New Quantum Search Model on Symmetric Ciphers and Its Applications

Yangru Zheng[1], Juntao Gao[1], and Baocang Wang[1]

School of Telecommunications Engineering, Xidian University, China
jtgao@mail.xidian.edu.cn

Abstract. It has been a long-standing viewpoint that doubling the length of key seeds in symmetric cipher can resist the quantum search attacks. This paper establishes a quantum key search model to deal with the post-quantum security of symmetric ciphers. The quantum search is performed in the punctured keystream/ciphertext space instead of the key space. On inputting the punctured keystreams/ciphertexts, we rule out the fake keys and find out the real key via the iterative use of the quantum singular value search algorithm. We find out several parameters, such as the length and min-entropy of the punctured keystream, the iterations, and the error in the search algorithm, and all of them can influence the resulting complexity. When these parameters are chosen properly, a better complexity can be obtained than Grover algorithm. Our search model can apply to any typical symmetric cipher. To demonstrate the power, we apply our model to analyze block cipher AES family, stream ciphers Grain-128 and ZUC-128. The resulting complexity of AES-128 is $\tilde{\mathcal{O}}(2^{30.8})$, $\tilde{\mathcal{O}}(2^{32.0})$ of AES-192, $\tilde{\mathcal{O}}(2^{32.7})$ of AES-256, $\tilde{\mathcal{O}}(2^{27.5})$ of Grain-128, and $\tilde{\mathcal{O}}(2^{39.8})$ of ZUC-128.
Our results show that increasing the length of key seeds is not an effective way anymore to resist the quantum search attacks, and it is necessary to propose new measures to ensure the post-quantum security of symmetric ciphers.

Keywords: Stream cipher · Block cipher · post-quantum security · quantum search algorithm.

1 Introduction

The symmetric cipher uses the same key in the encryption and decryption, which consists of block ciphers and stream ciphers.

The stream cipher [1] encryption process consists of an initialization process and a keystream output process. The initialization process doesn't output any keystream, with input of a fixed-length key seed and an optional initialization vector (IV). And it makes sure that the key seed and initialization vector are sufficiently mixed to make the states on each register more random and to prepare for the keystream output process. The keystream output process starts after the initialization process is completed, and each clock output one symbol (bit, byte or word), with updating states on each register. When the required length

of outputting keystream is in polynomial length, a secure stream cipher can be considered as a pseudo-random function.

The block cipher [1], $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m$, encrypts a block ($m$ bits) of plaintext into the ciphertext of the same length by a key in fixed length of n. Moreover, F is a keyed function such that, for all keys k, the function F_k defined by $F_k(x) \stackrel{def}{=} F(k,x)$ is a bijection, i.e., a permutation. And the main distinction between block ciphers and pseudo-random permutations is that the former typically only support a specific set of key/block lengths, and in particular do not support arbitrary-length keys.

As for stream ciphers, it is difficult to traverse the full key space for the large size due to the limitation of classical computers, so the distribution of each symbol in the outputting keystream cannot be determined. Grover's algorithm [2] is able to search for M specific elements in a set with size of N, achieving a squared speedup compared to classical search algorithms. Grover's algorithm can be illustrated in geometry. In the two-dimensional plane spanned by the target vector and its orthogonal vector, the angle between the current quantum state and the two vectors can be calculated by M and N. When Grover operator is applied, the current quantum state rotates fixed degree in the above plane. Based on the rotation degree, the number of the applied Grover operator can be calculated, so that the quantum state is rotated near the target vector, where the amplification of the target state is achieved. Grover's algorithm can attack against stream cipher algorithms with an initialization vector given, by traversing the full key space on a quantum computer and amplifying the amplitude of the quantum state corresponding to the key seed. The complexity is $\mathcal{O}(2^{n/2})$, where n denote the length of the key seed. In above attack, Grover's algorithm requires that the search target is the unique correct key, which means the attacker must be given a portion of the keystream such that it is uniquely mapped to the correct key by the encryption function (no other fake keys exist).

The search algorithm by quantum singular value transformation (QSVT) [3], combined with quantum signal process, can search for the unknown amount of specific data in the overall N data. By transforming between the left and right singular spaces and rotating within the spaces, in the Bloch sphere representation, the quantum state keeps spirally approaching and finally converges to the target quantum state. Because of convergence property, the search algorithm by QSVT only needs to know the size N of the search set. Besides, the amplification of the target state can be achieved even if the search oracle is applied too many times. As for Grover's algorithm, if only the size N of the search set is known, the optimal applied number of Grover search oracle can't be calculated. There is great probability that the Grover search oracle operates too many (or too few) times, so that the total rotation angle in the two-dimensional plane is too large (or too small), and the quantum state deviates from the target state, resulting in a failure. Therefore, without knowing the number of the target in the dataset, the Grover's algorithm is likely to fail, but the search algorithm by QSVT is still applicable.

In [4], a quantum signal processing framework is proposed, which uses $\mathcal{O}(d)$ elementary unitary quantum operations to achieve quantum subsystem's evolutionary transformation by simulation to nearly arbitrary d degree polynomials. The authors in [3] solves the synthesis problem of unitary quantum functions with a full characterization of achievable functions, and efficient techniques for their implementation. In [5], the authors propose a simulation algorithm required at most two auxiliary quits for the time-evolution operator $e^{-i\hat{H}t}$, such that the oracle in the algorithm is parameter-optimal in both asymptotic and non-asymptotic states. The key technology of the algorithm is qubitization, which uses a controlled oracle to embed the hermitian matrix $\hat{H}$ into the $SU(2)$ subspace. Qubitization forms a core tenet of quantum singular value transformation. In [6], the authors elaborate quantum singular value transformation combined with quantum signal process framework [4], and its applications on three central quantum problems, quantum search, factoring, and simulation.

As for block ciphers, Grover's algorithm is widely used in key search model. In [16], the authors apply Grover oracle in the AES key search. For detail, given a small number of plaintext pairs, the key of AES is searching by Grover's algorithm. And the AES attacking quantum circuit is designed with minimum qubits required and other quantum resources optimized, which has been adopted by the National Institute of Standards and Technology (NIST). In [17], the authors design an invertible quantum circuit for AES-128 algorithm with the same condition of minimum qubits, which decreases the number of quantum gates as well. The main optimized point is the quantum realization of S-box, which achieves the affine transformation in the S-box over finite field $GF(2^8)$ by Itoh-Tsujii algorithm [18]. In [13], the authors optimize the AES attacking quantum circuit based on [16], by searching two pairs of plaintext and ciphertext simultaneously in parallel in a quantum circuit, and prove that AES's quantum security is weaker than it NIST declares. Besides, they determine the relation between the key length and the pairs, and design the LowMC attacking quantum circuit in the same way. In [15], the off-line Simon algorithm is applied to the 2XOR-Cascade construction, and the attacking complexity is beyond the quadratic speedup of quantum search algorithm.

This paper searches in keystream/ciphertext space rather than key space. When the keystream/ciphertext space size is larger than the key space size, both Grover's algorithm and search algorithm by QSVT have squared acceleration effect. So consider a keystream/ciphertext space with much smaller size than key space. A short keystream/ciphertext may correspond to lots of keys, where a correct key and some fake keys exist. At this time, Grover's algorithm is not suitable, because of the unknown relation between key and keystream/ciphertext. Hence, using the search algorithm by QSVT [3], we propose a search model applicable to the quantum security analysis of symmetric ciphers. Firstly, we design a quantum algorithm to calculate the min-entropy of keystreams as preprocessing process, only for stream ciphers. And then, we design the single-round key search algorithm combined with the search oracle by QSVT, according to the operation rules of typical symmetric cipher. At last, based on the min-entropy

(or pseudo-randomness), run the single-round key search algorithm for r rounds, and return the correct key seed. Our search model attaches importance on how to obtain the key seed using quantum algorithms, so ultimately the attacking complexity against the symmetric cipher is our focus.

To verify the validity, we use our search model and instantiated block cipher AES family, stream ciphers Grain-128 and ZUC-128. The security of block cipher AES family is one of the most important issues in cryptanalysis. And these two stream ciphers are chosen for the representativity, and they both output the keystream after the initialization process. The keystream of the Grain type is one bit per symbol, and the ZUC type is one word per symbol.

The block cipher AES [19], released by NIST in 2001, is intended to replace DES as the widely used standard. In [20], the author publishes a probabilistic mixture-differential distinguisher on five rounds as well as a key-recovery attack on six rounds from [21], which costs $2^{72.8}$ required chosen plaintexts, 2^{105} time complexity and 2^{33} memory.

The stream cipher Grain is one of the hardware implementation-oriented stream ciphers solicited by the eSTREAM project. And the Grain-128 algorithm [7] is proposed as an improvement on the Grain v0 algorithm. The main classical attack algorithm against Grain-128 is chosen IV attack. Itai Dinur and others propose a dynamic cubic attack [8], which recovers the key seed belonging to a large subset of 2^{-10} of the key space. For 2^{118} key seeds applicable to Grain-128 stream cipher, an attacker can obtain 2^{15} of improvement than the exhaustive search.

The stream cipher ZUC [9] is identified as a next-generation international standard for LTE by the Third Generation Partnership Project Protocol (3GPP) in September 2011, and is established as a national standard in October 2016. The ZUC algorithm absorbs the advantages of cycle sequence generated by linear feedback shift register, bit-reorganization of Feistel structure with the nonlinearity, and S-box with the nonlinearity and strong diffusivity. The current classical attack algorithms are not effective in breaking the ZUC algorithm, so it is challenging to attack the ZUC algorithm.

By analyzing the number of quantum gates and the number of possible erroneous keys, we obtain optimal computational complexity of $\tilde{\mathcal{O}}(2^{27.5})$ for Grain-128, $\tilde{\mathcal{O}}(2^{39.8})$ for attacking ZUC-128, $\tilde{\mathcal{O}}(2^{30.8})$ for AES-128, $\tilde{\mathcal{O}}(2^{32.0})$ for AES-192, and $\tilde{\mathcal{O}}(2^{32.7})$ for AES-256.

Therefore, the attacking complexity in the quantum computing environment is much smaller than the classical attack. It indicates that two typical types of stream ciphers and a representative block cipher have some security risks in the quantum computing environment. Furthermore, increasing the key length of Grain-like or ZUC-like stream ciphers and AES family block cipher is no longer an effective method for enhancing the security to resist our search model.

2 Preliminaries

2.1 Symbol Description

The symbols and their representative meanings are shown in the Table 1.

Table 1. Symbol Description

Symbols	Representative Meaning
k	key seed
n	key seed length
IV/v_i	initialization vector
t	plaintext
m	initialization vector/block length
t	plaintext
z	punctured keystream
c'	punctured ciphertext
s	punctured keystream/ciphertext length
$Init(n)$	the complexity of full rounds initialization process
$Output(s,n)$	the complexity of outputting s-bit keystream(without initialization)
C_{block}	the complexity of block cipher quantum circuit
h	the min-entropy of keystream space
a	the proportion of the target in the multi-set
r	the iteration
K_i	key space

2.2 Grover's Algorithm

Grover's algorithm solves the problem of searching some specific elements in the set $S = \{1, \ldots, N\}$. As for the function $f : \{1, \ldots, N\} \to \{0, 1\}$, Grover's algorithm can find elements $\{\alpha\}$, for which $f(\alpha) = 1$. And for other elements $x \in \{1, \ldots, N\} \backslash \{\alpha\}$, $f(x) = 0$.

Set an operator $U_f : |x\rangle|y\rangle \to |x\rangle|y + f(x)\rangle$. If $x = \alpha$, $U_f|x\rangle|y + f(x)\rangle = |\alpha\rangle|y + 1\rangle$. Else, $U_f|x\rangle|y + f(x)\rangle = |x\rangle|y\rangle$. Specifically, if $|y\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, $U_f|x\rangle|y\rangle = (-1)^{f(x)}|x\rangle|y\rangle$. Else if $|y\rangle = |\phi\rangle$ is quantum state with arbitrary length,

$$U_f|\alpha\rangle|\phi\rangle = \sin(\theta)|\alpha\rangle|U(\phi)\rangle + \cos(\theta)|\psi^{\perp}\rangle,$$

where $(|\alpha\rangle\langle\alpha| \otimes I)\,|\psi^{\perp}\rangle = 0$.

Define Grover operator $G = (2|\alpha\rangle\langle\alpha| - I)\,U_f$, and operator $S = -U_f G U_f^{\dagger} G$. We can get the relation

$$S^t U_f|\alpha\rangle|\phi\rangle = \sin\left[(2t+1)\,\theta\right]|\alpha\rangle|U(\phi)\rangle + \cos\left[(2t+1)\,\theta\right]|\psi^{\perp}\rangle.$$

We apply above operators to the quantum state $|s\rangle$, which means $|s'\rangle = SU_f|s\rangle$. As is shown in Fig. 1, the above process can be seen as a rotation of 2θ

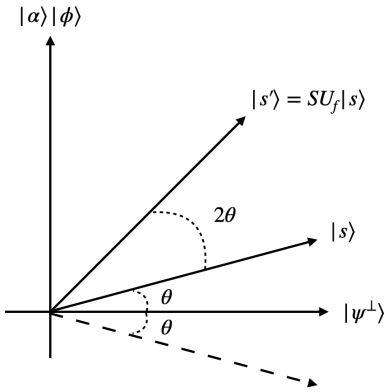

Fig. 1. Grover's algorithm

degree on the two-dimension plane, which is spanned by the superposition state corresponding to the special vector and its orthonormal state.

When the goal is searching M elements out of N, then

$$\sin^2(\theta) = M/N, 0 < \theta \leq \pi/2.$$

If $M \ll N$, $\theta \approx \sin(\theta) = \sqrt{M/N}$. We can successfully measure the special vectors $\{\alpha\}$ with probability of $1-\frac{M}{N}$, after applying $\left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor$ operator S. Above all, Grover algorithm can achieve quadratic acceleration compared to classical unordered database search algorithms.

2.3 Quantum Signal Processing

Set a quantum state $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$, where $\theta \in [0, \pi]$, $\phi \in [0, 2\pi]$. The parameters θ and ϕ can locate a point in Bloch sphere [10], as shown in Fig. 2.

Quantum signal processing(QSP) is built on the idea of interleaving two kinds of single-qubit rotations: a signal rotation operator W, and a signal processing rotation operator S. These rotation operations are about different axes through the Bloch sphere. For instances, $W(a) = \begin{bmatrix} a & i\sqrt{1-a^2} \\ i\sqrt{1-a^2} & a \end{bmatrix}$ is an x-rotation of $\theta = -2\cos^{-1}(a)$ degree, and $S(\phi) = e^{i\phi Z}$ is a z-rotation of -2ϕ degree.

Definition 1. *[11] For a tuple of phases $\overrightarrow{\phi} = (\phi_0, \phi_1, \cdots, \phi_d) \in \mathbb{R}^{d+1}$, the QSP operation sequence $U_{\overrightarrow{\phi}}$ is defined as*

$$U_{\overrightarrow{\phi}} = e^{i\phi_0 Z}\prod_{k=1}^{d} W(a)e^{i\phi_k Z}.$$

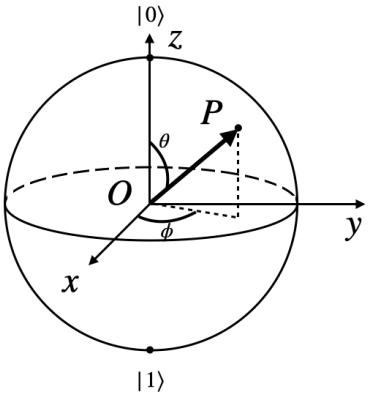

Fig. 2. Bloch sphere

Based on the definition 1, we have the following theorem:

Theorem 1. *[11] The QSP sequence $U_{\overrightarrow{\phi}}$ produces a matrix which may be expressed as a polynomial function of a:*

$$e^{i\phi_0 Z}\prod_{k=1}^{d}(W(a)e^{i\phi_k Z}) = \begin{bmatrix} P(a) & iQ(a)\sqrt{1-a^2} \\ iQ^*(a)\sqrt{1-a^2} & P^*(a) \end{bmatrix},$$

for $a \in [-1,1]$, and a $\overrightarrow{\phi}$ exists for any polynomial P, Q in a such that:

1. $\deg(P) \leq d$, $\deg(Q) \leq d-1$.

2. P has parity d $\mod 2$, *Q has parity* $(d-1) \mod 2$.

3. $|P(a)|^2 + (1-a^2)|Q(a)|^2 = 1$.

The authors in [4] indicate that, Remez-type exchange algorithm can compute a $\overrightarrow{\phi}$ that produces a good approximation to any feasible polynomials P and Q.

2.4 Search Algorithm by QSVT

Apply an operator U to initial state $|B_0\rangle$. Our goal is to search target state $|A_0\rangle$ among states $U|B_0\rangle$.

Let $a = \langle A_0|U|B_0\rangle$, and $a \neq 0$(If $a = 0$, searching set $U|B_0\rangle$ doesn't contain target state $|A_0\rangle$). If a is known, Grover's algorithm can be used for amplitude amplification. Else, if a is unknown but the low bound of a, the search algorithm by quantum singular value transformation(QSVT) can solve the problem. The search algorithm by QSVT is elaborated as following:

Let $|A_\perp\rangle = \frac{1}{\mathcal{N}}\left(I - |A_0\rangle\langle A_0|\right)U|B_0\rangle$, where $\mathcal{N}$ is the normalization factor needed to make $|A^\perp\rangle$ a unit vector. And $U|B_0\rangle = a|A_0\rangle + \sqrt{1-a^2}|A_\perp\rangle$, $U|B_\perp\rangle =$

$-a|A_\perp\rangle + \sqrt{1-a^2}|A_0\rangle$. Besides, the singular value decomposition is

$$U = a\left(|A_0\rangle\langle B_0| - |A_\perp\rangle\langle B_\perp|\right) + \sqrt{1-a^2}\left(|A_\perp\rangle\langle B_0| + |A_0\rangle\langle B_\perp|\right),$$

where $|A_0\rangle$, $|A_\perp\rangle$ are left singular vectors, and $|B_0\rangle$, $|B_\perp\rangle$ are right singular vectors.

Thus, the block encoding of the operator U is

$$U = \begin{bmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{bmatrix}.$$

The search algorithm by QSVT can measure the target state $|A_0\rangle$ with the probability approximate to $1(poly(a) \to 1)$, combined quantum signal processing in a way of applying a sequence of rotation operators and operator U. Based on Theorem 1, Theorem 2 can be deduced:

Theorem 2. *[11] Given a unitary U, its inverse $U^\dagger$, and operator $A_\phi = e^{i\phi|A_0\rangle\langle A_0|}$, $B_\phi = e^{i\phi|B_0\rangle\langle B_0|}$,*

$$\langle A_0| \left[\prod_{k=1}^{d/2} U B_{\phi_{2k-1}} U^\dagger A_{\phi_{2k}}\right] U|B_0\rangle = poly(a),$$

where $poly(a)$ is a polynomial in $a = \langle A_0|U|B_0\rangle$ of degree at most d, satisfying the conditions on P from Theorem 1.

As shown in Fig. 3, the geometry representation of Theorem 2 is as below:

Let $H_A = span(|A_0\rangle, |A_\perp\rangle)$ and $H_B = span(|B_0\rangle, |B_\perp\rangle)$ denote two invariant subspaces separately spanned by left/right singular vectors.

1. The operator U maps vectors in space H_B to vectors in space H_A with a rotation.

2. The operator A_ϕ works as a rotation around vector $|A_0\rangle$ with certain degree, and the operator B_ϕ works around vector $|B_0\rangle$.

3. The operator $U^\dagger$ maps vectors in space H_A to vectors in space H_B with a rotation.

Theorem 2 indicates that, search algorithm by QSVT combined with quantum signal proceeding, makes the final vector gradually converge on target vector $|A_0\rangle$ in a way of transforming between two singular vector spaces and rotations around singular vectors.

As for Grover's algorithm, on condition that only the low bound of $a = \langle A_0|U|B_0\rangle$ is known, we try to increase the number of applied Grover operator to solve the problem. Specifically, if the low bound of a works as parameter in Grover's algorithm but not exact value of a, the number of applied Grover operator can be much more than the optimal number corresponding to Grover's algorithm with exact value of a. As shown in Fig. 1, when the number of Grover operator is more than excepted, vector rotates a lager angle in total in the plane, and the probability $p = |\langle a \otimes \phi|S^t U_f|s\rangle|^2$ no longer tends to 1. In result, the final vector is diverging from target vector, which fails amplitude amplification.

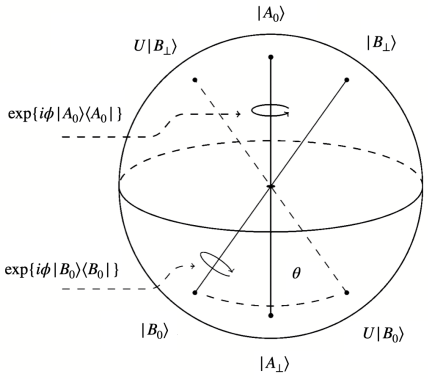

Fig. 3. The geometry representation of Theorem 2

In a word, compared to Grover's algorithm, search algorithm by QSVT don't have the problem that too many rotation operators fail amplitude amplification because of convergence.

In Theorem 2, the optimal function for $poly(a)$ is sign function

$$\Theta(x-c) = \begin{cases} -1 & x < c \\ 0 & x = c \\ 1 & x > c \end{cases}.$$

And sign function $\Theta(x-c)$ can be estimated with arbitrary precision by finding a polynomial approximation to gauss error function $erf\left(k\left[x-c\right]\right)$, for large enough k. Particularly, a degree $d = \mathcal{O}\left(\frac{1}{\Delta}\log\left(\frac{1}{\varepsilon}\right)\right)$ odd polynomial $P^{\Theta}_{\varepsilon,\Delta}(x-c)$ can be computed, where $\varepsilon \in \left(0, \sqrt{2/e\pi}\right)$, and such that

1. $|P^{\Theta}_{\varepsilon,\Delta}(x-c)| \leq 1$ for $x \in [-1,1]$.
2. $|\Theta(x-c) - P^{\Theta}_{\varepsilon,\Delta}(x-c)| \leq \varepsilon$ for $x \in [-1,1]\backslash(c-\frac{\Delta}{2}, c+\frac{\Delta}{2})$.

All in all, $P^{\Theta}_{\varepsilon,\Delta}(x-c)$ can ε-approximate sign function, as shown in Fig. 4.

Let N denote the searching set's size, and $|a| = |\langle A_0|U|B_0\rangle| \geq \frac{1}{\sqrt{N}}$. In Fig. 4, for an arbitrary value $|a| \geq \frac{1}{\sqrt{N}}$, $P^{\Theta}_{\varepsilon,\Delta}(a) \approx 1$, when $\Delta/2 \leq \frac{1}{\sqrt{N}}$. Thus, we get the Theorem 3.

Theorem 3. *[11] Given unitary operators U, $U^\dagger$, and rotation operators $A_\phi = e^{i\phi|A_0\rangle\langle A_0|}$, $B_\phi = e^{i\phi|B_0\rangle\langle B_0|}$,*

$$\langle A_0| \left[\prod_{k=1}^{d/2} U B_{\phi_{2k-1}} U^\dagger A_{\phi_{2k}}\right] U|B_0\rangle = P^{\Theta}_{\varepsilon,\Delta}(x-c),$$

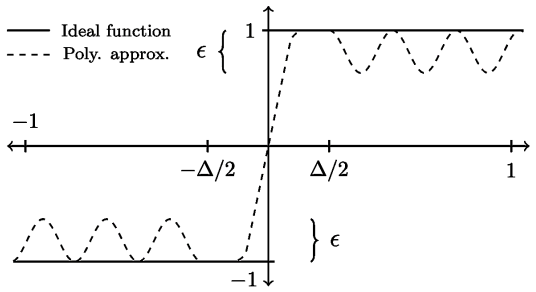

Fig. 4. Simulation of sign function

where $P^{\Theta}_{\varepsilon,\Delta}(x-c)$*is polynomial with degree at most* d*, satisfying the conditions on polynomial* P *in Theorem 1,*$\Delta \leq \frac{2}{\sqrt{N}}$*, and* $d = \mathcal{O}\left(\frac{1}{\Delta}\log\left(\frac{1}{\varepsilon}\right)\right) = \mathcal{O}\left(\sqrt{N}\log(1/\delta)\right)$.

In practice, odd polynomial $P^{\Theta}_{\varepsilon,\Delta}(x)$ is an approximation of $\Theta(x) = \begin{cases} -1, x < 0 \\ 0, x = 0 \\ 1, x > 0 \end{cases}$, satisfying the conditions in Theorem 1 with $poly(a) = \langle +|U_{\overrightarrow{\phi}}|+\rangle$. At this point, the operator sequence $U_{\overrightarrow{\phi}} = \left(P^{\Theta}_{\varepsilon,\Delta}\right)^{(SV)}(W) \approx \Theta^{(SV)}(W)$.

In summary, search algorithm by QSVT is as below:

Algorithm 1: Unstructured Search Algorithm by QSVT[11]

Input: Access to a controlled version of the oracle U which bit-flips an auxiliary qubit when given an unknown target state $|m\rangle$, an error tolerance $\delta = 2\varepsilon$, and a $\Delta \leq 2/\sqrt{N}$.

Output: The flagged state $|m\rangle$.

1 Use QSVT to construct the operator $\left(P^{\Theta}_{\delta/2,\Delta}\right)^{(SV)}(W)$, where W is the block encoding of U.

2 Apply $\left(P^{\Theta}_{\delta/2,\Delta}\right)^{(SV)}(W)$ to the uniform superposition. If the auxiliary is measured as $|+\rangle$, then $|m\rangle$ remains in the register. Else, repeat the above process.

Algorithm 1 successes in the probability of at least 1-δ, and costs 1 extra auxiliary qubit with complexity of $\tilde{\mathcal{O}}\left(\sqrt{N}\log\left(1/\delta\right)\right)$.

3 Quantum Search Model for Symmetric Ciphers

In this section, we design a search model which is universal to search the key both in block ciphers and stream ciphers. Because of the pseudo-randomness, the procedure in searching the block cipher key is simpler than stream cipher's.

3.1 Quantum Search Model for Stream Ciphers

Preprocessing Algorithm For saving quantum resources, we consider a punctured keystream, which only contains the first few bits rather than the full keystream.

Given an initialization vector IV, let

$$E_{IV,n} : \{0,1\}^n \to \{0,1\}^n, k_i \mapsto z_i$$

denote the encryption function with output of n-bit punctured keystream. Then, the image multi-set $E_{IV,n}(\{0,1\}^n)$ is n-bit punctured keystream space under the full key space.

Definition 2. *[12] A random variable X has min-entropy at least h iff* $\Pr(X = x) \leq \frac{1}{2^h}$ *for all x.*

By Definition 2, let the min-entropy of image multi-set $E_{IV,n}(\{0,1\}^n)$ be at least h. Specifically, $\Pr(X = z) \leq \frac{1}{2^h}$, for an arbitrary keystream $z \in E_{IV,n}(\{0,1\}^n)$ and a random variable X representing an element in the keystream multi-set $E_{IV,n}(\{0,1\}^n)$. Hence, let a denote the proportion of an arbitrary punctured keystream z in the multi-set $E_{IV,n}(\{0,1\}^n)$, then

$$a \leq \frac{1}{2^h}. \tag{1}$$

The following search model works as chosen IV attack, which means an attacker has access to hold many pairs of an initialization vector and corresponding punctured keystream. The main idea of search model is that, focus on several punctured keystream spaces with given initialization vectors held by attacker, and narrow down the key space by searching in above punctured keystream spaces in turns until only the correct key seed remains. Note that all of initialization vectors of the stream cipher can be applied in our search model, which is more feasible than the usual chosen IV attack.

The decreasing scale of key space at the first round is at least 2^h, because the punctured keystream is generated by at least 2^h key seeds with the given initialization vector according to the inequality (1). However, the following decreasing scale is unknown, because the min-entropy can't be sure under the narrowed key space. The specific process is discussed in next part. Thus, the value of h plays an important role in calculating exact searching rounds (the pairs of IV and z). In the following parameter selection part, we conclude the relation between the rounds r and the min-entropy h, to make sure that the final measurement is the correct key.

In this way, this paper designs Algorithm 2 to calculate the min-entropy of image multi-set $E_{IV,n}(\{0,1\}^n)$, by figuring out whether each bit of the punctured keystreams is uniformly distributed or not. If t bits of keystream are uniformly distributed, we can deduce that the proportion $a \leq \frac{1}{2^t}$, which means the min-entropy of $E_{IV,n}(\{0,1\}^n)$ is at least t.

Algorithm 2: Quantum Min-entropy Algorithm

Input: An initialization vector IV.

Output: The distribution of bits in keystream.

1 Prepare the state $|0\rangle^{\otimes n}|0\rangle^{\otimes m}$, stored separately in register K and register V.

2 Apply n Hadamard gates to n qubits in the first register K, and load initialization vector IV in the second register V, where $|0\rangle^{\otimes n}|0\rangle^{\otimes m} \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|v\rangle$.

3 Add n qubits in the third register Z, where $\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|v\rangle|0\rangle^{\otimes n}$.

4 Apply the oracle $\mathcal{O}_{stream}$, where $\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|v\rangle|0\rangle^{\otimes n} \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k'\rangle|v'\rangle|z\rangle$, $|k\rangle \rightarrow |k'\rangle, |v\rangle \rightarrow |v'\rangle$ is the state update process in registers, $|z\rangle$ is the n-bit keystream after encryption, and the oracle $\mathcal{O}_{stream}$ achieves the encryption of stream cipher.

5 Apply n Hadamard gates to n qubits in the third register Z, and measure.

In step 2, the application of n Hadamard gates to qubits in the register K, makes an initial state $|0\rangle$ evolve to a uniform superposition $\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle$, which means each component of the state stored on register K is uniformly distributed in the full key space. In step 4, each component of the state stored on register Z corresponds to every possible punctured keystream after encryption under the full key space.

Hence, calculate the distribution of each bit of punctured keystreams (whether is uniform or not), for the punctured keystream space $Z = \{z_i = (z_{i,0}, \cdots, z_{i,n-1}) \; | z_i = E_{IV,n}(k_i), i = 0, \cdots, 2^n - 1\}$:

1. If $z_{*,j}$ distributes uniformly in $\{0,1\}$, $|z_{*,j}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, meaning that the j_{th} bit of keystreams under the full key space is uniformly distributed on 0-1 space.

At this time, apply a Hadamard gate to this qubit, namely

$$H|z_{*,j}\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |0\rangle.$$

We get the measurement of $|0\rangle$.

2. If $z_{*,j}$ distributes non-uniformly in $\{0,1\}$, $|z_{*,j}\rangle = \sqrt{\frac{1}{2}+\varepsilon}|0\rangle + \sqrt{\frac{1}{2}-\varepsilon}|1\rangle$, where $\varepsilon \in [-\frac{1}{2}, \frac{1}{2}]$, meaning that the j_{th} bit of keystreams under the full key space is non-uniformly distributed on 0-1 space.

At this time, apply a Hadamard gate to this qubit,

$$\begin{aligned} H|z_{*,j}\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\frac{1}{2}+\varepsilon} \\ \sqrt{\frac{1}{2}-\varepsilon} \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \left(\sqrt{\frac{1}{2}+\varepsilon} + \sqrt{\frac{1}{2}-\varepsilon} \right) |0\rangle + \frac{1}{\sqrt{2}} \left(\sqrt{\frac{1}{2}+\varepsilon} - \sqrt{\frac{1}{2}-\varepsilon} \right) |1\rangle. \end{aligned}$$

The probability of measuring $|1\rangle$ is

$$\Pr(|1\rangle) = |\langle 1|H|z_{*,j}\rangle|^2 = \frac{1}{2}\left(1 - 2\sqrt{\frac{1}{4} - \varepsilon^2}\right).$$

Hence, the measurement of $|1\rangle$ means that the j_{th} bit is non-uniformly distributed.

As discussed above, run Algorithm 2 c times, where c is constant. Let $t_i = (t_{i,0}, \cdots, t_{i,n-1})$ denote the measurement each time, where $i = 0, \cdots, c-1$. Let $t^* = (t_0^*, \cdots, t_{n-1}^*)$ denote distributions of each bit(whether is uniform or not), where $t_j^* = t_{0,j} \vee \cdots \vee t_{c-1,j}$, and $j = 0, \cdots, n-1$. If $t_j^* = 0$, the j_{th} bit is uniformly distributed. Else, the j_{th} bit is non-uniformly distributed. Set $t^{*,s} = (t_0^*, \cdots, t_{s-1}^*)$ as the first s-bit of vector t_j^*, and h as the number of components equal to 0 in vector $t^{*,s}$, where $h = s - \sum_{i=0}^{s-1} t_i^{*,s}$.

Let $E_{IV,s} : \{0,1\}^n \to \{0,1\}^s$ denote stream cipher encryption function with output of s-bit punctured keystream. And the image multi-set $E_{IV,s}(\{0,1\}^n)$ is equal to s-bit punctured keystream multi-set under the full key space. By the Algorithm 2, set the min-entropy of image multi-set $E_{IV,s}(\{0,1\}^n)$ at least h. Namely, $a \leq \frac{1}{2^h}$, where a is the proportion of each possible s-bit punctured keystream z in the image multi-set $E_{IV,s}(\{0,1\}^n)$ under the full key space. Because $E_{IV,s}(\{0,1\}^n) \subset \{0,1\}^s$,

$$\frac{1}{2^s} \leq a \leq \frac{1}{2^h}.$$

In summary, we run Algorithm 2 constant times, and figure out which bits are uniformly distributed of n-bit punctured keystream. And we choose the length s, to make sure that the first s-bit punctured keystream has min-entropy at least h, where h meets the limitations as discussed in the following parameter selection part.

Quantum Key Search Model Now, we design the key search model for stream cipher, but actually, it's suitable for block cipher by changing the input parameters and encryption oracle, as discussed in next section.

1. Single-round Key Search Model

Suppose an attacker want to get the key k^* of stream cipher. The attacker can choose an initialization vector IV_i, and get the s-bit punctured keystream $z_i = (z_{i,0}, z_{i,1}, \ldots, z_{i,s-1})$, corresponding to (k^*, IV_i). And then, search z_i in punctured keystream space $Z = \{z | E_{IV_i,s}(k) = z, k \in \{0,1\}^n\}$ by Algorithm 1 to search k^* in full key space, where $E_{IV_i,s}(k^*) = z_i$. Noticed, we don't use search algorithm in key space but keystream space. If $s \geq n$, searching space Z has the almost same size of the key space where the key k in. Else, the size of searching space Z is $|E_{IV_i,s}(\{0,1\}^n)| \leq 2^s$. The reasonable value range of s is discussed later. The design of search algorithm is as following: (quantum circuit is shown in Fig. 5)

Algorithm 3: Single-round Key Search Model

Input: The m-bit initialization vector IV_i, and s-bit punctured keystream $z_i = (z_{i,0}, z_{i,1}, \ldots, z_{i,s-1})$.

Output: The n-bit key.

1 Prepare the state $|0\rangle^{\otimes n}|0\rangle^{\otimes m}$, stored separately in register K and register V.

2 Add s qubits in the third register Z, where $|0\rangle^{\otimes n}|0\rangle^{\otimes m}|0\rangle^{\otimes s}$.

3 Apply n Hadamard gates to n qubits in the first register K, where $|0\rangle^{\otimes n}|0\rangle^{\otimes m}|0\rangle^{\otimes s} \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|0\rangle^{\otimes(m+s)}$.

4 Load the initialization vector IV in the second register V, where $\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|0\rangle^{\otimes(m+s)} \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|v_i\rangle|0\rangle^{\otimes s}$.

5 Apply the oracle $\mathcal{O}_{stream}$, where $\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|v_i\rangle|0\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k'\rangle|v_i'\rangle|z\rangle$, $|k\rangle \rightarrow |k'\rangle, |v\rangle \rightarrow |v'\rangle$ is the state update process in registers of stream cipher, $|z\rangle$ is the s-bit punctured keystream after encryption, and the oracle $\mathcal{O}_{stream}$ achieves the encryption of stream cipher.

6 Add one auxiliary qubit stored in register $flag$, and initialize the state to $|-\rangle$. Apply function $f_i(z)$ to bit-flip, where $|flag\rangle \rightarrow |(-1)^{f_i(z)} flag\rangle$, $f_i(z) = \begin{cases} 1 & z = z_i \\ 0 & z \neq z_i \end{cases}$.

7 Apply the search algorithm by QSVT for amplitude amplification of $|flag\rangle = |+\rangle$, where $a_1 \left(\sum_{k\in K_i} |k'\rangle\right) \otimes |\phi\rangle|+\rangle + a_2|\psi^{\perp}\rangle|-\rangle$, $a_1 \rightarrow \frac{1}{\sqrt{|K_i|}}$, $a_2 \rightarrow 0$, and $K_i = \{k|E(k, IV_i) = z_i\}$.

8 Measure register $flag$, where the quantum state collapses to $\frac{1}{\sqrt{|K_i|}} \sum_{k\in K_i} |k'\rangle|\phi\rangle|+\rangle$.

9 Uncompute operations in step 5 and 4, where $\frac{1}{\sqrt{|K_i|}} \sum_{k\in K_i} |k\rangle|0\rangle^{\otimes m+s}|+\rangle$.

10 Measure and return the state in register K.

Here are some implementation details about Algorithm 3. The search algorithm in step 6 is Algorithm 1, where the parameter Δ is related to s and n. And the oracle $\mathcal{O}_{stream}$ in step 4, which achieves the encryption of stream cipher consisting of an initialization process and a keystream output process, takes a key seed, an initialization vector and value s (punctured keystream's length) as input, and s-bit punctured keystream as output. In step 9, the uncomputing operation aims for rollbacking the state on registers to recover the original key and all-zero state for the measurement or the input of an initialization vector in next round search (Algorithm 4).

Let $Init(n)$ denote the complexity of full-round initialization in stream cipher, and $Output(s, n)$ denote the complexity of outputting s-bit punctured keystream (excluding initialization process). Therefore, the complexity of the oracle $\mathcal{O}_{stream}$ is $Init(n) + Output(s, n)$.

1. If $s \leq n$, $\Delta \leq 2/\sqrt{2^s}$. The complexity of Algorithm 3 is

$$\tilde{\mathcal{O}}\left(\sqrt{2^s}\log(1/\delta)\cdot\mathcal{O}_{stream}\right) = \tilde{\mathcal{O}}\left(2^{\frac{s}{2}}\log(1/\delta)\left(Init(n)+Output(s,n)\right)\right).$$

2. If $s > n$, $\Delta \leq 2/\sqrt{2^n}$. The complexity of Algorithm 3 is

$$\tilde{\mathcal{O}}\left(\sqrt{2^n}\log(1/\delta)\cdot\mathcal{O}_{stream}\right) = \tilde{\mathcal{O}}\left(2^{\frac{n}{2}}\log(1/\delta)\left(Init(n)+Output(s,n)\right)\right).$$

Let h denote the min-entropy of the s-bit punctured keystream space. If $h \geq n$, run Algorithm 3 and return the correct key k^*, where $E_{IV_i,s}(k^*) = z_i$. Because $s \geq h \geq n$, the attacking complexity is

$$\tilde{\mathcal{O}}\left(2^{\frac{n}{2}}\log(1/\delta)\left(Init(n)+Output(s,n)\right)\right),$$

which is still larger than $2^{\frac{n}{2}}$. Consequently, the designer can extend the key's length to resist the attack from Algorithm 3. We design a search model, multi-round key search model by QSVT, based on Algorithm 3. In brief, for the reduction of attacking complexity, shrink the length of punctured keystream, and repeat Algorithm 3 r times.

2. Multi-round Key Search Model

Algorithm 4 reduces the complexity of search algorithm in a smaller searching space of s-bit punctured keystreams, which in return has a better effect than quadratic speedup. The quantum circuit of Algorithm 4 is shown in Fig. 6.

Algorithm 4: Multi-round Key Search Model

Input: The r initialization vector s $IV_0, \ldots, IV_{r-1}$, and r corresponding punctured keystreams $z_0, \ldots, z_{r-1}$ (s-bit).

Output: The n-bits key.

1 Run step 1 to step 9 of Algorithm 3 with inputs of initialization vector IV_0 and keystream z_0. The state in registers K, V, Z is $\sum_{k\in K_0} a_k|k\rangle|0\rangle^{\otimes(m+s)}$, where $K_0 = \{k|E_{IV_0,s}(k) = z_0\}$. Quantum state collapses because of the measurement on the register $flag$.

2 For $i = 1, \ldots, r-1$, repeat step 4 to step 9 of Algorithm 3 with inputs of initialization vector IV_i and keystream z_i.

3 Measure and return the state in register K, which belongs to key space $K_{r-1} = \{k|E_{IV_i,s}(k) = z_i, i = 0, \ldots, r-1\}$.

In Algorithm 4, the decreasing process of key space in each round is

$$\begin{aligned}\{0,1\}^n \xrightarrow{E_{IV_0,s}(\cdot)=z_0} K_0 = \{k|E_{IV_0,s}(k) = z_0\} &\xrightarrow{E_{IV_1,s}(\cdot)=z_1} \\ K_1 = \{k|E_{IV_1,s}(k) = z_1, E_{IV_0,s}(k) = z_0\} &\xrightarrow{E_{IV_2,s}(\cdot)=z_2} \cdots \\ \xrightarrow{E_{IV_{r-1},s}(\cdot)=z_{r-1}} K_{r-1} = \{k|E_{IV_i,s}(k) = z_i, i = 0, \ldots, r-1\}&\end{aligned}$$

Besides, the decreasing scale of key space at the first round is at least 2^h, because the punctured keystream can be generated by at least 2^h key seeds

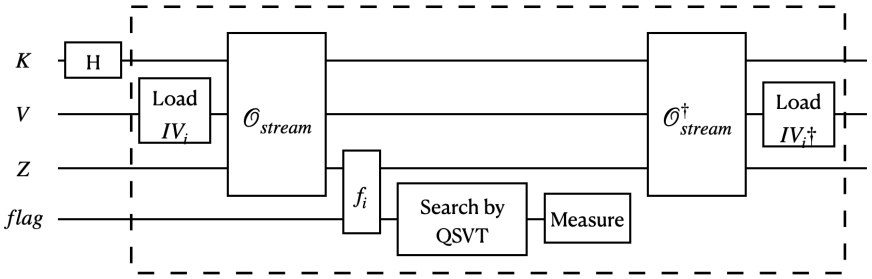

Fig. 5. Single-round key search model

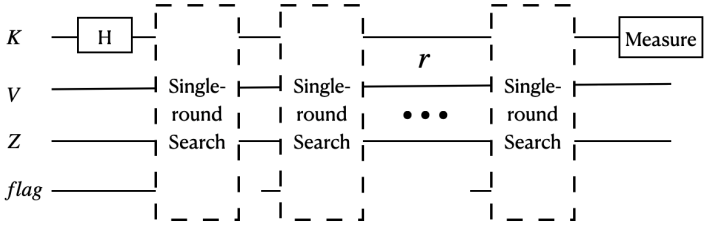

Fig. 6. Multi-round key search model

under the full key space with the given initialization vector, according to the inequality (1). However, the following decreasing scale is unknown, because the min-entropy can't be sure under the narrowed key space. But, in next part, we can guarantee that the result is the unique and correct key seed by meeting some certain conditions.

In the last round of searching, the size of punctured keystream space may be smaller than 2^h, where the percentage a of the target vector in searching space is more than $\frac{1}{2^h}$. Specifically, $a \geq \frac{1}{2^h} \geq \frac{1}{2^s}$, by the inequality $s \geq h$. Because of the convergence of search algorithm by QSVT, it still works well. Hence, Algorithm 4 can return the unique key, even if searching space has constant size in the last round.

Here is our search model's procedure. At the beginning, run preprocessing algorithm with an initialization vector IV several times, and make that the min-entropy of image multi-set $E_{IV,s}(\{0,1\}^n)$ is at least h with the chosen of s. Then, prepare r pairs of the initialization vector and keystream, put into Algorithm 4 as inputs, and measure the correct key, where the parameter r is related with h.

Consider the design idea of stream cipher, we make such reasonable assumption:

Assumption 1 *For most initialization vectors IV_i, the min-entropy of image multi-set $E_{IV_i,s}(\{0,1\}^n)$ is at least h.*

This assumption is reasonable, because only weak initialization vector will generate keystreams with abnormal min-entropy, while stream cipher algorithms widely analyzed always have very few weak initialization vectors. Based on Assumption 1, run one round search algorithm with cost of 1 auxiliary qubit and complexity of $\tilde{\mathcal{O}}(2^{\frac{s}{2}} \log \frac{1}{\delta} (Init(n) + Output(s, n)))$, to narrow down corresponding key space. And measure the register $flag$, which causes the collapse of state components while the corresponding keystreams aren't expected, in case that the wrong keys re-entry the following search process. Thus, before next round of searching, it is needed to prepare a new auxiliary qubit again.

All in all, the multi-round key search model costs

$$n + m + s + r + q$$

qubits, with complexity of

$$\tilde{\mathcal{O}}\left(r \cdot 2^{\frac{s}{2}} \log (1/\delta) (Init(n) + Output(s, n))\right), \tag{2}$$

where n represents key length, m represents initialization vector length, r represents the number of auxiliary qubits or the iteration, and q represents the number of extra qubits in the quantum realization of stream cipher.

Based on Assumption 1, we can see that our search model is of generality against stream ciphers. The initialization process of stream cipher is aimed to fully mix the key seed (and optional initialization vector) into some seemingly random initial states as input of outputting process, which in deed we take use of. Furthermore, the attacking complexity in (2) is mostly dependent on the value of s. A smaller s means the smaller complexity. It seems that we can decrease the attacking complexity by choosing a shorter keystream. However, the parameter s is restricted by the min-entropy of the punctured keystreams, and we cannot choose it on one's own will. In particular, more random the initial states are, more bits of keystreams under the full key space are uniformly distributed, and smaller value of s can be chosen where s-bits punctured keystreams have min-entropy at least h settled by Algorithm 2, which results in smaller attacking complexity. On the other hand, as for some stream ciphers whose initial states appear not so random, distinguishing attack must be a huge security threat.

Parameters Selection According to paper [13], we give analysis as follows.

Let k denote the n-bit key, k^* denote the correct key, IV denote an initialization vector, z denote s-bit keystream, and

$$E_{IV,s} : \{0,1\}^n \to \{0,1\}^s, k \mapsto z$$

denote the funtion of outputting s-bit keystream.

Definition 3. *For two initialization vectors IV_0, IV_1, if key k' satisfies $E_{IV_0,s}(k^*) = E_{IV_0,s}(k'), E_{IV_1,s}(k^*) = E_{IV_1,s}(k')$ and $k' \neq k^*$, k' is called fake key.*

It's of great possibility for the existence of fake keys because of the short length of keystream. By Assumption 1, $\Pr_{k^* \neq k'}\left(E_{IV,s}(k^*) = E_{IV,s}(k')\right) \leq 2^{-h}$. Given r initialization vectors $IV_0, \cdots, IV_{r-1}$,

$$\begin{aligned} p &= \Pr_{k^* \neq k'}\left(\left(E_{IV_0,s}(k^*), \cdots, E_{IV_{r-1},s}(k^*)\right) = \left(E_{IV_0,s}(k'), \cdots, E_{IV_{r-1},s}(k')\right)\right) \\ &\leq \prod_{i=0}^{r-1} \frac{1}{2^h - i}. \end{aligned}$$

For the condition

$$r^2 \ll 2^h, \tag{3}$$

we have

$$p \leq \prod_{i=0}^{r-1} \frac{1}{2^h - i} \approx 2^{-rh}. \tag{4}$$

Set the fake key set as $FK = \{k' | k' \neq k^*, \left(E_{IV_0,s}(k^*), \ldots, E_{IV_{r-1},s}(k^*)\right) = \left(E_{IV_0,s}(k'), \cdots, E_{IV_{r-1},s}(k')\right)\}$. By the inequality (4), $|FK| \leq (2^n - 1)\, 2^{-rh}$.

Let random variable X be the number of fake keys $|FK|$, where X follows the binomial distribution, and

$$\Pr(X = \alpha) = C_{2^n-1}^{\alpha} p^{\alpha} (1-p)^{2^n - 1 - \alpha}.$$

By the inequality (4),

$$\Pr(X = 0) = C_{2^n-1}^{0} p^0 (1-p)^{2^n-1} \geq \left(1 - 2^{-rh}\right)^{2^n - 1} \approx e^{-2^{n-rh}}. \tag{5}$$

By the inequality (5), if the pairs r of initialization vector and keystream, and min-entropy h of punctured keystream space, satisfy

$$r = \left\lfloor \frac{n}{h} + 1 \right\rfloor, \tag{6}$$

it is $p \geq e^{-2^{n-rh}}$ probability to confirm the unique and correct key.

Above all, if the parameters r and h satisfy the inequalities (3) and (6), we can guarantee that the probability of returning a unique key is $p \geq e^{-2^{n-rh}}$.

3.2 Quantum Search Model for Block Ciphers

As for searching block cipher key, we adopt the similar approach as above. Concluded from the structure of block cipher, we can make the assumption.

Assumption 2 *The encryption function of block cipher is a strong pseudo-random function.*

Set the encryption function of block cipher as

$$E_k : \{0,1\}^m \to \{0,1\}^m,\ t \mapsto c,$$

where k denotes the key of n bits, t denotes the plaintext of m bits, and c denotes corresponding ciphertext. By Assumption 2, function $E_k(\cdot)$ is a pseudo-random function.

Set the puncture function

$$f_s\left((x_1, \cdots, x_m)\right) = (x_1, \cdots, x_s),$$

where $s \leq m$. And define a compound function $g_{k,s} : \{0,1\}^m \to \{0,1\}^s$,

$$g_{k,s}(t) = f_s \circ E_k(t) = f_s(c) = c'.$$

It is easy to conclude that function $g_{k,s}(\cdot)$ is a strong pseudo-random function as well. By the definition and property of $g_{k,s}(\cdot)$, we can map each plaintext t to the first s bits of corresponding ciphertext c' uniformly, which means

$$\forall c' \in \{0,1\}^s, \Pr_{t \in \{0,1\}^m}\left(g_{k,s}(t) = c'\right) = \frac{1}{2^s}. \tag{7}$$

We define the first s bits of ciphertext as punctured ciphertext.

Hence, we can apply Algorithm 4 with input of r pairs of plaintext and punctured ciphertext by replacing initialization vector with plaintext, keystream with punctured ciphertext, and stream cipher encryption oracle with block cipher encryption oracle. Deduced by equation (7), it's no need to run preprocessing algorithm because of pseudo-randomness, i.e., each bit in ciphertext is uniformly distributed in 0-1 space.

All in all, the complexity of searching block cipher's key by Algorithm 4 is

$$\tilde{\mathcal{O}}\left(r \cdot 2^{\frac{s}{2}} \log\left(1/\delta\right) C_{block}\right), \tag{8}$$

and the qubits are

$$n + m + s + r + q,$$

where n represents key length, m represents block length, r represents the iteration or the number of auxiliary qubits, s represents punctured ciphertext length, q represents the number of extra qubits in the block cipher quantum circuit, δ represents the error tolerance in search algorithm by QSVT, and C_{block} represents the complexity of quantum block cipher circuit.

Parameters Selection The rules of parameters selection for block cipher are similar for stream cipher but a little different.

Let k' denote the n-bit key, k^* denote the correct key, and $(t_1, c_1'), (t_2, c_2')$ denote two pairs of plaintext and punctured ciphertext.

Definition 4. *For two pairs $(t_1, c_1'), (t_2, c_2')$, if the key k' satisfies $g_{k',s}(t_1) = g_{k^*,s}(t_1) = c_1'$, $g_{k',s}(t_2) = g_{k^*,s}(t_2) = c_2'$ and $k' \neq k^*$, then k' is called fake key.*

It's of great possibility for the existence of fake keys because of the short length of punctured ciphertext.

By the pseudo-randomness of function $g_{k,s}(\cdot)$,

$$\forall t \in \{0,1\}^m, \Pr_{k' \neq k^*}(g_{k',s}(t) = g_{k^*,s}(t)) = \frac{1}{2^s}.$$

Given r pairs of plaintext and punctured ciphertext $(t_1, c'_1), \cdots, (t_r, c'_r)$,

$$p = \Pr_{k' \neq k^*}(g_{k^*,s}(t_i) = g_{k',s}(t_i), i = 1, \cdots, r) = \prod_{i=0}^{r-1} \frac{1}{2^s - i}.$$

For the condition

$$r^2 \ll 2^s, \tag{9}$$

we have

$$p = \prod_{i=0}^{r-1} \frac{1}{2^s - i} \approx 2^{-rs}. \tag{10}$$

Set the fake key set as $FK = \{k' | k' \neq k^*, (g_{k^*,s}(t_i) = g_{k',s}(t_i), i = 1, \cdots, r)\}$. By equation (10), $|FK| \approx (2^n - 1)2^{-rs}$.

Let random variable X be the number of fake keys $|FK|$, where X follows the binomial distribution, and

$$\Pr(X = \alpha) = C^{\alpha}_{2^n-1} p^{\alpha} (1-p)^{2^n - 1 - \alpha}.$$

By equation (10),

$$\Pr(X = 0) = C^0_{2^n-1} p^0 (1-p)^{2^n-1} = (1-p)^{2^n-1} \approx e^{-2^{n-rs}}.$$

Hence, it is $p \approx e^{-2^{n-rs}}$ probability to return a unique and correct key.

In order to guarantee the success probability and use fewer pairs of plaintext and punctured ciphertext, we let the pairs r satisfy

$$r = \left\lfloor \frac{n}{s} + 1 \right\rfloor. \tag{11}$$

Above all, if the parameters r and s satisfy the inequality (9) and equation (11), we can guarantee that the probability of returning a unique key is $p \approx e^{-2^{n-rs}}$.

4 Implementation of Multi-round Key Search Model

To evaluate the specific searching complexity on symmetric ciphers, we implement our search model on block cipher AES family, two kinds of stream ciphers Grain-128 and ZUC-128.

As for stream ciphers Grain-128 and ZUC-128, we need to calculate quantum gates number in circuits of two stream ciphers with the initialization and keystream output process. As for block cipher AES, we have to figure out the complexity of quantum AES's encryption oracle.

4.1 Stream Cipher Grain-128

Stream Cipher Grain-128 Oracle The Grain-128 algorithm [7] is proposed in 2006, with input of 128-bit key and 96-bit initialization vector, and output of keystream with arbitrary length. The Grain-128 algorithm consists of two processes, 256-round initialization process and keystream output process. The specific components include linear feedback shift registers, nonlinear feedback shift registers, and filter function generators, etc.

The construction of the oracle for Grain-128 is as follows.

1. Initialization process

The state update equations on each component during initialization are as follows.

a. Linear feedback shift register(LFSR):

$$s_{i+128} = s_i + h(x) + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}.$$

b. Non-linear feedback shift register(NFSR):

$$\begin{aligned} b_{i+128} = s_i + b_i + h(x) + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + \\ b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}. \end{aligned}$$

c. Filter function:

$$h(x) = b_{i+12}s_{i+8} + s_{i+13}s_{i+20} + b_{i+95}s_{i+42} + s_{i+60}s_{i+79} + b_{i+12}b_{i+95}s_{i+95}.$$

We implement the Grain-128 algorithm into a quantum circuit by state update equations on each register, which is simpler and easier than constructed by feedback polynomials. It is known that the number of quantum gates required in one round is 36. And the number of quantum gates within 256 rounds of initialization is

$$Init(128) = 9216. \tag{12}$$

2. Keystream output process

Similarly, the state update equations are as follows:

a. Linear feedback shift register(LFSR):

$$s_{i+128} = s_i + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}.$$

b. Non-linear feedback shift register(NFSR):

$$\begin{aligned} b_{i+128} = s_i + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + \\ b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}. \end{aligned}$$

c. Filter function:

$$h(x) = b_{i+12}s_{i+8} + s_{i+13}s_{i+20} + b_{i+95}s_{i+42} + s_{i+60}s_{i+79} + b_{i+12}b_{i+95}s_{i+95}.$$

d. Keystream output:

$$k_i = h(x) + s_{i+93} + b_{i+2} + b_{i+15} + b_{i+36} + b_{i+45} + b_{i+64} + b_{i+73} + b_{i+89}.$$

By the keystream output equation, if the bit s of punctured keystream required in Algorithm 4 is smaller than 32, it is no need to update states on registers. At this time, the number of gates required to output 1 bit is 18. Else, if $s > 32$, there are $s - 32$ bits needed to update, and the number of gates required to output 1 bit is 42. Hence, the complexity of outputting s-bit keystream is

$$Output(s, 128) = \begin{cases} 18s \ , & s \leq 32 \\ 42s - 768 \ , & s > 32 \end{cases}. \tag{13}$$

Multi-round Key Search Model Attacking Effect on Grain-128 For stream cipher Grain-128, the key's length $n = 128$. By the inequality (3), it might be good that

$$1000 \cdot r^2 \leq 2^h. \tag{14}$$

Put equation (6) into (14), and get the solution $h \geq 17$. Let $h = 17$, and then $r = 8$. Hence, after constant times of Algorithm 2, get the optimal value of parameter s, where there are at least 17 bits uniformly distributed at the first s bits of the punctured keystreams. Denote $\delta = 0.01$ as the error tolerance of search algorithm, put it into equation (2), and the complexity is

$$\tilde{\mathcal{O}}\left(8 \cdot \log(1/0.01) \cdot (9256 + 18s) 2^{\frac{s}{2}}\right).$$

Hence, according to the quantitative relation between s and h, the attacking complexity against Grain-128 is shown in Table 2.

Table 2. Searching Complexity of the Stream Cipher Grain-128

s	h/s	Searching Complexity	Qubits
17	1	$\tilde{\mathcal{O}}(2^{27.5})$	280
19	0.9	$\tilde{\mathcal{O}}(2^{28.5})$	282
21	0.8	$\tilde{\mathcal{O}}(2^{29.5})$	284
24	0.7	$\tilde{\mathcal{O}}(2^{31.0})$	287
28	0.6	$\tilde{\mathcal{O}}(2^{33.0})$	291
32	0.5	$\tilde{\mathcal{O}}(2^{35.0})$	296

4.2 Stream Cipher ZUC-128

Stream Cipher ZUC-128 Oracle The ZUC-128 algorithm [9] is a synchronous stream cipher algorithm with input of a 128-bit key seed and a 128-bit initialization vector, and output of a 32-bit keystream at a time. The ZUC-128 algorithm consists of two processes.

1. Initialization process

Divide the key k and the initialization vector IV by 8 bits, where $k = k_0 \| k_1 \| \cdots \| k_{15}$, and $IV = IV_0 \| IV_1 \| \cdots \| IV_{15}$. Load them into linear feedback

shift registers, where $s_i = k_i \| d_i \| IV_i,\ 0 \leq i \leq 15$, and d_i is a 15-bit constant. Set memory unit variables $R_1 = R_2 = 0$, run Initialization process 32 rounds.(The output W of the nonlinear function F needs to round off the last 1 bit to participate in the state update process of the LFSR)

2. Keystream output process

After loading the key, the iterative process of bit-reorganization, nonlinear function, and LFSR state update is first executed in sequence, but no keystream is output. After that, the word keystream output process begins. Every iteration, a 32-bit (one word) keystream $z = W \oplus X_3$ is output.

As for the construction of the oracle for ZUC-128, a quantum circuit for the stream cipher ZUC-128 is designed in [14], where a round of initialization process requires 3000 Toffoli gates, 9488 CNOT gates and 736 Pauli X gates, the first round of operations in working mode executed after initialization requires 2754 Toffoli gates, 8849 CNOT gates, 672 Pauli X gates, and a round of operation in working mode (outputs a 32-bit keystream) requires 2754 Toffoli gates, 8913 CNOT gates, and 672 Pauli X gates.

To sum up, the complexity $Init(128)$ is 435443 of 32 rounds initialization processes and first round of operations in working mode, and the complexity of outputting 32-bit keystream is $Output(32, 128) = 12339$.

As for the qubits, 496 qubits hold each state on the linear feedback shift register, 64 qubits hold the two memory unit variables R_1 and R_2, s qubits hold the values of the output keystream, and 64 auxiliary qubits count.

Multi-round Key Search Model Attacking Effect on ZUC-128 For stream cipher ZUC-128, the key length $n = 128$, and one word keystream length is 32 at one time. Thus, $32|s$. Similarly, Algorithm 4 works well if $h \geq 17$. Set $s = 32$, and our search model needs $656 + \lfloor 128/h + 1 \rfloor$ bits of qubits. Run constant times of preprocessing algorithm 2, compute the number of uniformly distributed bits h (min-entropy) in the first 32 bits of the keystream. The relation between the min-entropy h and the attacking complexity can be seen in the following Fig. 7.

As for the worst case that $h = 17$, which means only 17 bits are uniformly distributed in the one word keystream, the attacking complexity is $\tilde{\mathcal{O}}\left(2^{40.5}\right)$, and the number of required quantum bits is 664. And as for the best case that $h = 32$, which means each bit is uniformly distributed in the one word keystream, the attacking complexity is $\tilde{\mathcal{O}}\left(2^{39.8}\right)$, and the number of required quantum bits is 661.

4.3 Block Cipher AES

Block Cipher AES Oracle The three main kinds in block cipher AES family [19] are AES-128, AES-192, and AES-256, whose key length is 128, 192, and 256 bits, and round is 10, 12, and 14, separately. Besides, all three algorithm encrypt with block length of 128 bits.

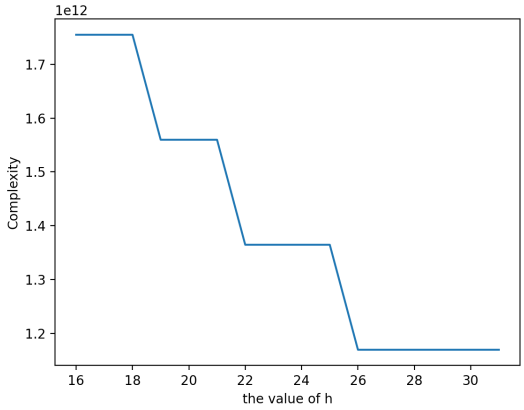

Fig. 7. Searching complexity against ZUC-128

Block cipher AES consists of a rounding function and key schedule, based on the substitution-permutation network structure. Firstly, there are three subroutines of a round function, SubBytes, ShiftRows, MixColumns, and AddRoundKey. Secondly, for key schedule, it consists of three subroutines, SubWord, RotWord, and Rcon. In SubBytes and SubWord subroutines, S-box substitution is applied to build up the whole encryption system's nonlinearity. And in ShiftRows and RotWord subroutines, some particular permutations are implemented by appropriate rewiring. As for MixColumns subroutine, a specific matrix is used to operate the entire column. In AddRoundKey subroutine, the bitwise XOR is operated of the 128-bit roundkey to the internal AES state. At last, Rcon is a round constant.

In [22], the authors design four kinds of quantum circuits for each AES-128/192/256 separately, which can be used as an oracle implemented in Grover's key search model and our search model as well. And we select one designed quantum circuit for each block ciphers in Table 3.

Table 3. Quantum Circuit of the Block Cipher AES-128/192/256

Block Cipher	Qubits	Toffoli Depth	Total Number of Gates
AES-128	400	1108	99824
AES-192	464	1340	115256
AES-256	528	1540	139919

Multi-round Key Search Model Attacking Effect on AES Like the section 3.2 discussed, we solve three inequalities

$$1000 \cdot \left\lfloor \frac{n}{s} + 1 \right\rfloor \leq 2^s,$$

where the key length of AES-128/192/256 is separately put into as the value of n. And we get

$$s \geq \begin{cases} 17, n = 128 \\ 18, n = 192 \\ 18, n = 256 \end{cases}.$$

For the sake of lower searching complexity, we choose the lower bound as the value of s, get the corresponding value of r by equation (11), and calculate the searching complexity of each block cipher by equation (8), as Table 4 shows.

It can be concluded from Table 4, increasing the key length doesn't enhance the security against our key search model.

Table 4. Searching Complexity of the Block Cipher AES-128/192/256

Block Cipher	s	r	Searching Complexity	Qubits
AES-128	17	8	$\tilde{\mathcal{O}}(2^{30.8})$	408
AES-192	18	11	$\tilde{\mathcal{O}}(2^{32.0})$	475
AES-256	18	15	$\tilde{\mathcal{O}}(2^{32.7})$	543

5 Conclusion

Our model takes use of the security property of symmetric ciphers. For stream ciphers, more random the initial states appears, more secure stream cipher gets, more bits of keystreams under the full key space distribute uniformly, smaller length s of punctured keystream has, and lower complexity of search model will be. Besides, for block ciphers, the pseudo-randomness is our standpoint, which guarantees each bit of ciphertext is uniformly distributed. Thus, our search model can function well on block ciphers without the preprocessing algorithm.

As for searching cost, the complexity is $\tilde{\mathcal{O}}\left(r \cdot 2^{\frac{s}{2}} \log\left(1/\delta\right) \cdot C\right)$, where n denotes key length, s denotes the length of punctured keystream/ciphertext, r denotes the iteration in our model, and C denotes the complexity of quantum stream/block cipher's encryption oracle. Besides, the required bits of qubits is $n + m + s + r + q$, where m denotes initialization vector/block length, r denotes the number of auxiliary qubits or the iteration, and q denotes the number of extra qubits in the quantum stream/block cipher's encryption oracle.

According to the parameter selection rules, the attacking complexity of our search model outperform Grover algorithm against symmetric ciphers. Furthermore, our search model shows that increasing the key seed length have little

influence on the resulting complexity. Thus, this common countermeasure to resist the quantum search attacks does not work anymore in the quantum computation environment. It is necessary to propose new design idea for symmetric ciphers in the future.

References

1. Katz, J., Lindell Y.: Introduction to modern cryptography. 3rd edn. CRC press, Boca Raton (2020)
2. Grover, L., K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pp. 212–219. Association for Computing Machinery, New York (1996)
3. Guang, H., L.: Quantum signal processing by single-qubit dynamics. Cambridge: Massachusetts Institute of Technology (2017)
4. Guang, H., L., Theodore J., Y., Isaac, L., C.: Methodology of resonant equiangular composite quantum gates. Physical Review X **6**(4), 041067 (2016)
5. Guang, H., L., Isaac, L., C.: Hamiltonian simulation by qubitization. Quantum **3**, 163 (2019)
6. András, G., Yuan, S., Guang, H., L., Nathan, W.: Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pp. 139–204. Association for Computing Machinery, New York (2019)
7. Hell, M., Johansson, T., Maximov, A., et al.: A stream cipher proposal: Grain-128. In: 2006 IEEE International Symposium on Information Theory, pp. 1614–1618. IEEE, Seattle (2006)
8. Dinur, I., Shamir, A.: Breaking Grain-128 with dynamic cube attacks. In: Fast Software Encryption: 18th International Workshop, pp. 167-187. Springer, Berlin Heidelberg (2011)
9. Xiutao, F.: The ZUC Stream Cipher Algorithm. Journal of Information Security Research **2**(11), 1028–1041 (2016)
10. Michael, A., N., Isaac, L., C.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2010)
11. Martyn, J. M., Rossi, Z., M., Tan, A., K., et al.: Grand unification of quantum algorithms. PRX Quantum **2**(4), 040203 (2021)
12. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM Journal on Computing **17**(2), 230-261 (1988)
13. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing Grover Oracles for Quantum Key Search on AES and LowMC. In: Canteaut, A., Ishai, Y. (eds) Advances in Cryptology - EUROCRYPT 2020. EUROCRYPT 2020. Lecture Notes in Computer Science(), vol 12106. Springer, Cham (2020)
14. Zhuang, S.: Quantum circuit implementations of symmetric ciphers. Institute of Information Engineering, Chinese Academy of Sciences, Beijing (2021)
15. Bonnetain, X., Schrottenloher, A., Sibleyras, F.: Beyond Quadratic Speedups in Quantum Attacks on Symmetric Schemes. In: Dunkelman, O., Dziembowski, S. (eds) Advances in Cryptology - EUROCRYPT 2022. EUROCRYPT 2022. Lecture Notes in Computer Science(), vol 13277. Springer, Cham (2022)
16. Grassl M., Langenberg B., Roetteler M., Steinwandt R.: Applying Grover's algorithm to AES: quantum resource estimates. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 29–43. Springer, Heidelberg (2016).

17. Almazrooie M., Samsudin A., Abdullah R., Mutter K N.: Quantum reversible circuit of AES-128. Quantum Inf **17**(5), 1–30 (2018)
18. Guajardo J., Paar C.: Itoh-Tsujii inversion in standard basis and its application in cryptography and codes. Codes Cryptogr **25**(2), 207 (2002)
19. Rijmen, V., Daemen, J.: in Proceedings of Federal Information Processing Standards Publications 197 (National Institute of Standards and Technology, Springfield, 2001)
20. Grassi, L.: Probabilistic Mixture Differential Cryptanalysis on round-reduced AES. In: Paterson, K., Stebila, D. (eds) Selected Areas in Cryptography - SAC 2019. SAC 2019. Lecture Notes in Computer Science(), vol 11959. Springer, Cham (2020).
21. Grassi, L.: Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES. IACR Cryptology ePrint Archive, 2017:832, (2017)
22. Li, Z., Cai, B., Sun, H. et al.: Novel quantum circuit implementation of Advanced Encryption Standard with low costs. Sci. China Phys. Mech. Astron. 65, 290311 (2022)