Cybersecurity

Research Open Access

Quantum key recovery attack on SIMON32/64

Hui Liu[1,2] and Li Yang[1,2*]

Abstract

The quantum security of lightweight block ciphers is receiving more and more attention. However, the existing quantum attacks on lightweight block ciphers only focused on the quantum exhaustive search, while the quantum attacks combined with classical cryptanalysis methods haven't been well studied. In this paper, we study quantum key recovery attack on SIMON32/64 using Quantum Amplitude Amplification algorithm in Q1 model. At first, we reanalyze the quantum circuit complexity of quantum exhaustive search on SIMON32/64. We estimate the Clifford gates count more accurately and reduce the T gate count. Also, the T-depth and full depth is reduced due to our minor modifications. Then, using four differentials given by Biryukov in FSE 2014 as our distinguisher, we give our quantum key recovery attack on 19-round SIMON32/64. We treat the two phases of key recovery attack as two QAA instances separately, and the first QAA instance consists of four sub-QAA instances. Then, we design the quantum circuit of these two QAA instances and estimate their corresponding quantum circuit complexity. We conclude that the quantum circuit of our quantum key recovery attack is lower than quantum exhaustive search. Our work firstly studies the quantum dedicated attack on SIMON32/64. And this is the first work to study the complexity of quantum dedicated attacks from the perspective of quantum circuit complexity, which is a more fine-grained analysis of quantum dedicated attacks' complexity.

Keywords: Quantum cryptanalysis, Lightweight block ciphers, Quantum amplitude amplification, Differential cryptanalysis, Key recovery attack, SIMON32/64

Introduction

The devolvement of quantum computation poses a threat to classical cryptosystems. Shor's algorithm (Shor 1994) can break the security of public-key cryptosystems based on integer factorization and discrete logarithm, which gives rise to post-quantum cryptography. As for the symmetric cryptosystems, before Simon's algorithm (Simon 1997) is applied in quantum cryptanalysis, there is only Grover's algorithm (Grover 1997) that helps get a quadratic speed-up.

Quantum cryptanalysis against block ciphers receives much attention in recent years. Following the notions for PRF security in quantum setting proposed by Zhandry et al. (Zhandry 2012), there are two security models in quantum cryptanalysis against block ciphers, called Q1 model and Q2 model by Kaplan et al. in (Kaplan et al. 2016b).

Q1 model: The adversary is only allowed to make classical online queries and do quantum offline computation.

Q2 model: The adversary is allowed to do offline quantum computation and make online quantum superposition queries. That is, the adversary could query in a superposition state to the oracle and get a superposition state as a query result.

We can observe that Q1 model is more realistic than Q2 model for the reason that it's up to the oracle whether to allow superposition access. However, it's still meaningful to study Q2 model to prepare for the future with highly developed quantum communication technology.

*Correspondence: yangli@iie.ac.cn
[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, 100093 Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, 100049 Beijing, China

In fact, quantum cryptanalysis in Q2 model has been going on for a long time. In 2010, Kuwakado and Morii constructed a quantum distinguisher on 3-round Feistel structure (Kuwakado and Morii 2010) using Simon's algorithm in Q2 model. Then they also recovered the key of Even-Mansour also using Simon's algorithm in 2012(Kuwakado and Morii 2012). At Crypto2016, Kaplan et al. extended the result in (Kuwakado and Morii 2010; 2012) and applied Simon's algorithm to attack a series of encryption modes and authenticated encryption such as CBC-MAC, PMAC, OCB (Kaplan et al. 2016a). In Q2 model, Simon's algorithm can be combined with Grover's algorithm to apply in quantum cryptanalysis against block ciphers. Leander and May (2017) firstly used this idea to attack FX-construction in Q2 model. Inspired by this work, Dong et al. (2020a) gave a quantum key recovery attack on full-round GOST also in Q2 model. Besides, Bernstein-Vazarani (BV) algorithm (Bernstein and Vazirani 1997) can also be applied in quantum cryptanalysis. Li and Yang (2015) proposed two methods to execute quantum differential cryptanalysis based on BV algorithm. Then, Xie and Yang extended the result in Li and Yang (2015) and present several new methods to attack block ciphers using BV algorithm (Xie and Yang 2019).

In Q1 model, it seems as if quantum cryptanalysis becomes less powerful. The most trivial quantum attack is quantum exhaustive search that defines the general security of block ciphers in quantum setting. Grassl et al. present quantum circuits to implement an exhaustive key search on AES and estimate quantum resources in Q1 model (Grassl et al. 2016). After that, there are also some other results exploring the quantum circuit design of AES (Almazrooie et al. 2018; Jaques et al. 2020; Zou et al. 2020; Langenberg et al. 2020). Besides, there are many attempts of quantum dedicated attacks combined with classical cryptanalysis methods, e.g. differential and linear cryptanalysis (Kaplan et al. 2016b), meet-in-the-middle attacks (Hosoyamada and Sasaki 2018; Bonnetain et al. 2019), and rebound attacks (Hosoyamada and Sasaki 2020; Dong et al. 2020b).

The research of lightweight block ciphers has received much attention in a decade. Several lightweight primitives have been proposed by the researchers, to just name some, SIMON (Beaulieu et al. 2015), SPECK (Beaulieu et al. 2015), SKINNY (Beierle et al. 2016), PRESENT(Bogdanov et al. 2007). To prepare for the future with large-scale quantum computers, it's necessary to study the quantum security of lightweight block ciphers. There are several attempts to study the quantum generic attacks on some lightweight block ciphers (Anand et al. 2020c; Jang et al. 2020; Anand et al. 2020b). In this paper, we focus on the quantum security of SIMON. The family of SIMON algorithm (Beaulieu et al. 2015) is a lightweight block cipher proposed by NSA in 2013, which has outstanding hardware implementation performance. In classical setting, there have been many dedicated attacks aimed at SIMON. However, in quantum setting, the only quantum attack on SIMON is in Anand et al. (2020c) where Anand et al. present the quantum circuit of Grover's algorithm on SIMON variants and give corresponding quantum resources estimate, which is a quantum generic attack. To further explore the quantum security of SIMON, we need to study the dedicated quantum attacks of SIMON. Notably, when measuring the attack complexity, the existing quantum dedicated attacks all studied the encryption complexity, while we use the quantum circuit resources cost as a measure of complexity in our study for the first time.

Attack model We consider the chosen-plaintext attack to SIMON32/64 in Q1 model, where the adversary is allowed to make classical online queries of encryption oracle and can choose random message pairs with input differential Δx. To achieve such a attack, the adversary needs to implement transformation:

$$\begin{aligned}\sum_{i=1}^{q}|0\rangle|0\rangle|0\rangle &\to \sum_{i=1}^{q}|m_i\rangle|0\rangle|0\rangle\\ &\to \sum_{i=1}^{q}|m_i\rangle|E(m_i)\rangle|E(m_i\oplus\Delta x)\rangle\end{aligned}$$

when given q pairs of classical plaintext-ciphertext pair as input. We suppose this process is efficient. Thus we can ignore the quantum resources cost of this process.

Our contribution In this paper, we study the quantum key recovery attack on SIMON32/64 using Quantum amplitude Amplification(QAA) in Q1 model. Our contributions can be summarized in the following two aspects.

1 We reanalyze the quantum circuit complexity of quantum master-key search on SIMON32/64. On one hand, we give more accurate estimate result of Clifford gates count and reduced T gate count. We reduce the execution number of key expansion process, which brings down the number of NOT gates and CNOT gates. Besides, counting the Clifford gates decomposed by Toffoli gates into the total number of Clifford gates helped us give a more accurate estimate of Clifford gates count. And we reduce the number of T gates using the decomposition of multi-control NOT gates with ancilla qubits. On the other hand, we give a more thorough analysis of circuits' depth. The depth we foucs on here is the depth of such quantum circuits that only are composed of Clifford + T gates. We make some modifications to the code of implementing SIMON32/64, which reduces the T-depth and full depth of circuits. Compared to (Anand

et al. 2020c), we give a more accurate and thorough complexity analysis of $\mathcal{QMKS}$'s quantum circuit.

2 We present our quantum round-key recovery attack on 19-round SIMON32/64 combined with $\mathcal{CRKR}$ in (Biryukov et al. 2014). We treat the partial key guessing phase and exhaustive search phase as two QAA instances separately and design the corresponding quantum circuit. The first QAA instance includes four sub-QAA instances corresponding to the four processes of key recovery using four differentials. Then we estimate the comlexity of our quantum circuits. At last, we make a a simple comparison among $\mathcal{QMKS}$, $\mathcal{QRKR}$ and $\mathcal{CRKR}$. We conclude that the encryption complexity is lowest among these three attacks and the quantum circuit complexity of $\mathcal{QRKR}$ is lower than $\mathcal{QMKS}$. That is, we give a quantum dedicated attacks on 19-round SIMON32/64 that has lower complexity than quantum generic attack both in terms of encryption complexity and quantum circuit complexity. Different from the former quantum dedicated attacks that only focused on encryption complexity, our work takes the first step of studying the quantum cirucuit complexity of quantum dedicated attacks.

Outline The rest of the paper is organized as follows. In "Preliminaries" section, we introduce the notations used in this paper and the background knowledge of SIMON block cipher, QAA algorithm and quantum circuit. In "The quantum master-key exhaustive search attack on 19-round SIMON32/64" section, we reanalyze the quantum circuit complexity of quantum exhaustive search attack on SIMON32/64. In "The quantum round-key key recovery attack on 19-round SIMON32/64" section, we describe the quantum round-key key recovery attack on 19-round SIMON32/64. In "The complexity analysis" section, we compare the complexity of our attack, quantum master-key search attack and classical differential attack. In "Conclusion" section, we make a summary of this paper.

Preliminaries

Notations

In this section, we list the notations used in this paper in Table 1.

Table 1 Notations

Notation	Description
&	The bitwise AND operation
$\oplus$	The bitwise XOR operation
$\lll$	The cyclic left rotation operation
Round-i	The i-th round of SIMON32/64
(L^{i-1}, R^{i-1})	The input block of Round-i in SIMON32/64
$L^i[j]$	The j-th bit of L^i(the index of rightmost bit is 0)
K^{i-1}	The round key of Round-i in SIMON32/64
$\Delta^{i-1} = (\Delta L^{i-1}, \Delta R^{i-1})$	The input difference to Round-i
ΔAnd^i	$\Delta And^i := (L^i \lll 1)\&(L^i \lll 8) \oplus ((L^i)' \lll 1)\&((L^i)' \lll 8)$
ΔRot^i	$\Delta Rot^i := \Delta L^i \lll 2$
$E(\cdot)$	The encryption function of 19-round SIMON32/64 with real key k
$E_k(\cdot)$	The encryption function of 19-round SIMON32/64 with guessed key k
$D_k^j(\cdot)$	The decryption function that decrypts the given ciphertext in j rounds with key k
$\mathcal{QMKS}$	The quantum master key exhaustive search attack on 19-round SIMON32/64
$\mathcal{QRKR}$	The quantum round-key key recovery attack on 19-round SIMON32/64
$\mathcal{CRKR}$	The key recovery attack on 19-round SIMON32/64 present in (Biryukov et al. 2014)
#iter	The number of iteration in a QAA instance
#Toff-C	The number of CNOT gate decomposed by Toffoli gate
#Toff-H	The number of H gate decomposed by Toffoli gate

Brief Description of SIMON

In this section, we describe SIMON briefly. SIMON is a Feistel structure lightweight block cipher. There are many SIMON variants to adapt to different computing scenarios, the differences between which lie at block size, key size, word size and round number. The block size of SIMON is $2n$ bits while the key size is mn bits. We could use SIMON$2n/mn$ to denote all SIMON variants, where $n \in \{16, 24, 32, 48, 64\}$ and $m \in \{2, 3, 4\}$. All the SIMON variants are summarized in Table 2.

Round function The i-th iteration structure of SIMON$2n/mn$ is shown in Fig. 1. We can easily see that the round function of SIMON$2n/mn$ consists of bit-wise AND, cyclic left rotation and bit-wise XOR. For SIMON, $f : \{0,1\}^n \rightarrow \{0,1\}^n$ is defined as $f(x) = (x \lll 1)\&(x \lll 8) \oplus (x \lll 2)$. The round function is defined as follows:

$$F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$$
$$F(L^{i+1}, R^{i+1}) = (R^i \oplus f(L^i) \oplus K^i, L^i)$$

Key schedule For r-round SIMON$2n/mn$, the round key SIMON is derived from primary key $\{K^0, K^1, \cdots, K^{m-1}\}$. The specific key expansion scheme is defined as:

1. When $i = 0, 1, \cdots, m-1$, $K^i = K^i$;

Table 2 All SIMON variants

Block Size($2n$)	Key Size($k = mn$)	Word Size(n)	Key Words(m)	Rounds(T)
32	64	16	4	32
48	72,96	24	3,4	36,36
64	96,128	32	3,4	42,44
128	128,192,256	64	2,3,4	68,69,72

2. When $i = m, m+1, \cdots, r-1$, $K^i = c\oplus(z_j)^{i-m}\oplus K^{(i-m)}\oplus K^{(i-m+1)}\oplus(K^{(i-m+1)} \lll 15)\oplus(K^{(i-m+3)} \lll 13)\oplus(K^{(i-m+3)} \lll 12)$;

z_j is a constant sequence and $c = 2^n - 4$. The key schedule is linear. Thus we can derive the master key from any mn independent bits of subkeys. Particularly, for SIMON32/64, as long as we get the round keys of any four adjacent rounds, the master key can be easily deduced.

Related works In classical setting, there already have been some attack results on SIMON. We make a simple summary of some attacks on SIMON32/64 in Table 3. However, in quantum setting, the only quantum attack on SIMON is the quantum exhaustive search in Anand et al. (2020c). To furthur explore the quantum security of SIMON block cipher, we study the quantum dedicated attack on SIMON32/64 in this paper. According to the analysis in "The complexity analysis", we also list the complexity of quantum generic attack and our quantum dedicated attack in Table 3 for comparison.

Brief Description of QAA algorithm

In this section, we describe QAA algorithm briefly. QAA algorithm is a natural generalization of Grover's algorithm that searches all solutions in an unsorted database. Compared to classical algorithm, QAA algorithm can achieve quadratic speed-up. According to (Brassard et al. 2002), QAA algorithm can be summarized in the following theorem.

Theorem 1 *Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $g : \{0,1\}^n \rightarrow \{0,1\}$ be any Boolean function. Let p be the initial success probability of $\mathcal{A}$. Suppose $p > 0$ and set $m = \lfloor \frac{\pi}{4\theta} \rfloor$, where $\sin^2(\theta) = p$. We define $G = U_s U_g = -\mathcal{A}S_0\mathcal{A}^{-1}U_g$, where $S_0 = 2|0\rangle\langle 0| - I$. If we compute $G^m\mathcal{A}|0\rangle$ and measure the system, the outcome is good with probability at least max$(1-p, p)$.*

The quantum circuit for QAA algorithm is displayed in Fig. 2. For simplicity, we call a search problem using QAA algorithm to settle as a QAA instance. Every iteration of a QAA instance is called QAA iteration. For a QAA instance with M solutions in N elements, we define elements that are solutions as GOOD while the elements that are not solutions as BAD. We define a function $g : \{0,1\}^n \rightarrow \{0,1\}$

$$g(x) = \begin{cases} 1, \text{if } x \text{ is GOOD} \\ 0, \text{if } x \text{ is BAD} \end{cases}$$

Based on function g, we construct an oracle U_g, which is defined as

$$U_g|x\rangle = \begin{cases} -|x\rangle, \text{if } x \text{ is GOOD} \\ |x\rangle, \text{if } x \text{ is BAD} \end{cases}$$

The process of QAA is described as follows:

1 Apply $\mathcal{A}$ on the initial state $|\psi\rangle = |0\rangle$, we can get $|\psi\rangle = \mathcal{A}|0\rangle = |GOOD\rangle + |BAD\rangle$.
2 Call QAA iteration $m = \lfloor \frac{\pi}{4\theta} \rfloor$ times. In each iteration, there are two steps. The first step is to apply U_g to quantum state, after which we can get $U_g|\psi\rangle = -|GOOD\rangle + |BAD\rangle$. The second step is to apply diffusion operator $2|s\rangle\langle s| - I$ to $|\psi\rangle$, where $|s\rangle$ is the equal superposition of all elements.
3 Measure the first register and obtain one of all solutions.

We can observe that compared to the original Grover's algorithm, the operator H is replaced by a random unitary operator $\mathcal{A}$. We must carry out plenty of measurements to

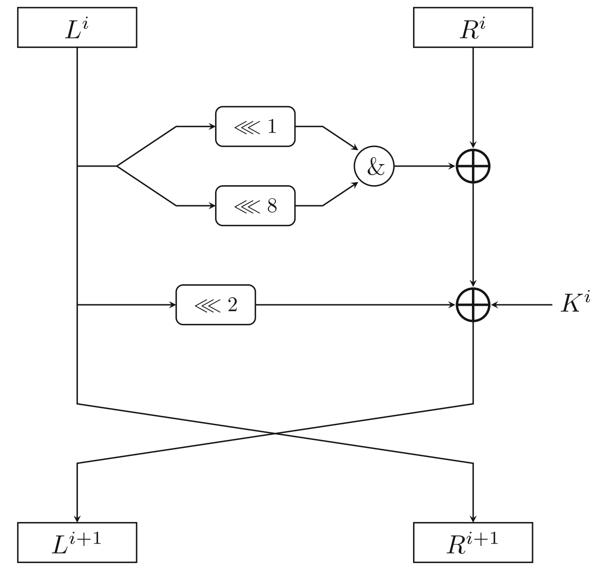

Fig. 1 Round function of SIMON

Table 3 Summary of some attacks on SIMON32/64

Cipher	Attacked rounds	Technique	Time	Data	Memory(bytes)	Reference
SIMON32/64	18	Differential	2^{46}	$2^{31.2}$	2^{15}	(Abed et al. 2014)
SIMON32/64	19	Differential	2^{34}	2^{31}	-	(Biryukov et al. 2014)
SIMON32/64	21	Differential	$2^{55.25}$	2^{31}	-	(Wang et al. 2018)
SIMON32/64	21	Zero-correlation	$2^{59.4}$	2^{32}	2^{31}	(Sun et al. 2015)
SIMON32/64	21	Integral	2^{63}	2^{31}	2^{54}	(Wang et al. 2014)
SIMON32/64	21	Linear	$2^{60.99}$	$2^{28.99}$	-	(Shi et al. 2017)
SIMON32/64	23	Linear	$2^{56.3}$	$2^{31.19}$	-	(Chen and Wang 2016)
SIMON32/64	24	Integral	2^{63}	2^{32}	$2^{33.64}$	(Chu et al. 2018)
SIMON32/64	19	Quantum generic	$2^{33.5}$	3	-	This paper
SIMON32/64	19	Quantum dedicated	$2^{31.4}$	2^{30}	-	This paper

get all solutions because the output of QAA algorithm is the superposition of M solutions.

Quantum circuit

In this section, we introduce the related knowledge of quantum circuits briefly. Quantum logic gates are the foundation of quantum circuits. A quantum circuit can be seen as a sequence of quantum logic gates. In order to measure the complexity of a quantum circuit, we should consider the number of gates, and the number of qubits and the depth. When computating the depth of a quantum circuit, we also adopt the *full parrellism assumption* as in Jaques et al. (2020), which means a quantum circuit can apply any number of gates simultaneously so long as these gates act on disjoint sets of qubits.

The Clifford + T gate set form a set of universal quantum gates. The Clifford group is defined as the group of unitary operators that map the group of Pauli operators to itself under conjugation. The Clifford gates are then defined as elements in the Clifford group. The basic Clifford gates includes H gate, S gate and CNOT gate. However, we cannot achieve universal quantum computation only with Clifford gates. This is, non-Clifford gate should be added into the gate set. And T gate is ususlly the choice to be added in. The matrix representations of Clifford + T gate set in shown in Eq.(1).

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$
$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \; T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \tag{1}$$

According to (Amy et al. 2013), all Clifford group operations have transversal implementations and thus are relatively simple to implement while non-Clifford gates require much more sophisticated and costly techniques to implement. The surface codes, which promise higher thresholds than concatenated code schemes, also have a significantly more complicated T gate implementation than any of the Clifford group generators. As a result, it's significant to study the number of T gate in a quantum circuit in order to measure the complexity of quantum computation. Besides, Amy et al. proposed T-depth as a cost function of quantum circuits in Amy et al. (2013). We can observe that the research on reducing the T depth of quantum circuits has been paid more and more attention.

In classical computation, the Toffoli gate is a universal classical reversible logic gate, while for quantum computation it needs to be decomposed into Clifford + T gates for real implementation. According to (Nielsen and Chuang 2001), the decomposition of Toffoli gate is shown in Fig. 3.

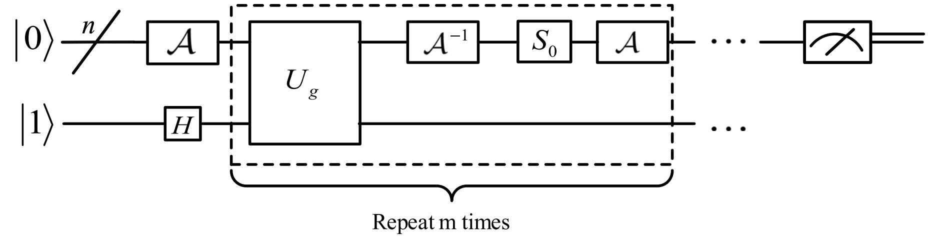

Fig. 2 Quantum circuit of QAA algorithm

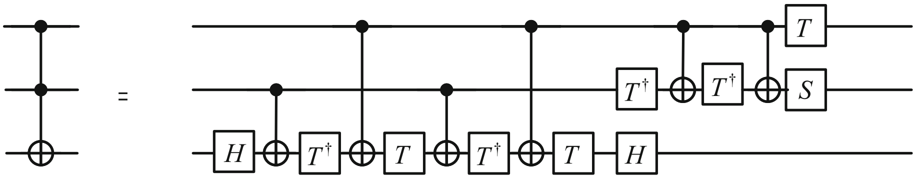

Fig. 3 The decomposition of Toffoli gate in (Nielsen and Chuang 2001)

That is, a Toffoli gate can be decomposed into 7 T gates, 6 CNOT gates, 2 H gates and 1 S gate with T-depth 7 and full depth 13. Then, to reduce T-depth, Amy et al. proposed a decomposition scheme of Toffoli gate in Amy et al. (2013) with T-depth 3 and full depth 10, shown in Fig. 4. And Amy et al. conjectured that this T-depth is optimal for circuits without ancillas. Although T-depth could be reduced to 1 further with ancilla qubits according to the Figure 1 in Selinger (2013), the number of CNOT gates increases much. After a overall consideration of gate counts and T-depth of quantum circuits, we adpot the method in Fig. 4 to decompose Toffoli gate in this paper.

In QAA iterator G, there two multi controlled-NOT gates. For the real implementation of QAA algorithm, we need to decompose the mutli controlled-NOT gate into a series of Toffoli gates. Then we need to decompose the Toffoli gate into Clifford + T gates. According to (Nielsen and Chuang 2001), the n-fold controlled-NOT could be decomposed into $2n - 3$ Toffoli gates using $n - 2$ ancilla qubits. We show the decomposition of n-fold controlled-NOT in Fig. 5. Here, we offer a concept, Toffoli-depth, which is similar to T-depth, meaning the number of stages in the circuit involving Toffoli gates. In our analysis, computing the Toffoli-depth is the first step to compute the T-depth and full depth of quantum circuits. We can observe that the Toffoli-depth of Fig. 5 is $2n-3$. Thus the full depth of implementing a n-fold controlled-NOT is $20n-30$,and the T-depth is $6n - 9$. It is worth noting that the depth we're talking about refers to the depth of the quantum circuits only containing Clifford gates and T gates. This is, we need to decompose all Toffoli gates into Clifford + T gates before computing the depth of quantum circuits.

The quantum master-key exhaustive search attack on 19-round SIMON32/64

In this section, to put the comparison standard on the same scale, we reanalyze the quantum circuit complexity of $\mathcal{QMKS}$ using QAA algorithm based on the result in Anand et al. (2020c) where Anand et al. present Grover's search algorithm on SIMON variants and estimate the quantum resources to implement such attack.

At first, we present the quantum circuit complexity of implementing 19-round SIMON32/64. From Table 3 in Anand et al. (2020c), we can easily derive the gate count of implementing 19-round SIMON32/64. However, when computing the circuit depth, we got different results from (Anand et al. 2020c). Anand et al. implemented all SIMON variants in QISKIT(Koch et al. 2019). The circuit depth can be calculated using the Qiskit function. After running the code of implementing SIMON32/64 given by Anand et al. (2020c) in Anand et al. (2020a), we found that the Qiskit function computes the the depth of quantum circuit without decomposing Toffoli gate which leads to the incompleteness of the circuit depth calculation. In our estimate, Toffoli gates should be decomposed into Clifford + T gates before computing the circuit depth. Besides, we made some small modifications to the code of implementing SIMON32/64, which brought in reduction of full depth and T-depth. We performed one operation on all bits firstly, and then performed the next operation on all bits, instead of performing all operations on each bit one by one in our modifications. We gave our modified code in (Lau I 2021). We list the quantum circuit complexity of implementing SIMON32/64 in Table. 4.

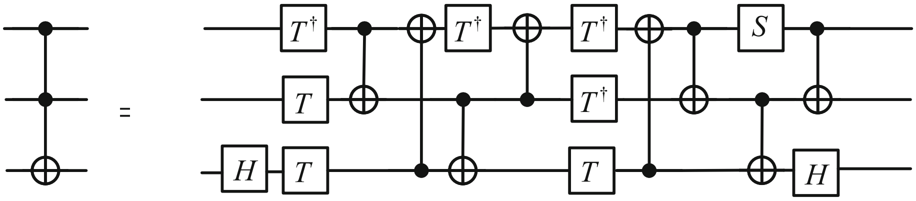

Fig. 4 The decomposition of Toffoli gate in (Amy et al. 2013)

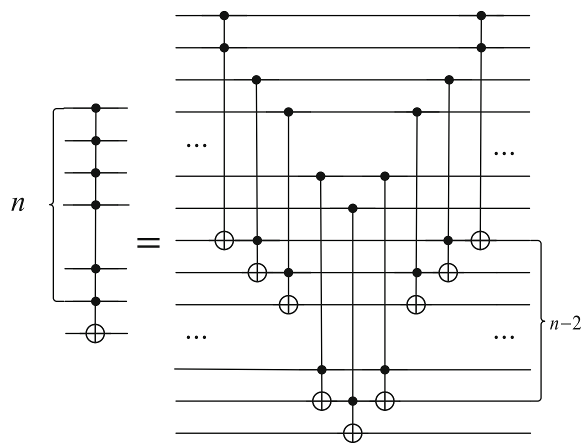

Fig. 5 The decomposition of *n*-fold controlled-NOT

Then we reanalyze the quantum circuit complexity of $\mathcal{QMKS}$'s quantum circuit, shown in Fig. 6. To implement the circuit in Fig. 6, we need to implement the QAA iterator $G = U_sU_g$. The implementation of U_g is in Fig. 7, in which 3 plaintext-ciphertext pairs are chosen for the uniqueness of solution. The operator U_s consists of two 64-fold Hardmard gates and one 64-fold controlled-NOT gate. Here, we reanlyze the quantum circuit complexity of quantum exhaustive search on SIMON32/64 from the following three points.

1 It is enough to perform key expansion in U_g twice, one computation and one uncomputation. In Anand et al.'s estimate, six key expansion processes for six SIMON instances were performed separately in U_g, which made the number of NOT gates and CNOT gates were overestimated. There are 448 NOT gates and 1792 CNOT gates during a key expansion process. It's easy to derive that #NOT=$448 \times 2 = 896$. Besides, the CNOT gates come from two key expansion processes and implementation of six SIMON instances. That is, #CNOT=$28 \times 64 \times 2 + 32 \times 32 \times 6 = 9728$.

2 The Clifford gates decomposed by Toffoli gates should be taken into account in resources estimate. Anand et al. ignored the Clifford gates decomposed by Toffoli gates. The Toffoli gates of quantum circuit in Fig. 6 come from implementation of SIMON and the decomposition of two multi controlled-NOT gates. There are $512 \times 6 = 3072$ Toffoli gates in six SIMON instances. Besides, according to the decomposition of Toffoli gate in Fig. 5, 96-fold controlled-NOT gate in U_g and 64-fold controlled-NOT gate in U_s can be decomposed into $2 \times 96 - 3 + 2 \times 64 - 3 = 314$ Toffoli gates using 94 ancilla qubits at most. So we have #Toff-C=$(3072 + 314) \times 7 = 23702$, #Toff-H=$(3072 + 314) \times 2 = 6772$. Anand et al. adopted the result in (Roetteler and Wiebe 2016) to estimate the number of T gates while we use Fig. 5 to estimate the number of T gates, which reduces the number of T gates via increasing the number of qubits.

3 The circuit depth estimate result should be more thorough, and the T-depth and full depth of QAA iterator G could be reduced. We decompose the two multi-control NOT gates in operator G into Toffoli gates, and then decompose all Toffoli gates into Clifford + T gates. We consider the circuit depth of this circuit with only Clifford + T gates. Although there are six SIMON instances in U_g, since three SIMON instances are executed in parallel, we only need to consider the depth of two SIMON instances. However, we found that Anand et al. counted the depth of the six SIMON instances into the total depth of G in Anand et al. (2020c), which overestimated the full depth and T-depth of G. We estimated that the Toffoli-depth of QAA iterator G is 96. Then we can easily get the full depth and T-depth of G, as shown in the second line of Table 4. We can observe that our estimated depth are smaller than the results in Anand et al. (2020c). This is due to the slight modification we made to the circuit implementation of SIMON32/64. In addition, we didn't ignore the depth of implementing the two multi-control NOT gates, which makes our estimate more accurate and thorough.

Through the above analysis, we present our more accurate estimate results of QAA iterator G in Table 5. To find the master key in the key space $\{0,1\}^{64}$, we need to iterate QAA iterator $G = U_sU_g$ for $\lfloor \frac{\pi}{4} 2^{32} \rfloor$ times. From the result in Table 5, we can easily get the quantum circuit complexity of quantum exhaustive search on SIMON32/64 in Table 6. In our estimate results, the number of Clifford gates is a little higher than that in

Table 4 Cost of SIMON32/64

Round	#NOT	$\#CNOT_{sum}$		$\#H_{sum}$		#Cliff	#T	T-depth	Full depth	#qubit	Refer.
		#CNOT	#Toff-C	#H	#Toff-H						
32	448	2816	3072	0	1024	7360	3584	2048	946	96	(Anand et al. 2020c)
32	448	2816	3584	0	1024	7872	3584	288	1024	96	This paper
19	240	1568	2128	0	608	4544	2128	171	608	96	This paper

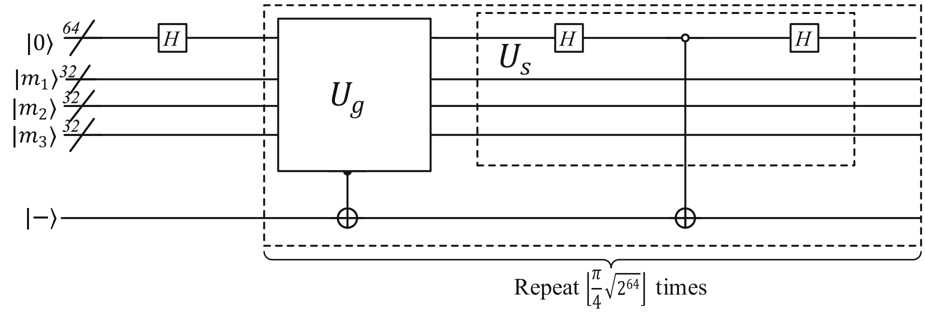

Fig. 6 The quantum circuit of $\mathcal{QMKS}$

Anand et al. (2020c) because we consider the number of Clifford gates decomposed by Toffoli gates. Besides, we reduce the number of T gates by adopting the decomposition of multi controlled-NOT gate, which also increases the number of qubits. Also we reduced the T-depth and full depth because of small modifications to the implementation of SIMON32/64. In summary, our estimate result is more accurate and detailed.

The quantum round-key key recovery attack on 19-round SIMON32/64

In this section, we describe the quantum round-key key recovery attack on 19-round SIMON32/64 and give the corresponding quantum circuit as well as its quantum resources estimate. At first, we recall the classical key recovery attack on 19-round SIMON32/64 in Biryukov et al. (2014) where Biryukov et al. present four 13-round differentials with which they recovered the round keys from Round-16 to Round-19. Then we use the four 13-round differentials in Biryukov et al. (2014) as our distinguisher and apply QAA algorithm into the two phases of key recovery attack on 19-round SIMON32/64. At last, we compare the complexity of our key recovery attack and exhaustive search on 19-round SIMON32/64 in terms of encryption complexity and quantum resources separately.

The classical key recovery attack on SIMON 32/64

In this section, we describe the key recovery attack in Biryukov et al. (2014).

At first, we list the four 13-round differentials given by Biryukov et al. as follows:

$$\mathcal{D}_1 : \Delta_{in}^1 = (0000, 0020), \Delta_{out}^1 = (2000, 0000)$$
$$\mathcal{D}_2 : \Delta_{in}^2 = (0000, 0040), \Delta_{out}^2 = (4000, 0000)$$
$$\mathcal{D}_3 : \Delta_{in}^3 = (0000, 0400), \Delta_{out}^3 = (0004, 0000)$$
$$\mathcal{D}_4 : \Delta_{in}^4 = (0000, 0800), \Delta_{out}^4 = (0008, 0000)$$

Then we add two rounds on the top and append four rounds on the bottom to carry out the key recovery attack on 19-round SIMON 32/64. The input truncated differential at the beginning of Rould-1 should be

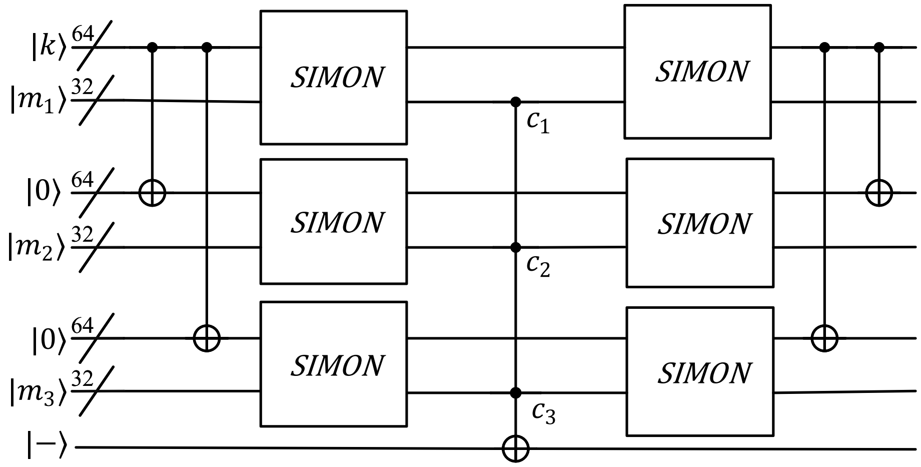

Fig. 7 The quantum circuit of U_g in Fig. 6

Table 5 The cost of QAA iterator $G = U_s U_g$ in Fig. 7

Round	#NOT	*#CNOT$_{sum}$*		*#H$_{sum}$*		#Cliff	#T	T-depth	Full depth	#qubit	Refer
		#CNOT	#Toff-C	#H	#Toff-H						
32	2688	17152	0	0	0	19840	24492	12288	27180	161	(Anand et al. 2020c)
32	896	9728	23702	128	6772	41226	23702	1230	5318	255	This paper
19	480	5568	14966	128	4276	25418	14966	1113	4434	255	This paper

$$\Delta x_1 = (\texttt{00}\star\texttt{0 0000 1}\star\texttt{00 0000, }\star\star\texttt{00 001}\star\ \star\texttt{0}\star\texttt{0 0000})$$

$$\Delta x_2 = (\texttt{0}\star\texttt{00 0001 }\star\texttt{000 0000, }\star\texttt{000 01}\star\star\texttt{ 0}\star\texttt{00 000}\star)$$

$$\Delta x_3 = (\texttt{0001 }\star\texttt{000 0000 0}\star\texttt{00, 01}\star\star\texttt{ 0}\star\texttt{00 000}\star\ \star\texttt{000})$$

$$\Delta x_4 = (\texttt{001}\star\texttt{ 0000 0000 }\star\texttt{000, 1}\star\star\texttt{0 }\star\texttt{000 00}\star\star\texttt{ 0000})$$

Then, we describe the process of key recovery process in Biryukov et al. (2014).

1 *Plaintexts Collecting*: Similar to (Biryukov et al. 2014), we construct a set $\mathcal{P}$ with 2^{23} plaintexts with 9 bits fixed. While different from (Biryukov et al. 2014), we just need one right pair. By varying some fixed bits of plaintexts in $\mathcal{P}$ and guessing 2 bits of the round key K^0, we can identify $2^{28.5}$ pairs which satisfy the input difference Δx_i to Round-3 for each $\mathcal{D}_i$ and for each guessed two bits of K^0. In total we can get a set with $2^{30.5}$ plaintext pairs for each $\mathcal{D}_i$ and there must be a right pair in this set.

2 *Filtering*: $2^{30.5}$ pairs of plaintexts is filtered by verifying the fixed 14 bits of the corresponding difference Δ^{18}. After filtering, the number of plaintext pairs can be reduced to $2^{30.5-18} = 2^{12.5}$ for each differential.

3 *Partial key guessing*: For each differential, we need to recover the following 25 key bits.

$$\mathcal{D}_1^K = \{K^{18}, K^{17}[3, 5-8, 12, 14], K^{16}[6] \oplus K^{17}[4], K^{16} \oplus K^{17}[2]\}$$

$$\mathcal{D}_2^K = \{K^{18}, K^{17}[4, 6-9, 13, 15], K^{16}[7] \oplus K^{17}[5], K^{16}[5] \oplus K^{17}[3]\}$$

$$\mathcal{D}_3^K = \{K^{18}, K^{17}[8, 10-13, 1, 3], K^{16}[11] \oplus K^{17}[9], K^{16}[9] \oplus K^{17}[7]\}$$

$$\mathcal{D}_4^K = \{K^{18}, K^{17}[9, 11-14, 2, 4], K^{16}[12] \oplus K^{17}[10], K^{16}[10] \oplus K^{17}[8]\}$$

The key recovery process of using four differentials is quite similiar. So we only describe the key recovery process using $\mathcal{D}_2$. We denote all the key bits in $\mathcal{D}_2^K$ by k_1 and denote the input ciphertext pair by $C = (L^{19}, R^{19}), C' = ((L^{19})', (R^{19})')$. The keys that satisfy Eq.(2) are called candidate keys.

$$D_{k_1}^4(C) \oplus D_{k_1}^4(C') = \Delta_{out}^2 \tag{2}$$

Eq.(2) holds with probability 2^{-14}, which means there are $2^{25} \times 2^{12.5}/2^{14} = 2^{23.5}$ plaintext-key pairs that satisfy Eq.(2). In expectation, we can get $2^{23.5}$ candidate keys for $\mathcal{D}_2^K$. Then we use the other three differentials to carry out the similiar key recovery process and can get $2^{23.5}$ candidate keys for $\mathcal{D}_1^K, \mathcal{D}_3^K, \mathcal{D}_4^K$ separately. Because there are some common bits among $\mathcal{D}_1^K, \mathcal{D}_2^K, \mathcal{D}_3^K, \mathcal{D}_4^K$, we can obtain $(2^{23.5})^4/(2^{19} \times 2^{20} \times 2^{22}) = 2^{33}$ candidate keys for 39 key bits in last 3 round-keys, i.e. $\mathcal{D}^c = \{K^{18}, K^{17}[1-15], K^{16}[6] \oplus K^{17}[4], K^{16} \oplus K^{17}[2], K^{17}[4, 6-9, 13, 15], K^{16}[7] \oplus K^{17}[5], K^{16}[5] \oplus K^{17}[3], K^{16}[11] \oplus K^{17}[9], K^{16}[9] \oplus K^{17}[7], K^{16}[12] \oplus K^{17}[10], K^{16}[10] \oplus K^{17}[8]\}$. For simplicity, we denote the 39 key bits by k_1'.

4 *Exhaustive search*: We randomly pick two plaintexts m_1, m_2 and get its corresponding ciphertext c_1, c_2. We run an exhaustive search on 2^{33} candidate keys for 39 key bits in $\mathcal{D}^c$ denoted by k_1' and 2^{25} remaining 25 key bits denoted by k_2 to get the unique and correct key that satisfies $E_{k_1'||k_2}(m_1) = c_1 \wedge E_{k_1'||k_2}(m_2) = c_2$.

The quantum partial key guessing phase in $\mathcal{QRKR}$

In this section, we give the quantum circuit of Step 3 and the corresponding quantum resources estimate. We consider Q1 model as our attack model where both Step 1 and Step 2 are classical processes. Thus to design the quantum circuit of quantum key recovery, we only need to regard Step 3 and Step 4 as two QAA instances separately.

In Step 3, four differentials are used to get candidate keys for 39 key bits in $\mathcal{D}_c$. So the QAA instance of Step 3 is actually the combination of four sub-QAA instances

Table 6 The cost of quantum exhaustive search on SIMON32/64

Round	#NOT	*#CNOT$_{sum}$*		*#H$_{sum}$*		#Cliff	#T	T-depth	Full depth	#qubit	Refer.
		#CNOT	#Toff-C	#H	#Toff-H						
32	2^{43}	$1.62 \cdot 2^{45}$	0	0	0	$1.35 \cdot 2^{45.5}$	$1.27 \cdot 2^{46}$	$1.18 \cdot 2^{45}$	$1.05 \cdot 2^{46.3}$	161	(Anand et al. 2020c)
32	$1.41 \cdot 2^{41}$	$1.87 \cdot 2^{44}$	$1.15 \cdot 2^{46}$	$1.62 \cdot 2^{38}$	$1.32 \cdot 2^{44}$	2^{47}	$1.15 \cdot 2^{46}$	$1.87 \cdot 2^{41}$	2^{44}	255	This paper
19	$1.52 \cdot 2^{40}$	$1.07 \cdot 2^{44}$	$1.41 \cdot 2^{45}$	$1.62 \cdot 2^{38}$	$1.62 \cdot 2^{43}$	$1.23 \cdot 2^{46}$	$1.41 \cdot 2^{45}$	$1.74 \cdot 2^{41}$	$1.74 \cdot 2^{43}$	255	This paper

corresponding to the four processes of partial key guessing using four differentials. The input of every sub-QAA instance is 2^{25} partial keys and $2^{12.5}$ plaintext pairs, while the output is a superposition state of $2^{23.5}$ plaintext-key pairs. We need to design quantum circuit for each sub-QAA instance. Once we have the quantum circuit of one sub-QAA instance using one differential, we can easily design the other three quantum circuits for the other three sub-QAA instances because the four key recovery processes using four differentials are quite similar. Besides, after our analysis, the cost of these four quantum circuits are totally the same. Thus here we only provide the quantum circuit of key recovery process using $\mathcal{D}_2$.

Our sub-QAA instance searches the key-plaintext pairs that satisfy Eq. (2). The quantum circuit of this sub-QAA instance is in Fig. 8. To achieve our attack, we need to implement two operators C_1 and C_2 when given classical tuples $(m_i, E(m_i), E(m_i \oplus \Delta x_2)), i = 1, \cdots, 2^{12.5}$. The operator C_1 is defined as $C_1|0\rangle = \sum_{i=1}^{2^{12.5}} |m_i\rangle$. And the operator C_2 is defined as $C_2 \sum_{i=1}^{2^{12.5}} |m_i\rangle|0\rangle|0\rangle = \sum_{i=1}^{2^{12.5}} |m_i\rangle|E(m_i)\rangle|E(m_i \oplus \Delta x_2)\rangle$. We suppose the implementation of operator C_1 and C_2 is efficient so that the cost of operator C_1 and C_2 can be ignored. To implement the quantum circuit in Fig. 8, we need to implement U_g and U_s separately. The main cost of operator U_s comes from one 57-fold controlled-NOT gate. The main cost of operator U_g comes from the computation of h and one 32-fold controlled-NOT gate. The operator h corresponds to the process of computing Δ^{15} from given ciphertext pairs, denoted by $(E(m), E(m \oplus \Delta x_2))$ and 25 key bits in $\mathcal{D}_2^K$, denoted by k_1.

Here, we describe the implementation of U_g. At first, we define a function h as follows.

$$\begin{aligned} h: \{0,1\}^{32} \times \{0,1\}^{25} &\rightarrow \{0,1\}^{32} \\ (m, k_1) &\rightarrow D_{k_1}^4(E(m)) \oplus D_{k_1}^4(E(m \oplus \Delta x_2)) \end{aligned}$$

Then we define a function g as follows based on h.

$$g(m, k_1) = \begin{cases} 1, \text{if } h(m, k_1) = \Delta_{out}^2 \\ 0, \text{if } h(m, k_1) \neq \Delta_{out}^2 \end{cases}$$

Naturally, the operator U_g is defined as follows:

$$U_g|k_1\rangle|m\rangle|0\rangle|0\rangle|0\rangle = \begin{cases} |k_1\rangle|m\rangle|0\rangle|0\rangle|0\rangle, \text{if } g(m, k_1) = 0 \\ -|k_1\rangle|m\rangle|0\rangle|0\rangle|0\rangle, \text{if } g(m, k_1) = 1 \end{cases}$$

Next, we describe the computation process of h. We denote the input ciphertext pair by $E(m) = (L^{19}, R^{19}), E(m \oplus \Delta x_2) = ((L^{19})', (R^{19})')$. The computation process to get Δ^{15} using $\mathcal{D}_2$ is as follows:

1. From the given ciphertext pair, we can easily get $\Delta^{19} = (L^{19} \oplus (L^{19})', R^{19} \oplus (R^{19})')$.
2. With guessed 16 bits of K^{18}, we can get $L^{18} = L^{19}, R^{18} = f(L^{18}) \oplus K^{18} \oplus R^{19}$ and $\Delta^{18} = (L^{18} \oplus (L^{18})', R^{18} \oplus (R^{18})')$.
3. On one hand, we compute Δ^{17} in Eq.(3).

$$\begin{cases} \Delta L^{17} = \Delta R^{18}, \\ \Delta R^{17}[i] = \Delta And^{17}[i] \oplus \Delta Rot^{17}[i], \quad i = 0, 1, 2, 4, 6, 8, 9, 15 \\ \Delta R^{17}[3] = L^{17}[11] \oplus \Delta Rot^{17}[3] \\ \qquad = (L^{17})'[11] \oplus \Delta Rot^{17}[3] \\ \Delta R^{17}[10] = L^{17}[9] \oplus \Delta Rot^{17}[10] \\ \qquad = (L^{17})'[9] \oplus \Delta Rot^{17}[10] \end{cases} \tag{3}$$

   On the other hand, we can get $R^{16}[4, 6-9, 13, 15]$ and $(R^{16})'[4, 6-9, 13, 15]$ with guessed $K^{17}[4, 6-9, 13, 15]$ for the computation in the following Step.
4. On one hand, we compute Δ^{16} in Eq.(4).

$$\begin{cases} \Delta L^{16} = \Delta R^{17} \\ \Delta R^{16}[i] = \Delta And^{16}[i], \qquad i = 0, 7, 14 \\ \Delta R^{16}[1] = L^{16}[9] \oplus \Delta Rot^{16}[1] \\ \qquad = (L^{16})'[11] \oplus \Delta Rot^{16}[1] \\ \Delta R^{16}[2] = \Delta Rot^{16}[2] \\ \Delta R^{16}[8] = L^{16}[7] \oplus \Delta Rot^{16}[8] \\ \qquad = (L^{16})'[7] \oplus \Delta Rot^{16}[8] \end{cases} \tag{4}$$

   On the other hand, we compute the $R^{15}[7]$ and $(R^{15})'[5]$ with guessed $K^{16}[7] \oplus K^{17}[5], K^{16}[5] \oplus K^{17}[3]$.
5. We compute Δ^{15} in Eq.(5).

$$\begin{cases} \Delta L^{15} = \Delta R^{16}, \\ \Delta R^{15}[0] = \Delta L^{15}[14] \\ \Delta R^{15}[15] = L^{15}[7] = (L^{15})'[7] \\ \Delta R^{15}[6] = L^{15}[5] = (L^{15})'[5] \end{cases} \tag{5}$$

According to the above process, we provide our quantum circuit of h in Fig. 9. After a simple analysis of the circuit, we can easily get there are 232 CNOT gates and 60 Toffoli gates in the implementation of h. As for the circuit depth, the total depth of h is 99 and the T-depth of h is 24.

Having the quantum circuit of h, we could easily estimate the cost of quantum partial key guessing using differential $\mathcal{D}_2$ in Table 7. Following the same process, we can easily design the quantum circuit of the other three sub-QAA instances using $\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$ separately. And the cost of other three sub-QAA instances can also be seen in Table 7.

At last, we describe our method of generating candidate keys. Our defined sub-QAA instance of Step 3 outputs a superposition state of $2^{23.5}$ plaintext-key pair that satisfies Eq. (2) among $2^{12.5}$ plaintext pairs and 2^{25} partial keys after $\lfloor \frac{\pi}{4}\sqrt{2^{14}} \rfloor$ iterations. To get candidate keys, we measure the key register many times. The probability of measuring right partial key is $2^{-23.5}$. That is, we expect that we can get the right partial key after running this

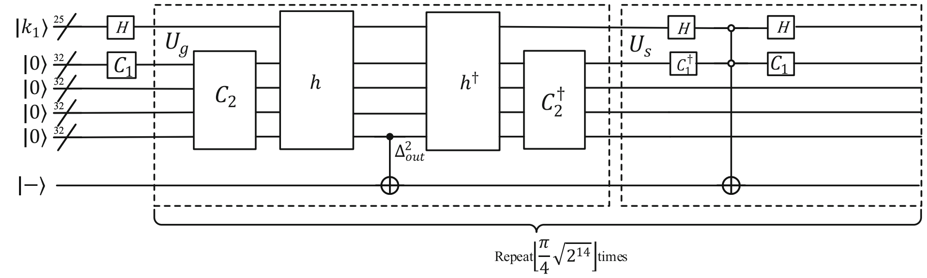

Fig. 8 The quantum circuit of partial key guessing using $\boldsymbol{\mathcal{D}}_2$

sub-QAA instance for $2^{23.5}$ times. And in expectation, we can get $2^{23.5}[1-(1-\frac{1}{2^{23.5}})^{23.5}] \approx 2^{22.8}$ different candidate keys for 25 key bits in $\mathcal{D}_2^K$ from $2^{23.5}$ measurements. After combining the results of the other three sub-QAA instances, we can get $(2^{22.8})^4/(2^{19}\times 2^{20}\times 2^{22}) = 2^{30.2}$ candidate keys for 39 key bits in $\mathcal{D}^c$. Despite that the cost of the process is a little high, we failed to find more efficient ways to get all candidate keys. Actually, Kaplen et al. also adopted a similar method to generate all candidate keys by measuring the key register for many times in Kaplan et al. (2016b). However, in their method, they ensured that the new gotten candidate key was different from the ones gotten before by excluding the keys that had been gotten in the QAA oracle. To implement their method using quantum circuit, a sequence of multi controlled-NOT gates need to be added in QAA oracle. That is, for every run, we need to design a new quantum circuit, which would greatly increase the quantum resources. Besides, the number of iteration increases with the increase of the number of elements needed to be excluded, which makes their encryption complexity also high. In our method, despite that we need to measure many times, we do not need to design a new quantum circuit in each run, which saves quantum resources.

Remark 1 *We consider a practical model, Q1 model. In Fig. 8, the operator C_1 achieves the process of preparing a superposition of $2^{12.5}$ classical plaintexts $m_i, i = 1, 2, \cdots, 2^{12.5}$. And the operator C_2 achieves the process of preparing a superposition of $2^{12.5}$ classical tuples $(m_i, E(m_i), E(m_i \oplus \Delta x_2))$. Actually, it's not known whether there exists such operators that could achieve such transformation, the difficulty of which is equal to preparing the superposition of random states. The choice of classical tuples may influence the efficiency of operator C_1 and C_2. If there are structures in the classical tuples, it may be efficient to get the target superposition state.*

The quantum exhaustive key search phase in $\mathcal{QRKR}$

In this section, we give the quantum circuit of Step 4 and estimate its quantum resources.

We define another QAA instance in search space of $2^{30.2}$ candidate keys for 39 bits in $\mathcal{D}^c$ denoted by k_1' and 2^{25} remaining 25 key bits denoted by k_2. According to (Jaques

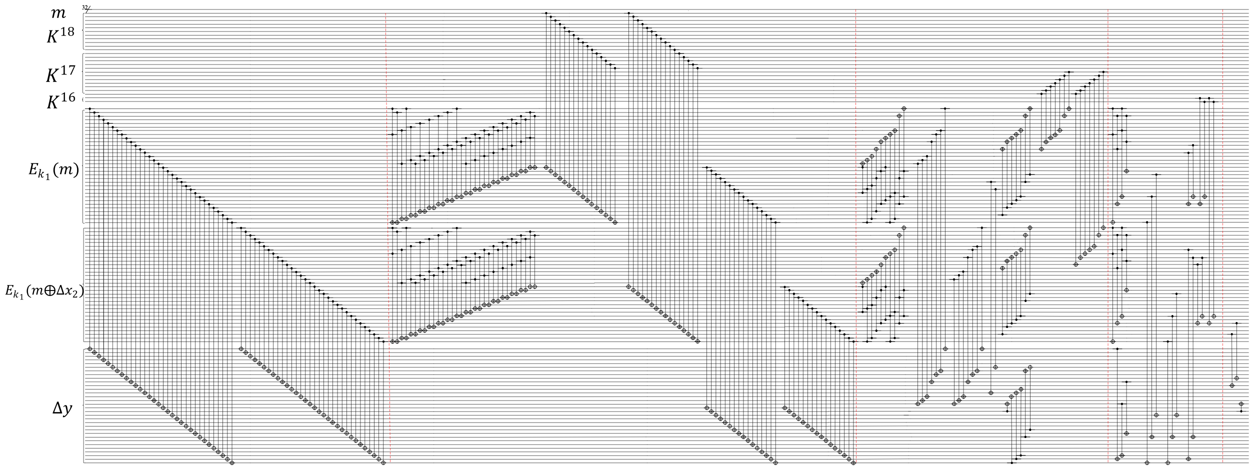

Fig. 9 The quantum circuit of function h. The input register Δy is equal to $E_{k_1}(m) \oplus E_{k_1}(m \oplus \Delta x_2)$. We use red dotted lines to split the calculation process of $\Delta^{19}, \Delta^{18}, \Delta^{17}, \Delta^{16}, \Delta^{15}$

Table 7 The cost of quantum partial key guessing using $\boldsymbol{\mathcal{D}}_i (i = 1, 2, 3, 4)$ in $\boldsymbol{\mathcal{QRKR}}$

#iter	#NOT	#CNOT$_{sum}$		#H$_{sum}$		#Cliff	#T	T-depth	Full depth	#qubit
		#CNOT	#Toff-C	#H	#Toff-H					
1	0	464	2044	50	584	3142	2044	732	2560	209
$\lfloor \frac{\pi}{4}\sqrt{2^{14}} \rfloor$	0	$1.41 \cdot 2^{15}$	$1.52 \cdot 2^{17}$	$1.23 \cdot 2^{12}$	$1.74 \cdot 2^{15}$	$1.23 \cdot 2^{18}$	$1.52 \cdot 2^{17}$	$1.15 \cdot 2^{16}$	2^{18}	209

et al. 2020), we need to choose two plaintexts m_1, m_2 and get their corresponding ciphertexts c_1, c_2 in QAA oracle to ensure the uniqueness of solution. The quantum circuit of Step 4 is in Fig. 10. The C operator is a creation operator, which creates the superposition state of $2^{30.2}$ candidate keys for 39 key bits in $\mathcal{D}^c$ from the all-zero state, which is defined as $C|0\rangle = \sum_{i=1}^{i=2^{30.2}} |(k_1')^i\rangle$. As previously assumed, we also assume that this process is efficient so that the cost of operator C could be ignored. Then, we need to implement the quantum circuit of U_g and U_s separately. The main cost of U_s is one 64-fold controlled-NOT gate. The main cost of U_g is four SIMON instances, and the circuit of U_g is shown in Fig. 11.

At first, we define a function h as follows, which corresponds to the encryption process of m_1, m_2 with given $k_1' || k_2$.

$$
\begin{aligned}
h: \{0,1\}^{39} \times \{0,1\}^{25} &\to \{0,1\}^{32} \times \{0,1\}^{32} \\
(k_1', k_2) &\to (E_{k_1'||k_2}(m_1), E_{k_1'||k_2}(m_2))
\end{aligned}
$$

Then based on h, we define a function g as follows:

$$
g(k_1', k_2) = \begin{cases} 1, \text{if } f(k_1', k_2) = (c_1, c_2) \\ 0, \text{if } f(k_1', k_2) \neq (c_1, c_2) \end{cases}
$$

Naturally, the operator U_g is defined as follows:

$$
U_g |k_1'\rangle |k_2\rangle |0\rangle |0\rangle = \begin{cases} |k_1'\rangle |k_2\rangle |0\rangle |0\rangle, \text{if } g(k_1', k_2) = 0 \\ -|k_1'\rangle |k_2\rangle |0\rangle |0\rangle, \text{if } g(k_1', k_2) = 1 \end{cases}
$$

We need to iterate the QAA operator $G = U_s U_g$ for $\lfloor \frac{\pi}{4}\sqrt{2^{55.2}} \rfloor$ times. We can easily deduce the cost of Step 4 in Table 8.

The complexity analysis

Our research is related to three attacks, $\mathcal{QMKS}$, $\mathcal{QRKR}$, $\mathcal{CRKR}$. In this section, we compare the complexity of these three attacks. On one hand, we compare the encryption complexity and data complexity of $\mathcal{QMKS}$, $\mathcal{QRKR}$ and $\mathcal{CRKR}$. On the other hand, we compare the quantum circuit complexity of $\mathcal{QMKS}$ and $\mathcal{QRKR}$.

Encryption complexity and data complexity comparison

In this section, we compare the complexity of $\mathcal{QMKS}$, $\mathcal{QRKR}$ and $\mathcal{CRKR}$ in terms of encryption complexity and data complexity.

In $\mathcal{QMKS}$, to recover the master key, we need to carry out $\lfloor \frac{\pi}{4} 2^{32} \rfloor \times \frac{19}{32} \times 6 \approx 2^{33.5}$ encryptions, where 6 represents six SIMON instances in one QAA iteration. In our $\mathcal{QRKR}$, $4 \times 2^{23.5} \times \lfloor \frac{\pi}{4} 2^{\sqrt{14}} \rfloor \times \frac{4}{19} \times 2 + \lfloor \frac{\pi}{4} 2^{\sqrt{55.2}} \rfloor \times 4 \approx 2^{31.3}$ encryptions are needed. In the first term, 4 represents four sub-QAA instances using four differentials, and $\frac{4}{19}$ represents the complexity of 4-round decryption. In the second term, 4 represents four SIMON instances. On the whole, the encryption complexity of $\mathcal{QRKR}$ is slightly lower than $\mathcal{QMKS}$. Besides, the encryption complexity of $\mathcal{CRKR}$ is 2^{34} from Table 3. That is, the encryption complexity of $\mathcal{QRKR}$ is also lower than $\mathcal{CRKR}$. We can observe that the main encryption complexity comes from Step 3, generating candidate keys. As a result, if the complexity of Step 3 could be reduced further, $\mathcal{QRKR}$ could achieve much lower encryption complexity.

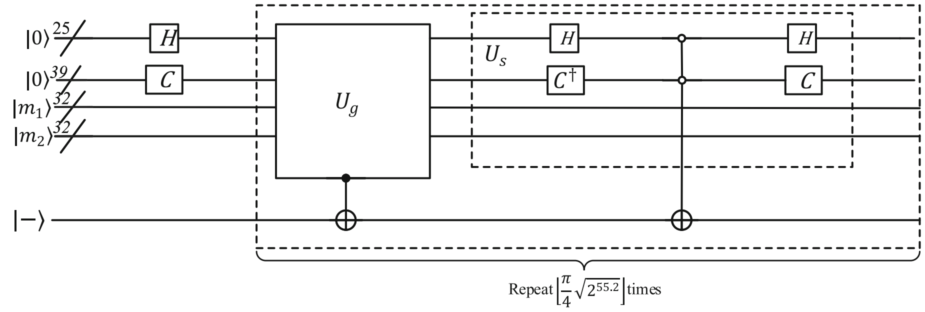

Fig. 10 The quantum circuit of exhaustive search in $\boldsymbol{\mathcal{QRKR}}$

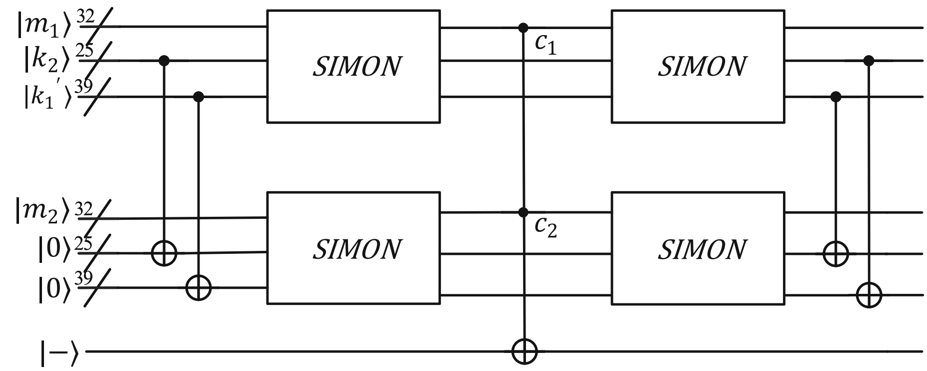

Fig. 11 The quantum circuit of U_g in Fig. 10

Although the data complexity isn't our focus, we still offer the comparison here for completeness. In $\mathcal{QMKS}$, 3 plaintexts are enough for the uniqueness of solution. In $\mathcal{CRKR}$, the data complexity is 2^{31} to get 4 right pairs in expectation. However, in our $\mathcal{QRKR}$, we only need to get one right pair in expectation. So the data complexity of $\mathcal{QRKR}$ is 2^{30}. That is, the data complexity of $\mathcal{QRKR}$ is lower than $\mathcal{CRKR}$.

Quantum circuit complexity comparison

In this section, we compare the complexity of $\mathcal{QMKS}$ and $\mathcal{QRKR}$ in terms of quantum circuit complexity.

We need to run four sub-QAA instances in Step 3. So multiplying the gate count in Table 7 by 4, we can get the quantum resources of Step 3 in the second line of Table 9. And the cost of Step 4 is listed in the third line of Table 9. From Table 9, we can observe that the cost of Step 3 in $\mathcal{QRKR}$ is far lower than that of Step 4 so that it can be omitted. The main cost of $\mathcal{QRKR}$ comes from Step 4 and it is lower than that of $\mathcal{QMKS}$. Thus we have that the quantum circuit complexity of $\mathcal{QRKR}$ is lower than that of $\mathcal{QMKS}$.

In summary, we gain a quantum dedicated attack that has lower encryption complexity and quantum circuit complexity than quantum generic attack on SIMON32/64. Besides, both the encryption complexity and data complexity of our attack are lower than the classical key-recovery attack in (Biryukov et al. 2014). However, we find it's not a big complexity gap between our attack and exhaustive search in quantum setting due to the big complexity of generating candidate keys.

Conclusion

In this paper, we studied the quantum key recovery attack on SIMON32/64 using QAA algorithm in Q1 model. We reanalyzed the quantum circuit complexity of quantum exhaustive search on SIMON32/64 and firstly offered a quantum dedicated attacks on SIMON32/64. And our work studied quantum dedicated attacks from the perspective of quantum circuit complexity for the first time, which can provide a research basis for performing real attacks on quantum computers in the future. On one hand, we gave more accurate estimate results of the quantum circuit complexity of quantum exhaustive search on SIMON32/64 than the results in (Anand et al. 2020c). We considered the number of Clifford gates more comprehensively and reduced the number of T gates. And we reduced the T-depth and full depth via small modifications. On the other hand, using the four differentials in (Biryukov et al. 2014) as our differential distinguisher, we gave our quantum key recovery attack on 19-round SIMON32/64. We treated the two phases of key recovery attack as two QAA instances separately and gave their corresponding quantum circuits, as well as quantum circuit complexity analysis separately. And the first QAA instance is composed of four sub-QAA instances corresponding to four differentials. At last, we compare the complexity of our quantum key recovery attack, quantum exhaustive search

Table 8 The cost of quantum exhaustive search in $\mathcal{QRKR}$

#iter	#NOT	#$CNOT_{sum}$		#H_{sum}		#Cliff	#T	T-depth	Full depth	#qubit
		#CNOT	#Toff-C	#H	#Toff-H					
1	480	3392	10262	78	2932	17144	10262	921	3718	191
$\lfloor \frac{\pi}{4}\sqrt{2^{55.2}} \rfloor$	$1.15 \cdot 2^{36}$	2^{39}	$1.52 \cdot 2^{40}$	$1.41 \cdot 2^{33}$	$1.74 \cdot 2^{38}$	$1.23 \cdot 2^{41}$	$1.52 \cdot 2^{40}$	$1.07 \cdot 2^{37}$	$1.07 \cdot 2^{39}$	191

Table 9 The cost comparison between $\mathcal{QMKS}$ and $\mathcal{QRKR}$

Algorithm	#NOT	$\#CNOT_{sum}$		$\#H_{sum}$		#Cliff	#T	T-depth	Full depth	#qubit
		#CNOT	#Toff-C	#H	#Toff-H					
$\mathcal{QMKS}$	$1.52 \cdot 2^{40}$	$1.07 \cdot 2^{44}$	$1.41 \cdot 2^{45}$	$1.62 \cdot 2^{38}$	$1.62 \cdot 2^{43}$	$1.23 \cdot 2^{46}$	$1.41 \cdot 2^{45}$	$1.74 \cdot 2^{41}$	$1.74 \cdot 2^{43}$	255
$\mathcal{QRKR}$	0	$1.41 \cdot 2^{17}$	$1.52 \cdot 2^{19}$	$1.23 \cdot 2^{14}$	$1.74 \cdot 2^{17}$	$1.23 \cdot 2^{20}$	$1.52 \cdot 2^{19}$	$1.15 \cdot 2^{16}$	2^{18}	209
	$1.23 \cdot 2^{36}$	2^{39}	$1.52 \cdot 2^{40}$	$1.41 \cdot 2^{33}$	$1.74 \cdot 2^{38}$	$1.15 \cdot 2^{41}$	$1.52 \cdot 2^{40}$	$1.07 \cdot 2^{37}$	$1.07 \cdot 2^{39}$	191

attack and classical key recovery attack. We found our attack has lowest encryption complexity and the quantum circuit complexity of our attack is lower than quantum exhaustive search attck. However, we used the method of measuring many times to generate all the candidate keys and failed to find a better way to generate candidate keys, which is the bottleneck of reducing complexity. In the following work, we may try to combine other key recovery techniques with our quantum dedicated attack, such as the dynamic key-guessing techniques proposed by Wang et al. (Wang et al. 2018). Besides, more efforts should be made to study how to reduce the complexity of generating candidate keys. Further, we could investigate the physical feasibility of our attack by considering the decoherence time of quantum computers and the time of CNOT operation because the two-qubit operation takes a longer time than single-qubit operations.

Acknowledgements
Not applicable.

Authors' contributions
This work was done by Hui Liu under the supervision of Prof. Li Yang. The initial idea of measuring quantum attack complexity from quantum resources cost was proposed Yang. Liu developed the original idea and implemented an attack in this paper. The calculations and drawing work were done by Liu. Yang offered some meaningful guidance and modifications. The author(s) read and approved the final mauscript.

Funding
This work was supported by National Natural Science Foundation of China (Grant No. 61672517), National Natural Foundation of China (Key program, Grant No. 61732021), National Cyrptography Development Fund (Grant No. MMJJ20170108) and Beijing Municipal Science & Technology Commission (Grant No. Z191100007119006).

Availability of data and materials
All data and materials are included in this article.

Declarations

Competing interests
Not applicable.

Received: 27 December 2020 Accepted: 13 April 2021
Published online: 02 August 2021

References

Abed F, List E, Lucks S, Wenzel J (2014). International Workshop on Fast Software Encryption. https://doi.org/10.1109/access.2019.2894337

Almazrooie M, Samsudin A, Abdullah R, Mutter KN (2018) Quantum reversible circuit of aes-128. Quantum Inform Process 17:112

Amy M, Maslov D, Mosca M, Roetteler M (2013) A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Trans Comput-Aided Des Integr Circ Syst 32:818–830

Anand R, Maitra A, Mukhopadhyay S (2020a). https://github.com/raviro/quantsimon. Accessed 05 March 2021

Anand R, Maitra A, Mukhopadhyay S (2020b) Evaluation of quantum cryptanalysis on speck. International Conference on Cryptology in India. https://doi.org/10.1007/978-3-030-65277-7_18

Anand R, Maitra A, Mukhopadhyay S (2020c) Grover on simon. arXiv preprint arXiv:200410686

Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L (2015) Simon and speck: Block ciphers for the internet of things. IACR Cryptol ePrint Arch 2015:585

Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, Sasaki Y, Sasdrich P, Sim SM (2016) The skinny family of block ciphers and its low-latency variant mantis. In: Annual International Cryptology Conference. pp 123–153. https://doi.org/10.1007/978-3-662-53008-5_5

Bernstein E, Vazirani U (1997) Quantum complexity theory. SIAM J Comput 26(5):1411–1473

Biryukov A, Roy A, Velichkov V (2014) Differential analysis of block ciphers simon and speck. International Workshop on Fast Software Encryption. pp 546–570. https://doi.org/10.1007/978-3-662-46706-0_28

Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJ, Seurin Y, Vikkelsoe C (2007) Present: An ultra-lightweight block cipher. Springer, Berlin. pp 450–466

Bonnetain X, Naya-Plasencia M, Schrottenloher A (2019) Quantum security analysis of AES. IACR Trans Symmetric Cryptol:55–93. https://doi.org/10.46586/tosc.v2019.i2.55-93

Brassard G, Hoyer P, Mosca M, Tapp A (2002) Quantum amplitude amplification and estimation. Contemp Math 305:53–74

Chen H, Wang X (2016) Improved linear hull attack on round-reduced simon with dynamic key-guessing techniques. In: International Conference on Fast Software Encryption. Springer, Berlin. pp 428–449

Chu Z, Chen H, Wang X, Dong X, Li L (2018) Improved integral attacks on simon32 and simon48 with dynamic key-guessing techniques. Secur Commun Netw:2018. https://doi.org/10.1155/2018/5160237

Dong X, Dong B, Wang X (2020a) Quantum attacks on some feistel block ciphers. Designs. Codes Crypt 88:1–25

Dong X, Sun S, Shi D, Gao F, Wang X, Hu L (2020b) Quantum collision attacks on aes-like hashing with low quantum random access memories. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer. pp 727–757. https://doi.org/10.1007/978-3-030-64834-3_25

Grassl M, Langenberg B, Roetteler M, Steinwandt R (2016) Applying grover's algorithm to AES: quantum resource estimates. Post-Quantum Cryptography. https://doi.org/10.1007/978-3-319-29360-8_3

Grover LK (1997) Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett 79:325

Hosoyamada A, Sasaki Y (2018) Quantum demiric-selçuk meet-in-the-middle attacks: applications to 6-round generic feistel constructions. In: International Conference on Security and Cryptography for Networks. Springer, Cham. pp 386–403

Hosoyamada A, Sasaki Y (2020) Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 249–279

Jang K, Choi S, Kwon H, Kim H, Park J, Seo H (2020) Grover on korean block ciphers. Appl Sci 10:6407

Jaques S, Naehrig M, Roetteler M, Virdia F (2020) Implementing grover oracles for quantum key search on AES and lowmc. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp 280–310. https://doi.org/10.1007/978-3-030-45724-2_10

Kaplan M, Leurent G, Leverrier A, Naya-Plasencia M (2016a) Breaking symmetric cryptosystems using quantum period finding. In: Annual International Cryptology Conference. pp 207–237. https://doi.org/10.1007/978-3-662-53008-5_8

Kaplan M, Leurent G, Leverrier A, Naya-Plasencia M (2016b) Quantum differential and linear cryptanalysis. IACR Trans Symmetric Cryptol:71–94. https://doi.org/10.46586/tosc.v2016.i1.71-94

Koch D, Wessing L, Alsing PM (2019) Introduction to coding quantum algorithms: A tutorial series using pyquil. arXiv preprint arXiv:190305195

Kuwakado H, Morii M (2010) Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: 2010 IEEE International Symposium on Information Theory. pp 2682–2685. https://doi.org/10.1109/isit.2010.5513654

Kuwakado H, Morii M (2012) Security on the quantum-type even-mansour cipher. In: 2012 International Symposium on Information Theory and its Applications. IEEE, New York. pp 312–316

Langenberg B, Pham H, Steinwandt R (2020) Reducing the cost of implementing the advanced encryption standard as a quantum circuit. IEEE Trans Quantum Eng 1:1–12

Lau I (2021). https://github.com/aliceQuantum/SIMONQ. Accessed 05 March 2021

Leander G, May A (2017) Grover meets simon–quantumly attacking the fx-construction. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham. pp 161–178

Li H, Yang L (2015) Quantum differential cryptanalysis to the block ciphers. In: International Conference on Applications and Techniques in Information Security. pp 44–51. https://doi.org/10.1007/978-3-662-48683-2_5

Nielsen MA, Chuang IL (2001) Quantum computation and quantum information. Phys Today 54(2):60

Roetteler M, Wiebe N (2016) Quantum arithmetic and numerical analysis using repeat-until-success circuits. Quantum Inform Comput 16:134–178

Selinger P (2013) Quantum circuits of t-depth one. Phys Rev A 87(4):042,302

Shi D, Hu L, Sun S, Song L, Qiao K, Ma X (2017). Improved linear (hull) cryptanalysis of round-reduced versions of simon. ence China(Information ences) 60(3):1–3

Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp 124–134. https://doi.org/10.1109/sfcs.1994.365700

Simon DR (1997) On the power of quantum computation. SIAM J Comput 26:1474–1483

Sun L, Fu K, Wang M (2015) Improved zero-correlation cryptanalysis on simon. In: International Conference on Information Security and Cryptology. Springer. pp 125–143. https://doi.org/10.1007/978-3-319-38898-4_8

Wang N, Wang X, Jia K, Zhao J (2018) Differential attacks on reduced simon versions with dynamic key-guessing techniques. Sci China Inform Sci 61:098,103

Wang Q, Liu Z, Varıcı K, Sasaki Y, Rijmen V, Todo Y (2014) Cryptanalysis of reduced-round simon32 and simon48. In: International Conference on Cryptology in India. pp 143–160. https://doi.org/10.1007/978-3-319-13039-2_9

Xie H, Yang L (2019) Using bernstein–vazirani algorithm to attack block ciphers. Designs. Codes Crypt 87:1161–1182

Zhandry M (2012) How to construct quantum random functions. In: 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science. pp 679–687. https://doi.org/10.1109/focs.2012.37

Zou J, Wei Z, Sun S, Liu X, Wu W (2020) Quantum circuit implementations of AES with fewer qubits. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer. pp 697–726. https://doi.org/10.1007/978-3-030-64834-3_24

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.