

Article

Quantum Key Distribution with Post-Processing Driven by Physical Unclonable Functions

Georgios M. Nikolopoulos ^{1,2,*}  and Marc Fischlin ³ 

¹ Institute of Electronic Structure and Laser, Foundation for Research and Technology-Hellas (FORTH), GR-70013 Heraklion, Greece

² Institut für Angewandte Physik, Technische Universität Darmstadt, D-64289 Darmstadt, Germany

³ Cryptoplexity, Technische Universität Darmstadt, D-64289 Darmstadt, Germany; marc.fischlin@tu-darmstadt.de

* Correspondence: nikolg@iesl.forth.gr

Abstract: Quantum key distribution protocols allow two honest distant parties to establish a common truly random secret key in the presence of powerful adversaries, provided that the two users share a short secret key beforehand. This pre-shared secret key is used mainly for authentication purposes in the post-processing of classical data that have been obtained during the quantum communication stage, and it prevents a man-in-the-middle attack. The necessity of a pre-shared key is usually considered to be the main drawback of quantum key distribution protocols, and it becomes even stronger for large networks involving more than two users. Here, we discuss the conditions under which physical unclonable functions can be integrated in currently available quantum key distribution systems in order to facilitate the generation and the distribution of the necessary pre-shared key with the smallest possible cost in the security of the systems. Moreover, the integration of physical unclonable functions in quantum key distribution networks allows for real-time authentication of the devices that are connected to the network.

Keywords: quantum key distribution; physical unclonable functions; authentication



Citation: Nikolopoulos, G.M.; Fischlin, M. Quantum Key Distribution with Post-Processing Driven by Physical Unclonable Functions. *Appl. Sci.* **2024**, *14*, 464. <https://doi.org/10.3390/app14010464>

Academic Editors: Nuno Silva and Mariana Ferreira Ramos

Received: 23 November 2023

Revised: 26 December 2023

Accepted: 29 December 2023

Published: 4 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Quantum key distribution (QKD) is the most mature quantum technology [1–8]. Different QKD systems by various vendors are available on the market, and national and international initiatives aim to test the integration of the available QKD systems into existing communication infrastructures, as well their operation in various use cases under realistic conditions.

Typically, the establishment of a secret key between two honest users (Alice and Bob) pertains to a quantum communication and a post-processing stage (see Figure 1). In the first stage, Alice encodes random bits on non-orthogonal photonic states, which are sent to Bob over a quantum channel. Bob measures each received signal and records the outcomes. Subsequently, the two users publicly post-process their classical data in order to obtain a final information-theoretically secure secret key. The main steps involved in the post-processing stage are key sifting, error reconciliation, error verification, and privacy amplification. A successful implementation of all of these steps, without any detected failure, is expected to lead to a final secret key. However, for any of these steps, there is always a probability of undetected failure, in which case the security of the protocol is compromised.

While processing their data, the two users have to verify the origin of the exchanged messages, otherwise the protocol is prone to a man-in-the-middle attack outlined in Figure 2. In this rather simple attack, an adversary (Eve) cuts the quantum and classical service channels that connect Alice and Bob. She connects her QKD devices to the loose ends of the channels and pretends to be Bob to Alice and Alice to Bob. In this way, she can

establish two secret keys, one with Alice and one with Bob, which allow her, e.g., to decrypt and modify any message sent from Alice to Bob (or vice versa). Such an attack can be prevented only by authenticating the service channel, and given that QKD protocols promise information-theoretically secure (ITS) distribution of random keys, authentication by means of ITS message authentication protocols is the natural choice for QKD [9–11]. Standard ITS message authentication codes (MACs) [12–18], however, require that the two users share a secret truly random key before they run their QKD system for the first time (see also Appendix A).

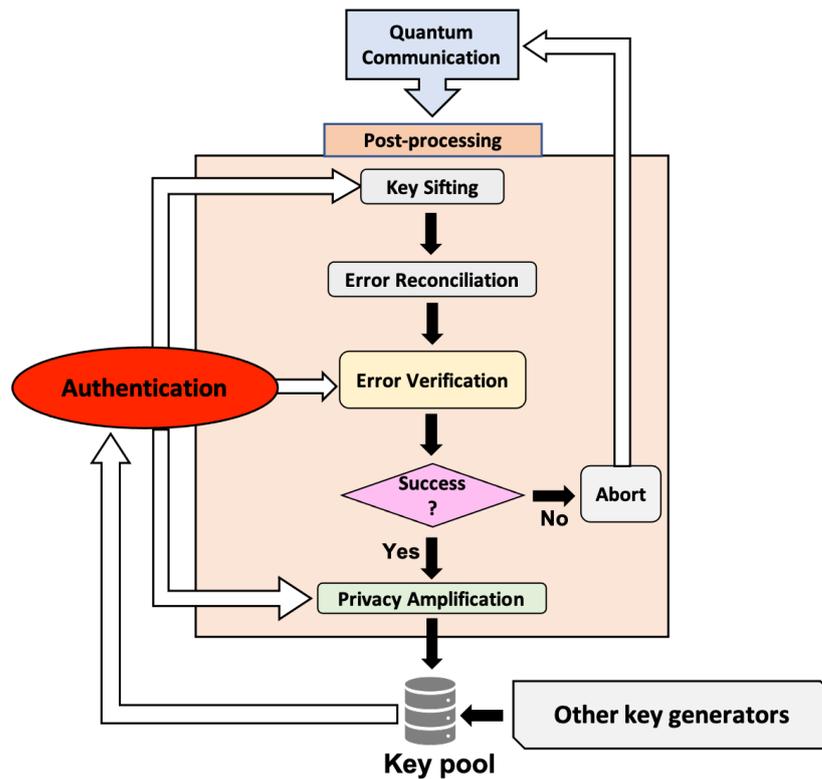


Figure 1. Schematic presentation of the main steps and the flow of data in a QKD protocol.

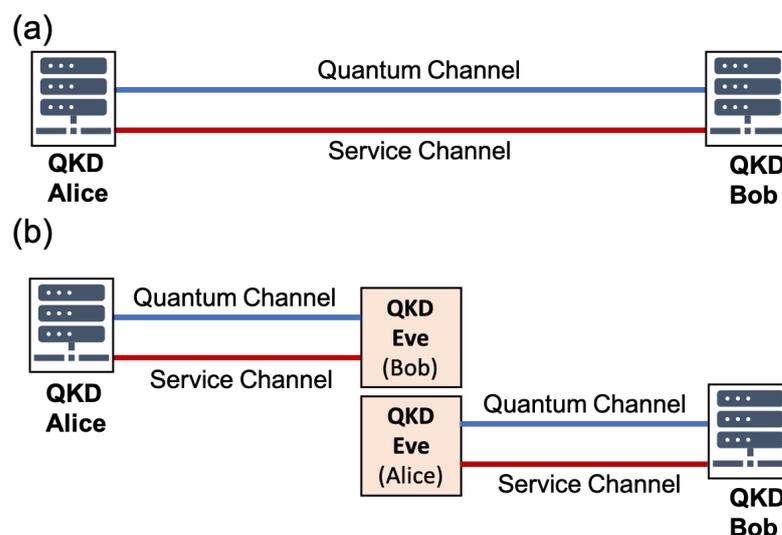


Figure 2. Schematic representations of a point-to-point QKD link (a) and of the man-in-the-middle attack (b).

For this reason, a typical commercially available pair (sender–receiver) of QKD devices comes with a pre-shared secret key, or the users who purchase the system have to generate the key by some means and transfer it to their devices. Among other tasks, this key is used for authentication in the first QKD session until a sufficient amount of fresh secret key material has been produced. From that point on, message authentication for the subsequent QKD sessions is expected to consume part of the existing secret key that has been generated through QKD. Clearly, a QKD protocol is meaningful only if key consumption is smaller than the generation of fresh key in a session. For large distances and/or lossy channels, however, the generation of fresh secret bits may become very costly in terms of the amount of the pre-shared key information required for their production. Another problem is that the need for a pre-shared secret key considerably complicates the design of large full-mesh QKD networks. These drawbacks, together with the strong dependence of QKD on ITS MACs, have made individuals as well as public authorities and organizations question the usefulness of QKD protocols and hold a negative position with respect to further proliferation of the technology.

The complications caused by the necessity for authentication and the management of pre-shared secret keys have not gone unnoticed by the community. When a pre-shared key is stored or distributed in plaintext, one cannot guarantee that a malicious third party will not obtain access to it and launch a successful man-in-the-middle attack in the first QKD session, which will compromise the security of the next sessions as well. The problem is not solved by encrypting the pre-shared key or by storing it in a password-protected electronic device, but rather it is transferred to the management and the distribution of an additional key (or password), which is needed in order to ensure the secrecy of the pre-shared key.

There have been efforts in the past to decrease the length of the pre-shared key in existing QKD protocols, as well as to facilitate the distribution and management of pre-shared keys. Peev et al. [19] proposed an authentication scheme which relies on a two-step hash function evaluation and involves two-step non-ITS hashing, resulting in low key consumption. Later, it was shown that the proposed scheme is not secure [20,21]. More recently, Pan and co-workers [22,23] experimentally studied the performance of QKD when the authentication relied on a post-quantum cryptographic (PQC) protocol [24]. Although this hybrid protocol has efficient key management and new users can be easily added to the QKD network, it is not ITS, because the adopted PQC is computationally and not information-theoretically secure. In fact, the key that is distributed by means of the hybrid protocol is secure provided that the underlying PQC has not been broken during the execution of the protocol, which can be guaranteed only in the framework of computational security. Unless the adversary is present during the execution, they cannot later obtain the distributed key, even if they have unlimited resources.

In the present work, our aim is to propose a scheme for the generation, distribution, and management of pre-shared keys that relies on physical unclonable functions (PUFs). In contrast to post-quantum public-key cryptosystems, the judicious integration of PUFs in existing QKD systems promises ITS QKD under limited assumptions pertaining mainly to the performance of the PUF under consideration. Furthermore, our scheme allows for the real-time authentication of the QKD devices that are connected to the network.

PUFs were proposed by Pappu et al. [25] as a means to generate random numerical keys from the laser speckle that is produced from a multiple-scattering optical medium. The optical medium plays the role of a token, which is technologically hard to clone and it can produce a large number of independent, almost truly random keys, each one associated with different parameters of the input laser light (e.g., wavelength, power, wavefront, angle of incidence, etc.). Following the work of Pappu et al., PUFs have attracted considerable attention over the last two decades or so, and the related literature is rather rich [26–33]. Currently, there are different types of PUFs, including optical (non-silicon) PUFs, time-delay-based silicon PUFs, intrinsic silicon PUFs, magnetic PUFs, etc. Their main applications are in entity authentication, and in the development of anti-counterfeiting methods, and there is also a number of companies commercializing PUF products [27]. For

our purposes, it is sufficient to know the basic operation of a PUF, which is the generation of random keys, leaving aside details that go beyond the scope of the present work. An interested reader may refer to the related review articles [26–33], and the references therein.

In a nutshell, as shown in Figure 3, a PUF is a mathematical function that refers to the behaviour of a physical object or device (to be referred to hereafter as the PUF token or tag). The PUF tag is characterized by internal random disorder, which plays the role of a fingerprint, and it is technologically hard to clone. A PUF operates as a pseudo-random number generator (PRNG) in the sense that it produces a random numerical key as a response to an input stimulus (to be referred to hereafter as a challenge). The nature of the challenge depends on the PUF under consideration, while the response to a given challenge depends strongly on the internal disorder of the device. The raw noisy response is classically processed by means of a fuzzy extractor to yield a nearly perfect robust binary key. The fuzzy extractor typically involves two separate processes: reconciliation and hashing. The former aims at a reconciled key that is not affected by environmental variations and ageing of the token, whereas the latter compresses the reconciled key further, so that the final key is nearly uniformly distributed. A useful PUF is robust, easy to evaluate, difficult to replicate, unique (highly unlikely for two independent devices to return the same key for the same challenge), and unpredictable (very difficult or impossible to predict the response to a given challenge). Many of the PUFs that have been discussed in the literature satisfy these requirements. The main advantage of PUFs is that they remove the necessity of secret key storage, as the key is essentially physically stored in the disordered token and can be recovered on demand when the appropriate challenge is given. The only requirement is that the user has access to the token.

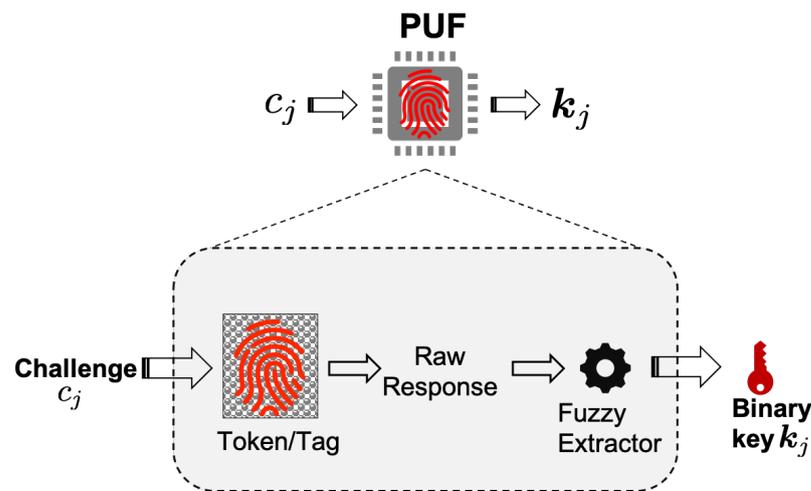


Figure 3. Schematic representation of a physical unclonable function (PUF). The token (sometimes also referred to as PUF tag), is a device with internal physical disorder. The internal disorder of the token is imprinted into its response to a physical challenge. The raw response is processed classically in order to yield a nearly perfect and robust random key.

The strength of a PUF is generally determined by the number of supported challenge–response pairs (CRPs), or to be more precise, by the way the number of potential CRPs scales with the increasing token size. In general, a PUF for which the number of supported CRPs grows exponentially with the size of the token is considered to be strong, while linear or polynomial growth typically refers to weak PUFs. Given that weak PUFs support a relatively small number of CRPs, an attacker can obtain the responses to all possible challenges, provided they have access to the PUF for a sufficiently long period of time. Hence, they can emulate a particular PUF in any cryptographic protocol without having an actual clone of the token. By contrast, strong PUFs support a much larger set of CRPs and an attacker must have access to the PUF for a time period that is considerably longer than the one required for the readout of a weak PUF. Ring-oscillator and SRAM PUFs are

considered to be weak, whereas optical and arbiter PUFs (a type of delay-based silicon PUF) are considered to be strong [26–33].

2. Results and Discussion

We discuss now how PUFs can be integrated in existing QKD systems, facilitating the generation and distribution of the necessary pre-shared key between Alice and Bob.

2.1. Integration of PUFs in Point-to-Point QKD Links

The manufacturer of the QKD devices associates a PUF with each of the QKD boxes (sender or receiver). For instance, a PUF tag is attached to each QKD box in a fashion similar to the barcodes or QR codes that accompany various products on the market. In contrast to these codes, however, as mentioned above, the PUF tags are technologically hard to copy, and they serve as unique fingerprints for the boxes they are attached to. Alternatively, we can assume that the PUF token associated with a QKD device is given to the owner of the device, e.g., in the form of a smart card [34].

We will first consider a point-to-point communication scenario, with a pair of QKD boxes intended for Alice (sender) and Bob (receiver). A pair PUF tokens/tags is associated with them. Each token is interrogated by a number of challenges $\{c_1, c_2, \dots\}$, and the corresponding keys $\{k_1^{(u)}, k_2^{(u)}, \dots\}$ are recorded, where $u = \{A, B\}$ is the label of the user. One of the honest users, say Alice, keeps a database of the challenge–response pairs (CRPs) in the following form [34,35]

$$\begin{array}{c|c}
 c_1 & k_1^{(A)} \oplus k_1^{(B)} \\
 c_2 & k_2^{(A)} \oplus k_2^{(B)} \\
 \vdots & \vdots \\
 c_j & k_j^{(A)} \oplus k_j^{(B)} \\
 \vdots & \vdots
 \end{array} \tag{1}$$

A schematic representation of the process is shown in Figure 4a. As will be discussed in detail below, the particular form of storage protects the individual keys against an adversary who obtains undetected access to the database.

A question arises here: who creates and who has access to this database? There are two options. One possibility is that the manufacturer creates the database and gives it to either of the two honest users during the delivery of the QKD boxes. Alternatively, one of the honest users (say Alice), has access to both boxes (and thus to both PUFs) simultaneously, and she creates the database before she gives one of the boxes to Bob. In both of these scenarios, only trusted parties are involved, in the sense that by definition Alice, Bob, and the manufacturer of the QKD devices are honest. In Figure 4, we assume that Alice possesses the database and hence there is no need for including an additional trusted third party. As will be discussed in the next subsection, however, the inclusion of a third party is inevitable in the case of larger QKD networks.

As discussed in the introduction, before Alice and Bob operate their QKD boxes for the first time, they must generate a common secret key, which will be used for post-processing purposes until a sufficiently large number of secret bits has been generated from the QKD sessions. To this end, Alice chooses at random one of the available challenges in the database, say the j th one, and sends it to Bob (see Figure 4b) over a classical channel which may not be authenticated (see related discussion in Section 2.4). The two users run their PUFs independently in order to recover their individual keys $k_j^{(A)}$ and $k_j^{(B)}$. Alice adds her key to the joint key $k_j^{(A)} \oplus k_j^{(B)}$ in order to recover Bob’s key which will seed the first QKD session. The corresponding entry is permanently deleted from the database, while Bob also keeps a blacklist of the used challenges so that they are not used again for the particular PUFs.

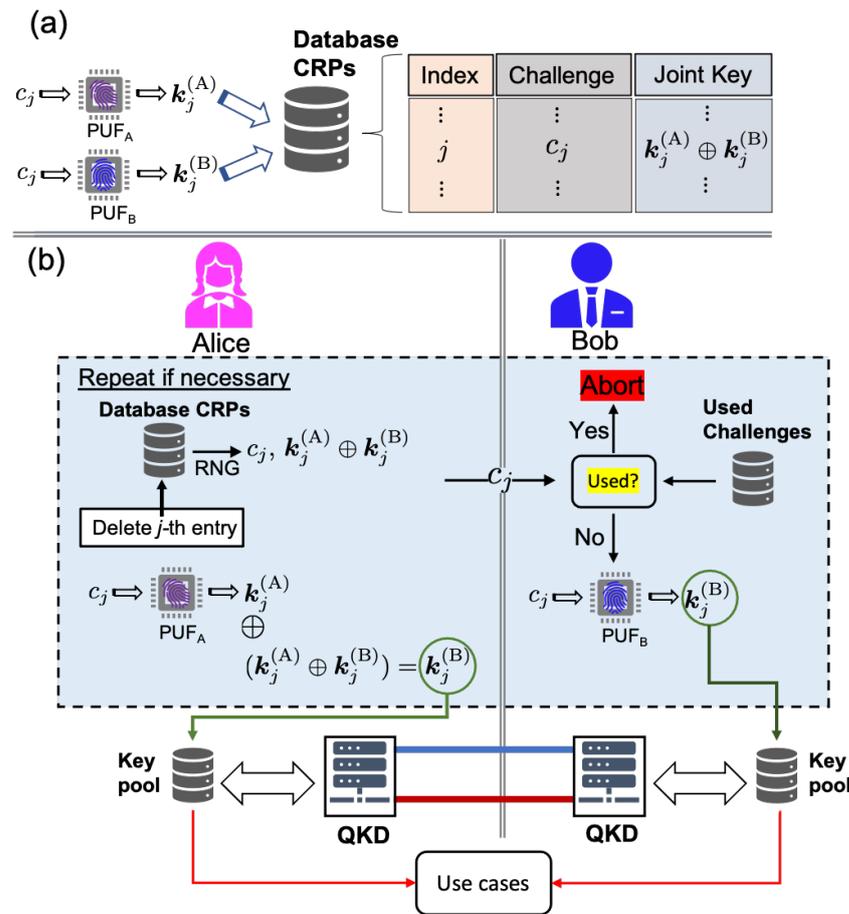


Figure 4. Integration of PUFs in a point-to-point QKD link. (a) Each pair of QKD boxes is associated with two PUFs namely, PUF_A and PUF_B. A PUF generates a random key as a response to a challenge. The manufacturer creates a database of challenge–response pairs (CRPs), where only the joint keys are stored. (b) With the purchase of the QKD boxes, the users also have access to the corresponding PUFs. Moreover, one of them (say Alice), receives a copy of the database. For the generation of a common random key, which will seed the first QKD session, Alice and Bob interrogate their PUFs independently with the same randomly chosen challenge. The corresponding entry is permanently removed from the database, while Bob also keeps track of the used challenges. This procedure can be performed again, e.g., if the first QKD session aborts, and a new QKD session is necessary.

Under normal conditions, the key material in the corresponding pools of Alice and Bob will grow through the operation of QKD sessions, and there will be no need for the users to execute the above procedure again. However, if for any reason, at any time, QKD alone cannot support the needs for keys and additional key material is needed, the procedure can be repeated again.

2.2. Large QKD networks

The previous subsection focuses on a typical point-to-point QKD scenario, where only two honest parties are involved and a single pre-shared key is required for the seeding of the first QKD session. The design of large full-mesh QKD networks with n users, however, requires $n(n - 1)/2$ pre-shared keys (see Figure 5a). For large n , there is a large amount of key material that has to be distributed in a secure manner among the users in order for the first pairwise QKD sessions to be implemented. Moreover, it is very hard to add a new user to the network. Indeed, as soon as a new user (let us say Charlie) receives his QKD device, this should come with a list of keys pertaining to each one of the other users with whom Charlie may want to establish a QKD link. Each one of these users

has to be informed about Charlie joining the network, and he/she should receive a copy of the corresponding symmetric key to be used for the implementation of the first QKD session with him. Overall, the addition of a new user in a network with n users should be accompanied by the generation and the distribution of n additional symmetric secret keys, a rather tedious task.

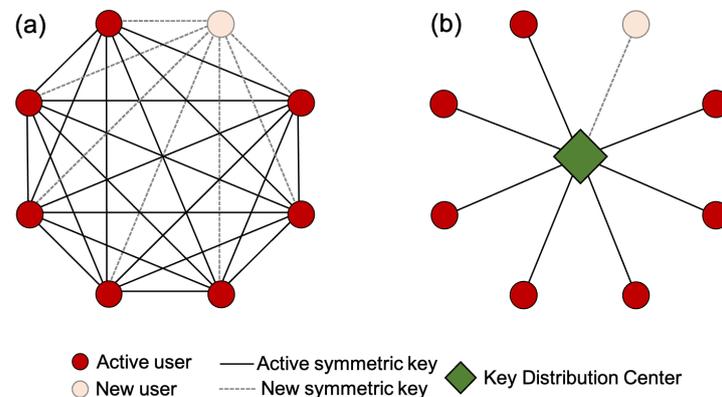


Figure 5. Full-mesh QKD network involving n users. (a) In the absence of a key distribution centre (KDC), the total number of pre-shared keys is $n(n - 1)/2$, while each new user has to share n keys with each one of the other existing users. (b) In the presence of a KDC, each user shares a key with the KDC only.

One solution to this problem is the use of a trusted centre, which shares a secret key with each one of the n users, thereby significantly reducing the number of pairwise shared secret keys (see Figure 5b). In this case, it is relatively easy for Charlie to join the existing network, as he has to obtain, together with his QKD device, only one secret key which is known to the key-distribution centre (KDC). The establishment of a QKD link with any other user can be achieved through the KDC, which can help the users to agree on a common secret key required for the post-processing in the first QKD session. The second solution to the aforementioned problems relies on the use of classical or quantum asymmetric cryptosystems, where each user holds a pair of keys (a private key and a public key). Standard widely used asymmetric cryptosystems rely on the difficulty of certain mathematical problems, and thus they offer computational security only [12–15,22,24,36,37]. On the other hand, a small number of ITS quantum public-key cryptosystems have been proposed in the literature, but their implementation is beyond reach of current technology [38–40].

As long as we are interested in preserving the ITS character of QKD protocols, the only currently available route towards the realization of large QKD networks involves, in one way or another, the use of a trusted authority, such as a KDC, which is trusted by all the parties participating in the network. This solution is by no means optimal, because the functionality and the security of the entire QKD network depends strongly on the KDC, which automatically becomes the main target of potential adversaries. Given the current status of quantum technologies, however, the presence of a KDC in large networks facilitates and guarantees the generation and the distribution of pre-shared keys between users, which are essential for the establishment of the first secure pairwise QKD sessions. In the literature of QKD, the inclusion of a KDC is also considered essential for one more reason, namely, it can operate as a relay, thereby allowing two users that are separated by a distance well beyond the reach of current QKD systems to establish a secret key [6].

The ideas discussed in the previous subsection can be generalized in the context of large QKD networks if we assume that the KDC holds all the databases of CRPs, as shown in Figure 6. The manufacturer assigns a PUF tag to each QKD device (sender or receiver), and a separate database of CRPs is created along the lines discussed in the previous subsection. For the sake of simplicity, let us assume that the manufacturer of the QKD devices also controls the KDC, while the manager of the KDC has one PUF for each database, i.e., for

each user. So, the KDC manages as many databases and PUFs as the user devices in the QKD network. For the sake of concreteness, let us consider the database for the QKD device C, intended for Charlie, who will join the QKD network. Let PUF_{MC} denote the PUF used by the KDC for the encryption of the entries in Charlie’s database of the CRPs. The j th entry in the database involves the challenge c_j and the joint key $k_j^{(MC)} \oplus k_j^{(C)}$, where $k_j^{(MC)}$ and $k_j^{(C)}$ denote the keys associated with the responses of PUF_{MC} and PUF_C to challenge c_j , respectively. As soon as Charlie is ready to join the network, his first task is to establish a common secret key with the KDC, following the steps discussed in the previous subsection. The procedure is recapitulated in Figure 6b, with the necessary changes in the labels.

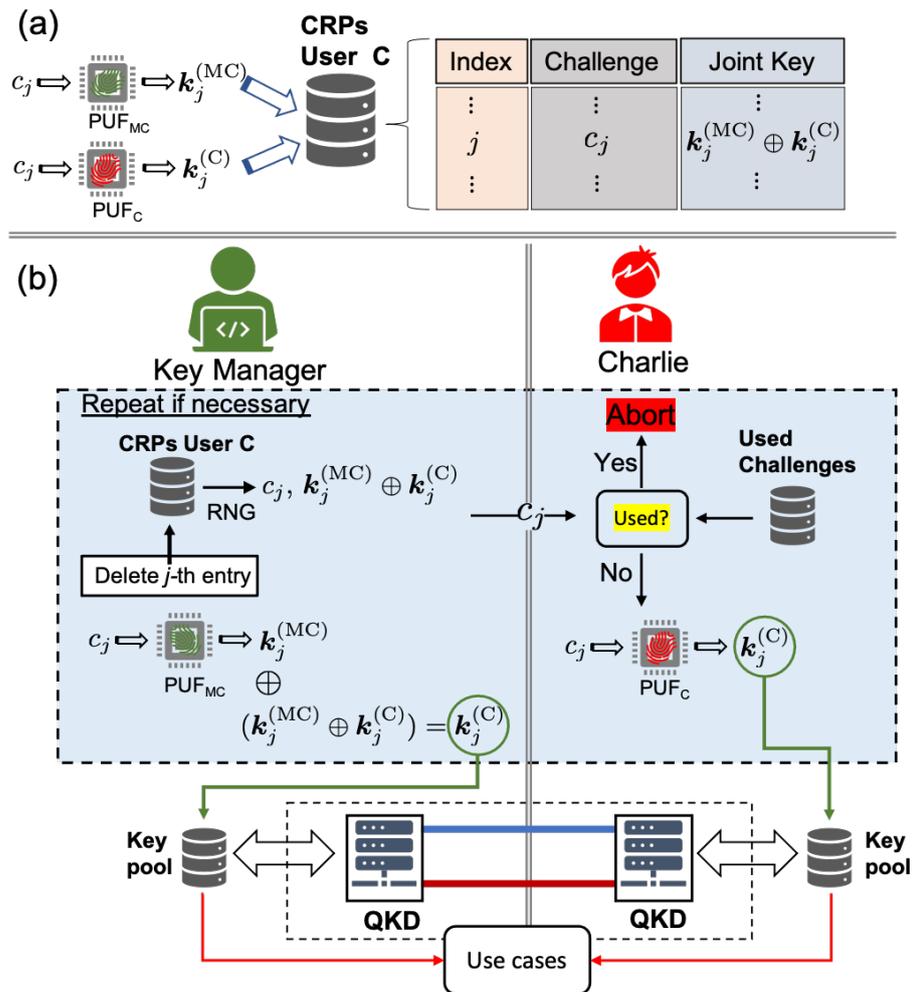


Figure 6. Integration of PUFs in a large QKD network. (a) Each QKD device (sender or receiver) is associated with a PUF. The PUF generates a random key as a response to a challenge. The KDC has its own PUFs, and it is controlled by the manufacturer, who creates a database of challenge–response pairs for each QKD device. The KDC has access to the databases of CRPs for all of the QKD devices that have been or will be connected to it, while a different PUF is used for the encryption of the entries in each database. (b) Each time that fresh key material is needed for a user, the user can generate a common random key with the KDC by running the protocol in the shaded box. Note that the presence of the QKD link between the KDC and the user is not necessary, as keys can be generated from the PUFs. In practice, the addition of such a QKD link will limit the distance between the user and the KDC, but it can make the network self sustainable in terms of key generation and consumption.

Having established a common secret key with the KDC, Charlie can establish a common secret key with any other existing user in the network, which will allow them to run the post-processing in the first QKD sessions until a sufficiently large number of fresh secret random bits has been generated through the QKD. The procedure is outlined in Figure 7 and it is as follows. The key manager of the KDC shares a common secret key $\tilde{k}_{M,C}$ with Charlie and another key with Alice, a user who is already in the network and with whom Charlie wants to communicate. For the sake of clarity, we write the latter key as a concatenation of two smaller keys, i.e., $\tilde{k}_{M,A}||\kappa_{M,A}$. Charlie contacts the KDC and requests connection with Alice, who is already in the network. The key manager calculates $\tilde{k}_{M,C} \oplus \tilde{k}_{M,A}$ and sends it to Alice over a classical channel. The message is authenticated with the key $\kappa_{M,A}$. Upon receipt of the message, Alice confirms its origin, and she adds her key to the received message to obtain $\tilde{k}_{M,A} \oplus (\tilde{k}_{M,C} \oplus \tilde{k}_{M,A}) = \tilde{k}_{M,C}$. Hence, Alice and Charlie share a common secret key, which can be expanded through QKD.

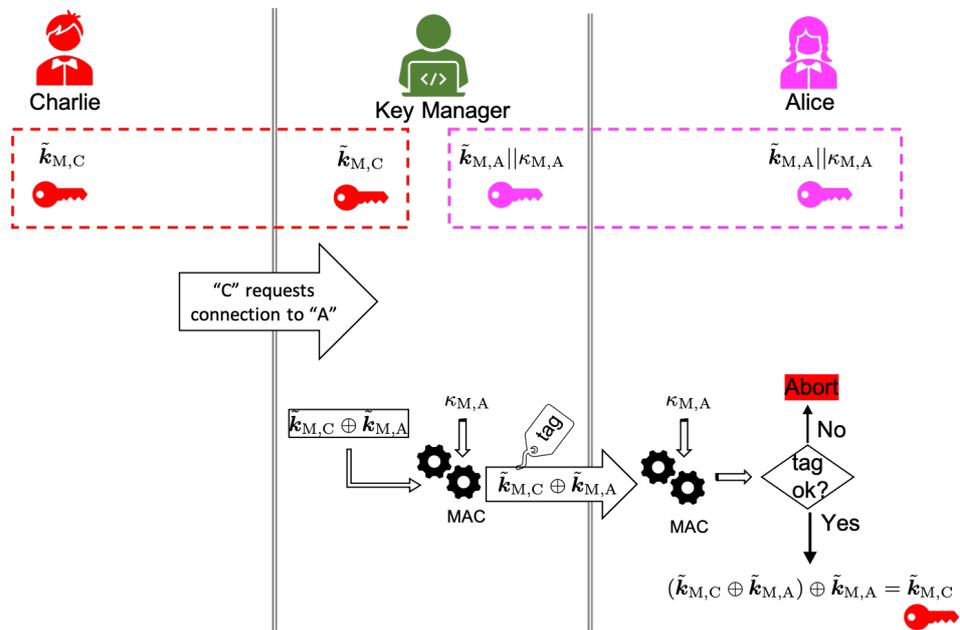


Figure 7. Schematic representation of the procedure through which Alice and Charlie can establish a common secret key, with a third party acting as a relay.

It is worth emphasizing here that the realization of the ideas we have just presented does not require a QKD link between each user and the KDC. As shown in Figure 6, the manager and the user can agree on a common secret key by means of classical communication and local operations on their PUFs. The only requirement is for the manager to communicate a randomly chosen challenge to Charlie over a public channel. This allows a lot of flexibility with respect to the physical topology of the network, in the sense that there are no limitations associated with the transmission of quantum signals on the spatial separation between a user and the KDC. In the absence of QKD links between the KDC and the users, all the key material is provided only by the PUFs, and thus it is limited by the number of challenge–response pairs that can be supported by the PUFs under consideration. In this context, strong PUFs are preferable. However, even though a QKD link between Charlie and the KDC imposes limitations on the distance, at the same time it allows for the expansion of an initial key that has been obtained through PUFs. In this case, the network may become self-sustainable, with respect to the generation and the consumption of key material, and its operation is not expected to depend strongly on the type of used PUFs.

2.3. QKD Device Authentication

Message authentication is what prevents a man-in-the-middle attack against QKD systems, and the ideas discussed in the previous subsections facilitate the generation and the distribution of pre-shared secret keys to this end. Entity authentication (also known as identification) is a different and very important cryptographic task, which allows the identity of a user or device to be confirmed [12–15]. By means of entity authentication, one can control which users or devices have access to certain physical or virtual resources. Entity authentication should not be confused with message authentication. They are different cryptographic tasks with distinct goals, and one cannot replace the other. A fundamental difference between the two is that an identification scheme provides real-time evidence about the identity of a user or a device, whereas MACs (and signature schemes) allow data origin authentication, which can be performed any time after the relevant message has been tagged or signed.

The integration of PUFs in QKD networks allows the KDC to confirm the identity of the QKD devices that are connected to the network at any time, thereby preventing the connection of counterfeit or unauthorized devices through which an attacker may try to connect to the KDC or to other users in the network in order to access databases or to sabotage the operation of the network. Moreover, an honest user can confirm the authenticity of a purchased QKD device. A PUF tag that is attached to a QKD device serves as a unique fingerprint and protects against counterfeiting.

Let us consider a scenario where a new user, Charlie, joins the network, and that the KDC wants to authenticate his QKD device. The procedure is outlined in Figure 8. As in the previous subsections, we will assume that Charlie's QKD device is associated with PUF_C . The related database of CRPs is generated by the manufacturer, and it is available to the KDC. Moreover, we will assume that each user (or QKD device) is uniquely identified by a binary string (a serial number). Given that Charlie is a new user who does not share a secret key with the KDC, the key manager randomly chooses one of the available challenges from the database of CRPs associated with PUF_C . Moreover, the manager generates a random bit string s , and they send it to Charlie together with a challenge chosen at random c_j . Charlie obtains the response k_j^C of PUF_C to challenge c_j , whereas the manager recovers the same response after they add the response of PUF_{MC} to the same challenge to the joint key. Charlie passes the concatenation of his identification string ID_C with s through a MAC with key k_j^C , and the resulting tag, $\tau = h_{k_j^C}(ID_C||s)$, is sent to the KDC. The key manager computes the tag for the same message and they accept the QKD device as authentic only if this agrees with the tag received from Charlie.

In the protocol outlined in Figure 8, we have assumed that the user does not share a common secret key with the KDC and it has to be generated locally through their PUFs. If the user shares a secret key with the KDC, then the procedure in the shaded region of Figure 8 can be omitted, and the entity authentication can rely on the existing key material. Moreover, the protocol can be easily extended to mutual entity authentication by adding a second round of authentication, where the user chooses at random another binary string, which is sent to the KDC [12,14].

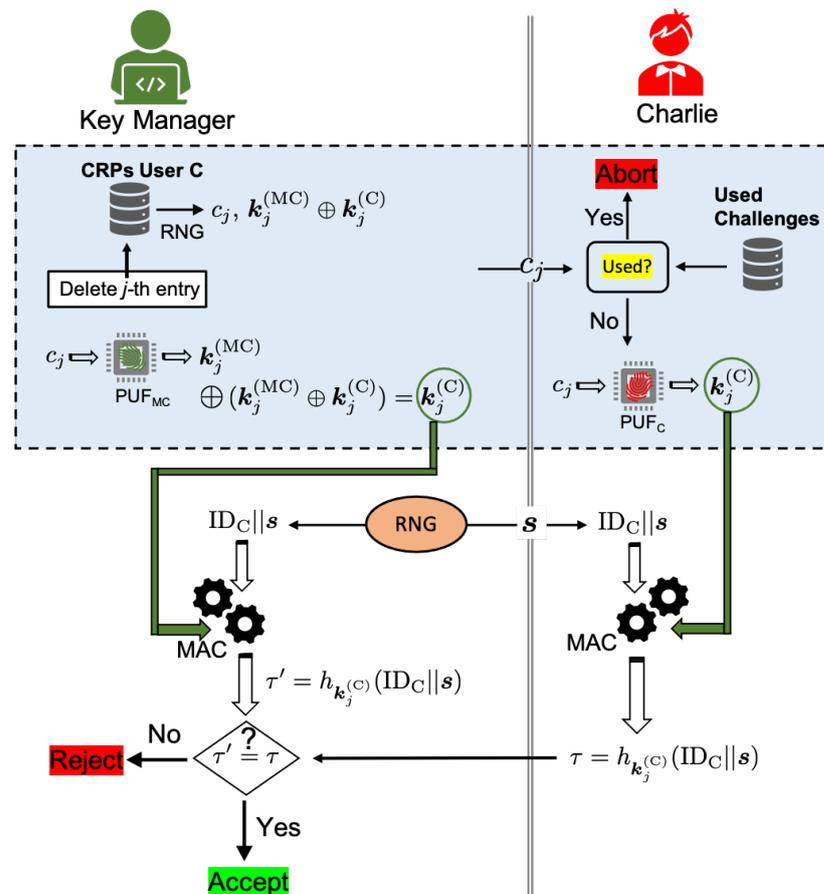


Figure 8. Schematic representation of an entity authentication session. In order for the KDC to confirm the identity of the newly connected QKD device of Charlie, they run the protocol in the shaded region to obtain a common secret key through their PUFs. The key manager also chooses a random binary string s , and they send it to the user. The user runs a publicly known MAC to obtain a tag τ for the concatenated message involving their unique identity (in binary format) ID_C and the received random binary string. The tag is sent to the KDC, where it is compared to the tag produced locally by the key manager, and the identity of the QKD device is accepted only if the two tags agree. If a secret key is already shared between the user and the KDC, then the steps in the shaded box can be omitted.

2.4. Security Considerations

The Wegman–Carter authentication (WCA) scheme [17] that is usually employed in standard QKD systems is ITS only if the pre-shared secret key is truly random, or at least very close to perfect. In particular, it has been shown that a WCA scheme that is executed with an ϵ' -perfect key, is indistinguishable from the ideal scenario, with a probability of $(1 - \epsilon - \epsilon')$, where ϵ refers to the security of the ideal scheme [16,41]. The protocols outlined in Figures 4, 6, and 7, can provide two honest users with the secret key that is necessary for ITS post-processing in the first QKD session if the following conditions are satisfied:

- (C1) The numerical keys produced by the PUFs under consideration are close to truly random.
- (C2) The legitimate users never make their PUF tokens/tags available to other parties.
- (C3) Each entry in the database of challenge–response pairs is used only once.
- (C4) The MAC that is used for the distribution of the key in Figure 7 is ITS.

For various PUFs that have been discussed in the literature, the generated keys have been shown to successfully pass widely accepted tests of random sequence certification, such as the ones provided by the NIST suite [25,31,35,42]. For all practical purposes, such a

key can be considered to be close to truly random, and there are no correlations between different elements or parts of the key. Moreover, keys that have been generated from different PUFs or from the same PUF but for different challenges can be considered as uniformly distributed independent random strings. As a result, the entries in a database of CRPs are ITS, because the extraction of the individual keys $k_j^{(u)}$ and $k_j^{(v)}$ from the joint key $k_j^{(u)} \oplus k_j^{(v)}$ is impossible by virtue of a one-time-pad (OTP) encryption, unless the attacker has also access to at least one of the PUFs. Furthermore, an attacker cannot launch a machine learning attack [27] in order to create a model of either of the two PUFs. This is because machine learning requires direct access to a set of CRPs $\{(c_j, k_j^{(u)}) : j = 1, 2, \dots\}$, whereas in our case, the individual keys are not accessible and they are stored encrypted in a secure manner.

For the sake of simplicity, in Figures 4 and 6 we have assumed that one challenge suffices for the seeding of the first QKD session. This may not be always the case. A typical QKD system requires at least a 256-bit key (e.g., see Appendix A), while the length of the key that can be generated from a PUF for a single challenge ranges from a few hundred to a few thousand bits. The precise length depends on the type of the PUF under consideration as well as on the details of the fuzzy extractor. If a single challenge cannot provide a key of an appropriate length for seeding the first QKD session, then one can concatenate two or more keys pertaining to different challenges. Indeed, as long as different challenges yield independent nearly perfect keys, the resulting longer key is also close to truly random.

Let us turn now to the entity authentication protocol in Figure 8. One can readily show that it is information-theoretically secure if an l -time ϵ -secure MAC is used [12,14,43] with $l \ll 2^{|s|}$ (see also appendix). In particular, the probability that the KDC uses the same random string s in the i th session is $(i-1)/2^{|s|}$. Hence, the probability for a repeat in any of the l sessions is $l(l-1)/2^{|s|+1}$. The same result can be obtained through the birthday bound for $l \ll 2^{|s|}$. On the other hand, there is a probability of at most ϵ for an adversary to construct the correct tag. Hence, the overall probability for an adversary to deceive the KDC is at most $l(l-1)/2^{|s|+1} + \epsilon$. Of course, one should not forget here that unconditionally secure l -time ϵ -secure MACs require a uniform truly random key. Hence, from the previous discussion, the quality of the keys that are produced from PUFs and the security of the databases also affect the security of the entity authentication protocol in Figure 8.

Another issue that deserves our attention is the communication of the randomly chosen challenge c_j in the procedures outlined in Figures 4b, 6b, and 8. The challenge is sent in plain text over a public, possibly unauthenticated channel. An adversary who learns the challenge and does not have access to the PUF of the sender or the receiver cannot use it to their advantage, because, as previously discussed, the PUFs operate as PRNGs, and their output to a given challenge cannot be predicted with a probability much better than random guessing. Even if the adversary has access to the database of CRPs, they cannot deduce the individual keys from the joint key by virtue of the OTP encryption. Finally, given that the entry pertaining to the communicated challenge is deleted from the database of CRPs and is added to a blacklist, an adversary cannot repeat it so as to fool either of the two users. The adversary can launch a type of DoS attack by changing the challenge. In this case, the two users will essentially not share the same key, as required for successful realization of a QKD session, and it is almost certain that the different keys will make the QKD protocol abort, and the users will have to start all over again. We have a similar situation in Figure 7, where Charlie's request for communication with Alice is also sent in plain text, and its origin is not verified by the receiver. In principle, an attacker can send multiple such requests to the KDC, thereby enforcing the manager to consume keys without reason and depleting the relevant databases and key pools. In all of these scenarios, the attacks are possible because the classical communication is not authenticated. However, none of these attacks threaten the security of the QKD network; rather, they target the resources of the QKD network. For this reason, there is no need

for ITS message authentication, and they can be prevented by means of computationally secure MACs [12–15,44] or public key cryptosystems [12,15,36,37].

Finally, throughout this work we have assumed that the manager of the KDC uses a different PUF for each user participating in the network, namely, PUF_{MA} , PUF_{MB} , PUF_{MC} , etc. In this way, we have added an additional level of security in the sense that if, for any reason, a database of CRPs is compromised, the security of the other databases is not affected, as long as different PUFs produce independent random keys. In particular, the j th entry in the databases of CRPs for two different users, say A and B, pertains to the joint keys $K_1 := k_j^{(\text{MA})} \oplus k_j^{(\text{A})}$ and $K_2 := k_j^{(\text{MB})} \oplus k_j^{(\text{B})}$, where all of the individual keys refer to the same challenge but to different independent PUFs, namely, PUF_{MA} , PUF_{A} , PUF_{MB} and PUF_{B} . By virtue of the OTP encryption, even if user A is malicious, they cannot deduce either $k_j^{(\text{MB})}$ or $k_j^{(\text{B})}$ from K_1 and PUF_{A} , provided that conditions (C1) and (C2) are satisfied for the PUFs under consideration.

Perhaps one way to avoid the use of different PUFs for each user is to consider the use of PUF duplicates in the spirit of the recent work by Marakis et al. [45]. In this preprint, the authors demonstrate for a particular type of PUF, that the original PUF manufacturer (which is a trusted authority in our case), can fabricate multiple identical structures (tokens) that possess essentially the same optical scattering behaviour (i.e., the same challenge–response pairs), but at the same time, they remain unclonable for external adversaries who do not know their internal features and inner construction plan. The use of PUF duplicates deserves a thorough analysis, which goes beyond the scope of the present work because it may open up new security loopholes which are not present at the moment.

3. Conclusions

We have considered the distribution of a common secret key between two honest users of a QKD network, which is necessary for authentication and encryption purposes during the post-processing stage in the first pairwise QKD session. Such a key (usually referred to as a pre-shared key in the QKD literature), is necessary for any QKD system because it prevents a man-in-the-middle attack.

We have discussed the generation and the distribution of pre-shared keys by means of PUFs. Each QKD device (sender or receiver) is associated with a disordered token/tag, which plays the role of a unique fingerprint and allows the users to generate nearly perfect random keys as well as to authenticate their devices. Two users can generate a common secret random key by locally interrogating their tokens with the same randomly chosen challenge. The main advantages offered by the proposed scheme are the following: (i) It does not require quantum communication. (ii) In contrast to PQC protocols, the distributed pre-shared keys are information-theoretically secure, which ensures the ITS character of the entire QKD session (see also related discussion in Section 1). (iii) There is no need for distributing additional secret keys or passwords to Alice and Bob to ensure the secrecy of the pre-shared key. (iv) There is no need to keep the database of CRPs private. Even if an adversary obtains access to the database, they cannot deduce the possible pre-shared keys, as discussed in Section 3. (v) By definition, a PUF is hard to clone, even if one has access to it, because its response to a particular challenge is very sensitive to the internal randomness of the token (tag), which also depends on the conditions of fabrication. (vi) There is no need to ensure the authenticity of the randomly chosen challenge that is used for the establishment of the pre-shared key, unless one wishes to prevent a DoS attack (see the related discussion in Section 3). All of these features make our scheme more attractive than other currently used methods, such as the inclusion of the pre-shared key in the software that accompanies the QKD boxes, its storage in password-protected devices, the use of a trusted courier, or the distribution by means of PQC.

For two users, a single CRP suffices for establishing the necessary pre-shared key for the first QKD session, provided that the length of the key that can be generated from the PUF under consideration is at least equal to the length of the pre-shared key required by the used QKD system (see also the related discussion in Section 3). In other words, by

employing the right type of PUF, the present protocol does not require more pre-shared keys (or equivalently challenges) than the number required by the current QKD systems. Of course, having more CRPs available in the database, in case something goes wrong, is an additional feature which is provided by our protocol and, to the best of our knowledge, it is not an option in the QKD systems that are available on the market today. Extending this reasoning to a case of n users with a trusted KDC, one can easily see that the number of challenges (entries) required by the present protocol are as many as the number of pre-shared keys required in the QKD network without PUFs. The additional entries that are provided by a PUF is a benefit that can be exploited by the users if needed.

As discussed above, the inclusion of a trusted KDC is essential in large QKD networks (many users) in order to preserve the ITS character of QKD systems. Moreover, even for two users, a trusted intermediate node may operate as a relay, which allows the distribution of a secret key beyond the distances that can be covered by a direct QKD link. In other words, the trusted KDC is not a requirement that originates from our protocol, but rather it is a necessity which originates from the point-to-point character of QKD links, the limitations of the available QKD systems, and the absence of quantum repeaters. If one is willing to accept a computationally secure system, PUFs can be combined with standard public key cryptosystems in order to generate and distribute keys between users. In particular, the key that is generated from a PUF device may serve as a private key, or it can be used as a random seed for the generation of a private key. Subsequently, building upon the private key, the corresponding public key is generated, which becomes publicly available through a trusted public key server. An advantage of such an asymmetric cryptosystem is that it allows for the implementation of various cryptographic tasks beyond the generation and the distribution of a key, including digital signatures. At the same time, the pair of private–public keys is bound to the PUF device.

Throughout this work, we have adopted a rather general theoretical framework, which is not restricted to a particular type of PUF. The present results pave the way for the integration of PUFs in currently used QKD systems, but there are many practical issues that deserve further investigation in order to identify which type of PUF better serves the needs of QKD networks. To this end, one has to take into account various facts including the required key lengths, the size of the network, the robustness of the PUF in the presence of environmental fluctuations, etc. Optical PUFs may be a very promising candidate because they are considered to be among the strong PUFs, and they can typically support long keys (thousand of bits) [25,35,42]. Moreover, they are compatible with QKD infrastructure and, in principle, they are amenable to remote quantum readout [46]. Remote quantum readout of optical PUFs is very attractive if one is interested in saving classical resources, at the cost of introducing limitations in the spatial separation between the users and the KDC. However, all the known quantum readout schemes [47–49] are currently limited to very short distances (of the order of 1 km), and there is a need for their extension to distances comparable to the ones that can be covered by standard QKD systems. Only experimental research may shed light on the integration of PUFs in operational QKD networks and provide answers to many of these questions.

Author Contributions: Conceptualization, G.M.N.; methodology, G.M.N. and M.F.; formal analysis, G.M.N. and M.F.; writing—original draft preparation, G.M.N.; writing—review and editing, G.M.N. and M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded by the European Union under the Digital Europe Program grant agreement number 101091504. Moreover, it has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—SFB 1119—236615297.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors are grateful to G. Alber for useful discussions.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QKD	Quantum key distribution
PUF	Physical unclonable function
KDC	Key distribution center
MAC	Message authentication code
CRP	Challenge-response pair
ITS	Information-theoretically secure
OTP	One-time pad
WCA	Wegman-Carter authentication
DoS	Denial of Service

Appendix A. The Size of the Pre-Shared Key

In a QKD system, the pre-shared key is mainly used for authentication and encryption purposes during the post processing. The length of the pre-shared key depends strongly on the details of the post-processing, and one may wonder about the typical values. In an effort to discuss the typical lengths without deviating from the main focus of this work, we have kept this appendix as general and simple as possible. Following Refs. [10,11], we will assume that authentication takes place only in the key sifting, the error verification, and the privacy amplification for the extraction of the final secret key, whereas encryption takes place only during error reconciliation. Let us look separately at the key cost for these processes.

Appendix A.1. Error Reconciliation

For the sake of simplicity, let us consider a BB84-type protocol where the two bases are equally likely. We will also assume that the error rate in the sifted key will be the same for both bases, and it will be denoted by q_e . Let n_s denote the length of the sifted key to be reconciled. According to Shannon's theory, Alice and Bob have to exchange at least $n_{er} = n_s H(q_e)$ bits for error reconciliation, where $H(q_e)$ is the binary entropy. To prevent the leakage of information to an adversary, the data can be encrypted by means of an OTP encryption using the pre-shared key of length $k_{er} = n_s H(q_e)$ bits [10,11]. Assuming $n_s = 10^4$ and $q_e = 1\%$, we have $k_{er} \sim 800$ bits, while for $q_e = 2\%$, we have $k_{er} \sim 1400$ bits. In a protocol where this information is not encrypted, one has to pay the cost in the privacy amplification by shortening further the final key.

Appendix A.2. Message Authentication

Data integrity or message authentication is achieved by means of the so-called message authentication codes (MACs), and one can distinguish between ITS (or else unconditionally secure) and computationally secure MACs [12–16]. MACs that rely on block ciphers or collision-resistant hash functions offer computational security [12,44], as opposed to the Wegman and Carter scheme, which exploits ϵ -Almost Strongly Universal₂ (ϵ -ASU₂) hash functions and offers unconditional security [14–17,43]. Given that QKD protocols promise the highest level of security, the Wegman-Carter authentication (WCA) scheme has been widely employed in the post-processing of data [9].

In the WCA, Alice and Bob share a common secret key k uniformly distributed over \mathbb{K} and let $\mathbb{H} := \{h_k : k \in \mathbb{K}\}$ denote a publicly known set of ϵ -ASU₂ hash functions. The shared key identifies uniquely a hash function, which is used by the two users for hashing messages $m \in \mathbb{M}$. In particular, when Alice wants to send an authenticated message m to Bob, she evaluates the tag τ , through $\tau := h_k(m)$. The message and the tag (m, τ) are sent to Bob in plain text over a classical channel. In general, as a result of forgery or noise, Bob receives (m', τ') and he accepts the message only if $\tau' = h_k(m')$.

In the framework of a man-in-the-middle attack, Eve intercepts the valid message–tag pair transmitted from Alice to Bob, and her task is to produce another message–tag pair that will be accepted by Bob as originated from Alice. This is a *substitution* or chosen-message attack, and the probability of Eve succeeding is quantified by the deception probability $P_1^{(D)}$. Of course, Eve may always try to impersonate Alice without having access to a valid message–tag pair, but in this case, the associated deception probability $P_0^{(D)}$ is at most as large as $P_1^{(D)}$. In general, an l -time ϵ -secure MAC is defined as follows.

Definition A1. An MAC is l -time ϵ -secure, if for all (including unbounded) adversaries, $P_l^{(D)} \leq \epsilon$.

The WCA scheme described above is one-time ϵ -secure, with $\epsilon \geq 1/|\mathbb{T}|$, where \mathbb{T} denotes the set of all possible tags. Various ϵ -ASU₂ hash functions have been proposed in the literature (e.g., see references in [14–16]), which achieve

$$P_0^{(D)} = \frac{1}{|\mathbb{T}|}, \quad P_1^{(D)} = \epsilon. \tag{A1}$$

The precise value of the security parameter ϵ and the length of the key vary from scheme to scheme. One can readily show the following fundamental theorem [15,17,43]

Theorem A1. A 1-time ϵ -secure MAC must have keys of length at least $2|\log_2(\epsilon)|$.

To understand this theorem, one may consider the scenario discussed above, where Eve obtains a valid message–tag pair. We ask for the probability of Eve guessing the correct tag for any message to be at most ϵ , even after she has seen the valid message–tag pair. This implies that the tags in total should have at least $2|\log_2(\epsilon)|$ bits of entropy. This entropy, however, can be obtained only from the key because the message is known. The theorem can be extended to any $l \geq 1$.

Wegman and Carter have shown that one can have a one-time $2/|\mathbb{T}|$ -secure MAC with a key length that scales linearly with the length of the tag and logarithmically with the length of the message [16,17], i.e.,

$$|k| \simeq 4\log_2(|\mathbb{T}|)\log_2[\log_2(\mathbb{M})]. \tag{A2}$$

In the framework of QKD, we usually deal with binary messages, key and tags, and in this case, this expression reads

$$|k| \simeq 4|\tau|\log_2(|m|), \tag{A3}$$

where $|\tau|$ and $|m|$ denote the lengths of the tag and the message, respectively. Accordingly, the security of the WCA scheme is

$$\epsilon = 2^{-|\tau|+1} \tag{A4}$$

A family of ϵ -ASU₂ hash functions can be also constructed efficiently by means of Toeplitz matrices [18]. In this case, the tag τ is obtained by multiplication of a $|\tau| \times |m|$ Toeplitz matrix with the message m . The elements of the matrix are either 0 or 1, whereas the matrix is uniquely identified by a binary string of length $|\tau| + |m| - 1$ which determines the elements of the first column and the first row. In the authentication scheme discussed by Krawczyk, the tag is also encrypted by means of an OTP encryption with $|\tau|$ -bit random key. This authentication scheme is 1-time ϵ -secure, with [16,18]

$$\epsilon = \frac{2|m|}{2^{|\tau|}} + \frac{1}{2^{|\tau|}}. \tag{A5}$$

for key length $|k| > 3|\tau|$ or else $|k| > 3\log(\varepsilon)$.

As mentioned above, throughout this work we assume that channel authentication takes place in the key sifting, the error verification, and the privacy amplification [11]. Let $\varepsilon^{(ks)}$, $\varepsilon^{(ev)}$, and $\varepsilon^{(pa)}$ denote the corresponding deception probabilities. In view of the union bound, the overall probability of undetected failure (failure of security) for the QKD protocol is bounded from above as follows

$$P_{\text{fail}} \leq P_{\text{fail}}^{(\text{auth})} + P_{\text{fail}}^{(\text{other})}, \quad (\text{A6})$$

where the first term refers to the probability of failure due to the authentication in these phases of the post-processing, and the second term refers to the failure due to other reasons. The former probability is related to the aforementioned deception probabilities for the phases involving authentication. The deception probabilities must be as low as possible so that the overall probability for the QKD to undergo an undetected failure is very small. For instance, asking for the right-hand side of the inequality to be equal to 10^{-9} so that $P_{\text{fail}} \leq 10^{-9}$, we need $\varepsilon^{(ks)}$, $\varepsilon^{(ev)}$, $\varepsilon^{(pa)} < 10^{-9} \simeq 2^{-30}$. Recalling Theorem 1 for $\varepsilon = 2^{-30}$, we have that the required key length for each authentication session satisfies $|k^{(j)}| > 60$ bits. Under these conditions, the total length of the pre-shared key required for ITS authentication in QKD should satisfy

$$|k_{\text{auth}}| \geq 2|k^{(ks)}| + |k^{(ev)}| + |k^{(pa)}| > 240 \text{ bits}, \quad (\text{A7})$$

where we have taken into account the fact that the service channel has to be authenticated both ways in the case of key sifting [11]. This is a low bound on the required key length, which is independent of the applied ITS MAC, as Theorem 1 is very general and depends only on the value of ε . More tight bounds can be obtained by considering specific authentication schemes, such as the ones mentioned above. Taking into account the typical key lengths required for error reconciliation, the total length of the pre-shared key in a typical QKD protocol may increase considerably. Unconditionally secure l -time ε -secure MACs for $l \gg 1$ typically require very long keys. MACs that rely on block ciphers or collision resistant hash functions may require considerably smaller keys, but they offer computational security only [44], thereby affecting the security of the entire QKD protocol.

References

- Gisin, N.; Ribordy, G.; Tittel, W.; Zbinden, H. Quantum cryptography. *Rev. Mod. Phys.* **2002**, *74*, 145–195. [[CrossRef](#)]
- Scarani, V.; Bechmann-Pasquinucci, H.; Cerf, N.J.; Dušek, M.; Lütkenhaus, N.; Peev, M. The security of practical quantum key distribution. *Rev. Mod. Phys.* **2009**, *81*, 1301–1350. [[CrossRef](#)]
- Lo, H.-K.; Curty, M.; Tamaki, K. Secure quantum key distribution. *Nat. Photon.* **2014**, *8*, 595–604. [[CrossRef](#)]
- Diamanti, E.; Lo, H.K.; Qi, B.; Yuan, Z. Practical challenges in quantum key distribution. *npj Quantum Inf.* **2016**, *2*, 16025. [[CrossRef](#)]
- Pirandola, S.; Andersen, U.L.; Banchi, L.; Berta, M.; Bunandar, D.; Colbeck, R.; Englund, D.; Gehring, T.; Lupo, C.; Ottaviani, C.; et al. Advances in quantum cryptography. *Adv. Opt. Photonics* **2020**, *12*, 1012–1236. [[CrossRef](#)]
- Mehic, M.; Niemiec, M.; Rass, S.; Ma, J.; Peev, M.; Aguado, A.; Martin, V.; Schauer, S.; Poppe, A.; Pacher, C.; et al. Quantum Key Distribution: A Networking Perspective. *ACM Comput. Surv.* **2020**, *53*, 1–41. [[CrossRef](#)]
- Xu, F.; Ma, X.; Zhang, Q.; Lo, H.-K.; Pan, J.-W. Quantum cryptography with realistic devices. *Rev. Mod. Phys.* **2020**, *92*, 025002. [[CrossRef](#)]
- Paraíso, T.K.; Woodward, R.I.; Marangon, D.G.; Lovic, V.; Yuan, Z.; Shields, A.J. Advanced Laser Technology for Quantum Communications. *Adv. Quantum Technol.* **2021**, *4*, 2100062. [[CrossRef](#)]
- Bennett, C.H.; Brassard, G.; Crépeau, C.; Maurer, U. Generalized privacy amplification. *IEEE Trans. Inf. Theory* **1995**, *41*, 1915–1923. [[CrossRef](#)]
- Lütkenhaus, N. Estimates for practical quantum cryptography. *Phys. Rev. A* **1999**, *59*, 3301–3319. [[CrossRef](#)]
- Fung, C.H.F.; Ma, X.; Chau, H.F. Practical issues in quantum-key-distribution postprocessing. *Phys. Rev. A* **2010**, *81*, 012318. [[CrossRef](#)]
- Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
- Martin, K.M. *Everyday Cryptography: Fundamental Principles and Applications*; Oxford University Press: New York, NY, USA, 2012.
- Stinson, D.R.; Paterson, M.B. *Cryptography: Theory and Practice*; CRC Press: Boca Raton, FL, USA, 2019.
- Katz, J.; Lindell, Y. *Introduction to Modern Cryptography*; CRC Press: Boca Raton, FL, USA, 2015.

16. Abidin, A. Authentication in Quantum Key Distribution: Security Proof and Universal Hash Functions. Ph.D. Thesis, Linköping University, Linköping, Sweden, 2013.
17. Wegman, M.N.; Carter, J.L. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci.* **1981**, *22*, 265–279. [[CrossRef](#)]
18. Krawczyk, H. *Advances in Cryptology—CRYPTO '94, Lecture Notes in Computer Science*; Springer: New York, NY, USA, 1994; Volume 839, p. 129.
19. Peev, M.; Nölle, M.; Maurhardt, O.; Lorünser, T.; Suda, M.; Poppe, A.; Ursin, R.; Fedrizzi, A.; Zeilinger, A. A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography. *Int. J. Quantum Inf.* **2005**, *3*, 225–231. [[CrossRef](#)]
20. Abidin A.; Larsson, J.-Å. Vulnerability of “A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography. *Int. J. Quantum Inf.* **2009**, *7*, 1047–1052. [[CrossRef](#)]
21. Pacher, C.; Abidin, A.; Lorünser, T.; Peev, M.; Ursin, R.; Zeilinger, A.; Larsson, J.-Å. Attacks on quantum key distribution protocols that employ non-ITS authentication. *Quantum Inf. Process.* **2016**, *15*, 327–362. [[CrossRef](#)]
22. Wang, L.-J.; Zhang, K.-Y.; Wang, J.-Y.; Cheng, J.; Yang, Y.-H.; Tang, S.-B.; Yan, D.; Tang, Y.-L.; Liu, Z.; Yu, Y.; et al. Experimental authentication of quantum key distribution with post-quantum cryptography. *npj Quant. Inf.* **2021**, *7*, 67. [[CrossRef](#)]
23. Yang, Y.-H.; Li, P.-Y.; Ma, S.-Z.; Qian, X.-C.; Zhang, K.-Y.; Wang, L.-J.; Zhang, W.-L.; Zhou, F.; Tang, S.-B.; Wang, J.-Y.; et al. All optical metropolitan quantum key distribution network with post-quantum cryptography authentication. *Opt. Express* **2021**, *29*, 25859. [[CrossRef](#)] [[PubMed](#)]
24. Mosca, M.; Stebila, D.; Ustaoglu, B. Quantum key distribution in the classical authenticated key exchange framework. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 136–154.
25. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical One-way Functions. *Science* **2002**, *297*, 2026–2030. [[CrossRef](#)]
26. McGrath, T.; Bagci, I. E.; Wang, Y. M.; Roedig, U.; Young, R.J. A PUF taxonomy. *Appl. Phys. Rev.* **2019**, *6*, 011303. [[CrossRef](#)]
27. Gao, Y.; Al-Sarawi, S.F.; Abbott, D. Physical unclonable functions. *Nat. Electron.* **2020**, *3*, 81–91. [[CrossRef](#)]
28. Covic, A.; Chowdhury, S.; Acharya, R.Y.; Ganji, F.; Forte, D. Post-Quantum Hardware Security. In *Emerging Topics in Hardware Security*; Springer: Berlin/Heidelberg, Germany, 2021.
29. Chowdhury, S.; Covic, A.; Acharya, R.Y.; Dupee, S.; Ganji, F.; Forte, D. Physical security in the post-quantum era. *J. Crypt. Eng.* **2022**, *12*, 267–303. [[CrossRef](#)]
30. Arppe, R.; Just Sørensen, T. Physical unclonable functions generated through chemical methods for anti-counterfeiting. *Nat. Rev. Chem.* **2017**, *1*, 0031. [[CrossRef](#)]
31. Herder, C.; Yu, M.D.; Koushanfar, F.; Devadas, S. Physical unclonable functions and applications: A tutorial. *Proc. IEEE* **2014**, *102*, 1126–1141. [[CrossRef](#)]
32. Shamsoshoara, A.; Korenda, A.; Afghah, F.; Zeadally, S. A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. *Comput. Netw.* **2020**, *183*, 107593. [[CrossRef](#)]
33. Rührmair, U.; Devadas, S.; Koushanfar, F. *Introduction to Hardware Security and Trust*; Springer: Berlin/Heidelberg, Germany, 2012; Chapter 4.
34. Nikolopoulos, G.M. Remote Quantum-Safe Authentication of Entities with Physical Unclonable Functions. *Photonics* **2021**, *8*, 289. [[CrossRef](#)]
35. Horstmayr, R.; Judkewitz, B.; Vellekoop, I.M.; Assaworarith, S.; Yang, C. Physical key-protected one-time pad. *Sci. Rep.* **2013**, *3*, 3543. [[CrossRef](#)] [[PubMed](#)]
36. Bernstein, D.J.; Buchmann, J.; Dahmen, J. *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009.
37. Bernstein, D.J.; Lange, T. Post-quantum cryptography. *Nature* **2017**, *549*, 188–194. [[CrossRef](#)] [[PubMed](#)]
38. Nikolopoulos, G.M. Applications of single-qubit rotations in quantum public-key cryptography. *Phys. Rev. A* **2008**, *77*, 032348. [[CrossRef](#)]
39. Kawachi, A.; Koshihara, T.; Nishimura, H.; Yamakami, T. Computational Indistinguishability Between Quantum States and Its Cryptographic Application. *J. Cryptol.* **2012**, *25*, 528–555. [[CrossRef](#)]
40. Kabashima, Y.; Murayama, T.; Saad, D. Cryptographical Properties of Ising Spin Systems. *Phys. Rev. Lett.* **2000**, *84*, 2030–2033. [[CrossRef](#)]
41. Abidin A.; Larsson, J.-Å. Direct proof of security of Wegman–Carter authentication with partially known key. *Quantum Inf. Process.* **2014**, *13*, 2155–2170. [[CrossRef](#)]
42. Mesaritakis, C.; Akriotou, M.; Kapsalis, A.; Grivas, E.; Chaintoutis, C.; Nikas, T.; Syvridis, D. Physical Unclonable Function based on a Multi-Mode Optical Waveguide. *Sci. Rep.* **2018**, *8*, 9653. [[CrossRef](#)] [[PubMed](#)]
43. Nikolopoulos, G.M.; Fischlin, M. Information-Theoretically Secure Data Origin Authentication with Quantum and Classical Resources. *Cryptography* **2020**, *4*, 31. [[CrossRef](#)]
44. Bellare, M.; Kilian, J.; Rogaway, P. The Security of the Cipher Block Chaining Message Authentication Code. *J. Comput. Syst. Sci.* **2000**, *61*, 362–399. [[CrossRef](#)]
45. Marakis, E.; Rührmair, U.; Lachner, M.; Uppu, R.; Škorić, B.; Pinkse, P.W.H. Clones of the Unclonable: Nanoduplicating Optical PUFs and Applications. *arXiv* **2022**, arXiv:2212.12495. [[CrossRef](#)]
46. Škorić, B. Quantum readout of physical unclonable functions. *Int. J. Quantum. Inform.* **2012**, *10*, 1250001. [[CrossRef](#)]

47. Goorden, S.A.; Horstmann, M.; Mosk, A.P.; Škorić, B.; Pinkse, P.W.H. Quantum-secure authentication of a physical unclonable key. *Optica* **2014**, *1*, 421–424. [[CrossRef](#)]
48. Nikolopoulos, G.M.; Diamanti, E. Continuous-variable quantum authentication of physical unclonable keys. *Sci. Rep.* **2017**, *7*, 46047. [[CrossRef](#)]
49. Nikolopoulos, G.M. Continuous-variable quantum authentication of physical unclonable keys: Security against an emulation attack. *Phys. Rev. A* **2018**, *97*, 012324. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.