Synthesizing Quantum Circuits of AES with Lower T-depth and Less Qubits

Zhenyu Huang and Siwei Sun

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

Aisacryt 2022

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

中国科学院大学
University of Chinese Academy of Sciences

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 for the encryption process** is a part 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 for the encryption process** is a part 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

From classical circuits to quantum circuits:

- Classicial gates: XOR, NOT, AND
  $\Rightarrow$ CNT gate set: CNOT, NOT(Pauli-X), Toffoli
  $\Rightarrow$ Clifford+T gates: $\{$Pauli gates, CNOT, S, H$\}$ + T

Optimization Goals:

- Width: the number of qubits
- Gate count
- Depth: The number of layers of the circuit (gates acting on disjoint sets of qubits can be applied in parallel)
- In fault-tolerant quantum computation (Surface code), the cost of the T **gate is greatly higher** than that of a Clifford gate, and the running time of a circuit is **dominated by the T-depth**.

From classical circuits to quantum circuits:

- Classicial gates: XOR, NOT, AND
  $\Rightarrow$ CNT gate set: CNOT, NOT(Pauli-X), Toffoli
  $\Rightarrow$ Clifford+T gates: {Pauli gates, CNOT, S, H} + T

Optimization Goals:

- Width: the number of qubits
- Gate count
- Depth: The number of layers of the circuit (gates acting on disjoint sets of qubits can be applied in parallel)
- In fault-tolerant quantum computation (Surface code), the cost of the T **gate is greatly higher** than that of a Clifford gate, and the running time of a circuit is **dominated by the T-depth**.

Outline

The Pipeline Structure

Round Transformation:

- $Round_i : (key_i, x) \rightarrow (key_i, O(R_i))$
- $O(R_i)$: the output of the round function
- $\mathcal{R}_i : |key_i\rangle |x\rangle |0\rangle \rightarrow |key_i\rangle |x\rangle |O(R_i)\rangle$, **an out-of-place implementation**

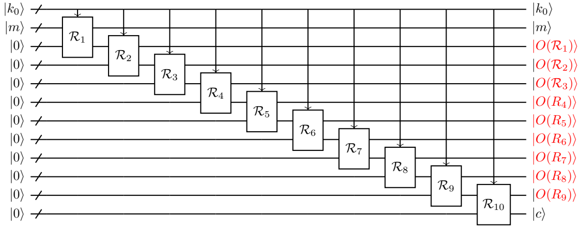


Figure: The pipeline structrue for AES-128

- Generates redundant output $O(\mathcal{R}_i)$ after each round

The Zig-zag Structure

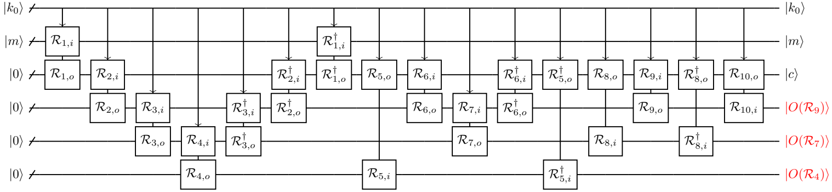


Figure: The zig-zag structure for AES-128

- The reverse circuit $R^\dagger$ is used to clean some redundant outputs.

The Out-of-Place Based Round-in-Place Structure

- For symmetric ciphers, each round is invertible, so theoretically there is an in-place quantum circuit for each round.
- However, directly obtain a such in-place circuit is very hard.
- We can construct it by combing two out-of-place sub-circuits.
  - Round transformation $R : (k, x) \to (k, T(x, k))$
  - T': the inverse function of T, $T'(k, T(x, k)) = x$

Figure: The op-based in-place circuit

The Out-of-Place Based Round-in-Place Structure

- For symmetric ciphers, each round is invertible, so theoretically there is an in-place quantum circuit for each round.
- However, directly obtain a such in-place circuit is very hard.
- We can construct it by combing two out-of-place sub-circuits.
  - Round transformation $R : (k, x) \to (k, T(x, k))$
  - T': the inverse function of T, $T'(k, T(x, k)) = x$

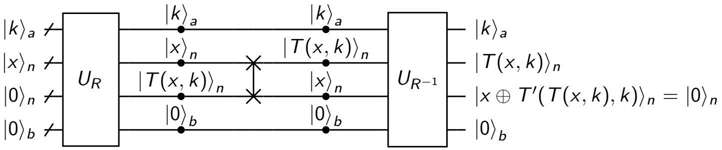


Figure: The op-based in-place circuit

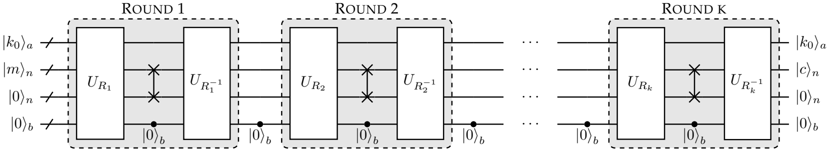


Figure: The OP-based round-in-place structure

- The width does not increase after each round

Comparison of Different Structures

- n qubits for input, n qubits for output, and r rounds.
- Same out-of-place round circuit using αn ancilla qubits .

Table: The widths (number of qubits) of different structures, where t is the minimal number such that $\sum_{i=1}^{t} i > r$.

Pipeline	Zig-zag	Round-in-place
$(r+\alpha+1)n$	$(t+1+\alpha)n \approx (\sqrt{2r}+\alpha)n$	$(2+\alpha)n$

Table: The depths and DW-costs of the oracles based on different structures

Metric	Type	Pipeline	Zig-zag	Round-in-place
Depth	Grover	$2r \cdot d$	$\approx 4r \cdot d$	$4r \cdot d$
	Encrypt	$2r \cdot d$	$\approx 4r \cdot d$	$2r \cdot d$
DW-cost	Grover	$2r(r+1+\alpha)nd$	$2r(\sqrt{2r}+\alpha)nd$	$4r(2+\alpha)nd$
	Encrypt	$2r(r+1+\alpha)nd$	$2r(\sqrt{2r}+\alpha)nd$	$2r(2+\alpha)nd$

Outline

Synthesizing Optimal CNOT Circuits

- Invertible linear transformation :
  $|x_1, x_2, \ldots, x_n\rangle \to |L_1(x_1, \ldots, x_n), \ldots, L_n(x_1, \ldots, x_n)\rangle$
- **Invertible linear transformation $\Rightarrow$ In-place CNOT circuit**
  - CNOT gate: $|x_1, x_2\rangle \to |x_1, x_1 \oplus x_2\rangle$, seen as a row addition elementary matrix
  - PLU decomposition: number of gates is large
  - Heuristic algorithm (Xiang et al. FSE 2020): greatly reduce the number of gates, but is not optimal.
- A new SAT-based method for implementing linear transformations with minimal number of CNOT gates
  - Encode the problem of finding a circuit with k gates into a SAT problem
  - $k-1$ (UNSAT), k (SAT) $\Rightarrow$ using k gates is optimal

Synthesizing Optimal CNOT Circuits

- Invertible linear transformation :
  $|x_1, x_2, \ldots, x_n\rangle \rightarrow |L_1(x_1, \ldots, x_n), \ldots, L_n(x_1, \ldots, x_n)\rangle$
- **Invertible linear transformation $\Rightarrow$ In-place CNOT circuit**
  - CNOT gate: $|x_1, x_2\rangle \rightarrow |x_1, x_1 \oplus x_2\rangle$, seen as a row addition elementary matrix
  - PLU decomposition: number of gates is large
  - Heuristic algorithm (Xiang et al. FSE 2020): greatly reduce the number of gates, but is not optimal.

- A new SAT-based method for implementing linear transformations with minimal number of CNOT gates
  - Encode the problem of finding a circuit with k gates into a SAT problem
  - $k-1$ (UNSAT), k (SAT) $\Rightarrow$ using k gates is optimal

The Way of Encoding

- Variable sets: $B = (b_{ij})_{k\times n}$, $C = (c_{ij})_{k\times n}$, $F = (f_{ij})_{n\times n}$, $\Psi = \{\psi_{i,j,s}\}_{k\times n\times n}$.
- B, C: $b_{ij_1} = c_{ij_2} = 1 \Rightarrow$ CNOT$_i$: Adds Wire$_{j_1}$ to Wire$_{j_2}$.
- F: $f_{ij} = 1 \Rightarrow L_i$ is the output of Wire$_j$.
- Ψ: $\psi_{i,j,k} = 1 \Rightarrow$ After CNOT$_i$, in the boolean expression (ANF) of Wire$_j$, coeff(x_k) is 1.

Boolean Equations for the CNOT circuit problem

$$EQN_b = \left\{ \begin{array}{c} b_{ij_1} b_{ij_2} = 0, \\ b_{i1} + b_{i2} + \cdots + b_{in} + 1 = 0, \\ \text{for } 1 \leq i \leq k, 1 \leq j_1 \neq j_2 \leq n \end{array} \right\} \quad EQN_c = \left\{ \begin{array}{c} c_{ij_1} c_{ij_2} = 0, \\ c_{i1} + c_{i2} + \cdots + c_{in} + 1 = 0, \\ \text{for } 1 \leq i \leq k, 1 \leq j_1 \neq j_2 \leq n \end{array} \right\}$$

$$EQN_a = \left\{ \begin{array}{c} f_{i,j}(\psi_{k,j,s} + a_{is}) = 0 \\ \text{for } 1 \leq i,j \leq n, 1 \leq s \leq m \end{array} \right\} \quad EQN_f = \left\{ \begin{array}{c} f_{ij_1} f_{ij_2} = 0, \\ f_{i1} + f_{i2} + \cdots + f_{in} + 1 = 0, \\ \text{for } 1 \leq i \leq n, 1 \leq j_1 \neq j_2 \leq n \end{array} \right\}$$

$$EQN_\psi = \left\{ \begin{array}{c} \psi_{i,j,s} + \sum_{t=1}^{n} c_{ij} b_{it} \psi_{i-1,t,s} + \psi_{i-1,j,s} = 0, \\ \text{for } 1 \leq i \leq k, 1 \leq j \leq n, 1 \leq s \leq m \end{array} \right\}$$

- Problems with size < 9 bits can be solved in a reasonable time.
  - 8-bit: 56 threads, SAT:200-300 sec, UNSAT: 1 day

Boolean Equations for the CNOT circuit problem

$$EQN_b = \left\{ \begin{array}{c} b_{ij_1} b_{ij_2} = 0, \\ b_{i1} + b_{i2} + \cdots + b_{in} + 1 = 0, \\ \text{for } 1 \le i \le k, 1 \le j_1 \ne j_2 \le n \end{array} \right\} \quad EQN_c = \left\{ \begin{array}{c} c_{ij_1} c_{ij_2} = 0, \\ c_{i1} + c_{i2} + \cdots + c_{in} + 1 = 0, \\ \text{for } 1 \le i \le k, 1 \le j_1 \ne j_2 \le n \end{array} \right\}$$

$$EQN_a = \left\{ \begin{array}{c} f_{i,j}(\psi_{k,j,s} + a_{is}) = 0 \\ \text{for } 1 \le i,j \le n, 1 \le s \le m \end{array} \right\} \quad EQN_f = \left\{ \begin{array}{c} f_{ij_1} f_{ij_2} = 0, \\ f_{i1} + f_{i2} + \cdots + f_{in} + 1 = 0, \\ \text{for } 1 \le i \le n, 1 \le j_1 \ne j_2 \le n \end{array} \right\}$$

$$EQN_\psi = \left\{ \begin{array}{c} \psi_{i,j,s} + \sum_{t=1}^{n} c_{ij} b_{it} \psi_{i-1,t,s} + \psi_{i-1,j,s} = 0, \\ \text{for } 1 \le i \le k, 1 \le j \le n, 1 \le s \le m \end{array} \right\}$$

- Problems with size < 9 bits can be solved in a reasonable time.
  - 8-bit: 56 threads, SAT:200-300 sec, UNSAT: 1 day

$\mathfrak{C}^0$-circuit and $\mathfrak{C}^*$-circuit

$\mathfrak{C}^0$-circuit of f: $|x\rangle_a |0\rangle_b |0\rangle_c \to |x\rangle_a |f(x)\rangle_b |0\rangle_c$.

$\mathfrak{C}^*$-circuit of f : $|x\rangle_a |y\rangle_b |0\rangle \to |x\rangle_a |y \oplus f(x)\rangle_b |0\rangle_c$

- A $\mathfrak{C}^*$-circuit is always a $\mathfrak{C}^0$-circuit.
- Building a $\mathfrak{C}^0$-circuit is much easier than building a $\mathfrak{C}^*$-circuit.
- Some circuits using the output wires as temporary storage space to save the cost of qubits, are $\mathfrak{C}^0$-circuits but not $\mathfrak{C}^*$-circuits.
  - AES S-box circuits proposed in [GLRS16,ASAM18,LPS19]

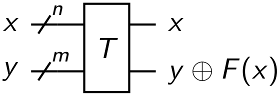
$\mathfrak{C}^0$-circuit and $\mathfrak{C}^*$-circuit

$\mathfrak{C}^0$-circuit of f: $|x\rangle_a |0\rangle_b |0\rangle_c \rightarrow |x\rangle_a |f(x)\rangle_b |0\rangle_c$.

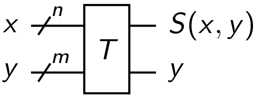
$\mathfrak{C}^*$-circuit of f : $|x\rangle_a |y\rangle_b |0\rangle \rightarrow |x\rangle_a |y \oplus f(x)\rangle_b |0\rangle_c$

- A $\mathfrak{C}^*$-circuit is always a $\mathfrak{C}^0$-circuit.
- Building a $\mathfrak{C}^0$-circuit is much easier than building a $\mathfrak{C}^*$-circuit.
- Some circuits using the output wires as temporary storage space to save the cost of qubits, are $\mathfrak{C}^0$-circuits but not $\mathfrak{C}^*$-circuits.
    - AES S-box circuits proposed in [GLRS16,ASAM18,LPS19]

Implementing nonlinear transformations in-place (I)



1) Feistel-like



2) Substitution-like

Figure: Two kinds of classical invertible nonlinear transformations

- **Feistel-like** (Fesitel cipher, NFSR, Key schedule).

$$T : (x, y) \to (x, y \oplus F(x))$$

To implement T in-place, we only need a $\mathfrak{C}^*$-circuit of F:
$|x\rangle\,|y\rangle \to |x\rangle\,|y \oplus F(x)\rangle$;

Implementing nonlinear transformations in-place (II)

- **Substitution-like** (S-box):

$$T : (x, y) \to (S(x, y), y)$$

can be implemented by the OP-based in-place circuit

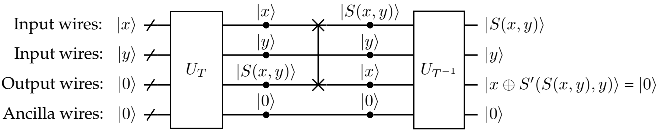


Figure: An OP-based in-place circuit for a substitution-like transformation. S': a function satisfying $S'(S(x, y), y) = x$.

1. $|x\rangle |y\rangle |0\rangle \to |x\rangle |y\rangle |S(x, y)\rangle$: $\mathfrak{C}^0$-circuit of S
2. $|S(x, y)\rangle |y\rangle |x\rangle \to |S(x, y)\rangle |y\rangle |x \oplus S'(S(x, y), y)\rangle$: we don't need a $\mathfrak{C}^*$-circuit of S'.

   $z = S(x, y)$, $|z\rangle |y\rangle |S'(z, y)\rangle \to |z\rangle |y\rangle |0\rangle$. Only need to design a $\mathfrak{C}^0$-circuit of S', and use its reverse circuit.

Implementing nonlinear transformations in-place (II)

- **Substitution-like** (S-box):

$$T : (x, y) \to (S(x, y), y)$$

can be implemented by the OP-based in-place circuit

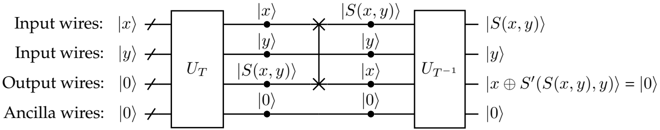


Figure: An OP-based in-place circuit for a substitution-like transformation. S': a function satisfying $S'(S(x, y), y) = x$.

1. $|x\rangle\,|y\rangle\,|0\rangle \to |x\rangle\,|y\rangle\,|S(x, y)\rangle$: $\mathfrak{C}^0$-circuit of S
2. $|S(x, y)\rangle\,|y\rangle\,|x\rangle \to |S(x, y)\rangle\,|y\rangle\,|x \oplus S'(S(x, y), y)\rangle$: we don't need a $\mathfrak{C}^*$-circuit of S'.

   $z = S(x, y)$, $|z\rangle\,|y\rangle\,|S'(z, y)\rangle \to |z\rangle\,|y\rangle\,|0\rangle$. Only need to design a $\mathfrak{C}^0$-circuit of S', and use its reverse circuit.

Constructing a $\mathfrak{C}^*$-circuit from a $\mathfrak{C}^0$-circuit

- Some criteria for efficiently designing $\mathfrak{C}^*$-circuits are proposed
- Under these criteria, a $\mathfrak{C}^*$-circuit of f can be constructed from a special $\mathfrak{C}^0$-circuit called **Simplex $\mathfrak{C}^0$-circuit**:

$$|x\rangle_a |y\rangle_b |0\rangle_c \to |x\rangle_a |A(y) \oplus f(x)\rangle_b |0\rangle_c\,, \quad A: \text{ a linear function}$$

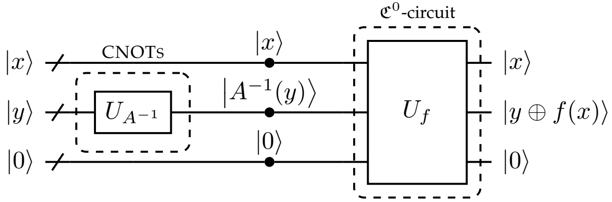


Figure: A $\mathfrak{C}^*$-circuit based on a simplex $\mathfrak{C}^0$-circuit

- $U_{A^{-1}}$: a CNOT sub-circuit; in most times uses ≤ 8 qubits.

Application in AES Key Schedule

- We can construct a $\mathfrak{C}^*$-circuit of AES S-box (used in the key schedule) from the $\mathfrak{C}^0$-circuit proposed in previous works without increasing #qubit and #Toffoli

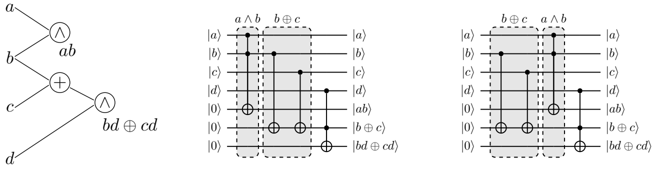
Table: Quantum resources for implementing the S-box of AES

	#ancilla	Toffoli-depth	#Toffoli	#CNOT	#NOT	source
$\mathfrak{C}^0$-S-box	6	41	52	326	4	Asiacrypt2020
$\mathfrak{C}^*$-S-box	7	60	68	352	4	Asiacrypt2020
	6	41	52	336	4	This paper

Outline

Classical AND-depth v.s. Quantum T-depth

- T gates only appears in the Toffoli gates, quantum AND gates (Eurocrypt 2020, a $\mathfrak{C}^0$ circuit of AND), and their adjoint (all have **T-depth-1 implementations**).
- Classical AND-depth = Quantum T-depth ?



(1) classical circuit (2) T-depth-2 (3) T-depth-1

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

- AES S-box circuit in Eurocrypt 2020: AND-depth 4, but T-depth 6.

The Lowest-T-depth Circuit

Theorem

Given a classical circuit with **AND-depth s**, the T-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 **T-depth s** .

- Based on Boyar and Peralta's classical circuit for AES S-box (AND-depth-4), we construct a T-depth-4 quantum circuit for AES S-box.
- We construct a new improved classical circuit for AES S-box (AND-depth-3), and induce a T-depth-3 quantum circuit for AES S-box.
- AES S-box has algebraic degree 7. Needs at least 3 multiplication layers, hence T-depth-3 is optimal.

The Lowest-T-depth Circuit

Theorem

Given a classical circuit with **AND-depth s**, the T-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 **T-depth s** .

- Based on Boyar and Peralta's classical circuit for AES S-box (AND-depth-4), we construct a T-depth-4 quantum circuit for AES S-box.
- We construct a new improved classical circuit for AES S-box (AND-depth-3), and induce a T-depth-3 quantum circuit for AES S-box.
- AES S-box has algebraic degree 7. Needs at least 3 multiplication layers, hence T-depth-3 is optimal.

Outline

Low-width Circuits for AES

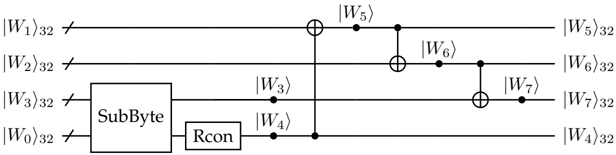


Figure: An in-place circuit for generating the first round key

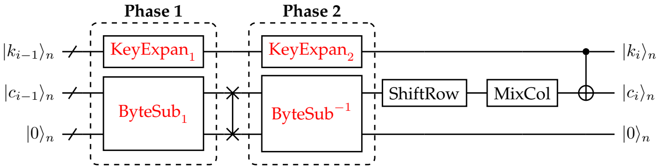


Figure: The in-place implementation of the i-th round of AES-128

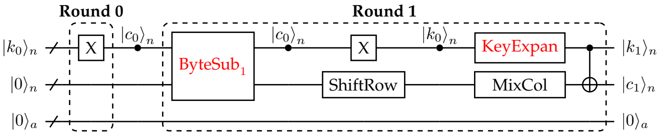


Figure: The implementation of the round 0 and round 1 of AES

Table: Quantum resources for implementing AES-128

Width	Toffoli-Depth	#Toffoli	#CNOT	#Pauli-X	source
512	2016	19788	128517	4528	Asiacrypt2020
492	820	17888	126016	2528	$p = 18$
374	1558	17888	126016	2528	$p = 9$

- p: number of S-boxes applied in parallel

Low-depth circuits for AES

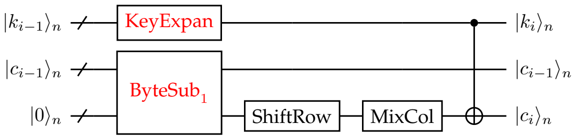


Figure: The out-of-place implementation of the i-th round of AES-128

Table: Quantum resources for implementing AES and AES†.

#CNOT	#1qClifford	$\#T$	#M	T-depth	Full depth	width	source
291150	83116	54400	13600	120 (60)	2827	3936	Eurocrypt 2020
298720	83295	54400	13600	80 (40)	2198	3936	with S-box$_3$
570785	189026	124800	31200	60 (30)	2312	5576	with S-box$_4$

- S-box$_4$: T-depth-4 S-box, S-box$_3$: T-depth-3 S-box
- ($*$) : for only implementing the forward circuit of AES

Tradeoff between Width and T-depth

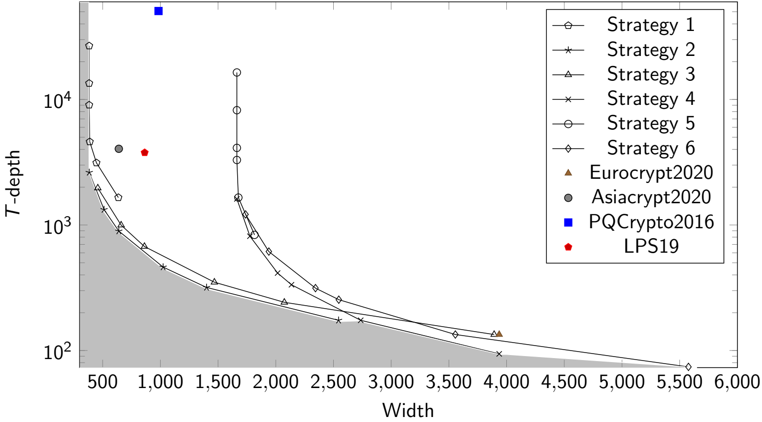


Figure: The width and T-depth for implementing the Grover oracle of AES-128

Thank you for your attention!

huangzhenyu@iie.ac.cn