Impact of Optimized Operations $A \cdot B$, $A \cdot C$ for Binary Field Inversion on Quantum Computers

Kyoungbae Jang, Seung Ju Choi Hyeokdong Kwon, and Zhi Hu Hwajeong Seo[⋆]

IT Department, Hansung University, Seoul, South Korea,
{starj1023, bookingstore3, korlethean}@gmail.com, huzhi_math@csu.edu.cn
hwajeong84@gmail.com

Abstract. The inversion circuit based on the Itoh-Tsujii algorithm, used for many cryptography functions, requires a number of multiplication and squaring operations in circuits. In the past, the optimized inversion implementation has been actively studied in modern computers. However, there are very few works to optimize the inversion on the quantum computer. In this paper, we present the optimized implementation of binary field inversion in quantum circuits. Reversible and non-reversible multiplication circuits are finely combined to reduce the number of CNOT gate. In particular, we optimized the reversible circuit for $A \cdot B$ and $A \cdot C$ case in the inversion operation. Afterward, the multiplication and squaring routine efficiently initializes some of the qubits used for the routine into zero value. Lastly, the-state-of-art multiplication and squaring implementation techniques, such as Karatsuba algorithm and shift-based squaring are utilized to obtain the optimal performance. In order to show the effectiveness of the proposed implementation, the inversion is applied to the substitute layer of AES block cipher. Furthermore, the proposed method can be applied to other cryptographic functions, such as binary field inversion for public key cryptography (i.e. Elliptic Curve Cryptography).

Keywords: Quantum Computers · Itoh-Tsujii Algorithm · Karatsuba Algorithm · Binary Field Multiplication.

1 Introduction

The binary field is a finite field of characteristic 2, which is a binomial polynomial consisting of an irreducible polynomials of n degrees. The binary field arithmetic is widely used in cryptographic applications. For the high performance of cryptography analysis, the optimized binary field arithmetic in the quantum circuit is a fundamental building block. Among binary field arithmetic operations, the most expensive operation is an inversion operation, which is a multiplicative computation of finding a^{-1} of element $a \in GF(2^m)$, such that $a \cdot a^{-1} = 1$. The inversion operation is mainly used in both symmetric and asymmetric cryptography, such as the substitute layer of AES and inversion of

[⋆] Corresponding Author

Elliptic Curve Cryptography (ECC) [1]. A number of optimization methods have been proposed for computing the multiplicative inverse [2–4]. One of the well known algorithm is the Itoh-Tsujii multiplicative inverse algorithm [5], which is a inverse algorithm based on Fermat's Little Theorem (FLT).

When computing the binary field inversion, multiplication and squaring operations are required. The multiplication in binary field involves multiplying two polynomial multiplication and a modular reduction with an irreducible polynomial. The reduction operation is a relatively simpler operation than the polynomial multiplication, because the reduction consists of only eXclusive-or operations [6]. For this reason, an optimized polynomial multiplication for binary fields has been studied [7–9]. Among them, Karatsuba algorithm is widely used in practice. Karatsuba algorithm replaces the one n-bit multiplication operation into three $\frac{n}{2}$-bit multiplication operations. Although, Karatsuba multiplication requires several extra addition operations (i.e. eXclusive-or), the method significantly reduces the complexity of multiplication. By applying the Karatsuba algorithm to quantum computing, the computation complexity in quantum circuits is also optimized.

By using the Karatsuba multiplication for the Itoh-Tsujii inversion algorithm, the multiplication part of the inversion is optimized [10]. During the calculation, there is a multiplication routine that proceeds $A \cdot B$ and $A \cdot C$ in this pattern, where A, B, and C are operands for each operation. The value of A should be maintained during the computation since it gets reused in both $A \cdot B$ and $A \cdot C$ multiplications. Due to the nature of Karatsuba multiplication, operand A gets changed to other value after $A \cdot B$. For this reason, the direct reuse of operand A in following $A \cdot C$ is not available. Therefore, a reverse circuit must be added to revert the A value to its original value after $A \cdot B$ operation.

In this paper, we present a state-of-art multiplication based on [11] and squaring method to implement the Itoh-Tsujii algorithm. [11] minimizes the number of multiplications by recursively applying Karatsuba multiplication on quantum computers. However, only $A \cdot B$ multiplication is considered. In order to apply [11] in $A \cdot B$ and $A \cdot C$ multiplication, a additional reversible circuit is required to calculate the $A \cdot B$ and $A \cdot C$ multiplication. We have successfully enhanced the [11] multiplication. This can be applied to $A \cdot B$ and $A \cdot C$ multiplication in Karatsuba approach. $A \cdot B$ and $A \cdot C$ structures are optimized by omitting the reverse circuit. Furthermore, we proposed qubit re-use techniques for squaring and multiplication routine. By initializing qubits, the circuit is implemented with minimal circuit gates. Finally, the proposed technique is applied to the substitute layer of AES in order to show practicality and efficiency. The algorithm can be applied to other cryptographic functions, such as binary field inversion of Elliptic Curve Cryptography (ECC).

1.1 Research Contributions

- **Optimized implementation of binary field inversion** The quantum circuit for multiplicative inversion based on Itoh-Tsujii algorithm is optimized by utilizing the $A \cdot B$ and $A \cdot C$ pattern. By changing the reversible structure of the algorithm into non-reversible structure, the number of CNOT gate is optimized. Furthermore, the qubit of B is initialized with optimal routine. The initialized qubit gets used in the following operation, which reduces the total number of qubits required for the inversion operation. The binary field polynomial multiplication is optimized with the-state-of-art Karatsuba multiplication and shift-based squaring method, which reduced the total number of Toffoli and CNOT gates.

- **Optimized cryptographic primitives for AES and ECC** The proposed method can be applied to the substitute layer of AES and binary field inversion of ECC. We show the impact of proposed method by implementing the substitute layer. The proposed circuit significantly reduces the required resource in terms of CNOT gates and qubits.

1.2 Organization of the paper

The organization of this paper is as follows. Section 2 presents the background of binary field inversion. In Section 3, the proposed inversion operation is presented. In Section 4, we evaluate the proposed inversion method for AES. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Itoh-Tsujii Multiplicative Inverse Algorithm

Itoh-Tsujii multiplicative inverse algorithm is an exponentiation based algorithm for the inversion in binary field [5]. In a normal basis representation, it reduces the complexity of computing the inverse of a non-zero element in $GF(2^n)$ with the binary exponentiation method.

2.2 Karatsuba Multiplication

Karatsuba algorithm reduces the complexity of multiplication by additional addition operations. When multiplying the polynomial f and g of size n through $h = f \cdot g$, two input polynomials are divided into $s = n/2$ units as follows:

$$\begin{aligned} f &= f_1 x^s + f_0 \\ g &= g_1 x^s + g_0 \end{aligned} \tag{1}$$

After splitting two input polynomials, Karatsuba multiplication can be performed as follows:

$$f_0 \cdot g_0 + \{(f_0 + f_1) \cdot (g_0 + g_1) + f_0 \cdot g_0 + f_1 \cdot g_1\} x^s + f_1 \cdot g_1 x^{2s} \tag{2}$$

With the Karatsuba multiplication, multiplication of $O(n^2)$ can be performed in $O(n^{\log_2 3})$ with a few addition operations.

2.3 Quantum Gates

Quantum computers have several gates that can represent the classical gates [12]. Two most representative gates are CNOT and Toffoli gates. The CNOT gate performs a NOT gate operation on the second qubit when the first input qubit of the two input qubits is one. This gate performs the same role as the add operation on the binary field. The circuit configuration is shown in left side of Figure 1. The Toffoli gate receives three qubits. When the first and second qubits are one, the gate performs a NOT gate operation on the last qubit. This serves as an AND operation on the binary field, and the circuit configuration is shown in right side of Figure 1.

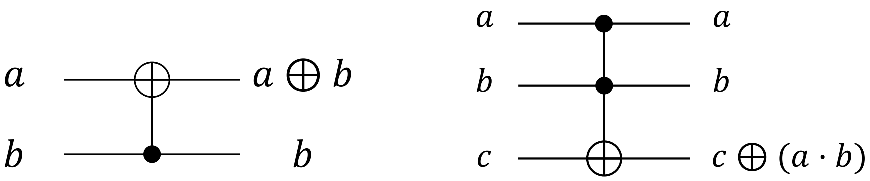

Fig. 1. Circuit configuration of the (left) CNOT and (right) Toffoli gate.

3 Proposed Method

In this paper, the Itoh-Tsujii algorithm for binary field inversion was optimized on the quantum computer. First, the multiplication of Itoh-Tsujii algorithm was optimized by applying the Karatsuba multiplication technique. Second, the quantum circuit is optimized by changing the reversible circuit to a non-reversible circuit when calculating $A \cdot B$ and $A \cdot C$ pattern in the Itoh-Tsujii algorithm with Karatsuba algorithm. Lastly, some of qubits are reuse during squaring and multiplication operations by using the initialization technique. Initialized qubits are used for following computations. With this technique, the total number of qubits required for the operation is optimized.

3.1 Optimization of $A \cdot B$, $A \cdot C$ structure in inversion

The proposed method optimized the $A \cdot B$ and $A \cdot C$ structure in inversion algorithm by using non-reversible circuits rather than reversible circuits. Itoh-Tsuji algorithm consists of squaring and multiplication operation. The squaring operation is designed with few CNOT gates on a quantum circuit [13]. Unlike the squaring operation, the multiplication operation is an expensive operation. Therefore, it is necessary to apply an efficient multiplication technique to achieve the optimal performance. There are a number of studies to implement the multiplication. Among them, one of the well known efficient multiplication method

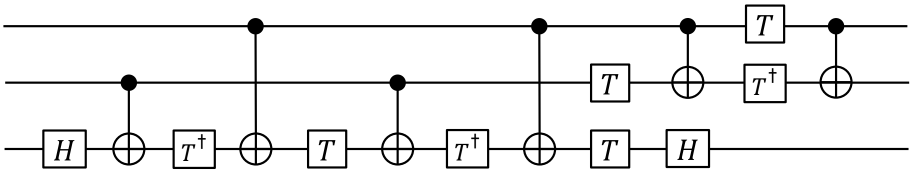

Fig. 2. Circuit configuration of the CNOT gate.

Algorithm 1 Itoh-Tsuji-based inversion for $p = x^8 + x^4 + x^3 + x + 1$

```
Require: Integer $z$ satisfying $1 \leq z \leq p - 1$.
Ensure: Inverse $t = z^{p-2} \bmod p = z^{-1} \bmod p$.
1: $z_2 \leftarrow z^2 \cdot z$                                   {cost: 1S+1M}
2: $z_3 \leftarrow z_2^2 \cdot z$                                 {cost: 1S+1M}
3: $z_6 \leftarrow z_3^{2^3} \cdot z_3$                           {cost: 3S+1M}
4: $z_7 \leftarrow z_6^2 \cdot z$                                 {cost: 1S+1M}
5: $t \leftarrow z_7^2$                                           {cost: 1S}
6: return $t$
```

is the Karatsuba multiplication. When performing the two polynomial $f \cdot g$ multiplication on a quantum gate, the Karatsuba algorithm can reduce the usage of Toffoli gates while increasing the number of CNOT gates. The Toffoli gate consists of 6 CNOT gates and 9 single qubit gates as shown in Figure 2. For this reason, it is important to design a circuit with a minimum number of Toffoli gates to implement the quantum algorithm.

In order to show the impact of proposed method, the substitute layer of AES is selected as an example, which uses $x^8 + x^4 + x^3 + x + 1$ as a target polynomial. The substitute layer requires binary field inversion operation of $x^8 + x^4 + x^3 + x + 1$. The inversion operation of AES is given in Algorithm 1.

In Step 1, the squaring operation is performed. The squaring operation can be obtained with 11 CNOT gates as shown in Figure 3. The squaring directly reduces the result due to the nature of squaring on binary field [13]. After the squaring operation, a multiplication operation is performed on the squared value (i.e. z^2) and the input integer z. The integer z is reused in Step 2. This process follows $A \cdot B$, $A \cdot C$ structure (i.e $z = A, z_2 = B, z_2^2 = C$).

When the Karatsuba method is applied to the multiplication, the input value (i.e. z) is updated due to the operand addition step of Karatsuba. Generally, the operand is restored to the original z value for the following operation. The condition is described with an example on 2-bit case in Figure 4. The circuit describes the condition after the 2-bit multiplication.

When the 2-bit multiplication operations on operands $A(a_0, a_1)$ and $B(b_0, b_1)$ are performed, a_1 and b_1 are changed to $a_0 + a_1$ and $b_0 + b_1$, respectively. If only the result of the multiplication is needed, reversible gates are not required. However, the value of $A(a_0, a_1)$ is used again in following calculations, (i.e. $A \cdot C$).

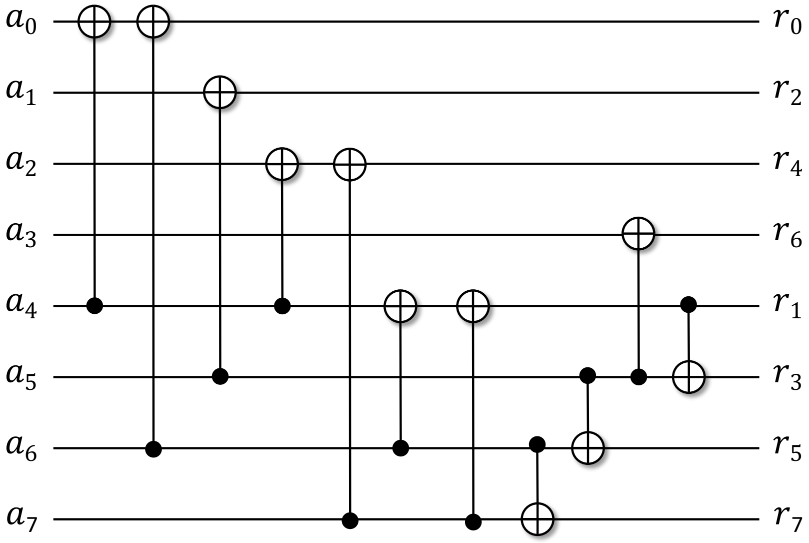

Fig. 3. Circuit configuration of the squaring operation on $x^8 + x^4 + x^3 + x + 1$.

For this reason, the reversible circuit should be performed for the operand A. The reversible gate is described in Figure 5. In each n-bit multiplication, the reversible gate requires $(n - 1)$ CNOT gates for each operand.

In order to reduce this overhead, we present a non-reversible gate based $A \cdot B$ and $A \cdot C$ structure of inversion. Detailed descriptions for 2-bit case are given in Figure 6.

First, the $a_0 \cdot c_0$ operation is calculated. When the operation of $a_0 \cdot c_0$ is completed, the a_0 value is no longer used in following operations, because the value will not affect the following calculation. The second qubit, $a_0 + a_1$, then performs a CNOT operation with a_0 to change the value of the first qubit to a_1. Afterward, $a1 \cdot c1$ operation gets calculated using c_1 with the a_1 value. The $(c_0 + c_1)$ is simply made through CNOT operation between c_0 and c_1. Finally, the $(a_0 + a_1) \cdot (c_0 + c_1)$ gets calculated.

In conclusion, it is possible to make the same result of $A \cdot B$ and $A \cdot C$, through $A \cdot B$ and $A' \cdot C$ structure, which excludes reversible circuits.

By using the proposed method, the inversion for $x^8 + x^4 + x^3 + x + 1$ is implemented. This implementation reduced the number of Toffoli gates used compared to the basic multiplication method. Generic multiplication uses 64 (n^2) Toffoli gates per multiplication, while Karatsuba only uses 27 Toffoli gates per multiplication. However, the Karatsuba method for the inversion operation uses 108 additional CNOT gates than the generic multiplication method. However the number of Toffoli gates should be considered than CNOT gates since the Toffoli gate consists of 6 CNOT gates with 9 one-gates.

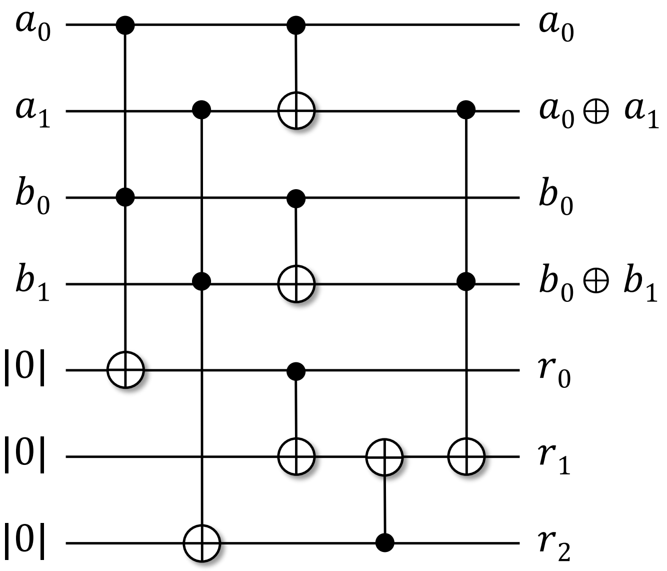

Fig. 4. Non-reversible multiplication of $A \cdot B$ with Karatsuba multiplication.

The Karachuba multiplication circuit of [11] is not designed to be reversible, and all of the remaining operators A, B and input qubits get treated as garbage qubits after multiplication, except for the qubits that contain the result of the operation. This is because the calculation of $A \cdot C$ was not considered which comes after $A \cdot B$ during multiplication of Itoh-Tsuji algorithm. Conventionally, in order to perform Itoh-Tsuji multiplication with Karatsuba, a reversible circuit is required to recover the original values of the operand. In the case of inversion for $x^8+x^4+x^3+x+1$, 14 CNOT gates are required for the reverse circuit for the Karatsuba algorithm. In the proposed method, the reverse process is omitted by utilizing the $A \cdot B$ and $A' \cdot C$ pattern, which optimizes 14 CNOT gates.

3.2 Reducing the number of qubits

In this section, we present the technique to reduce the total number of qubits required for the operation during the multiplication and squaring operation of Itoh-Tsujii's algorithm.

In Steps 1 and 2 of Algorithm 1, the value of B is the square of the A value (i.e. z on the algorithm) and the value C is created through the square of $A \cdot B$. Both B and C are originated from the value A.

The value of B is computed with the squaring operation as described in Figure 3. The squaring operation is performing the left-shifting and modular operations

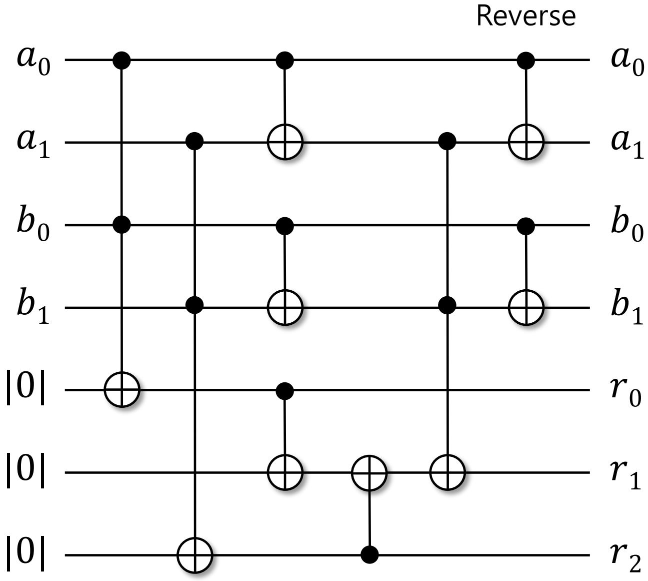

Fig. 5. Reversible multiplication of $A \cdot B$ with Karatsuba multiplication.

on the input qubit. In other words, the value B is the result of left-shifting the qubit of A and the modular operation on it.

Using these features, some of the qubits are initialized to zero, which allows the calculation to use less qubits than original multiplication computation. Detailed descriptions of the method are presented in Figure 7.

The first Step represents the value B, the second Step represents the value A, and the third Step represents the value C from $A \cdot B$, $A \cdot C$ multiplication of Itoh-Tsujii algorithm.

First, B value is formed from A value through the square operation. In Step 2, $A \cdot B$ multiplication is performed. After the multiplication, the values of B and A get changed into B' and A', respectively. The result of $A \cdot B$ calculation gets stored in the third row. A gets changed into A', because the calculation does not require the reverse circuit. In Step 3, $A \cdot B$ operation is calculated, and C value is formed with the result. In Step 4, values A' and C are multiplied. During the multiplication process of Step 4, the value of A' gets changed in order to proceed with the Karatsuba operation (i.e $A' \longrightarrow A''$). During this process, the changed value of multiplication (A'') forms the same value with B' during the multiplication process of $A' \cdot C$. By performing the CNOT operation on the same value between A'' and B', some of the qubits of B' is initialized back to zero (See Step 5). In Algorithm 1, the value B is not used again after the $A \cdot B$ operation.

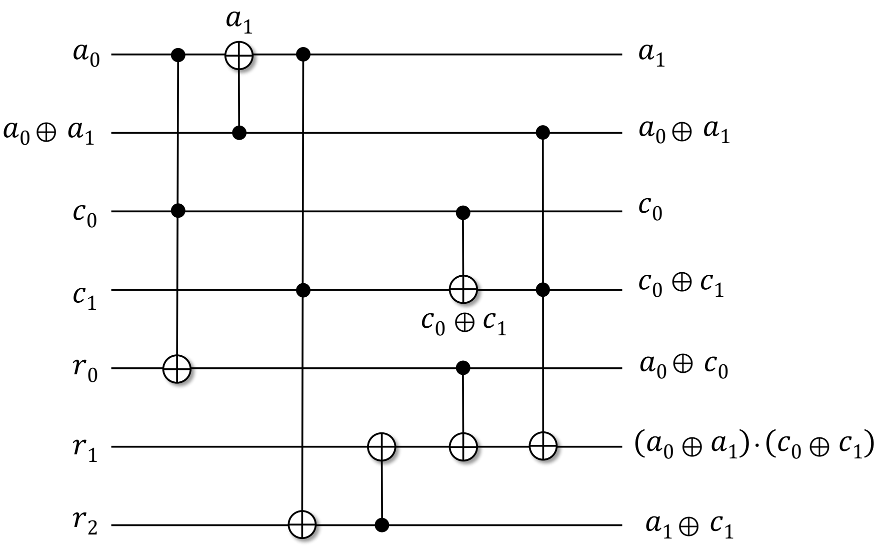

Fig. 6. Proposed non-reversible multiplication of $A \cdot C$ with Karatsuba multiplication.

For this reason, we can utilize these qubits for other purposes. Initialized qubits of B are used as a extra qubit space during the following inversion operation.

Table 1 presents the values contained in the value B' after $A \cdot B$ multiplied by B and a formed through the squaring of A on $x^8 + x^4 + x^3 + x + 1$.

Table 1. Combination of A values of B' after $A \cdot B$ computation on $GF(2^8)$.

k	B_k	k	B_k
0	$a_0 + a_2 + a_6 + a_7$	4	$a_2 + a_3 + a_4 + a_5 + a_7$
1	$a_4 + a_5 + a_7$	5	$a_5 + a_7$
2	$a_1 + a_3$	6	$a_3 + a_5 + a_6 + a_7$
3	$a_4 + a_5$	7	$a_6 + a_7$

As shown in Table 2, it can be seen that the value of B' after the operation of $A \cdot B$ consists of the addition of the A values. These combinations of A values are made through the Karatsuba operation. This combination is also observed in A'', which are values that get formed during $A' \cdot C$ multiplication. Values are shown in Table 2.

Using both Table 1 and Table 2, the combination can initialize the value of B to zero. For example, in Table 1, the element of k_0 is presented as $a_0+a_6+a_2+a_7$. This value is initialized by using k_6 and k_{14} value from Table 2 which are $a_0 + a_2$ in red and $a_6 + a_7$ in orange, respectively. By performing the CNOT operation on B_0 with k_6 and k_{14}, B_0 is initialized into zero.

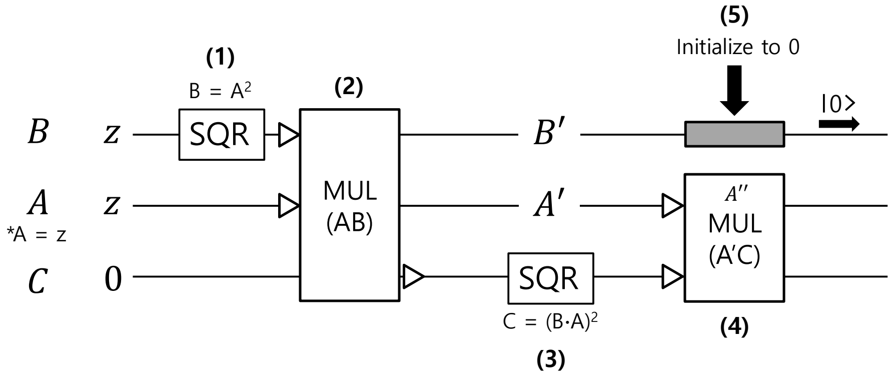

Fig. 7. Overview of proposed method for initializing some of the qubits during Karatsuba multiplication.

If the initialization is performed for multiplication with the conventional multiplication method, it would require 18 additional CNOT gates in order to form elements of k_n. However, the Karatsuba multiplication makes it possible to initialize the value of B to zero without having to take extra steps to form elements of k_n. During the multiplication process of Karatsuba, the elements of k_n gets created eventually in order to be used as a multiplication factor. By utilizing this feature, we were able to initialize 8 qubits with 11 CNOT gates.

In the process of calculating the Step 3 of the Algorithm 1, the qubit can be reduced in a similar way. Unlike Steps 1 and 2 of the algorithm 1, the Step 3 does not have a exact same structure of $A \cdot B$ and $A \cdot C$. However there is still a part that can initialize the qubit to zero. 8 additional qubits can be initialized to zero. Finally, 16 qubits can be initialized to zero and be used for the following calculation.

4 Evaluation

In order to evaluate quantum gates, we utilized the quantum computer emulator. We utilized the well-known IBM's ProjectQ framework for evaluation of the proposed method[1]. The framework provides quantum computer compiler and quantum resource estimator. This is useful for accurate evaluation. Proposed quantum gates are written in Python and follow the ProjectQ grammar.

We compare the number of Toffoli gates, CNOT gates, and qubit of implementation method based on $x^8+x^4+x^3+x+1$ inversion, which is used in the substitute layer of AES. The inversion operation of $x^8+x^4+x^3+x+1$ consists of 4 multiplications and 7 squaring operations using the Itoh-Tsuji's algorithm. We present two methods including CNOT reduction version and qubit recycle

[1] https://github.com/ProjectQ-Framework/ProjectQ

Table 2. Combination of A values of A'' during $A \cdot C$ computation on $GF(2^8)$.

k	A_k	R_k
0	a_0	a_0c_0
1	a_1	a_1c_1
2	a_0+a_1	$(a_0+a_1)(c_0+c_1)$
3	a_2	a_2c_2
4	a_3	a_3c_3
5	a_2+a_3	$(a_2+a_3)(c_2+c_3)$
6	a_0+a_2	$(a_0+a_2)(c_0+c_2)$
7	a_1+a_3	$(a_1+a_3)(c_1+c_3)$
8	$a_0+a_1+a_2+a_3$	$(a_0+a_1+a_2+a_3)(c_0+c_1+c_2+c_3)$
9	a_4	a_4c_4
10	a_5	a_5c_5
11	a_4+a_5	$(a_4+a_5)(c_4+c_5)$
12	a_6	a_6c_6
13	a_7	a_7c_7
14	a_6+a_7	$(a_6+a_7)(c_6+c_7)$
15	a_4+a_6	$(a_4+a_6)(c_4+c_6)$
16	a_5+a_7	$(a_5+a_7)(c_5+c_7)$
17	$a_4+a_5+a_6+a_7$	$(a_4+a_5+a_6+a_7)(c_4+c_5+c_6+c_7)$
18	a_0+a_4	$(a_0+a_4)(c_0+c_4)$
19	a_1+a_5	$(a_1+a_5)(c_1+c_5)$
20	$a_0+a_1+a_4+a_5$	$(a_0+a_1+a_4+a_5)(c_0+c_1+c_4+c_5)$
21	a_2+a_6	$(a_2+a_6)(c_2+c_6)$
22	a_3+a_7	$(a_3+a_7)(c_3+c_7)$
23	$a_2+a_3+a_6+a_7$	$(a_2+a_3+a_6+a_7)(c_2+c_3+c_6+c_7)$
24	$a_0+a_2+a_4+a_6$	$(a_0+a_2+a_4+a_6)(c_0+c_2+c_4+c_6)$
25	$a_1+a_3+a_5+a_7$	$(a_1+a_3+a_5+a_7)(c_1+c_3+c_5+c_7)$
26	$a_0+a_1+a_2+a_3+a_4+a_5+a_6+a_7$	$(a_0+a_1+a_2+a_3+a_4+a_5+a_6+a_7)$ $(c_0+c_1+c_2+c_3+c_4+c_5+c_6+c_7)$

version. The CNOT reduction version performs $A \cdot B$ and $A \cdot C$ pattern with non-reversible multiplications. For the qubit recycle version, few CNOT gates are used more than CNOT reduction version but some qubits are recycled. The required quantum resources of proposed method is given in the Table 3. Compared with Karatsuba approach by [11], the number of Toffoli gates is identical. For the CNOT gate, CNOT reduction and qubit recycle versions reduces 14 and 3 CNOT gates, respectively. In particular, the qubit recycle version reduces the number of qubit by 8.

Table 3. Comparison of quantum resource for $A \cdot B$ and $A \cdot C$ computation on $GF(2^8)$.

Method	Toffoli gate	CNOT gate	Qubit
Kepley et al. [11]	54	252	70
This work (CNOT reduction)	54	**238**	70
This work (qubit recycle)	54	**249**	**62**

The proposed method can be applied to the Itoh-Tsuji-based inversion of binary field ECC. In Algorithm 2, the inversion algorithm for `sect283k1` and `sect283r1` is given. In Step 1, 5, 7, and 10, $A \cdot B$ and $A \cdot C$ pattern is observed. This case can be optimized by using the proposed method in terms of CNOT gates and qubits.

```
Algorithm 2 Itoh-Tsuji-based inversion for $p = x^{283} + x^{12} + x^7 + x^5 + 1$
Require: Integer $z$ satisfying $1 \leq z \leq p - 1$.
Ensure: Inverse $t = z^{p-2} \bmod p = z^{-1} \bmod p$.
$z_2 \leftarrow z^2 \cdot z$                                  {cost: 1S+1M}
$z_4 \leftarrow z_2^{2^2} \cdot z_2$                          {cost: 2S+1M}
$z_8 \leftarrow z_4^{2^4} \cdot z_4$                          {cost: 4S+1M}
$z_{16} \leftarrow z_8^{2^8} \cdot z_8$                       {cost: 8S+1M}
$z_{17} \leftarrow z_{16}^2 \cdot z$                          {cost: 1S+1M}
$z_{34} \leftarrow z_{17}^{2^{17}} \cdot z_{17}$              {cost: 17S+1M}
$z_{35} \leftarrow z_{34}^2 \cdot z$                          {cost: 1S+1M}
$z_{70} \leftarrow z_{35}^{2^{35}} \cdot z_{35}$              {cost: 35S+1M}
$z_{140} \leftarrow z_{70}^{2^{70}} \cdot z_{70}$             {cost: 70S+1M}
$z_{141} \leftarrow z_{140}^2 \cdot z$                        {cost: 1S+1M}
$z_{282} \leftarrow z_{141}^{2^{141}} \cdot z_{141}$          {cost: 141S+1M}
$t \leftarrow z_{282}^2$                                      {cost: 1S}
return $t$
```

5 Conclusion

In this paper, we presented the optimized implementation of binary field inversion in quantum circuits for $A \cdot B$ and $A \cdot C$ structure. First, non-reversible circuits are used for $A \cdot B$ and $A \cdot C$ patterns. Second, qubit reuse technique is suggested. Both techniques reduce the required number of CNOT gates and qubits. The-state-of-art optimization techniques, such as Karatsuba algorithm and modular squaring, are also utilized to reduce the number of Toffoli gates and qubits. Finally, the quantum circuit for binary field inversion achieved the optimal number of Toffoli gates, CNOT gates and qubits. The proposed method is used to implement the substitute layer of AES. The result shows that proposed method uses lesser CNOT and qubits than previous Karatsuba based approach. Furthermore, the proposed method can be used for the binary field inversion of ECC.

The future work is going to find another arithmetic structures to optimize the quantum circuit for other cryptographic algorithms. In the inversion, the consecutive multiplication and squaring structure is frequently used. We will find the optimal computation routine for this structure.

6 Acknowledgement

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). This work of Zhi Hu is supported by the Natural Science Foundation of China (Grants No. 61972420, 61602526) and Hunan Provincial Natural Science Foundation of China (Grants No. 2019JJ50827, 2020JJ3050).

References

1. D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.
2. A. A.-A. Gutub, A. F. Tenca, E. Savaş, and R. K. Koç, "Scalable and unified hardware to compute Montgomery inverse in GF(p) and GF(2^n)," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 484–499, Springer, 2002.
3. S. Yen, "Improved normal basis inversion in GF(2^m)," *IEE Electronic Letters*, vol. 33, pp. 196–197, 1997.
4. M. A. Hasan, "Efficient computation of multiplicative inverses for cryptographic applications," *15th IEEE Symposium on Computer Arithmetic*, 2001.
5. T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in GF(2^m) using normal basis," *Information and Computing*, vol. 78, pp. 171–177, 1988.
6. N. H. B. Michael Kaminski, "Multiplicative complexity of polynomial multiplication over finite fields," *Journal of the ACM*, vol. 36, no. 3, pp. 150–170, 1989.
7. O. Karatsuba, "Multiplication of multidigit numbers on automata," in *Soviet physics doklady*, pp. 595–596, Springer, 1963.
8. D. Bernstein, "Batch binary edwards," in *Advances in Cryptology - CRYPTO 2009*, pp. 317–336, Springer, 2009.
9. S. Cook, "On the minimum computation time of functions. phd thesis, harvard university," 1966.
10. P. Schwabe and B. Westerbaan, "Solving binary $\mathcal{MQ}$ with grover's algorithm," pp. 303–322, 12 2016.
11. S. Kepley and R. Steinwandt, "Quantum circuits for F_2^n-multiplication with subquadratic gate count," *Quantum Information Processing*, vol. 14, no. 7, pp. 2373–2386, 2015.
12. X. Li, Y. Wu, D. Steel, D. Gammon, T. H. Stievater, D. S. Katzer, D. Park, C. Piermarocchi, and L. J. Sham, "An all-optical quantum gate in a semiconductor quantum dot," *Science*, vol. 301, no. 5634, pp. 809–811, 2003.
13. E. Muñoz-Coreas and H. Thapliyal, "Design of quantum circuits for Galois field squaring and exponentiation," in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 68–73, 2017.