Depth-Optimized Quantum Implementation of CHAM

Kyungbae Jang[1], Yujin Oh[1], Hwajeong Seo[1]

[1] Division of IT Convergence Engineering, Hansung University, Seoul, South Korea

Presentation at WISA 2025

August 25, 2025

Contents

Contents...

- CHAM
  - **We improve the quantum circuit implementations of CHAM** (Jang et al., Quantum Information Processing, 2022; Yang et al., Applied Sciences, 2023).
  - We optimize circuit depth for the implementation by allocating additional ancilla qubits.
    - To be more specific, we parallelize the inner quantum additions in the round function.

Contribution

- CHAM
  - **We improve the quantum circuit implementations of CHAM** (Jang et al., Quantum Information Processing, 2022; Yang et al., Applied Sciences, 2023).
  - We optimize circuit depth for the implementation by allocating additional ancilla qubits.
    - To be more specific, we parallelize the inner quantum additions in the round function.
- Grover's Attack
  - Based on our quantum circuits for CHAM, we estimate the required quantum resources for Grover's key recovery attack on CHAM.
    - Depth optimization is more effective for Grover's attack (strictly speaking, for parallelized Grover's search).

Contribution

- CHAM
  - **We improve the quantum circuit implementations of CHAM** (Jang et al., Quantum Information Processing, 2022; Yang et al., Applied Sciences, 2023).
  - We optimize circuit depth for the implementation by allocating additional ancilla qubits.
    - To be more specific, we parallelize the inner quantum additions in the round function.
- Grover's Attack
  - Based on our quantum circuits for CHAM, we estimate the required quantum resources for Grover's key recovery attack on CHAM.
    - Depth optimization is more effective for Grover's attack (strictly speaking, for parallelized Grover's search).
- Post-quantum security evaluation of CHAM
  - We assess the security of CHAM against Grover's algorithm (i.e., quantum exhaustive search).

Contents. . .

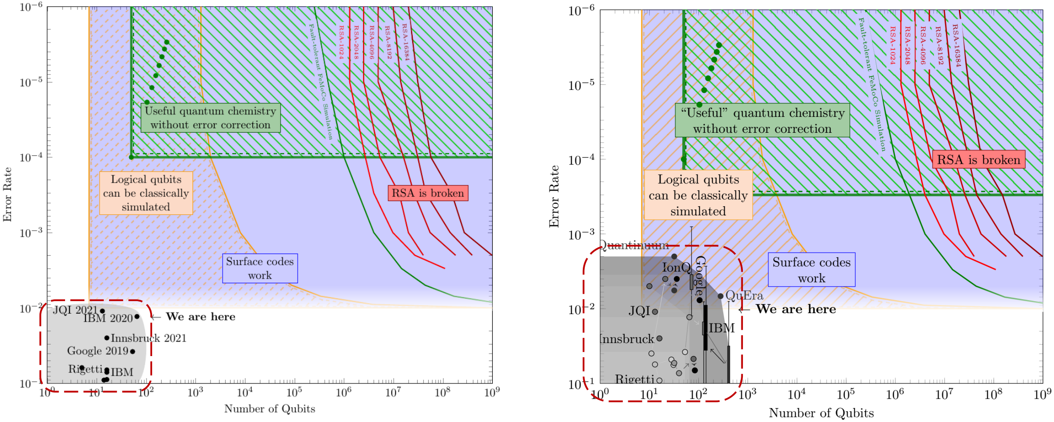


Figure 1: Quantum computing landscape in 2021 (left) and 2024 (right).

Backdrop and Motivation: Quantum Computing Land Scape

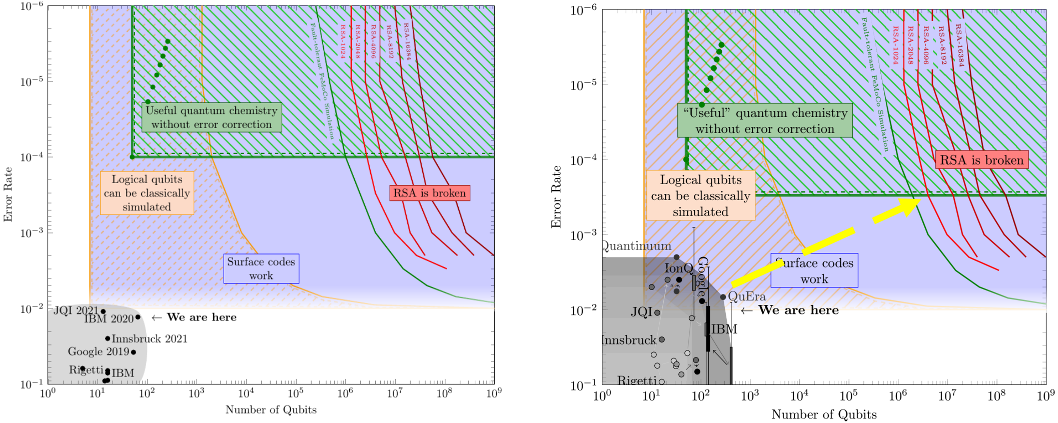


Figure 1: Quantum computing landscape in 2021 (left) and 2024 (right).

Quantum key search using Grover's search algorithm

1. A k-qubit key is prepared in superposition $|\psi\rangle$

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

2. In oracle $f(x)$, the plaintext is encrypted with the key in the superposition state.

$$f(x) = \begin{cases} 1 \text{ if } Enc(k) = c \\ 0 \text{ if } Enc(k) \neq c \end{cases}$$

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

3. Lastly, the diffusion operator amplifies the amplitude of the negative sign state.

Repeat steps 2 and 3 about $\sqrt{2^k}$ times ($O(2^k)$ in classical)

- The NIST call for proposals indicates several security categories that are related to the hardness of a quantum key search attack (based on [GLRS]) on a block cipher, like AES
  - Level 1: Cipher is at least as hard to break as AES-128 (2^{157}).
  - Level 3: Cipher is at least as hard to break as AES-192 (2^{221}).
  - Level 5: Cipher is at least as hard to break as AES-256 (2^{285}).

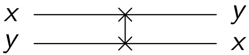
- The cost (i.e., complexity) of a quantum attack is calculated as (total number of gates x total full depth).
    - Ex) Level 3 (AES-192) $\rightarrow 2^{110} \times 2^{111} = 2^{221}$.

Backdrop and Motivation: NIST post-quantum secutiy level

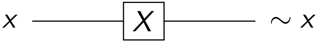
- The cost (i.e., complexity) of a quantum attack is calculated as (total number of gates x total full depth).
  - Ex) Level 3 (AES-192) $\rightarrow 2^{110} \times 2^{111} = 2^{221}$.
- NIST does not include the number of qubits when estimating the cost per level.
- This is because NIST considers the extreme depth due to sequential iterations in Grover's algorithm to be more burdensome rather than the number of qubits.

- The cost (i.e., complexity) of a quantum attack is calculated as (total number of gates x total full depth).
  - Ex) Level 3 (AES-192) $\rightarrow 2^{110} \times 2^{111} = 2^{221}$.
- NIST does not include the number of qubits when estimating the cost per level.
- This is because NIST considers the extreme depth due to sequential iterations in Grover's algorithm to be more burdensome rather than the number of qubits.
- This shows the effectiveness of our implementation: **increasing the number of qubits and reducing the depth.**

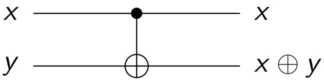
- There are several commonly used quantum gates to implement ciphers into quantum circuit, such as:



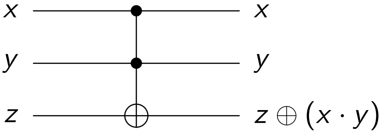
(a) Swap gate



(b) X (NOT) gate



(c) CNOT gate



(d) Toffoli (CCNOT) gate

Figure 2: Quantum gates.

In our work, we use the quantum programming tool **ProjectQ** to implement and simulate quantum circuits.

- We use two internal libraries ClassicalSimulator and ResourceCounter of ProjectQ to verify the test vector and then estimate the required quantum resources.
  - **ClassicalSimulator:** simulate large-scale quantum circuits by limiting only quantum gates with Boolean functions such as X, CNOT, and Toffoli gates.
  - **ResourceCounter:** estimate the resources required for the implemented quantum circuit (qubits, quantum gates, circuit depth).

```
def CDKM_adder(eng, a, b, c, n):
    for i in range(n-1):
        CNOT | (a[i+1], b[i+1])

    CNOT | (a[1], c)
    Toffoli_gate(eng,a[0], b[0], c)
    CNOT | (a[2], a[1])
    Toffoli_gate(eng,c, b[1], a[1])
    CNOT | (a[3], a[2])
                 ⋮
```

```
Estimate cost...
Gate class counts:

Gate counts:
    Allocate : 98
    CCX : 1247
    CX : 4179
    Deallocate : 98
    X : 1160

Depth : 814.
```

Quantum Programming and Simulation

- The results of the paper are reported from **error-free** quantum simulations (i.e., logical level).
- All relevant codes are released in public:
  https://github.com/starj1023/CHAM_Parallel_QC

Contents. . .

- Key schedule is consists of XOR operations → **linear layer**.

$$RK[i] = K[i] \oplus (K[i] \lll 1) \oplus (K[i] \lll 8)$$
$$RK[(i + k/w) \oplus 1] = K[i] \oplus (K[i] \lll 1) \oplus (K[i] \lll 11),$$

- There are **two designs** for linear layer:
  - **In-place**: the result is computed on the input.
  - **Out-of-place**: the result is computed on a separate output.
- Unlike in-place implementation, out-of-place implementation requires additional qubits for the output.
- To reduce the number of qubits, we adopt the **in-place method**.

Quantum Circuit Implementation of CHAM

- Linear layer can be represented by binary matrix:

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- In general, **PLU decomposition** can be applied for in-place implementation.
- For further optimization, **linear-layer optimization techniques can be applied**.
  - Various methods have been presented in IACR ToSC (FSE), and we adopt one[1] of these.

[1]Xiang, Z.; Zeng, X.; Lin, D.; Bao, Z.; Zhang, S. Optimizing implementations of linear layers. IACR Trans. Symmetric Cryptol. 2020.

- Round function of CHAM

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

- Most quantum resources are required for quantum addition in the round function.
- There are various quantum adder designs based on classical designs (such as ripple-carry adders, carry-lookahead adders, etc.).
  - **The choice of quantum adder is important**, and we adopt the **CDKM adder**[2] (balanced performance in terms of qubit count and circuit depth).

[2]Cuccaro, S., Draper, T., Kutin, S., Moulton, D.: A new quantum ripple-carry addition circuit. arXiv (2008)

- The arrangement of quantum additions determines the overall performance.

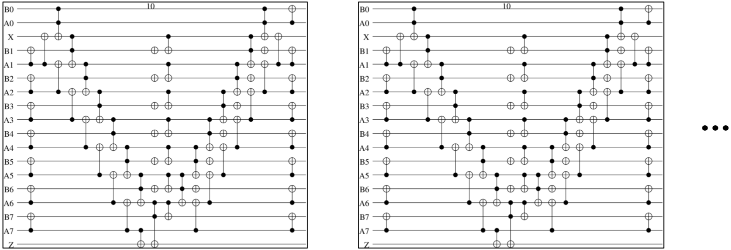


Figure 3: Sequential quantum additions.

- It has the benefit of reducing the qubit count; however, it increases the circuit depth.

Quantum Circuit Implementation of CHAM

- Parallel design has the benefit of reducing the circuit depth.

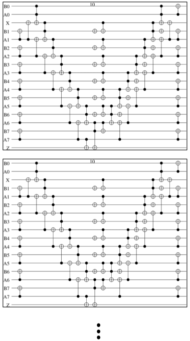


Figure 3: Parallel quantum additions.

- To do this, Dependencies between additions and the need for additional ancilla qubits should be considered.

- In CHAM, at most **3 rounds can be performed in parallel**.
  (since the 4th round uses the output of the 1st round)

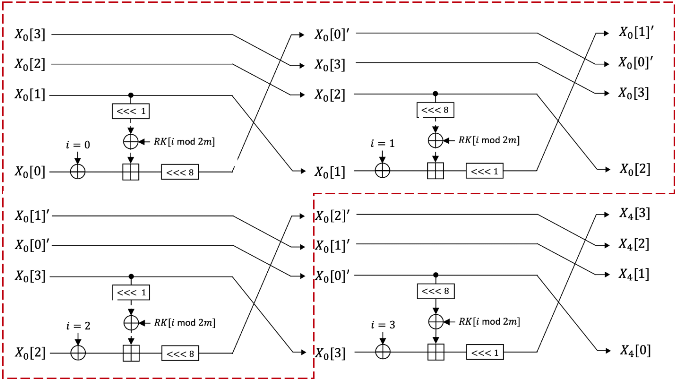


Figure 4: 4 rounds of CHAM.

- In the previous works[3] [4], round key (RK) was reused sequentially (to save qubits).
  - Causing a sequential round flow.

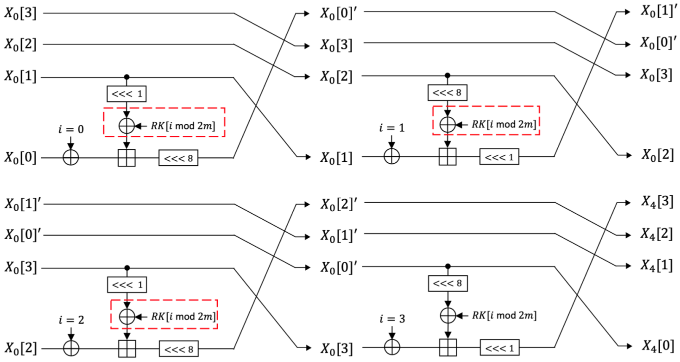


Figure 4: 4 rounds of CHAM.

3 [10] Jang, K. et al., Parallel quantum addition for korean block ciphers. Quantum Information Processing, 2022.
4 [21] Yang, Y. et al., Optimized implementation and analysis of cham in quantum computing. Applied Sciences, 2023.

- We allocate additional qubits and **generate three round keys (RKs)** to enable the parallelization of three rounds.
  - As a result, three quantum additions are performed in parallel → **low depth**.

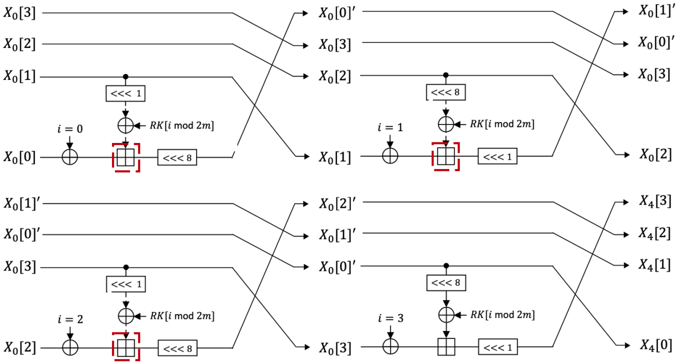


Figure 4: 4 rounds of CHAM.

Contents. . .

Results

- We use more qubits for our 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

Figure 5: Required quantum resources for CHAM quantum circuits[3] [4]

[3] [10] Jang, K. et al., Parallel quantum addition for korean block ciphers. Quantum Information Processing, 2022.

[4] [21] Yang, Y. et al., Optimized implementation and analysis of cham in quantum computing. Applied Sciences, 2023.

- We use more qubits for our quantum circuits but reduce the circuit depth.
  - Suitable optimization approach for Grover's key recovery (strictly speaking, for its parallelization).

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

Figure 5: Required quantum resources for CHAM quantum circuits[3] [4]

[3] [10] Jang, K. et al., Parallel quantum addition for korean block ciphers. Quantum Information Processing, 2022.

[4] [21] Yang, Y. et al., Optimized implementation and analysis of cham in quantum computing. Applied Sciences, 2023.

- Based on the presented quantum circuits of CHAM, we estimate the required quantum resources for **Grover's key recovery** (*FD*: full depth, *M*: qubit count).

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	$1.395 \cdot 2^{155}$	$1.395 \cdot 2^{297}$

- We should focus on **Cost (= Total gates $\times$ Total depth)**.
- To evaluate the post-quantum security of CHAM, we compare the costs of Grover's key search for AES variants.
  - NIST Level 1: 2^{157} (AES-128), NIST Level 3: 2^{221} (AES-192), NIST Level 5: 2^{285} (AES-256)

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	$1.395 \cdot 2^{155}$	$1.395 \cdot 2^{297}$

Results

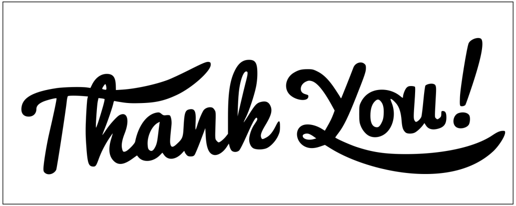
- Compared to AES, **CHAM is more difficult to break with Grover's algorithm**.
  - CHAM-I: $2^{158} > 2^{157}$ (AES-128)
  - CHAM-III: $2^{159} > 2^{157}$ (AES-128)
  - CHAM-V: $2^{289} > 2^{285}$ (AES-256)
- CHAM variants successfully achieve **Levels 1 and 5 (post- quantum security)**.

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	$1.395 \cdot 2^{155}$	$1.395 \cdot 2^{297}$

Contents. . .

Conclusion

- We present **depth-optimized** quantum circuits of CHAM.
  - We adopt in-place optimization technique and parallelize quantum additions.
- We find security bounds of CHAM against quantum attacks (based on NIST standards).
  - CHAM variants (-I, -III) that use 128-bit key: **Level 1** ($> 2^{157}$)
  - CHAM-V that uses 256-bit key: **Level 5** ($> 2^{285}$);
- We anticipate our work would be useful to the broader community **when analyzing the quantum security of ciphers** in the coming future.


Thank You!