A New Perspective on Key Expansion for QKD BB84: The RanA Model

Thiago Luigi Mello[1], Isys Sant'Anna [1], Marcus Freire[1], Adriano Maia[1], Rodrigo Moreira[2], Roberto Rivelino[3], Maycon Peixoto[1]

[1]Instituto de Computação, Universidade Federal da Bahia (UFBA)
{thiagomello,isys.nogueira,marcus.elias,adriano.maia,maycon.leone}@ufba.br

[2]Instituto de Física Gleb Wataghin, Universidade Estadual de Campinas (Unicamp)
rirm@ifi.unicamp.br

[3]Instituto de Física, Universidade Federal da Bahia (UFBA)
rivelino@ufba.br

Abstract. *Quantum cryptography, exemplified by the BB84 protocol, ensures secure key distribution through quantum mechanical principles such as superposition and the no-cloning theorem. However, limitations in key generation rates and scalability, along with attenuation and noise in quantum channels, hinder its practical deployment in high-demand scenarios. This study proposes a hybrid approach that combines BB84 with classical key expansion, reducing dependence on quantum channels while maintaining efficiency and robustness against quantum attacks. The experimental results validate the scalability and security of the protocol.*

1. Introduction

Quantum computing exploits quantum mechanics to perform tasks beyond the capabilities of classical systems [Peixoto 2024, Maia et al. 2025]. This advancement has direct implications for quantum cryptography, a promising approach to ensuring information security in a world where quantum technologies are increasingly integrated into everyday life. Protocols like BB84 [Bennett and Brassard 2014] uses quantum mechanics characteristics, such as superposition and the no-cloning theorem [Wootters and Zurek 1982], to enable the distribution of secure encrypted keys. The growing viability of quantum computers represents a threat to classical cryptography systems, highlighting the importance of hybrid strategies to mitigate these vulnerabilities [Kamel et al. 2024, Nielsen and Chuang 2010].

Although BB84 are secure against eavesdropping, its low key generation rate is responsible for poor scalability in scenarios with high demand for secure traffic [Lee et al. 2022]. Furthermore, quantum channels are inherently susceptible to physical limitations such as attenuation and noise, which compromise the efficiency and practical feasibility of the protocol over long distances [Brassard et al. 2000].

Several approaches have been proposed to improve the efficiency and security of BB84. Among them, quantum channel optimization methods, integration with key derivation algorithms, and adoption of post-quantum schemes stand out. However, these solutions often present challenges, such as dependence on a sophisticated infrastructure, computational complexity, and high costs, limiting their practical application

[Rahmanpour et al. 2024, Kaur and Singh 2024, Jiang et al. 2021]. Despite recent advances, there remains a gap between the demands for large-scale security and the capabilities of current quantum protocols. In particular, the inability to generate keys long enough for high-demand applications and the exclusive reliance on quantum channels are critical gaps that require innovative approaches to overcome [Liao et al. 2022].

Thus, this work proposes the **RanA** (Random Amplification), a hybrid approach that combines the BB84 protocol with Key Derivation Function (KDF) [Biryukov et al. 2016]. The main contribution of this study is the development of a protocol for expanding the encryption key, increasing key generation rate and reducing dependence on the quantum channel, characteristics necessary for the adoption of QKD methods in modern applications.

2. RanA: Random Amplification Protocol for Quantum Key Augmentation

The advancement of quantum computing has increased the demand for secure protocols for communications, driving the development of QKD. Among these protocols, BB84 uses the no-cloning theorem to allow two parties, Alice and Bob, to establish a secure cryptographic key through qubit measurements: due to this theorem, some interception attempts by a potential attacker alter the transmitted quantum states, which makes them detectable by Alice and Bob, in order to guarantee the security of the communication [Wolf 2021]. As a consequence, the BB84 protocol allows the shared key to be used only if both Alice and Bob can verify its integrity.

After the raw key K generated in BB84 is generated, it goes through a reconciliation process to correct errors resulting from noise in the quantum channel or interception attempts. The CASCADE protocol, available at [Liao et al. 2022], is applied in this step, allowing Alice and Bob to obtain a corrected key K' and reducing discrepancies in the shared bit sequence, improving the transmission rate of secret keys in noisy scenarios. Overall, Figure 1 illustrates the key distribution and reconciliation process in RanA.

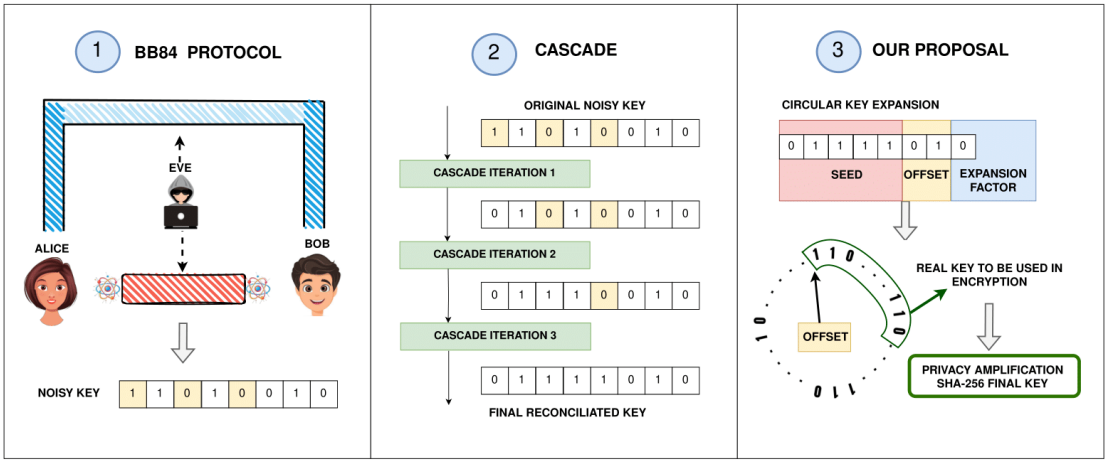

Figure 1. RanA schematic: on the left, the BB84 protocol performs error correction via CASCADE to reconcile the key between Alice and Bob. On the right, the post-quantum step expands the key and uses it to encrypt the message.

Since the protocol is one-time pad based, the amount of data that can be encrypted

directly by the generated keys is proportional to the key distribution rate, becoming a critical factor for the scalability of the system. In order to overcome this obstacle, RanA presents a hybrid scheme that combines BB84 with key derivation techniques, increasing the scalability of the system without compromising security.

2.1. Expanding BB84 Random Key

The corrected key K' is divided into three segments with fixed proportions, as illustrated in Figure 1. The first segment, called **seed** (S_{seed}), is used as input for the key derivation function (Argon2 and PBKDF2 are the algorithms that will be studied in our experiment). The second, called **offset** (S_{offset}), determines the offsets applied during encryption. Finally, the **expansion factor** (S_{exp}) defines the final size of the expanded key. Starting from K'_{seed}, Argon2 is applied to generate the expanded key K_{exp}. The size of the expanded key is defined by KeySizeExp $= |M| \times$ expandFactor, where $|M|$ is the message size, and expandFactor is calculated based on K'_{numExp}.

The derivation occurs iteratively until the required size is reached, increasing the entropy for encryption. To ensure synchronization between Alice and Bob, a circular shift defined by offset $\leftarrow$ BitsToInt$(K'_{\text{offset}}) \bmod |K_{\text{exp}}|$ is performed, where BitsToInt converts the bits of the segment to an integer. This value identifies the starting point of the key block used in encryption.

Alice selects from K_{exp} a block of size $|M|$ to XOR the message M, producing the ciphertext C. Algorithm 1 presents the correction, key expansion and validation hash generation, steps common to Alice and Bob that follow the generation of K via BB84.

Algorithm 1 Procedure for key expansion

Inputs:
K: Raw Key Generated by BB84
L: Key Size
S: Segmentation proportions
Output:
K_{segment}: Expanded Key

procedure RANA(K, L)
  $K' \leftarrow$ CASCADE(K) ▷ *Error correction*
  $K'_{seed} \leftarrow S_{seed} \times |K'|;\ K'_{offset} \leftarrow S_{offset} \times |K'|;\ K'_{exp} \leftarrow S_{exp} \times |K'|$ ▷ *segment of K'*
  ExpFactor $\leftarrow$ BitsToInt(K'_{exp}) % ExpFactor $>= 1$ ▷ *Expansion factor*
  ExpKeySize $\leftarrow L \times$ ExpFactor
  $K_{\text{exp\_initial}} \leftarrow \emptyset$ ▷ *Expanded key derivation (K_{exp})*
  $K_{\text{exp}} \leftarrow$ Concat(KDF$(K'_{seed}, t_cost, m_cost, parallelism, 256), ...)$[: ExpKeySize]
  offset $\leftarrow$ BitsToInt$(K'_{offset}) \bmod |K_{\text{exp}}|$ ▷ *Circular Deviation Calculation*
  $K_{\text{segment}} \leftarrow$ CircularKeySegment$(K_{\text{exp}}$, offset, $L)$
end procedure

2.2. Reception and Verification by Bob

Upon receiving the encrypted message C, Bob recreates the expanded key K_{exp}, using the same set of parameters in the KDF function of choice and the same form of segmentation applied by Alice during seed generation.

To determine the circular offset and extract the key block, Bob obtains $\delta = \text{BitsToInt}(K'_{\text{offset}}) \bmod |K_{\text{exp}}|$ and selects the segment $K_{\text{segment}} \subset K_{\text{exp}}$, whose size is $|C_{\text{dec}}|$. Bob then recovers the original message by computing $M_{\text{recovered}} = C \oplus K_{\text{segment}}$.

The protocol seeks to minimize dependence on the quantum channel by leveraging corrected keys to increase the volume of encrypted data in each session.

3. Experimental Design

To evaluate the effectiveness of the RanA protocol, a full factorial design 2^3 was employed, considering three factors, each at two levels, resulting in eight experimental conditions. The first factor is the key derivation algorithm applied after BB84, with levels PBKDF2 and Argon2. PBKDF2 was selected as a baseline due to its widespread use in cryptographic applications, while Argon2 was chosen for its superior memory-hardness and resilience against GPU-based attacks, providing enhanced security. The second factor refers to the proportion of the reconciled key allocated to the SEED during the derivation process, set at 50% and 75%. A 50% allocation was chosen to evaluate the impact of a balanced key split, while the 75% allocation aims to increase entropy for more secure key derivation, improving robustness in noisy environments. The third factor corresponds to the reconciled key size after error correction using the CASCADE protocol, evaluated at 100 and 150 bits. The 100-bit key size represents a typical trade-off between security and efficiency, while 150 bits was tested to assess the system's performance under higher security demands, simulating more challenging conditions.

The response variables analyzed were entropy of the derived key, avalanche percentage (which quantifies the sensitivity of the derivation process to small changes in the SEED), and execution time of the derivation procedure, reflecting the overall performance and security of the system. The factorial model is expressed as $Y = q_0 + q_A x_A + q_B x_B + q_C x_C + q_{AB} x_A x_B + q_{AC} x_A x_C + q_{BC} x_B x_C + q_{ABC} x_A x_B x_C$, where Y denotes the observed response variable, x_A, x_B, x_C are the coded levels of each factor, and the coefficients q represent the main and interaction effects.

To ensure statistical reliability, each experiment used ten distinct SEEDs, and for each SEED, 50 keys were simulated to represent the output of the BB84 protocol. This approach enabled the analysis of both individual factor effects and their interactions. The results highlight the system's behavior under different input conditions, supporting the selection of appropriate parameters in some quantum key distribution scenarios.

4. Results

Based on the experiments conducted, we selected the Argon2 algorithm as the key derivation function (KDF) and adopted a seed proportion of 0.75, considering the trade-off between computational performance and security.

The first three plots in Figure 2 presents the execution time comparison between the evaluated algorithms (Argon2 and PBKDF2) under different input conditions. It can be observed that PBKDF2 yields considerably higher execution times, exceeding 16 seconds in scenarios with larger key sizes, whereas Argon2 exhibits stable performance with execution times remaining below 0.06 seconds across all tested configurations.

Additionally, when analyzing the impact of the seed proportion used in the KDF input, it is evident that a value of 0.75 results in slightly lower execution times compared to 0.5, for both Argon2 and PBKDF2. For example, with Argon2, the execution time decreases from 0.0583s (seed=0.5) to 0.0571s (seed=0.75), while with PBKDF2 the reduction is more substantial (from 12.7372s to 9.4413s). Although the performance gain

in Argon2 is modest, this configuration does not compromise the entropy of the generated key. Thus, the selection of Argon2 aims to ensure efficiency and scalability, while the configuration with seed=0.75 represents fine-tuning that contributes to reduced key derivation time while preserving the proposed application's cryptographic security requirements.

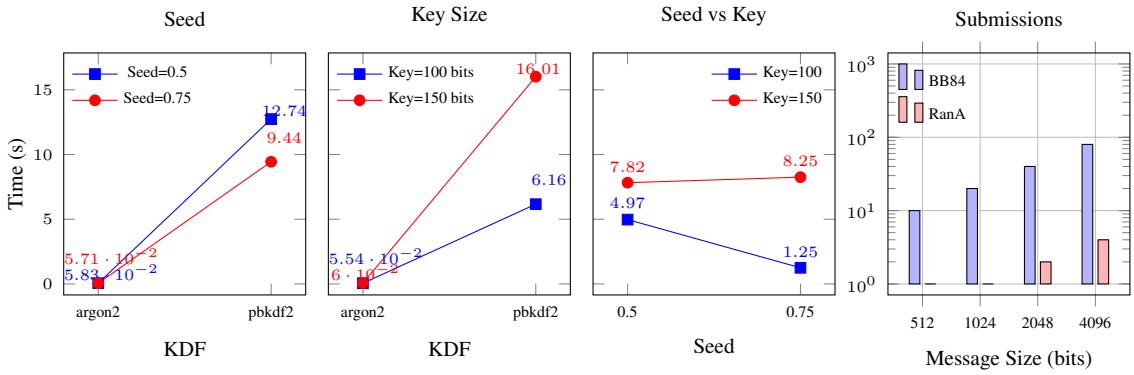

Figure 2. Execution time comparison by KDF algorithm, seed proportion, and key size.

The fourth plot in Figure 2 illustrates the relationship between the number of protocol executions and the message size for both BB84 and the RanA approach with package size of 1500 bytes. In the case of BB84, the number of execution cycles increases exponentially as the message size grows, resulting in significantly higher total transmission time. This occurs because key generation must be repeated several times before the message can be encrypted and transmitted. In contrast, RanA drastically reduces the number of executions by deriving the key directly to match the MTU size, eliminating the need for multiple execution rounds for shorter messages.

5. Conclusion

This work presented a comparative evaluation of key derivation function (KDF) configurations for secure key generation in environments with computational and communication constraints. We evaluated the performance and entropy of keys derived using Argon2 and PBKDF2, considering different seed proportions and key sizes. The results show that Argon2 provides lower and more stable execution times than PBKDF2, while maintaining high entropy across all configurations. Based on these observations, we adopted Argon2 with a seed proportion of 0.75 as a suitable configuration, balancing computational efficiency and cryptographic quality.

To address the limited bandwidth of quantum channels, we incorporated the RanA protocol for key expansion, enabling secure key reuse without compromising entropy or confidentiality. This integration reduces reliance on frequent raw quantum bit exchanges, enhancing the system's practicality. The main contributions of this study include analyzing lightweight and secure KDF configurations, justifying parameter choices with experimental evidence, and integrating classical post-processing (RanA) to alleviate quantum channel limitations.

Acknowledgements

This study was funded, in part, CAPES, Financing Code 001, by FAPESB, and by the National Council for Scientific and Technological Development (CNPq), Brazil, under

grant No. 403231/2023-0.

References

Bennett, C. H. and Brassard, G. (2014). Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11. Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84.

Biryukov, A., Dinu, D., and Khovratovich, D. (2016). Argon2: new generation of memory-hard functions for password hashing and other applications. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 292–302. IEEE.

Brassard, G., Lütkenhaus, N., Mor, T., and Sanders, B. C. (2000). Limitations on practical quantum cryptography. *Physical review letters*, 85(6):1330.

Jiang, Y., Liu, B., Guo, C., and Zhao, J. (2021). A quantum pseudo-random number generation scheme. *Journal of Physics: Conference Series*, 2004(1):012001.

Kamel, O. H. A., Raslan, A. T. N. E.-D., Aly, T., and Gheith, M. (2024). Quantum computing's impact on data encryption: Methodologies, implementation, and future directions: Exploring the bb84 protocol and comparative analysis with classical cryptographic techniques. In *2024 Intelligent Methods, Systems, and Applications (IMSA)*, pages 213–217. IEEE.

Kaur, H. and Singh, J. S. P. (2024). Software defined network implementation of multi-node adaptive novel quantum key distribution protocol. *AIMS Electronics & Electrical Engineering*, 8(4).

Lee, C., Sohn, I., and Lee, W. (2022). Eavesdropping detection in bb84 quantum key distribution protocols. *IEEE Transactions on Network and Service Management*, 19(3):2689–2701.

Liao, C.-T., Bahrani, S., da Silva, F. F., and Kashefi, E. (2022). Benchmarking of quantum protocols. *Scientific Reports*, 12(1):5298.

Maia, A. H. O., Freire, M., Melo, T., Rodrigues-Filho, R., Almeida, E. S., Prazeres, C. V. S., Figueiredo, G. B., and Peixoto, M. L. M. (2025). Q-edge: Leveraging quantum computing for enhanced software engineering in vehicular networks. In *40th ACM/SIGAPP Symposium On Applied Computing (ACM-SAC)*.

Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge ; New York, 10th anniversary ed edition.

Peixoto, M. L. M. (2024). Quantum edge computing for data analysis in connected autonomous vehicles. In *2024 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6.

Rahmanpour, M., Erfanian, A., Afifi, A., Khaje, M., and Fahimifar, M. H. (2024). A new quantum key distribution protocol to reduce afterpulse and dark counts effects. *Results in Optics*, page 100718.

Wolf, R. (2021). *Quantum Key Distribution: An Introduction with Exercises*, volume 988 of *Lecture Notes in Physics*. Springer, 1 edition.

Wootters, W. K. and Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature*, 299(5886):802–803.