Quantum Synthesis of Large S-Boxes: Heuristic and MILP-Based Transpiled-Depth Optimization

Tarun Yadav, Shweta Singh and Sudha Yadav

Scientific Analysis Group, DRDO, Metcalfe House Complex, Delhi-110 054, INDIA
{tarunyadav.sag,shweta.singh2416,sudhayadav.sag}@gov.in

Abstract. Quantum cryptanalysis of block ciphers with Grover's search requires synthesis of round function, where the non-linear S-boxes dominate the circuit cost. Efficient quantum implementations of these S-boxes are a bottleneck for cryptanalysis. In this work, we address this problem and present new generic strategy for synthesis of quantum circuit for large S-boxes that reduces the NISQ-era *transpiled depth* after decomposition into the hardware-oriented universal basis gate set `u+cx`. We introduce *two-phase MILP-based, ancilla-aware synthesis framework* for large S-boxes. Phase 1 determines which monomials will be synthesised globally, and how they are reused across outputs. This reduces redundancy and avoids high-degree terms that would lead to deep ladders. Phase 2 arranges the selected monomials into parallel layers. Here the solver explicitly accounts for ancilla usage, balancing the trade-off between fewer layers (smaller depth) and larger ancilla. MILP based synthesis show decisive, multi-fold reductions in transpiled depth in universal basis gate set `u+cx`. For SKINNY and ZUC S0 S-boxes, our synthesis reduces transpiled depth by factors of 18 and 9, respectively, with ancilla usage raised only to 10 and 13 qubits. For higher-degree S-boxes such as AES, SM4, and ZUC S1, we achieve 5 times reduction in transpiled depth by trading additional ancillas, increasing the budget from 5 to 22. To our knowledge, this is the first demonstration of ancilla-aware, globally optimised synthesis of 8-bit cryptographic S-boxes. By aligning primitive synthesis with transpiled cost, our method establishes a new baseline for depth-optimised resource estimation in the quantum cryptanalysis of symmetric primitives.

Keywords: Quantum circuits, Transpiled depth, S-box, MILP optimization

1 Introduction

Substitution boxes (S-boxes) are the core nonlinear components of symmetric ciphers such as AES, SM4, SKINNY, and ZUC. Their efficiency directly impacts both classical cryptanalytic resistance and quantum cost in Grover's search. Hence, low-cost quantum synthesis of S-boxes is crucial for estimating post-quantum security.

1.1 Small vs. Large S-boxes

The synthesis cost depends strongly on S-box size:

- **Small S-boxes (4-bit)**: Used in lightweight ciphers like PRESENT and GIFT [1][2]. They require only a few Toffoli/CNOT gates, leading to low transpiled depth. In practice, SWAP costs from linear layers often dominate.
- **Large S-boxes (8-bit)**: Found in AES, SM4, SKINNY, and ZUC. Their higher nonlinearity results in hundreds of Toffoli gates and deep circuits. After decomposition, depths expand to thousands of Clifford+T or `u+cx` gates, making them the bottleneck in quantum implementations.

1.2 Quantum Gates (Brief)

Reversible quantum circuits for S-boxes are built from:

- **X (NOT)**: Flips $|0\rangle$ to $|1\rangle$ and vice versa.
- **CNOT**: Flips target if control is $|1\rangle$; quantum XOR.
- **Toffoli (CCNOT)**: Flips target if both controls are $|1\rangle$; backbone of reversible synthesis.
- **MCX (multi-controlled X)**: Generalizes Toffoli to k controls, implementing Boolean monomials.
- **Hadamard (H)**: Creates superpositions, essential for Grover's algorithm.
- **Phase (S) and T gates**: Introduce phase shifts; T is expensive in fault-tolerant models.
- **Universal u gate**: General single-qubit rotation in Qiskit's `u+cx` basis.

1.3 Quantum Cost Metrics

Common metrics for evaluating quantum S-box synthesis include:

- **Qubit count**: Total logical qubits used (ancilla trade-offs matter).
- **Gate count**: Total operations in a chosen basis.
- **Primitive depth**: Sequential layers of multi-controlled gates before decomposition.
- **Transpiled depth**: Circuit depth after decomposition into Clifford+T or $u + cx$; most realistic measure of NISQ/fault-tolerant cost.

Among these, transpiled depth best reflects practical feasibility of Grover-based attacks, motivating synthesis strategies that explicitly minimize it.

2 Methodology

The synthesis of efficient quantum circuits for 8-bit S-boxes begins with their representation as Boolean functions. Each S-box output can be expressed as a polynomial over $\mathbb{F}_2$, where monomials correspond to logical ANDs of subsets of input variables and addition is realized as XOR. This representation, known as

the *algebraic normal form* (ANF), provides a canonical algebraic specification that is particularly suitable for reversible logic synthesis.

From a circuit perspective, the ANF serves as base since each linear term maps to a CNOT, each quadratic term to a Toffoli, and higher-degree terms to multi-controlled Toffolis requiring either ancilla qubits or decompositions into universal gate basis. Thus, the efficiency of a quantum S-box implementation depends on how these monomials are selected, shared across outputs, and scheduled with respect to ancilla usage.

2.1 Maslov–Miller–Dueck (MMD) Synthesis

The Maslov–Miller–Dueck (MMD) method [7] is a classical truth-table–driven approach for reversible logic synthesis. Unlike ANF-based strategies, MMD operates directly on the $2^n \times m$ truth table of an $n \times m$ S-box, systematically applying multi-controlled Toffoli (MCX) gates until the mapping reduces to the identity. This constructive procedure guarantees correctness without requiring persistent ancilla qubits, which explains its widespread adoption as a baseline in cryptanalytic resource estimates [13].

Algorithm. MMD proceeds as follows:

1. **Normalization:** Map the all-zero input to the all-zero output using NOT gates if needed.
2. **Row iteration:** For each input $i \in \{1, \ldots, 2^n - 1\}$ and each output bit j, if $T[i][j]$ differs from the identity mapping, apply an MCX gate with controls given by the prefix of row i and target j.
3. **Update:** Modify the truth table accordingly and continue until all rows are corrected.

Resource model. Each d-controlled Toffoli is decomposed into $(2d-3)$ standard Toffolis using a ladder construction with one scratch qubit. The primitive depth is estimated via bucketized ladder costs:

$$\mathrm{Depth}_{\mathrm{ladder}}(d) = \begin{cases} 1, & d \in \{1,2\}, \\ 3, & d \in \{3,4\}, \\ 5, & d \in \{5,6\}, \\ 7, & d = 7. \end{cases} \tag{1}$$

and the normalized circuit depth is obtained by replacing each d-controlled gate with its corresponding depth contribution:

$$\mathit{Depth}_N \;=\; \mathrm{Base} - \sum_{d \geq 3} n_d \;+\; \sum_{d \geq 3} n_d \cdot \mathrm{Depth}_{\mathrm{ladder}}(d). \tag{2}$$

Discussion. The appeal of MMD lies in its determinism, ancilla-freeness, and reproducibility. Variants based on Reed–Muller expansions or template simplification have yielded modest improvements in primitive gate counts, but they remain limited to local optimizations. The key drawback is that MMD introduces high-degree Toffolis whenever required, without exploiting algebraic structure or inter-output sharing. After decomposition, these gates inflate into long chains of Toffolis and CNOTs, leading to very large transpiled depths (on the order of 10^4–10^5). For this reason, while MMD serves as a clear and conservative baseline for benchmarking, it motivates the search for depth-oriented alternatives such as MILP synthesis.

2.2 Two-phase MILP: global, ancilla-aware synthesis

To capture global trade-offs between ancilla, monomial reuse, and depth, we design a two-phase synthesis framework based on mixed-integer linear programming (MILP). The objective is to minimize transpiled depth in the hardware-oriented `u+cx` basis while explicitly constraining ancilla usage.

Phase 1: Monomial selection. Each output S_j of an $n \times m$ S-box is written in ANF as

$$S_j(x) = \bigoplus_{u \in \{0,1\}^n} \lambda_{j,u} \chi_u(x), \quad \chi_u(x) = \prod_{i=1}^{n} x_i^{u_i}.$$

Binary variables z_u indicate whether a monomial χ_u is globally synthesized, and $x_{j,u}$ indicate whether it contributes to output j. Correctness is ensured by enforcing truth-table consistency:

$$\sum_{u:\, u \subseteq s} x_{j,u} \;=\; Y_{j,s} + 2\, k_{j,s}, \qquad \forall j, s, \tag{3}$$

where $Y_{j,s}$ is the output value at input s. Degree bounds $deg(u) \leq d_{\max}$ restrict the appearance of costly high-control Toffolis. The Phase 1 objective penalizes heavy or high-degree terms, encourages reuse across outputs, and promotes compact factorizations:

$$\min \; w_{\text{mono}} \sum_u c_u z_u + w_{\text{use}} \sum_{j,u} x_{j,u} + w_{\text{depth}}\, D_{\max} + w_{\text{deg5+}} \sum_{\deg(u) \geq 5} z_u, \tag{4}$$

with $c_u = \max(\deg(u) - 1, 0)$. Each term has a clear meaning:

- $\sum_u c_u z_u$: penalises the total "heaviness" of global monomials (encouraging reuse of small-degree terms).
- $\sum_{j,u} x_{j,u}$: penalises widespread usage (encouraging sparse factorizations).
- $D_{\max}$: penalises outputs that rely on many sequential monomials (encouraging shallow ANFs).
- $\sum_{\deg(u) \geq 5} z_u$: applies an extra penalty to degree-5 or higher monomials.

This phase thus selects a balanced, low-degree set of building blocks that preserves functional correctness.

Phase 2: Ancilla-aware layered scheduling. The selected monomials are assigned to sequential layers via binary variables $y_{u,\ell}$. Each monomial u of degree d requires $r(u) = \max(0, d-2)$ ancillas if retained. Constraints ensure that:

$$\sum_u r_u \, y_{u,\ell} \le A_\ell, \qquad \forall \ell, \tag{5}$$

$$A \ge A_\ell, \qquad \forall \ell. \tag{6}$$

so that the peak ancilla demand A is explicitly tracked. Target contention rules prevent multiple monomials from toggling the same output in one layer. The objective is

$$\min \; D_{\max} \; + \; w \cdot A. \tag{7}$$

where $D_{\max}$ is the number of layers and w is a tunable weight. This balances time (depth) against space (ancilla), yielding Pareto-efficient schedules.

Discussion. The MILP framework integrates algebraic factoring, ancilla budgeting, and parallel scheduling into a single optimization problem. In contrast to MMD, which introduces high-control Toffolis greedily, MILP exploits global structure to reduce bottlenecks and expose concurrency.

3 Results and Discussion

The synthesis methods described in Section 2 are implemented in Python, with `u+cx` decompositions handled through Qiskit's transpiler and verified against the truth-table specifications of each S-box. Our toolchain automatically extracts ANF representations, applies the chosen synthesis strategy, and compiles the resulting circuits to the same universal gate set `u+cx`. The complete source code and scripts are available on GitHub[1], ensuring all reported numbers are reproducible.

In this section we present a structured comparison of the three synthesis strategies. We begin with the baseline MMD synthesis (Section 3.1), which represents the canonical ANF-to-ladder construction used in prior resource estimates (e.g., Dutta et al. [13]). Then, we present the MILP-based synthesis (Section 3.2), which incorporates ancilla allocation and global scheduling into a single optimization framework. There are too many metrics to compare and therefore, we have restricted our results to the metrics which are significant in the cost computation. These metrics include ancilla count, primitive-level normalized depth and transpiled-level depth. A comparative discussion after these subsections analyzes the trade-offs and cryptanalytic implications across all methods.

3.1 MMD Based Synthesis

We first evaluate the baseline MMD construction on five representative 8-bit S-boxes: AES, SM4, SKINNY, and the two ZUC S-boxes (S0 and S1). Table 1 and

[1] `https://github.com/tarunyadav/QuantumSynthesis_LargeSBoxes`

Table 1: Primitive-level MMD synthesis results with full Toffoli breakdown (ancilla=5). MCX = total multi-controlled-X gates (sum of T3–T7). Toffoli$_N$ and Depth$_N$ denote normalized metrics using Dutta's $(2d-3)$ weighting for C^dX.

S-box	NOT	CNOT	Toffoli	T3	T4	T5	T6	T7	MCX	Depth$_P$	Toffoli$_N$	Depth$_N$
AES	4	123	206	244	205	123	42	7	621	945	3279	2545
SM4	5	104	198	220	214	136	53	9	632	931	3456	2609
SKINNY	4	95	173	212	205	134	52	8	611	877	3328	2503
ZUC S0 [13]	5	102	199	236	202	136	53	10	637	936	3456	2628
ZUC S1 [13]	4	110	211	221	189	132	62	9	613	930	3400	2580

Table 2: Transpiled-level MMD synthesis results (ancilla=5). Reported are normalized depth, size, and dominant decomposed gate counts.

S-box	Ancilla	Depth$_T$	u	cx
AES	5	54,124	46,377	29,907
SM4	5	58,989	51,001	32,900
SKINNY	5	57,081	49,244	31,777
ZUC S0	5	59,021	50,775	32,916
ZUC S1	5	58,916	50,692	33,022

2 summarize the primitive-level metrics with a full Toffoli breakdown and MCX totals, and the transpiled-level metrics after `u+cx` decomposition and mapping.

Analysis. The MMD baseline produces circuits with primitive depths in the range 877–945. The per-degree breakdown (Table 1) shows a heavy reliance on high-degree Toffoli gates: across all five S-boxes, MCX totals range from 611 (SKINNY) to 637 (ZUC S0), with many of T_6 and T_7 gates still present. These high-control constructs account for the majority of the normalized Toffoli costs (e.g., 3,456 for SM4 and ZUC S0) and inflate the normalized depth estimates.

After `u+cx` decomposition and transpilation (Table 2), the final depths explode to 5.4×10^4–5.9×10^4, and gate count exceeds 8×10^4. Two-qubit operations (`cx`) alone exceed 3×10^4 in each case. This blow-up confirms that while MMD provides a straightforward and general synthesis, it is pessimistic as a baseline because it directly maps monomials into long Toffoli ladders without factoring or scheduling.

3.2 MILP Based Synthesis

The Mixed-Integer Linear Programming (MILP) pipeline described in Section 2.2. The Phase 1 results (Depth$_\mathrm{P}$, Depth$_\mathrm{P}$ and Depth$_\mathrm{T}$) closely resemble those of the MMD with a few ancilla usages. However, after applying Phase 2, the transpiled depth drops by a large margin due to ancilla-depth tradeoff optimization (Table 3). This highlights that primitive depths (Depth$_P$ and Depth$_N$) are not a

Table 3: MILP Phase-2 results: peak ancilla (A) and transpiled depth (Depth_T) as a function of depth bound $d_{\max}$. Bold entries mark the minimum Depth_T and minimum *ancilla* per cipher. – represents not achieving full bound $d_{\max}$ and therefore, results are same as one of the previous row.

$d_{\max}$	AES		SM4		SKINNY		ZUC S0		ZUC S1	
	A	Depth_T	A	Depth_T	A	Depth_T	A	Depth_T	A	Depth_T
1	**516**	**11772**	**516**	**11779**	**119**	**3082**	**259**	**6428**	**513**	**11708**
2	258	11840	258	11850	60	3128	130	6493	257	11792
3	172	11876	172	11823	40	3119	87	6521	171	11804
4	129	11842	129	11854	30	3117	65	6531	129	11792
5	104	11842	104	11849	24	3104	52	6503	103	11792
6	86	11809	86	11861	20	3132	44	6476	86	11804
7	74	11827	74	11852	17	3125	37	6498	74	11787
8	65	11865	65	11878	15	3129	33	6535	65	11806
9	58	11845	58	11891	–	–	29	6516	57	11811
10	52	11859	52	11882	12	3137	26	6545	52	11832
11	47	11844	47	11855	11	3140	24	6516	47	11799
12	43	11860	43	11876	**10**	**3138**	22	6561	43	11830
13	40	11850	40	11862	–	–	20	6520	40	11807
14	37	11882	37	11886	–	–	19	6550	37	11789
15	35	11858	35	11873	–	–	18	6543	35	11778
16	33	11848	33	11879	–	–	17	6536	33	11809
17	31	11874	31	11880	–	–	16	6528	31	11837
18	29	11863	29	11878	–	–	15	6518	29	11834
19	28	11853	28	11832	–	–	14	6505	27	11837
20	26	11882	26	11885	–	–	**13**	**6561**	26	11818
21	25	11871	–	–	–	–	–	–	25	11770
22	24	11862	24	11899	–	–	–	–	–	–
23	–	–	–	–	–	–	–	–	23	11799
24	**22**	**11887**	**22**	**11902**	–	–	–	–	**22**	**11829**

reliable indicator of the true execution cost as circuits with similar Depth_P may have multifold difference in transpiled depth (Depth_T).

In Phase 2, $d_{\max}$ controls the maximum number of sequential layers and *ancilla* records the peak ancilla load. At small $d_{\max}$, the solver packs many monomials per layer: ancilla are large but depth is minimal. At large $d_{\max}$, monomials are spread across more layers and ancilla shrink, while transpiled depth increases only slightly and quickly saturates. Thus the Phase 2 sweep directly exposes the trade-off between workspace and wall-clock logical time. This is an essential contrast to MMD which either lack ancilla-awareness or use only local heuristics.

Comparison with MMD. MMD's transpiled depths (Table 2) are all around 5.4×10^4–5.9×10^4 with $A = 5$. MILP improves dramatically even at very low ancilla:

- AES: 54,124 (MMD) vs. 11,887 (MILP at $A = 22$),
- SM4: 58,989 (MMD) vs. 11,902 (MILP at $A = 22$),
- SKINNY: 57,081 (MMD) vs. 3,138 (MILP at $A = 10$),
- ZUC S0: 59,021 (MMD) vs. 6561 (MILP at $A = 13$),
- ZUC S1: 58,916 (MMD) vs. 11,829 (MILP at $A = 22$),

Thus even in the ancilla regime close to MMD, MILP provides an order of magnitude reduction in transpiled depth.

Why MILP outperforms. MILP's superiority arises from three interacting mechanisms.

- *Global monomial selection:* Phase 1 balances degree, reuse, and a depth proxy, eliminating long chains and redundant recomputation.
- *Ancilla-aware scheduling:* Phase 2 distributes heavy monomials across layers to reduce peak ancilla without introducing sequential bottlenecks, maintaining compiler-friendly DAG shapes.
- *Compiler synergy:* these globally balanced DAGs decompose into `u+cx` circuits with shorter ladders, more parallelism, and better cancellation opportunities.

Together, these yield circuits that are not only dramatically shallower than heuristic methods at the same ancilla, but also remain shallow even when ancilla budgets are tightened to realistic device scales. This defines a Pareto frontier of ancilla–depth trade-offs, producing circuits that are both depth-efficient and compiler-friendly.

4 Conclusion

This work addressed the NISQ-era depth-efficient synthesis of large cryptographic S-boxes, a central challenge for quantum implementations of block ciphers under Grover's algorithm. The classical MMD baseline, and the two-phase MILP formulation have beem evaluated.

Our results show that:

- **MMD synthesis** is deterministic and ancilla-free, but produces circuits with excessive transpiled depths ($\sim 10^5$), making it impractical as an implementation strategy though useful as a baseline.
- **MILP synthesis** achieves the best trade-offs by globally optimizing monomial selection and layered scheduling. It reduces transpiled depths by factors of 5–18 while requiring only 10–22 ancillas, thus defining the Pareto frontier of ancilla–depth trade-offs.

Overall, this is the first demonstration of ancilla-aware, MILP based globally optimized synthesis of 8-bit S-boxes that directly minimizes transpiled depth in the `u+cx` basis. The methodology not only establishes realistic resource estimates for quantum cryptanalysis of AES, SM4, SKINNY, and ZUC, but also provides a general framework adaptable to other non-linear components.

References

1. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, LNCS 4727, pp. 450–466, 2007. doi:10.1007/978-3-540-74735-2_31.
2. S. Banik, S.K. Pandey, T. Peyrin, Y. Sasaki, S.M. Sim, Y. Todo, "GIFT: A Small Present," in *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, LNCS 10529, pp. 321–345, 2017. doi:10.1007/978-3-319-66787-4_16
3. J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. Springer, 2002. doi:10.1007/978-3-662-04722-4.
4. A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization," in *ASIACRYPT*, LNCS 2248, pp. 239–254, 2001. doi:0.1007/3-540-45682-1_15
5. D. Canright, "A very compact S-Box for AES," in *CHES*, LNCS 3659, pp. 441–455, 2005. doi:10.1007/11545262_32
6. O. Golubitsky and D. Maslov, "A study of optimal 4-bit reversible Toffoli circuits and their synthesis," *IEEE Trans. Comput.*, vol. 61, no. 9, pp. 1341–1353, 2012.
7. D. Maslov, G. W. Dueck and D. M. Miller, "Toffoli network synthesis with templates," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, no. 6, pp. 807-817, June 2005, doi: 10.1109/TCAD.2005.847911.
8. D. Maslov, G. Dueck, and M. Miller, "Techniques for the synthesis of reversible Toffoli networks," *ACM Trans. Des. Autom. Electron. Syst.,* vol. 12, no. 4, 2007. doi:10.1145/1278349.1278355.
9. V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 6, pp. 710–722, 2003. doi: 10.1109/TCAD.2003.811448.
10. R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *46th ACM/IEEE Design Automation Conference*, pp. 270-275, 2009, doi: 10.1145/1629911.1629984
11. K. N. Patel, I. L. Markov, and J. P. Hayes, "Optimal synthesis of linear reversible circuits," *Quantum Information and Computation*, vol. 8, no. 3, pp. 282–294, 2008. doi:10.26421/QIC8.3-4-4
12. M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying Grover's algorithm to AES: quantum resource estimates," in *PQCrypto*, LNCS 9606, pp. 29–43, 2016. doi:10.1007/978-3-319-29360-8_3
13. S. Dutta, A. Ghatak, A. Chattopadhyay, and S. Maitra, "Quantum cryptanalysis of ZUC and related resource estimation," in *INDOCRYPT 2024*, LNCS 15495, pp. 329–355, 2025. doi:10.1007/978-3-031-80308-6_15
14. M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, 2013. doi:10.1109/TCAD.2013.2244643
15. P. Selinger, "Quantum circuits of T-depth one," *Phys. Rev. A*, vol. 87, no. 4, 042302, 2013. doi:10.1103/PhysRevA.87.042302
16. D. Maslov, "Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization," *Phys. Rev. A*, vol. 93, no. 2, 022311, 2016. doi:10.1103/PhysRevA.93.022311

17. C. Gidney, "Halving the cost of quantum addition," *Quantum*, vol. 2, p. 74, 2018. doi:10.48550/arXiv.1709.06648
18. S. Cuccaro, T. Draper, S. Kutin, and D. Moulton, "A new quantum ripple-carry addition circuit," arXiv:quant-ph/0410184, 2004. doi:10.48550/arXiv.quant-ph/0410184
19. M. Soeken, S. Frehse, R. Wille, R. Drechsler, "RevKit: A toolkit for reversible circuit design," in *Journal of Multiple-Valued Logic and Soft Computing*, 2010.
20. R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schoenberger, Y. Stade, and L. Burgholzer, "The MQT handbook: A summary of design automation tools and software for quantum computing," in *IEEE International Conference on Quantum Software (QSW)*, 2024. doi:10.1109/QSW62656.2024.00013
21. G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. Jose Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C. Chen, J. Chow, A. Córcoles-Gonzales, A. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. González, E. Torre, D. Ding, . . . ,C. Zoufal, "Qiskit: An open-source framework for quantum computing," Zenodo, 2019. doi:10.5281/zenodo.2562111
22. S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington and R. Duncan, "t|ket⟩: a retargetable compiler for NISQ devices," *Quantum Sci. Technol.*, vol. 6, no. 1, 014003, 2020. doi:10.1088/2058-9565/ab8e92
23. L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, "Report on Post-Quantum Cryptography," NISTIR 8105, 2016. Available at: `http://dx.doi.org/10.6028/NIST.IR.8105`
24. B. Langenberg, H. Pham, and R. Steinwandt, "Reducing the cost of implementing AES as a quantum circuit," in *in IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1-12, 2020, Art no. 2500112. doi: 10.1109/TQE.2020.2965697.
25. M. Zhang, T. Shi, W. Wu and H. Sui, "Optimized Quantum Circuit of AES with Interlacing-Uncompute Strategy," *in IEEE Transactions on Computers*, vol. 73, no. 11, pp. 2563-2575, 2024. doi: 10.1109/TC.2024.3449094
26. Q. Luo, Q. Li, X. Li, G. Yang, J. Shen, M. Zheng. (2023). "Quantum Implementation of SM4 Block Cipher with Less Qubits," in Research Square; 2023. doi:10.21203/rs.3.rs-3105531/v1.
27. T. Yadav, M. Kumar, A. Kumar, and S. K. Pal, "A Practical-Quantum Differential Attack on Block Ciphers," *Cryptography and Communications*, vol. 17, pp. 337–357, 2025. doi:10.1007/s12095-023-00650-6.
28. K. Jang, G. Song, H. Kim, H. Kwon, H. Kim, and H. Seo, "Efficient Implementation of PRESENT and GIFT on Quantum Computers," *Applied Sciences*, vol. 11, no. 11, p. 4776, 2021. doi:10.3390/app11114776.