

# Q-CSKDF: A Continuous and Security Key Derivation Function for Quantum Key Distribution

Lutong Chen, *Graduate Student Member, IEEE*, Kaiping Xue, *Senior Member, IEEE*, Jian Li, *Senior Member, IEEE*, Zhonghui Li, *Member, IEEE*, Nenghai Yu

**Abstract**—Quantum Key Distribution (QKD) offers a novel approach to address the challenges posed by quantum computations in classical ciphers. Due to hardware limitations and the stochastic nature of QKD protocols, the secure key generation rate is currently constrained and subject to fluctuations. However, users expect a consistent supply of keys at a stable rate while maintaining a secure threshold. To address this issue, we present a Quantum Continuous and Secure Key Derivation Function named Q-CSKDF that utilizes QKD keys to produce stable, high-rate, but secure derived keys. Our approach satisfies both security and rate requirements in a full-period and continuous perspective, with a dynamic derivation expansion ratio determined by users' requirement of security level and key quantity and the amount of generated QKD keys. Extensive semi-physical experiments demonstrate Q-CSKDF's capability to consistently generate high-rate derived keys while ensuring the desired level of security.

**Index Terms**—Quantum networks, quantum key distribution, key derivation function.

## I. INTRODUCTION

Quantum Key Distribution (QKD) stands at the frontier of modern cryptography, offering unprecedented security in the post-quantum age [1]. It helps two nodes in a quantum network to produce secure keys. These secure keys can subsequently serve as the foundation for establishing secure communication tunnels, ensuring the confidentiality of sensitive data over the network. Thus, in recent years, significant progress has been made in this research hotspot, along with functional QKD networks constructed in major countries and areas. In 2022, the EU provided funding for the OpenQKD project [2] to establish multiple QKD testbeds across several countries. There are also plans to construct a large-scale QKD network with over 100 nodes in Japan by 2024 [3]. Furthermore, China's quantum network [4] has been extended over a distance of 4,600 km.

However, the current QKD network can still not adequately meet the growing demand for secure encryption, mainly due to two aspects. The first issue is the Secure Key Rate (SKR) of the QKD networks cannot satisfy the encryption needs. Recent study [5] achieves a rate of 110 Mbps in a laboratory environment. However, classic backbone networks typically operate at bandwidths exceeding 10 Gbps. This disparity between QKD's key provisioning and the demands

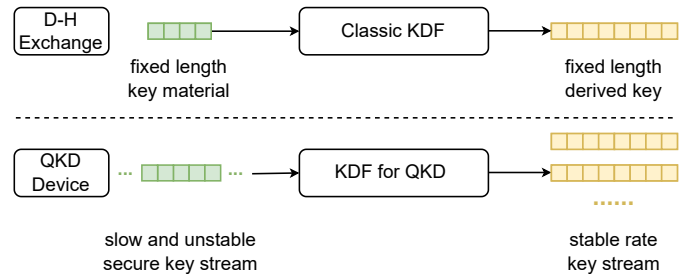


Fig. 1. Structure difference between classic KDF and that in QKD networks

is significant. The second issue is about the stable delivery of secure keys. The networks must continuously provide secret keys at a relevant constant rate to support critical encryption applications such as video meetings and calls. However, achieving this stability is not a straightforward task. Firstly, the QKD procedure itself is inherently stochastic. For instance, in the BB84 protocol [6], the sifting phase randomly filters out about half of the qubits, leading to a probability output rate. In the experiments, we observe the Coefficient of Variation (CoV) of a QKD key rate is 1.52, indicating such fluctuation. Secondly, the key pool is used to cache the keys, but it serves multiple requests. It may occasionally become depleted and unable to fulfill incoming requests. Lastly, in specific scenarios, like satellite-based QKD networks, QKD only operates within a limited time.

One potential solution to address the challenge is to adopt Key Derivation Functions (KDFs) [7], [8] that are used in many classic secure protocols to extend the seed keys (QKD keys in this scenario) to larger-size session keys while guaranteeing security. However, the security property of KDFs used in QKD networks is different from the classic use cases. QKD keys are usually considered Information-Theoretic Security (ITS). Meanwhile, KDFs here need to derivate more keys using limited QKD keys considering its scarcity, and eventually introduce a security downgrade.

Another difference in applying classic KDFs directly for QKD networks is the I/O property, as shown in Fig. 1. On the one hand, the input key materials in QKD networks possess distinct properties. QKD secure keys (serve as the input of KDFs) are produced continuously at a low and unstable rate, requiring our proposed KDF to run continuously. In contrast, classic protocols usually use Diffie-Hellman exchange to obtain a single fixed-length key. On the other hand, the KDF's output keys in QKD networks should satisfy both

L. Chen, K. Xue, J. Li, Z. Li, and N. Yu are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China.

K. Xue and N. Yu are also with Hefei National Laboratory, Hefei, Anhui 230088, China.

Corresponding Author: K. Xue (kpxue@ustc.edu.cn)

security and real-time encryption requirements. Alternatively, traditional KDFs typically extend keys at a fixed time without requirements for stabilizing the extended key stream. Although several key expansion schemes have been proposed, they usually do not consider providing a secure, high-rate, and stable key stream. Bebrov *et al.* [9] proposed a similar key expansion method for MDI-QKD protocols, but it expands keys at a constant 1.5 ratio and does not stabilize the key rate. Zhang *et al.* [10] proposed a quantum tensor network for key expansion based on entanglement distribution, which is currently in the experimental stage. Huang *et al.* [11] proposed a stream Privacy Amplification (PA) to guarantee QKD's output security, but they do not output secure keys at a user-desired stable rate.

In this article, we propose a Continuous and Security Key Derivation Function for Quantum Key Distribution (Q-CSKDF) for the aforementioned challenges. It can operate at both the network layer (e.g., in QKD network devices) or the application layer (e.g., integrated into specific applications) to supply high-rate secure keys. Q-CSKDF operates continuously over continuous periods, and each period consists of the following three schemes: Firstly, Q-CSKDF employs a dynamic sampling technique to collect secure keys from QKD devices. Due to the inherent instability of QKD key generation, the amount of collected keys may vary. Q-CSKDF determines the optimal sampling period based on the input/output key rate and the security threshold. Secondly, it dynamically determines the expansion ratio based on theoretical security analysis. The idea is to extend secure keys at the desired output key rate whenever it satisfies the security threshold. Lastly, Q-CSKDF adopts our key derivation module inspired by the widely used HMAC [8] schema. Note that we re-design HMAC to better leverage the ITS property of QKD secure keys and enhance the dynamic security level across multiple periods. With the above three schemes, Q-CSKDF can efficiently derivate a stable high-rate key stream with a corresponding computational security analysis for key derivation. We further conduct extensive experiments that use independent QKD and key management devices to expose the detailed QKD procedures. Our platform can also optionally retrieve secure keys from real QKD devices. The results demonstrate that the Q-CSKDF effectively utilizes scarce and unstable QKD key resources, generating high-quality key streams with guaranteed security. The contributions of the article are summarized as follows:

- We propose Q-CSKDF to provide secure and high-quality keys for QKD networks, using a continuous model and dynamic sampling technique to utilize the scarce and unstable QKD keys. We determine the sampling period based on the analysis of QKD procedures.
- We provide a dynamic key expansion design with comprehensive security analysis from a full-period perspective. With a dynamic key expansion ratio that considers both real-time encryption and security requirements, Q-CSKDF can provide keys at a desired output rate while ensuring security.
- To demonstrate the efficient performance of Q-CSKDF,

we conduct semi-physical experiments that expose the detailed QKD secure key generation. The results confirm the high performance achieved by our proposed scheme.

This article is organized as follows. In Section II, we brief the background and the problem statement. Then, we present our KDF design and corresponding analysis in Section III. We conduct experiments to demonstrate the performance in Section IV, and conclude this article in Section V.

## II. BACKGROUND AND PROBLEM STATEMENT

### A. QKD Networks

QKD is a cryptographic protocol designed to establish secure keys between two entities. It leverages the fundamental properties of quantum physics, such as the quantum uncertainty principle, to ensure ITS. The pioneering QKD protocol, known as the BB84 protocol [6], was developed by Bennett and Brassard. Subsequently, MDI-QKD [12] and TF-QKD [13] aim to enhance security and enable longer communication distances. However, the practical implementation of QKD devices is constrained by hardware limitations, resulting in a limited SKR, which is typically below 100 kbps [4] in the constructed QKD network.

QKD networks are designed to deliver QKD services to more users and expand coverage areas. Like classical networks, quantum networks are often structured using a layered architecture [1]. One possible architecture consists of three layers. The first layer is the QKD layer, where the QKD devices are responsible for generating secure keys. The second layer is the Key Management Service (KMS) layer. It contains a key pool structure to cache secure keys and serve users. Additionally, leveraging key repeater technologies, the KMS layer also serves secure keys between non-neighbour nodes. Lastly, the upper layer is the controller layer that governs the entire network. These layers significantly impact the secure key supplement and the performance of the proposed Q-CSKDF. Accordingly, we conduct comprehensive analyses of the dynamic key supplement in theory and experiments.

### B. Problem Statement: Tailored KDF for QKD networks

KDF is a cryptography algorithm that derivates key materials (or source for short) into the session keys. These session keys are further used in the ciphers. Based on the function, KDFs have two categories: 1) The first category of KDFs includes HKDF [8], SS-KDF, and *etc.*. They derivate the fixed-length binary master keys into several session keys. 2) The second category refers to the password-based KDFS [14]. They convert human-readable passwords into binary format secure keys. PBKDF2 is the one option in this category.

In this article, we design a KDF to extend QKD secure keys to provide high-quality keys for QKD networks. However, we observe that both of the two categories are unsuitable for QKD network scenarios, specifically in terms of I/O and security aspects. Regarding the I/O aspect, there are a few considerations to address. Firstly, classic KDFs accept fixed-length strings as input, whereas QKD devices continuously provide QKD keys, which may vary in quantity over

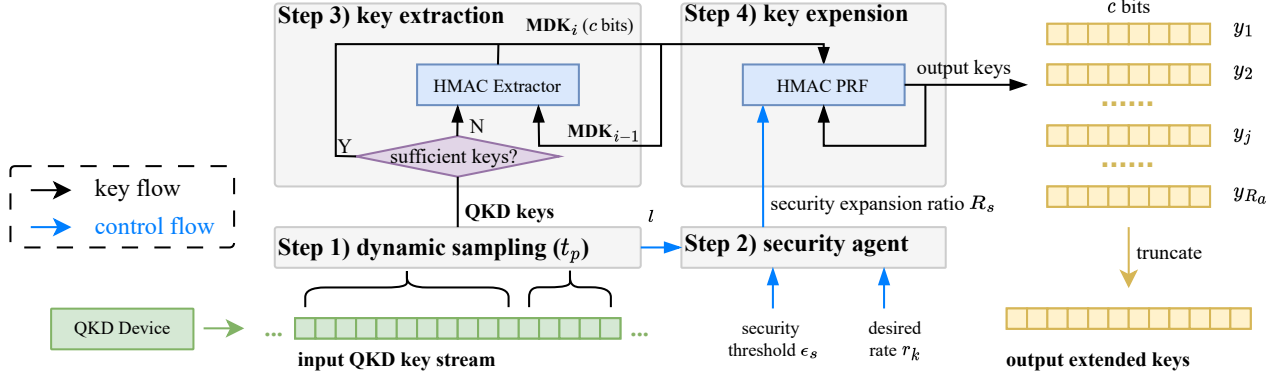


Fig. 2. Structure of the proposed Q-CSKDF algorithm.

time. Assuming that the QKD device generates secure keys periodically, with each period lasting  $t_p$  times. In each period, the QKD device generates keys following a distribution denoted as  $\mathcal{S}$ . For instance, in BB84 protocol, it roughly conforms to a Binomial distribution where  $\mathcal{S} \sim f_{pp} \cdot B(r_s t_p, 1/2)$ , where  $r_s$  is the physic level qubit reception rate in BB84 protocol,  $1/2$  is the probability of successful sifting, and  $f_{pp}$  denotes the depletion during post-processing. Note that we use the BB84 protocol as an example to analyze the performance, but our scheme can also work for other QKD protocols without further adaptation. Secondly, considering the output side, traditional KDFs produce keys with a predetermined length. However, to meet the real-time encryption requirement, the proposed Q-CSKDF should generate derived keys at a desired key rate  $r_k$ , *i.e.*, extend the keys to a length of  $r_k t_p$  in each period.

The security property also differs from that in the class KDFs. We require the Q-CSKDF to have multiple-period security, or  $(t, q, N, \epsilon)$ -secure with respect to a QKD key resource  $\Sigma$  runs for  $N$  periods if no attacker  $\mathcal{A}$  running in time  $t$  and making at most  $q$  queries can win the distinguishing game (similar to  $(t, q, \epsilon)$ -secure defined in [8]) with probability larger than  $1/2 + \epsilon$ . The three differences are:

- 1) Single-period security should also be considered as it guarantees the key's security in each period and is the foundation of achieving multiple-period security. Similar to the security definition in classic KDFs, a  $(t, q, 1, \epsilon_s)$ -secure should also be met in any period, where  $\epsilon_s$  is a security threshold for a single-round game.
- 2) Security in Q-CSKDF depends on the distribution of QKD key materials  $\mathcal{S}$ . The security of classical KDFs is tied to the entropy of the key material, and they usually require the input key material to possess a minimum entropy of  $m$  bits. However, meeting this requirement becomes challenging in QKD scenarios due to the variable amount of QKD keys. Accordingly, Q-CSKDF's security is constructed over  $\mathcal{S}$  instead of a fixed length of  $m$  bits threshold.
- 3) Given that QKD keys are statistically indistinguishable from a random binary string due to the quantum inherent randomness, we assume a  $l$ -bit QKD key is a random binary vector with  $l$  bit entropy.

Consequently, the above I/O and security properties in QKD secure keys are different from these classic networks, but the proposed Q-CSKDF should still achieve the following two goals: **security** and **efficiency**. Our foremost goal is to ensure **security** by meeting a predetermined security threshold. For example, if the QKD key material generated within a given period is insufficient to satisfy the required security level, the Q-CSKDF should either terminate the key derivation process or produce a limited number of keys. Furthermore, we aim to achieve **efficiency** by extending the keys at a desired output rate  $r_k$ . It is crucial for Q-CSKDF to utilize QKD keys efficiently, given their scarcity.

### III. Q-CSKDF DESIGNS

#### A. Overview

The proposed Q-CSKDF operates continuously. It reads QKD key streams and generates extended keys at a period  $t_p$ . As shown in Fig. 2, Q-CSKDF consists of four main components: dynamic sampling, security agent, key extraction, and key extension.

The dynamic sampling collects QKD keys from a key pool at a fixed period  $t_p$ . The collected keys serve as the key materials for subsequent derivation. During the dynamic sampling, Q-CSKDF considers the scarcity and fluctuation of the QKD keys and only collects a restricted number of keys while ensuring both security and efficiency requirements. Further details regarding the designs are in Section III-B.

The key extraction and key extension components in Q-CSKDF adopt an Extract-Then-Expand structure inspired by the HKDF algorithm. However, we introduce several innovations tailored to QKD scenarios. Firstly, we employ a segmentation strategy in the extraction step to improve efficiency. When a sufficient number of QKD keys is available, Q-CSKDF fully leverages the inherent ITS property of QKD keys. However, when the QKD keys are insufficient, we employ a cross-period extraction approach to supply entropy over multiple periods. This segmentation strategy enables us to meet security requirements while minimizing the consumption of QKD keys. Secondly, we introduce an additional security agent that dynamically determines the expansion ratio, which indicates the number of keys to be derivated in each period. This decision is based on various factors, including the

quantity of input QKD keys, the desired security threshold, and the desired output key rate. The key extraction process is presented in Section III-B, while the key expansion mechanism is explained in Section III-C. The security agent and security analysis are discussed in Section III-D. Lastly, in Section III-E, we optimize and evaluate the performance in Q-CSKDF, such as the collecting period  $t_p$  and the average output key rate.

### B. Dynamic Sampling and Key Extraction

Dynamic sampling periodically collects QKD keys from a key pool. The key extraction process extracts the entropy from these input QKD keys to produce a fixed-length Master Derivation Key (MDK). MDK serves as the seed for producing extended keys in the current period. Let  $|\text{MDK}|$  represent the length of MDK, and  $\text{MDK}_i$  denote the MDK in the  $i$ -th period. Without losing generality, let  $\text{MDK}_0$  be  $\{0\}^{|\text{MDK}|}$ . Additionally, we denote  $s_i$  as the QKD key in the  $i$ -th period.

The MDK generation contains two primary challenges: 1) When the key pool contains enough keys, how many keys should be used in one period? 2) What should be done when there are insufficient keys? To handle these questions, Q-CSKDF adopts a segmentation strategy. When the QKD key pool contains more than  $|\text{MDK}|$  bits of keys, Q-CSKDF directly collects exactly  $|\text{MDK}|$  bits of keys and utilizes them as the MDK. In this case, Q-CSKDF uses  $s_i$  directly as MDK since QKD keys are ITS, and the privacy amplification procedure already has a similar function to key extraction. Moreover, by utilizing a maximum of  $|\text{MDK}|$  bits of keys, Q-CSKDF ensures that no QKD keys are compressed, thereby optimizing the utilization of the scarce QKD keys. Unused QKD keys can be reserved for subsequent periods, thereby enhancing the stability of SKR.

Otherwise, when the QKD key pool contains fewer than  $|\text{MDK}|$  keys, we introduce a cross-period extraction approach. Generally speaking, insufficient QKD keys cannot provide enough entropy in the current period. Thus, Q-CSKDF leverages the keys from previous periods as complementary entropy sources by utilizing the HMAC algorithm to aggregate entropies from two distinct key sampling periods, ensuring the generation of a secure and reliable MDK:  $\text{MDK}_i = \text{HMAC}(H, s_i, \text{MDK}_{i-1})$ , where  $H$  is the hash function, and  $s_i$  is all available keys in the key pool. Now we require the length of MDK to be the same as the output length  $c$  of the hash function  $H$  (i.e.,  $|\text{MDK}| = c$ ). For example, if SHA2-256 is used as  $H$ ,  $|\text{MDK}|$  should be 256 bit.

### C. Key Expansion

The key expansion employs MDK to generate derived keys. The major challenge in this stage involves the determination of the expansion ratio  $R_a$ , the ratio of the length of the MDK to the length of the derived keys. It should be determined considering both security and efficiency requirements.

The real-time encryption requirement of Q-CSKDF requires the derivation of keys at a stable and desired rate  $r_k$ . This requirement can be quantified as:

$$R_i = \lceil \frac{t_p r_k}{c} \rceil, \quad (1)$$

where  $R_i$  represents the ideal expansion ratio. This equation ensures that over a time of  $t_p$ , the Q-CSKDF generates at least  $t_p r_k$  keys.

However, Eq. (1) alone does not guarantee security when the collected QKD keys  $s_i$  do not possess sufficient entropy to be expanded to  $t_p r_k$  derived keys. Therefore, We propose an additional secure expansion ratio  $R_s$  to restrict the key expansion. It is a dynamic variable that depends on both the user's desired security level and the number of input QKD keys. The specific methodology for calculating  $R_s$  will be presented in Section III-D.

Finally, the actual expansion ratio  $R_a$ , is constrained by both  $R_i$  and  $R_s$  (i.e.,  $R_a = \min(R_i, R_s)$ ). A similar procedure as in HKDF is adopted to produce the derived keys. In each period, it produces  $R_a$  blocks of output derived keys. Let  $y_j$  be the  $j$ -th block of the output keys, we use the following HMAC feedback mode  $y_j = \text{HMAC}(H, \text{MDK}_i, y_{j-1} || \text{ctx} || j)$  to produce the next block of the output keys, where  $\text{ctx}$  is an optional context-specific string defined in HMAC [8]. Finally, we concatenate all blocks  $y_1, y_2, \dots, y_{R_a}$  and truncate it to at most  $t_p r_k$  bits as the output derived keys for the current period. Here,  $\text{MDK}_i$  is utilized as the seed for this specific period and  $y_0$  is set to  $\{0\}^c$ .

### D. Security Analysis

In this section, we provide a brief overview of evaluating secure expansion ratio  $R_s$  and proof of the security for Q-CSKDF. We use a non-asymptotic method to evaluate the possibility that the adversary to conquer the Q-CSKDF is lower than the secure threshold  $\epsilon_s$  and  $\epsilon$  for a single round and multiple rounds, respectively.

**The security analysis in one single period.** Similar to HKDF, the security of single-period Q-CSKDF is related to two aspects:  $\epsilon_s = \epsilon_x + \epsilon_e$ . Here,  $\epsilon_x$  represents the security in the extraction step, while  $\epsilon_e$  denotes the security during the expansion step. However, Q-CSKDF contains two major modifications compared to the standard HKDF.

First, Q-CSKDF requires a larger expansion ratio. In HKDF, the maximal expansion ratio is typically limited to 255 as the counter  $j$  is encoded in one octet. Q-CSKDF uses HMAC in a feedback mode to construct a variable-length output PRF and requires a larger expansion ratio. It raises the adversaries' queries  $q$  [15], and the security of an HMAC-based PRF degrades quadratically with an increase in queries. Consequently, given a specific expansion ratio  $R_s$ , it is possible to determine the maximum  $q$  and  $\epsilon_e$ , respectively.

Second, the QKD key collected  $s_i$  does not guarantee a fixed minimum entropy and further influences the security threshold  $\epsilon_x$ . For instance, when the compression function  $h$  satisfies  $\delta^h$ -almost universality (AU), HMAC serves as a  $\epsilon_x = \sqrt{2^{-l} + \delta^h}$ -secure extractor [8], where  $l$  denotes the entropy of the input keys. Note that when the number of QKD keys is smaller than  $c$ , Q-CSKDF introduces a cross-period extraction approach to supply the entropy. Therefore, the above evaluated  $\epsilon_x$  is a lower bound.

Consequently, we find that the maximum expansion ratio  $R_s$  of a single period is related to the single-period security

threshold  $\epsilon_s$ . For example, when the compression function  $h$  of  $H$  is a  $\epsilon^h$ -secure PRF and a  $\epsilon_{NA}^h$ -secure PRF, and the QKD keys are ITS, previous research has demonstrated that HMAC functions as a  $\epsilon_e$ -secure PRF [15]. Consequently, it is feasible to obtain a concise estimation of  $(t', q, 1, \epsilon_s)$ -security, where  $\epsilon_s \leq \epsilon_x + \epsilon_e$ , against a probabilistic polynomial-time adversary with  $t'$  time and  $q$  queries [8]. Notably, the aforementioned analysis pertains to NMAC (a generalized variant of HMAC) but can be reduced to HMAC. Overall, given a desired security level  $\epsilon_s$ , it is possible to evaluate the adversary's query  $q$  and security expansion ratio  $R_s$ , respectively. The specific form of the equation relies on the security properties of the hash functions  $H$  and QKD keys.

**The security of Q-CSKDF over multiple periods.** Now, we take the distribution of QKD keys  $\mathcal{S}$  into account. Let's define the probability of having fewer QKD keys than  $|\text{MDK}|$  as  $p_c = Pr[\mathcal{S} < c]$ . The single-period security with respect to  $\mathcal{S}$  can be expressed as  $\epsilon_s^S = (1 - p_c)\epsilon_s + p_c$ . Considering a simplified low-bound model where we disable the cross-period entropy supplement, it means that all periods are operated independently. In this model, all  $N$  periods are secured with the possibility of  $\epsilon_i$ , that is  $\epsilon_i \leq 1 - (1 - \epsilon_s^S)^N \leq N\epsilon_s^S$ . Since our Q-CSKDF uses the cross-period entropy supplement to achieve higher entropy in each period, and thus achieves  $\epsilon \leq \epsilon_i \leq N\epsilon_s^S$ -security.

### E. Performance Analysis

We now analyze the performance of Q-CSKDF, including determining the sampling period  $t_p$  and the average expansion ratio.

The key sampling period  $t_p$  affects the Q-CSKDF's security and efficiency. For example, if the QKD SKR is high,  $t_p$  can be shrunk to enhance security. Or, when the QKD keys exhibit large fluctuations, it should extend  $t_p$  to prevent occasional shortages of keys during certain periods. Accordingly, we employ the Value at Risk (VaR) model to determine  $t_p$ , which ensures that the probability of collecting fewer than  $c$  bits QKD keys is smaller than a predefined threshold  $\mu$ . For example in BB84 protocol,  $\mathcal{S}$  approximately follows a normal distribution  $\mathcal{S} \sim f_{pp} \cdot N(\frac{1}{2}r_s t_p, \sqrt{\frac{1}{4}r_s t_p})$  (a binomial distribution but approximate to a normal distribution). Thus, the following formulation is formed to determine the  $t_p$ :

$$\frac{1}{2}r_s t_p + \Phi^{-1}(\mu)\sqrt{\frac{1}{4}r_s t_p} \geq c/f_{pp}, \quad (2)$$

where  $\Phi^{-1}$  is the inverse of the Cumulative Distribution Function (CDF) of  $N(0, 1)$ . For example,  $t_p \geq 1574.53/r_s$ , when  $c = 256$ ,  $\mu = 0.01$  and  $f_{pp} = 0.3$ .

Furthermore, we calculate the maximal achieved derived key rate using the formula  $r_{kM} = \frac{cR_s}{t_p}$ . Considering the average QKD SKR  $= \frac{f_{pp}}{2}r_s$ , we present the maximal expansion ratio  $R_{sM} = \frac{r_{kM}}{\text{SKR}} = \frac{2cR_s}{f_{pp}r_s t_p}$ . In the above example,  $R_{sM} \approx 1.084R_s$ , indicating the high key utility efficiency.

The above analysis is based on the BB84 protocol as an example. However, it can also be adapted to other protocols. On one hand, a similar probabilistic analysis can be employed to model the probability distribution of the QKD key rate

and to further analyze the appropriate sampling period  $t_p$ . On the other hand, the determination of  $t_p$  can be achieved experimentally, as users can observe the key generation and find a time period that the probability of achieving a key length of  $|\text{MDK}|$  is close to 1.

## IV. EVALUATIONS

In this section, we perform comprehensive experiments to evaluate the performance and security of Q-CSKDF under various conditions. We construct a semi-physical experimental testbed, as depicted in Fig. 3. It is a full-stack QKD communication system involving two separate nodes. Each node has three distinct devices: the QKD device, the KMS device, and the Q-CSKDF module. All devices operate independently in real-time and communicate with each other using the same protocols used in real QKD products. To expose the detailed key production, it can either simulate the detailed QKD behaviors (including the sifting and post-processing) or read physical keys from commercial QKD devices directly as it implements the real protocols.

In most experiments, we set the parameter  $f_{pp}$  to a typical value of 0.3 in many QKD devices. Consistent with real devices, these keys are transmitted to the KMS in blocks of 1024 bytes. In Q-CSKDF, we employ the SHA2-256 as the hash function and set the parameter  $\mu = 0.01$ . In the experiments, we use SHA2-256 as an implement of an ideal hash function and assume it is  $2^{-128}$ -security considering the brute-force collision. Other SOTA attacks can be easily merged into our security analysis, or users can intuitively use other hash functions such as SHA3 or BLACK2 in practice.

### A. Experiments for Sampling Period

We first illustrate the impact of the period  $t_p$  on the overall performance of the Q-CSKDF. The QKD device generates keys at an average SKR = 1 kbps, and we observe that QKD keys are sent to the key pool in 1024-byte blocks at approximately  $8.44 \pm 0.05$  seconds. We vary the period  $t_p$  of the Q-CSKDF from 100ms to 400ms, and the theoretical period, in Eq. (2), is about 242ms. Here, we do not care about the expansion ratio and fix it to 1, leading to QKD keys not being derived.

We introduce two metrics to evaluate security and efficiency. The first metric is  $p_c$  mentioned in Section III-D with  $l_c = c = 256$ . It represents the probability that the key pool does contain a sufficient number of QKD keys for security. The second metric is the key utilization  $U$  as the ratio of the consumed QKD keys to the total number of produced QKD keys. A higher key utilization indicates that the Q-CSKDF effectively uses the QKD keys.

The results are illustrated in Fig. 4. Fig. 4(a) depicts the  $p_c$  and  $U$  under different  $t_p$ . We observe  $p_c$  initially increases and then approaches 1 as  $t_p$  reaches 250ms, which is close to the theoretical value of 242ms. When  $p_c$  exceeds 250ms, it becomes highly probable to obtain a sufficient number of QKD keys from the key pool. However, we also notice that  $U$  decreases because the keys are generated faster than they are utilized, resulting in underutilization of the QKD keys. In

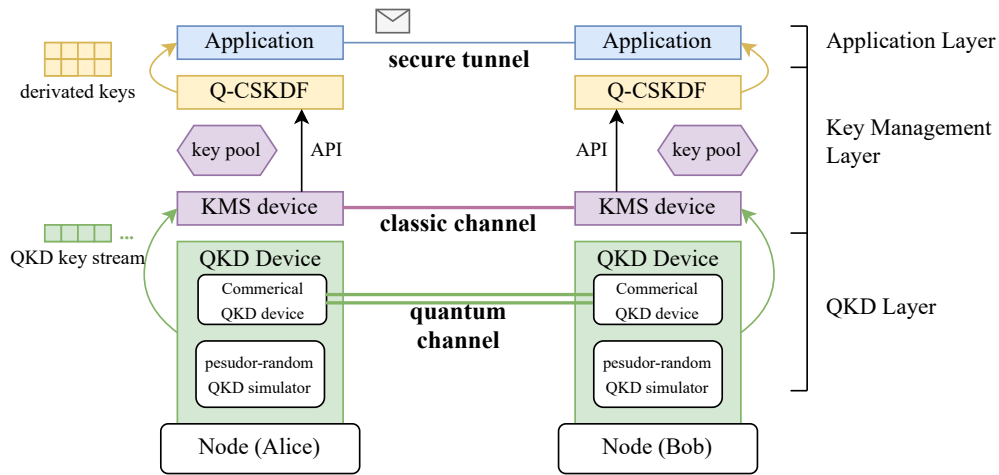
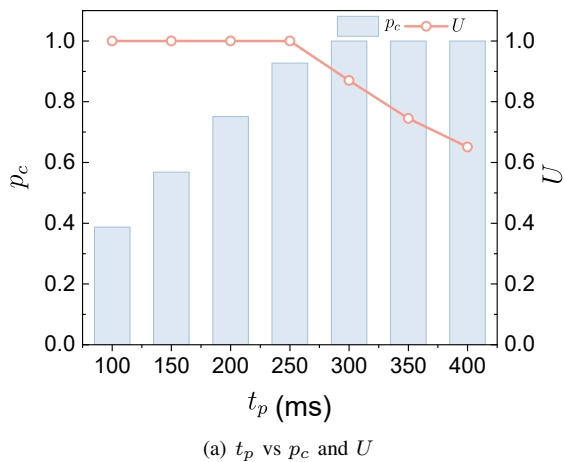
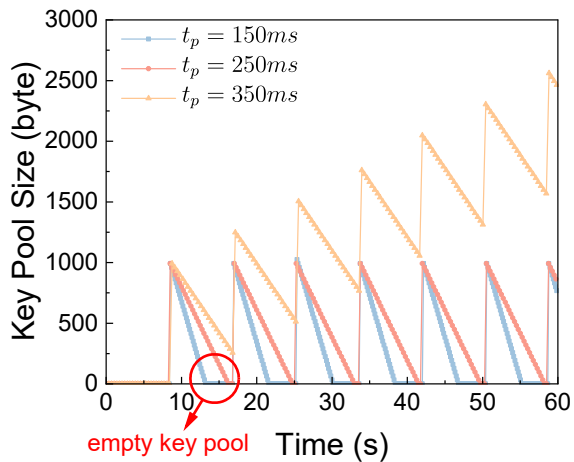


Fig. 3. The architecture of the semi-physical experiment environment



(a)  $t_p$  vs  $p_c$  and  $U$



(b) Available keys in key pool

Fig. 4. Experiments on the sampling period  $t_p$

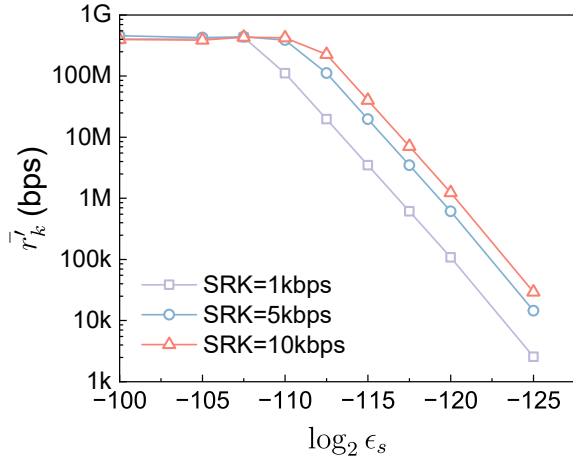
Fig. 4(b), we provide a detailed view of the key pool status. When  $t_p$  is set to 150ms, the keys run out periodically, leading to approximately 43.1% of the periods not receiving any QKD keys, as shown in the red circle. Oppositely, when  $t_p$  is 350ms, the key pool experiences an overstocking of keys. The best key efficiency is observed when  $t_p$  is 250ms, where QKD keys are consumed optimally, with all keys being utilized by the Q-CSKDF. Only a few periods experience key insufficient due to fluctuations in key generation. In conclusion, this experiment confirms that the theoretical value derived from Eq. (2) achieves the best performance.

### B. Experiments for Security Threshold

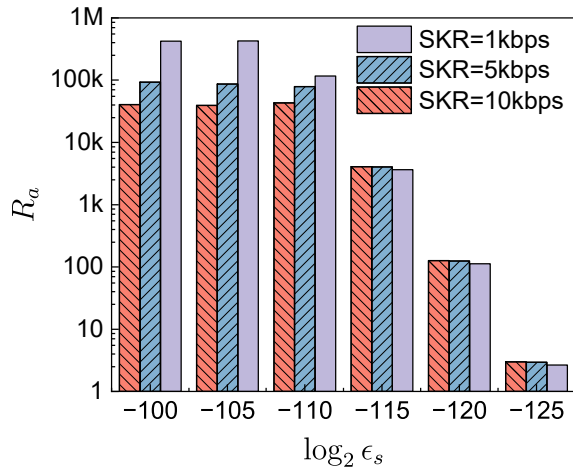
We evaluate the security of the Q-CSKDF. The QKD keys are generated at SKR = 1 kbps, 5 kbps, and 10 kbps, while we require the Q-CSKDF to produce extended keys at a desired rate of  $r_k = 1$  Gbps under different security thresholds. The results are given in Fig. 5. We observe that the produced secure key rate starts to decrease at about  $\epsilon_s \approx 2^{-105}$  and reaches close to 0 when  $\epsilon_s \leq 2^{-125}$ .

In Fig. 5(a), we show the relationship between the Q-CSKDF's average actual output key rate  $r'_k$  and the security threshold  $\epsilon_s$ . When  $\epsilon_s$  is larger than  $2^{-105}$ ,  $r'_k$  remains constant at about 400 Mbps, regardless of the QKD key rate. It indicates that the security threshold is not reached. However, it does not achieve the desired 1 Gbps due to hardware limitations. This experiment is conducted in real-time using a consumer Intel CPU, which imposes constraints on the computational capabilities. Despite this limitation, a 400 Mbps key stream is still sufficient for most encryption scenarios. Furthermore, such limitations can be overcome by utilizing more computing resources or parallel computing.

Additionally, we find that as  $\epsilon_s$  decreases, the derived key rate  $r'_k$  exhibits an exponential decrease. Moreover, when more QKD keys are utilized, the decline in  $r'_k$  occurs later and is less pronounced. For example, when the input QKD SKR is 10 kbps, the output rate is 29.4 kbps, whereas the output rate drops to 2.5 kbps when the input key rate is reduced to 1 kbps. This result is quite intuitive, as more QKD production rate



(a) The produced key rate  $r_k$  vs  $\epsilon_s$



(b) The expansion ratio  $R_a$  vs  $\epsilon_s$

Fig. 5. Experiments results for the security threshold

reduces the period  $t_p$ . Consequently, given a fixed expansion ratio  $R_s$  (constrained by  $\epsilon_s$ ), more periods can generate more derived keys.

For a detailed analysis, we conduct experiments as shown in Fig. 5(b), illustrating the relationship between the expansion ratio  $R_a$  and the security threshold. It shows that a higher QKD SKR effectively reduces the  $R_a$  within a single period, subsequently lowering the security requirements when  $\epsilon_s$  surpasses  $2^{-110}$ . Moreover, we demonstrate that as  $\epsilon_s$  reaches  $2^{-110}$ , the expansion ratio  $R_a$  becomes independent of the QKD SKR. When  $\epsilon_s = 2^{-120}$ ,  $R_a$  approximates 110. In such scenarios, the output key rate exhibits a proportional relationship with the input key rate.

### C. Experiments for Overall Performance

Finally, we demonstrate the performance of the proposed Q-CSKDF. We fix  $\epsilon_s$  to  $2^{-120}$ , while the corresponding  $R_s$  is approximately 127. For the QKD devices, we vary the SKR to 2.5, 5, 7.5, and 10.0 kbps. We verify whether the desired rates  $r_k$  can be achieved. We also observe the stability of the output keys.

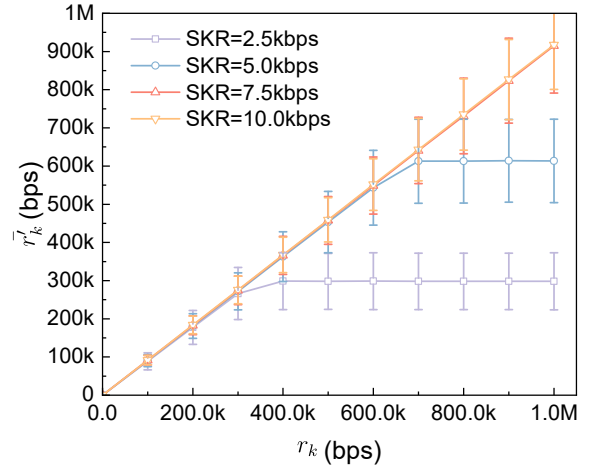


Fig. 6. Experiments results on the extended key rate

The results are illustrated in Fig. 6. Q-CSKDF is capable of achieving the desired key rate  $r_k$  whenever the security threshold is allowed. The actual key rate  $\bar{r}_k$  attained is 92.6% of  $r_k$ . It slightly deviates from 100% due to fluctuations in QKD key supply. It also shows that as the QKD SKR increases, Q-CSKDF is able to achieve a higher  $\bar{r}_k$ . For example, when the QKD SKR is 2.5 kbps, the  $\bar{r}_k$  reaches approximately 313.7 kbps. When the QKD SKR increases to 5.0 kbps, the  $\bar{r}_k$  can be up to 623.8 kbps.

The key rate stability is also evaluated and presented as error bars in Fig. 6. We observe that with a higher desired output key rate  $r_k$ , the stability drops. Also, if the SKR increases, the fluctuation is reduced (compared to the corresponding SKR). Besides, we also use the Coefficient of Variation (CoV), the ratio of the mean to the standard deviation of a key stream, to quantify the fluctuation. The results show that Q-CSKDF can efficiently stabilize the key rate compared to the original QKD keystream. For example, a QKD key stream produces keys at an average SKR of 2.5 kbps, and its corresponding CoV reaches 1.52. However, after processing by Q-CSKDF, the CoV of the output keys reduces significantly to approximately 0.09, indicating that the Q-CSKDF is highly effective in stabilizing the key stream.

In conclusion, we show that Q-CSKDF can both increase the total amount of the keys within a desired security threshold and stabilize the key rate fluctuation.

## V. CONCLUSION

The scarcity and volatility of QKD keys constitute the pivotal factors hindering the practical implementation of QKD protocols. In this article, we introduced Q-CSKDF, a specialized key expansion method for QKD networks. It leverages a continuous QKD key stream as its input and generates an extended key stream at the desired rate. Q-CSKDF introduces a dynamic expansion ratio predicated on a comprehensive security analysis to ensure security through the entire derivation process. We also presented several mechanisms to enhance efficiency, including dynamic key sampling with a theoretical analysis. By conducting extensive

real-time and semi-physical experiments, we exposed the intricate structures underlying QKD key generation, and the results proved the high performance and security of Q-CSKDF in producing a stable and high-rate key stream.

#### ACKNOWLEDGEMENTS

This work is supported in part by Innovation Program for Quantum Science and Technology under Grant No. 2021ZD0301301, Anhui Initiative in Quantum Information Technologies under Grant No. AHY150300, and Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093.

#### REFERENCES

- [1] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S. X. Ng, and L. Hanzo, "The Evolution of Quantum Key Distribution Networks: On the Road to the Qinternet," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 839–894, 2022.
- [2] "OpenQKD," <https://openqkd.eu/>, 2024, Accessed: Apr., 2024.
- [3] Y. Yamamoto, M. Sasaki, and H. Takesue, "Quantum information science and technology in Japan," *Quantum Science and Technology*, vol. 4, no. 2, p. 020502, 2019.
- [4] Y.-A. Chen, Q. Zhang, T.-Y. Chen, W.-Q. Cai, S.-K. Liao, J. Zhang, K. Chen, J. Yin, J.-G. Ren, Z. Chen *et al.*, "An integrated space-to-ground quantum communication network over 4,600 kilometres," *Nature*, vol. 589, no. 7841, pp. 214–219, 2021.
- [5] W. Li, L. Zhang, H. Tan, Y. Lu, S.-K. Liao, J. Huang, H. Li, Z. Wang, H.-K. Mao, B. Yan *et al.*, "High-rate quantum key distribution exceeding  $110 \text{ mb s}^{-1}$ ," *Nature Photonics*, vol. 17, no. 5, pp. 416–421, 2023.
- [6] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum cryptography without Bell's theorem," *Physical Review Letters*, vol. 68, no. 5, p. 557, 1992.
- [7] E. Barker, L. Chen, and R. Davis, "Recommendation for key-derivation methods in key-establishment schemes," *NIST Special Publication*, vol. 800, p. 56C, 2018.
- [8] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *proceedings of the 30th Annual Cryptology Conference (CRYPTO)*. Springer, 2010, pp. 631–648.
- [9] G. Bebrov, "Key Expanding in Measurement-Device-Independent Quantum Key Distribution," *International Journal of Theoretical Physics*, vol. 60, no. 9, pp. 3566–3577, 2021.
- [10] Q. Zhang, H. Lai, and J. Pieprzyk, "Quantum-key-expansion protocol based on number-state-entanglement-preserving tensor network with compression," *Physical Review A*, vol. 105, no. 3, p. 032439, 2022.
- [11] Y. Huang, X. Zhang, and X. Ma, "Stream privacy amplification for quantum cryptography," *PRX Quantum*, vol. 3, no. 2, p. 020353, 2022.
- [12] H.-K. Lo, M. Curty, and B. Qi, "Measurement-device-independent quantum key distribution," *Physical Review Letters*, vol. 108, no. 13, p. 130503, 2012.
- [13] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, "Overcoming the rate–distance limit of quantum key distribution without quantum repeaters," *Nature*, vol. 557, no. 7705, pp. 400–403, 2018.
- [14] B. Kaliski and A. Rusch, "RFC 8018: PKCS# 5: Password-Based Cryptography Specification Version 2.1," 2017, RFC 8018, Accessed on Nov., 2023. [Online]. Available: <https://www.ietf.org/rfc/rfc8018.txt>
- [15] P. Gaži, K. Pietrzak, and M. Rybár, "The exact PRF-security of NMAC and HMAC," in *Proceedings of the 34th Annual Cryptology Conference (CRYPTO)*. Springer, 2014, pp. 113–130.

LUTONG CHEN (lutong98@mail.ustc.edu.cn) received his bachelor's degree from the School of Cyber Science and Technology, University of Science and Technology of China (USTC) in 2020. He is currently working toward a Ph.D degree in Network Security from the School of Cyber Science and Technology, USTC. His research interests include quantum networking and network security.

KAIPING XUE [M'09-SM'15] (kpxue@ustc.edu.cn) received his bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. Currently, he is a Professor in the School of Cyber Science and Technology, USTC. He is also a director of Network and Information Center, USTC. His research interests include future Internet architecture design, transmission optimization, network security, and quantum networking. He is an Fellow of the IET.

JIAN LI [M'20-SM'23] (lijian9@ustc.edu.cn) received his bachelor's degree from the Department of Electronics and Information Engineering, Anhui University, in 2015, and received his Ph.D degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2020. He is currently an associate researcher with the School of Cyber Science and Technology, USTC. His research interests include future Internet architecture design, satellite-terrestrial integrated networks, and quantum networking.

ZHONGHUI LI (leestone@mail.ustc.edu.cn) received his bachelor's degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, in 2018. He is currently working toward the Ph.D degree from the School of Cyber Science and Technology, University of Science and Technology of China. His research interests include quantum networking and network security.

NENGHAI YU (ynh@ustc.edu.cn) received his bachelor's degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987, the M.E. degree from Tsinghua University, Beijing, China, in 1992, and his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), Hefei, China, in 2004. Currently, he is a Professor in the School of Cyber Science and Technology, USTC. He is the Executive Dean of the School of Cyber Science and Technology, USTC, and the Director of the Information Processing Center, USTC. His research interests include multimedia security and quantum networking.