Constructing Quantum Implementations with the Minimal T-depth or Minimal Width and Their Applications

Zhenyu Huang, Fuxin Zhang, Dongdai Lin

Institute of Information Engeering, Chinease Academy of Sciences
University of Chinese Academy of Sciences

Eurocrypt 2025

Outline

Outline

Motivation

- Quantum Cryptanlysis for Symmetric Ciphers
  - Grover's algorithm: the attacker needs to construct a Grover oracle to search the key.
  - Simon's algorithm (Kuwakado and Mori, ISIT 2010; Kaplan et al. Crypto 2016): The attacker needs to access an online quantum encryption oracle.
  - Offline Simon's algorithm (Bonnetain et al. Asiacrypt 2019): the attacker needs to construct different quantum encryption oracles for different keys.
  - **The quantum circuit that implements the cipher** is a primary component of the Grover oracle or the quantum encryption oracle.
- NIST's call for proposals for PQC
  - **The complexity of quantum key search circuit for AES** is used as a baseline to categorize the post-quantum public-key schemes.

Motivation

- Quantum Cryptanlysis for Symmetric Ciphers
  - Grover's algorithm: the attacker needs to construct a Grover oracle to search the key.
  - Simon's algorithm (Kuwakado and Mori, ISIT 2010; Kaplan et al. Crypto 2016): The attacker needs to access an online quantum encryption oracle.
  - Offline Simon's algorithm (Bonnetain et al. Asiacrypt 2019): the attacker needs to construct different quantum encryption oracles for different keys.
  - **The quantum circuit that implements the cipher** is a primary component of the Grover oracle or the quantum encryption oracle.
- NIST's call for proposals for PQC
  - **The complexity of quantum key search circuit for AES** is used as a baseline to categorize the post-quantum public-key schemes.

Circuit Complexity

Cost Metrics:

- **Width** (the number of qubits); Gate count, especially the T-count;
- Depth: The number of layers of the circuit (gates acting on disjoint sets of qubits can be applied in parallel)
- T-**depth**: the number of layers for T gates, which dominates the running time of a circuit in fault-tolerant quantum computation (Surface code).

Previous Works (PQCrypto 2016, Asiacrypt 2020, Eurocrypt 2020, Aisacrypt 2022, Asiacrypt 2023, Asiacrypt 2024, etc):

- AES Width: $2953 \rightarrow 512 \rightarrow 374$, T-depth: $120 \rightarrow 80 \rightarrow 60$
- Heuristic improvements that are only applicable for AES.

Both T-depth and Width have theoretical minimums.

Generic methods for constructing circuits that achieve these minimums?

Circuit Complexity

Cost Metrics:

- **Width** (the number of qubits); Gate count, especially the T-count;
- Depth: The number of layers of the circuit (gates acting on disjoint sets of qubits can be applied in parallel)
- T-**depth**: the number of layers for T gates, which dominates the running time of a circuit in fault-tolerant quantum computation (Surface code).

Previous Works (PQCrypto 2016, Asiacrypt 2020, Eurocrypt 2020, Aisacrypt 2022, Asiacrypt 2023, Asiacrypt 2024, etc):

- AES Width: $2953 \rightarrow 512 \rightarrow 374$, T-depth: $120 \rightarrow 80 \rightarrow 60$
- Heuristic improvements that are only applicable for AES.

Both T-depth and Width have theoretical minimums.

Generic methods for constructing circuits that achieve these minimums?

Circuit Complexity

Cost Metrics:

- **Width** (the number of qubits); Gate count, especially the T-count;
- Depth: The number of layers of the circuit (gates acting on disjoint sets of qubits can be applied in parallel)
- T-**depth**: the number of layers for T gates, which dominates the running time of a circuit in fault-tolerant quantum computation (Surface code).

Previous Works (PQCrypto 2016, Asiacrypt 2020, Eurocrypt 2020, Aisacrypt 2022, Asiacrypt 2023, Asiacrypt 2024, etc):

- AES Width: $2953 \rightarrow 512 \rightarrow 374$, T-depth: $120 \rightarrow 80 \rightarrow 60$
- Heuristic improvements that are only applicable for AES.

Both T-depth and Width have theoretical minimums.

Generic methods for constructing circuits that achieve these minimums?

Convert Classical Circuits to Quantum Circuits

Boolean Function Implementation Problem

Given a vectorial Boolean function $F(x)$, construct a quantum circuit that maps

- $|x, 0\rangle \rightarrow |F(x), 0\rangle$: An **in-place** implementation of an invertible F.
- $|x, y\rangle \rightarrow |x, y \oplus F(x)\rangle$ for any y (sometimes only for $y = 0$): An **out-of-place** implementation of F.

Convert Classical Circuits to Quantum Circuits

Boolean Function Implementation Problem

Given a vectorial Boolean function $F(x)$, construct a quantum circuit that maps

- $|x, 0\rangle \rightarrow |F(x), 0\rangle$: An **in-place** implementation of an invertible F.
- $|x, y\rangle \rightarrow |x, y \oplus F(x)\rangle$ for any y (sometimes only for $y = 0$): An **out-of-place** implementation of F.

From classical circuits to quantum circuits:

- Classicial gates: NOT, XOR, AND $\Rightarrow$ NCT gate set: NOT(Pauli-X), CNOT, Toffoli

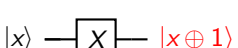


Figure: NOT gate

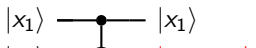


Figure: CNOT gate

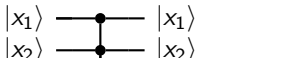
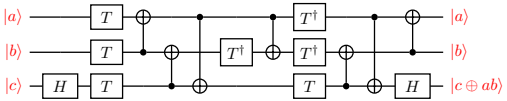


Figure: Toffoli gate

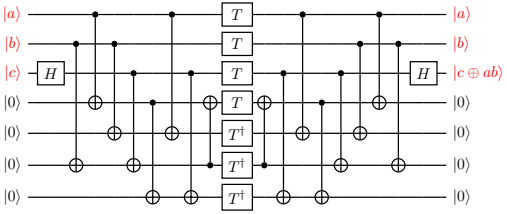
$\Rightarrow$ Clifford+T gates: {Pauli gates, CNOT, S, H} + T

Clifford+T Implementations for the Toffoli gate

- No ancilla qubit:



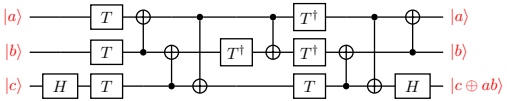
- T-depth-1:



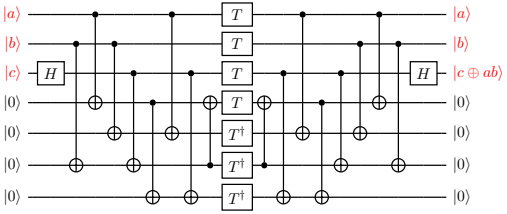
Design NCT circuits that achieve the minimal Toffoli-depth or minimal width

Clifford+T Implementations for the Toffoli gate

- No ancilla qubit:



- T-depth-1:

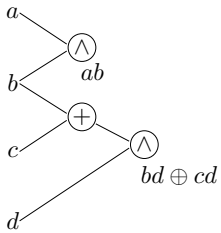


Design NCT circuits that achieve the minimal Toffoli-depth or minimal width

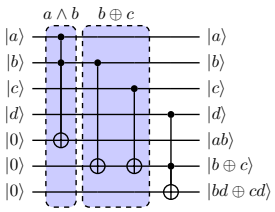
Outline

Convert Classical Circuits to NCT Circuits

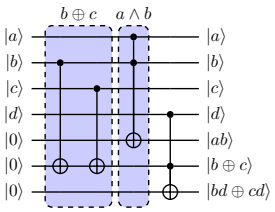
- Problems of the trivial conversion (XOR→CNOT, AND→Toffoli):
  - Need lots of ancilla qubits.
  - The depth may change, especially **Toffoli-depth≠ AND-depth** in sometimes.



(1) AND-depth-1



(2) Toffoli-depth-2



(3) Toffoli-depth-1

Figure: Quantum implementations of a classical circuit with AND-depth 1

Quantum Circuit with the Minimal Toffoli-depth

Theorem (Asiacrypt 2022, Huang & Sun)

Given a classical circuit with **AND-depth s**, the Toffoli-depth of the quantum circuit implementing all the nodes of the classical circuit **is not smaller than s**. Moreover, **with sufficiently many ancillae**, we can construct a quantum circuit implementing all the nodes of the classical circuit with **Toffoli-depth s**.

1. Construct a classical circuit with the minimal AND-depth ($= \lceil \log_2(D) \rceil$);
   - D is the algebraic degree of the vectorial Boolean function F (With an AND-layer, one can mostly double the degree).
   - AES S-box (degree 7): the minimal AND-depth is 3.
2. Convert it to a Toffoli-depth-$\lceil \log_2(D) \rceil$ NCT circuit.

Quantum Circuit with the Minimal Toffoli-depth

Theorem (Asiacrypt 2022, Huang & Sun)

Given a classical circuit with **AND-depth s**, the Toffoli-depth of the quantum circuit implementing all the nodes of the classical circuit **is not smaller than s**. Moreover, **with sufficiently many ancillae**, we can construct a quantum circuit implementing all the nodes of the classical circuit with **Toffoli-depth s**.

1. Construct a classical circuit with the minimal AND-depth ($= \lceil \log_2(D) \rceil$);
   - D is the algebraic degree of the vectorial Boolean function F (With an AND-layer, one can mostly double the degree).
   - AES S-box (degree 7): the minimal AND-depth is 3.
2. Convert it to a Toffoli-depth-$\lceil \log_2(D) \rceil$ NCT circuit.

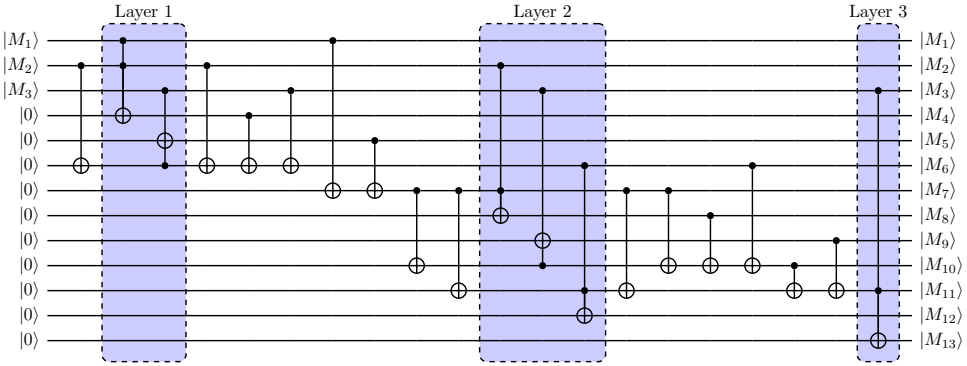
Quantum Circuit with the Minimal Toffoli-depth

Theorem (Asiacrypt 2022, Huang & Sun)

Given a classical circuit with **AND-depth s**, the Toffoli-depth of the quantum circuit implementing all the nodes of the classical circuit **is not smaller than s**. Moreover, **with sufficiently many ancillae**, we can construct a quantum circuit implementing all the nodes of the classical circuit with **Toffoli-depth s**.

1. Construct a classical circuit with the minimal AND-depth ($= \lceil \log_2(D) \rceil$);
   - D is the algebraic degree of the vectorial Boolean function F (With an AND-layer, one can mostly double the degree).
   - AES S-box (degree 7): the minimal AND-depth is 3.
2. Convert it to a Toffoli-depth-$\lceil \log_2(D) \rceil$ NCT circuit.

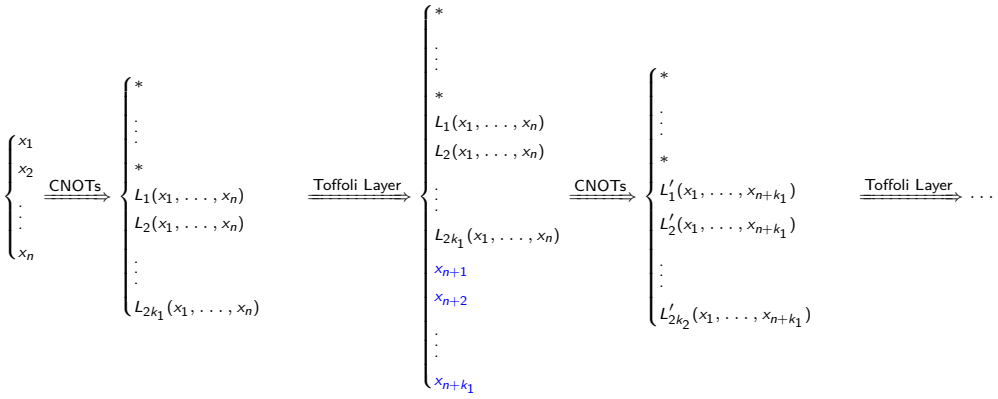
Automatic Conversion with Low Quantum Resource Cost



- Our idea: **Reconstruct the CNOT sub-circuits** that connect the Toffoli layers while maintaining the **lowest width and a low gate-count**.

Inputs and Outputs of the CNOT Sub-circuits

- CNOT gates generate linear expressions.
- A Toffoli layer generates some new Boolean variables.



Lemma 2

Let $\{L_1, L_2 \ldots, L_t\}$ and $\{T_1, T_2 \ldots, T_m\}$ be two sequences of linear functions w.r.t Boolean variables $x_1, x_2, \ldots, x_n$. Suppose **the rank of $L_1, L_2, \ldots, L_t$ is n, and the rank of $T_1, T_2, \ldots, T_m$ is k.** If $|L_1, L_2, \ldots, L_t\rangle$ is the input of a t-qubit register, then to **output the state $|T_1, T_2, \ldots, T_m\rangle$ using a CNOT circuit**, **$m - k - (t - n)$ additional qubits are necessary and sufficient**. Additionally, if $m - k - (t - n) < 0$, it means no additional qubits are required. Instead, $t - n - m + k$ qubits can be returned to $|0\rangle$.

Table: Different NCT circuits originated from an AND-depth-4 classical circuit for the AES S-box. Toffoli-depth=2× AND-depth, since uncomputation is included.

Type	#NOT	#CNOT	#Toffoli	Toffoli-depth	Full Depth	Width	Source
$\mathfrak{C}^*$	4	312	68	8	78	90	Aisacrypt 2023
$\mathfrak{C}^*$	4	368	68	8	105	76	Aisacrypt 2023
$\mathfrak{C}^0/\mathfrak{C}^*$	4	**227/240**	**60**	8	**60**	**66**	This work

Lemma 2

Let $\{L_1, L_2 \ldots, L_t\}$ and $\{T_1, T_2 \ldots, T_m\}$ be two sequences of linear functions w.r.t Boolean variables $x_1, x_2, \ldots, x_n$. Suppose **the rank of $L_1, L_2, \ldots, L_t$ is n, and the rank of $T_1, T_2, \ldots, T_m$ is k.** If $|L_1, L_2, \ldots, L_t\rangle$ is the input of a t-qubit register, then to **output the state $|T_1, T_2, \ldots, T_m\rangle$ using a CNOT circuit**, **$m - k - (t - n)$ additional qubits are necessary and sufficient**. Additionally, if $m - k - (t - n) < 0$, it means no additional qubits are required. Instead, $t - n - m + k$ qubits can be returned to $|0\rangle$.

Table: Different NCT circuits originated from an AND-depth-4 classical circuit for the AES S-box. Toffoli-depth=2× AND-depth, since uncomputation is included.

Type	#NOT	#CNOT	#Toffoli	Toffoli-depth	Full Depth	Width	Source
$\mathfrak{C}^*$	4	312	68	8	78	90	Aisacrypt 2023
$\mathfrak{C}^*$	4	368	68	8	105	76	Aisacrypt 2023
$\mathfrak{C}^0/\mathfrak{C}^*$	4	**227/240**	**60**	8	**60**	**66**	This work

Construct Classical Implementations with the Minimal AND-depth

- Trivial approach:
  - 1st AND layer: generate monomials with **degree** 2 simultaneously: $x_1x_2, \ldots, x_{n-1}x_n$;
  - 2rd AND layer: generate monomials with **degree** $3, 4$ simultaneously;
    $\ldots$
  - k-th AND layer: generate monomials with **degree from** $2^{k-1}+1$ **to** 2^k;
  - Construct F from these monomials by XOR gates.

- χ function of SHA3 : $(f_1, f_2, f_3, f_4, f_5) =$
  $(x_1 + (x_2 + 1)x_3, x_2 + (x_3 + 1)x_4, x_3 + (x_4 + 1)x_5, x_4 + (x_5 + 1)x_1, x_5 + (x_1 + 1)x_2)$.
  Construct $x_2 \cdot x_3, x_3 \cdot x_4, x_4 \cdot x_5, x_5 \cdot x_1, x_1 \cdot x_2$ in **one AND layer**.

Construct Classical Implementations with the Minimal AND-depth

- Trivial approach:
  - 1st AND layer: generate monomials with **degree** 2 simultaneously: $x_1x_2, \ldots, x_{n-1}x_n$;
  - 2rd AND layer: generate monomials with **degree** $3, 4$ simultaneously;
    $\ldots$
  - k-th AND layer: generate monomials with **degree from** $2^{k-1}+1$ **to** 2^k;
  - Construct F from these monomials by XOR gates.

- χ function of SHA3 : $(f_1, f_2, f_3, f_4, f_5) =$
  $(x_1 + (x_2 + 1)x_3, x_2 + (x_3 + 1)x_4, x_3 + (x_4 + 1)x_5, x_4 + (x_5 + 1)x_1, x_5 + (x_1 + 1)x_2)$.
  Construct $x_2 \cdot x_3, x_3 \cdot x_4, x_4 \cdot x_5, x_5 \cdot x_1, x_1 \cdot x_2$ in **one AND layer**.

- **Top-down approach**:
  - $\mathbb{M}_s$: **Polynomials in layer** s. Boolean polynomials with degree in the range $[2^{s-1}+1, 2^s]$, whose minimal AND-depth is s.
  - $f = f_d + f_{d-1} + \cdots + f_1$, where $f_i \in \mathbb{M}_i$. Generate $f_d \to f_{d-1} \to \cdots$
  - Compute **the max-depth cover** $\{C_1, C_2, \ldots, C_k\} \subset \mathbb{M}_d$ of f:

    $$f = C_1 + C_2 + \cdots + C_k + R,$$

    where $C_i = D_{i,1} \cdot D_{i,2}$, and $D_{i,j}, R \in \mathbb{M}_{d-1}$.

    Then recursively compute the covers of these $D_{i,j}$ and R.
  - Trivial cover (all monomials contained in f_d) is equivalent to the trivial approach.

- **Top-down approach**:
  - $\mathbb{M}_s$: **Polynomials in layer** s. Boolean polynomials with degree in the range $[2^{s-1}+1, 2^s]$, whose minimal AND-depth is s.
  - $f = f_d + f_{d-1} + \cdots + f_1$, where $f_i \in \mathbb{M}_i$. Generate $f_d \to f_{d-1} \to \cdots$
  - Compute **the max-depth cover** $\{C_1, C_2, \ldots, C_k\} \subset \mathbb{M}_d$ of f:

$$f = C_1 + C_2 + \cdots + C_k + R,$$

    where $C_i = D_{i,1} \cdot D_{i,2}$, and $D_{i,j}, R \in \mathbb{M}_{d-1}$.

    Then recursively compute the covers of these $D_{i,j}$ and R.

  - Trivial cover (all monomials contained in f_d) is equivalent to the trivial approach.

- **Top-down approach**:
  - $\mathbb{M}_s$: **Polynomials in layer** s. Boolean polynomials with degree in the range $[2^{s-1}+1, 2^s]$, whose minimal AND-depth is s.
  - $f = f_d + f_{d-1} + \cdots + f_1$, where $f_i \in \mathbb{M}_i$. Generate $f_d \to f_{d-1} \to \cdots$
  - Compute **the max-depth cover** $\{C_1, C_2, \ldots, C_k\} \subset \mathbb{M}_d$ of f:

    $$f = C_1 + C_2 + \cdots + C_k + R,$$

    where $C_i = D_{i,1} \cdot D_{i,2}$, and $D_{i,j}, R \in \mathbb{M}_{d-1}$.

    Then recursively compute the covers of these $D_{i,j}$ and R.
  - Trivial cover (all monomials contained in f_d) is equivalent to the trivial approach.

- **Top-down approach**:
  - $\mathbb{M}_s$: **Polynomials in layer** s. Boolean polynomials with degree in the range $[2^{s-1}+1, 2^s]$, whose minimal AND-depth is s.
  - $f = f_d + f_{d-1} + \cdots + f_1$, where $f_i \in \mathbb{M}_i$. Generate $f_d \to f_{d-1} \to \cdots$
  - Compute **the max-depth cover** $\{C_1, C_2, \ldots, C_k\} \subset \mathbb{M}_d$ of f:

    $$f = C_1 + C_2 + \cdots + C_k + R,$$

    where $C_i = D_{i,1} \cdot D_{i,2}$, and $D_{i,j}, R \in \mathbb{M}_{d-1}$.

    Then recursively compute the covers of these $D_{i,j}$ and R.
  - Trivial cover (all monomials contained in f_d) is equivalent to the trivial approach.

Construct Nontrivial Covers

Example 1

Let

$$f = x_1x_2x_3x_4x_5x_6 + x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_6 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6 + x_2x_3x_4x_5x_6.$$

$$\mathcal{C} = \{(x_1x_2x_3 + x_2x_3 + x_1x_2 + x_1x_3) \cdot (x_4x_5x_6 + x_4x_5 + x_4x_6 + x_5x_6)\}$$

is a max-depth cover of f with size 1.

- Greedy approach: gradually enlarge the monomial sets S_1, S_2, such that the product $(\sum_{p_i \in S_1} p_i)(\sum_{q_j \in S_2} q_j)$ covers more monomials in F.
- SAT-based Method: encode the relation $f = \sum_{i=1}^{k} \mathrm{D}_i^1 \cdot \mathrm{D}_i^2 + \mathrm{R}$ to Boolean equations, then solve them by an off-the-shelf SAT-solver.

Construct Nontrivial Covers

Example 1

Let

$$f = x_1x_2x_3x_4x_5x_6 + x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_6 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6 + x_2x_3x_4x_5x_6.$$

$$\mathcal{C} = \{(x_1x_2x_3 + x_2x_3 + x_1x_2 + x_1x_3) \cdot (x_4x_5x_6 + x_4x_5 + x_4x_6 + x_5x_6)\}$$

is a max-depth cover of f with size 1.

- Greedy approach: gradually enlarge the monomial sets S_1, S_2, such that the product $(\sum_{p_i \in S_1} p_i)(\sum_{q_j \in S_2} q_j)$ covers more monomials in F.
- SAT-based Method: encode the relation $f = \sum_{i=1}^{k} \mathrm{D}_i^1 \cdot \mathrm{D}_i^2 + \mathrm{R}$ to Boolean equations, then solve them by an off-the-shelf SAT-solver.

Construct Nontrivial Covers

Example 1

Let

$$f = x_1x_2x_3x_4x_5x_6 + x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_6 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6 + x_2x_3x_4x_5x_6.$$

$$\mathcal{C} = \{(x_1x_2x_3 + x_2x_3 + x_1x_2 + x_1x_3) \cdot (x_4x_5x_6 + x_4x_5 + x_4x_6 + x_5x_6)\}$$

is a max-depth cover of f with size 1.

- Greedy approach: gradually enlarge the monomial sets S_1, S_2, such that the product $(\sum_{p_i \in S_1} p_i)(\sum_{q_j \in S_2} q_j)$ covers more monomials in F.
- SAT-based Method: encode the relation $f = \sum_{i=1}^{k} \mathrm{D}_i^1 \cdot \mathrm{D}_i^2 + \mathrm{R}$ to Boolean equations, then solve them by an off-the-shelf SAT-solver.

Applications

- AES S-box (degree 7): AND-depth 3,**AND-count 76**.
  - Asiacrypt 2022: AND-depth 3, **AND-count 78**, modified from an AND-depth-4 implementation with some heuristics.
- SKINNY S-box (degree 7): AND-depth 3, AND-count 10.

- Top-down approach only needs the ANF (algebraic normal form) of F.
- For Boolean functions with specific structure, can we find minimal-AND-depth implementations with lower AND-count?

Applications

- AES S-box (degree 7): AND-depth 3,**AND-count 76**.
  - Asiacrypt 2022: AND-depth 3, **AND-count 78**, modified from an AND-depth-4 implementation with some heuristics.
- SKINNY S-box (degree 7): AND-depth 3, AND-count 10.

- Top-down approach only needs the ANF (algebraic normal form) of F.
- For Boolean functions with specific structure, can we find minimal-AND-depth implementations with lower AND-count?

Applications

- AES S-box (degree 7): AND-depth 3,**AND-count 76**.
  - Asiacrypt 2022: AND-depth 3, **AND-count 78**, modified from an AND-depth-4 implementation with some heuristics.
- SKINNY S-box (degree 7): AND-depth 3, AND-count 10.

- Top-down approach only needs the ANF (algebraic normal form) of F.
- For Boolean functions with specific structure, can we find minimal-AND-depth implementations with lower AND-count?

Outline

Minimal AND-depth for Implementing the Inversion

The problem

- $\alpha \in \mathbb{F}_{2^n}$: $x_1\beta_1 + x_2\beta_2 + \cdots + x_n\beta_n$, where $x_1, \ldots, x_n \in \mathbb{F}_2$.
- α^{-1} can be written as
  $f_1(x_1, x_2, \ldots, x_n)\beta_1 + f_2(x_1, x_2, \ldots, x_n)\beta_2 + \cdots + f_n(x_1, x_2, \ldots, x_n)\beta_n$.
- Goal: implementing $F = (f_1, f_2, \ldots, f_n)$.

- It is easy to prove that $deg(F) = n - 1$.

Theorem

The minimal AND-depth for implementing the multiplicative inversion in $\mathbb{F}_{2^n}$ is $\lceil \log_2(n-1) \rceil$.

Minimal AND-depth for Implementing the Inversion

The problem

- $\alpha \in \mathbb{F}_{2^n}$: $x_1\beta_1 + x_2\beta_2 + \cdots + x_n\beta_n$, where $x_1, \ldots, x_n \in \mathbb{F}_2$.
- α^{-1} can be written as
  $f_1(x_1, x_2, \ldots, x_n)\beta_1 + f_2(x_1, x_2, \ldots, x_n)\beta_2 + \cdots + f_n(x_1, x_2, \ldots, x_n)\beta_n$.
- Goal: implementing $F = (f_1, f_2, \ldots, f_n)$.

- It is easy to prove that $deg(F) = n - 1$.

Theorem

The minimal AND-depth for implementing the multiplicative inversion in $\mathbb{F}_{2^n}$ *is* $\lceil \log_2(n-1) \rceil$.

Addition Chain and Exponentiation

- $\alpha^{2^n-2} \cdot \alpha = \alpha^{2^n-1} = 1$, and if $\alpha = 0$, $\alpha^{2^n-2} = 0 \Rightarrow \alpha^{-1} = \alpha^{2^n-2}$.
- Compute $\alpha^{-1} = \alpha^{62} \in \mathbb{F}_{2^6}$:

  $1_2(1) \xrightarrow{\wedge 2} 10_2(2) \xrightarrow{\times} 11_2(3) \xrightarrow{\wedge 2} 110_2(6) \xrightarrow{\times} 111_2(7) \xrightarrow{\wedge 4} 11100_2(28) \xrightarrow{\times} 11111_2(31) \xrightarrow{\wedge 2} 111110_2(62)$

  - Squaring can be implemented without AND gate, squaring will not change the Hamming weight.
  - The change of the Hamming weight: $1 \to 2 \to 3 \to 5$
  - 3 additions correspond to 3 multiplications in $\mathbb{F}_{2^6}$.

Lemma

If $k = 2^r s$, for some positive number r and odd number s, then the multiplication of two elements in $\mathbb{F}_{2^k}$ can be implemented by **one AND layer and $\omega(k) = 3^r s^2$ AND gates**.

  - AND-depth 3, AND-count $3 \cdot 27$.

- An addition chain for $n-1$ can induce an implementation of the inversion in $\mathbb{F}_{2^n}$.

Addition Chain and Exponentiation

- $\alpha^{2^n-2} \cdot \alpha = \alpha^{2^n-1} = 1$, and if $\alpha = 0$, $\alpha^{2^n-2} = 0 \Rightarrow \alpha^{-1} = \alpha^{2^n-2}$.
- Compute $\alpha^{-1} = \alpha^{62} \in \mathbb{F}_{2^6}$:

$1_2(1) \xrightarrow{\wedge 2} 10_2(2) \xrightarrow{\times} 11_2(3) \xrightarrow{\wedge 2} 110_2(6) \xrightarrow{\times} 111_2(7) \xrightarrow{\wedge 4} 11100_2(28) \xrightarrow{\times} 11111_2(31) \xrightarrow{\wedge 2} 111110_2(62)$

  - Squaring can be implemented without AND gate, squaring will not change the Hamming weight.
  - The change of the Hamming weight: $1 \to 2 \to 3 \to 5$
  - 3 additions correspond to 3 multiplications in $\mathbb{F}_{2^6}$.

Lemma

If $k = 2^r s$, for some positive number r and odd number s, then the multiplication of two elements in $\mathbb{F}_{2^k}$ can be implemented by **one AND layer and $\omega(k) = 3^r s^2$ AND gates**.

  - AND-depth 3, AND-count $3 \cdot 27$.

- An addition chain for $n-1$ can induce an implementation of the inversion in $\mathbb{F}_{2^n}$.

Addition Chain and Exponentiation

- $\alpha^{2^n-2} \cdot \alpha = \alpha^{2^n-1} = 1$, and if $\alpha = 0$, $\alpha^{2^n-2} = 0 \Rightarrow \alpha^{-1} = \alpha^{2^n-2}$.
- Compute $\alpha^{-1} = \alpha^{62} \in \mathbb{F}_{2^6}$:

  $1_2(1) \xrightarrow{\wedge 2} 10_2(2) \xrightarrow{\times} 11_2(3) \xrightarrow{\wedge 2} 110_2(6) \xrightarrow{\times} 111_2(7) \xrightarrow{\wedge 4} 11100_2(28) \xrightarrow{\times} 11111_2(31) \xrightarrow{\wedge 2} 111110_2(62)$

  - Squaring can be implemented without AND gate, squaring will not change the Hamming weight.
  - The change of the Hamming weight: $1 \to 2 \to 3 \to 5$
  - 3 additions correspond to 3 multiplications in $\mathbb{F}_{2^6}$.

Lemma

If $k = 2^r s$, for some positive number r and odd number s, then the multiplication of two elements in $\mathbb{F}_{2^k}$ can be implemented by **one AND layer and $\omega(k) = 3^r s^2$ AND gates**.

  - AND-depth 3, AND-count $3 \cdot 27$.

- An addition chain for $n-1$ can induce an implementation of the inversion in $\mathbb{F}_{2^n}$.

Addition Chain and Exponentiation

- $\alpha^{2^n-2} \cdot \alpha = \alpha^{2^n-1} = 1$, and if $\alpha = 0$, $\alpha^{2^n-2} = 0 \Rightarrow \alpha^{-1} = \alpha^{2^n-2}$.
- Compute $\alpha^{-1} = \alpha^{62} \in \mathbb{F}_{2^6}$:

  $1_2(1) \xrightarrow{\wedge 2} 10_2(2) \xrightarrow{\times} 11_2(3) \xrightarrow{\wedge 2} 110_2(6) \xrightarrow{\times} 111_2(7) \xrightarrow{\wedge 4} 11100_2(28) \xrightarrow{\times} 11111_2(31) \xrightarrow{\wedge 2} 111110_2(62)$

  - Squaring can be implemented without AND gate, squaring will not change the Hamming weight.
  - The change of the Hamming weight: $1 \to 2 \to 3 \to 5$
  - 3 additions correspond to 3 multiplications in $\mathbb{F}_{2^6}$.

Lemma

If $k = 2^r s$, for some positive number r and odd number s, then the multiplication of two elements in $\mathbb{F}_{2^k}$ can be implemented by **one AND layer and $\omega(k) = 3^r s^2$ AND gates**.

  - AND-depth 3, AND-count $3 \cdot 27$.

- An addition chain for $n-1$ can induce an implementation of the inversion in $\mathbb{F}_{2^n}$.

Addition Chain and Exponentiation

- $\alpha^{2^n-2} \cdot \alpha = \alpha^{2^n-1} = 1$, and if $\alpha = 0$, $\alpha^{2^n-2} = 0 \Rightarrow \alpha^{-1} = \alpha^{2^n-2}$.
- Compute $\alpha^{-1} = \alpha^{62} \in \mathbb{F}_{2^6}$:

  $1_2(1) \xrightarrow{\wedge 2} 10_2(2) \xrightarrow{\times} 11_2(3) \xrightarrow{\wedge 2} 110_2(6) \xrightarrow{\times} 111_2(7) \xrightarrow{\wedge 4} 11100_2(28) \xrightarrow{\times} 11111_2(31) \xrightarrow{\wedge 2} 111110_2(62)$

  - Squaring can be implemented without AND gate, squaring will not change the Hamming weight.
  - The change of the Hamming weight: $1 \to 2 \to 3 \to 5$
  - 3 additions correspond to 3 multiplications in $\mathbb{F}_{2^6}$.

Lemma

If $k = 2^r s$, for some positive number r and odd number s, then the multiplication of two elements in $\mathbb{F}_{2^k}$ can be implemented by **one AND layer and $\omega(k) = 3^r s^2$ AND gates**.

  - AND-depth 3, AND-count $3 \cdot 27$.

- An addition chain for $n-1$ can induce an implementation of the inversion in $\mathbb{F}_{2^n}$.

Parallel Addition Chain

- Shortest addition chain $\neq$ Minimal-AND-depth implementation:
  - $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \Rightarrow$ 4 AND layers
  - $\alpha^{-1} \in \mathbb{F}_{2^8}$: the minimal AND-depth is 3.

Parallel Addition Chain

- Shortest addition chain $\neq$ Minimal-AND-depth implementation:
  - $1 \to 2 \to 3 \to 5 \to 7 \Rightarrow$ 4 AND layers
  - $\alpha^{-1} \in \mathbb{F}_{2^8}$: the minimal AND-depth is 3.
- Parallel Addition Chain:

$$A = \begin{bmatrix} 1 & 2 & 4 & 7 \\ 1 & 0 & 3 & 0 \end{bmatrix}$$

$$\mathcal{C}_1 : 1 \xrightarrow{+1} 2 \xrightarrow{+2} 4 \xrightarrow[\mathcal{C}_2]{+3} 7$$
$$\mathcal{C}_2 : 1 \longrightarrow 1 \xrightarrow[\mathcal{C}_1]{+2} 3$$

$$\alpha \xrightarrow{\wedge 2} \alpha^{10_2} \xrightarrow{\times \alpha^{1}} \alpha^{11_2} \xrightarrow{\wedge 4} \alpha^{1100_2} \xrightarrow{\times \alpha^{11_2}} \alpha^{1111_2} \xrightarrow{\wedge 16} \alpha^{11110000_2} \xrightarrow{\times \alpha^{1110_2}} \alpha^{11111110_2}$$
$$\alpha^{10_2} \longrightarrow \alpha^{10_2} \longrightarrow \alpha^{10_2} \xrightarrow{\times \alpha^{1100_2}} \alpha^{1110_2}$$

Lemma

For any n, there is a parallel addition chain for n with the minimal depth $\lceil \log_2(n) \rceil$ and involving $\mathrm{HW}(n) + \lfloor \log_2(n) \rfloor - 1$ additions.

$$\begin{bmatrix} 1 & 2 & \cdots & 2^{k_1} & 2^{k_1+1} & \cdots & 2^{k_2} & 2^{k_2+1} & \cdots & 2^{k_3} & \cdots & 2^{k_m} & 0 \\ 0 & 0 & \cdots & 2^{k_1} & 0 & \cdots & 0 & \sum_{i=1}^{2} 2^{k_i} & \cdots & 0 & \cdots & 0 & \sum_{i=1}^{m} 2^{k_i} \end{bmatrix}$$

Theorem

There is a classical circuit implementing the inversion in $\mathbb{F}_{2^n}$ with AND-depth $\lceil \log_2(n-1) \rceil$ and AND-count $\omega(n)(\mathrm{HW}(n-1) + \lfloor \log_2(n-1) \rfloor - 1)$.

- RAIN-128 (an MPC-friendly block cipher, CCS 2022) S-box, the inversion in $\mathbb{F}_{2^{128}}$:

$$A = \begin{bmatrix} 1 & 2 & 4 & 8 & 16 & 32 & 64 & 127 \\ 1 & 0 & 3 & 7 & 15 & 31 & 63 & 0 \end{bmatrix},$$

  AND-depth 7 and AND-count 24057.

Lemma

For any n, there is a parallel addition chain for n with the minimal depth $\lceil \log_2(n) \rceil$ and involving $\mathrm{HW}(n) + \lfloor \log_2(n) \rfloor - 1$ additions.

$$\begin{bmatrix} 1 & 2 & \cdots & 2^{k_1} & 2^{k_1+1} & \cdots & 2^{k_2} & 2^{k_2+1} & \cdots & 2^{k_3} & \cdots & 2^{k_m} & 0 \\ 0 & 0 & \cdots & 2^{k_1} & 0 & \cdots & 0 & \sum_{i=1}^{2} 2^{k_i} & \cdots & 0 & \cdots & 0 & \sum_{i=1}^{m} 2^{k_i} \end{bmatrix}$$

Theorem

There is a classical circuit implementing the inversion in $\mathbb{F}_{2^n}$ with AND-depth $\lceil \log_2(n-1) \rceil$ and AND-count $\omega(n)(\mathrm{HW}(n-1) + \lfloor \log_2(n-1) \rfloor - 1)$.

- RAIN-128 (an MPC-friendly block cipher, CCS 2022) S-box, the inversion in $\mathbb{F}_{2^{128}}$:

$$A = \begin{bmatrix} 1 & 2 & 4 & 8 & 16 & 32 & 64 & 127 \\ 1 & 0 & 3 & 7 & 15 & 31 & 63 & 0 \end{bmatrix},$$

AND-depth 7 and AND-count 24057.

Lemma

For any n, there is a parallel addition chain for n with the minimal depth $\lceil \log_2(n) \rceil$ and involving $\mathrm{HW}(n) + \lfloor \log_2(n) \rfloor - 1$ additions.

$$\begin{bmatrix} 1 & 2 & \cdots & 2^{k_1} & 2^{k_1+1} & \cdots & 2^{k_2} & 2^{k_2+1} & \cdots & 2^{k_3} & \cdots & 2^{k_m} & 0 \\ 0 & 0 & \cdots & 2^{k_1} & 0 & \cdots & 0 & \sum_{i=1}^{2} 2^{k_i} & \cdots & 0 & \cdots & 0 & \sum_{i=1}^{m} 2^{k_i} \end{bmatrix}$$

Theorem

There is a classical circuit implementing the inversion in $\mathbb{F}_{2^n}$ with AND-depth $\lceil \log_2(n-1) \rceil$ and AND-count $\omega(n)(\mathrm{HW}(n-1) + \lfloor \log_2(n-1) \rfloor - 1)$.

- RAIN-128 (an MPC-friendly block cipher, CCS 2022) S-box, the inversion in $\mathbb{F}_{2^{128}}$:

$$A = \begin{bmatrix} 1 & 2 & 4 & 8 & 16 & 32 & 64 & 127 \\ 1 & 0 & 3 & 7 & 15 & 31 & 63 & 0 \end{bmatrix},$$

  AND-depth 7 and AND-count 24057.

Implementing the Inversion in $\mathbb{F}_{2^{2m}}$

- Tower field structure: $\alpha = a_0\beta^{2^m} + a_1\beta$,

$$\alpha^{2^m} = a_1\beta^{2^m} + a_0\beta, \quad b = \alpha \cdot \alpha^{2^m} = a_0a_1\beta^{2^{m+1}} + (a_0^2 + a_1^2)\beta^{2^m+1} + a_0a_1\beta^2$$

  and $\alpha^{-1} = b^{-1}a_1\beta^{2^m} + b^{-1}a_0\beta$.

- Three Steps: (1) compute $b \in \mathbb{F}_{2^m}$; (2) b^{-1}; (3) $b^{-1}a_1, b^{-1}a_0$;

Implementing the Inversion in $\mathbb{F}_{2^{2m}}$

- Tower field structure: $\alpha = a_0\beta^{2^m} + a_1\beta$,

$$\alpha^{2^m} = a_1\beta^{2^m} + a_0\beta, \quad b = \alpha \cdot \alpha^{2^m} = a_0a_1\beta^{2^{m+1}} + (a_0^2 + a_1^2)\beta^{2^m+1} + a_0a_1\beta^2$$

  and $\alpha^{-1} = b^{-1}a_1\beta^{2^m} + b^{-1}a_0\beta$.

- Three Steps: (1) compute $b \in \mathbb{F}_{2^m}$; (2) b^{-1}; (3) $b^{-1}a_1, b^{-1}a_0$;

- **Merge Step (2)** (compute a parallel addition chain of $m-1$) **and (3)**:

$$A_1 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 0 \end{bmatrix} \Rightarrow A_2 = \begin{bmatrix} 1 & 2 & 3 \boxplus a_1 \\ 1 & 1 \boxplus a_0 & 3 \boxplus a_0 \\ 1 & 1 \boxplus a_1 & 0 \end{bmatrix}$$

$$\begin{array}{l} b \xrightarrow{\wedge 4} b^{100_2} \xrightarrow{\times b^1} b^{101_2} \xrightarrow{\wedge 2} b^{1010_2} \xrightarrow{\times b^{100_2}a_1} b^{1110_2}a_1 = b^{-1}a_1 \\ \quad\quad b^{100_2} \xrightarrow{\times a_0} b^{100_2}a_0 \rightarrow b^{100_2}a_0 \xrightarrow{\times b^{1010_2}} b^{1110_2}a_0 = b^{-1}a_0 \\ \quad\quad b^{100_2} \xrightarrow{\times a_1} b^{100_2}a_1 \rightarrow b^{100_2}a_1 \end{array}$$

Theorem

Let $m-1=\sum_{i=1}^{s}2^{k_i}$ *with* $0\le k_1<k_2<\cdots<k_s$ *and* $s\ge 2$. *If* $k_s>\lceil\log_2(\sum_{i=1}^{s-1}2^{k_i})\rceil$, *then* $\alpha^{-1}\in\mathbb{F}_{2^{2m}}$ *can be implemented by a classical circuit with the minimal AND-depth* $\lceil\log_2(2m-1)\rceil$ *and AND-count* $\omega(m)(HW(m-1)+\lfloor\log_2(m-1)\rfloor+3)$.

$$A=\begin{bmatrix} 1 & 2 & \cdots & 2^{k_1} & 2^{k_1+1} & \cdots & 2^{k_2} & 2^{k_2+1} & \cdots & 2^{k_{s-1}} & 2^{k_{s-1}+1} & \cdots & \mathbf{2^{k_s}} & 0 \\ 0 & 0 & \cdots & 2^{k_1} & 0 & \cdots & 0 & \sum_{i=1}^{2}2^{k_i} & \cdots & 0 & \sum_{i=1}^{s-1}\mathbf{2^{k_i}} & \cdots & 0 & m-1 \end{bmatrix}.$$

- AES S-box: AND-depth 3 , **AND-count 42** (Asiacrypt 2022, AND-count 78).

- Convert to Clifford+T circuit with our new conversion algorithm.

Table: Clifford+T circuits (including uncomputation) for the AES S-box with T-depth 3.

Type	#CNOT	#1qClifford	#T	#M	T-depth	Full Depth	Width	Source
$\mathfrak{C}^0/\mathfrak{C}^*$	1396/1398	494	312	78	3	119	218/226	Asiacrypt 2022
$\mathfrak{C}^*$	1110	448	264	66	3	92	129	IEEE TC 2024
$\mathfrak{C}^0/\mathfrak{C}^*$	**827/856**	**266/298**	**168**	**34/42**	3	**85/87**	**89/97**	This work

New Results for Implementing Quantum Oracles for AES

Table: The Costs of Grover Oracles based on the Pipeline Structure.

#CNOT	#1qClifford	#T	#M	T-depth	Full Depth	Width	Source
456040	179200	105600	26400	60	1802	3796	IEEE TC 2024
353160	**119200**	**67200**	**16800**	60	**1782**	**3156**	This work

Table: The Costs of Encryption Oracles based on the Interlacing-Uncompute Structure.

#CNOT	#1qClifford	#T	#M	T-depth	Full Depth	Width	Source
364360	144584	84480	21120	33	1078	4128	IEEE TC 2024
281896	**96584**	**53760**	**13440**	33	**1066**	**3104**	This work

Outline

Minimal-width NCT implementations

- An invertible vectorial Boolean function F is **a permutation $\mathcal{P}$ on $\mathbb{F}_2^n$.**
- It always has an in-place NCT implementation:
  - $\mathcal{P}$ is even: 0 ancilla qubit, minimal-width$= n$
  - $\mathcal{P}$ is odd: 1 ancilla qubit, minimal-width$= n + 1$

Vivek V. Shende, Aditya K. Prasad, Igor L. Markov, John P. Hayes: Synthesis of reversible logic circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 22(6): 710-722 (2003)

- How to obtain the minimal-width implementations?
- AES S-box: [PQCrypto 2016, Grassl et al.] 9 qubits with no more than 9695 T-gates and 12631 Clifford gates. The specific circuit was not presented.

Minimal-width NCT implementations

- An invertible vectorial Boolean function F is **a permutation $\mathcal{P}$ on $\mathbb{F}_2^n$.**
- It always has an in-place NCT implementation:
  - $\mathcal{P}$ is even: 0 ancilla qubit, minimal-width$= n$
  - $\mathcal{P}$ is odd: 1 ancilla qubit, minimal-width$= n + 1$

Vivek V. Shende, Aditya K. Prasad, Igor L. Markov, John P. Hayes: Synthesis of reversible logic circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 22(6): 710-722 (2003)

- How to obtain the minimal-width implementations?

- AES S-box: [PQCrypto 2016, Grassl et al.] 9 qubits with no more than 9695 T-gates and 12631 Clifford gates. The specific circuit was not presented.

SAT-based Method

- ≤ 5 qubit: at most one Toffoli gate in each layer.
- $F = \mathcal{S}_k \circ \mathcal{T}_k \circ \mathcal{S}_{k-1} \circ \cdots \circ \mathcal{S}_2 \circ \mathcal{T}_2 \circ \mathcal{S}_1 \circ \mathcal{T}_1 \circ \mathcal{S}_0$: $\mathcal{T}$ corresponds a Toffoli gate, $\mathcal{S}$ corresponds to all possible affine transformation.
  - **Affine Layers**: $\forall i \in \{1, \ldots, w\}, \forall j \in \{0, \ldots, k\}$, $B_{i,j} = c_{1,i}^{(j)} A_{1,j} + \cdots + c_{w,i}^{(j)} A_{w,j} + d_i^{(j)}$, for some Boolean variables $c_{1,i}^{(j)}, \ldots, c_{w,i}^{(j)}$ and $d_i^{(j)}$.
  - **Toffoli layers**: $\forall j \in \{1, \ldots, k\}$, $A_{3,j} = B_{3,(j-1)} + B_{1,(j-1)} \cdot B_{2,(j-1)}$, and $A_{i,j} = B_{i,(j-1)}$ if $i \neq 3$.
  - **Inputs and outputs**: $\forall i \in \{1, \ldots, n\}$, $A_{i,0} = x_i$, $B_{i,k} = f_i(x_1, x_2, \ldots, x_n)$.
  - **Ancilla qubit (for an odd F)**: If $w = n + 1$, $A_{w,0} = 0$ and $B_{w,k} = 0$.

SAT-based Method

- ≤ 5 qubit: at most one Toffoli gate in each layer.
- $F = \mathcal{S}_k \circ \mathcal{T}_k \circ \mathcal{S}_{k-1} \circ \cdots \circ \mathcal{S}_2 \circ \mathcal{T}_2 \circ \mathcal{S}_1 \circ \mathcal{T}_1 \circ \mathcal{S}_0$: $\mathcal{T}$ corresponds a Toffoli gate, $\mathcal{S}$ corresponds to all possible affine transformation.
  - **Affine Layers**: $\forall i \in \{1, \ldots, w\}, \forall j \in \{0, \ldots, k\}$, $\mathrm{B}_{i,j} = c^{(j)}_{1,i}\mathrm{A}_{1,j} + \cdots + c^{(j)}_{w,i}\mathrm{A}_{w,j} + d^{(j)}_i$, for some Boolean variables $c^{(j)}_{1,i}, \ldots, c^{(j)}_{w,i}$ and $d^{(j)}_i$.
  - **Toffoli layers**: $\forall j \in \{1, \ldots, k\}$, $\mathrm{A}_{3,j} = \mathrm{B}_{3,(j-1)} + \mathrm{B}_{1,(j-1)} \cdot \mathrm{B}_{2,(j-1)}$, and $\mathrm{A}_{i,j} = \mathrm{B}_{i,(j-1)}$ if $i \neq 3$.
  - **Inputs and outputs**: $\forall i \in \{1, \ldots, n\}$, $\mathrm{A}_{i,0} = x_i$, $\mathrm{B}_{i,k} = f_i(x_1, x_2, \ldots, x_n)$.
  - **Ancilla qubit (for an odd F)**: If $w = n + 1$, $\mathrm{A}_{w,0} = 0$ and $\mathrm{B}_{w,k} = 0$.

Improve the Encoding Scheme

- Exclude equivalent solutions (Fix the Toffoli gate):
  - $\mathcal{T}_{i,j,k} = \text{Rewire} \circ \mathcal{T}_{1,2,3} \circ \text{Rewire}^{-1}$

$$\Downarrow$$

$$\mathcal{S}_1 \circ \mathcal{T}_{i,j,k} \circ \mathcal{S}_2 = (\mathcal{S}_1 \circ \text{Rewire}) \circ \mathcal{T}_{1,2,3} \circ (\text{Rewire}^{-1} \circ \mathcal{S}_2) = \mathcal{S}_1' \circ \mathcal{T}_{1,2,3} \circ \mathcal{S}_2'$$

- Meet-in-the-Middle: **Quantum circuits are reversible.** Build the equations forward and backward respectively, then meet in the middle.
  - Should add some constrains to ensure the affine layers are invertible.

Improve the Encoding Scheme

- Exclude equivalent solutions (Fix the Toffoli gate):
  - $\mathcal{T}_{i,j,k} = \text{Rewire} \circ \mathcal{T}_{1,2,3} \circ \text{Rewire}^{-1}$

$$\Downarrow$$

$$\mathcal{S}_1 \circ \mathcal{T}_{i,j,k} \circ \mathcal{S}_2 = (\mathcal{S}_1 \circ \text{Rewire}) \circ \mathcal{T}_{1,2,3} \circ (\text{Rewire}^{-1} \circ \mathcal{S}_2) = \mathcal{S}_1' \circ \mathcal{T}_{1,2,3} \circ \mathcal{S}_2'$$

- Meet-in-the-Middle: **Quantum circuits are reversible.** Build the equations forward and backward respectively, then meet in the middle.
  - Should add some constrains to ensure the affine layers are invertible.

Applications

Table: Costs of the 5-qubit quantum circuits for the χ function of SHA3.

Type	#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth	Source
NCT	12	0	**7**	**5**	7	10	This work
Type	#CNOT	#1qClifford	$\#T$	Width	T-depth	Full Depth	Source
Clifford+T	79	24	70	12	30	103	eprint 2023
	49	**24**	**49**	**5**	**21**	**66**	This work

- Implementing χ requires at least 5 Toffoli gates without width limit. Toffoli-count 7 is almost optimal.

Table: Costs of the 5-qubit quantum circuit for the S-box of ASCON.

Type	#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth	Source
NCT	10	38	7	5	7	44	This work

Applications

Table: Costs of the 5-qubit quantum circuits for the χ function of SHA3.

Type	#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth	Source
NCT	12	0	**7**	**5**	7	10	This work
Type	#CNOT	#1qClifford	$\#T$	Width	T-depth	Full Depth	Source
Clifford+T	79	24	70	12	30	103	eprint 2023
	49	**24**	**49**	**5**	**21**	**66**	This work

- Implementing χ requires at least 5 Toffoli gates without width limit. Toffoli-count 7 is almost optimal.

Table: Costs of the 5-qubit quantum circuit for the S-box of ASCON.

Type	#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth	Source
NCT	10	38	7	5	7	44	This work

Method Based on MCT Implementations

- C^mX gate: an MCT (Multiple Controlled Toffoli) gate with m control qubits, maps $|x_1, x_2, \cdots, x_k\rangle |x_{k+1}\rangle$ to $|x_1, x_2, \cdots, x_k\rangle |x_{k+1} \oplus x_1 x_2 \cdots x_k\rangle$
- $C^0X =$ NOT, $C^1X =$ CNOT, $C^2X =$ Toffoli.
- **Our idea**:
  1. Construct an MCT implementation without ancilla qubit;
  2. Decompose each C^kX to Toffoli gates with **at most one ancilla qubit**.

Method Based on MCT Implementations

- C^mX gate: an MCT (Multiple Controlled Toffoli) gate with m control qubits, maps $|x_1, x_2, \cdots, x_k\rangle |x_{k+1}\rangle$ to $|x_1, x_2, \cdots, x_k\rangle |x_{k+1} \oplus x_1 x_2 \cdots x_k\rangle$
- $C^0X =$ NOT, $C^1X =$ CNOT, $C^2X =$ Toffoli.
- **Our idea**:
  1. Construct an MCT implementation without ancilla qubit;
  2. Decompose each C^kX to Toffoli gates with **at most one ancilla qubit**.

Decomposition of C^mX

Lemma (Physical Review A 1995, Barenco et al.)

A C^mX gate can be implemented by two C^pX gates and two (one) C^qX gate and one dirty (clean) ancilla qubit, where $p + q = m + 1$

Figure: Implementing a C^7X gate with one ancilla qubit.

Decomposition of C^mX

Lemma (Physical Review A 1995, Barenco et al.)

A C^mX gate with $m > 3$ can be implemented by $4(m-2)$ Toffoli gates and $m-2$ ancilla qubits(dirty qubits).

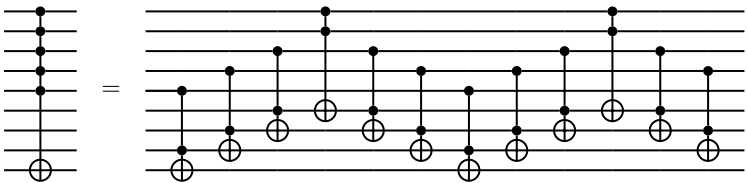


Figure: Implementing a C^5X gate with three **dirty ancilla qubits**.

- With on ancilla qubit (dirty or clean), one can decompose C^mX into Toffoli gates.

Decomposition of C^mX

Lemma (Physical Review A 1995, Barenco et al.)

A C^mX gate with $m > 3$ can be implemented by $4(m-2)$ Toffoli gates and $m-2$ ancilla qubits(dirty qubits).

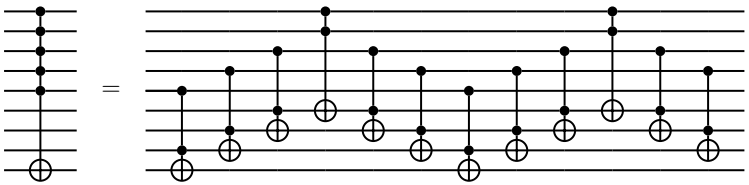


Figure: Implementing a C^5X gate with three **dirty ancilla qubits**.

- With on ancilla qubit (dirty or clean), one can decompose C^mX into Toffoli gates.

MCT Implementations from Tensor Decomposition

Lee et al.: An Algorithm for Reversible Logic Circuit Synthesis Based on Tensor Decomposition. arXiv:2107.04298, 2021

- $M_k M_{k-1} \cdots M_1 \mathcal{P}_n = \mathcal{P}_{n-1} \otimes I_2$
- $\mathcal{P}_{n-1}$ only involve the first $n-1$ wires. Recursively decompose $\mathcal{P}_{n-1}$.
- We can obtain a MCT decomposition consisting of $\{X, C^1X, C^2X, \ldots, C^{n-1}X\}$
  - For an even permutation, the decomposition does not contain $C^{n-1}X$ in most cases.
  - Combing the decomposition of C^mX, we can obtain the minimal-width NCT circuit for $\mathcal{P}_n$.

MCT Implementations from Tensor Decomposition

Lee et al.: An Algorithm for Reversible Logic Circuit Synthesis Based on Tensor Decomposition. arXiv:2107.04298, 2021

- $M_k M_{k-1} \cdots M_1 \mathcal{P}_n = \mathcal{P}_{n-1} \otimes I_2$
- $\mathcal{P}_{n-1}$ only involve the first $n-1$ wires. Recursively decompose $\mathcal{P}_{n-1}$.
- We can obtain a MCT decomposition consisting of $\{X, C^1X, C^2X, \ldots, C^{n-1}X\}$
  - For an even permutation, the decomposition does not contain $C^{n-1}X$ in most cases.
  - Combing the decomposition of C^mX, we can obtain the minimal-width NCT circuit for $\mathcal{P}_n$.

Minimal-width Implementations for the AES S-box

- Tensor Decomposition based method + SAT-based method (for 5-qubit sub-circuits)

Table: Costs of the 9-qubit NCT circuit for the AES S-box.

#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth
233	885	833	9	793	1594

Table: Costs of different 9-qubit Clifford+T circuits for the AES S-box.

#Clifford (CNOT, 1qClifford)	#T	Width	T-depth	Full Depth	Source
$\leq$ 12631(- , -)	$\leq$ 9295	9	-	-	PQCrypt 16
7465 (6028, 1437)	**3783**	**9**	**1501**	**7180**	This work
13008 (10633, 2375)	**3447**	**9**	**1274**	**9954**	This work (T-par)

Minimal-width Implementations for a Pair of S-boxes

- Implementing a pair of S-box, (S_1, S_2): use a qubit allocated for implementing S_2 as the dirty ancilla qubit when implementing S_1, and vice versa.

Table: Costs of the 16-qubit quantum circuits for a pair of AES S-boxes.

Type	#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth
NCT	502	1770	2140	16	1714	2990
Type	#1qClifford	#CNOT	#T	Width	T-depth	Full Depth
Clifford+T	3628	14786	9008	16	2949	15253
Clifford+T (T-par)	5066	23976	8360	16	2774	18883

- We can construct a 256-qubit quantum circuit of AES-128, achieving the theoretical minimum.

Minimal-width Implementations for a Pair of S-boxes

- Implementing a pair of S-box, (S_1, S_2): use a qubit allocated for implementing S_2 as the dirty ancilla qubit when implementing S_1, and vice versa.

Table: Costs of the 16-qubit quantum circuits for a pair of AES S-boxes.

Type	#NOT	#CNOT	#Toffoli	Width	Toffoli-depth	Full Depth
NCT	502	1770	2140	16	1714	2990
Type	#1qClifford	#CNOT	#T	Width	T-depth	Full Depth
Clifford+T	3628	14786	9008	16	2949	15253
Clifford+T (T-par)	5066	23976	8360	16	2774	18883

- We can construct a 256-qubit quantum circuit of AES-128, achieving the theoretical minimum.

Thank you for your attention!

huangzhenyu@iie.ac.cn