Towards Optimizing Quantum Implementation of AES S-box

Doyoung Chung[1,2], Jooyoung Lee[2], Seungkwang Lee[2], and Dooho Choi[2,*]

[1] School of Computing, KAIST, Daejeon, 34141, Korea
{wordspqr, hicalf}@kaist.ac.kr
[2] Information Security Research Division, ETRI, Daejeon, 34129, Korea
{thisisdoyoung, skwang, dhchoi}@etri.re.kr

Abstract. Grover's search algorithm allows a quantum adversary to find a k-bit secret key of a block cipher by making $O(2^{k/2})$ block cipher queries. Resistance of a block cipher to such an attack is evaluated by quantum resources required to implement Grover's oracle for the target cipher. The quantum resources are typically estimated by the T-depth of its circuit implementation and the number of qubits used by the circuit (width).

Since the AES S-box is the only component which requires T-gates in a quantum implementation of AES, recent research has put its focus on efficient implementation of the AES S-box. However, any efficient implementation with low T-depth will not be practical in the real world without considering qubit consumption of the implementation.

In this work, we propose four methods of trade-off between time and space for the quantum implementation of the AES S-box. In particular, one of our methods turns out to use the smallest number of qubits among the existing methods, significantly reducing its T-depth.

Keywords: Quantum implementation, quantum cryptanalysis, Grover's algorithm, AES, multiplicative inversion.

1 Introduction

Most cryptographic primitives are under new threats with the advent of quantum computers. Public key cryptosystems such as RSA, ECDSA, and ECDH will be completely broken by Shor's algorithm [16], a quantum algorithm that solves the order finding problem in polynomial time. When it comes to symmetric key cryptography, exhaustive key search using Grover's algorithm [8] is becoming a new threat. For example, Grover's search algorithm allows a quantum adversary to find a k-bit secret key of a block cipher by making $O(2^{k/2})$ block cipher queries. Resistance of a block cipher to such an attack is evaluated by quantum resources required to implement Grover's oracle for the target cipher. The quantum resources are typically estimated by the circuit depth of the circuit implementation and the number of qubits used by the circuit (width) [7, 10, 13].

Quantum circuits involve error-prone qubits, and fault-tolerant quantum computation (FTQC) is made possible by using error correcting codes, where the surface code is one of the most feasible candidates for this purpose. Since T-gates are exceptionally expensive in the implementation of the surface code, *T-depth*, counting the number of sequential T-gates, dominates the overall efficiency of the quantum circuit in terms of the processing time [2]. For this reason, T-depth is widely used as a metric to estimate the time complexity of a quantum circuit.

The only component of AES that requires T-gates for its quantum implementation is the multiplicative inversion used in the AES S-box. Therefore, recent research [7, 10, 12, 13] has put its main focus on lightweight implementation of the multiplicative inversion, using tower field constructions of the underlying finite field $\mathbf{GF}(2^8)$. However, any efficient implementation with low T-depth will not be practical in the real world without considering its qubit consumption since qubits are arguably considered as the most valuable resources in quantum computation.

A classical implementation of the AES S-box based on a tower-field construction of $\mathbf{GF}(2^8)$ consists of XOR and AND gates. An XOR gate in a classical circuit can be converted to a CNOT gate in the corresponding quantum circuit, while an AND gate is converted to a Toffoli gate or a quantum AND gate. Since both gates are built on T-gates, the T-depth of the quantum circuit is determined by the AND-depth of the classical circuit.

1.1 Our Contribution

In this work, we propose four methods of trade-off between T-depth (time) and width (space) for the quantum implementation of the AES S-box. In particular, one of our methods turns out to use the smallest number of qubits among the existing methods, significantly reducing its T-depth. Precisely, it uses 32 qubits in a quantum circuit of T-depth 36. We note that the implementation by Langenberg et al. [13] uses the same number of qubits, while its T-depth is 120.

Two of our methods, balancing depth and width in their quantum implementation, improve on the "balanced" method proposed by Jaques et al. [10] in terms of both depth and width.

The key idea behind our methods is to adopt efficient tower-field constructions studied in [11] to reduce the AND-depth of multiplicative inversion over $\mathbf{GF}(2^8)$ from 6 (as proposed by Boyar et al. [3]) to 4 in a classical implementation. In order to further optimize the T-depth of the corresponding quantum implementation, we decomposed the 8-bit inversion into three 4-bit multiplications, one 4-bit inversion and other minor operations; we applied a different quantum implementation to each subfield operation, carefully recycling ancilla qubits, and hence reducing the overall depth-width of the resulting circuit. The cost of our methods is summarized in Table 1.

method	type	width	T-depth	# CNOT	# 1qCliff	# T	# M
Grassl et al. [7]		44	217	8683	1028	3584	0
Langenberg et al. [13]		32	120	314	4	385	0
Jaques et al. [10]	balanced	41	35	818	264	164	41
	minimum depth	137	6	654	184	136	34
this work	minimum width	32	36	958	366	268	40
	balanced 1	34	31	1000	408	232	46
	balanced 2	36	30	985	390	241	43
	minimum depth	54	20	1016	408	232	46

Table 1: Comparison of our methods with the existing ones.

1.2 Related Work

Quantum implementation of AES was first proposed by Grassl *et al.* [7]. They showed that every AES operation except S-box can be implemented by using Clifford gates only, and then evaluated the T-depth of the multiplicative inversion in the AES S-box. Later, Kim *et al.* improved on Grassl *et al*'s work by reducing the T-depth of the multiplicative inversion [12]. Based on classical implementations of AES with low depths of AND gates as studied in [3, 4], Langenberg *et al.* significantly reduced T-depth in its corresponding quantum implementation [13].

An AND gate is converted into a Toffoli gate in its quantum implementation. So far, the most shallow implementation of a Toffoli gate was known to have T-depth 3 [1]. In [10], they proposed to convert an AND gate into a quantum AND gate; a Toffoli gate is of T-depth 3, while a quantum AND gate is of T-depth 1 [6, 10]. By taking advantage of this property, Jaques *et al.* further improved on Langenberg *et al.*'s implementation in terms of T-depth. Specifically, they proposed two quantum circuits with different cost advantages. The first one, based on [3], reduces T-depth balancing time and space, while the other, based on [4], minimizes the depth of the circuit without considering space limit.

2 Preliminaries

2.1 S-box of AES

The AES S-box is a 8-bit permutation used in the nonlinear confusion layer of AES, where the set of 8-bit strings is identified with a finite field $\mathbf{GF}(2^8) = \mathbf{GF}(2)[x]/(x^8+x^4+x^3+x+1)$. This permutation can also be represented by a polynomial over $\mathbf{GF}(2)$; the input to this S-box is mapped to its multiplicative inverse in $\mathbf{GF}(2^8)$ (with zero mapped to itself by definition), followed by an affine transformation. Precisely, the S-box can be defined in the matrix form as follow:

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

where $\left[s_7, ..., s_0\right]$ is the output of S-box and $\left[b_7, ..., b_0\right]$ is the multiplicative inversion of the input of S-Box as a vector.

In classical computing environment, an 8-bit to 8-bit lookup table is generally used for S-box in most of software implementation of AES. However, in quantum computing environment, it is efficient to perform the multiplicative inversion and the affine transformation due to the limited number of qubits. The affine transformation which follows the multiplicative inversion can be computed in-place with only X- and CNOT gates. So, it does not consume additional T-depth and qubits.

For this reason, the main issue in implementing AES S-box quantum circuit is how to implement multiplicative inversion efficiently. The following explains a technique of tower-field construction to perform this operation efficiently.

2.2 Tower-field construction

A tower of fields is an extension sequence of some fields, $\mathbb{F}$. The tower-field construction for the implementation of the AES S-box is representing the operations over $\mathbb{F}_{2^{2k}}$ with operations over $\mathbb{F}_{2^k}$ recursively. The computational cost of AES operations that are performed on $\mathbf{GF}(2^8)$ can be reduced by using isomorphic composite fields which generated by the tower-field construction. When using subfield arithmetic, it is costly to convert the original into the isomorphic composite field and vice-versa. Such conversion and re-conversion can be implemented with only CNOT gates in quantum circuits by using PLU decomposition. One of the known tower-field representations is as follows [3]:

$\mathbf{GF}(2^2)$ by adjoining a root W of $x^2 + x + 1$ over $\mathbf{GF}(2)$
$\mathbf{GF}(2^4)$ by adjoining a root Z of $x^2 + x + W^2$ over $\mathbf{GF}(2^2)$
$\mathbf{GF}(2^8)$ by adjoining a root Y of $x^2 + x + WZ$ over $\mathbf{GF}(2^4)$.

In this paper, we will present our representation of tower-field construction that is improved for lightweight quantum circuits of the multiplicative inversion in AES S-box. This tower-field construction reduces AND-depth, imposing a dominant effect on the execution time of a quantum circuit.

2.3 Grover's algorithm

For a Boolean function $f : \{0,1\}^k \longmapsto \{0,1\}$, Grover's algorithm takes a *Grover's Oracle*, U_f, that implements $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, where $x \in \{0,1\}^k$ and $y \in \{0,1\}$. Basically, Grover's algorithm finds an element x_0 such that $f(x_0) = 1$ by repeatedly applying a *Grover iteration* defined below to the initial state $|\psi\rangle = H^{\otimes k}|0\rangle$.

$$Q = -H^{\otimes k}(I - 2|0\rangle\langle 0|)H^{\otimes k}U_f$$

After applying $\lfloor \frac{\pi}{4}\sqrt{\frac{K}{N}} \rfloor$ iterations on the initial state, a solution x_0 will be found with at least $1 - (N/K)$ probability by measuring the entire quantum register, where K is the total number of candidates ($K = 2^k$), and N is the number of solutions ($N = |\{x : f(x) = 1\}|$).

In [5], the authors analyzed that $r_k = \lceil k/128 \rceil$ known plaintext-ciphertext pairs are sufficient to avoid false positives in an exhaustive key search for AES-k, where $k \in \{128, 192, 256\}$. In order to build U_f, each plaintext-ciphertext pair requires AES and its inverse. This gives us that the number of AES instances should be twice as many as the number of plaintext-ciphertext pairs. For each key size k, the number of Grover's operations is then given by

- 2 AES instances for $k = 128$
- 4 AES instances for $k = 192$
- 4 AES instances for $k = 256$.

2.4 Quantum AND gate

In [10], the authors used a T-depth 1 circuit for an AND gate which is a combination of Selinger [15], and Gidney [6], and that was designed by Mathias Soeken.

This gate requires one more ancilla qubit, instead of reducing T-depth compared to Toffoli gate. It has an assymetric relationship with its dagger gate. Its dagger gate requires only 3 qubits same as Toffoli gate, but does not include any T-depth.

The diagrams for the quantum AND gate and AND† gate are depicted in Fig.1 and Fig.2.

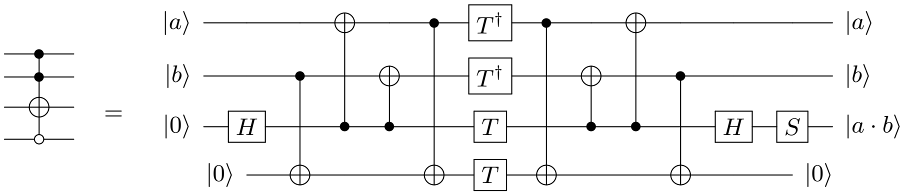


Fig. 1: Quantum AND gate.

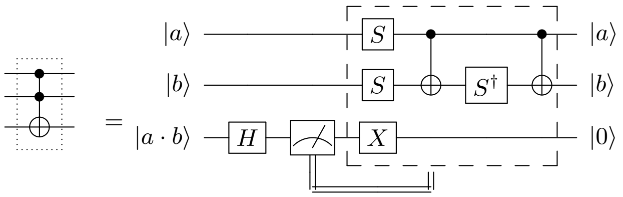


Fig. 2: Quantum AND† gate.

3 Improvement on tower-field construction

In this section, we improve the tower-field construction by using the following representations of Galois fields:

$$\begin{cases} \mathbf{GF}(2^8): & x^8 + x^4 + x^3 + x + 1 \\ \mathbf{GF}(2^4): & x^4 + x + 1 \\ \mathbf{GF}(2^2): & x^2 + x + 1, \end{cases}$$

and we suggest the following composite fields $\mathbf{GF}((2^4)^2)$ and $\mathbf{GF}((2^2)^2)$:

$$\begin{cases} \mathbf{GF}((2^4)^2): & y^2 + y + \lambda \\ \mathbf{GF}((2^2)^2): & z^2 + z + \phi, \end{cases}$$

where

- $\lambda := \omega^{11} = \omega^3 + \omega^2 + \omega = \{1110\}_2$
- ω is a root of $x^4 + x + 1$
- $\phi = \{10\}_2$
- ϕ is a root of $x^2 + x + 1$.

3.1 Isomorphic mapping quantum circuit

The $\mathbf{GF}(2^8)$ of AES and other representations of $\mathbf{GF}(2^8)$ are isomorphic. The matrix of mapping between $\mathbf{GF}(2^8)$ of AES and our representation of $\mathbf{GF}(2^8) = \mathbf{GF}((2^4)^2)$, and its inverse mapping are defined as M and M^{-1}:

$$M = \begin{bmatrix} 1&0&1&0&0&0&0&0 \\ 1&0&1&0&1&1&0&0 \\ 1&1&0&1&0&0&1&0 \\ 0&1&1&1&0&0&0&0 \\ 0&0&0&1&0&1&0&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&1&1&0 \\ 0&1&1&1&0&0&0&1 \end{bmatrix}, \text{ and } M^{-1} = \begin{bmatrix} 1&0&1&1&0&1&0&0 \\ 1&0&0&1&1&1&1&0 \\ 0&0&1&1&0&1&0&0 \\ 1&0&1&1&1&0&1&0 \\ 0&1&1&1&0&0&1&0 \\ 1&0&1&1&0&0&1&0 \\ 1&0&1&1&0&0&0&0 \\ 0&0&0&1&0&0&0&1 \end{bmatrix}.$$

Next, the matrix of mapping between the proposed $\mathbf{GF}(2^4)$ and $\mathbf{GF}((2^2)^2)$ representations, and its inverse mapping are given by

$$M_4 = \begin{bmatrix} 1&0&0&0 \\ 1&1&1&0 \\ 1&1&0&0 \\ 0&0&0&1 \end{bmatrix}, \text{ and } M_4^{-1} = \begin{bmatrix} 1&0&0&0 \\ 1&0&1&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{bmatrix}.$$

The matrix α of the isomorphic mapping from the input qubits for AES $\mathbf{GF}(2^8)$ to our composite field is then obtained by the multiplication of the matrices above:

$$\alpha = \begin{bmatrix} M_4 & 0 \\ 0 & M_4 \end{bmatrix} \circ M = \begin{bmatrix} 1&0&1&0&0&0&0&0 \\ 1&1&0&1&1&1&1&0 \\ 0&0&0&0&1&1&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&0&0&1&0&1&0&0 \\ 1&0&0&1&0&0&0&0 \\ 1&0&0&1&0&1&1&0 \\ 0&1&1&1&0&0&0&1 \end{bmatrix}.$$

Here, α can be implemented with only CNOT gates in a quantum circuit by using PLU decomposition. For example, $\alpha \coloneqq P \cdot L \cdot U$, where

$$\mathrm{P} \coloneqq \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}, \mathrm{L} \coloneqq \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&1&0&0&1&0&0&0 \\ 1&0&1&0&0&1&0&0 \\ 0&0&0&0&1&0&1&0 \\ 0&1&0&0&1&0&0&1 \end{bmatrix}, \mathrm{U} \coloneqq \begin{bmatrix} 1&0&1&0&0&0&0&0 \\ 0&1&1&1&1&1&1&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&1&0&1&0&0 \\ 0&0&0&0&1&1&1&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}.$$

Based on the PLU decomposition above, the quantum circuit of the isomorphic mapping, $\alpha \coloneqq (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0) \mapsto ((b_{hh1}, b_{hh0}, b_{hl1}, b_{hl0}), (b_{lh1}, b_{lh0}, b_{ll1}, b_{ll0}))$ is depicted as Fig. 3.

3.2 Inversion method with our composite field

The multiplicative inversion for composite fields $\mathbf{GF}((2^n)^m)$ can be computed as a combination of operations in the sub-fields $\mathbf{GF}(2^n)$ [14].

For $P \in \mathbf{GF}((2^n)^m)$ and $r = \frac{2^{nm}-1}{2^n-1}$, $P^{-1} = (P^r)^{-1}P^{r-1}$ where $P^r \in \mathbf{GF}(2^n)$ [9]. Throughout this paper, the multiplicative inversion, P^{-1}, is represented as $P^{-1} = (P^{17})^{-1} \cdot P^{16}$ for $n = 4$ and $m = 2$. Then P^{-1} can be calculated by the following four steps:

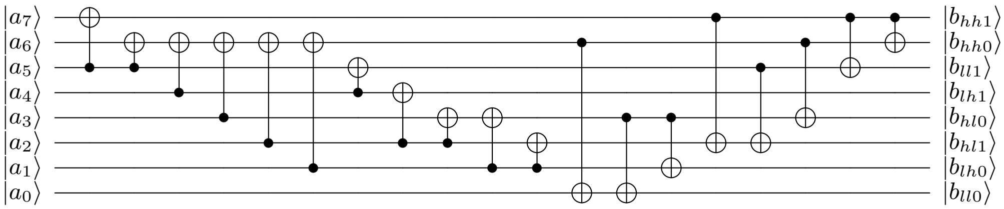


Fig. 3: Quantum circuit of the isomorphic mapping α.

1. $P^{r-1} = P^{16}$
2. $P^{r} = (P^{16}) \cdot P$
3. Compute $(P^{r})^{-1}$ in $\mathbf{GF}(2^4)$
4. Compute $(P^{r})^{-1} \cdot P^{r-1}$ using $\mathbf{GF}((2^2)^2)$ arithmetic

Step 1. First, we need to compute P^{16}.

For $P = ((b_{hh1}, b_{hh0}, b_{hl1}, b_{hl0}), (b_{lh1}, b_{lh0}, b_{ll1}, b_{ll0})) = b_h y + b_l$,

$P^{16} = (b_h y + b_l)^{16} = b_h y^{16} + b_l$
$y^{16} = y + 1$

Thus, $P^{16} = b_h y + (b_h + b_l)$.

Step 2. Then we need to compute $P^{r} = P^{16} \cdot P$. In here, $\lambda = map_4(\lambda) = \{1100\}_2$ in $\mathbf{GF}((2^2)^2), z^2 + z + \phi, \phi = \{10\}_2$

$$\begin{aligned}
P = P^{16} \cdot P &= (b_h y + (b_h + b_l))(b_h y + b_l) \\
&= b_h^2 y^2 + b_h^2 y + (b_h + b_l) b_l \\
&= b_h^2 (y + \lambda) + b_h^2 y + (b_h + b_l) b_l \\
&= b_h^2 \times \lambda + (b_h + b_l) b_l .
\end{aligned}$$

Squaring, multiplication by λ, and multiplication in $\mathbf{GF}((2^2)^2)$ in our composite field should be implemented in quantum circuit to calculate above equations. The squaring can be calculated as

$$\begin{aligned}
(p_h z + p_l)^2 &= (p_{h1} x + p_{h0})^2 z^2 + (p_{l1} x + p_{l0})^2 \\
&= (p_{h1} x^2 + p_{h0}) z^2 + (p_{l1} x^2 + p_{l0}) \\
&\coloneqq (q_{h1} + q_{h0}) z + (q_{l1} + q_{l0}) .
\end{aligned}$$

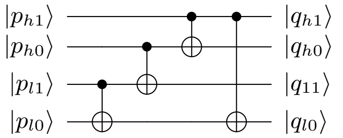


Fig. 4: Quantum circuit of the squaring in $\mathbf{GF}((2^2)^2)$.

The quantum circuit for the above formula, squaring, is shown as Fig. 4. The multiplication by λ can be calculated as

$$\begin{aligned}(p_h z + p_l) \times \lambda &= ((p_{h0} + p_{l0})x + p_{h1} + p_{h0} + p_{l1} + p_{l0})z + p_{h1}x + p_{h0} \\ &\coloneqq (q_{h1} + q_{h0})z + (q_{l1} + q_{l0}).\end{aligned}$$

The quantum circuit for the above formula, multiplication by λ, is depicted as Fig. 5. Both quantum circuits for squaring and multiplication by λ are implemented using only CNOT gates and wiring. The rest of arithmetic operations in $\mathbf{GF}((2^2)^2)$ is multiplication which can be calculated as

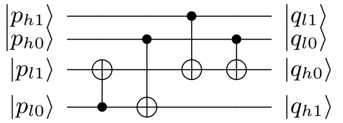


Fig. 5: Quantum circuit of the multiplication by λ in $\mathbf{GF}((2^2)^2)$.

$$\begin{aligned}(p_h z + p_l)(q_h z + q_l) &= ((p_h + p_l)(q_h + q_l) + p_l q_l)z + (p_h q_h \phi + p_l q_l) \\ &\coloneqq r_h z + r_l.\end{aligned}$$

To minimize depth-width of the quantum circuit of multiplicative inversion in $\mathbf{GF}(2^8)$, we adopt various implementation of quantum circuit for multiplication in $\mathbf{GF}((2^2)^2)$. We present those quantum circuits in Section 4. To clarify the concept of the multiplication in $\mathbf{GF}((2^2)^2)$, we present the classical circuit diagram in Fig. 6.

Step 3. We compute $(P^{16})^{-1}$ in $\mathbf{GF}((2^2)^2)$, $z^2 + z + \phi$, $\phi = \{00\}_2$.

Given $p_h, p_l, q_h, q_l \in \mathbf{GF}(2^2) = \mathbf{GF}(2)[x]/(x^2+x+1)$ and $(p_h z + p_l), (q_h z + q_l) \in \mathbf{GF}((2^2)^2)$, suppose that $(p_h z + p_l)^{-1} = (q_h z + q_l)$. Then, we have

$$\begin{aligned}(p_h z + p_l)(q_h z + q_l) &= ((p_h + p_l)(q_h + q_l) + p_l q_l)z + (p_h q_h \phi + p_l q_l) \\ &= 1,\end{aligned}$$

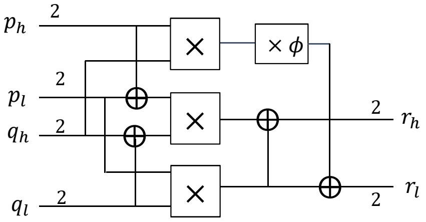


Fig. 6: Classical circuit of the multiplication in $\mathbf{GF}((2^2)^2)$.

and this gives us $(p_h + p_l)q_h + p_h q_l = 0$, and $p_h q_h \phi + p_l q_l = 1$. Thus, it is easy to know that $q_h = p_h d^{-1}$ and $q_l = (p_h + p_l)d^{-1}$, where $d = p_h^2 \phi + p_l(p_h + p_l)$. The classical circuit diagram for the multiplicative inversion on $\mathbf{GF}(2^2)^2)$ designed on the above formula is depicted in Fig. 7.

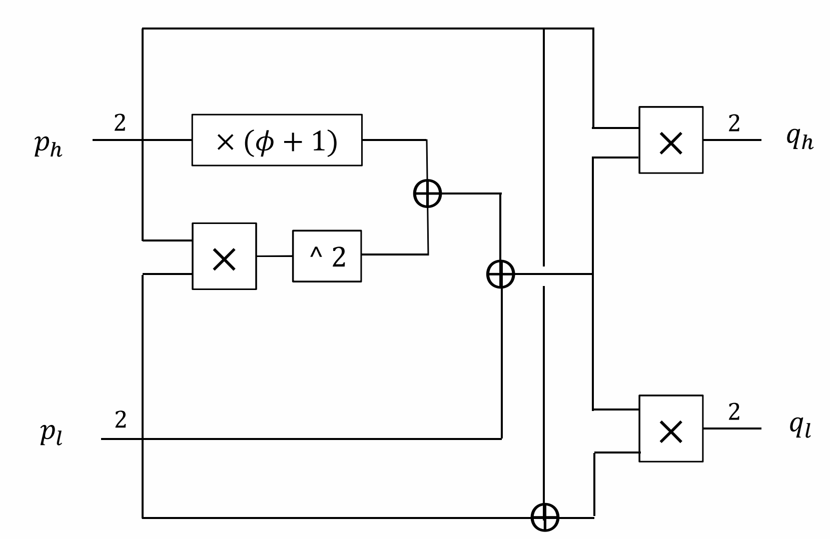


Fig. 7: Circuit diagram of the multiplication inversion in $\mathbf{GF}((2^2)^2)$.

Step 4. Finally, we compute $(P^r)^{-1} \cdot P^{16}$ in $\mathbf{GF}((2^2)^2)$, $z^2 + z + \phi$, $\phi = \{00\}_2$.

For $p = (p_h z + p_l)$ in $\mathbf{GF}((2^2)^2)$, where $p_h, p_l \in \mathbf{GF}(2^2)$, $x^2 + x + 1$,

$$\begin{aligned} p \cdot P^{16} &= p \cdot (b_h y + (b_h + b_l)), \\ &= p \cdot b_h y + p \cdot (b_h + b_l), \end{aligned}$$

because $P^{16} = b_h y + (b_h + b_l)$ in Step 1.

Following Itoh and Tsujii's inversion algorithm, $p \cdot P^{16}$ is the multiplicative inversion for the composite field, $\mathbf{GF}(2^8)$, which we suggested. The classical circuit diagram for calculating $p \cdot P^{16}$ is illustrated in Fig. 8. The square box written × means a multiplication operation in $\mathbf{GF}(2^4)$, and the others x^2, x^{-1}, and $\times\lambda$ represent squaring, multiplicative inversion, and multiplication by λ in $\mathbf{GF}(2^4)$, respectively.

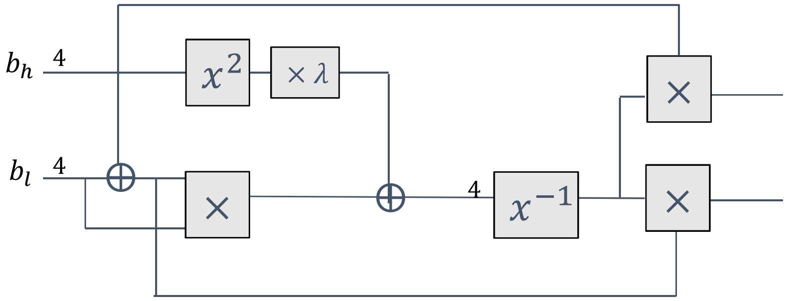


Fig. 8: Circuit diagram of the multiplication inversion in $\mathbf{GF}(2^8)$.

Summary. A hardware implementation of Galois field operations generally uses AND and XOR (some NOR) gates in classical computing. Among them, an XOR gate can be converted into a quantum gate as a relatively inexpensive CNOT gate. In the case of AND gate, it is converted to a Toffoli gate or a quantum AND gate [10]. Currently, a Toffoli gate requires T-depth 3 [1] and a quantum AND gate requires T-depth 1 [10]. Therefore, the AND-depth in a classical computing environment decides the T-depth in a quantum computing environment, and a hardware implementation with a low AND-depth will be preferred to reduce operation time when converting to a quantum circuit.

Compared to prior work [3], our method reduces the AND-depth of the multiplicative inversion from 6 to 4, which is about a 33% decreasement. Note that we only modified the multiplicative inversion; the rest part for constructing the S-box, including the isomorphic mapping, its inverse operation, and the affine mapping, can be implemented with only XOR gates. Therefore, the AND-depth of the S-box using the proposed multiplicative inversion remains the same, 4.

4 Proposed Quantum Circuit

In order to build up a shallow implementation of multiplicative inversion in $\mathbf{GF}(2^8)$, we should take into account depth-width trade-offs on the corresponding quantum arithmetic circuits of $\mathbf{GF}(2^4)$ and $\mathbf{GF}(2^2)$. From now on, we propose several low-cost quantum circuits of multiplication inversion in $\mathbf{GF}(2^8)$ with a bottom-up approach. Meanwhile, we mainly focus on the optimal combinations of quantum circuits for composite Galois field operations under consideration of depth-width trade-offs.

4.1 $\mathbf{GF(2^2)}$ arithmetic quantum circuits

The operations in $\mathbf{GF}(2^2)$ are the basic operation in our scheme. We utilize multiplication and its inverse (dagger) operation. For $(a_1x + a_0), (b_1x + b_0) \in \mathbf{GF}(2^2), x^2 + x + 1$, the multiplication operation can be written as

$$(a_1x + a_0)(b_1x + b_0) = ((a_1 + a_0)(b_1 + b_0) + a_0b_0)x + a_1b_1 + a_0b_0.$$

This equation can be implemented as a quantum circuit using CNOT, Toffoli and quantum AND gates. A Toffoli gate performs CCNOT for either $|0\rangle$ or $|1\rangle$ to the target register. In contrary, a quantum AND gate does CCNOT when the input state $|0\rangle$ is given to the target register. Although a quantum AND gate uses one more qubit temporarily, it reduces T-depth compared to Toffoli gate [10]. By using these characteristics of Toffoli and quantum AND gates, quantum circuits for multiplication in $\mathbf{GF}(2^2)$ of various depth-width can be designed.

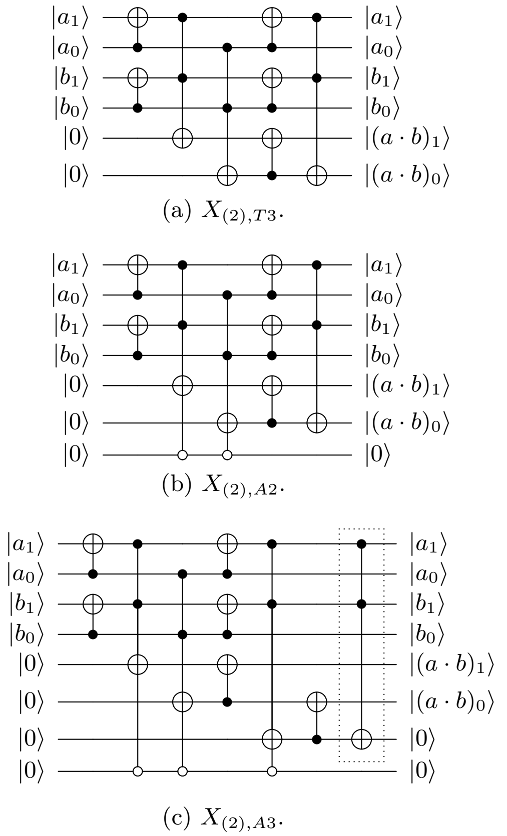
We propose three types of quantum circuits for multiplication in $\mathbf{GF}(2^2)$ and two types of quantum circuits for its dagger operation. The three types of quantum circuits for multiplication are summarized in Table 2, and are illustrated in Fig. 9. In addition, the two types of quantum circuits for multiplication dagger are summarized in Table 3, and are described in Fig. 10.

Notation	Composition of gates
$X_{(2),T3}$	Toffoli × 3
$X_{(2),A2}$	AND × 2, Toffoli × 1
$X_{(2),A3}$	AND × 3, AND† × 1

Table 2: Three quantum circuits of multiplication in $\mathbf{GF}(2^2)$.

4.2 $\mathbf{GF(2^4)}$ arithmetic quantum circuits

The arithmetic in $\mathbf{GF}(2^4)$ is performed as $\mathbf{GF}((2^2)^2)$ as explained in Section. 3. We use constant(λ) multiplication, squaring, multiplication and multiplicative



(a) $X_{(2),T3}$.

(b) $X_{(2),A2}$.

(c) $X_{(2),A3}$.

Fig. 9: Three quantum circuits for multiplication in $\mathbf{GF}(2^2)$.

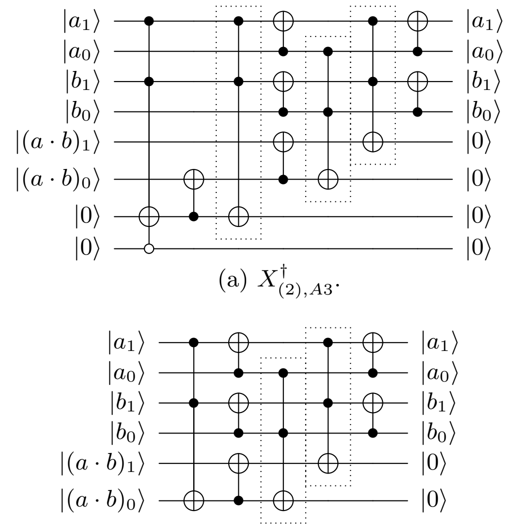
inversion in $\mathbf{GF}(2^4)$. The quantum circuits of squaring and constant multiplication were previously described in Fig. 4 and Fig. 5, respectively. Note that those quantum circuits were implemented by using only Clifford gates. Now we describe our improvement on the quantum circuits of multiplication and multiplicative inversion in $\mathbf{GF}(2^4)$.

Multiplication As depicted in Fig. 6, multiplication in $\mathbf{GF}(2^4)$ consists of addition, constant(ϕ) multiplication, and three times of multiplication in $\mathbf{GF}(2^2)$. The quantum circuits for addition and constant multiplication in $\mathbf{GF}(2^2)$ can be implemented with only CNOT gates. Three times of multiplication in $\mathbf{GF}(2^2)$ can be performed either in parallel or in a combination of parallel and series, depending on the available amount of qubits. Also, the arrangement of the quantum circuits for multiplication in $\mathbf{GF}(2^2)$ introduced in Section 4.1 will have an influence on depth-width trade-offs.

Hereafter, we analyze the number of multiplication in $\mathbf{GF}(2^2)$ that can be performed in parallel, and the proper type of quantum circuits for the multipli-

Notation	Composition of gates
$X^\dagger_{(2),A3}$	AND × 1, AND† × 3
$X^\dagger_{(2),A2}$	Toffoli × 1, AND† × 2

Table 3: Two quantum circuits of multiplication dagger in $\mathbf{GF}(2^2)$.



(a) $X^\dagger_{(2),A3}$.

(b) $X^\dagger_{(2),A2}$.

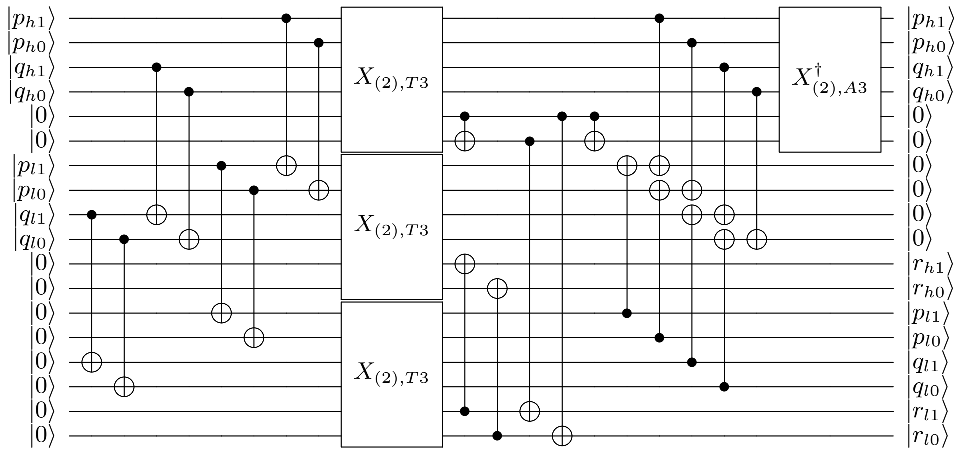
Fig. 10: Two quantum circuits of multiplication dagger in $\mathbf{GF}(2^2)$.

cation. These will have a significant effect on depth-width cost of the quantum circuit for multiplication in $\mathbf{GF}(2^4)$.

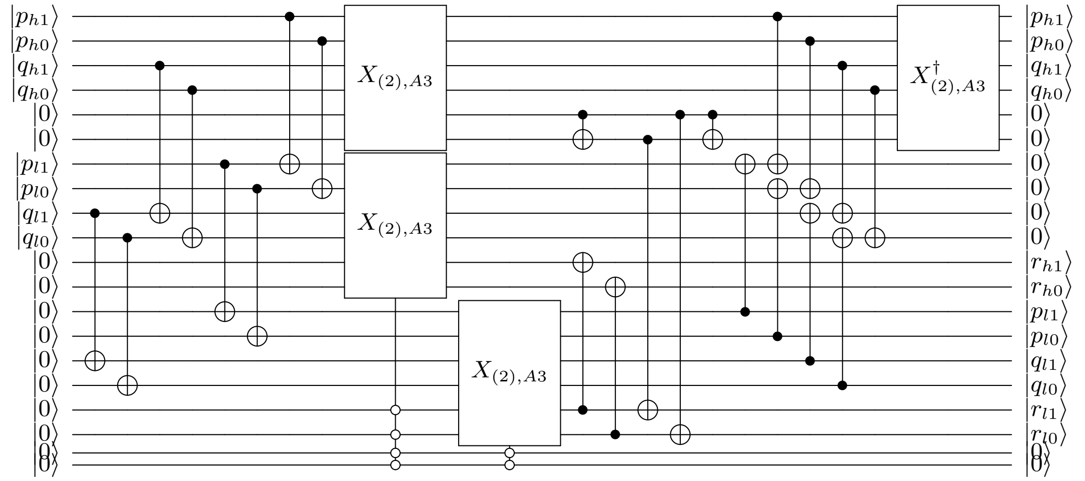
We use four types of quantum circuits for multiplication in $\mathbf{GF}(2^4)$. We denote each of them by $X_{(4),T3}$, $X_{(4),AA3}$, $X_{(4),A2}$, and $X_{(4),A3}$, and these are shown in Fig. 11. Each name of the circuits characterizes the type of multiplication in $\mathbf{GF}(2^2)$ and the arrangement in the circuit.

On the other hand, we use only one type of quantum circuit for dagger operation of multiplication in $\mathbf{GF}(2^2)$, $X^\dagger_{(2),A3}$. Because the proposed quantum circuits for multiplication in $\mathbf{GF}(2^4)$ have enough qubits to perform $X^\dagger_{(2),A3}$ in all circuit design, we use $X^\dagger_{(2),A3}$ which has the smallest T-depth.

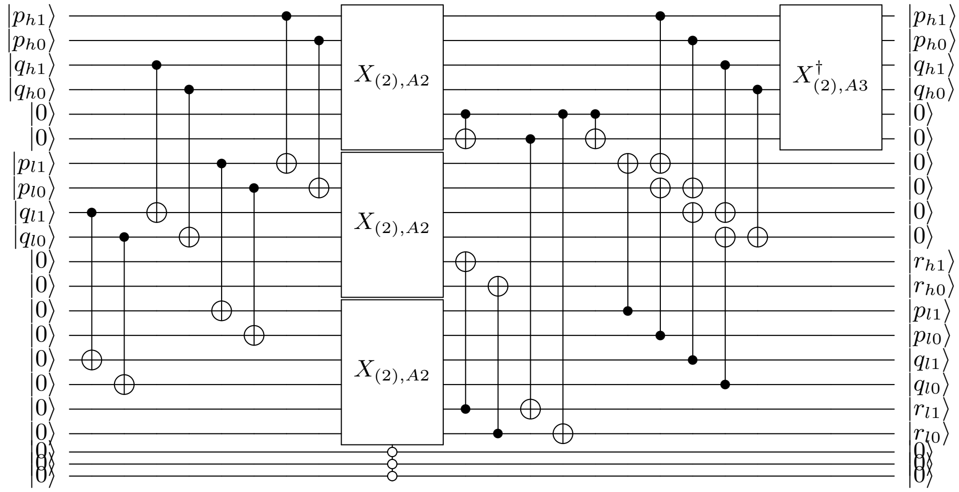
The three types of quantum circuit for multiplication in $\mathbf{GF}(2^2)$ are arranged to reduce the T-depth as much as possible within the available number of qubits. The quantum circuits for multiplication in $\mathbf{GF}(2^2)$ arranged in the same column represent parallel execution, and those arranged in different columns as shown in Fig. 11b represent operations performed in series.



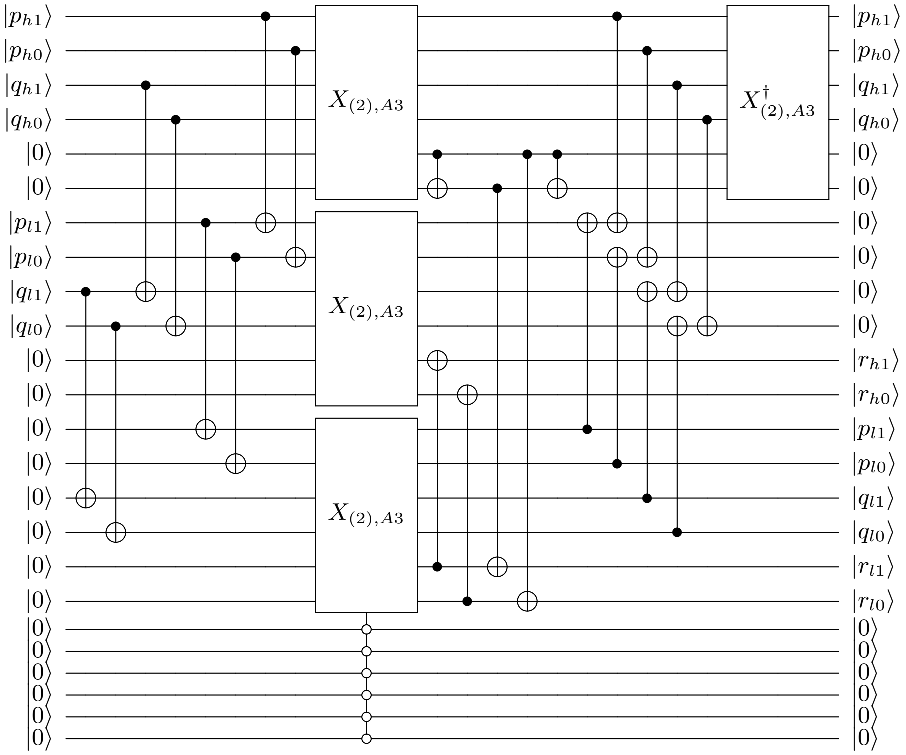
(a) $X_{(4),T3}$.



(b) $X_{(4),AA3}$.



(c) $X_{(4),A2}$.



(d) $X_{(4),A3}$.

Fig. 11: Four quantum circuits for multiplication in $GF(2^4)$.

Multiplicative inversion The result of the multiplicative inversion in $\mathbf{GF}(2^4)$ are $q_h = p_h d^{-1}$ and $q_l = (p_h + p_l)d^{-1}$, where, $d^{-1} = p_h(\phi + 1) + p_l + p_h^2 p_l^2$. To calculate q_h and q_l, the quantum circuit of multiplicative inversion in $\mathbf{GF}(2^4)$ requires three quantum circuits for multiplication in $\mathbf{GF}(2^2)$ and two of three can be executed parallel. This quantum circuit requires additional two qubits for saving d^{-1} which is required during clean-up process of the multiplicative inversion in $\mathbf{GF}(2^8)$.

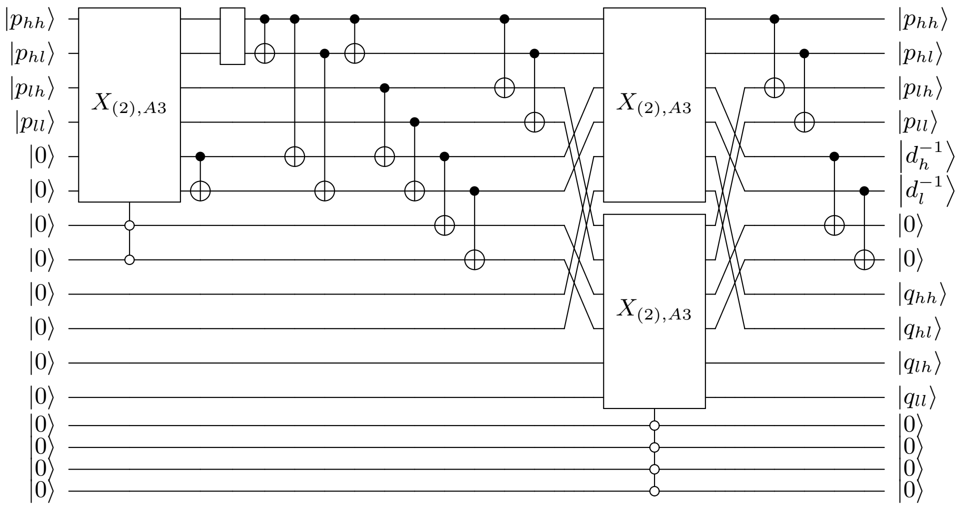


Fig. 12: Quantum circuit of the multiplicative inversion in $\mathbf{GF}(2^4)$.

The quantum circuit for multiplicative inversion in $\mathbf{GF}(2^4)$ is depicted in Fig.12. Because the quantum circuit for multiplicative inversion in $\mathbf{GF}(2^8)$ has sufficient number of available qubits, we use only $X_{(2),A3}$ to build the quantum circuit for multiplicative inversion in $\mathbf{GF}(2^4)$.

4.3 GF(2^8) multiplicative inversion quantum circuits

The multiplicative inversion in $\mathbf{GF}(2^8)$ consists of squaring, constant(λ) multiplication, multiplicative inversion, and multiplications in $\mathbf{GF}(2^4)$ as depicted in Fig. 8. Squaring and constant multiplication can be implemented by CNOT gates only.

The quantum circuit for multiplicative inversion in $\mathbf{GF}(2^8)$ requires clean-up process. The clean-up process means to make the qubits that store the intermediate value generated during the calculation process into a specific state (usually $|0\rangle$). This process is necessary to improve the reusability of quantum resources such as qubits when the proposed quantum circuit is combined in other quantum circuits such as AES S-box. The quantum circuit for clean-up process is similar to that of the target operation in reverse, but it is not symmetric because the result of the target operation must be maintained. Due to this reason, the suggested quantum circuit has a complicated structure compared to the circuit diagram in Fig. 8.

As shown in Fig. 13 to Fig. 16, the quantum circuit can be largely divided into two parts. Forward operation part calculates multiplicative inversion in $\mathbf{GF}(2^8)$ and another part performs clean-up process. In the forward operation, unlike the circuit diagram in Fig.8, one more $\mathbf{GF}(2^4)$ multiplication is included. This is because keeping $b_h \times b_l$ reduces T-depth of the entire quantum circuit which includes clean-up process.

What we mainly describe here are what kinds of quantum circuit for multiplication in $\mathbf{GF}(2^4)$ are used and how these are arranged in the quantum circuit for multiplicative inversion in $\mathbf{GF}(2^8)$. These two factors have a significant effect on the depth-width cost of the quantum circuit for multiplicative inversion in $\mathbf{GF}(2^8)$.

We depict the minimum width quantum circuits in Fig.13. This is the starting point for describing the four quantum circuits that we have presented; minimum width, balanced 1, 2, and minimum depth. While designing quantum circuits, blank squares and circles are used. Blank square means logical swap which change the index of qubits conceptually to improve understanding. In the actual quantum circuit implementations, such index change of qubits is not mandatory. Blank circle represents ancilla qubits as in Fig.11a to Fig.11d.

In the figure, each horizontal line corresponds to two qubits and receives corresponding b_{hh}, b_{hl}, b_{lh}, and b_{ll} as 8-bit input. These input are b_{hh} to b_{ll} from above. We denote as $b_h = b_{hh}|b_{hl}$ and $b_l = b_{lh}|b_{ll}$ for convenience. First, we copied b_h using two CNOT gates. Then we change state of the qubits from b_h to $b_h + b_l$ with two CNOT gates. By applying $X_{(4),A3}$ to the qubits corresponding $b_h + b_l$ and b_l, $(b_h + b_l)b_l$ is calculated. We change the index of qubits by applying logical swap; move b_l to the top 2-qubits of the blank square, move $b_h + b_l$ to the bottom 2-qubits of the blank square, and keep the index of $(b_h + b_l)b_l$. Now, the b_hb_l is calculated by applying $X_{(4),A3}$ to the b_h and b_l. The qubits corresponding to b_l become 0 by applying four CNOT-gates. We generate $(b_h + b_l)b_l + b_h^2\lambda$ using squaring, constant(λ) multiplication, and two CNOT gates. Before applying $(b_h + b_l)b_l + b_h^2\lambda$ to $x^{-1}_{(4),A3}$, we change index of these qubits to the bottom of blank square. Then the output of $x^{-1}_{(4),A3}$ to be d^{-1}, q, and $(b_h + b_l)b_l + b_h^2\lambda$ from above. The qubits corresponding to $(b_h + b_l)b_l + b_h^2\lambda$ become b_l by applying four CNOT gates and x^{-2}. While perform such operation, $b_h^2\lambda$ becomes b_h by using the dagger of $x^2\lambda$. Then we change b_l to 0 by applying four CNOT gates. Now we calculate r_l which the lower 4-bit of the result of suggested $\mathbf{GF}(2^8)$ multiplicative inversion. This value can be calculated by applying $X_{(4),A2}$ to q and $b_h + b_l$. Before applying $X_{(4),A2}$, we swap the index of q and 0. Thus the output of $X_{(4),A2}$ to be r_l, q, and $b_h + b_l$ from above. At this point we apply a complex logical swap, $(b_h, 0, 0, 0, 0, 0, r_l, q, b_h + b_l) \mapsto (0, 0, b_h, q, 0, b_h + b_l, 0, 0, r_l)$. In here, $b_h, r_l, q, b_h + b_l$ are 4-qubits and 0 is 2-qubits. By applying two CNOT gates, $b_h + b_l$ becomes b_l. Finally, we compute r_h which the higher 4-bit of the result of suggested $\mathbf{GF}(2^8)$ multiplicative inversion by applying $X_{(4),T3}$ to b_h

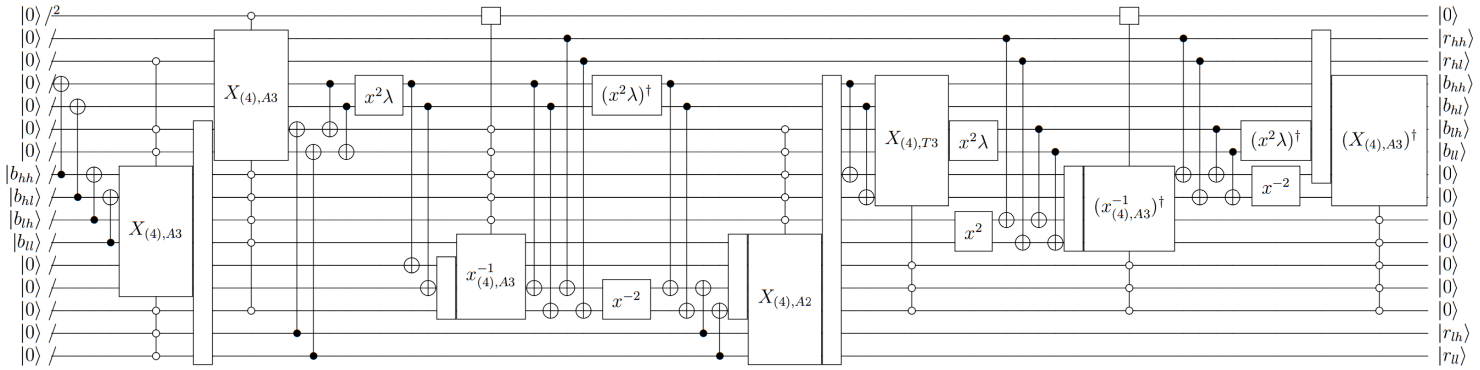


Fig. 13: Quantum circuit for the multiplicative inversion in $\mathbf{GF}(2^8)$, minimum width.

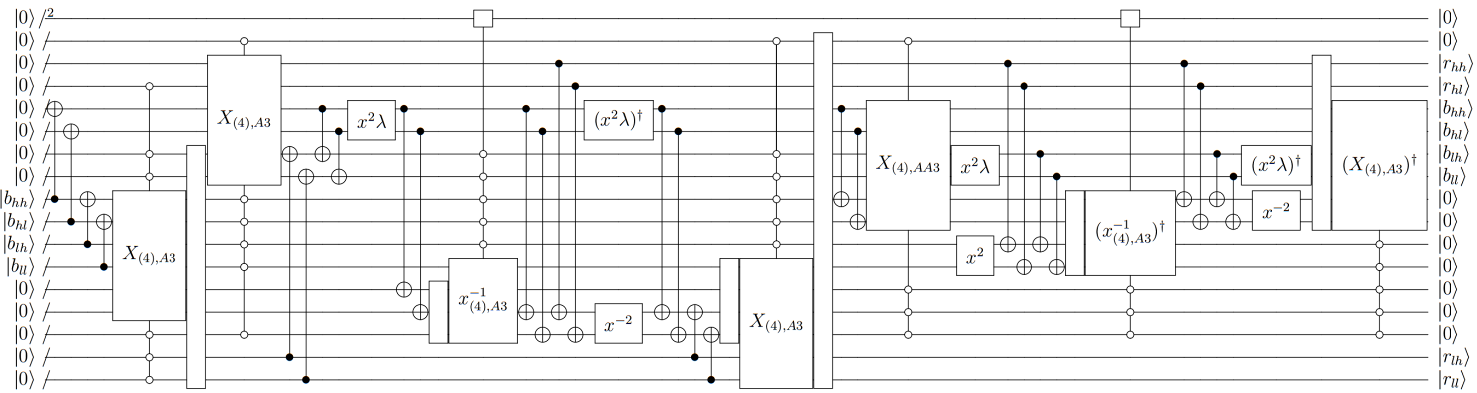


Fig. 14: Quantum circuit for the multiplicative inversion in $\mathbf{GF}(2^8)$, balanced 1.

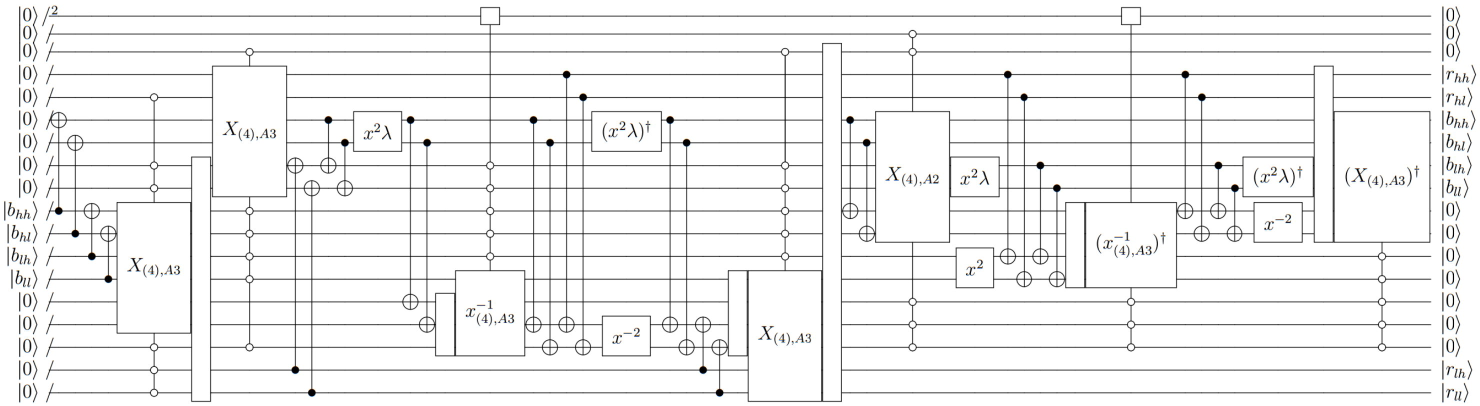


Fig. 15: Quantum circuit for the multiplicative inversion in $\mathbf{GF}(2^8)$, balanced 2.

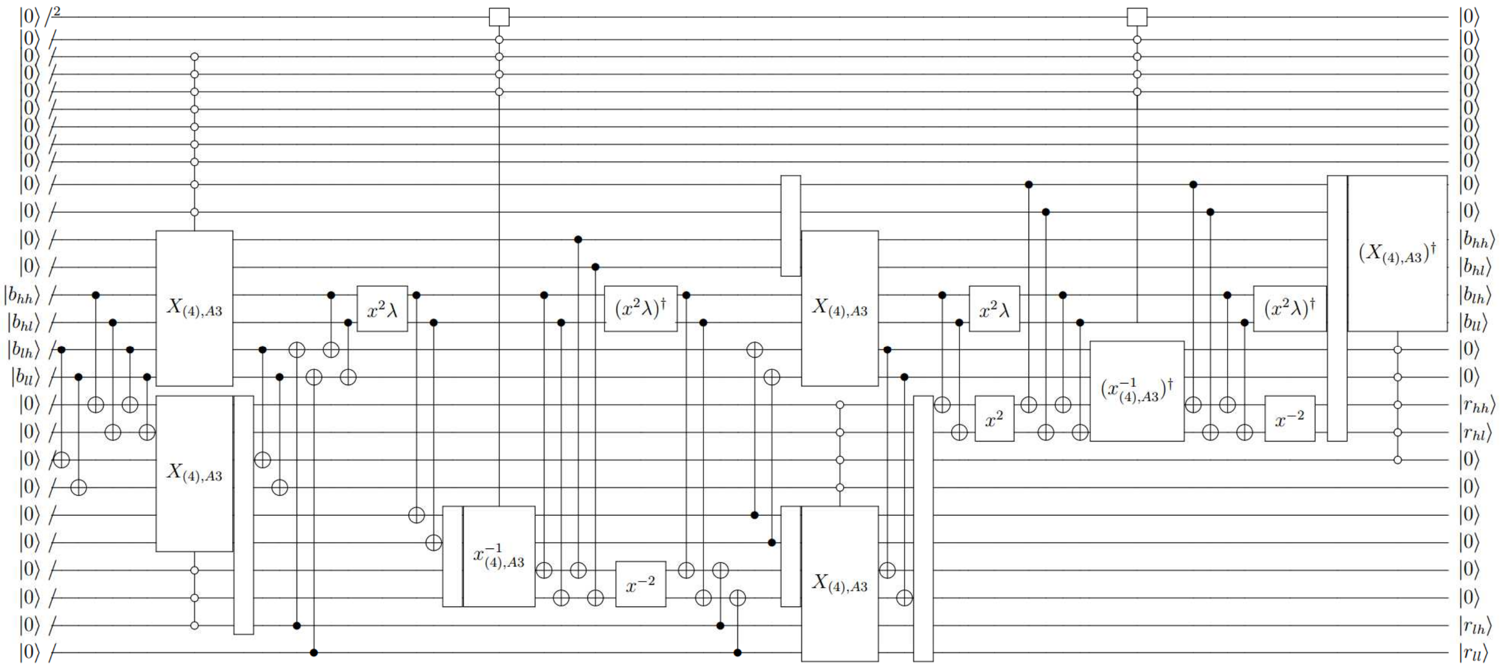


Fig. 16: Quantum circuit for the multiplicative inversion in $\mathbf{GF}(2^8)$, minimum depth.

and q. In here, we have computed r_h and r_l which the upper and lower 4-bit of the result, thus the forward operation is completed.

The rest of proposed quantum circuit is for clean-up process. By clean-up process, we restore all of qubits except r_h, r_l, b_h and b_l to 0 for reusing. In here, we regenerate $(b_h + b_l)b_l + b_h^2\lambda$ by applying $x^2\lambda$ on b_h, x^2 on b_l and add b_hb_l using four CNOT gates. Then we swap this value with q and apply these to $(x^{-1}_{(4),A3})^\dagger$ with d^{-1}. By applying $(X^{-1}_{(4),A3})^\dagger$, d^{-1} and q are restored as 0. The remaining value, $(b_h + b_l)b_l + b_h^2\lambda$ is changed to b_l^2 by using four CNOT gates. At this point we return $b_h^2\lambda$ to b_h by using $(x^2\lambda)^\dagger$ and b_l^2 to b_l by using x^{-2}. Finally, b_hb_l is the only qubit to be cleaned-up. To clean-up this qubit, we change the indices thus move r_h to the top of blank square and move b_hb_l to the input of $(X_{(4),A3})^\dagger$. Then b_hb_l is restored as 0, and the state of qubits are only r_h, b_h, b_l, and r_l except 0s.

For the other two quantum circuits, balanced 1 and 2, the process is same except that the $\mathbf{GF}(2^4)$ multipliers, which has smaller T-depth, are used as the number of available qubits increases. For example, the quantum circuit requiring 34-qubits uses $X_{(4),A3}$ and $X_{(4),AA3}$ instead of $X_{(4),A2}$ and $X_{(4),T3}$ in the center of the quantum circuit requiring 32-qubits. And the quantum circuit requiring 36-qubits uses $X_{(4),A2}$ instead of $X_{(4),AA3}$ in the quantum circuit requiring 34-qubits.

On the other hand, the quantum circuit with minimum depth has a bigger change. The two pairs of $\mathbf{GF}(2^4)$ multipliers are previously operated in sequential, but each pair of these is operated in parallel because sufficient number of qubits are provided. This change occurs minor changes in other parts of the circuit. For each $\mathbf{GF}(2^4)$ multiplier pairs, the values that used for both parallel multipliers, such as b_l and q, have to be copied before multiplications and restored to 0 after multiplications. To do this, several CNOT gates are added.

In section 5, we analyze depth-width for four quantum circuits for multiplicative inversion in $\mathbf{GF}(2^8)$ which have significant improvement. Each is a quantum circuit that minimizes qubit consumption, a quantum circuit that minimizes T-depth, and quantum circuits that balances width and T-depth.

4.4 Affine transformation quantum circuit

The affine transformation is expressed as the following equation.

$$\{b\} = M\{b^{'}\} \oplus \{v\}$$

where M is the matrix below, $\{v\}$ is a fixed vector and $\{b^{'}\} = (b^{'}_0, b^{'}_1, b^{'}_2, b^{'}_3, b^{'}_4, b^{'}_5, b^{'}_6, b^{'}_7)$ is the result of the multiplicative inversion for the input of AES S-box.

$$M\{b^{'}\} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix} \begin{bmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ b_3^{'} \\ b_4^{'} \\ b_5^{'} \\ b_6^{'} \\ b_7^{'} \end{bmatrix} \text{ and } \{v\} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

The matrix M can be decomposed into three matrices, P, L and U by PLU decomposition. These matrices can be implemented in a quantum circuit with only CNOT gates. On the other hand, the modular addition of $\{v\}$ can be implemented in a quantum circuit with only X-gates. In Fig. 17, we depict a quantum circuit for affine transformation of AES S-box which combined the above two operations.

$$\mathrm{P} := \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,0 \end{bmatrix}, \mathrm{L} := \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,0\,0\,0\,0\,0 \\ 1\,1\,1\,0\,0\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0\,0\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0 \\ 0\,1\,1\,1\,0\,0\,1\,0 \\ 0\,0\,1\,1\,1\,0\,0\,1 \end{bmatrix}, \mathrm{U} := \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 0\,1\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}.$$

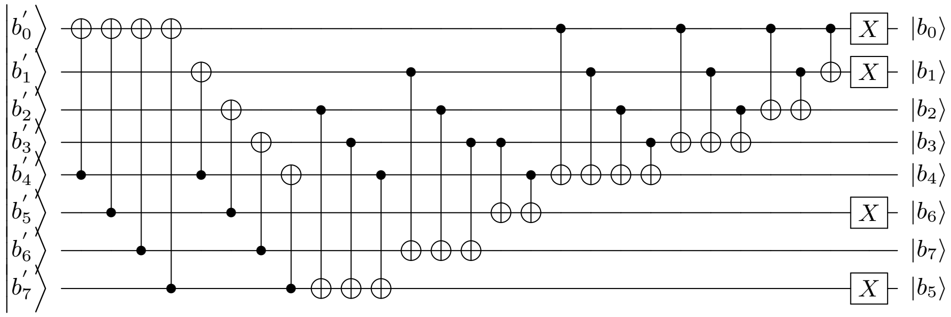


Fig. 17: Quantum circuit of the Affine transformation.

As shown in Fig.17, the quantum circuit of the affine transformation is executable in-place and does not include T-gate. Therefore, the affine transformation does not occur additional T-depth or qubits.

4.5 Merging inverse of isomorphic mapping and affine transformation

By combining the inverse of isomorphic mapping and the affine transformation, the number of CNOT gates of the quantum circuit can be reduced. The inverse of isomorphic mapping are represented as the matrix form, α^{-1}, from α in Section 3.1:

$$\alpha^{-1} = \begin{bmatrix} 1~0~1~0~0~0~0~0 \\ 1~1~0~1~1~1~1~0 \\ 0~0~0~0~1~1~0~0 \\ 0~1~1~1~0~0~0~0 \\ 0~0~0~1~0~1~0~0 \\ 1~0~0~1~0~0~0~0 \\ 1~0~0~1~0~1~1~0 \\ 0~1~1~1~0~0~0~1 \end{bmatrix}^{-1} = \begin{bmatrix} 1~1~1~1~1~0~1~0 \\ 1~0~0~1~0~1~0~0 \\ 0~1~1~1~1~0~1~0 \\ 1~1~1~1~1~1~1~0 \\ 1~1~0~1~0~1~1~0 \\ 1~1~1~1~0~1~1~0 \\ 1~1~1~1~0~0~0~0 \\ 0~0~0~1~0~0~0~1 \end{bmatrix}$$

then we can merge the inverse of isomorphic mapping and the affine transformation as follow:

$$M\{r\} = \begin{bmatrix} 1~0~0~0~1~1~1~1 \\ 1~1~0~0~0~1~1~1 \\ 1~1~1~0~0~0~1~1 \\ 1~1~1~1~0~0~0~1 \\ 1~1~1~1~1~0~0~0 \\ 0~1~1~1~1~1~0~0 \\ 0~0~1~1~1~1~1~0 \\ 0~0~0~1~1~1~1~1 \end{bmatrix} \begin{bmatrix} 1~1~1~1~1~0~1~0 \\ 1~0~0~1~0~1~0~0 \\ 0~1~1~1~1~0~1~0 \\ 1~1~1~1~1~1~1~0 \\ 1~1~0~1~0~1~1~0 \\ 1~1~1~1~0~1~1~0 \\ 1~1~1~1~0~0~0~0 \\ 0~0~0~1~0~0~0~1 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

where $\{r\} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7)$ is the result of the proposed multiplicative inversion on the isomorphic $\mathbf{GF}(2^8)$. In here,

$$\begin{bmatrix} 0~0~1~1~1~0~1~1 \\ 0~1~1~1~1~0~0~1 \\ 1~1~1~1~0~1~0~1 \\ 1~1~1~1~1~0~1~1 \\ 0~0~1~1~1~1~0~0 \\ 0~0~1~1~0~0~0~0 \\ 0~1~0~1~0~1~0~0 \\ 0~0~1~1~1~1~1~1 \end{bmatrix} = \begin{bmatrix} 1~0~0~0~1~1~1~1 \\ 1~1~0~0~0~1~1~1 \\ 1~1~1~0~0~0~1~1 \\ 1~1~1~1~0~0~0~1 \\ 1~1~1~1~1~0~0~0 \\ 0~1~1~1~1~1~0~0 \\ 0~0~1~1~1~1~1~0 \\ 0~0~0~1~1~1~1~1 \end{bmatrix} \begin{bmatrix} 1~1~1~1~1~0~1~0 \\ 1~0~0~1~0~1~0~0 \\ 0~1~1~1~1~0~1~0 \\ 1~1~1~1~1~1~1~0 \\ 1~1~0~1~0~1~1~0 \\ 1~1~1~1~0~1~1~0 \\ 1~1~1~1~0~0~0~0 \\ 0~0~0~1~0~0~0~1 \end{bmatrix}$$

and this matrix can be decomposed by matrix P, L, and U by PLU decomposition. The matrix P, L, and U are as follow:

$$\mathrm{P} := \begin{bmatrix} 0&0&1&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}, \mathrm{L} := \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&1&1&1&0&0&0&0\\ 0&0&1&0&1&0&0&0\\ 0&0&1&0&0&1&0&0\\ 1&0&0&0&1&1&1&0\\ 0&0&1&0&0&1&1&1 \end{bmatrix}, \mathrm{U} := \begin{bmatrix} 1&1&1&1&0&1&0&1\\ 0&1&1&1&1&0&0&1\\ 0&0&1&1&1&0&1&1\\ 0&0&0&1&0&1&1&0\\ 0&0&0&0&1&0&1&1\\ 0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}.$$

therefore, we can rearrange the $M\{r\}$ as

$$M\{r\} = \begin{bmatrix} 0&0&1&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&1&1&1&0&0&0&0\\ 0&0&1&0&1&0&0&0\\ 0&0&1&0&0&1&0&0\\ 1&0&0&0&1&1&1&0\\ 0&0&1&0&0&1&1&1 \end{bmatrix} \begin{bmatrix} 1&1&1&1&0&1&0&1\\ 0&1&1&1&1&0&0&1\\ 0&0&1&1&1&0&1&1\\ 0&0&0&1&0&1&1&0\\ 0&0&0&0&1&0&1&1\\ 0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} r_0\\ r_1\\ r_2\\ r_3\\ r_4\\ r_5\\ r_6\\ r_7 \end{bmatrix} \oplus \begin{bmatrix} 1\\ 1\\ 0\\ 0\\ 0\\ 1\\ 1\\ 0 \end{bmatrix}$$

and it can be implemented in quantum circuit as Fig.18. This quantum circuit reduces the number of CNOT gates from 45 to 29, compared to implementing the inverse of isomorphic mapping and affine transformation, separately.

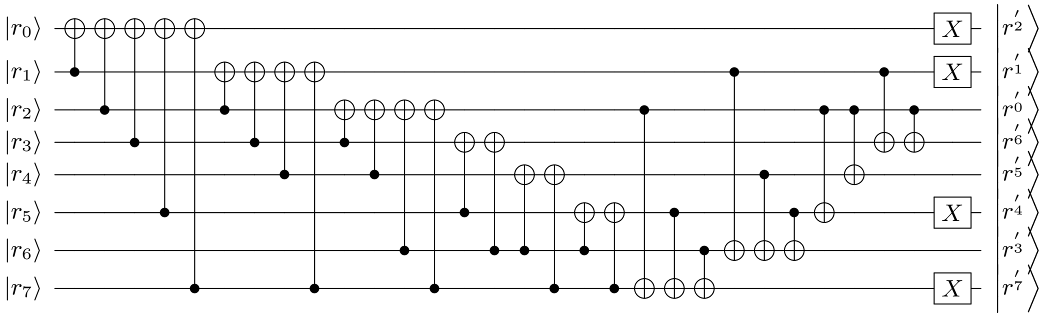


Fig. 18: Quantum circuit of merging the inverse of isomorphic mapping and affine transformation.

5 Evaluation

We evaluate our proposed method in terms of the number of qubits and T-depth. To design a quantum circuit that calculates multiplicative inversion in $\mathbf{GF}(2^8)$, we proposed quantum circuit designs with varying depth-width for smaller Galois field, such as $\mathbf{GF}(2^4)$ and $\mathbf{GF}(2^2)$. By selecting the quantum circuit for the smaller Galois field operation according to the situation, the depth-width of the quantum circuit that calculates the multiplicative inversion in $\mathbf{GF}(2^8)$ can be reduced in a trade-off manner.

5.1 Multiplication and dagger in $\mathbf{GF(2^2)}$

Symbol	# qubit	T-depth	Composition of gates
$X_{(2),T3}$	6	9	Toffoli × 3
$X_{(2),A2}$	7	5	AND × 2, Toffoli × 1
$X_{(2),A3}$	8	3	AND × 3, AND† × 1
$X^\dagger_{(2),A3}$	10	1	AND × 1, AND† × 3
$X^\dagger_{(2),A2}$	6	3	Toffoli × 1, AND† × 2

Table 4: The number of qubits and T-depth of multiplication and its dagger in $\mathbf{GF}(2^2)$.

As explained in Section 3, the multiplication operation quantum circuit in $\mathbf{GF}(2^2)$ varies in depth-width according to the combination of AND gates (and its dagger) and Toffoli gates. In the case of AND gate, compared to Toffoli gate, qubit consumption is increased, but T-Depth is reduced. The summary of multiplication operation in $\mathbf{GF}(2^2)$ used in our proposed scheme is depicted as Table 4

5.2 Multiplication in $\mathbf{GF(2^4)}$

Symbol	# qubit	T-depth	Composition of gates
$X_{(4),T3}$	18	10	$X_{(2),T3}\times$ 3, $X^\dagger_{(2),A3} \times 1$
$X_{(4),AA3}$	20	7	$X_{(2),A3}\times$ 2, $X_{(2),A3}\times$ 1, $X^\dagger_{(2),A3} \times 1$
$X_{(4),A2}$	21	6	$X_{(2),A2}\times$ 3, $X^\dagger_{(2),A3} \times 1$
$X_{(4),A3}$	24	4	$X_{(2),A3}\times$ 3, $X^\dagger_{(2),A3} \times 1$
$X^{(4),-1}$	18	6	$X_{(2),A3}\times$ 1, $X_{(2),A3} \times 2$

Table 5: The number of qubits and T-depth of multiplication and its dagger in $\mathbf{GF}(2^4)$.

The quantum circuits for multiplication in $\mathbf{GF}(2^4)$ have different depth-width. The different is depending on which quantum circuit for multiplication in $\mathbf{GF}(2^2)$ is used, and whether the quantum circuits are arranged in sequential or parallel. The summary of multiplication operation in $\mathbf{GF}(2^2)$ used in our proposed scheme is depicted as Table 5.

5.3 Multiplicative Inversion in $\mathbf{GF(2^8)}$

Scheme	type	# qubit	T-depth	# CNOT	# 1qCliff	# T	# M
Grassl et al.		44	217	8683	1028	3584	0
Langenberg et al.		32	120	314	4	385	0
Jaques et al.	[BP10]	41	35	818	264	164	41
	[BP12]	137	6	654	184	136	34
Our scheme	minimum width	32	36	958	366	268	40
	balanced 1	34	31	1000	408	232	46
	balanced 2	36	30	985	390	241	43
		42	27	1004	408	232	46
		46	25	986	372	250	40
		50	22	986	372	250	40
	minimum depth	54	20	1016	408	232	46

Table 6: Comparison of our methods with the existing ones.

The quantum circuit of multiplicative inversion in $\mathbf{GF}(2^8)$ has a more complex structure than the quantum circuit of $\mathbf{GF}(2^4)$ and $\mathbf{GF}(2^2)$. However, the dominant factor on its cost is which quantum circuits of $\mathbf{GF}(2^4)$ is used and how they are arranged. The Table 6 summarizes the number of qubits and T-depth of multiplicative inversion in $\mathbf{GF}(2^8)$ for each possible combination.

Acknowledgment

This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (⟨Q|Crypton⟩, No.2019-0-00033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity).

References

1. Matthew Amy, Dmitri Maslov, Mchele Mosca, and Martin Roetteler. 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, 2013.
2. Fowler Austion G., Stephens Ashley M., and Groszkowski Peter. High-threshold universal quantum computation on the surface code. *Phys. Rev. A*, 80(5):052312, 2009. Full version available at `https://journals.aps.org/pra/cited-by/10.1103/PhysRevA.80.052312`.

3. Joan Boyar and René Peralta. A New Combinational Logic Minimization Technique with Applications to Cryptology. In Paola Festa, editor, *Experimental Algorithms. SEA 2010*, volume 6049 of *LNCS*, pages 178–189. Springer, 2010.
4. Joan Boyar and René Peralta. A small depth-16 circuit for the AES S-Box. In Dimitris Gritzalis, Steven Furnell, and Marianthi Theoharidou, editors, *Information Security and Privacy Research. SEC 2012*, volume 376 of *IFIPAICT*, pages 287–298. Springer, 2012.
5. Amento-Adelmann Brittanney, Markus Grassl, Brandon Langenberg, Yi-Kai Liu, Eddie Schoute, and Rainer Steinwandt. Quantum Cryptanalysis of Block Ciphers: A Case Study. In *Poster at Quantum Information Processing QIP*, 2018.
6. Craig Gidney. Halving the cost of quantum addition. *Quantum*, 2:74, 2018.
7. Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. Applying Grover's algorithm to AES: quantum resource estimates. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - PQCrypto 2016*, volume 9606 of *LNCS*, pages 29–43. Springer, 2016.
8. Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, pages 212–219. ACM, 1996.
9. Toshiya Itoh and Shigeo Tsujii. A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. *Information and Computations*, 78(3):171–177, 1988.
10. Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing Grover oracles for quantum key search on AES and LowMC. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 (Proceedings, Part II)*, volume 12106 of *LNCS*, pages 280–310. Springer, 2020.
11. Junki Kang, Dooho Choi, Yong-Je Choi, and Dong-Guk Han. Secure Hardware Implementation of ARIA Based on Adaptive Random Masking Technique. *ETRI Journal*, 34(1), 2012.
12. Panjin Kim, Daewan Han, and Kyung Chul Jeong. Time-space complexity of quantum search algorithms in symmetric cryptanalysis: applying to AES and SHA-2. *Quantum Information Processing*, 17(12):339, 2018.
13. Brandon Langenberg, Hai Pham, and Rainer Steinwandt. Reducing the Cost of Implementing the Advanced Encryption Standard as a Quantum Circuit. *IEEE Transactions on Quantum Engineering*, 1:1–12, 2020.
14. Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A compact Rijndael Hardware Architecture with S-Box Optimization. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 239–254. Springer, 2001.
15. Peter Selinger. Quantum circuits of T-depth one. *Phys. Rev. A*, 87(4):042302, 2013. Full version available at `https://journals.aps.org/pra/abstract/10.1103/PhysRevA.87.042302`.
16. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 4(2):303–332, 1999.