Efficient bit sifting scheme of post-processing in quantum key distribution

Qiong Li · Dan Le · Xianyan Wu · Xiamu Niu · Hong Guo

Abstract Bit sifting is an important step in the post-processing of quantum key distribution (QKD). Its function is to sift out the undetected original keys. The communication traffic of bit sifting has essential impact on the net secure key rate of a practical QKD system. In this paper, an efficient bit sifting scheme is presented, of which the core is a lossless source coding algorithm. Both theoretical analysis and experimental results demonstrate that the performance of the scheme is approaching the Shannon limit. The proposed scheme can greatly decrease the communication traffic of the post-processing of a QKD system, which means the proposed scheme can decrease the secure key consumption for classical channel authentication and increase the net secure key rate of the QKD system, as demonstrated by analyzing the improvement on the net secure key rate. Meanwhile, some recommendations on the application of the proposed scheme to some representative practical QKD systems are also provided.

Keywords Quantum cryptography · Quantum key distribution · Post-processing · Bit sifting · Source coding · Unconditionally secure authentication · Net secure key rate

© Springer Science+Business Media New York 2015

Xiamu Niu
xm.niu@hit.edu.cn

1 Information Countermeasure Technique Institute, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150080, China

2 The State Key Laboratory of Advanced Optical Communication, Systems and Networks, School of Electronics Engineering and Computer Science and Center for Quantum Information Technology, Peking University, Beijing 100871, China
1 Introduction

The quantum key distribution (QKD) is the most developed branch of quantum cryptography, and its security is based on the principles of quantum mechanics. QKD can not only enhance the security of traditional symmetric/asymmetric cryptographic systems, but also construct an information-theoretic secure cryptographic system by combining with Vernam one-time pad cipher \[1\]. QKD comprises two phases: the transmission of the photons over the quantum channel and the post-processing over the authenticated classical channel. In the first phase, by transmitting the modulated photons, Alice and Bob obtain a partially shared bit string, so-called original key. A representative high-performance QKD system can transmit original keys at rates in the order of Gbps. In the second phase, by performing sifting, parameter estimation, error reconciliation and privacy amplification in an authenticated classical channel, Alice and Bob obtain the identical and unconditionally secure key, so-called secure key. The highest secure key rate is about 1 Mbps according to the published literatures \[2,3\]. The estimation methods of secure key rates for different QKD protocols are similar. Taking BB84 protocol as an example, the secure key rate \( R \) is given by \[4\]

\[
R = \max \left\{ g_p Q_\mu \left[ -f \left( E_\mu \right) H_2 \left( E_\mu \right) + \Delta_1 \left( 1 - H_2 \left( e_1 \right) \right) \right], 0 \right\},
\]

where \( g_p \) is the protocol gain, which is 0.5 for the classical BB84 protocol \[5\], \( Q_\mu \) is the count rate, \( f \left( E_\mu \right) \) is the reconciliation efficiency, and \( H_2 \left( x \right) = -x \log_2 \left( x \right) - \left( 1 - x \right) \log_2 \left( 1 - x \right) \). One of the basic assumptions of the security analysis for QKD protocols is that there is an authenticated classical channel between Alice and Bob \[6–8\]. However, the classical channel in a QKD system cannot be authenticated by itself unless an unconditionally secure authentication algorithm is employed to authenticate all interactive messages between Alice and Bob. The algorithm is usually based on the family of almost strongly universal hash functions at the cost of some key consumption. For the first round of the QKD system, a pre-shared key must be available, which is exchanged through a secret channel, such as face to face or other ways. For the following rounds, a part of the secure key generated by the QKD system is used as the authentication key. Since the above formula does not consider the secure key consumption due to the authentication of classical channel, the net secure key rate \( R' \) after the withdrawal of the authentication consumption, which is also named the authenticated key rate in \[9\], is written as follows,

\[
R' = R - K_{\text{Aut}}
\]

\[
K_{\text{Aut}} = K_{\text{Aut} - \text{Sift}} + K_{\text{Aut} - \text{PE}} + K_{\text{Aut} - \text{ER}} + K_{\text{Aut} - \text{PA}},
\]

where \( K_{\text{Aut}} \) is the total secure key consumption of authentication and \( K_{\text{Aut} - \text{Sift}} \), \( K_{\text{Aut} - \text{PE}} \), \( K_{\text{Aut} - \text{ER}} \) and \( K_{\text{Aut} - \text{PA}} \) are the secure key consumption of authentication during sifting procedure, parameter estimation procedure, error reconciliation procedure and privacy amplification procedure, respectively. Besides, \( K_{\text{Aut} - \text{Sift}} \) is much larger than the others, for instance, \( K_{\text{Aut} - \text{Sift}} \geq 0.94 \cdot K_{\text{Aut}} \) in the QKD system of \[9\]. In order to improve \( R' \), many works have been done on the methods to decrease...
Table 1 Secure key consumptions of some representative authentication algorithms for given security parameter $\varepsilon$ and message length $m$

| Authentication algorithm          | Secure key consumption                                      |
|----------------------------------|-------------------------------------------------------------|
| den Boer [15]                    | $\approx -2\log_2 \varepsilon + 2\log_2 m$                |
| Bierbrauer etc. [16]             | $\approx -3\log_2 \varepsilon + 2\log_2 m$                |
| Krawczyk [17]                    | $-3\log_2 \varepsilon + 3\log_2 (1 + 2m) + 1$              |
| Abidin etc. [18]                 | $-4\log_2 \varepsilon + 3\log_2 m + 8$                     |

$f(E_{\mu})$ since the mid-1990s [10–14], while this study is trying to decrease $K_{\text{Aut-Sift}}$ to improve $R'$, which has not been paid too much attention so far.

The secure key consumptions of some representative authentication algorithms are listed in Table 1, as functions of the security parameter and the authenticated message length. It can be found that the secure key consumption monotonically increases with the message length $m$. Therefore, the communication traffic during sifting procedure should be reduced as much as possible.

In 2010, in order to save communication traffic, Koll-mitzer et al. stated that Bob could inform Alice of the detection position represented by $\lceil \log_2 m \rceil$ bits, where $m$ is the number of original keys to be processed [19]. The scheme can reduce the amount of exchanged messages to some extent, but the compression efficiency is far from the optimum. This scheme was implemented by Li et al. [20]. In 2014, Walenta et al. [9] pointed out that the amount of bits exchanged during sifting should be kept as small as possible due to the authentication cost. Their sifting scheme is to encode the detection time indexes between two adjacent detection events at Bob side. Their compression efficiency is less than twice the Shannon limit when the count rate is between $10^{-4}$ and $10^{-1}$, while the performance falls sharply when the count rate is out of this range. The theoretical analysis and experimental results demonstrate that the proposed scheme always performs better than the scheme in [9].

The sifting procedure consists of bit sifting sub-procedure and basis sifting sub-procedure. The function of bit sifting is to get rid of the undetected original keys, and the basis sifting aims to sift out the original keys whose bases are incompatible. On one hand, the amount of interactive messages of bit sifting is far more than that of basis sifting. On the other hand, from the point of information theory, the great redundancy due to the very low count rate makes it possible to significantly decrease the amount of interactive messages of bit sifting. On the contrary, the interactive messages during basis sifting can hardly be compressed because of the low redundancy. Hence the key point to decrease the communication traffic of sifting procedure is to reduce the amount of interactive messages during bit sifting step to a maximum extent, and this study only focuses on bit sifting.

The method to decrease the communication traffic of sifting should not only have good compression performance but also be performed as fast as possible so that Alice could remove the undetected original keys from her buffer in time. With the fast increase in the repetition frequency of quantum channel, post-processing devices are facing unprecedented challenges. Taking BB84 protocol as an example, the input data rate of sifting procedure at Alice side is twice the repetition frequency and three times at Bob side. For decoy BB84 protocol [21], the input data rate of sifting is even more
than that of BB84. To date, the repetition frequency of a high-speed QKD system has been up to about 10 GHz [2], so the post-processing devices are facing huge storage pressure. In 2007, Mink indicated that the storage of sifting was one bottleneck for his QKD system [22], whose repetition frequency is 3.125 Gbps. It is noted that although the input data rate at Bob side is more than that at Alice side, Bob can immediately sift out the undetected original keys whose amount is far more than the amount of the detected, while Alice cannot sift out them until Bob announces which original keys he has detected, so the storage pressure of the device at Alice side is much heavier.

In this paper, we firstly present a lossless source coding-based bit sifting scheme. Considering the expected codelength of the source coding algorithm as the optimization object, an efficient iteration algorithm is proposed to solve the optimization problem. Both theoretical analysis and experimental results demonstrate that the performance of our scheme is approaching the Shannon limit and also better than the competitive scheme within the entire reasonable interval of count rate. Besides, the improvement of our scheme on $R'$ is analyzed for the QKD system in [9], and some suggestions on how to apply our scheme to some representative practical QKD systems are provided. Although some special QKD protocols are mentioned as examples, any existing single-photon QKD protocols that contain the sifting procedure can benefit from our works.

The rest of this paper is organized as follows. In Sect. 2, the proposed lossless source coding-based bit sifting scheme and the theoretical analysis on its performance are discussed in detail. The experimental results and analysis are presented in Sect. 3. Finally, some conclusions are drawn in Sect. 4.

2 Proposed bit sifting scheme

2.1 Description of bit sifting scheme

The schematic diagram of the proposed bit sifting scheme with the preceding and following steps is shown in Fig. 1. A basic QKD protocol starts with the preparation, transmission and detection of a random sequence-modulated photons, also called qubits or quantum states, which are transferred into the original key at both sides. The original key constitutes the input of QKD post-processing system. Since a large fraction of qubits cannot be detected due to the loss of the transmission and the imperfection of the detection device, Bob needs to announce the detected validity of each original key. As mentioned above, the data amount of the announcements is extremely large, which requires a huge secure key consumption for the corresponding authentication. In order to save the secure key consumption, a source encoder and decoder are designed in our bit sifting scheme at Bob and Alice sides, respectively. The optimal compression performance of the source coding algorithm is pursued to minimize the secure key consumption for the authentication of bit sifting. Since Alice has to buffer all original keys until she receives the validity announcement from Bob, the storage pressure would be too much to bear if the source encoding and decoding cannot be implemented in real time. Therefore, another desired performance of the source coding algorithm is low computation complexity.
2.2 Description of the MZRL source coding algorithm

Generally, the announcement is a binary string, in which the value of each bit indicates the detected validity of the corresponding original key. Without loss of generality, we assume that “0” represents the case of undetected, and “1” represents the case of detected. Since the number of photons in one pulse, the noise of quantum channel and the response of detection device are all almost random, the detected validity is nearly random. So the announcement of detected validities can be considered as a binary memoryless information source which is just the object that we need to compress via a source coding.

Considering that the number of “0” in the binary string is far more than the number of “1,” a modified zero run length (MZRL) source coding algorithm is designed. First of all let us recall the traditional zero run length coding. Suppose that there is a binary string “0010001100000001,” the traditional zero run length coding result would be “2-3-0-7.” Such simple coding algorithm is not completely suitable in the context of QKD. Since a QKD system may run continuously, the length of zero run may be any element from the set of natural number, i.e., \{0, 1, 2, \ldots, +\infty\}. That is to say, the binary information source is transferred to a non-binary source with infinite and countable source letters. While it is not realistic to represent infinite numbers in a practical system, in many cases, the run lengths larger than a preset threshold are truncated because the probabilities of the big run lengths are usually so small that they can be neglected in some error-tolerant applications. But such truncation scheme does not fit for QKD since “lossless” is the basic requirement for the bit sifting scheme and any error is not acceptable.

To losslessly represent infinite possible run lengths by using finite resources, we design the MZRL algorithm based on a straightforward and efficient idea, i.e., segmentation. The encoding schematic diagram of MZRL algorithm is shown in Fig. 2.

The function of the Source Parser in Fig. 2 is to divide the binary source output sequence into messages, which are the objects to be assigned codewords by the Mes-
Fig. 2  Encoding schematic diagram of MZRL codes

Table 2  MZRL codes

| Source message | Codeword |
|----------------|----------|
| \( s_0 = "1" \) | \( c_0 = 00\ldots0 \) |
| \( s_1 = "01" \) | \( c_1 = 00\ldots1 \) |
| \( \vdots \) | \( \vdots \) |
| \( s_{n-2} = "00\ldots01" \) | \( c_{n-2} \) |
| \( n-2 \) '0's | \( n-2 \) '0's |
| \( s_{n-1} = "00\ldots00" \) | \( c_{n-1} \) |
| \( n-1 \) '0's | \( n-1 \) '0's |

**sage Encoder.** In Fig. 2, the output of **Source Parser** is \( n \) variable-length messages. The function of the **Message Encoder** is to map each message into a codeword. To simplify the decoding operation, the length of every codeword is set to \( \lceil \log_2 n \rceil \). The output messages of the **Source Parser** and their corresponding codewords produced by the **Message Encoder** in the MZRL coding are presented in Table 2.

As shown in Table 2, the definition of messages is the same as the traditional zero run length coding except the \( n \)th message \( s_{n-1} \). The first \( n-1 \) messages \( s_i \) follow the same pattern whose length is \( i+1 \) and the last digit is 1. But the \( n \)th message \( s_{n-1} \) is a sequence of all 0s of length \( n-1 \). It is obvious that \( n \) must be greater than or equal to 2.

It is clear that the codeword \( c_0 \) to \( c_{n-2} \) can represent the run length from 0 to \( n-2 \). How to represent a run length that is greater than or equal to \( n-1 \) is the next problem we need to solve. Our method is to segment the long binary sequence into one or more \( s_{n-1} \)'s and one message \( s_i \) (0 ≤ \( i \) ≤ \( n-2 \)), which can be represented by \( c_{n-1} \) and \( c_i \) (0 ≤ \( i \) ≤ \( n-2 \)), respectively. That is to say, for an arbitrary zero run length \( RL(0) = m \ast (n-1)+i \), where \( m, i \in \mathbb{N} \) and 0 ≤ \( i \) ≤ \( n-2 \), the codeword sequence is \( c_{n-1}c_{n-1}c_{n-2}\ldotsc_{n-1}c_i \). For example, if \( n = 4 \), the MZRL codeword sequence for the binary string “0010001100000001” is “c_2c_3c_0c_0c_3c_3c_1.”

According to Table 2, both encoding and decoding are quite simple and efficient. For the encoder, the **Source Parser** stores letters from the **Binary Source** until it sees that these letters form a valid message as defined in Table 2 and the **Message Encoder** outputs the corresponding codeword. For the decoder, it decodes the received codeword \( c_i \) to the corresponding message \( s_i \) which is just the final output of the decoder. Since MZRL is a non-singular fix-length code, it is by nature an instantaneous code, which means the end of a codeword is immediately recognizable and a codeword can be decoded without reference to future codewords. Such property makes the decoding of MZRL more efficient.

The simple encoding and decoding principles of MZRL guarantee that the algorithm can be implemented very fast. Except the computation complexity, what we care about
most in the scenario of QKD is the compression performance of the source coding algorithm. Hence, in the following sections, we will explore the optimal expected code length of MZRL codes.

2.3 Expected code length of MZRL codes

Suppose that the count rate of a QKD system is \( q \), which means the probability of “1” and “0” is \( q \) and \( 1 - q \) for the binary source \( X \) in Fig. 2. It is easy to deduct that the probability of the zero run length \( l \) is

\[
P(l) = (1 - q)^l q.
\] (2)

For any given \( i \in \{0, 1, \ldots, n - 2\} \), the message \( s_i \) only appears once when the zero run lengths \( l \in \{l | l = m(n - 1) + i, m \in \mathbb{N}\} \). Therefore \( P(s_i) \) is given by

\[
P(s_i) = \sum_{m=0}^{+\infty} P(m(n - 1) + i)
= \frac{(1 - q)^{i+1} q}{1 - (1 - q)^n - q}, \quad \forall i \in \{0, 1, \ldots, n - 2\}.
\] (3)

While the message \( s_{n-1} \) would appear \( \lfloor \frac{l}{n-1} \rfloor \) times when the zero run lengths \( l \in \{l | l \geq n - 1 \cap l \in \mathbb{N}\} \). So \( P(s_{n-1}) \) is given by

\[
P(s_{n-1}) = \sum_{l=n-1}^{+\infty} \left\lfloor \frac{l}{n-1} \right\rfloor P(l).
\] (4)

Since \( P(l) \geq 0 \) and

\[
P(s_{n-1}) = \sum_{l=n-1}^{+\infty} \left\lfloor \frac{l}{n-1} \right\rfloor P(l)
\leq \sum_{l=0}^{+\infty} l P(l)
= \frac{1}{q} - 1,
\] (5)

the series \( P(s_{n-1}) \) in the (4) converges absolutely [23]. Then any rearranged series of the series \( P(s_{n-1}) \) also converges absolutely to the same sum. In order to compute the sum of \( P(s_{n-1}) \), it is rearranged as follows,

\[
P(s_{n-1}) = \sum_{m=1}^{+\infty} m \sum_{k=0}^{n-2} P(m(n - 1) + k)
= \frac{(1 - q)^n}{1 - (1 - q)^n - q}.
\] (6)
Since the codelength of the codeword $c_i$ corresponding to the message $s_i$ is a constant $\lceil \log_2 n \rceil$ and the probability mass function $P(s_i)$ is given by (3) and (6), the expected codelength of MZRL code $C$ for the $n$-ary source, i.e., the random variable $S$, is given by

$$\bar{L}_C = \sum_{i=0}^{n-1} P(s_i) \lceil \log_2 n \rceil$$

$$= \frac{\lceil \log_2 n \rceil}{1 - (1 - q)^{n-1}}.$$  \hspace{1cm} (7)

To obtain the expected codelength for the binary source, i.e., the random variable $X$, we also need to compute the average length of the source message of $S$ by (8)

$$\bar{L}_S = \sum_{i=0}^{n-2} P(s_i) (i + 1) + P(s_{n-1}) (n - 1)$$

$$= \frac{1}{q}.$$  \hspace{1cm} (8)

Hence the expected codelength for the binary source $X$ is

$$\bar{L}(n) = \frac{\bar{L}_C}{\bar{L}_S}$$

$$= \frac{q \lceil \log_2 n \rceil}{1 - (1 - q)^{n-1}}.$$  \hspace{1cm} (9)

According to (9), the expected codelength is an expression of $n$, which is the size of code alphabet of MZRL, and the count rate $q$. For a QKD system, $n$ is a parameter that should be adjusted carefully depending upon the requirements and available resources, while the count rate $q$ is almost constant. To analyze the optimization of the expected codelength, we need to confine the possible range of count rate. In general, the count rate $q$ is determined by the mean photon number $\mu$, the fiber loss coefficient $\alpha$, the distance $d$ between two parties, the inner loss $\gamma_{\text{Bob}}$ of the optical devices of Bob, the detection efficiency $\eta_D$ of Bob’s detector and the dark count rate $P_d$. The relationship is given by

$$q = 1 - e^{-\mu \cdot 10^{-(\alpha \cdot d + \gamma_{\text{Bob}})/10} \cdot \eta_D} + P_d.$$  \hspace{1cm} (10)

Table 3 illustrates the typical value of these parameters above[24]. To the best of our knowledge, the current maximal communication distance is about 260 km [25,26], in which case the count rate $q$ is about $10^{-6}$. Besides, the count rates of most practical QKD systems are always less than 0.1 [27]. So it is reasonable to set the range of count rate $q$ as $[10^{-15}, 10^{-1}]$, which covers all possible values of current QKD systems. So the goal is to minimize the value of $\bar{L}$ under the constraint $q \in [10^{-15}, 10^{-1}]$, i.e., solve the optimization problem (11),
Table 3 Typical parameters related to the count rate of QKD

| $\mu$ | $\alpha$ (dB/km) | $d$ (km) | $\gamma_{\text{Bob}}$ (dB) | $\eta_{D}$ (%) | $P_{d}$ |
|-------|------------------|--------|--------------------------|--------------|-------|
| 0.5   | 0.2              | 0–260 | 4                         | 10           | 10$^{-5}$ |

\[
\min_{n} L(n) = \frac{q \left\lceil \log_2 n \right\rceil}{1-(1-q)^{n-1}} \\
s.t. n \in \mathbb{N} \\
\quad n \geq 2 \\
\quad q \in [10^{-15}, 0.1] \tag{11}
\]

2.4 Optimization of the expected code length

For simplicity of expression, $g(n)$ is used to denote $1 - (1 - q)^{n-1}$, then \( L(n) \) can be rewritten as

\[
L(n) = \frac{q \left\lceil \log_2 n \right\rceil}{g(n)}.
\]

It is easy to conclude that $g(n)$ is monotonically increasing function with respect to the variable $n$ for any given $0 < q < 1$, so

\[
g(n) \leq g\left(2^{k}\right), \quad \forall n \in \left(2^{k-1}, 2^{k}\right) \cap \mathbb{N}^+, k \in \mathbb{N}^+.
\]

Besides, the value of the following expression

\[
q \left\lceil \log_2 n \right\rceil
\]

is invariant in the range $n \in \left(2^{k-1}, 2^{k}\right) \cap \mathbb{N}^+$. So

\[
L(n) \geq L(2^{k}), \quad \forall n \in \left(2^{k-1}, 2^{k}\right) \cap \mathbb{N}^+. \tag{12}
\]

Therefore we only need to consider the function values at $2^{k}$, and the optimization problem (11) is equivalent to

\[
\min_{k} L(k) = \frac{q^{k}}{1-(1-q)^{2^{k-1}}} \\
s.t. \quad k \in \mathbb{N}^+ \\
\quad q \in [10^{-15}, 0.1] \tag{13}
\]

To explore the properties of the function $L(k)$ with respect to the variable $k$, the domain of $k$ is extended from $\mathbb{N}^+$ to real numbers no less than 1. That is

\[
L(z) = \frac{q z}{1 - (1 - q)^{2z-1}}, \quad z \in [1, +\infty). \tag{14}
\]
Lemma 1 For any given $q \in (0, 0.1]$, there exists a constant $z_0 \in \left( -\log_2(-\ln (1-q)), +\infty \right)$ satisfying that the function $L(z)$ monotonically decreases in the domain $z \in [1, z_0)$, monotonically increases in the domain $z \in (z_0, +\infty)$ and reaches the global minimum at the point $z = z_0$. In other words,

$$
\begin{align*}
\frac{\partial L}{\partial z} &< 0, \text{ when } z \in [1, z_0) \\
\frac{\partial L}{\partial z} &= 0, \text{ when } z = z_0 \\
\frac{\partial L}{\partial z} &> 0, \text{ when } z \in (z_0, +\infty)
\end{align*}
$$

Then it can be inferred that the optimal solution of (13) is reached at $k = \lfloor z_0 \rfloor$ or $k = \lceil z_0 \rceil$.

Proof Please refer to “Appendix A.” \qed

An example curve of $L(z)$ demonstrating the Lemma 1 is shown as Fig. 3, where $q = 0.05$.

So far, the existence of the optimal solution of (13) has been proved, and the optimal parameter $k_{opt}$ has been determined as $\lfloor z_0 \rfloor$ or $\lceil z_0 \rceil$. So the next task is to solve the key value $z_0$.

Theorem 1 For any given $q \in [10^{-15}, 0.1]$, the point $z_0$ which leads to the global minimum of function $L(z)$ is bounded by

$$y < z_0 < y + 3,$$

where $y = -\log_2(-\ln (1-q))$. Then the optimal parameter $k_{opt}$ of (13) is one of the following five values, i.e., $\lfloor y \rfloor$, $\lfloor y \rfloor + 1$, $\lfloor y \rfloor + 2$, $\lfloor y \rfloor + 3$, $\lceil y \rceil + 3$.

Proof Please refer to “Appendix B.” \qed
Subsequently, an efficient iterative solution of (13) can be presented according to Theorem 1, which is called Algorithm 1.

**Algorithm 1** The solution of optimization problem (13).

**Require:** Count rate $q$.

**Ensure:** The optimal solution $L_{opt}$; the optimal parameter $k_{opt}$.

1: $k = \left\lfloor -\log_2 (-\ln (1 - q)) \right\rfloor$.
2: $L_1 = L(k)$.
3: while (1) do
4:  $k = k + 1$.
5:  $L_2 = L(k)$.
6:  if ($L_1 \leq L_2$) then
7:    break.
8:  else
9:    $L_1 = L_2$.
10:  end if
11: end while
12: $L_{opt} = L_1$.
13: $k_{opt} = k - 1$.

For convenience, the steps 3–11 of Algorithm 1 are called one iteration. It is obvious that the algorithm converges within five iterations for any given $q \in [10^{-15}, 0.1]$. Some count rates $q$ between $10^{-6}$ and $10^{-1}$ are chosen as the input of Algorithm 1, and the corresponding numbers of iterations are demonstrated in Fig. 4. The experimental results show that the largest number of iterations is 4 and the average number of iterations is 3.28.

To sum up, the optimal solution of (11) stated in the previous section is solved when the size of codeword alphabet $n = 2^{k_{opt}}$. 
2.5 Theoretical performance analysis

In the following sections, we will analyze the performance of the proposed scheme in terms of compression efficiency, time complexity and space complexity.

2.5.1 Compression efficiency

The compression efficiency is denoted as \( f \) and defined as \( f = \frac{\bar{L}}{H(X)} \), where \( \bar{L} \) is the expected codelength and \( H(X) \) is the entropy of the source \( X \). Since the theoretical lower bound of \( \bar{L} \) is \( H(X) \) \[28\], the compression efficiency \( f \) is always greater than or equal to 1. Besides, the smaller \( f \) indicates the better compression performance.

To compute the compression efficiency \( f \), first of all we need to calculate the expected codelength \( \bar{L} \). One hundred different count rates are selected in the range \([10^{-6}, 10^{-1}]\) on the logarithmic scale. The optimal codelength of n-ary source \( S \), i.e., \( k_{opt} \), is computed via Algorithm 1 for each count rate, and the corresponding optimal size of codeword alphabet \( n_{opt} \) is \( 2^{k_{opt}} \). The expected codelength \( \bar{L}(n) \) can be hereby computed according to (9). The entropy of binary source is \( H(X) = -n \log_2 q - (1 - n) \log_2 (1 - q) \), denoted as \( h(q) \). So far, the compression efficiency \( f = \frac{\bar{L}}{h(q)} \) can be obtained. The theoretical results of \( k_{opt}, \bar{L} \) and \( f \) are shown as Figs. 5, 6, 7, respectively.

The compression efficiency \( f \) is less than 1.10 during the whole domain of count rate. So it is demonstrated that the compression performance of our MZRL source coding is very close to the Shannon’s limit.

2.5.2 Time complexity

In this section, we will analyze the time complexity of the proposed bit sifting scheme in Fig. 1.

![Fig. 5 Optimal codelength of \( S k_{opt} (q) \)](image-url)
Bit sifting at Bob side consists of the Undetected Bit Removal and the Source Encoder. For each original_key $B$, the Undetected Bit Removal determines whether it is a valid detection or not and outputs the detected validity to the Source Encoder. The Source Parse of the Source Encoder in Fig. 2 determines whether a valid message $s_i$ is formed, i.e., the current input detected validity is “1” or the current zero counter $i$ is equal to $n - 1$. If not, $i = i + 1$. Otherwise, the message $s_i$ is output to the Message Encoder which outputs the corresponding codeword $c_i$, and the zero counter $i$ is reset to 0. Since the time complexity of both the Undetected Bit Removal and the Source Encoder is constant, the time complexity of bit sifting for each original_key $B$ is also
constant. Hence, when \( m \) original_key\(_B\) are input, the time complexity of bit sifting at Bob side is \( O(m) \).

Once receiving a codeword \( c_i \), the Source Decoder at Alice side in Fig. 1, whose time complexity is constant, decodes it to the corresponding message \( s_i \) and outputs the \( s_i \) to the Undetected Bit Removal. If \( i \) is equal to \( n - 1 \), the Undetected Bit Removal discards \( n - 1 \) consecutive original_key\(_A\) from the Buffer. Otherwise the Undetected Bit Removal discards the former \( i \) original_key\(_A\) and reserves the \( i + 1 \)th as a raw_key\(_A\). Therefore the time complexity of bit sifting for the input codeword \( c_i \) is \( O(i) \). Assuming that the received \( w \) codewords are \( c_{i_0}, c_{i_1}, \ldots, c_{i_{w-1}} \), the time complexity of bit sifting for the input codeword \( c_i \) is \( O(\sum_{j=0}^{w-1} i_j) \). In fact, \( \sum_{j=0}^{w-1} i_j \) is the number of processed original_key\(_A\) so the time complexity of bit sifting at Alice side is also \( O(m) \).

In summary, the time complexity of the bit sifting at both Alice and Bob sides is linearly dependent on the number of original keys \( m \).

### 2.5.3 Space complexity

In this section, we will analyze the space complexity of the proposed bit sifting scheme in Fig. 1. Here, we assume that Bob does not cache more than one codeword but send a codeword to Alice as soon as it is formed. In this case, the bit sifting at Bob side only needs to store two temporary variables, i.e., one zero counter and one codeword. Both of them are represented by \( \lceil \log_2 n \rceil \) bits, so the space complexity of bit sifting at Bob side is \( O(\log_2 n) \).

Since Alice has to store the original_key\(_A\) in the Buffer till she receives a codeword carrying the detected validity of the original_key\(_A\) from Bob, the required storage consists of some temporary variables and the Buffer to store original_key\(_A\). The temporary variables are the received codeword \( c_i \) and the message \( s_i \) which are both represented by \( \lceil \log_2 n \rceil \) bits. While the size of the Buffer depends on the time difference \( t_{\text{diff}} \) between the time when an original_key\(_A\) is stored in the Buffer and the time it is removed from the Buffer by the Undetected Bit Removal. According to the MZRL codes, Alice has to send Bob at most \( n - 1 \) qubits to form a codeword. So the maximum time difference is

\[
 t_{\text{diff}} = (n - 2) t_{\text{rf}} + t_2 + t_3 + t_4 + t_5. \tag{16}
\]

\((n - 2) t_{\text{rf}} \) means the time that Alice prepares \( n - 1 \) qubits, where \( t_{\text{rf}} \) is the reciprocal of the repetition frequency of QKD.

\( t_2 \) is the time that the \( n - 1 \)th qubit is transmitted from Alice to Bob over quantum channel, which depends on the distance \( d \) between Alice and Bob.

\( t_3 \) is the time that the bit sifting at Bob side processes the \( n - 1 \)th original_key\(_B\), which is a constant according to the analysis in Sect. 2.5.2. By then, the codeword \( c_{n-2} \) or \( c_{n-1} \) is formed.

\( t_4 \) is the time that the codeword is transmitted from Bob to Alice over authenticated classical channel, which also depends on the distance \( d \).

\( t_5 \) is the time that the Source Decoder at Alice side decodes the codeword to the corresponding message, which is also a constant according to the analysis in Sect. 2.5.2. So far, the Undetected Bit Removal can begin to discard these \( n - 1 \) original_key\(_A\) from the Buffer.
Hence the number of the cached original_key_A in the Buffer is

\[
\frac{t_{\text{diff}}}{t_{\text{rf}}} + 1 = (n - 1) + \frac{t_2 + t_3 + t_4 + t_5}{t_{\text{rf}}},
\]

which is \( O(n) \). Therefore, the space complexity of bit sifting at Alice side is \( O(n) \).

Since the size of code alphabet \( n \) is exponentially dependent on the optimal parameter \( k_{\text{opt}} \), the required storage at Alice side may be very large. For instance, let \( q = 10^{-6} \), then the optimal parameter \( k_{\text{opt}} \) is 22 according to Algorithm 1, and the required storage at Alice side is multiple times of 4Mb. The times depend on the protocol of the QKD systems. In fact, memory resource sometimes may be very expensive, such as FPGA-based QKD system [9,25,29]. Although the storage of FPGA can be extended by attaching several SRAMs or DDRs, the performance of SRAM or DDR is not as good as the inner storage of FPGA. In the case of limited storage resource, the optimization problem (11) can be rewritten as

\[
\begin{align*}
\min_n & \quad \overline{L}(n) = q \left\lceil \frac{\log_2 n}{1-(1-q)^{n-1}} \right\rceil, \\
\text{s.t.} & \quad n \in \mathbb{N}, \\
& \quad n_{\text{max}} \geq n \geq 2 \\
& \quad q \in [10^{-15}, 0.1],
\end{align*}
\]

(17)

where \( n_{\text{max}} \) is the possible maximal code alphabet size, which can be evaluated according to the available storage size. The optimal solution of (17) is stated in Theorem 2.

**Theorem 2** For any given \( q \in [10^{-15}, 0.1] \) and \( n_{\text{max}} \), if \( n_{\text{max}} \geq 2^{k_{\text{opt}}} \) then the optimal solution of (17) is reached at \( n = 2^{k_{\text{opt}}} \). Otherwise it is reached at \( n = 2^\left\lfloor \log_2 n_{\text{max}} \right\rfloor \) or \( n = n_{\text{max}} \).

**Proof** Please refer to “Appendix C.”

Figure 8 shows an example curve of \( \overline{L}(n) \) with three preset \( n_{\text{max}} \), which demonstrates the different aspects of Theorem 2. Here the count rate \( q = 0.05 \), \( k_{\text{opt}} \) is 6 according to Algorithm 1, and the optimal solution is reached at

\[
n = \begin{cases} 
2^{k_{\text{opt}}} = 64, & \text{when } n_{\text{max}} = n_{\text{max}0} = 80 \\
n_{\text{max}} = 30, & \text{when } n_{\text{max}} = n_{\text{max}1} = 30 \\
2^\left\lfloor \log_2 n_{\text{max}} \right\rfloor = 16, & \text{when } n_{\text{max}} = n_{\text{max}2} = 18
\end{cases}
\]

According to Theorem 2, Algorithm 2 is presented to obtain the optimal solution of (17). Its convergence can be deduced directly from the convergence of Algorithm 1.
Algorithm 2 The solution of optimization problem (17)

Require: Count rate $q$; possible maximal code alphabet size $n_{\text{max}}$.
Ensure: The optimal solution $L_{\text{opt}}$; the optimal parameter $n_{\text{opt}}$.

1: $k_{\text{max}} = \left\lfloor \log_2 n_{\text{max}} \right\rfloor$.
2: Let $q$ be the input of Algorithm 1, then $L_{\text{opt}}$ and $k_{\text{opt}}$ can be obtained.
3: if $(n_{\text{max}} \geq 2^{k_{\text{opt}}})$ then
4: $n_{\text{opt}} = 2^{k_{\text{opt}}}$.
5: else if $L(2^{k_{\text{max}}}) > L(n_{\text{max}})$ then
6: $n_{\text{opt}} = n_{\text{max}}$.
7: $L_{\text{opt}} = L(n_{\text{max}})$.
8: else
9: $n_{\text{opt}} = 2^{k_{\text{max}}}$.
10: $L_{\text{opt}} = L(2^{k_{\text{max}}})$.
11: end if

3 Experimental results and analysis

3.1 Compression efficiency

In the experiment, one hundred different count rates are selected in the range $[10^{-6}, 10^{-1}]$ on the logarithmic scale and the simulation results are obtained by processing $10^{10}$ original keys for each count rate.

Figure 9 demonstrates the compression efficiency $f$ of the proposed bit sifting scheme and the bit sifting scheme of [9], where $f = \frac{L}{H(X)}$. Since the Shannon limit of $L$ is $H(X)$, the Shannon limit of $f$ is 1. The performance of the scheme of [9] is near the Shannon limit for $q \in [10^{-4}, 10^{-1}]$, while the performance falls sharply as the count rate is outside of the range. The curves in the Fig. 9 demonstrate that our scheme is superior to the scheme of [9] in the whole range of count rate. In addition, the performance of our scheme approaches the Shannon limit uniformly, since the average,
maximal and minimal compression efficiencies within the whole range of count rate are 1.066, 1.093 and 1.045, respectively. Since the better compression performance indicates the less key consumption and higher net secure key rate, our scheme can always lead to higher net secure key rate than the scheme of [9].

Since finite original keys are used in our experiment, the statistic fluctuation is inevitable, which leads that the actual count rate may be $q'$ when we plan to simulate the case of count rate $q$. Because our scheme is designed for the count rate $q$, the straightforward application on the count rate $q'$ will influence the performance of our scheme. To quantify the influence, we compare the performance of our scheme in both cases of infinite and finite original keys, i.e., the theoretical case and the practical case. The theoretical compression efficiency is $f_{\text{theoretical}} = \bar{L}/h(q)$, while the practical compression efficiency is calculated as $f_{\text{practical}} = M/\overline{N}h(q')$, where $N$ is the length of finite original keys, $M$ is the length of their codewords, and $q'$ is the actual count rate of the $N$ original keys. The compression results are shown in Fig. 10. It can be seen that the performance of our scheme is only affected slightly by the statistic fluctuation of count rate.

### 3.2 Improvement on the net secure key rate

To the best of our knowledge, the reference [9] is one of the very few articles that discuss the net secure key rate after the withdrawal of the authentication consumption. Hence in this section, as an example, we will apply our scheme to the QKD system in [9] and demonstrate the improvement of our scheme on the net secure key rate $R'$ which is defined in (1).

In Walenta et al. [9], use a combination [30] of $\varepsilon$-almost strongly universal hash functions and a family of strongly universal hash functions named polynomial hashing...
Fig. 10 Quantitative influence of the performance of our scheme by the statistic fluctuation

[31,32] to achieve information theoretically secure authentication. The authentication algorithm produces a 127-bit authentication tag for every $2^{20}$ bits of classical communication and consumes 383 secure keys to select a hash function for every tag. According to the result of [33], the same hash function can be reused for multiple authentication rounds. If the tags attached to the messages are one-time pad encrypted, then only 127 secure keys are consumed for the classical communication of every $2^{20}$ bits and the key consumption can be reduced to one-third. Although the authentication scheme is very efficient, they show that $K_{\text{Aut}} = 0.027 \cdot R$ and $K_{\text{Aut}} = 0.05 \cdot R$, i.e., $R' = 0.973 \cdot R$ and $R' = 0.95 \cdot R$, when the fiber length is 1 and 25 km, respectively.

Let $M$ be the amount of classical communication, then the key consumption is

$$K_{\text{Aut}} = 127 \cdot \left\lceil \frac{M}{2^{20}} \right\rceil.$$  

Especially, the key consumption for the bit sifting is

$$K_{\text{Aut-bs}} = 127 \cdot \left\lceil \frac{m \cdot h(q) \cdot f}{2^{20}} \right\rceil,$$

where $m$ is the number of original keys to be processed, $q$ is the count rate, and $f$ is the compression efficiency. Since

$$\frac{m \cdot h(q)}{2^{20}} f \leq \left\lceil \frac{m \cdot h(q) \cdot f}{2^{20}} \right\rceil < \frac{m \cdot h(q)}{2^{20}} f + 1$$

and $\frac{m \cdot h(q)}{2^{20}} f$ is usually very large because $f \geq 1$ and QKD is a continuous high-speed system which leads to the large $\frac{m \cdot h(q)}{2^{20}}$, we have

$$\left\lceil \frac{m \cdot h(q) \cdot f}{2^{20}} \right\rceil \approx \frac{m \cdot h(q)}{2^{20}} f.$$

Springer
Table 4  Comparison between the performances of proposed scheme and Walenta’s scheme [9] for the QKD system in [9]

| Fiber length (km) | Count rate   | $f_w$  | $f_p$  | $f_p/f_w$ |
|-------------------|--------------|--------|--------|-----------|
| 1                 | $2.76 \times 10^{-3}$ | 1.51   | 1.07   | 0.71      |
| 12.5              | $1.18 \times 10^{-3}$ | 1.34   | 1.08   | 0.81      |
| 25                | $7.87 \times 10^{-4}$ | 1.27   | 1.06   | 0.83      |

The subscript $w$ indicates Walenta’s scheme, while the subscript $p$ indicates the proposed scheme.

So

$$\frac{K_{\text{Aut-bs-A}}}{K_{\text{Aut-bs-B}}} \approx \frac{f_A}{f_B}, \quad (18)$$

where subscript $A$ and $B$ indicate two different bit sifting schemes. The (18) demonstrates that the key consumption depends linearly on the compression efficiency.

Based on (18), more experiment results of the proposed scheme and the Walenta’s scheme are listed in Table 4, including compression efficiency and the ratio of the two compression efficiencies. It needs to be explained that the count rate is not given explicitly in [9], while it can be inferred according to the sifted key rate and the repetition frequency of the QKD system. Table 4 shows that if the QKD system in [9] employs our proposed scheme for bit sifting, then 29, 19 and 17% of the secure key consumption for bit sifting can be saved when the fiber length is 1, 12.5 and 25 km, respectively. Besides, more than 88% of the key consumption of the post-processing comes from the bit sifting step, which is evaluated according to the sifting scheme and the communication rate among the procedures of post-processing presented in [9]. So if our scheme is applied to the QKD system in [9], then $K_{\text{Aut,p}} \leq 0.0201 \cdot R$ and $K_{\text{Aut,p}} \leq 0.0425 \cdot R$, i.e., $R'_p \geq 0.9798 \cdot R$ and $R'_p \geq 0.9574 \cdot R$, which means that the net secure key rate is increased about $0.7\% \cdot R$ in both cases of 1 and 25 km. The calculation is as follows,

$$K_{\text{Aut,p}} \leq K_{\text{Aut}} \cdot 0.88 \cdot \frac{f_p}{f_w} + K_{\text{Aut}} \cdot 0.12$$

$$R'_p = R - K_{\text{Aut,p}}.$$  

In a word, our scheme can obviously improve the net secure key rate $R'$.

3.3 Some suggestions for some representative QKD systems

Many QKD systems have been developed since the first QKD system was developed in 1984. Most of them failed to take into account the authentication of the classical channel, so they did not pay much attention to the communication traffic. The parameters of four representative QKD systems are given in Table 5, which are used to compute the corresponding count rate of these systems by (10). Upon the calculated...
Table 5 Parameters of four QKD systems

| QKD system     | Dark count rate $P_d$ | Mean photon number $\mu$ | Distance $d$ (km) | Loss (dB)$^a$ | Detection efficiency $\eta_D$ (%) |
|----------------|-----------------------|--------------------------|-------------------|--------------|----------------------------------|
| Dixon etc. [34]| $1.36 \times 10^{-5}$ | 0.55                     | 20                | 8.01         | 10.00                            |
| Stucki etc. [25]| $1.60 \times 10^{-5}$ | 0.50                     | 250               | 42.60        | 2.65                             |
| Zhang etc. [29]| $1.00 \times 10^{-5}$ | 0.60                     | 20                | 7.20         | 12.00                            |
| Tanaka etc. [2] | $2.00 \times 10^{-5}$ | 0.40                     | 50                | 14.00        | 10.00                            |

$^a$ The parameter includes both the loss of transmission and the inner loss $\gamma_{Bob}$ of Bob’s optical devices

Table 6 Recommendations for four representative QKD systems

| QKD system     | Count rate $q$ | Theoretical $n$ | Theoretical $f$ | Available storage | Recommended $n$ | Actual $f$ |
|----------------|---------------|----------------|-----------------|------------------|-----------------|-------------|
| Dixon etc. [34]| $8.68 \times 10^{-3}$ | $2^8$          | 1.08           | $\geq 2$ GB$^a$ | $2^8$           | 1.08        |
| Stucki etc. [25]| $7.44 \times 10^{-7}$ | $2^{22}$       | 1.06           | 32 Kb$^b$       | $12 \times 2^{10}$ | 70.57$^c$  |
| Zhang etc. [29]| $1.36 \times 10^{-2}$ | $2^8$          | 1.08           | 32 Mb$^d$       | $2^8$           | 1.08        |
| Tanaka etc. [2] | $1.42 \times 10^{-3}$ | $2^{11}$       | 1.07           | 833 MB$^e$      | $2^{11}$        | 1.07        |

$^a$ The system is implemented in PC, and the available storage is estimated to be larger than 2 GB
$^b$ The system is implemented in Virtex II Pro FPGA and 32 Kb memory for sifting
$^c$ If the storage for sifting is extended to 8 Mb, the actual compression efficiency would be 1.06
$^d$ The system is implemented in two Cyclone III series FPGAs (EP3C120) and 32 Mb memory for sifting
$^e$ The system is implemented in several FPGAs, and average 833 Mb memory is used for each FPGA

count rates, the optimal code alphabet sizes are suggested for them in Table 6 according to their available storages. The theoretical $n$ is calculated by $n = 2^k$ without considering the constraint of storage, where $k$ is obtained by Algorithm 1, and the corresponding compression efficiency is named theoretical $f$. The recommended $n$ is calculated after taking into account the storage constraint, and the corresponding compression efficiency is named as actual $f$. The systems of [2,29,34] have sufficient storage, and all of their actual compression efficiencies are near 1. However, the system of [25] just has 32 Kb storage for sifting which is not enough for the theoretical optimal $n = 2^{22}$. The protocol adopted by [25] is coherent one way (COW)[35], which needs two bits to describe each original_key$_A$. One bit indicates whether it is a decoy or a signal, and the other determines its value when it is a signal. Hence Alice should store two bits for each original_key$_A$. If the memory is extended to 8Mb, then the compression efficiency of the system would be 1.06. Otherwise, if 8 Kb is allocated to basis sifting step, then the rest 24 Kb is for bit sifting. Therefore the possible maximum code alphabet size $n_{\text{max}}$ is set as $12 \times 2^{10}$, i.e., 12 K. According to Algorithm 2, the optimal code alphabet size $n$ and optimal solution are $12 \times 2^{10}$ and 70.57, respectively. It can be seen that the performance falls sharply due to the lack of the storage resource.
4 Conclusion

In this paper, an efficient bit sifting scheme for QKD is proposed, of which the core is a modified zero run length source coding algorithm with performance near Shannon’s limit. The existence of optimal codelength of the source coding algorithm is proved, and a fast iteration algorithm is presented to solve the optimal parameter. Both the theoretical analysis and the experimental results demonstrate that our scheme can reduce the classical communication traffic greatly and hereby evidently save the secure key consumption for authentication. As an example, our scheme is applied to the QKD system in [9], and the experimental results demonstrate that our scheme can improve the net secure key rate obviously. The impact of storage resource of a QKD system on the application of our scheme is also discussed. Some recommendations on how to apply our scheme into four representative QKD systems are given.

Acknowledgments We thank Z. Li from the Peking university for the helpful discussions of the count rate of QKD system and the COW protocol and thank Q. Zhao from the Harbin Institute of Technology for the discussions of the inference of some formulas. This work is supported by the National Natural Science Foundation of China (Grant Numbers: 61471141, 61361166006, 61301099) and the Fundamental Research Funds for the Central Universities (Grant Numbers: HIT. KISTP. 201416, HIT. KISTP. 201414).

Appendix A: Proof of Lemma 1

Proof For convenience, we denote \( p = 1 - q \), then \( p \in [0.9, 1) \) and \( q = 1 - p \). The \( \overline{L}(z) \) in (14) can be rewritten as

\[
\overline{L}(z) = \frac{(1 - p) z}{1 - p^{2z - 1}},
\]

and the partial derivative of the expected codelength \( \overline{L} \) with respect to the variable \( z \) is given by

\[
\frac{\partial \overline{L}}{\partial z} = \frac{(1 - p)p}{(p - p^{2z})^2} \left( p - p^{2z} + 2z p^{2z} \ln 2 \ln p \right). \tag{19}
\]

Due to \( p \in [0.9, 1) \),

\[
\frac{(1 - p)p}{(p - p^{2z})^2} > 0.
\]

We only need to focus on the sign of

\[
r(z) = p - p^{2z} + 2z p^{2z} \ln 2 \ln p. \tag{20}
\]

The partial derivative of \( r(z) \) with respect to the variable \( z \) can be evaluated as

\[
\frac{\partial r}{\partial z} = 2z p^{2z} \ln 2 \ln p \left( 1 + 2z \ln p \right). \tag{21}
\]
Due to $z^2 p^2 \ln^2 2 \ln p < 0$, the sign of $\frac{\partial r}{\partial z}$ is determined by the sign of the expression $1 + 2^z \ln p$. Let $1 + 2^z \ln p > 0$; then

$$z < -\log_2 (-\ln p) \equiv z_m.$$ 

So

$$\begin{align*}
\frac{\partial r}{\partial z} < 0, & \quad \text{when } 1 \leq z < z_m \\
\frac{\partial r}{\partial z} = 0, & \quad \text{when } z = z_m \\
\frac{\partial r}{\partial z} > 0, & \quad \text{when } z > z_m
\end{align*} .$$

(22)

Therefore function $r(z)$ is monotonically decreasing in the domain $z \in [1, z_m)$ and monotonically increasing in domain $(z_m, +\infty)$ and reaches the global minimum value at the point $z = z_m$, which is

$$r(z_m) = \frac{-1 + ep + \ln (-\ln p)}{e} .$$

(23)

Since

$$\frac{\partial r(z_m)}{\partial p} = 1 + \frac{1}{ep \ln p} < -2.87, \forall p \in [0.9, 1),$$

function $r(z_m)$ is monotonically decreasing with respect to the variable $p$ and comes to the maximum value at $p = 0.9$, which is about $-0.30$. So

$$r(z_m) < 0, \forall p \in [0.9, 1).$$

(24)

In addition, as $z$ approaches $+\infty$, the limit of $r(z)$ is

$$\lim_{z \to +\infty} r(z) = p, \forall p \in [0.9, 1).$$

(25)

Upon (24), (25) and the continuity of $r(z)$ in the domain $z \in [z_m, +\infty)$, it can be inferred that there exists at least one $z_0 \in (z_m, +\infty)$ satisfying that $r(z_0) = 0$ according to the intermediate value theorem [23]. Besides, since $r(z)$ is monotonically increasing function in the range $z \in [z_m, +\infty)$, the root of $r(z) = 0$ is unique, and

$$\begin{align*}
r(z) < 0, & \quad \text{when } z \in [z_m, z_0) \\
r(z) = 0, & \quad \text{when } z = z_0 \\
r(z) > 0, & \quad \text{when } z \in (z_0, +\infty)
\end{align*} .$$

(26)

Since

$$z_m \geq -\log_2 (-\ln p) \big|_{p=0.9} > 3.24,$$

we have to discuss the sign of $r(z)$ in the domain $z \in [1, z_m)$. Now our concern is the sign of

$$r(1) = p - p^2 + 2p^2 \ln 2 \ln p .$$

(27)
The partial derivative of \( r(1) \) with respect to the variable \( p \) is given by
\[
\frac{\partial r(1)}{\partial p} = 1 - 2p + 2p \ln 2 + 4p \ln 2 \ln p, \tag{28}
\]
and the second partial derivative of \( r(1) \) with respect to the variable \( p \) is given by
\[
\frac{\partial^2 r(1)}{\partial p^2} = -2 + 6 \ln 2 + 4 \ln 2 \ln p. \tag{29}
\]

Obviously in the domain \( p \in [0.9, 1) \), the function \( \frac{\partial^2 r(1)}{\partial p^2} \) is monotonically increasing with respect to the variable \( p \) and reaches the minimum value at \( p = 0.9 \), which is about 1.87. Thus the function \( \frac{\partial r(1)}{\partial p} \) with respect to the variable \( p \) is also monotonically increasing and also reaches the minimum value at \( p = 0.9 \), which is about 0.18. Therefore \( r(1) \) is a monotonically increasing function with respect to the variable \( p \) and reaches the supremum 0 at \( p = 1 \), so
\[
\quad r(1) < 0, \forall p \in [0.9, 1). \tag{30}
\]

Since \( r(z) \) is a monotonically decreasing function in the range \( z \in [1, z_m) \), we have
\[
\quad r(z) \leq r(1) < 0, \forall z \in [1, z_m). \tag{31}
\]

Combining (26) and (31), we have
\[
\left\{ \begin{array}{l}
r(z) < 0, \text{ when } z \in [1, z_0) \\
r(z) = 0, \text{ when } z = z_0 \\
r(z) > 0, \text{ when } z \in (z_0, +\infty)
\end{array} \right. \tag{32}
\]

Since the sign of \( \frac{\partial T}{\partial z} \) is the same as the sign of \( r(z) \),
\[
\left\{ \begin{array}{l}
\frac{\partial T}{\partial z} < 0, \text{ when } z \in [1, z_0) \\
\frac{\partial T}{\partial z} = 0, \text{ when } z = z_0 \\
\frac{\partial T}{\partial z} > 0, \text{ when } z \in (z_0, +\infty)
\end{array} \right.
\]
where \( z_0 \in (z_m, +\infty) \), i.e.,
\[
z_0 \in (-\log_2 (-\ln (1 - q)), +\infty).
\]

On the other hand, since \( k \in \mathbb{N}^+ \), the minimum of \( \bar{L}(k) \) is reached at \( k = \lfloor z_0 \rfloor \) or \( k = \lceil z_0 \rceil \), i.e., the optimal solution of (13) is reached at \( k = \lfloor z_0 \rfloor \) or \( k = \lceil z_0 \rceil \).
Appendix B: Proof of Theorem 1

Proof For briefness, we denote \( y = -\log_2 (-\ln (1 - q)) \). Then Lemma 1 indicates that

\[
z_0 > y, \forall q \in [10^{-15}, 0.1].
\]

At the same time, the value of the partial deviation \( \frac{\partial \overline{L}}{\partial z} \) at \( z = y+3 \) is given by

\[
\left. \frac{\partial \overline{L}}{\partial z} \right|_{z=y+3} = \frac{e^8 (1 - q)q}{(1 - e^8 (1 - q))^2} A(q),
\]

where

\[
A(q) = -1 - 24 \ln 2 + e^8 (1 - q) + 8 \ln (-\ln (1 - q)).
\]

Since

\[
\frac{e^8 (1 - q)q}{(1 - e^8 (1 - q))^2} > 0, \forall q \in [10^{-15}, 0.1]
\]

the sign of (34) is same as the sign of \( A(q) \). The partial derivative of \( A(q) \) can be evaluated as

\[
\frac{\partial A}{\partial q} = -e^8 - \frac{8}{(1 - q) \ln (1 - q)},
\]

which is a monotonically decreasing function with respect to the variable \( q \) and

\[
\begin{cases}
\frac{\partial A}{\partial q} > 7.20 \times 10^{15}, & \text{when } q = 10^{-15} \\
\frac{\partial A}{\partial q} < -2.80 \times 10^3, & \text{when } q = 0.1
\end{cases}
\]

So the function \( A(q) \) is firstly monotonically increasing and then monotonically decreasing in the range \( q \in [10^{-15}, 0.1] \). The minimum must occur at the point \( q = 10^{-15} \) or \( q = 0.1 \), where \( A(10^{-15}) \approx 2687.85 \) and \( A(0.1) \approx 2647.22 \), so

\[
A(q) > 0, \forall q \in [10^{-15}, 0.1].
\]

Hence, combining (34), (35) and (36),

\[
\left. \frac{\partial \overline{L}}{\partial z} \right|_{z=y+3} > 0, \forall q \in [10^{-15}, 0.1].
\]
Efficient bit sifting scheme of post-processing in quantum… 3809

Making use of (37) and the Lemma 1, it can be concluded that
\[ z_0 < y + 3, \forall q \in \left[10^{-15}, 0.1\right]. \] (38)

Combining (33) and (38), we have
\[ y < z_0 < y + 3, \forall q \in \left[10^{-15}, 0.1\right]. \]

On the other hand, Lemma 1 shows that the optimal solution of (13) is reached at \( k = \lfloor z_0 \rfloor \) or \( k = \lceil z_0 \rceil \), so the optimal parameter \( k_{\text{opt}} \) of (13) is one of the following five values, i.e., \( \lfloor y \rfloor, \lfloor y \rfloor + 1, \lfloor y \rfloor + 2, \lfloor y \rfloor + 3, \lceil y \rceil + 3. \) \( \square \)

Appendix C: Proof of Theorem 2

Proof For brevity, let \( \{a \ldots b\} \overset{\Delta}{=} [a, b] \cap \mathbb{N}. \)

(a) When \( n_{\text{max}} \geq 2^{k_{\text{opt}}} \), the optimal code alphabet size
\[ n_{\text{opt}} = 2^{k_{\text{opt}}} \leq n_{\text{max}}, \]
which is reachable in the domain \( \{2 \ldots n_{\text{max}}\} \). So the optimal solution of (17) is reached at \( n = 2^{k_{\text{opt}}} \) in this case.

(b) When \( n_{\text{max}} < 2^{k_{\text{opt}}} \), the optimal code alphabet size
\[ n_{\text{opt}} = 2^{k_{\text{opt}}} > n_{\text{max}}, \]
which cannot be reachable in the domain \( \{2 \ldots n_{\text{max}}\} \). In the case, the domain \( \{2 \ldots n_{\text{max}}\} \) is divided into \( \{2 \ldots 2^{k_{\text{max}}}\} \) and \( \{2^{k_{\text{max}}} + 1 \ldots n_{\text{max}}\} \), where \( k_{\text{max}} = \lceil \log_2 n_{\text{max}} \rceil \).

(b.1) Since
\[ \{2 \ldots 2^{k_{\text{max}}}\} = \bigcup_{k=1}^{k_{\text{max}}} \left\{2^{k-1} + 1 \ldots 2^k\right\} \]
and the minimum of \( \overline{L} (n) \) in the range \( \{2^{k-1} + 1 \ldots 2^k\} \) occurs at the point \( 2^k \) according to (12), \( 1 \leq k \leq k_{\text{max}} \), we just need to consider the minimum of \( \overline{L} (k) \) in the range \( \{1 \ldots k_{\text{max}}\} \). According to Lemma 1, it can be inferred that \( k_{\text{opt}} \leq \lfloor z_0 \rfloor \), then
\[ k_{\text{max}} = \lceil \log_2 n_{\text{max}} \rceil \leq k_{\text{opt}} - 1 \leq \lfloor z_0 \rfloor - 1 < z_0. \]

So \( \overline{L} (k) \) is monotonically decreasing in the range \( \{1 \ldots k_{\text{max}}\} \) and reaches the minimum at the point \( k = k_{\text{max}} \). Therefore, the minimum of \( \overline{L} (n) \) in the range \( \{2 \ldots 2^{k_{\text{max}}}\} \) occurs at the point \( n = 2^{k_{\text{max}}} \), i.e., \( n = 2^\lfloor \log_2 n_{\text{max}} \rfloor \).
Since the function $L(n)$ is monotonically decreasing in the range $\{2^{k_{\text{max}}} + 1 \ldots n_{\text{max}}\}$, the minimum of $L(n)$ in the range occurs at the point $n = n_{\text{max}}$.

Combining (b.1) and (b.2), it can be seen that the optimal solution of (17) is reached at $n = 2^{\left\lfloor \log_2 n_{\text{max}} \right\rfloor}$ or $n = n_{\text{max}}$ when $n_{\text{max}} < 2^{k_{\text{opt}}}$.

Combining the case (a) and the case (b), the theorem is proved. $\square$

References

1. Shannon, C.E.: A mathematical theory of communication. Bell Tech. J. 27(4), 379 (1948)
2. Tanaka, A., Fujiwara, M., Yoshino, K., Takahashi, S., Nambu, Y., Tomita, A., Miki, S., Yamashita, T., Wang, Z., Sasaki, M., et al.: High-speed quantum key distribution system for 1-Mbps real-time key generation. Quantum Electron IEEE J 48(4), 542 (2012)
3. Dixon, A., Yuan, Z., Dynes, J., Sharpe, A., Shields, A.: Continuous operation of high bit rate quantum key distribution. Appl. Phys. Lett. 96(16), 161102 (2010)
4. Lütkenhaus, N.: Security against individual attacks for realistic quantum key distribution. Phys. Rev. A 61(5), 052304 (2000)
5. Bennett, C.H., Brassard, G.: In: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, vol. 175, New York (1984)
6. Mayers, D.: Advances in CryptologyCRYPTO96, pp. 343–357. Springer, New York (1996)
7. Mayers, D.: Unconditional security in quantum cryptography. J. ACM 48(3), 351 (2001)
8. Ben-Or, M., Horodecki, M., Leung, D.W., Mayers, D., Oppenheim, J.: Theory of Cryptography, pp. 386–405. Springer, New York (2005)
9. Walenta, N., Burg, A., Caselunghe, D., Constantin, J., Gisin, N., Guinnard, O., Houlmann, R., Junod, P., Kozh, B., Kulesza, N., et al.: A fast and versatile quantum key distribution system with hardware key distillation and wavelength multiplexing. N. J. Phys. 16(1), 013047 (2014)
10. Brassard, G., Salvail, L.: Advances in CryptologyEUROCRYPT93, pp. 410–423. Springer, New York (1994)
11. Buttler, W., Lamoreaux, S., Torgerson, J., Nickel, G., Donahue, C., Peterson, C.G.: Fast, efficient error reconciliation for quantum cryptography. Phys. Rev. A 67(5), 052303 (2003)
12. Traisililanun, W., Sririmanawat, K., Sangaroon, O.: Secret key reconciliation using BCH code in quantum key distribution. A A 3(2), 1 (2007)
13. Martínez-Mateo, J., Elkouss, D., Martín, V.: Blind reconciliation. Quantum Inf. Comput. 12(9&10), 0791 (2012)
14. Jouguet, P., Kunz-Jacques, S.: High performance error correction for quantum key distribution using polar codes. Quantum Inf. Comput. 14(3–4), 329 (2014)
15. den Boer, B.: A simple and key-economical unconditional authentication scheme. J. Comput. Secur. 2(1), 65 (1993)
16. Bierbrauer, J., Johansson, T., Kabatianskii, G., Smeets, B.: Advances in CryptologyCrypto93, pp. 331–342. Springer, New York (1994)
17. Krawczyk, H.: Advances in CryptologyCRYPTO94, pp. 129–139. Springer, New York (1994)
18. Abidin, A., Larsson, J.A.: Research in Cryptology, pp. 99–108. Springer, New York (2012)
19. Kollmitzer, C., Pivi, M.: Applied Quantum Cryptography, vol. 797. Springer, New York (2010)
20. Li, Q., Le, D., Rao, M.: In: Instrumentation, measurement, computer, communication and control (IMCCC), 2012 Second International Conference on, IEEE, pp. 272–275. (2012)
21. Hwang, W.Y.: Quantum key distribution with high loss: toward global secure communication. Phys. Rev. Lett. 91(5), 057901 (2003)
22. Mink, A.: In: Optics East 2007, International Society for Optics and Photonics, pp. 678,014–678,014 (2007)
23. Johnsonbaugh, R., Pfaffenberger, W.E.: Foundations of Mathematical Analysis. Dover Publications, New York (2012)
24. Xu, B.: The practical security of quantum key distribution system. Ph.D. thesis, Peking University (2012)
25. Stucki, D., Walenta, N., Vannel, F., Thew, R.T., Gisin, N., Zbinden, H., Gray, S., Towery, C., Ten, S.: High rate, long-distance quantum key distribution over 250 km of ultra low loss fibres. N. J. Phys. 11(7), 075003 (2009)
26. Wang, S., Chen, W., Guo, J.F., Yin, Z.Q., Li, H.W., Zhou, Z., Guo, G.C., Han, Z.F.: 2 GHz clock quantum key distribution over 260 km of standard telecom fiber. Opt. Lett. 37(6), 1008 (2012)
27. Sasaki, M., Fujiwara, M., Ishizuka, H., Klaus, W., Wakui, K., Takeoka, M., Miki, S., Yamashita, T., Wang, Z., Tanaka, A., et al.: Field test of quantum key distribution in the Tokyo QKD Network. Opt. Express 19(11), 10387 (2011)
28. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Wiley, New Jersey (2006)
29. Zhang, H.F., Wang, J., Cui, K., Luo, C.L., Lin, S.Z., Zhou, L., Liang, H., Chen, T.Y., Chen, K., Pan, J.W.: A real-time QKD system based on FPGA. J. Lightw. Technol. 30(20), 3226 (2012)
30. Stinson, D.R.: Universal hashing and authentication codes. Des. Codes Cryptogr. 4(3), 369 (1994)
31. Wegman, M.N., Carter, J.L.: Universal classes of hash functions. J. Comput. Syst. Sci. 18(2), 143 (1979)
32. Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. 22(3), 265 (1981)
33. Portmann, C.: arXiv preprint arXiv:1202.1229 (2012)
34. Dixon, A., Yuan, Z., Dynes, J., Sharpe, A., Shields, A.: Gigahertz decoy quantum key distribution with 1 Mbit/s secure key rate. Opt. Express 16(23), 18790 (2008)
35. Stucki, D., Brunner, N., Gisin, N., Scarani, V., Zbinden, H.: Fast and simple one-way quantum key distribution. Appl. Phys. Lett. 87(19), 194108 (2005)