_{temp}$)
$RK_{temp} \leftarrow$ CNOT16(RK, RK_{temp})
$X[1] \leftarrow X[1] \ggg 8$

//Addition
$X[0] \leftarrow ADD(RK_{temp}, X[0], c)$

return $X[1], X[2], X[3], X[0] \lll 1$

Algorithm 2: Quantum circuit for round function of CHAM-128/128, -128/256 (i is *odd*).

Input: $X[0] \sim [3], RK, c$ (an ancilla qubit for quantum addition)
Output: $X[0] \sim [3]$
//Add round constant(i)
for $j = 0$ to 7 **do**
  if $(i \gg j)$ & 1 **then**
    $X[0] \leftarrow$X($X[0][j]$)
  end if
end for

//Store the round key in the ancilla qubits
$X[1] \leftarrow X[1] \lll 8$
$RK_{temp} \leftarrow$ Allocate 32 ancilla qubits for RK_{temp}
$RK_{temp} \leftarrow$ CNOT32($X[1], RK_{temp}$)
$RK_{temp} \leftarrow$ CNOT32(RK, RK_{temp})
$X[1] \leftarrow X[1] \ggg 8$

//Addition
$X[0] \leftarrow ADD(RK_{temp}, X[0], c)$

return $X[1], X[2], X[3], X[0] \lll 1$

Table 1: Quantum resources required for the CHAM quantum circuits.

Cipher	#CNOT	#1qCliff	$\#T$	#Qubit (M)	Full depth (FD)
CHAM-64/128 [10]	27120	6960	16240	204	17034
CHAM-128/128 [10]	58040	14640	34160	292	37766
CHAM-128/256 [10]	70080	17584	40992	420	45252
CHAM-64/128 [21]	29960	6960	16240	195	17031
CHAM-128/128 [21]	58080	14640	34160	259	37768
CHAM-128/256 [21]	69696	17584	40992	387	44904
CHAM-64/128 (This work)	27120	6960	16240	1484	7105
CHAM-128/128 (This work)	58040	14640	34160	2852	14772
CHAM-128/256 (This work)	70080	17584	40992	3492	17712

4 Post-Quantum Security Evaluation of CHAM

In this section, we examine the post-quantum security of CHAM. Specifically, we estimate the cost of Grover's key search for CHAM and compare it with the costs of Grover's key search for various AES variants. The AES costs used for this post-quantum security evaluation are based on NIST estimates [14,15], which, in turn, rely on the findings of Grassl et al. [6] and Jaques et al. [11].

As detailed in Section 2.1, Grover's key search for a cipher using a k-bit key requires approximately $\sqrt{2^k}$ iterations of the Grover oracle and diffusion operator. A tighter analysis of Grover's algorithm [4] suggests that the optimal number of iterations is $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$, and our cost estimates are based on this calculation. When estimating the cost of Grover's key search, we exclude the diffusion operator, as its overhead is typically negligible (a common assumption in related studies [9,11]). Consequently, our cost estimation focuses only on the oracle.

The Grover oracle comprises the CHAM quantum circuit for encryption, an n-controlled NOT gate (where n represents the ciphertext size) to compare the ciphertext with the known ciphertext, and the reverse operation of the CHAM quantum circuit for the subsequent iteration. The n-controlled NOT gate is estimated to require $(32 \cdot n - 64)$ T gates, according to the decomposition method in [19]. Thus, the total cost of Grover's key search for CHAM 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 1 $\times$ 2). Since these iterations are executed sequentially, the number of qubits does not increase from what is indicated in Table 1, except for the addition of a single decision qubit used to compare the ciphertext against the known ciphertext. Table 2 summarizes the costs associated with Grover's key search for CHAM.

We also include the metrics $TD^2 \times M$ and $FD^2 \times M$ in Table 2. Grover's key search is characterized by a significant circuit depth, which makes it challenging to execute efficiently. Therefore, performing a parallelized search to reduce the circuit depth is more feasible in practice. However, the effectiveness of parallelizing Grover's search is notably limited (as discussed in Section 2.3). To reduce the depth by a factor of S, S^2 instances of Grover's algorithm must be run in parallel [12,11], which results in a qubit count increase by a factor of S. Hence,

when considering a parallel search, the key metrics to optimize are $TD^2 \times M$ and $FD^2 \times M$. This explains why minimizing circuit depth is particularly advantageous for quantum circuits designed for Grover's key search.

Table 2: Costs of the Grover's key search for CHAM

Cipher	Total gates	Total depth	Cost (complexity)	#Qubit	$FD \times M$	$FD^2 \times M$
CHAM-I	$1.254 \cdot 2^{81}$	$1.362 \cdot 2^{77}$	$1.709 \cdot 2^{158}$	2841	$1.889 \cdot 2^{88}$	$1.287 \cdot 2^{166}$
CHAM-III	$1.304 \cdot 2^{81}$	$1.416 \cdot 2^{78}$	$1.847 \cdot 2^{159}$	2853	$1.973 \cdot 2^{89}$	$1.397 \cdot 2^{168}$
CHAM-V	$1.566 \cdot 2^{146}$	$1.698 \cdot 2^{142}$	$1.33 \cdot 2^{289}$	6729	$1.395 \cdot 2^{155}$	$1.395 \cdot 2^{297}$

To evaluate the post-quantum security of CHAM, we compare the costs of Grover's key search for AES variants. By comparing the updated Grover's key search costs from [15] with the costs for CHAM-64/128, -128/128, and -128/256, we find that these CHAM variants successfully achieve Levels 1, 3, and 5 (post-quantum security), as 2^{158}, 2^{159}, and 2^{289} exceed 2^{157}, 2^{157}, and 2^{285}, respectively.

5 Conclusion

This paper introduces the depth-optimized quantum circuit implementation of the lightweight block cipher CHAM. To minimize the cost of Grover's key search, developing an efficient quantum circuit for CHAM is crucial. Our approach achieves this by reducing circuit depth while maintaining a manageable number of qubits.

Using our depth-optimized quantum circuits for CHAM, we estimated the costs of Grover's key search for various CHAM variants. When comparing CHAM against several recent standards [14,15], we determined that CHAM-64/128, -128/128, and -128/256 consistently achieve post-quantum security Levels 1, 3, and 5, respectively.

Evaluating the post-quantum security of cryptographic systems against possible attacks by quantum computers, which present substantial threats, is vital for establishing a secure post-quantum framework. In this regard, our future work will focus on assessing post-quantum security across different scenarios for cryptographic algorithms. Approaches like Simon's algorithm [18] for period finding in ciphers or applying Grover's algorithm to other PQC algorithms, such as quantum sieving [5] for lattice-based cryptography and Quantum Information Set Decoding (QISD) [3] for code-based cryptography, could be prominent directions to explore.

6 Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea govern-

ment(MIST) (No. RS-2019-II190033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 75%) and this work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00264, Research on Blockchain Security Technology for IoT Services, 25%).

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 5, 6
2. Banegas, G., Bernstein, D.J., Van Hoof, I., Lange, T.: Concrete quantum cryptanalysis of binary elliptic curves. Cryptology ePrint Archive (2020) 1
3. 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) 9
4. 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 8
5. 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) 9
6. 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) 4, 8
7. 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
8. 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 4
9. 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 4, 8
10. 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) 2, 6, 8
11. 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 4, 8

12. 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) 4, 8
13. Koo, B., Roh, D., Kim, H., Jung, Y., Lee, D.G., Kwon, D.: Cham: A family of lightweight block ciphers for resource-constrained devices. In: International conference on information security and cryptology. pp. 3–25. Springer (2017) 2
14. 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 2, 3, 4, 8, 9
15. 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 2, 3, 4, 8, 9
16. NIST.: Stateless hash-based digital signature standar (2023), https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.ipd.pdf 4
17. 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
18. Simon, D.R.: On the power of quantum computation. SIAM journal on computing **26**(5), 1474–1483 (1997) 9
19. Wiebe, N., Roetteler, M.: Quantum arithmetic and numerical analysis using repeat-until-success circuits. arXiv preprint arXiv:1406.2040 (2014) 8
20. Xiang, Z., Zeng, X., Lin, D., Bao, Z., Zhang, S.: Optimizing implementations of linear layers. IACR Transactions on Symmetric Cryptology (2020) 6
21. Yang, Y., Jang, K., Baksi, A., Seo, H.: Optimized implementation and analysis of cham in quantum computing. Applied Sciences **13**(8), 5156 (2023) 2, 6, 8
22. 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) 4