Research Article

Optimization and Implementation of Efficient and Universal Quantum Key Distribution

Da-jun Huang,¹ Wen-zhe Zhong,² Jin Zhong ^{1,3} Dong Jiang ⁴ and Hao Wu¹

¹School of Computer Science and Technology, Hefei Normal University, Hefei 230601, China

²College of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China

³College of Computer Science and Technology, China University of Science and Technology, Hefei 230026, China

⁴State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China

Correspondence should be addressed to Dong Jiang; jiangd@nju.edu.cn

Received 14 January 2020; Accepted 28 April 2020; Published 25 May 2020

Academic Editor: Jit S. Mandeep

Copyright © 2020 Da-jun Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since quantum key distribution (QKD) can provide proven unconditional security guaranteed by the fundamental laws of quantum mechanics, it has attracted increasing attention over the past three decades. Its low bit rate, however, cannot meet the requirements of modern applications. To solve this problem, recently, an efficient and universal QKD protocol based on chaotic cryptography and middleware technology was proposed, which efficiently increases the bit rate of the underlying QKD system. Nevertheless, we find that this protocol does not take the bit errors into account, and one error bit may lead to the failure of the protocol. In this paper, we give an optimized protocol and deploy it on a BB84 QKD platform. The experimental results show that the optimized version provides resistance to bit errors compared with the original version. And the statistical properties of the generated bits are fully assessed using different methods. The evaluation results prove that the proposed protocol can generate bits with outstanding properties.

1. Introduction

Quantum key distribution (QKD), invented by Bennett and Brassard in 1984 [1], is a technology for sharing keys between two legitimate users, Alice and Bob. Since the quantum mechanics laws guarantee that the keys are perfectly secret from the eavesdropper, Eve [2], it has attracted widespread attention, and many advanced works have been proposed over recent years [3, 4]. Especially, the successful launch of the Micius quantum satellite [5] proves the arrival of the global quantum communication era. Despite these developments, the bit rates of existing QKD systems cannot meet the requirements of high-speed applications [6]. Many research communities are dedicated to solve this problem, and some advanced protocols have been proposed in recent years [7, 8]. However, these protocols are neither efficient enough to fit the modern applications nor compatible with different QKD protocols or optical platforms.

To solve these problems, recently, Jiang et al. proposed an efficient and universal quantum key distribution (EUQKD) protocol [9] based on chaotic cryptography and middleware techniques. It uses a chaotic pseudorandom bit generator (PRBG) as a middleware, directly deploying on a QKD platform. The middleware initially takes the bits generated by the platform as input, periodically updates the chaotic systems, and efficiently generates the bit strings. By employing PRBG, EUQKD improves the bit rate of the underlying QKD system, resulting in higher efficiency. And by using the middleware technologies, it divides the cryptosystem to different layers and isolates the details of QKD platform from the users, leading to better compatibility. And EUQKD can be easily realized by software or hardware.

However, we find that EUQKD does not consider the bit errors. Due to the impact of channel noise or Eve's attacks, errors are common in the generated bits. If Alice and Bob use a set of bits to update their input-sensitive middleware, even though there is only one error bit in these bits, the

execution results will be completely different, resulting in the failure of the communication. To solve this problem, in this paper, we optimize EUQKD protocol and provide it with the ability to resist bit errors by checking part of the execution results. Then, we deploy a polarized state-based QKD platform and realize the optimized protocol. The experimental results present that compared with the original version, the proposed scheme achieves better performance in terms of error resistance. We also use different approaches to assess the statistical properties of the generated bits. The evaluation results show that the optimized protocol can efficiently output bits with outstanding properties.

This paper is organized as follows: we first give a brief review of EUQKD protocol, followed by analyzing the problem of the protocol, introducing the optimized scheme, and realizing it on a BB84 QKD platform. Then, the statistical properties of the generated bits are evaluated by different methods. Finally, a brief conclusion is given.

2. Review of EUQKD

As shown in Figure 1(a), EUQKD can be divided into three layers. The bottom layer is a QKD platform, the middle layer is a chaos-based PRBG, and the top layer is a software application which implements encrypting, decrypting, and communicating components to transmit messages between two legitimate users.

The core of EUQKD protocol is the middle layer; it contains two piecewise linear chaotic maps (PLCMs) defined as follows:

$$F(x, p) = \begin{cases} \frac{x}{p}, & x \in [0, p), \\ \frac{x-p}{(1/2)-p}, & x \in [p, \frac{1}{2}], \\ F(1-x, p), & x \in (\frac{1}{2}, 1], \end{cases} \quad (1)$$

where $x \in [0, 1]$ is referred to as the initial condition and $p \in (0, 0.5)$ is referred to as the control parameter. The bit strings can be generated by iterating the PLCM according to $x_n = F(x_{n-1}, p)$. As plotted in Figure 1(b), the middle layer runs as follows:

- (1) Alice randomly selects initial conditions x_0^1, x_0^2 and control parameters p_1, p_2 , encrypts them with QKD bits, and sends the ciphertext to Bob. Then, Alice and Bob initialize their PLCMs with the selected parameters.
- (2) Alice and Bob iterate their PLCMs to generate two sets of iteration results $R_1\{r_{36}^1, \dots, r_m^1\}$, $R_2\{r_{36}^2, \dots, r_m^2\}$ ($\{r_1, \dots, r_{35}\}$ are discarded for security), where m is the upper bound of iterations. Then, they construct two integers z, z' with QKD bits, act XOR on mantissa parts of r_z^1 and $r_{z'}^2$ to obtain a set of bits (Cbits). Alice and Bob repeat this step n times, and then they jump to the next step.

- (3) Alice and Bob use QKD bits to construct four integers $z_1, z_1', z_2, z_2' \in \{36, \dots, m\}$ and select $r_{z_1}^1, r_{z_1'}^1$ as x_0^1 and p_1 (if $p_1 > 0.5, p_1 = 1 - p_1$) and select $r_{z_2}^2, r_{z_2'}^2$ as x_0^2 and p_2 (if $p_2 > 0.5, p_2 = 1 - p_2$) to update their PLCMs.
- (4) Alice and Bob repeat steps 2 and 3 until they generate enough Cbits for encrypting and decrypting the secret message.

The middle layer isolates the bottom layer to achieve compatibility; it can be easily realized by both software and hardware and can be rapidly deployed on any QKD platform. It also takes QKD bits as input and efficiently outputs Cbits, resulting in higher efficiency.

3. Optimization and Realization of EUQKD

3.1. Optimization of EUQKD. EUQKD significantly improves the efficiency of the physical layer; however, it does not consider bit errors. Generally, a QKD protocol consists of quantum and classical parts [10]. In the former part, Alice and Bob generate a set of bits (raw key) by preparing and measuring photon states. Then, they broadcast their basis choices. For each bit of the generated raw key, if Alice's basis is the same as Bob's, it is stored as the sifted key; otherwise, it is discarded. Due to the channel noise and Eve's attacks, there may exist some errors in the sifted key, and part of the information of the sifted key may be leaked. Thus, Alice and Bob need to execute the classical part, i.e., postprocessing [11], to correct the errors and cover the information leakage. They first run the first step known as the parameter estimation [12] to estimate the error rate of the sifted key. The second step, i.e., the key reconciliation [13], uses the estimated error rate as input to correct the errors. Finally, they run the privacy amplification [14] step to cover the leaked information.

Existing key reconciliation algorithms, e.g., belief propagation [15, 16], shuffled belief propagation [17], layer belief propagation [18, 19], etc., use a piecewise function to extract useful information from Alice's syndrome to correct Bob's QKD bits. To avoid information leakage, such information is not accurate and reliable. Thus, there may be some bit errors after key reconciliation. According to EUQKD protocol, if an error bit affects the second step, it will cause 52-bit errors in the Cbits (IEEE 754 standard represents a double-precision float number in the form $(-1)^1 \times \text{mantissa}(52 \text{ bits}) \times 2^{11}$). We know that chaotic system is sensitive to initial condition and control parameter. As shown in Figure 2, the iteration results of PLCM (1) with very close initial conditions and control parameters are totally different. Thus, if the error bit affects the third step, it will lead to the failure of the protocol. To solve this problem, we optimize EUQKD protocol as follows:

- (1) Alice randomly selects initial conditions x_0^1, x_0^2 and control parameters p_1, p_2 , encrypts them with QKD bits, and sends the ciphertext to Bob. Then, Alice and Bob initialize their PLCMs with the selected parameters.

FIGURE 1: Review of EUQKD. (a) Hierarchical structure of EUQKD. (b) Workflow diagram of the middle layer.

FIGURE 2: Iteration results of PLCM (1) with very close initial conditions and control parameters.

- (2) Alice and Bob iterate their PLCMs to generate two sets of iteration results $R_1\{r_{36}^1, \dots, r_m^1\}$, $R_2\{r_{36}^2, \dots, r_m^2\}$. Then, they broadcast part of the results over the classical channel and calculate the error rate. If the error rate exceeds the threshold, they jump to step 1.
- (3) They construct two integers z, z' with QKD bits. If the selected iteration result r_z^1 or $r_{z'}^2$ has been broadcasted, they select the next iteration result until they find an iteration result that has not been broadcasted. Then, they act XOR on mantissa parts of the two iteration results to obtain a set of Cbits. Alice and Bob repeat this step n times, and then they jump to the next step.
- (4) Alice and Bob use QKD bits to construct four integers $z_1, z_1', z_2, z_2' \in \{36, \dots, m\}$ and select $r_{z_1}^1, r_{z_1'}^1$ as x_0^1 and p_1 (if $p_1 > 0.5$, $p_1 = 1 - p_1$) and select $r_{z_2}^2, r_{z_2'}^2$ as x_0^2 and p_2 (if $p_2 > 0.5$, $p_2 = 1 - p_2$) to update their PLCMs.
- (5) Alice and Bob repeat steps 2, 3, and 4 until they generate enough Cbits for encrypting and decrypting the secret message.

Clearly, in each round of the optimized protocol, Alice and Bob check part of the iteration results. Since PLCM (1) is

sensitive to the initial condition and the control parameter, they can easily detect the failure. And if the protocol fails, they will randomly select a set of parameters to synchronize their PLCMs. This can avoid the failure of the protocol and guarantee that the users can generate enough bits for encryption and decryption, even if there exist error bits in QKD bits. Compared with the original protocol, the optimized scheme, therefore, achieves higher robustness. Although the legitimate users broadcast part of their iteration results to resist bit errors, these iteration results are not used to generate the secret key. And many papers have proved that the eavesdropper cannot break digital chaotic PRBG, even if he obtains part of the iteration results [20–22]. Therefore, the eavesdropper cannot obtain useful information during the communication.

3.2. Realization of EUQKD. In this paper, a BB84 QKD platform based on polarization states is deployed as the physical layer. The optical setup of the platform is shown in Figure 3. Alice can control four lasers to send horizontal, vertical, $+45^\circ$, and -45° polarization pulses, which are denoted as $H, V, +$, and $-$, respectively. Because the intensity of these pulses may be different, we first attenuate them to the same level. Then, the pulses are combined by two

BS: beam splitter

PBS: polarizing beam splitter

SPD: single photon detector

FIGURE 3: The optical setup of the physical layer.

polarizing beam splitters (PBSs) to guarantee the orthogonality between H and V pulses and the orthogonality between $+$ and $-$ pulses. The manual polarization controllers adjust the angle between H and $+$ pulses and the angle between V and $-$ pulses to 45° . Finally, all pulses are combined by a beam splitter (BS), attenuated to single photon level, and sent to Bob over the quantum channel. In Bob's optical platform, the state first passes a 50/50 BS to randomly select a measurement basis. Since the optical fiber may change the state of the photon, two manual polarization controllers are used to adjust the state. In our platform, only one single photon detector (SPD) is deployed. Since the four states $\{|H\rangle, |V\rangle, |+\rangle, |-\rangle\}$ are distinguished by delay lines of different lengths, SPD can determine the state according to the time tag of the measurement.

We run BB84 protocol on our optical platform and generate 30 sets of sifted keys. Then, we use the post-processing algorithm proposed in Ref. [23] to process the generated sifted keys. The error rates of the generated sifted keys and the results of parameter estimation are plotted in Figure 4(a). The error rates of the generated QKD bits are shown in Figure 4(b).

Because the bit error rates of the generated QKD bits are very low, we slightly change Bob's QKD bits and use the QKD bits to update Alice's and Bob's PLCMs. We simulate this process for 10,000 times, and all errors are detected in the second step of the optimized protocol. Our protocol, thus, provides the ability to resist bit errors.

4. Statistical Evaluation

In this section, we evaluate the statistical properties of the generated Cbits from aspects of uniformity and correlation. Cbits should show perfect uniformity; otherwise, Eve may extract some information from the ciphertext. To evaluate the uniformity, we set the rounds to 100, set the upper bound of iterations to 1000, and run the proposed protocol 100 times. The distribution of iteration results, control

parameters, and initial conditions is plotted in Figures 5(a)–5(c), respectively. Obviously, the results show perfect uniformity.

To evaluate the correlation of the generated bits, first, the Hamming distance and Pearson's correlation coefficient are calculated [24]. For two binary sequences $S_b^1 = [b_1^1, b_2^1, \dots, b_n^1]$ and $S_b^2 = [b_1^2, b_2^2, \dots, b_n^2]$, Hamming distance $H(S_b^1, S_b^2)$ can be calculated as follows:

$$H(S_b^1, S_b^2) = \sum_{i=1}^n (b_i^1 \oplus b_i^2). \quad (2)$$

If S_1 and S_2 are two truly random sequences, $H(S_b^1, S_b^2)$ should approach 0.5. For two integer sequences $S_I^1 = [I_1^1, I_2^1, \dots, I_n^1]$ and $S_I^2 = [I_1^2, I_2^2, \dots, I_n^2]$, Pearson's correlation coefficient can be calculated according to the following equation:

$$P(S_I^1, S_I^2) = \frac{\sum_{i=0}^n (I_i^1 - \bar{I}^1)(I_i^2 - \bar{I}^2)}{\sqrt{\sum_{i=0}^n (I_i^1 - \bar{I}^1)^2} \sqrt{\sum_{i=0}^n (I_i^2 - \bar{I}^2)^2}}, \quad (3)$$

where $\bar{I}^1$ and $\bar{I}^2$ are the average values of S_I^1 and S_I^2 , respectively. If S_I^1 and S_I^2 are uncorrelated, $P(S_I^1, S_I^2)$ should approach 0. We generate 20 pairs of Cbits. The Hamming distances of each pair of Cbits are shown in Figure 6(a). Then, we generate 20 pairs of Cbits and construct 20 pairs of integer sequences. Pearson's correlation coefficient of each pair of integer sequences is plotted in Figure 6(b).

Second, we calculate the autocorrelation of generated Cbits, and the cross correlation of two pair of Cbits generated with very close initial conditions. The results are drawn in Figures 6(c) and 6(d), respectively. Clearly, the results prove that the generated Cbits have outstanding correlation properties.

5. Image Encryption and Decryption

In this section, we use image encryption to further evaluate the statistical properties of Cbits. First, the histogram

FIGURE 4: Experimental results of quantum key distribution.

FIGURE 5: Uniformity evaluation. (a) Uniformity of iteration results. (b) Uniformity of control parameters. (c) Uniformity of initial conditions.

FIGURE 6: Continued.

FIGURE 6: Correlation evaluation. (a) Results of Hamming distance. (b) Results of Pearson's correlation coefficient. (c) Results of autocorrelation. (d) Results of cross correlation.

FIGURE 7: Histogram analysis. (a) The plain image. (b) The histogram of the plain image. (c) The cipher image. (d) The histogram of the cipher image.

analysis method is employed to evaluate the uniformity of the generated bits. As shown in Figure 7(a), we choose a 256×256 gray scale image as the plain image. The histogram of the plain image is plotted in Figure 7(b). Then, we

generate a set of Cbits and encrypt the plain image. The cipher image and its histogram are drawn in Figures 7(c) and 7(d), respectively. Clearly, the generated bits uniformly conceal the information of the plain image. The variances of

FIGURE 8: The distribution of two adjacent pixels. w and h denote the abscissa and the ordinate of the pixels. (a–c) The distribution of two horizontally, vertically, and diagonally adjacent pixels selected in the plain image. (d–f) The distribution of two horizontally, vertically, and diagonally adjacent pixels selected in the cipher image.

TABLE 1: Pearson's correlation coefficient of adjacent pixels.

Direction of the adjacent pixels	Plain image	Cipher image
Horizontal	0.964428	−0.060663
Vertical	0.954564	0.008813
Diagonal	0.920223	0.037429

the histograms can be calculated according to the following equation:

$$V(G) = \frac{1}{(w \times h)^2} \sum_{i=1}^{w \times h} \sum_{j=1}^{w \times h} \frac{1}{2} (g_i - g_j)^2, \quad (4)$$

where w and h are the width and the height of the image, g_i is the number of pixels with a gray value equal to i , and $G = \{g_1, g_2, \dots, g_{256}\}$ is the gray value vector of the image. The higher value of variance indicates the lower uniformity of the image. The variances of the plain and the cipher images are 66200 and 272.53, respectively. The generated Cbits, thus, show perfect uniformity properties.

Generally, two adjacent pixels in an image show strong correlation. Therefore, we analyze the correlation between two adjacent pixels in the plain and cipher images to evaluate the correlation properties of the generated Cbits. We randomly choose 1000 pairs of two horizontally, vertically, and diagonally adjacent pixels. Figures 8(a)–8(c) plot the distribution of two horizontally, vertically, and diagonally adjacent pixels selected in the plain image. And Figures 8(d)–8(f) present the distribution of two horizontally, vertically, and diagonally adjacent pixels selected in the cipher image. Obviously, the plain image is decorrelated by the generated bits.

Pearson's correlation coefficient of adjacent pixels is shown in Table 1. Obviously, the adjacent pixels selected from the plain image are highly correlated. And there is negligible correlation between the adjacent pixels selected from the cipher image. The generated bits, therefore, show outstanding correlation properties.

For a well-designed cryptosystem, high key sensitivity is also required, i.e., Eve cannot obtain any information of the plain image, even if she uses a key which is very close to the correct key to decrypt the plain image. To evaluate the key sensitivity, we select a set of initial conditions and control parameters as the key to synchronize Alice's and Bob's PLCMs, use a set of QKD bits to generate Cbits to encrypt the plain image, and plot the cipher image in Figure 9(a). Then, we use the correct key and QKD bits to generate Cbits, decrypt the cipher image, and show the decrypted image in Figure 9(b). Finally, we slightly change the key, use the changed key and QKD bits to generate Cbits, decrypt the cipher image, and draw the decrypted image in Figure 9(c).

Pearson's correlation coefficient between encrypted and decrypted images is shown in Table 2. Obviously, Eve cannot obtain any information even if she obtains a key which is very close to the correct key, thus proving the high key sensitivity of our protocol.

FIGURE 9: Key sensitivity evaluation. (a) The cipher image. (b) The image decrypted with the correct keys. (c) The image decrypted with a slightly different key.

TABLE 2: Pearson's correlation coefficient between two images.

Image 1	Image 2	Pearson's correlation coefficient
Encrypted image	Decrypted image 1	-0.026842
Encrypted image	Decrypted image 2	-0.038520
Decrypted image 1	Decrypted image 2	0.009568

6. Conclusion

EUQKD efficiently improves the efficiency of the underlying QKD platform; however, it does not consider the bit errors, which may lead to the failure of the protocol. In this paper, we optimize EUQKD against bit errors and realize the protocol with a BB84 QKD platform. Experimental results show that the optimized protocol significantly reduces the impact of error bits and efficiently generates Cbits with outstanding statistical properties.

Data Availability

All data generated or analyzed during this study are included within this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was financially supported by the Natural Science Foundation of Jiangsu Province (no. BK20190297) and the Natural Science Foundation of Anhui Province (no. 1708085qf157).

References

- [1] H. Bennett Ch and G. Brassard, "Quantum cryptography: public key distribution and coin tossing," in *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, p. 175, Bangalore, India, December 1984.
- [2] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of Modern Physics*, vol. 81, no. 3, pp. 1301–1350, 2009.
- [3] H.-K. Lo, X. Ma, and K. Chen, "Decoy state quantum key distribution," *Physical Review Letters*, vol. 94, no. 23, Article ID 230504, 2005.
- [4] H.-K. Lo, M. Curty, and B. Qi, "Measurement-device-independent quantum key distribution," *Physical Review Letters*, vol. 108, no. 13, Article ID 130503, 2012.
- [5] S.-K. Liao, W.-Q. Cai, W.-Y. Liu et al., "Satellite-to-ground quantum key distribution," *Nature*, vol. 549, no. 7670, pp. 43–47, 2017.
- [6] H.-K. Lo, H. F. Chau, and M. Ardehali, "Efficient quantum key distribution scheme and a proof of its unconditional security," *Journal of Cryptology*, vol. 18, no. 2, pp. 133–165, 2005.
- [7] P. Xue, C.-F. Li, and G.-C. Guo, "Efficient quantum-key-distribution scheme with nonmaximally entangled states," *Physical Review A*, vol. 64, no. 3, Article ID 032305, 2001.
- [8] X.-H. Li, F.-G. Deng, and H.-Y. Zhou, "Efficient quantum key distribution over a collective noise channel," *Physical Review A*, vol. 78, no. 2, Article ID 022321, 2008.
- [9] D. Jiang, Y. Chen, X. Gu, L. Xie, and L. Chen, "Efficient and universal quantum key distribution based on chaos and middleware," *International Journal of Modern Physics B*, vol. 31, no. 2, Article ID 1650264, 2017.
- [10] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145–195, 2002.
- [11] M. Tomamichel, C. C. W. Lim, N. Gisin, and R. Renner, "Tight finite-key analysis for quantum cryptography," *Nature Communications*, vol. 3, p. 634, 2012.
- [12] X. Ma, C.-H. F. Fung, and M. Razavi, "Statistical fluctuation analysis for measurement-device-independent quantum key distribution," *Physical Review A*, vol. 86, no. 5, Article ID 052305, 2012.
- [13] J. Martinez-Mateo, D. Elkouss, and V. Martin, "Key reconciliation for high performance quantum key distribution," *Scientific Reports*, vol. 3, p. 1576, 2013.
- [14] C. H. Bennett, G. Brassard, C. Crepeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.
- [15] S.-Y. Sae-Young Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, 2001.

- [16] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.
- [17] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proceedings of the Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 1, IEEE, Pacific Grove, CA, USA, pp. 8–15, 2002.
- [18] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proceedings of the IEEE Workshop on Signal Processing Systems, SIPS 2004*, IEEE, New York, NY, USA, pp. 107–112, 2004.
- [19] E. Sharon, S. Litsyn, and J. Goldberger, "An efficient message-passing schedule for LDPC decoding," in *Proceedings of the 2004 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, IEEE, Tel-Aviv, Israel, pp. 223–226, September 2004.
- [20] S. Li, X. Mou, and Y. Cai, "Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography," in *International Conference on Cryptology in India*, pp. 316–329, Springer, Berlin, Germany, 2001.
- [21] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," *International Journal of Bifurcation and Chaos*, vol. 15, no. 10, pp. 3119–3151, 2005.
- [22] J. Zhong, D. Jiang, Q.-G. Huang, and Y. Cao, "A self-updating digital chaotic stream cipher," *International Journal of Modern Physics C*, vol. 29, no. 8, Article ID 1850074, 2018.
- [23] C. Gao, D. Jiang, Y. Guo, and L. Chen, "Multi-matrix error estimation and reconciliation for quantum key distribution," *Optics Express*, vol. 27, no. 10, pp. 14545–14566, 2019.
- [24] J. Lee Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.