Secure quantum key distribution with malicious devices

Víctor Zapatero* and Marcos Curty

Escuela de Ingeniería de Telecomunicación, Department of Signal
Theory and Communications, University of Vigo, Vigo E-36310, Spain

The malicious manipulation of quantum key distribution (QKD) hardware is a serious threat to its security, as, typically, neither end users nor QKD manufacturers can validate the integrity of every component of their QKD system in practice. One possible approach to re-establish the security of QKD is to use a redundant number of devices. Following this idea, we introduce an efficient distributed QKD post-processing protocol and prove its security in a variety of corruption models of the possibly malicious devices. We find that, compared to the most conservative model of active and collaborative corrupted devices, natural assumptions lead to a significant enhancement of the secret key rate and considerably simpler QKD setups. Furthermore, we show that, for most practical situations, the resulting finite-size secret key rate is similar to that of the standard scenario assuming trusted devices.

I. INTRODUCTION

Quantum key distribution [1–4] (QKD) allows for information-theoretically secure communications, unaffected by the long-term security weakening inherent to public-key cryptography [5, 6]. Its security relies on fundamental physical principles and various assumptions, a crucial one being that the legitimate QKD users, say Alice and Bob, hold honest devices that stick to the QKD protocol and do not intentionally leak their private information to an eavesdropper (Eve). However, this strong assumption is probably unjustified, considering the amount of hardware and software Trojan horse attacks (THAs) against conventional cryptographic systems reported in the last years [7–11]. After all, likewise conventional security hardware, QKD devices incorporate many sophisticated components typically provided by specialised companies, and neither QKD vendors nor users are capable of validating the security of all these components in practice [12]. However, a malicious component can totally compromise the security of QKD. Indeed, the fabrication process of QKD systems might provide Eve with plenty of opportunities to meddle with the QKD hardware, including both the optical equipment and the classical post-processing (CP) units. Moreover, Eve could evade post-fabrication tests by arranging attack triggers that depend on a sequence of unlikely events [10, 13].

Remarkably, not even device-independent (DI) QKD [14–18] can provide security against malicious devices, as shown in [19]. It is the classical nature of the secret keys that makes QKD systems vulnerable to classical hacking in both the DI and the non-DI scenarios, because classical keys are susceptible to copying.

A possible solution to foil malicious hardware and software in QKD was recently presented in [20], and then experimentally demonstrated in [21]. The triggering idea is that it might be more difficult for Eve to corrupt various devices than a single device, for example, if they originate from different providers. Therefore, one can use a redundant number of devices for both the raw key generation and the post-processing of QKD. As shown in [20], under the assumption that the number of devices controlled by Eve is restricted, secure QKD is possible by combining verifiable secret sharing (VSS) [22–26]—whose essential building block is secret sharing (SS) [27, 28], a standard technique in secure hardware design [29]—and privacy amplification (PA) [30, 31]. To be precise, VSS (PA) allows to protect a QKD system against malicious CP units (optical apparatuses, such as QKD transmitters or detection modules).

However, the analysis in [20] is restricted to the probably over-conservative scenario where all the corrupted devices are active and collaborative. What is more, an evaluation of the performance of QKD in the presence of corrupted devices is missing. To cope with these limitations, in this work we present an efficient distributed QKD post-processing scheme tailored for a setting with corrupted devices, and prove its security in a variety of eavesdropping models. For this purpose, we introduce a suitable cryptographic primitive weaker than standard VSS, which we refer to as conditional VSS. In addition, we combine our post-processing protocol with well-known QKD schemes to evaluate the resulting secret key rate. We find that, compared to the conservative scenario considered in [20], some eavesdropping models of practical relevance enable a significant enhancement of the secret key rate, while requiring fewer honest devices and classical communications. Moreover, for practical data block-sizes and moderate numbers of corrupted devices, our results show that the increased authentication cost of our scheme (compared to that of standard QKD post-processing) is negligible with respect to the extractable secret key length.

*Electronic address: vzapatero@com.uvigo.es
II. RESULTS

We start by describing the general formalism we consider. Without loss of generality, a standard QKD setup can be divided into two parts with separate roles: a QKD module and a classical post-processing (CP) unit. Alice’s and Bob’s QKD modules form a so-called QKD pair, whose role is to generate raw correlated data between the parties via quantum communication. Each module transfers its raw data to its local CP unit, and the two distant CP units distill a pair of secret keys from the raw data via coordinated classical post-processing and authenticated classical communication.

The focus of this work is the general scenario where not all the devices are trusted, thus forcing the parties to use a redundant number of them [20]. Throughout the paper, we shall assume that Alice and Bob share \( n_q \) QKD pairs (or simply “pairs”), and that each of them holds \( n_c \) CP units (or simply “units”). Nevertheless, our results could be trivially adapted to the case where each party has a different number of units. For \( j = 1, \ldots, n_q \), Alice’s (Bob’s) module \( \text{QKD}_{A_j} \) (\( \text{QKD}_{B_j} \)) is connected to all her (his) units \( \{\text{CP}_{A_i}\}_{i=1}^{n_c} \) \( \{\text{CP}_{B_i}\}_{i'=1}^{n_c} \) via secure channels, i.e., channels that provide both privacy and authentication. Also, all of Alice’s (Bob’s) units are pairwise connected by secure channels too. Since all these links take place within Alice’s (Bob’s) lab, in practice security could be enforced by using, say, physically protected cables. Similarly, the \( \text{CP}_{A_i} \) are connected to the \( \text{CP}_{B_j} \) by authenticated classical channels. And lastly, as usual, a quantum channel fully accessible to an eavesdropper links \( \text{QKD}_{A_j} \) to its partner \( \text{QKD}_{B_j} \). A schematic of this QKD setup is given in Fig. 1.

A. Active collaborative eavesdroppers

It was shown in [20] that the setting above—consisting of multiple QKD pairs and multiple CP units—allows for secure QKD in the presence of a restricted number of corrupted devices, even under the conservative assumption that all of them fully obey a single Eve who can access their internal information and make them arbitrarily misbehave from the protocol. We refer to this scenario as the active collaborative (AC) model. More precisely, Eve is treated here as a so-called threshold active adversary [26]. This approach amounts to assuming that she can corrupt up to, say \( t_q \) QKD pairs (a QKD pair is said to be corrupted when at least one of its modules is), and up to \( t_c \) CP units per lab. The solution presented in [20] to achieve secure QKD in the AC model is based on combining PA with VSS. On the one hand, PA is required to remove not only the information Eve gains through her intervention in the quantum channel (as it is done in standard QKD post-processing), but also the information she learns from the corrupted QKD pairs, which is possible as long as \( n_q > t_q \). On the other hand, VSS enables an honest QKD module to split a raw key into shares and redundantly distribute them among its \( n_c \) local CP units, in such a way that no group of at most \( t_c \) units can collaborate to reconstruct the key (privacy). In order to perform linear local operations on the key (i.e., the ones typically required by the post-processing of QKD), the units individually apply these operations on their shares, and the final output can be reconstructed from its shares due to linearity. In any such operation, VSS guarantees that all honest units can reconstruct a common output as long as \( n_c > 3t_c \) [26] (commitment), which indeed is the correct output if the QKD module is honest (correctness).

Due to the use of standard VSS, the proposal in [20] is resilient to the misbehaving of the CP units at the price of relying on the availability of a possibly simulated broadcast channel [32], a very stringent requirement in practice. Here, we devise a general post-processing protocol that replaces standard VSS by conditional VSS (defined in Sec. IV), which circumvents broadcast by contemplating the possibility of aborting. After all, abortion is an ultimately unavoidable feature of QKD due to the unrestricted access that Eve has to the quantum channel. Precisely, in a conditional VSS scheme, commitment...
(correctness) is only guaranteed upon non-abortation of the scheme.

Also, one drawback of the proposal in [20] is that its implementation is rather cumbersome, as it requires the execution of \( n_c + 1 \) separate PA steps. Instead, our protocol implements PA in a single step. In addition, it minimises the classical communication between labs to reduce the extra authentication cost provoked by the use of multiple devices.

1. QKD post-processing protocol

Prior to the QKD session, Alice and Bob agree on the number \( t_q \geq 0 \) (\( t_c \geq 0 \)) of corrupted QKD pairs (CP units per lab) they want to be protected against, which forces them to hold \( n_q = t_q + 1 \) QKD pairs (\( n_c = 3t_c + 1 \) CP units per lab) at least in the AC model under consideration. Similarly, they agree on a correctness (secrecy) parameter, \( \epsilon_{cor} (\epsilon_{sec}) \), and a total authentication error satisfying \( \epsilon_{AU} < \epsilon_{cor} \) and \( \epsilon_{AU} < \epsilon_{sec} \).

For \( j = 1, \ldots, n_q \), the pair \((\text{QKD}_A, \text{QKD}_B)\) runs a QKD session to generate the basis \( 2 \) raw key strings, \((r_A^j, r_B^j)\), to be kept private, and some non-private protocol information, \((\text{info}_A^j, \text{info}_B^j)\), typically including the basis and intensity settings, detection events, etc. Crucially, \((\text{info}_A^j, \text{info}_B^j)\) includes all the raw key material required for parameter estimation. The post-processing protocol starts next and is described below. Although the description assumes that the possibly corrupted devices do not deviate from the protocol, the protocol is indeed secure against active eavesdroppers, as proven later. Finally, although not explicitly stated, in case of abortion, the aborting party must notify the other party.

Protocol. Let us focus on, say, the \( j \)-th QKD pair.

1. Distribution of data. QKD\(_A\), (QKD\(_B\)) distributes shares of its raw key \( r_A^j \) (\( r_B^j \)) among the CP\(_A\), following the Share protocol of a conditional VSS scheme (see Sec. IV). We denote the set of units that receive the \( i \)-th share of \( r_A^j \) (\( r_B^j \)) by \( \sigma_i^A \) (\( \sigma_i^B \)), which without loss of generality is common for all \( j = 1, \ldots, n_q \). In addition, QKD\(_A\) (QKD\(_B\)) sends the protocol information \( \text{info}_A^j \) (\( \text{info}_B^j \)) to all CP\(_A\) \( \in \sigma_i^A \) (CP\(_B\) \( \in \sigma_i^B \)), and the latter perform a consistency test on this data: they pairwise check that their copies of \( \text{info}_A^j \) (\( \text{info}_B^j \)) match via authenticated channels. If a CP\(_A\) (CP\(_B\)) finds an inconsistency, she aborts the protocol (see the Share protocol in the Methods section for the two-step abortion procedure we consider).

2. Sifting. Each CP\(_A\) \( \in \sigma_1^A \) sends its copy of \( \text{info}_A^j \) to all CP\(_B\) \( \in \sigma_1^B \), which individually apply majority voting (MV) to decide on a single copy. Then, the CP\(_B\) \( \in \sigma_1^B \) forward some sifting information, \( \text{sift}_i^j \), computable from the pair \((\text{info}_A^j, \text{info}_B^j)\), to the CP\(_B\) \( \notin \sigma_1^B \), which apply MV too. Using \( \text{sift}_i^j \), every CP\(_B\) discards some key bits from their shares of \( r_B^j \) to obtain shares of the sifted key, \( s_B \).

3. Parameter estimation. Using \((\text{info}_A^j, \text{info}_B^j)\), each CP\(_B\) \( \in \sigma_1^B \) computes a hypothetical lower bound \( h^j \) (see the Supplementary Information for the details) on the \( \epsilon \)-smooth min-entropy of \( s_B \) conditioned on the information held by an eavesdropper up to the parameter estimation (PE) step, for a pre-agreed \( \epsilon \) determined by \( \epsilon_{sec} \).

Once steps 1 to 3 are implemented for \( j = 1, \ldots, n_q \), all CP\(_B\) construct their shares of the concatenated sifted key \( s_B = [s_1^B, \ldots, s_{n_q}^B] \), such that the \( k \)-th share of \( s_B \) is simply given by the concatenation of the \( k \)-th share of \( s_1^B \), the \( k \)-th share of \( s_2^B \), and so on. In addition, from all \( n_q \) hypothetical values \( h^j \), every CP\(_B\) \( \in \sigma_1^B \) computes a lower bound \( l \) on the secret key length extractable from \( s_B \) via PA (see Sec. III for the explicit formula of \( l \)). If a CP\(_B\) \( \in \sigma_1^B \) finds \( l \leq 0 \), it aborts the protocol. Otherwise, the post-processing proceeds as follows.

4. RBS generation. Every CP\(_B\) \( \in \sigma_1^B \) forwards \( l \) to the CP\(_B\) \( \notin \sigma_1^B \), which apply MV. All CP\(_B\) perform a RBS generation protocol to select two random \( 2 \)-universal hash functions, \( h_{EV} \) and \( h_{PA} \), respectively devoted to error verification (EV) and PA.

5. Information reconciliation. Every CP\(_A\) computes its shares of the string of concatenated syndromes, \( s_B = \{s(y(s_1^A), \ldots, s(y(s_{n_q}^A)))\} \), and the EV tag \( h_{EV,B} = h_{EV}(s_B) \). Here, \( sy(k) \) is a linear function specified by an error correction (EC) protocol for a pre-agreed quantum bit error rate (QBER).

All together, the CP\(_B\) reconstruct \( s_B \) and \( h_{EV,B} \) via the Reconstruct protocol of a conditional VSS scheme (see Sec. IV). Each CP\(_B\) \( \in \sigma_1^B \) sends the following items to every CP\(_A\) \( \in \sigma_1^A \):

(a) The total sifting information, \( \{\text{sift}_i^j\}_{i=1}^{n_q} \).

(b) The syndrome information, \( sy_B \), a description of \( h_{EV} \) and the EV tag, \( h_{EV,B} \).

(c) A description of \( h_{PA} \).

For all 3 items, each CP\(_A\) \( \in \sigma_1^A \) applies MV to decide on a single copy. Then, it forwards \( \{\text{sift}_i^j\}_{i=1}^{n_q} \), \( h_{EV} \) and \( h_{PA} \) to the CP\(_A\) \( \notin \sigma_1^A \) (which apply MV too), and every CP\(_A\) sifts its shares of the raw keys \( r_A^j \) to obtain shares of the concatenated sifted key \( s_A = [s_A^1, \ldots, s_A^{n_q}] \). Following the EC protocol, all CP\(_A\) compute their shares of the concatenated syndrome string, \( s_A = \{sy(s_A^1), \ldots, sy(s_A^{n_q}))\} \), and
jointly reconstruct it via the Reconstruct protocol of a conditional VSS scheme. From $sy_B$ and $sy_A$, each $CP_{A_j} \in \sigma^A_1$ computes the error pattern $\hat{e}$ and updates its copy of the first share of $s_A$ by XOR-ing it with $\hat{e}$. Thus, by construction, Alice’s corrected key is $\hat{s}_A = s_A \oplus \hat{e}$, “@” denoting bitwise XOR. Then, all $CP_{A_j}$ compute their shares of the EV tag $h_{EV,A} = h_{EV}(\hat{s}_A)$ and jointly reconstruct it via the Reconstruct protocol of a conditional VSS scheme. Finally, every $CP_{A_j} \in \sigma^A_1$ checks that $h_{EV,A} = h_{EV,B}$. Otherwise, it aborts the protocol.

6. Privacy amplification. In case of not aborting, every $CP_{A_j} (CP_{B_j})$ computes its shares of the final key $k_A = h_{PA}(\hat{s}_A)$ ($k_B = h_{PA}(s_B)$).

From steps 1 to 6, and given that $n_q > t_q$ and $n_c > 3t_c$, the following security claim holds irrespectively of the misbehaving of the corrupted devices.

**Claim.** Suppose that Protocol does not abort. Then, Alice and Bob can unambiguously determine unique $\epsilon_{\text{cor}}$-correct and $\epsilon_{\text{sec}}$-secret final keys.

Importantly, the determination of such final keys by Alice and Bob can be done by simply applying MV on the key shares held by their respective CP units, followed by an XOR operation. More generally, in the presence of untrusted units, the $CP_{A_j}$ (CP$_{B_j}$) can forward their final shares to a local key management layer [33, 34]. There, they could be stored in distributed memories or employed for applications such as message encryption, which in turn can be performed share-wise too in the presence of untrusted devices.

A proof of the claim based on the properties of conditional VSS and PA is given in the Supplementary Information. In particular, it is shown there that the secret key length extractable via Protocol is given by

$$l = \left\lfloor \min_j \left( h^j_{\epsilon} - |sy(s_B^j)| \right) - \log_2 \left( \frac{1}{\epsilon_{\text{cor}}} \right) \right\rfloor,$$

(1)

where $j = 1, \ldots, n_q$, $\epsilon$ and $h^j_{\epsilon}$ were introduced in the PE step, $|sy(s_B^j)|$ is the number of bits in $sy(s_B^j)$, $\epsilon_{\text{cor}} = \epsilon_{\text{cor}} - \epsilon_{\text{AU}}$, $\epsilon_{\text{PA}}$ is the error probability of PA, and $\delta > 0$. In addition, the overall secrecy parameter is given by

$$\epsilon_{\text{sec}} \geq 2\epsilon + \delta + \epsilon_{\text{PA}} + \epsilon_{\text{AU}},$$

(2)

where we recall that the total authentication error, $\epsilon_{\text{AU}}$, is selected a priori by Alice and Bob.

Let $l_{\text{AU}}$ denote the number of secret bits consumed for authentication during Protocol (quantified in the next section). Then, the secret key rate reads

$$K = \frac{l - l_{\text{AU}}}{n_qN},$$

(3)

where $N$ stands for the total number of transmission rounds per QKD pair. As discussed next, the authentication cost of a message scales logarithmically with its length, meaning that for most practical situations $l_{\text{AU}} \ll l$ and thus $K \approx l/(n_qN)$ (see Sec. III).

2. Authentication

Here, we quantify the amount of secret bits that Protocol consumes for the authentication of the classical communications between labs. This task requires every $CP_{A_j} \in \sigma^A_1$ to pre-share a so-called key pool of secret bits with every $CP_{B_j} \in \sigma^B_1$. Assuming a pre-fixed size $|k|$ for all the key pools and recalling that both $\sigma^A_1$ and $\sigma^B_1$ contain $2t_c + 1$ CP units, the overall authentication cost reads

$$l_{\text{AU}} = R^2 \times |k|$$

(4)

bits, where we defined $R = |\sigma^A_1| = |\sigma^B_1|$. Indeed, $R$ is the common size of all $\sigma^A_1$ ($\sigma^B_1$). If, for instance, we follow the scheme based on Toeplitz matrices described in [35] (see the Supplementary Information), the authentication of a message $m$ (containing $|m|$ bits) with an error probability $\gamma$ consumes $\lceil \log_2 (2|m|/\gamma) \rceil$ bits of the corresponding key pool.

According to Protocol, the sifting step requires the authentication of $n_q$ messages from Alice to Bob, say $\{m_A^j\}_{j=1}^{n_q}$ with $m_A^j = \text{info}_{A_j}^j$, and the information reconciliation step requires the authentication of a single message from Bob to Alice, say $m_B$ (consisting of various items). Therefore, although $\{m_A^j\}_{j=1}^{n_q}$ and $|m_B|$ are not known a priori, it is required that

$$|k| \geq \sum_{j=1}^{n_q} \left[ \log_2 \left( \frac{2|m_A^j|}{\gamma_{\text{AU}}} \right) + \log_2 \left( \frac{2|m_B|}{\gamma_{\text{AU}}} \right) \right]$$

(5)

secret bits, setting a common error probability, $\gamma_{\text{AU}}$, for every communication. In fact, since only an authentication error between two honest units may compromise the security, we find that $\gamma_{\text{AU}}$ and $\epsilon_{\text{AU}}$ must be related as

$$\epsilon_{\text{AU}} \geq (t_c + 1)^2(n_q + 1)\gamma_{\text{AU}}.$$ 

(6)

Finally, note that authentication errors may compromise both the secrecy and the correctness of the final keys due to the possibility that a malicious unit impersonates an honest one. Thus, $\epsilon_{\text{AU}}$ contributes to both $\epsilon_{\text{sec}}$ and $\epsilon_{\text{cor}}$.

B. Alternative corruption models

The previous section considers the most conservative adversarial scenario, that is, the AC model. However, there might be situations where this model is over-conservative. For instance, if Alice and Bob purchase devices from different vendors, it might be reasonable to expect that, even if they are corrupted, they do not collaborate, meaning that they do not share their private information with each other or cooperate in any way.
Also, if the information delivered by a certain device is different from the one prescribed by the protocol, it might be detected by Alice and Bob a posteriori. In this sense, some QKD users might only request security against passive (rather than active) corrupted devices, which follow the protocol prescriptions but covertly leak their internal information to an eavesdropper. For example, this might happen due to the existence of a side-channel that leaks information to Eve from an honest device.

In this section, we analyse a wide spectrum of such alternative corruption models to investigate their advantages in terms of secret key rate and necessary resources. Due to its generality, Protocol can be directly applied to all the models we consider, such that distinct models simply determine different protocol settings, compared to the AC scenario. Importantly, we decouple the analysis of the different corruption models for the QKD modules and the CP units, i.e., the results we present for the modules do not assume a specific model for the units and vice versa. For conciseness though, we restrict the analysis to the non-mixed corruption models where all the malicious devices of each kind belong to the same model.

1. QKD modules

The privacy of conditional VSS guarantees that the secret key length extractable via Protocol does not depend on the corruption model of the CP units, but only on that of the QKD pairs. In particular, compared to the AC model, three looser non-mixed corruption models exist: passive and collaborative (PC), active and non-collaborative (AN) and passive and non-collaborative (PN). Also, note that non-collaboration is obviously only defined for \( t_q > 1 \). In what follows, we show that only the PN model allows to enhance the secret key rate.

As long as the malicious QKD pairs are collaborative, an omniscient Eve could learn all the information they hold about the keys, and as long as they are active, they can deliver untrustworthy protocol information unsuitable for correct PE. Hence, although for different reasons, the intermediate scenarios PC and AN cannot lead to an enhancement of the secret key length with respect to the AC model: they require to remove all the key material that comes from corrupted QKD pairs via PA, thus also demanding \( n_q > t_q \). In particular, it can be shown that the extractable key length for \( n_q = t_q + 1 \) in the PC (AN) corruption model is given by Eq. (1) too.

In the PN corruption model, one assumes an independent Eve per malicious QKD pair who does not collaborate with the eavesdroppers possibly controlling the other pairs. Moreover, passivity implies that corrupted pairs deliver trustworthy protocol information which allows to quantify the ignorance (in secret bits) that the Eves possibly corrupting other pairs have about their raw data. In this case, it suffices to remove the information held by the most knowledgeable eavesdropper via PA in order to provide security against all of them. As a consequence, secure QKD is possible even if all the QKD pairs are corrupted in the PN model. Setting \( n_q = t_q \), the secret key length \( l \) extractable via Protocol in the PN model (see Appendix D in the Supplementary Information) is given by

\[
l = \min_v \frac{1}{\epsilon_{\text{cor}} \epsilon_{\text{PA}}^2} \left( H_{\text{min}}(s_B^v | E_v) - |s_y(s_B^v)| \right) - \log_2 \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{\text{PA}}^2} \delta_{n_q - 1} \right), \tag{7}
\]

which obviously leads to a greater secret key rate via Eq. (3) than Eq. (1) for any given \( t_q > 1 \) (we recall that non-collaboration is only defined in this case). Note that all the parameters in Eq. (7) are defined in Eq. (1), except from \( E_v \). This quantity stands for the information held by the \( v \)-th eavesdropper, i.e., the one that corrupts the \( v \)-th QKD pair, where \( v = 1, \ldots, n_q \). Also, we remark that no distinction is made between the bounds \( h^\epsilon \) computed by the CP \( B_v \in \sigma^B \) in the PE step of Protocol and the actual bounds on \( H_{\text{min}}(s_B^j | E_v) \) in Eq. (7). This is so because, by assumption, \( h^\epsilon \) certainly is a lower bound on \( H_{\text{min}}(s_B^j | E_v) \) for all \( v \neq j \) in the PN model.

The secrecy parameter attached to \( l \) is

\[
\epsilon_{\text{sec}} \geq (n_q - 1)(2\epsilon + \delta) + \epsilon_{\text{PA}} + \epsilon_{\text{AU}}. \tag{8}
\]

As a final remark, we note that, although an enhancement of the secret key rate is only achievable in the PN model, both passive models (PN and PC) allow for a straightforward simplification of Protocol. Namely, one can remove all the consistency tests in step 1, irrespectively of the corruption model of the CP units.

2. CP units

Although the corruption model of the CP units does not affect the extractable key length, \( l \), it determines the necessary resources to securely implement Protocol: the number of units per party, \( n_c \), the size of all sets \( \sigma^A \) and \( \sigma^B \), \( R \), and the total number of key shares managed per CP unit, say \( r \). On the one hand, \( n_c \) and \( R \) determine the necessary classical communications (both between labs and inside each lab), and the total authentication cost, \( l_{\text{AU}} \). On the other hand, \( r \) strongly affects the post-processing time, a usual concern in the performance of QKD.

In Table 1 we list the minimum values of \( n_c \), \( R \) and \( r \) required to implement Protocol within each corruption model of the CP units. The entries of the table follow from the requirements of our conditional VSS scheme in each case, and they are derived in Appendix D of the Supplementary Information. As we observe, all the restricted models allow to reduce \( n_c \), \( R \) and \( r \) with respect to the AC model. For instance, note that the number \( r \) of shares per unit grows exponentially with \( n_c \) for a fixed
fraction of corrupted units in the AC model. This might lead to prohibitively long post-processing times even for small values of $n_c$. Nevertheless, this problem disappears if one assumes that the possibly corrupted units are non-collaborative. As for the passive models, not only they preserve this advantage, but they also minimise the classical communications between labs and inside each lab: since $R = 1$, the consistency tests vanish and a single unit per lab conducts the lab-to-lab classical communications.

Lastly, as stated above, the corruption model of the units also determines the authentication cost, $I_{\text{AU}}$. To be precise, Protocol consumes $R^2$ key pools with a prefixed size of $|k|$ bits (see Eq. (4)), where $|k|$ is estimated in Eq. (5) and $R$ is given in Table 1 for each model. Also, using the same argument as in Sec. II A 2, the total authentication error probability, $\epsilon_{\text{AU}}$, and the individual error probability, $\gamma_{\text{AU}}$, are related as indicated in Eq. (6) for active corruption of the CP units, and simply as

$$\epsilon_{\text{AU}} \geq (n_q + 1)\gamma_{\text{AU}}$$

for passive corruption.

### III. DISCUSSION

Finally, we show the practicality of Protocol for the post-processing of QKD in the presence of corrupted devices. For this purpose, we combine it with well-known QKD schemes to evaluate the resulting secret key rate, $K$, given by Eq. (3), in the finite data regime.

For concreteness, we assume the same corruption model for the QKD modules and the CP units, a natural supposition in practice. Moreover, we restrict ourselves to the extreme corruption models, AC and PN, as the intermediate scenarios (AN and PC) do not allow to enhance the secret key rate, disregarding the authentication cost (see Sec. II B). We also assume that Alice and Bob use the minimum number of devices that allows for $K > 0$, which depends on the corruption model they consider. For AC corruption, this means that they agree on the number $t_q \geq 0$ ($t_c \geq 0$) of malicious QKD pairs (CP units per lab) they want to be protected against, and use $n_q = t_q + 1$ pairs ($n_c = 3t_c + 1$ units per lab). Alternatively, for PN corruption, they use $n_q = 2$ QKD pairs and $n_c = 2$ CP units per party, which suffices to achieve $K > 0$ even if all the devices are possibly malicious (see Sec. II B).

Also, the number of units in every $\sigma^A_i$ ($\sigma^B_j$), $R$, is fixed by the pre-agreed corruption model via Table 1. $R$ determines the total authentication cost, $I_{\text{AU}}$, via Eq. (4), and the value of the individual authentication error probability, $\gamma_{\text{AU}}$, via Eq. (6) in the AC model and via Eq. (9) in the PN model.

We consider two practical QKD protocols with decoy states: an efficient MDI-QKD scheme [36] with three decoy intensities in the basis X (devoted to PE) and one signal intensity in the basis Z (devoted to key distillation), and the standard decoy-state BB84 scheme [37] with three decoy intensities per basis. Detailed analyses of these protocols are provided in Appendices E and F of the Supplementary Information, respectively. For each protocol, we compute estimates of the secret key length, $l$ (given by Eq. (1) for the AC model and by Eq. (7) for the PN model), and the authentication cost, $I_{\text{AU}}$ (given by Eq. (4)), by setting the observables to their expected values according to respective channel models described in the cited appendices. These channel models depend on various common experimental parameters: the efficiency of the photodetectors, set to $\eta_{\text{det}} = 65\%$, their dark count probability, set to $d_4 = 7.2 \times 10^{-8}$ (both values matching the recent MDI-QKD experiment reported in [38]), and the polarization misalignment, set to, say $\delta_{\text{mis}} = 0.08$ for illustration purposes. Moreover, in both the MDI-QKD and the BB84 schemes, the weakest decoy intensity is set to $\omega = 10^{-3}$ for the numerics. In each case, we optimise the remaining protocol inputs (i.e., intensity settings, and basis and decoy probabilities) to maximize $K$ as a function of the channel loss between Alice and Bob.

For the finite key analysis, we select a post-processing block-size of $M$ bits. Then, for every value of the channel loss, we choose the smallest number of transmission rounds per QKD pair, $N$, that assures that all $n_q$ sifted keys reach this block-size except with a probability of, say $\gamma_{\text{sift}} = 5 \times 10^{-3}$, according to the channel model.

Regarding the EC leakage, we assume the typical model $|s_y(s^i_{\text{B}})| = M f_{\text{EC}} h(E_{\text{tol}})$ for every EC syndrome, where $f_{\text{EC}} = 1.16$ is the efficiency of the EC protocol, $h(\cdot)$ is the binary entropy function, and $E_{\text{tol}}$ is a prefixed threshold QBER. In particular, $E_{\text{tol}}$ is an upper bound on the QBER that any pair of sifted keys can reach according to the channel model, except with an error probability of $\gamma_{\text{EC}} = 5 \times 10^{-3}$.

Finally, the security parameters are set to $\epsilon_{\text{cor}} = \epsilon_{\text{sec}} = 10^{-8}$ and $\epsilon_{\text{AU}} = 5 \times 10^{-9}$. Given $\epsilon_{\text{sec}}$ and $\epsilon_{\text{AU}}$, the remaining parameters, $\epsilon_{\text{PA}}$ and $\delta$, entering the extractable key length, $l$, are determined by imposing a common value, $\gamma_{\text{sec}}$, for every error term that contributes to $\epsilon_{\text{sec}} = \epsilon_{\text{sec}} - \epsilon_{\text{AU}}$ (given by Eq. (2) and Eq. (8)).
In particular, from the PE procedure presented in Appendix E (Appendix F), it follows that \( \gamma_{\text{sec}} = \hat{\epsilon}_{\text{sec}}/48 \) in the MDI-QKD (BB84) scheme within both the AC and the PN scenarios, where we used the fact that \( n_q = 2 \) in the latter case.

Adhering to all the above, in Fig. 2, we plot the secret key rate, \( K \), as a function of the total channel loss, for the AC and the PN models in the MDI-QKD scheme, considering the symmetric case where both Alice and Bob are at the same distance of the central untrusted node. The corresponding figure for the BB84 scheme is included in Appendix F of the Supplementary Information. In both cases, for illustration purposes two different block-sizes are considered, \( M \in \{10^5, 10^6\} \). Within the AC corruption model, for concreteness we only address the symmetric case \( t_q = t_c = t \), such that \( n_q = t + 1 \) and \( n_c = 3t + 1 \). Hence, we use the notation \( K_{\text{AC},t} \) \( (l_{\text{AC},t}) \) for the secret key rate (length) secure against \( t \) corrupted devices of each kind in this model. Similarly, \( K_{\text{PN}} \) \( (l_{\text{PN}}) \) denotes the secret key rate (length) in the PN model, which, as explained above, unambiguously requires \( n_q = n_c = 2 \). Lastly, \( K_{\text{honest}} \) \( (l_{\text{honest}}) \) denotes the secret key rate (length) assuming all-honest devices, i.e., \( K_{\text{honest}} = K_{\text{AC},0} \) \( (l_{\text{honest}} = l_{\text{AC},0}) \). Note that the honest scenario corresponds to the standard situation where each party holds one QKD module and one CP unit and both of them are trusted.

The conclusions gathered from Fig. 2 are readily understood in view of the results of Sec. II. In the first place, for both \( M = 10^5 \) and \( M = 10^6 \), we find that the secret key rate in the PN model matches that of the AC model with \( t = 1 \) to a precision that cannot be distinguished in the figure, i.e., \( K_{\text{PN}} \approx K_{\text{AC},1} \). This follows from the fact that, in both cases, two raw keys are generated (as \( n_q = 2 \) and the parties need to remove the information from one of them via PA. Indeed, comparing Eq. (1) and Eq. (2) with Eq. (7) and Eq. (8), one observes that

\[
l_{\text{PN}} = l_{\text{AC},1},
\]

i.e., the secret key lengths coincide exactly for given security parameters. Thus, the minuscule difference between \( K_{\text{PN}} \) and \( K_{\text{AC},1} \) comes from the authentication cost, as \( l_{\text{AU}} \) is proportional to \( R^2 \), with \( R = 2t + 1 \) (\( R = 1 \)) in the AC (PN) setting.

An identical argument relates \( K_{\text{AC},t} \) and \( K_{\text{honest}}/t+1 \) for all \( t \). This is so because, for fixed security parameters, \( \epsilon_{\text{sec}}, \epsilon_{\text{corr}} \) and \( \epsilon_{\text{AU}} \), and fixed experimental inputs, \( N \) and \( E_{\text{tot}} \), the exact same secret key length can be extracted via PA in both scenarios (see Eq. (1) and Eq. (2)). That is to say,

\[
l_{\text{AC},t} = l_{\text{honest}}
\]

for all \( t \). This length corresponds to the key material coming from the honest QKD pair in the AC model, which matches the honest scenario. At the same time, in the presence of \( t \) malicious QKD pairs, the extraction of the above key length requires the generation of \( t + 1 \)

![FIG. 2: Secret key rate, \( K \), that results from combining Protocol with a decoy-state MDI-QKD scheme [36] in various adversarial scenarios with malicious devices, as a function of the total channel loss between Alice and Bob (assumed to be at the same distance of the untrusted measurement node). Two finite block-sizes are considered, (a) \( M = 10^5 \) and (b) \( M = 10^6 \). In both figures, the purple line is the secret key rate in the standard scenario—where each party holds one QKD module and one classical post-processing (CP) unit, both of them trusted by hypothesis—and green lines denote different adversarial scenarios. In particular, the dashed-dotted phosphorescent line is the extractable secret key rate assuming that the corrupted devices are passive and non-collaborative, which requires the use of two QKD pairs and two CP units per lab (all of them being possibly malicious) to provide security. A more conservative adversarial scenario is represented by the solid non-phosphorescent green lines, which assume active and collaborative corrupted devices. These lines further assume the same number, say \( t \), of malicious QKD pairs and malicious CP units per lab, which requires the use of at least \( n_q = t + 1 \) QKD pairs and \( n_c = 3t + 1 \) CP units per party to provide security. Specifically, the dark (light) green line corresponds to \( t = 3 \) \( (t = 5) \).]
raw keys in the AC model. Thus, as \( K = (l-l_{AU})/(n_{q}N) \) in general, it follows that

\[
K_{\text{honest}} \frac{1}{t+1} - K_{\text{AC},t} = \frac{\delta l_{AU}}{(t+1)N} \tag{12}
\]

for fixed experimental and security parameters, where \( \delta l_{AU} \) denotes the extra authentication cost of the AC model with \( t_{q} = t_{c} = t \), compared to the honest scenario. Due to the factor \( N^{-1} \) in the right-hand side of Eq. (12), larger block sizes lead to smaller differences between \( K_{\text{honest}}/(t+1) \) and \( K_{\text{AC},t} \). Finally, since \( K_{\text{AC},t} \propto (l_{\text{honest}} - l_{AU}) \), and \( l_{AU} \propto (2t+1)^{2} \) in the AC corruption model, \( K_{\text{AC},t} \) vanishes for any given block size if a large enough number of CP units is considered, as eventually \( l_{AU} > l_{\text{honest}} \). This is the case for \( M = 10^{5} \) and \( t = 5 \) in Fig. 2.

Putting it all together, this work is a fundamental step forward towards the development of practical and secure QKD setups in the presence of malicious devices possibly sabotaged by a third party, a major threat against classical cryptography today that cannot be put aside in the quantum-safe era.

IV. METHODS

A. Conditional verifiable secret sharing

Here, we introduce a modified version of the VSS scheme presented in [26] that contemplates the possibility of aborting, thus providing a weaker cryptographic primitive than standard VSS. For this reason, we refer to it as conditional VSS.

We consider a scenario with one possibly corrupted dealer, \( D \), and a set of \( n \) parties, \( \mathcal{P} = \{P_{1}, \ldots, P_{n}\} \), of which are possibly corrupted. In this setting, a conditional VSS scheme is a pair of protocols, (Share, Reconstruct), satisfying three properties: privacy, conditional commitment and conditional correctness (defined below). In full generality, Share and Reconstruct run as follows. During Share, \( D \) distributes an input \( m \) among the \( n \) parties, which pairwise perform consistency tests on their common information via secure channels and possibly abort. Upon non-abortion of Share, during Reconstruct the parties collaborate to retrieve \( m \). The defining properties of conditional VSS are given below:

1. Privacy. If \( D \) is honest, the information obtained by any set of \( t \) or less parties prior to Reconstruct is independent of \( m \).

2. Conditional commitment. Upon non-abortion of Share, Reconstruct yields the same output for all honest (and/or passively corrupted) parties.

3. Conditional correctness. Upon non-abortion of Share, if \( D \) is honest the common output of all honest (and/or passively corrupted) parties is the input \( m \).

As for the corrupted parties, all four non-mixed corruption models presented in the main text shall be addressed: AC, AN, PC and PN. However, we do not restrict to any of them yet. As for the dealer, a corrupted \( D \) can distribute incorrect or inconsistent information about his input to the parties. If, in addition, the corrupted parties are collaborative, a corrupted \( D \) can collaborate with them too, following Eve’s instructions and possibly sharing his private information with them.

In what follows, we describe a pair of protocols, (Share, Reconstruct), that depend on various settings, and such that adequate choices of these settings confer the pair the category of a conditional VSS scheme. We remark that the adequacy of some given settings depends on the corruption model one assumes for the parties. As in the main text, the protocol definitions below assume that the parties and the dealer do not misbehave, whether or not these protocols are robust against active corruption. Also, the dealer’s input \( m \) is assumed to be a binary string, and we recall that the symbol “\( \oplus \)” denotes bitwise XOR. In addition, this operation is generalised to a pair of strings with different lengths by padding the shortest one with as many zeros as necessary for the lengths to match. This said, Share runs as follows.

Share protocol

1. \( D \) uses a \( q \)-out-of-\( q \) SS scheme to split a message \( m \) into \( q \) random shares, by selecting the first \( q - 1 \) shares \( m_{i} \) at random and then choosing \( m_{q} = m \oplus m_{1} \oplus \ldots \oplus m_{q-1} \).

2. For \( i = 1, \ldots, q \), \( D \) sends \( m_{i} \) to all the parties in a certain subset, say \( \sigma_{i} \subseteq \mathcal{P} \), via secure channels. If any of these parties does not receive the share, she takes a zero bit string as default share.

3. If \( |\sigma_{i}| > 1 \), all pairs of parties in \( \sigma_{i} \) perform a consistency test: they send each other their copies of \( m_{i} \) over secure channels to check if they are equal. If any party finds an inconsistency, she aborts the protocol.

Importantly, abortion proceeds in two steps: the aborting party sends an abortion order to all other parties, and each receiving party resends the order to all the rest. Upon reception of an abortion order, the parties abort. Step two assures that the honest (and/or passively corrupted) parties always abort collectively. Upon non-abortion of Share, Reconstruct runs as follows.

Reconstruct protocol

1. All pairs of parties send each other their shares through authenticated channels.

2. For \( i = 1, \ldots, q \), each party uses MV to reconstruct the share \( m_{i} \), and then obtains \( m = \oplus_{i=1}^{q} m_{i} \).

In general, in order for MV to be well-defined, the output must be set to a default value in case of a tie.
Nevertheless, ties never occur for the adequate choices of the parameters $n$ and $q$ and the subsets $\sigma_i$ we present next.

**Proposition 1.** Let $t$ be the maximum number of corrupted parties, and let $\{T_1, \ldots, T_{\binom{q}{t}}\}$ be any ordered list of all possible combinations of $t$ parties. Under the following settings, (Share, Reconstruct) defines a conditional VSS scheme:

1. $n = 3t+1$, $q = \binom{q}{t}$ and $\sigma_i = P/T_i$ (AC corruption).
2. $n = 2t+2$, $q = n$ and $\sigma_i = P/P_i$ (AN corruption).
3. $n = t+1$, $q = n$ and $\sigma_i = P_i$ (PC corruption).
4. $n = 2$, $q = n$ and $\sigma_i = P_i$ (PN corruption).

What is more, the above settings are optimal in the number of parties.

The reader is referred to Appendix D in the Supplementary Information for a proof of Proposition 1.

Finally, we remark that the above conditional VSS scheme enables secure MPC of linear functions of the shared private input in a very simple way. Let $L(\cdot)$ be the linear function to be computed on $m$. Upon non-abortion of Share, each party applies $L$ to its shares of $m$, in so obtaining shares of $L(m)$. Since this step requires null communication, privacy, conditional commitment and conditional correctness are trivially maintained.

**B. Generation of random bit strings**

Protocol also relies on the possibility to generate unbiased random bit strings (RBS) of a pre-fixed length $L$ among $n$ parties (the CP units in Protocol), when up to $t$ of them are possibly corrupted. In what follows, we describe a RBS generation protocol suitable for the active corruption models, AC and AN, that builds on conditional VSS to safeguard the randomness of its output string (the passive models shall be addressed afterwards).

Let us set the total number of parties, $n$, the total number of shares, $q$, and the subsets of parties, $\sigma_i$, as specified in Proposition 1 for the considered model (AC or AN). The protocol runs as follows.

**RBS generation protocol**

1. For $k = 1, \ldots, t+1$, $P_k$ creates a random $L$-bit string, $R_k$, and distributes it among all $n$ parties (including itself) using Share. If, for some $k$, Share aborts, the RBS generation protocol aborts. If a party receives any share whose length differs from $L$, she aborts.

2. Upon non-abortion of step 1, the parties use Reconstruct to obtain $R_k$ for all $k = 1, \ldots, t+1$. Then, each of them individually calculates $R = \bigoplus_{k=1}^{t+1} R_k$.

**Proposition 2.** The RBS generation protocol outputs a common random $L$-bits string for all honest (and/or passively corrupted) parties.

The reader is referred to Appendix D in the Supplementary Information for a proof of Proposition 2.

Finally, using the standard notion of passivity given in the main text, one can avoid the use of conditional VSS for RBS generation in the passive models (PC and PN). Instead, any given unit can generate the strings directly, and such strings are truly random by assumption.

**V. ACKNOWLEDGEMENTS**

We thank Liu Zhang Chen-Da for useful discussions on verifiable secret sharing and secure multiparty computation. We acknowledge support from the Spanish Ministry of Economy and Competitiveness (MINECO), the Fondo Europeo de Desarrollo Regional (FEDER) through the grant TEC2017-88243-R, and the European Union’s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675662 (project QCALL) for financial support. VZ gratefully acknowledges support from a FPU scholarship from the Spanish Ministry of Education.

---

**Appendix A: Proof of the security claim**

Here, we give a detailed proof of Claim (see the main text), which establishes the security of Protocol when the corrupted QKD modules (CP units) belong to the AC corruption model. Notably, security follows identically for all the alternative models we consider, as it builds on (1) the defining properties of conditional VSS (established for each corruption model of the CP units in the main text), (2) the extractable secret key length (established for each corruption model of the QKD pairs in the main text), and (3) the redundancy of the classical communications of Protocol.

Below, we use asterisks to identify either the well-defined versions reconstructible by honest units of quantities with the conditional commitment property of conditional VSS (that is, the raw keys, the sifted keys, the EC syndromes, the EV and PA hash functions, the corrected keys, the EV tags and the final keys), or unique quantities consistently held by honest units, to which the QKD modules are committed too (that is, the protocol information, the sifting
information, the hypothetical lower bounds computed in the PE step, the error pattern and the secret key length).
Note that the latter are not divided into shares.

1. Correctness

We first prove the correctness established in Claim. Precisely, Claim asserts the $\epsilon_{\text{cor}}$-correctness of the output keys upon non-abort of Protocol (if Protocol aborts, correctness follows trivially). Hence, let us assume Protocol does not abort and refer to conditional commitment (conditional correctness) simply as commitment (correctness) for conciseness.

In the first place, the Share protocol in step 1 guarantees the commitment of the raw keys. In the second place, given the commitment of the raw keys, the commitment of the sifted keys follows from the uniqueness of the sifting information, say $\{\text{sift}^{*}\}_{t=1}^{n_t}$, used by the honest CP units to sift their shares of the raw keys. And, in particular, the uniqueness of $\{\text{sift}^{*}\}_{t=1}^{n_t}$ is trivially enforced by the consistency tests and the redundancy of the communications in step 2. In the third place, given the commitment of the sifted keys, the commitment of the corrected keys is enforced by the uniqueness of the error pattern $\hat{e}^*$ that all honest $\text{CP}_{A_t} \in \sigma^A_t$ apply on their copies of the first share of the sifted key. In turn, the uniqueness of $\hat{e}^*$ follows from the commitment of the syndrome strings $s_A$ and $s_B$, ensured by the commitment of the sifted keys and the redundancy of the communications. Fourthly, the commitment of the EV tags $h_{\text{EV,A}} (h_{\text{EV,B}})$ follows from the commitment of the corrected keys and that of the function $h_{\text{EV}}$ (assured by the RBS generation protocol). In particular, due to the redundancy of the classical communications in step 5, all honest $\text{CP}_{B_t} \in \sigma^B_t$ reach the well-defined copies $h_{\text{EV,A}}$ and $h_{\text{EV,B}}$, where $h_{\text{EV,A}} = h_{\text{EV}} (s_A^*) (h_{\text{EV,B}} = h_{\text{EV}} (s_B^*))$ is the well-defined EV tag reached by all honest $\text{CP}_{A_t}$ ($\text{CP}_{B_t}$), computed on the well-defined corrected key $\hat{s}_A^*$ (sifted key $s_B^*$). Furthermore, from step 5 of Protocol it follows that EV aborts if $h_{\text{EV, A}} \neq h_{\text{EV, B}}$. Conversely, no abortion of the EV step guarantees that $h_{\text{EV, A}} = h_{\text{EV, B}}$. The $\epsilon_{\text{cor}}$-correctness follows from this fact as long as $h_{\text{EV}}$ (well-defined EV function reached by all honest $\text{CP}_{B_t}$ at the RBS generation protocol) is a random 2-universal hash function with output length $[\log_2(2/\epsilon_{\text{cor}})]$ at least [39]. But this is indeed ensured by the correctness of conditional VSS within the RBS generation protocol. To finish with, the commitment of the final keys follows from the commitment of the corrected keys and that of the function $h_{\text{PA}}$. In turn, the latter follows from the commitment of conditional VSS within the RBS generation protocol, in which all honest $\text{CP}_{B_t} \in \sigma^B_t$ select a unique length $l^*$ due to the consistency tests in step 1 and the redundancy of the classical communications. Also note that the commitment of $h_{\text{PA}}$ (plus the redundancy of the communications) guarantees that correctness is not compromised in the final PA step.

Notably, one should not confuse the correctness of conditional VSS (see the Methods section in the main text) with the correctness of the output keys of Protocol. In fact, except from the implicit use of correctness in the RBS generation protocol, only the commitment (but not the correctness or the privacy) of conditional VSS is required to establish the correctness of the final keys.

Lastly, we remark that an authentication error may allow a corrupted CP unit to impersonate an honest one, thus possibly compromising the correctness. Therefore, one must compose the error probability, $\epsilon_{\text{cor}}$, of the EC —which presumes the successful authentication of all the classical communications—with the total error probability of the authentication, $\epsilon_{\text{AU}}$, pre-selected by Alice and Bob. In this way, the overall correctness parameter is

$$\epsilon_{\text{cor}} = \hat{\epsilon}_{\text{cor}} + \epsilon_{\text{AU}}.$$  \hspace{1cm} (A1)

2. Secrecy

In what follows, we prove the secrecy established in Claim. Importantly, the reasoning we present below does not assume a specific QKD protocol, but it applies to a wide variety of them.

We assume again that Protocol does not abort. Given the redundancy of the communications and the uniqueness of $l^*$ (established in the previous section), the RBS generation protocol guarantees that all honest CP units reach a well-defined function $h_{\text{PA}}$ (conditional commitment) and that this function is indeed a 2-universal hash function of length $l^*$ selected at random, as required for PA [31] (conditional correctness). Under these circumstances, the $\epsilon_{\text{sec}}$-secrecy of $k_A^*$ and $k_B^*$ asserted in Claim follows as long as $l^*$ is a valid lower bound on the extractable secret key length, which we prove in what follows.

By applying PA with 2-universal hashing [31], a $\hat{\epsilon}_{\text{sec}}$-secret key can be extracted from the well-defined concatenated sifted key $s_B^* = [s_B^{s_1}, \ldots, s_B^{n_{\text{sift}}}]$ as long as $l^*$ verifies [39]

$$l^* \leq \left[ H_{\text{min}}^c (s_B^{l^*}) - 2 \log_2 \left( \frac{1}{2\epsilon_{\text{PA}}} \right) \right].$$ \hspace{1cm} (A2)
for all \( \hat{\epsilon}_{\text{sec}} \geq \epsilon + \epsilon_{\text{PA}} \), where \( H_{\min}^{\epsilon}(s_B^*|E') \) is the \( \epsilon \)-smooth min-entropy of \( s_B^* \) conditioned on the (possibly quantum) information \( E' \) held by Eve, and \( \epsilon_{\text{PA}} \) is the error probability of PA. To derive a lower bound on \( H_{\min}^{\epsilon}(s_B^*|E') \), it is crucial to note that \( s_B^* \) is statistically uncorrelated to any subset of shares possibly held by any group of corrupted CP units due to the \textit{privacy} of conditional VSS. As a consequence, Eve cannot extract any information from the units beyond the one she learns from the corrupted QKD pairs, the classical communications between labs, and her interaction with the quantum channel. This said, the derivation goes as follows.

Without loss of generality, \( E' \) can be decomposed as \( E' = CE \), where \( C \) denotes the information gained by Eve when she learns the syndrome, \( sy_B^* \), and the EV tag, \( h_{\text{EV,B}}^* \), and \( E \) denotes the information she holds in advance of that. According to Protocol, EC is applied individually on each \( s_A^* \) to reconcile it with the corresponding \( s_B^* \). In particular, the well-defined syndrome information that the honest CP \( \sigma_B^A \) send to the CP \( \sigma_A^B \) splits as \( sy_B^* = [sy^*(s_A^*)] \ldots [sy^*(s_A^{n_q^*})] \). Clearly, all \( n_q \) items in \( sy_B^* \) but the one that comes from the honest QKD pair are possibly known to Eve a priori. If we denote the pair index of the honest QKD pair by “h”, this implies that only \( sy^*(s_B^{h^*}) \) contributes to \( C \), together with the error verification tag \( h_{\text{EV,B}}^* \), whose size is \( |h_{\text{EV,B}}^*| = \lceil \log_2(2/\epsilon_{\text{cor}}) \rceil \). Then, from a chain inequality for smooth entropies [39], \( H_{\min}^{\epsilon}(s_B^*|E') \geq H_{\min}^{\epsilon}(s_B^*|E) - |sy^*(s_B^{h^*})| - \left\lceil \log_2 \left( \frac{2}{\epsilon_{\text{cor}}} \right) \right\rceil \). (A3)

If we use the decomposition \( s_B^* = s_B^{h^*} s_B^{d^*} \) (where \( s_B^{d^*} \) includes all the substrings of \( s_B^* \) that come from dishonest QKD pairs), the following chain rule holds [40]. For all \( \epsilon, \epsilon' \geq 0 \) and for all \( \epsilon \) such that \( \epsilon > 2\epsilon + \epsilon' \),

\[
H_{\min}^{\epsilon}(s_B^*|E) \geq H_{\min}^{\epsilon}(s_B^{h^*}|s_B^{d^*}|E) + H_{\min}^{\epsilon}(s_B^{d^*}|E) - \log_2 \left( \frac{1}{\epsilon - 2\epsilon - \epsilon'} \right),
\]

(A4)

where \( \epsilon \) and \( \epsilon' \) are the smoothing parameters of the corresponding smooth min-entropies [39]. We recall that, following Protocol, Alice and Bob agree on \( \epsilon \) a priori. Also, one can set \( \epsilon' = 0 \) and use the trivial bound \( H_{\min}^{\epsilon}(s_B^{h^*}|E) \geq 0 \) valid for all \( \epsilon' \geq 0 \), as \( s_B^{d^*} \) could be entirely known to Eve. This amounts to say that \( s_B^{d^*} \) is included in \( E \), which further implies that \( H_{\min}^{\epsilon}(s_B^{h^*}|s_B^{d^*}|E) = H_{\min}^{\epsilon}(s_B^{h^*}|E) \). From these two results, inserting Eq. (A4) in Eq. (A3) one finds

\[
H_{\min}^{2\epsilon+\delta}(s_B^*|E') \geq H_{\min}^{\epsilon}(s_B^{h^*}|E) - |sy^*(s_B^{h^*})| - \log_2 \left( \frac{4}{\epsilon_{\text{cor}}\delta} \right),
\]

(A5)

where we use the fact that \( \lceil \log_2(2/\epsilon_{\text{cor}}) \rceil \leq \log_2 (4/\epsilon_{\text{cor}}) \) and also define \( \delta = \epsilon - 2\epsilon \), such that \( \delta > 0 \). Further inserting the previous equation in Eq. (A2), it follows that one can extract

\[
\hat{\epsilon}_{\text{sec}} \leq \left[ H_{\min}^{\epsilon}(s_B^{h^*}|E) - |sy^*(s_B^{h^*})| - \log_2 \left( \frac{1}{\epsilon_{\text{cor}}\epsilon_{\text{PA}}\delta} \right) \right]
\]

(A6)

\[
\hat{\epsilon}_{\text{sec}} \geq 2\epsilon + \delta + \epsilon_{\text{PA}},
\]

(A7)

and \( \delta > 0 \).

Notably, the analysis above is conditioned on the successful authentication of all the classical communications. Setting a common error probability, \( \gamma_{\text{AU}} \), for every authentication, it follows that the total authentication error verifies \( \epsilon_{\text{AU}} \geq (\epsilon + 1)^2(n_q + 1)\gamma_{\text{AU}} \), as explained in the Results section of the main text. Moreover, the overall secrecy parameter is given by

\[
\epsilon_{\text{sec}} = \hat{\epsilon}_{\text{sec}} + \epsilon_{\text{AU}}.
\]

(A8)

Crucially, the honest QKD pair is unknown and thus Eq. (A6) cannot be evaluated in practice. However, it implies a looser but more convenient bound that does not rely on the knowledge of the honest pair by assuming a worst case scenario. Precisely, let \( h_{j^*}^* \) denote the hypothetical lower bound on \( H_{\min}^{\epsilon}(s_B^{j^*}|E) \) determined by \((\text{info}_{A^*}^{j^*}, \text{info}_{B^*}^{j^*})\). As we also did in the main text, we use the term \textit{hypothetical} here because, even though the \( j \)-th QKD pair is committed to a single value \( h_{j^*}^* \) via \((\text{info}_{A^*}^{j^*}, \text{info}_{B^*}^{j^*})\), one cannot assure that such \( h_{j^*}^* \) is a valid lower bound on \( H_{\min}^{\epsilon}(s_B^{j^*}|E) \) unless \( j = h \). Let us further explain this point. On the one hand, if \( j \neq h \), \((\text{info}_{A^*}^{j^*}, \text{info}_{B^*}^{j^*})\) might be unfaithful information —thus, unsuitable for correct parameter estimation (PE)— and indeed \( H_{\min}^{\epsilon}(s_B^{j^*}|E) = 0 \) might hold for all \( \epsilon \). On the other hand, let us focus on the case \( j = h \). In Sec. A.1, we established the \textit{conditional commitment} of all the raw keys, the concatenated sifted keys, the concatenated corrected keys and the final keys. In an identical fashion, since
nq = t_q + 1, the conditional correctness of conditional VSS implies that the pair of raw keys coming from the honest modules QKD_A and QKD_B is generated, sifted, reconciled and subjected to PA correctly by an honest majority of CP units in each lab. Thus, in particular, h^*_e is a valid lower bound on H^{e}_{\min}(s^*_B | E). This said, the more convenient lower bound on H^{e}_{\min}(s^*_B | E) − |sy^*(s^*_B)| that we referred to above is given by \(\min_j \{h^*_e - |sy^*(s^*_B)|\}\), such that the well defined \(I^*\) reached by all honest CP units reads

\[
I^* = \left\lfloor \min_j \left\{ h^*_e - |sy^*(s^*_B)| \right\} - \log_2 \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{PA} \delta} \right) \right\rfloor.
\]

(Appendix B: Authentication scheme)

We consider the authentication scheme presented in [41] and described in [35], based on the construction of Toeplitz matrices using a linear feedback seed register (LFSR).

The sender and the receiver must pre-share a key pool of secret bits, and for every classical message \(m\) to be authenticated, they draw bits from this key pool to construct a LFSR-based Toeplitz matrix \(T\). Let \(\gamma_{\text{AU}} (|m|)\) be a pre-fixed error probability (the length of the message \(m\)), and let \(k = \lfloor \log_2 (2|m|/\gamma_{\text{AU}}) \rfloor\). The construction of \(T\) uses 2\(k\) secret bits and the size of the resulting matrix is \(k \times |m|\). The sender multiplies the matrix \(T\) by the message \(m\) to generate an authentication tag \(t = T \times m\), of \(k\) bits. Then, he encrypts the tag using the one-time-pad, thus consuming another \(k\) secret bits from the key pool. Nevertheless, the encryption of the tag guarantees that the first 2\(k\) bits used for the construction of \(T\) remain secure and can be reallocated in the key pool, in such a way that the net secret key cost of the authentication is \(k\) bits. Finally, the sender transmits both \(m\) and its encrypted tag through the public channel. The receiver calculates its own tag using \(T\) and the received message, and authentication succeeds if this tag matches the one sent by the sender after decrypting it.

(Appendix C: Secret key length in the PN corruption model for the QKD modules)

In this section we derive the secret key length of Protocol under the assumption that the possibly corrupted QKD pairs are passive and non-collaborative (i.e., they belong to the PN corruption model). The use of asterisks to identify well-defined/unique quantities is not required here because, by assumption, the possibly corrupted devices are passive.

As explained in the main text, in this scenario we assume \(n_q = t_q\). This choice allows to fairly compare the performance of the AC and the PN corruption models in terms of the secret key rate, and it means that every QKD pair might be corrupted by an independent eavesdropper, say Eve_j, with \(j = 1, \ldots, n_q\). Let us focus on one of them, say Eve_v. We denote by \(E_v\) the information held by Eve_v prior to the information reconciliation (IR) step of Protocol. Defining, for instance, \(Z_1 = s^*_B\), \(Z_j = s^*_B^{j-1}\) for \(j = 2, \ldots, v\) and \(Z_j = s^*_B^v\) for \(j = v + 1, \ldots, n_q\), the next holds:

1. \(H^{\epsilon_1}_{\min}(Z_1 | E_v) = 0\) for all \(\epsilon_1\).
2. \(H^{\epsilon_j}_{\min}(Z_j | Z_{j-1} \ldots Z_1 E_v) = H^{\epsilon_j}_{\min}(Z_j | E_v)\) for all \(\epsilon_j\) and \(j = 2, \ldots, n_q\).

Therefore, one can apply the simplified version, Eq. (H5), of the generalised chain rule for conditional smooth min-entropies presented in Appendix H. This yields,

\[
H^{(n_q-1)(2\epsilon+\delta)}_{\min}(s_B | E_v) \geq \sum_{j \neq v} H^{\epsilon_j}_{\min}(s^*_B^j | E_v) - \log_2 \left( \frac{1}{\delta n_q - 1} \right),
\]

(C1)

with \(\epsilon, \delta > 0\) and \(n_q \geq 2\). Coming next, we account for the information that Eve_v gains at the IR step. The total information held by Eve_v, a posteriori of IR, can be decomposed as \(E_v' = C_v E_v\), where \(C_v\) denotes the information she learns during IR. Precisely, \(C_v\) contemplates all the syndromes, \(|sy^j(s_B^v)|\), with \(j \neq v\), and the EV tag \(h_{EV,B}\), such that \(|h_{EV,B}| = \lfloor \log_2 (2/\epsilon_{\text{cor}}) \rfloor\). Therefore, from a chain inequality for smooth entropies [39] previously used in Appendix A, we have that

\[
H^{(n_q-1)(2\epsilon+\delta)}_{\min}(s_B | E_v') \geq \sum_{j \neq v} \left\{ H^{\epsilon_j}_{\min}(s^*_B^j | E_v) - |sy^j(s_B^v)| \right\} - \log_2 \left( \frac{4}{\epsilon_{\text{cor}} \delta n_q - 1} \right).
\]

(C2)
By applying PA with 2-universal hashing \[31\], a key that is \(\hat{\epsilon}_{\text{sec}}\)-secret with respect to \(E_v\) can be extracted from \(s_B\), as long as the output length satisfies \[39\]

\[
l \leq \left\lfloor \sum_{j \neq v} \left( H^\epsilon_{\min}(s^j_B | E_v) - |sy(s^j_B)| \right) - \log_2 \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{\text{PA}} \delta n^{-1}} \right) \right\rfloor,
\]

for all

\[
\hat{\epsilon}_{\text{sec}} \geq (n_q - 1)(2\varepsilon + \delta) + \epsilon_{\text{PA}}.
\]

Lastly, composing the total authentication error \(\epsilon_{\text{AU}}\) (pre-agreed by the parties), the overall secrecy parameter reads

\[
\epsilon_{\text{sec}} = \hat{\epsilon}_{\text{sec}} + \epsilon_{\text{AU}},
\]

where we recall that the pre-agreed \(\epsilon_{\text{AU}}\) depends on the corruption model of the CP units, as shown in the Results section of the main text.

Finally, note that Eq. (C3) determines the extractable key length that provides security with respect to the information \(E_v\) held by Eve\(_v\). Nevertheless, one can provide security against all \(\{Eve_v\}_{v=1}^{n_q}\) by taking

\[
l = \left\lfloor \min_{v} \sum_{j \neq v} \left( H^\epsilon_{\min}(s^j_B | E_v) - |sy(s^j_B)| \right) - \log_2 \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{\text{PA}} \delta n^{-1}} \right) \right\rfloor.
\]

**FIG. 3:** Depiction of a setting where \(n_q - 1\) non-collaborative eavesdroppers, \(\{Eve_v\}_{v \neq j}\) (where Eve\(_v\) is the eavesdropper controlling the \(v\)-th QKD pair), attack the quantum communication between QKD\(_A_j\) and QKD\(_B_j\). If the possibly corrupted QKD pairs are passive, by definition Eve\(_j\) has total access to the internal information of QKD\(_A_j\) and QKD\(_B_j\), but the latter do not deviate from Protocol.

To conclude this part, we remark that, according to the PE step of Protocol, every CP\(_B.v \in \sigma^B_1\) computes a hypothetical lower bound \(h^j_\varepsilon\) on the \(\varepsilon\)-smooth min-entropy of \(s^j_B\) conditioned on an eavesdropper’s information up to the PE step. As already discussed in Appendix A, in the AC corruption model, \(h^j_\varepsilon\) does not necessarily pose a lower bound on \(H^\epsilon_{\min}(s^j_B | E)\). However, by assumption, in the PN model the QKD modules deliver faithful protocol information and thus \(h^j_\varepsilon\) is indeed a lower bound on \(H^\epsilon_{\min}(s^j_B | E_v)\) for all \(v \neq j\). Finally, we recall that security against the eavesdroppers that corrupt the QKD modules guarantees security against possible eavesdroppers that corrupt the CP units (irrespectively of their corruption model) due to the *privacy* of conditional VSS (see the Methods section in the main text).

**Appendix D: Proof of propositions 1 and 2**

Here, we give detailed proofs of propositions 1 and 2 in the Methods section of the main text.

1. **Proof of Proposition 1**

Proposition 1 establishes adequate settings under which the pair of protocols (Share, Reconstruct) presented in the Methods section of the main text defines a conditional VSS scheme for every non-mixed corruption model of the parties. Here, we address all four scenarios one by one.
1. **AC corruption** \((t > 0)\). The considered settings are \(n = 3t + 1, q = \binom{n}{t}\) and \(\sigma_i = P_i/T_i\) for \(i = 1, \ldots, q\). Let \(\{T_1, \ldots, T_{n^t}\}\) be an ordered list of all possible combinations of \(t\) parties. Since an honest \(D\) distributes \(m\) according to the previous settings, every combination of \(t\) parties is missing exactly one distinct share. Thus, privacy follows. Let us now assume that, for some \(i\), two honest parties in \(\sigma_i\) receive different copies of \(m_i\) —note that such parties are guaranteed to exist for all \(i\), because \(|\sigma_i| = n - t = 2t + 1 \geq t + 2\) for all \(t > 0\). Then, Share certainly aborts. Conversely, upon non-abortion of Share, all honest parties in each \(\sigma_i\) hold identical copies of \(m_i\) (possibly, a default zero string). What is more, \(|\sigma_i| = 2t + 1\) implies that every \(\sigma_i\) contains a majority of honest parties, such that *conditional commitment* follows from the use of MV in Reconstruct. *Conditional commitment* implies that, upon non-abortion of Share, \(D\) is committed to an input with respect to the honest parties. *Conditional correctness* follows identically as *conditional commitment*, given the fact that an honest \(D\) commits to his actual input value \(m\). This completes the proof.

Note that, in the AC model, \(n > 3t\) is necessary to assure *conditional commitment* by enforcing the success of MV during Reconstruct. In fact, it is known to be a general necessary condition for secure MPC [23–25], such that setting \(n = 3t + 1\) is optimal. What is more, within our conditional VSS scheme, any attempt to reduce the total number of shares, \(q\), comes at the price of increasing the number of parties, \(n\). To see this, let us assume that such improved settings exist, satisfying all three properties of conditional VSS while keeping \(q < \binom{n}{t}\) for a given number of parties, \(n\). On the one hand, privacy implies that every combination of \(t\) parties is missing one share at least. On the other hand, by the pigeonhole principle, \(q < \binom{n}{t}\) implies that at least two distinct combinations of \(t\) parties, say \(T_k\) and \(T_l\), have one common missing share, say \(m_s\), for some \(s = 1, \ldots, q\). Since \(|T_k \cup T_l| \geq t + 1\), it follows that \(|\sigma_s| \leq n - t - 1\), and thus *conditional commitment* requires \(n \geq 3t + 2\) at least, in order for MV to certainly succeed when applied to all copies of \(m_s\).

2. **AN corruption** \((t > 1)\). The considered settings are \(n = 2t + 2, q = n\) and \(\sigma_i = P_i/P\) for \(i = 1, \ldots, q\). Since an honest \(D\) distributes \(m\) according to the previous settings, every party is missing exactly one distinct share. This suffices to establish privacy in a non-collaborative setting. Let us now assume that, for some \(i\), two honest parties in \(\sigma_i\) receive different copies of \(m_i\) —note that such parties are guaranteed to exist for all \(i\), because \(|\sigma_i| = n - t = 2t + 1 \geq t + 2\) for all \(t > 1\). Then, Share certainly aborts. Conversely, upon non-abortion of Share, all honest parties in each \(\sigma_i\) hold identical copies of \(m_i\) (possibly, a default zero string), and since every \(\sigma_i\) contains a majority of honest parties, *conditional commitment* follows from the use of MV in Reconstruct. *Conditional correctness* follows identically as in the AC model.

The optimality of the setting \(n = 2t + 2\) for the pair of protocols (Share, Reconstruct) in the AN model follows from the next lemma.

**Lemma.** If, for some \(i = 1, \ldots, q\), \(|\sigma_i| < 2t + 1\), the pair of protocols (Share, Reconstruct) does not provide a conditional VSS scheme in the AN model.

For the AC model, such an assertion is straightforward. However, at a first glance, it seems reasonable that non-collaboration of the corrupted parties may allow to overcome the restriction that each share is held by an honest majority of parties. This is so because, for any given share \(m_i\), the values declared by any two corrupted parties in \(\sigma_i\) that misbehave during Reconstruct are not expected to coincide, except with the minuscule probability of a random match. Nevertheless, Lemma states that this is not the case, and we prove it in what follows. For this purpose, let us consider that \(D\) is corrupted, and let us assume the worst-case scenario where, for some \(i\), \(\sigma_i\) contains all \(t\) corrupted parties. With a non-negligible probability of success, \(D\) could, for instance, select two distinct versions of the share \(m_i\), say \(m_i^h\) and \(m_i^d\), and deliver \(m_i^h\) \((m_i^d)\) to all honest (dishonest) parties in \(\sigma_i\). Note that this does not necessarily imply the abortion of Share, as the dishonest parties in \(\sigma_i\) can simply declare the copy \(m_i^h\) they receive from the honest ones during the consistency test of \(m_i\). If, in addition, \(|\sigma_i| < 2t + 1\), \(\sigma_i\) does not contain a majority of honest parties and thus *conditional commitment* is compromised, because one cannot assure the consistency of the copies of \(m_i\) reached by all honest parties via MV. This completes the proof.

In summary, \(|\sigma_i| \geq 2t + 1\) is necessary for (Share, Reconstruct) to define a conditional VSS scheme in the AN model. Since, in addition, \(\sigma_i \not\subseteq P\), the requirement \(n \geq 2t + 2\) follows, which means that our setting \(n = 2t + 2\) is optimal. In addition, as in the AC model, direct application of the pigeonhole principle implies that any attempt to reduce the total number of shares, \(q\), comes at the price of increasing the number of parties, \(n\), in order to maintain the defining properties of conditional VSS.

3. **PC corruption** \((t > 0)\). The considered settings are \(n = t + 1, q = n\) and \(\sigma_i = P_i\) for \(i = 1, \ldots, q\). Since an honest \(D\) distributes \(m\) according to the previous settings, every combination of \(t = n - 1\) parties is missing exactly one distinct share. Thus, privacy follows. In addition, *conditional commitment* holds due to
passive corruption of the parties and the fact that $|\sigma_i| = 1$ for all $i$ (which implies that MV trivially succeeds). \textit{Conditional correctness} follows identically as in the previous models.

Note that the optimality of $n = t + 1$ in the PC model is obvious in full generality, and not only within our specific protocols Share and Reconstruct. This is so because setting $n = t$ would compromise \textit{privacy} in the presence of collaborative corrupted parties. Also, as in the previous models, any attempt to reduce the total number of shares, $q$, comes at the price of increasing the number of parties (if one aims to preserve conditional VSS).

Remarkably, as a consequence of considering passive corruption, $|\sigma_i| = 1$ suffices for all $i = 1, \ldots, q$, in which case step 3 of Share vanishes and thus Share never aborts. This being the case, in the PC model, (Share, Reconstruct) with the above settings not only provides a conditional VSS scheme, but also a standard VSS scheme. We also remark that, in the absence of step 3 of Share, no consistency test occurs, which means that VSS reduces to secret sharing (SS) by definition.

4. PN corruption ($t > 1$). The considered settings are $n = 2$, $q = n$ and $\sigma_i = P_i$ for $i = 1, 2$. We clarify that $n = 2$ for all $t$ means that it suffices to select two parties out of all corrupted parties in order for (Share, Reconstruct) to define a conditional VSS scheme. An honest $D$ splits $m$ into two random shares and delivers each of them to a different party. \textit{Privacy} holds because each party is missing one share and they do not collaborate. \textit{Conditional commitment} follows due to passivity and the fact that $|\sigma_i| = 1$ for $i = 1, 2$. \textit{Conditional correctness} follows identically as in the previous models.

The optimality of $n = 2$ and $q = 2$ is trivial, and it is not restricted to our pair of protocols (Share, Reconstruct).

2. Proof of Proposition 2

Let us now prove Proposition 2 of the Methods section of the main text, which asserts that the RBS generation protocol yields a common random $L$-bits string for all honest (and/or passively corrupted) parties. The proposition only applies in the active corruption models.

We address the AC model first, and recall that the settings are selected as prescribed by Proposition 1. Let us assume that the RBS generation protocol does not abort. This implies that Share terminated successfully for all $k = 1, \ldots, t + 1$. In virtue of \textit{conditional commitment}, non-abortion of Share for, say $P_k$, means that all honest parties reach a common string $R_k$ via Reconstruct. Thus, all of them output a common final string $R = \oplus_{k=1}^{t+1} R_k$. What is more, non-abortion implies that $|R_k| = L$ bits for all $k$, such that $|R| = L$ too. Then, Proposition 2 follows if we prove the randomness of $R$. On the one hand, since at least one dealer party is honest, say $P_h$, for some $h = 1, \ldots, t + 1$, \textit{conditional correctness} assures that $R_h$ is random. What is more, in virtue of \textit{privacy}, the information obtained by any set of $t$ or less parties prior to Reconstruct is independent of $R_h$. In particular, given that no Share protocol aborts, all $t$ strings $R_k$ delivered by all $t$ dealers different from $P_h$ are statistically uncorrelated to $R_h$. Therefore, $R$ is indeed random.

In the AN model, in principle, \textit{privacy}, \textit{conditional commitment} and \textit{conditional correctness} of conditional VSS only hold if the corrupted parties do not collaborate with a corrupted dealer (in accordance with the capabilities of a corrupted dealer specified in the main text). However, when considering the RBS generation protocol, $t + 1$ parties act as dealers at the same time, such that collaboration between each dealer and exactly one party (i.e., himself) is unavoidable. Nevertheless, it is straightforward to show that this fact does not compromise the defining properties of conditional VSS, such that the randomness and the length of the output string $R$ are guaranteed using the exact same argument as in the AC model. Notably, since the parties do not collaborate, one could feel tempted to select two dealers instead of $t + 1$, as the strings they would generate would be uncorrelated to each other. Nevertheless, their bitwise XOR would not be necessarily random due to the active character of the two dealers.

Appendix E: Decoy-state MDI-QKD

Here, we combine Protocol (in the main text) with the efficient MDI-QKD scheme proposed in [36]. In this scheme, Alice and Bob use a single intensity for the basis $Z$, devoted to key extraction, and perform PE with the basis $X$ alone, for which they use three different intensities.
1. QKD protocol

The description below assumes that the possibly corrupted devices of any kind do not misbehave from the protocol description. Nevertheless, the scheme is secure against active eavesdroppers (see Claim in the main text). We define the sets $\sigma^A = \{\text{CP}_{A_i}\}_{i=1}^R$ and $\sigma^B = \{\text{CP}_{B_i}\}_{i=1}^R$, where we recall that $R$ is given in Table 1 of the main text for each corruption model of the CP units. For instance, $R = 2t_{c} + 1$ if the AC corruption model is considered.

For $j = 1, \ldots, n_q$, QKD$_A_j$ and QKD$_B_j$ create the pairs of strings $(r^j_A, a^j)$ and $(r^j_B, b^j)$, respectively. While $r^j_A$ and $r^j_B \in [0, 1]^N$ are fully random polarization bit strings, $a^j$ and $b^j \in \{\lambda, \mu, \nu, \omega\}$ are strings of intensities that verify $P[a^j_i = \lambda] = P[b^j_i = \lambda] = q_Z$ and $P[a^j_i = a] = P[b^j_i = a] = q_Xp_a$, for $a \in A = \{\mu, \nu, \omega\}$ and $i = 1, \ldots, N$. On each side, the intensity $\lambda$ determines the use of the basis $Z$, and the basis $X$ is used otherwise.

Let us now focus on, say, the $j$-th QKD pair. For $i$ ranging from 1 to $N$, steps (i) to (iii) are repeated.

(i) **State preparation** QKD$_A_j$ (QKD$_B_j$) prepares a phase-randomized weak coherent pulse (PR-WCP) with intensity $a^j_i$ (in the BB84 state defined by $a^j_i$) and $r^j_A, (b^j_i)$ and $r^j_B_i$.

(ii) **Transmission** QKD$_A_j$ and QKD$_B_j$ send the states to Charles via the quantum channel.

(iii) **Measurement** If Charles is honest, he measures the received signals with a Bell state measurement (BSM).

After the above quantum communication phase, the distributed QKD post-processing starts. Again, we focus on a single QKD pair (the $j$-th one).

1. Distribution of data. Charles sends a $N$-trit string $c^j$ to both modules. If he is honest, this is the string of successes, such that $c^j_i = 1$ ($c^j_i = 2$) if a successful BSM associated to the Bell state $|\psi^+\rangle = 1/\sqrt{2}(|01\rangle + |10\rangle)$ (where $|\psi^-\rangle = 1/\sqrt{2}(|01\rangle - |10\rangle)$) occurred at the $c^j_i$-th round, and $c^j_i = 0$ otherwise. Let $a^j_{i|c^j} (b^j_{i|c^j})$ be the restriction of the intensities string $a^j$ ($b^j$) to the non-zero entries of $c^j$. Also, let $r^j_{A|c^j,X} (r^j_{B|c^j,X})$ and $r^j_{A|c^j,Z} (r^j_{B|c^j,Z})$ be the restrictions of $r^j_A (r^j_B)$ to the non-zero entries of $c^j$ where Alice (Bob) uses basis $X$ and basis $Z$, respectively. QKD$_A_j$ (QKD$_B_j$) communicates $a^j_{i|c^j} (c^j_i, b^j_{i|c^j})$ and $r^j_{A|c^j,X} (r^j_{B|c^j,X})$ to every CP$_A_j \in \sigma^A$ (CP$_B_j \in \sigma^B$), and uses the Share protocol of a conditional VSS scheme to distribute shares of $r^j_{A|c^j,Z} (r^j_{B|c^j,Z})$ to all CP$_A_j$ (CP$_B_j$). All the CP$_A_j \in \sigma^A$ (CP$_B_j \in \sigma^B$) perform a consistency test on $a^j_{i|c^j} (c^j_i, b^j_{i|c^j})$ and $r^j_{A|c^j,X} (r^j_{B|c^j,X})$.

2. Sifting. Every CP$_A_j \in \sigma^A$ sends $a^j_{i|c^j} (c^j_i, b^j_{i|c^j})$ to every CP$_B_j \in \sigma^B$ and each of the latter applies MV. Using $a^j_{i|c^j}$ and $b^j_{i|c^j}$, each CP$_B_j \in \sigma^B$ unit builds the index sets

\[
Z^j = \left\{i | c^j_i \neq 0, a^j_i = b^j_i = \lambda\right\} \quad \text{and} \quad \mathcal{A}^{a,b}_{j} = \left\{i | c^j_i \neq 0, a^j_i = a, b^j_i = b\right\}
\]

for all $a, b \in A$, and checks if the sifting condition $|Z^j| \geq M$ is met for a pre-established threshold value $M$. If it is not met, the CP$_B_j \in \sigma^B$ abort the protocol. In case of not aborting, the CP$_B_j \in \sigma^B$ forward the set $Z^j$ to the rest of Bob’s units, which apply MV. All together, the CP$_B_j$ perform a RBS generation protocol to select a random subset $Z' \subseteq Z_j$, of size $M$. Then, Bob’s units proceed to the sifting. Precisely, every CP$_B_j$ builds its shares of the sifting key $s^j_B = r^j_B|Z'_j$ from those of $r^j_B|Z_j$ (discarding the data external to $Z'_j$).

3. Parameter estimation. For each pair $a, b \in A$, every CP$_B_j \in \sigma^B$ builds the PE strings $r^j_{A|A^{a,b}_j}$ and $r^j_{A|A^{a,b}_j}$ from the respective strings $r^j_{A|c^j,X}$ and $r^j_{A|c^j,X}$, discarding the data external to $\mathcal{X}^{a,b}_{j}$. Also, every CP$_B_j \in \sigma^B$ performs the required bit flips on the strings $r^j_{A|A^{a,b}_j}$ depending on the list of successes, $c^j$, declared by QKD$_B_j$ (see [42, 43]). In this way, $r^j_{A|\mathcal{X}^{a,b}_j}$ and $r^j_{B|\mathcal{X}^{a,b}_j}$ are properly correlated for all $a, b$. Then, each of them computes the numbers of bit errors

\[
eq 1} \sum_{k=1}^{M} r^j_{A_k|\mathcal{X}^{a,b}_j} \oplus r^j_{B_k|\mathcal{X}^{a,b}_j},
\]

(E2)
where $r^j_{A_i} \mid \chi_{A_i}^{a,b}$ denotes the $k$-th bit of the corresponding string. Using $|Z_j|$ and the different $|\chi_{A_i}^{a,b}|$ and $c^j_{a,b}$, every CP $B_{j'} \in \sigma^B_1$ computes a lower bound on the number $n^j_{11,Z}$ of single-photon successes in $Z_{j'}$ and an upper bound on the single-photon phase-error rate $\phi^j_{11,Z}$ associated to the single-photon successes in $Z_{j'}$.

The above steps 1 to 3 are performed for all $j = 1, \ldots, n_q$. At this stage, every CP $B_{j'} \in \sigma^B_1$ derives a lower bound $l$ (given in the next section) on the secret key length that can be extracted from the concatenated sifted key $s_B = s_B^1 \ldots s_B^n$ via PA. If a CP $B_{j'} \in \sigma^B_1$ finds $l \leq 0$, it aborts the protocol. Importantly, all CP $B_{j'} \in \sigma^B_1$ hold copies of the first share of $s_B$, so each of them performs the relevant bit flips (see [42, 43]) on its copy of this share to correctly correlate $s_A$ (defined below) and $s_B$.

4. RBS generation. If the protocol does not abort, every CP $B_{j'} \in \sigma^B_1$ forwards $l$ to the rest of Bob’s units, which apply MV. All CP $B_{j'}$ perform a RBS generation protocol to randomly select two 2-universal hash functions $h_{EV}$ and $h_{PA}$, respectively devoted to error verification (EV) and PA. Following [35], if Toeplitz matrices are used for this purpose, $2[\log_2(2/\epsilon_{cor})] (Mn_q + l - 1)$ bits are required to specify $h_{EV}$ ($h_{PA}$).

5. Information reconciliation. Every CP $B_{j'}$ computes its shares of (1) the concatenated syndromes string $s_{Bj'} = sy(s_B^1) \ldots sy(s_B^n)$ and (2) the EV tag $h_{EV,B} = h_{EV}(s_B)$. All together, the CP $B_{j'}$ reconstruct $s_{Bj'}$ and $h_{EV,B}$ via the Reconstruct protocol of a conditional VSS scheme (see the Methods section in the main text).

Each CP $B_{j'} \in \omega^B_1$ sends the following items to every CP $A_{j'} \in \omega^A_1$:

1. The string $s_{Zj'} = s_{Zj1} \ldots s_{Zjn}$, where $s_{Zj1}$ specifies, say, the positions in $r^j_{Ai} \mid c_{j,Z}$ that contribute to $Z_{j'}$.
2. The syndrome information $s_{YB} = s_{YB1} \ldots s_{YBn}$, together with the description of $h_{EV}$ and the EV tag $h_{EV}(s_B)$.
3. The description of $h_{PA}$.

Each CP $A_{j'} \in \omega^A_1$ decides on all three items via MV and communicate $s_{Zj'}$, $h_{EV}$ and $h_{PA}$ to the rest of Alice’s units, which apply MV too. Then, they proceed as follows. Using $s_{Zj'}$, all CP $A_{j'}$ shrink their shares of $r_{A} \mid c_{j,Z} = r_{A1} \mid c_{j1,Z} \ldots r_{An} \mid c_{jn,Z}$ into shares of $s_A = s_A^1 \ldots s_A^n$, where $s_A^j = r_{A}^j \mid x_{j'}$. All the CP $A_{j'}$ compute shares of $sy(s_A)$ from those of $s_A$ and then perform the Reconstruct protocol of a conditional VSS scheme to agree on $sy(s_A)$. Coming next, the CP $A_{j'} \in \omega^A_1$ compute the error pattern $\hat{e}$ from $sy(s_{Bj})$ and $sy(s_{A})$ and update the first share of $s_A$ XOR-ing it with $\hat{e}$ (i.e., key reconciliation is achieved by acting on a single share). We denote the corrected key by $\hat{s}_{A} = s_{A} \oplus \hat{e}$. Using $h_{EV}$, all the CP $A_{j'}$ compute their shares of $h_{EV}(\hat{s}_{A})$ and reconstruct it via the Reconstruct protocol of a conditional VSS scheme. Then, each CP $A_{j'} \in \omega^A_1$ checks that $h_{EV}(\hat{s}_{A}) = h_{EV}(s_{Bj})$. Otherwise, it aborts the protocol.

6. Privacy amplification. In case of not aborting, all the CP $A_{j'}$ compute their shares of Alice’s final key $S_A = h_{PA}(\hat{s}_{A})$. Similarly, if no abortion is notified, all the CP $B_{j'}$ compute their shares of Bob’s final key $S_B = h_{PA}(s_{Bj})$.

2. Secret key length formula

a. AC, AN and PC corruption models for the QKD modules

In this section, we particularise the extractable key length (Eq. (A9)) for the decoy-state MDI-QKD protocol of Sec. E 1. As seen in the main text, this formula is tight within the AC, AN and PC corruption models for the QKD modules, and we maintain the asterisks to emphasize that we refer to well-defined/unique quantities consistently held by honest (and/or passively corrupted) units. To evaluate Eq. (A9), it suffices to derive the explicit formula of $h_{\varepsilon}^{L_s}$. Assuming perfect state preparation, the entropic uncertainty relation [44] gives

$$H_{\min}^{s_B^*}(E) \geq n_{11,Z}^{h,L_s} \left[1 - h \left(\frac{h_{11,Z}^*}{\phi_{11,Z}^*}\right)\right],$$

(E3)

where $h(\cdot)$ is the binary entropy function, $n_{11,Z}^{h,L_s}$ stands for a lower bound on $n_{11,Z}^{h}$ and $\phi_{11,Z}$ stands for an upper bound on $\phi_{11,Z}$, $n_{11,Z}^{j}$ and $\phi_{11,Z}^{j}$ being defined in Sec. E 1. From the definition of the smooth min-entropies, it follows that $\varepsilon$ is upper-bounded by the sum of the error probabilities of the estimates of $n_{11,Z}^{h,L_s}$ and $\phi_{11,Z}^{h,L_s}$.

Eq. (E3) implies that, for all $j = 1, \ldots, n_q$, one should define

$$h_{\varepsilon}^{j,*} = n_{11,Z}^{j} \left[1 - h \left(\frac{\phi_{11,Z}^{j}}{\phi_{11,Z}^{*}}\right)\right],$$

(E4)
which indeed determines a lower bound on $H_{\min}^\ell(s_B^i|E)$ if the $j$-th QKD pair delivers faithful protocol information. Putting it all together, the extractable key length of the protocol in Sec. E1 reads

$$l^* = \min \left\{ \ell^j_{11,Z} \left[ 1 - h(\phi_{11,Z}^{j,U}) \right] - |\text{sy}(s_B^i)| \right\} - \log_2 \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{PA} \delta^q} \right), \tag{E5}$$

where we recall that $|\text{sy}(s_B^i)|$ is the size of the $j$-th EC syndrome, $\epsilon_{\text{cor}}$ is the correctness parameter, $\epsilon_{PA}$ is the error probability of the privacy amplification and $\delta > 0$. Also, as shown in Sec. A, the above key length is $\epsilon_{\text{sec}}$-secret for all

$$\epsilon_{\text{sec}} = \hat{\epsilon}_{\text{sec}} + \epsilon_{\text{AU}}, \tag{E6}$$

where $\hat{\epsilon}_{\text{sec}} \geq 2\epsilon + \delta + \epsilon_{PA}$ and $\epsilon_{\text{AU}}$ is the error probability of the authentication, which is pre-determined by the parties and depends on the corruption model of the CP units (see the Results section in the main text).

Explicit expressions of $n_{11,Z}^{j,L}$ and $\phi_{11,Z}^{j,U}$ in terms of the observables of the protocol are given in the next section, together with an upper bound on the smooth-parameter $\varepsilon$.

b. PN corruption model for the QKD modules

Similarly, the tighter secret key length formula valid for the PN corruption model is

$$l \leq \min \left\{ n_{11,Z}^{j,L} \left[ 1 - h(\phi_{11,Z}^{j,U}) \right] - |\text{sy}(s_B^i)| \right\} - \log_2 \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{PA} \delta^q - 1} \right), \tag{E7}$$

with $\epsilon_{\text{sec}} = \hat{\epsilon}_{\text{sec}} + \epsilon_{\text{AU}}$ and $\hat{\epsilon}_{\text{sec}} \geq (n_q - 1)(2\epsilon + \delta) + \epsilon_{PA}$. Again, $\epsilon_{\text{AU}}$ is pre-determined by Alice and Bob, and depends on the corruption model of the CP units as established in the main text.

3. Parameter estimation

Here, we compute the bounds $n_{11,Z}^{j,L}$ ($n_{11,Z}^{j,U}$) and $\phi_{11,Z}^{j,L}$ ($\phi_{11,Z}^{j,U}$) that enter the secret key length, Eq. (E5) (Eq. (E7)). Since the analysis below is common for every $j = 1, \ldots, n_q$, for simplicity of notation we drop the QKD pair index $j$ and refer to any of the QKD pairs. Also, we drop the asterisks for readability. PE is divided into two steps. In a first step, we use the observables of the protocol to calculate bounds on the number $S_{11,X}$ ($E_{11,X}$) of single-photon successes (errors) in $X = \cup_{a,b} X_{a,b}$. For this purpose, we adopt the decoy-state bounds presented in [43], although a slightly simpler technique is used to estimate the expected sizes of the sets $X_{a,b}$ given their realisations (see Sec. G). In a second step, since PE is only performed with the basis X data in the protocol (see Sec. E1), we use basis-indistinguishability arguments for the single-photon contributions and standard results from large deviation theory to compute a lower bound on $n_{11,Z}$ and an upper bound on $\phi_{11,Z}$ given the former bounds on $S_{11,X}$ and $E_{11,X}$.

a. $S_{11,X}^{L}, E_{11,X}^{U}$

Let $A = \{\mu, \nu, \omega\}$ be the set of intensities that the parties use when they select the basis X, such that $\mu > \nu > \omega$, and let $p_{\mu}, p_{\nu}$ and $p_{\omega}$ be the corresponding probabilities. Also, let us introduce a list $\mathcal{V} = \{(v_i, v_i')\}_{i=1}^9$ of pairs of vectors given by:

$$\begin{align*}
(v_1, v_1') &= (\mu, \mu, \mu, \mu, \mu, \mu, \mu, \mu, \mu), \quad (v_2, v_2') = (\nu, \nu, \mu, \nu, \mu, \nu, \nu, \mu, \nu), \\
(v_3, v_3') &= (\mu, \nu, \nu, \nu, \nu), \quad (v_4, v_4') = (\nu, \nu, \nu, \nu, \nu, \nu, \nu, \nu, \nu), \\
(v_5, v_5') &= (\mu, \mu, \mu, \mu, \mu, \mu, \mu, \mu, \mu), \quad (v_6, v_6') = (\nu, \nu, \mu, \nu, \mu, \nu, \nu, \mu, \nu), \\
(v_7, v_7') &= (\mu, \nu, \nu, \nu, \nu), \quad (v_8, v_8') = (\mu, \mu, \mu, \mu, \mu, \mu, \mu, \mu, \mu), \quad (v_9, v_9') = (\nu, \nu, \nu, \nu, \nu, \nu, \nu, \nu, \nu). \tag{E8}
\end{align*}$$

Then, the lower bound $S_{11,X}^{L}$ is given by [43]

$$S_{11,X}^{L} = \max_{(v_i, v_i') \in \mathcal{V}} \left\{ \frac{\tau_{11}}{\epsilon_{11}} \left( J_{v_i, v_i'} - \Gamma_{v_i, v_i'} \right) \right\} \tag{E9}$$
except with probability at most $\epsilon_{11,X} = \sum_{a,b} \epsilon_{a,b}$, for a series of error terms $\{\epsilon_{a,b}\}_{a,b \in A}$ specified by the parties, and some specific quantities $\tau_{11}$, $c_{11}$, $J_{uv'}$, and $\Gamma_{uv'}$ that we define in what follows. First,

$$\tau_{nm} = \frac{1}{n! m!} \sum_{a,b \in A} e^{-(a+b) \alpha_n b^m} p_{a,b, X},$$

(E10)

where $p_{a,b, X}$ stands for the probability of a basis $X$ coincidence with intensity settings $a \in A$ for Alice and $b \in A$ for Bob. That is, $p_{a,b, X} = p_a p_b q_{X}^2$. Regarding $c_{11}$, $J_{uv'}$, and $\Gamma_{uv'}$, we distinguish two cases depending on the sign of $(a_0 + a_1)/(a'_0 + a'_1) - (b_0 + b_1)/(b'_0 + b'_1)$, where for convenience we use the generic notation $v = [a_0, a_1, b_0, b_1]$ and $v' = [a'_0, a'_1, b'_0, b'_1]$ for the pairs of vectors in the list $V$.

**Case 1:** $(a_0 + a_1)/(a'_0 + a'_1) > (b_0 + b_1)/(b'_0 + b'_1)$.

In this case, the definitions are

$$c_{nm} = (b_0^2 - b_1^2)(a_0 - a_1)(a'_0 - a'_1)(b'_0 - b'_1)(a'_0 - a'_1)(a_0 - a_1)(b_0 - b_1)(a'_0 - a'_1)(b'_0 - b'_1),$$

(E11)

$$J_{uv'} = (b_0^2 - b_1^2)(a_0 - a_1)G_{uv'} - (b'_0^2 - b'_1^2)(a'_0 - a'_1)G_v,$$

(E12)

with

$$G_v = |\tilde{\mathcal{X}}^{a_0,b_0}| + |\tilde{\mathcal{X}}^{a_1,b_1}| - |\tilde{\mathcal{X}}^{a_0,b_1}| - |\tilde{\mathcal{X}}^{a_1,b_0}|,$$

(E13)

$$G_{uv'} = |\tilde{\mathcal{X}}^{a'_0,b'_0}| + |\tilde{\mathcal{X}}^{a'_1,b'_1}| - |\tilde{\mathcal{X}}^{a'_0,b'_1}| - |\tilde{\mathcal{X}}^{a'_1,b'_0}|$$

(E14)

and $|\tilde{\mathcal{X}}^{a,b}| = e^{a+b}|\mathcal{X}^{a,b}|/p_{a,b, X}$.

Lastly,

$$\Gamma_{uv'} = (b_0^2 - b_1^2)(a_0 - a_1)(\hat{\Gamma}_{a'_0,b'_0} + \hat{\Gamma}_{a'_1,b'_1} + \hat{\Gamma}_{a_0,b_0} + \hat{\Gamma}_{a_1,b_1}) + (b'_0^2 - b'_1^2)(a'_0 - a'_1)(\hat{\Gamma}_{a_0,b_0} + \hat{\Gamma}_{a_1,b_1} + \hat{\Gamma}_{a_0,b_1} + \hat{\Gamma}_{a_1,b_0}),$$

(E15)

where $\hat{\Gamma}_{a,b} = e^{a+b}\hat{\Delta}(\mathcal{X}^{a,b}, \epsilon_{a,b})/p_{a,b, X}$ and $\Gamma_{a,b} = e^{a+b}\Delta(\mathcal{X}^{a,b}, \epsilon_{a,b})/p_{a,b, X}$. The functions $\hat{\Delta}(x, y)$ and $\Delta(x, y)$ are defined in Sec. G. There, we explain the technique we use to relate the observed set sizes, $|\mathcal{X}^{a,b}|$, with their expected values, in order to set statistical bounds on the latter.

**Case 2:** $(a_0 + a_1)/(a'_0 + a'_1) \leq (b_0 + b_1)/(b'_0 + b'_1)$.

For this case,

$$c_{nm} = (a_0 - a_1)(b_0 - b_1)(a'_0 - a'_1)(b'_0 - b'_1)(a_0 + a_1 - a'_0 - a'_1),$$

(E16)

$$J_{uv'} = (a_0^2 - a_1^2)(b_0 - b_1)G_{uv'} - (a'_0^2 - a'_1^2)(b'_0 - b'_1)G_v,$$

(E17)

and

$$\Gamma_{uv'} = (a_0^2 - a_1^2)(b_0 - b_1)(\hat{\Gamma}_{a'_0,b'_0} + \hat{\Gamma}_{a'_1,b'_1} + \hat{\Gamma}_{a_0,b_0} + \hat{\Gamma}_{a_1,b_1}) + (a'_0^2 - a'_1^2)(b'_0 - b'_1)(\hat{\Gamma}_{a_0,b_0} + \hat{\Gamma}_{a_1,b_1} + \hat{\Gamma}_{a_0,b_1} + \hat{\Gamma}_{a_1,b_0}),$$

(E18)

where the definitions of $G_v$, $G_{uv'}$, $\hat{\Gamma}_{a,b}$ and $\Gamma_{a,b}$ are the same as in Case 1.

Coming next, we compute an upper bound $E_{11,X}^U$ on $E_{11,X}$. For this, let us introduce the list of vectors $W = \{[a_0, a_1, b_0, b_1] | a_0 > a_1, b_0 > b_1, a_0, a_1, b_0, b_1 \in A\}$. Then, the upper bound $E_{11,X}^U$ is given by [43]

$$E_{11,X}^U = \max_{v \in W} \left\{ \frac{\tau_{11}(F_v - \Gamma_v)}{(a_0 - a_1)(b_0 - b_1)} \right\},$$

(E19)
except with probability at most $\epsilon'_{11} = \sum_{a,b} \epsilon'_{a,b}$, for a series of error terms $\{\epsilon'_{a,b}\}_{a,b \in X}$ specified by the parties and some specific quantities $F_v$ and $\Gamma_v$ that we define in what follows:

$$ F_v = \tilde{e}^{a_0,b_0} + \tilde{e}^{a_1,b_1} - \tilde{e}^{a_0,b_1} - \tilde{e}^{a_1,b_0} \tag{E20} $$

with $\tilde{e}^{a,b} = e^{a,b} e_{a,b} / p_{a,b,x}$, and

$$ \Gamma_v = -\Gamma'_{a_0,b_0} - \Gamma'_{a_1,b_1} - \hat{\Gamma}'_{a_0,b_1} - \hat{\Gamma}'_{a_1,b_0} \tag{E21} $$

with $\Gamma'_{a,b} = e^{a,b} \Delta(e_{a,b}, e'_{a,b}) / p_{a,b,x}$ and $\hat{\Gamma}'_{a,b} = e^{a,b} \hat{\Delta}(e_{a,b}, e'_{a,b}) / p_{a,b,x}$. Also, we remind the reader that, for every $a, b \in A$, $e_{a,b}$ is the observed number of bit errors in the set $X^{a,b}$.

\begin{center}
\textit{b. $n_{11,Z}$, $\phi_{11,Z}$}
\end{center}

Let $N_{11,Z}$ ($N_{11,X}$) be the number of rounds where both Alice and Bob sent single photons and used the basis $Z$ ($X$). Of course, $N_{11} = N_{11,Z} + N_{11,X}$ is the overall number of rounds where a basis match occurred and both parties sent single photons. In the absence of state preparation flaws, the quantum states sent by Alice and Bob that contain single photons on both sides are basis independent, meaning that Eve cannot distinguish in which basis they are prepared. As a consequence, the probability that Charles declares a successful BSM cannot depend on the basis choice. Thus, given the number $S_{11,X}$ of rounds where both parties sent single photons in the basis $X$ and Charles declared a successful BSM, one can estimate the corresponding number for the basis $Z$, $S_{11,Z}$, via Serfling’s inequality [45]. Of course, this requires the knowledge of $N_{11,Z}$ and $N_{11,X}$ as well. Precisely,

$$ P \left( S_{11,Z} \leq N_{11,Z} \left( \frac{S_{11,X}}{N_{11,X}} \right) - (N_{11,Z} + N_{11,X}) \times \Upsilon \left( N_{11,Z}, N_{11,X}, \epsilon \right) \right) \leq \epsilon \tag{E22} $$

holds for any $0 < \epsilon < 1$ if we choose the deviation term $\Upsilon(N_{11,Z}, N_{11,X}, \epsilon)$ to be defined by the function

$$ \Upsilon(x, y, z) = \sqrt{(x + 1) \ln(z^{-1})/(2y(x + y))}. \tag{E23} $$

For simplicity, we shall set a common error probability, $\epsilon = \epsilon_S$, for each usage of Serfling’s inequality in this section.

Note that, as the quantities $N_{11,Z}$, $N_{11,X}$, and $S_{11,X}$ are not known, one should derive statistical bounds on them and assume the worst-case scenario, i.e., the one that minimises the value of $S_{11,Z}$. For the first two quantities one can use the standard Chernoff bound [46], as their expected values are known to be $\mu_{11,Z} = E[N_{11,Z}] = N_{11}^2 \rho_{11,Z}^2$ and $\mu_{11,X} = E[N_{11,X}] = N_{11}^2 \rho_{11,X}^2 (p_0 p_{11} + p_0 p_{11} + p_0 p_{11})^2$, where $p_{11}$ stands for the poissonian photon-number distribution with mean value $a$. Importantly, these expected values do not rely on the assumption of a particular channel model, but only on Alice’s and Bob’s state preparation process. Regarding, for instance, $N_{11,Z}$, we have that $P \left( N_{11,Z} > N_{11,Z}^U \right) < \epsilon'$ and $P \left( N_{11,Z} < N_{11,Z}^L \right) < \epsilon''$ respectively hold for any $\epsilon', \epsilon'' \in (0, 1)$ if we set

$$ N_{11,Z}^L = \min \left\{ \left[ \mu_Z + \Delta_L(\mu_Z, \epsilon') \right], N \right\} \quad \text{and} \quad N_{11,Z}^U = \max \left\{ \left[ \mu_Z - \Delta_L(\mu_Z, \epsilon'') \right], 0 \right\}, \tag{E24} $$

where the deviation functions are given by [46]

$$ \Delta_L(x, y) = \frac{\ln y - 1}{2} \left( 1 + \sqrt{1 + \frac{8x}{\ln y - 1}} \right) \quad \text{and} \quad \Delta_L(x, y) = \sqrt{2x \ln y^{-1}}. \tag{E25} $$

As usual, the superscript “L” (“U”) stands for “lower” (“upper”) bound, and the bounds on $N_{11,X}$ are obtained substituting $\mu_Z$ by $\mu_X$ in Eq. (E24). For simplicity, we shall set a common error probability, $\epsilon_C$, for each usage of the Chernoff bound, as we already did for Serfling’s inequality. In particular, we set $\epsilon' = \epsilon'' = \epsilon_C$.

Regarding $S_{11,X}$, a lower bound $S_{11,X}^L$ was already derived in the first part of this appendix, and the corresponding error probability is denoted by $\epsilon_{11,X}$. Coming next, we update the claim of Eq. (E22) by replacing $N_{11,Z}$, $N_{11,X}$ and $S_{11,X}$ with the appropriate bounds minimising $S_{11,Z}$, and by adding the corresponding error terms on the right-hand side. This yields $P \left( S_{11,Z} \leq S_{11,Z}^L \right) \leq \epsilon_S + \epsilon_{11,X} + 2\epsilon_C$, for

$$ S_{11,Z}^L = \max \left\{ \left[ N_{11,Z}^L \left( \frac{S_{11,X}^L}{N_{11,X}} \right) - (N_{11,Z}^L + N_{11,X}^U) \times \Upsilon \left( N_{11,Z}^L, N_{11,X}^U, \epsilon_S \right) \right], 0 \right\}. \tag{E26} $$
Finally, using Serfling’s inequality [45] one can easily relate the lower bound on the number \( n_{11, Z} \) of single-photon successes in the random sample \( Z' \subset Z \), with the lower bound on the number \( S_{11, Z} \) of single-photon successes in the original set \( Z \) (see Sec. E 1). Already incorporating Eq. (E26), it follows that \( P \left( n_{11, Z} \leq n_{11, Z}^L \right) \leq 2 \varepsilon_S + \varepsilon_{11, X} + 2 \varepsilon_C \) for

\[
\begin{align*}
\left( n_{11, Z}^L = \max \left\{ \left[ M \left( \frac{S_{11, Z}}{|Z|} - \Lambda (|Z|, M, \varepsilon_S) \right) \right], 0 \right\},
\right.
\end{align*}
\]

where \( \Lambda (x, y, z) = \sqrt{(x - y + 1) \ln(z^{-1})/(2xy)} \) and \( M \) is again the size of \( Z' \), which defines the post-processing block size (i.e., the size of the sifted keys).

In the derivation above, we used a basis indistinguishability argument to relate the ratio \( S_{11, Z}/N_{11, Z} \) to the ratio \( S_{11, X}/N_{11, X} \) via Serfling’s inequality [45]. The same argument also relates the ratio \( \varepsilon_{11, Z}/n_{11, Z} \) to the ratio \( E_{11, X}/S_{11, X} \), where \( \varepsilon_{11, Z} \) denotes the number of single-photon phase errors (bit errors) in the rounds indexed by \( Z' \) \( (X = \cup_{a,b} X_{a,b}) \). Precisely,

\[
P \left( \varepsilon_{11, Z} \geq \frac{n_{11, Z}}{S_{11, X}} \left( \frac{E_{11, X}}{S_{11, X}} \right) + \left( S_{11, X} + n_{11, Z} \right) \times Y \left( n_{11, Z}, S_{11, X}, \varepsilon_S \right) \right) \leq \varepsilon_S
\]

holds, where the deviation function \( Y(x, y, z) \) is defined in Eq. (E22). Again, the quantities \( n_{11, Z}, S_{11, X} \) and \( E_{11, X} \) are not known, in such a way that adequate bounds should be used instead. On the one side, a lower bound on \( n_{11, Z} \) was presented in the previous subsection, and the relevant bounds on \( S_{11, X} \) and \( E_{11, X} \) were derived in the first part of the appendix. Using these bounds and their respective error probabilities, one can update the claim of Eq. (E28) as \( P \left( \varepsilon_{11, Z} \geq \frac{n_{11, Z}}{S_{11, X}} \right) \leq 3 \varepsilon_S + \varepsilon_{11, X} + \varepsilon_{11, X} + 2 \varepsilon_C \), where

\[
\begin{align*}
\varepsilon_{11, Z}^U = \min \left\{ n_{11, Z}^L \left( \frac{E_{11, X}}{S_{11, X}} \right) + \left( S_{11, X} + n_{11, Z} \right) \times Y \left( n_{11, Z}, S_{11, X}, \varepsilon_S \right) \right\}, \left( n_{11, Z}^L \right) \right\}
\end{align*}
\]

To finish with, note that the single-photon phase error rate is, by definition, given by \( \phi_{11, Z} = \varepsilon_{11, Z}/n_{11, Z} \). Thus, it follows that \( P \left( \phi_{11, Z} \geq \frac{n_{11, Z}}{S_{11, X}} \right) \leq 3 \varepsilon_S + \varepsilon_{11, X} + \varepsilon_{11, X} + 2 \varepsilon_C \) for

\[
\phi_{11, Z}^U = \frac{n_{11, Z}}{n_{11, Z}^L}
\]

where \( n_{11, Z} \) is given by Eq. (E27) and \( n_{11, Z}^L \) is given by Eq. (E29).

From the above PE procedure, it follows that the smooth parameter \( \varepsilon \) (see Sec. E 2) is upper-bounded as

\[
\varepsilon \leq 3 \varepsilon_S + \varepsilon_{11, X} + \varepsilon_{11, X} + 2 \varepsilon_C.
\]

4. Authentication cost

Here, we estimate the secret key bits consumed for authentication purposes in the MDI-QKD protocol of Sec. E 1, assuming the authentication scheme presented in Sec. B of this Supplementary Information.

In general, as explained in the main text, the authentication cost reads

\[
l_{AU} = R^2 \times |k|
\]

for every corruption model of the CP units, where \( R \) is the common size of the sets \( \sigma_1^A \) and \( \sigma_1^B \) (specified in Table 1 of the main text for each model), and \( |k| \) is the number of secret bits of each key pool. Let us now estimate \( |k| \). For this purpose, let \( \left| m_A \right|_{j=1}^n \) and \( \left| m_B \right| \) be the lengths of the messages respectively sent by Alice and Bob in the MDI-QKD protocol of Sec. E 1 (where \( n_q \) is the required number of QKD pairs, which depends on the corruption model of the QKD modules). Then, it is required that

\[
|k| \geq \sum_{j=1}^n \left[ \log_2 \left( \frac{2 \left| m_A \right|}{\gamma_{AU}} \right) \right] + \left[ \log_2 \left( \frac{2 \left| m_B \right|}{\gamma_{AU}} \right) \right]
\]
secret key bits at least, assuming a common error probability, $\gamma_{AU}$, for every authenticated message. This is so because the right-hand side of Eq. (E33) defines the amount of secret bits that are consumed per pair (CP$_A$, CP$_B$) with CP$_A$ $\in$ $\sigma^A_1$ and CP$_B$ $\in$ $\sigma^B_1$, for the encryption of the authentication tags. Following the protocol description of Sec. E1, we have

$$\begin{align*}
|m^A_A| &= |a^l_{c,\ell}| + |r^A_{\ell,c,X}|, \\
|m^B_B| &= |s_{Z,\ell}| + |s_{y_{SB}}| + |h_{EV}(s_{SB})| + |h_{EV} \text{ description}| + |h_{PA} \text{ description}|,
\end{align*}$$

where the expected sizes of $a^l_{c,\ell}$ and $r^A_{\ell,c,X}$ for a typical channel model are given in Sec. E6, $|s_{Z,\ell}| = \sum_{j=1}^{n_q} |r^A_{\ell,c,Z}|$ (the expected sizes of all $r^A_{\ell,c,Z}$ being given in Sec. E6 too), the size of the syndrome $|s_{y_{SB}}|$ depends on the EC protocol (and a typical model is given in the Discussion of the main text), $|h_{EV}(s_{SB})| = \lceil \log_2(2/\varepsilon_{cor}) \rceil$ bits, $|h_{EV} \text{ description}| = 2\lceil \log_2(2/\varepsilon_{cor}) \rceil$ bits and $|h_{PA} \text{ description}| = M n_q + l - 1$ bits, $l$ denoting the extractable key length, given by Eq. (E5) (Eq. (E7)) for the AC, AN and PC corruption models (PN corruption model) for the QKD modules.

To finish with, we recall that the pre-agreed total error probability of the authentication, $\epsilon_{AU}$, is related to $\gamma_{AU}$ via $\epsilon_{AU} \geq (t_c + 1)^2 \times (n_q + 1) \gamma_{AU}$ for active corruption of the CP units, and via $\epsilon_{AU} \geq (n_q + 1) \gamma_{AU}$ for passive corruption, where we remind the reader that $t_c$ is the threshold number of corrupted CP units per party.

5. Calculation of $N$ and $E_{tol}$ for the simulations

Here, we derive proper values for the number of transmission rounds per QKD pair, $N$, and for the threshold bit error rate of the EC protocol, $E_{tol}$, based on respective restrictions on the abortion probabilities of the sifting step and the error verification step. The analysis relies on a typical channel model presented in Sec. E6.

a. $N$

Let us impose a common abortion probability $\gamma_{sift}/n_q$ for each sifting step ($n_q$ of them in total). That is, we demand that $P(|Z_j| < M) \leq \gamma_{sift}/n_q$ for all $j = 1, \ldots, n_q$, where $|Z_j|$ is the set of detection events when both parties use basis $Z$ and $M$ is the pre-specified size of the sifted keys (i.e., the block size). Using the Chernoff’s inequality [46], this condition is met if we set the number of signals transmitted per module in each QKD pair to $N = \zeta(M, G_{Z,Z}^{\lambda,\lambda}, \gamma_{sift}/n_q)$, where

$$\zeta(x,y,z) = \left[ \frac{x + \ln (1/z)}{y} \right] 1 + \sqrt{1 + \frac{2x}{\ln (1/z)}} \right] ,$$

and $G_{Z,Z}^{\lambda,\lambda}$ is the probability that any given round of the QKD session between QKD$_A_j$ and QKD$_B_j$ contributes to $Z_j$. An expression of $G_{Z,Z}^{\lambda,\lambda}$ for a typical channel model is given in Sec. E6.

b. $E_{tol}$

Following the MDI-QKD protocol of Sec. E1, the reconciliation of $s_A$ with $s_B$ is performed separately on each $s^l_A$. If, for simplicity, one assumes that the EC protocol corrects up to a fraction $E_{tol}$ of bit errors (and no more) with certainty, either $E_{j} \leq E_{tol}$ for all $j$ or the (single) EV step aborts, where $E_{j}$ denotes the actual error rate between $s^l_A$ and $s^l_B$. Thus, applying Chernoff’s inequality [46], $P$(EV aborts) $\leq \gamma_{EC}$ holds for any $\gamma_{EC} \in (0, 1)$ if

$$E_{tol} = \min \left\{ 1, E_{Z,Z}^{\lambda,\lambda} + \frac{\Delta_U(E_{Z,Z}^{\lambda,\lambda} M, \gamma_{EC}/n_q)}{M} \right\}$$

where $E_{Z,Z}^{\lambda,\lambda}$ is the expected bit error rate for the basis $Z$ and the common intensity $\lambda$, and the deviation function $\Delta_U(x,y)$ is defined in Eq. (E25). An expression of $E_{Z,Z}^{\lambda,\lambda}$ for a typical channel model is given in Sec. E6.
6. Channel model

In this section, we derive expressions for the expected values of the observables of the protocol, considering the setup illustrated in Fig. (4). To begin with, let us elaborate on the mathematical models we use.

a. Models

1. Laser sources. Alice’s and Bob’s photon sources emit PR-WCP of the form

$$\rho_\tau = \frac{1}{2\pi} \int_0^{2\pi} |\tau\rangle\langle\tau| \, d\gamma,$$  \hspace{1cm} (E37)

where $|\tau\rangle = \exp(\tau a^\dagger - \tau^* a) \, |0\rangle$ is a coherent state, with amplitude $\tau = |\tau| e^{i\gamma} \in \mathbb{C}$. Here, $a^\dagger$ (a) and $|0\rangle$ are the creation (annihilation) operator and the vacuum state for mode $a$, such that a Fock state with $n$ photons in this mode is given by $|n\rangle = a^n |n\rangle/[\sqrt{n}]!0\rangle$.

2. Channel and detector loss. An effective beam-splitter with transmittance $\eta = \eta_{\text{ch}}\eta_{\text{det}}$ is used to jointly model channel loss ($\eta_{\text{ch}}$) and detector loss ($\eta_{\text{det}}$) on each side. The transformation reads

$$p^\dagger \rightarrow \sqrt{\eta} r^\dagger + \sqrt{1-\eta} s^\dagger,$$

$$q^\dagger \rightarrow \sqrt{\eta} s^\dagger - \sqrt{1-\eta} r^\dagger,$$  \hspace{1cm} (E38)

where the quantum signal enters through the input port $p$, a vacuum state enters through the input port $q$, the output port $r$ leads to unit detection efficiency detectors, and the output port $s$ represents channel and detection loss. In turn, $\eta_{\text{ch}} = 10^{-\alpha_{\text{att}} L/10}$, $\alpha_{\text{att}}$ being the attenuation coefficient of the channel (in dB/km), and $L$ being the transmission length between each party and the central node (in km).

3. Basis choice and polarization misalignment. Let $a^\dagger_h$ ($a^\dagger_v$) denote the creation operator of a photon with horizontal (vertical) polarization in a pre-fixed basis $Z$. For each party, the selection of the basis setting $\theta \in \{0, \pi/4\}$ and the occurrence of a polarization misalignment $\delta_{\text{mis}} > 0$ jointly transform $a^\dagger_h$ and $a^\dagger_v$ according to the following unitary operation:

$$a^\dagger_h \rightarrow \cos(\theta + \delta_{\text{mis}}) a^\dagger_h - \sin(\theta + \delta_{\text{mis}}) a^\dagger_v,$$

$$a^\dagger_v \rightarrow \cos(\theta + \delta_{\text{mis}}) a^\dagger_v - \sin(\theta + \delta_{\text{mis}}) a^\dagger_h.$$  \hspace{1cm} (E39)

In short, for any given $\delta_{\text{mis}}$, setting $\theta = 0$ ($\theta = \pi/4$) in Eq. (E39) jointly models that the party selected basis $Z$ ($X$) and a polarization misalignment $\delta_{\text{mis}}$ occurred in the channel.

4. Photo-detectors. Threshold detectors are considered, meaning that each of them is modeled with a POVM consisting of only two elements: $\{E_{\text{no click}}, E_{\text{click}}\}$. As the detector loss is already accounted for in the channel model, the POVM here must describe unit efficiency photo-detectors, but having a non-zero dark count probability $p_d$. That is,

$$E_{\text{no click}} = (1 - p_d) |0\rangle\langle 0|, \hspace{1cm} E_{\text{click}} = \mathbb{I} - E_{\text{no click}}.$$  \hspace{1cm} (E40)

The operator $\mathbb{I}$ denotes the identity operator in the photon-number basis, i.e., $\mathbb{I} = \sum_{n=0}^{\infty} |n\rangle\langle n|$.

b. Relevant experimental parameters

First of all, let us introduce some convenient notation. At every round of the protocol, $\theta_A$ ($\theta_B$) $\in \{0, \pi/4\}$ denotes Alice’s (Bob’s) basis setting, where, as usual, $0$ ($\pi/4$) stands for basis $Z$ ($X$). Similarly, $i$ and $j$ $\in \{1, 2\}$ respectively denote Alice’s and Bob’s polarization states, such that, for basis $Z$ ($X$), 1 means “h” (“+”) and 2 means “v” (“”). Regarding the photo-detectors, they are numbered by $w \in \{1, 2, 3, 4\}$ as shown in Fig. 4. Also, for each photo-detector $w$, it is convenient to introduce an “arm index” $s_w \in \{1, 2\}$ specifying whether it is on the right arm ($s_w = 1$) or the left arm ($s_w = 2$) of the detection scheme, and another “polarization index” $k_w \in \{1, 2\}$ specifying whether they detect the horizontal ($k_w = 1$) or the vertical ($k_w = 2$) component of the pulses coming from the polarizing
FIG. 4: Schematic of the decoy-state MDI-QKD setup. Alice (Bob) holds a laser source that emits PR-WCPs in any of the four BB84 states, defined by a polarization setting \(i\) \((j)\) ∈ \(\{1, 2\}\) and a basis setting \(\theta_A\) \((\theta_B)\) ∈ \(\{0, \pi/4\}\). An intensity modulator (IM) selects the amplitude \(|\alpha|\) \(|\beta|\) of Alice’s (Bob’s) laser pulse. The overall one-sided efficiency is denoted by \(\eta = \eta_{ch}\eta_{det}\), where \(\eta_{det}\) is the detector efficiency (set to a common value for all the photo-detectors) and \(\eta_{ch} = 10^{-\alpha L/10}\) is the transmission efficiency, \(\alpha\) (dB/km) being the attenuation coefficient of the channel and \(L\) (km) being the common transmission length between each party and the central node. The angle \(\delta_A\) \((\delta_B)\) ≥ 0 denotes the polarization misalignment occurring in the left (right) arm of the setup (denoted by \(\delta_{\text{min}}\) in Eq. (E39)) and the symbol “⊗” stands for polarizing beam-splitter (PBS). Blue color is used for the intensities \(|\xi_{w}^{ij}\rangle\) that arrive at the detectors \(w \in \{1, 2, 3, 4\}\). Each detector has an “arm index” \(s_w\) ∈ \(\{1, 2\}\) that specifies whether it is on the right arm \((s_w = 1)\) or the left arm \((s_w = 2)\) of the detection scheme, and a polarization index \(k_w\) ∈ \(\{1, 2\}\) specifying whether it detects the horizontal \((k_w = 1)\) or the vertical \((k_w = 2)\) component of the pulses coming from the PBSs. For simplicity, this last index is not shown in the figure.

Let us assume for the moment that Alice’s (Bob’s) laser emits pure coherent states with complex amplitude \(\alpha\) \((\beta)\) in the BB84 state defined by \(i\) and \(\theta_A\) \((j\) and \(\theta_B)\). The quantum state at the input port of the detectors also factors as the product of four coherent states, \(|\phi_{\text{det}}\rangle = |\xi_{1w}^{ij}\rangle|\xi_{2w}^{ij}\rangle|\xi_{3w}^{ij}\rangle|\xi_{4w}^{ij}\rangle\), \(\xi_{w}^{ij}\) denoting the incoming amplitude to detector \(w\) for settings \(i\) and \(j\) (the dependence on the intensity settings, \(\alpha\) and \(\beta\), and the basis settings, \(\theta_A\) and \(\theta_B\), is omitted for readability). Precisely, it can be shown that

\[
|\xi_{w}^{ij}\rangle = \frac{\eta}{2} \left[ \alpha \Theta_{A,i,k_w} + (-1)^{s_w} \beta \Theta_{B,j,k_w} \right],
\]

where \(\eta = \eta_{ch}\eta_{det}\) is the overall one-sided efficiency (accounting for both the transmission efficiency of the channel, \(\eta_{ch}\), and the detection efficiency of Charles’ detectors, \(\eta_{det}\)), and \(\Theta_{A,l,m}(\Theta_{B,l,m})\) is the \((l, m)\)-th element of the matrix \(\Theta_A\) \((\Theta_B)\), which incorporates Alice’s (Bob’s) measurement setting, \(\theta_A\) \((\theta_B)\), and the polarization misalignment occurring in her (his) side of the channel, \(\delta_A\) ≥ 0 \((\delta_B \geq 0)\):

\[
\Theta_A = \begin{bmatrix}
\cos(\theta_A + \delta_A) & \sin(\theta_A + \delta_A) \\
-\sin(\theta_A + \delta_A) & \cos(\theta_A + \delta_A)
\end{bmatrix}
\quad \text{and} \quad
\Theta_B = \begin{bmatrix}
\cos(\theta_B + \delta_B) & \sin(\theta_B + \delta_B) \\
-\sin(\theta_B + \delta_B) & \cos(\theta_B + \delta_B)
\end{bmatrix}.
\]

The cases of interest are \(\theta_A = \theta_B = 0\) (basis Z match) and \(\theta_A = \theta_B = \pi/4\) (basis X match). If, without loss of generality, we set \(\alpha = |\alpha|\) and \(\beta = |\beta|e^{i\gamma}\), the intensities (squared modulus of the amplitudes) at the detectors read

\[
|\xi_{w}^{ij}\rangle^2 = \frac{\eta}{2} \left[ |\alpha|^2 \Theta_{A,i,k_w}^2 + |\beta|^2 \Theta_{B,j,k_w}^2 + (-1)^{s_w} 2 |\alpha||\beta| \Theta_{A,i,k_w} \Theta_{B,j,k_w} \cos \gamma \right], \quad w \in \{1, 2, 3, 4\}.
\]

Since a success at the central node is heralded by the click of exactly two detectors referred to orthogonal polarizations, the set of possible successful events reads

\[
\Omega = \{(1, 2), (3, 4), (1, 4), (2, 3)\}.
\]

As an example, let us compute the probability \(P_{(1,2)}^{ij}\) of the successful event \((1, 2)\). This probability factors as

\[
P_{(1,2)}^{ij} = p(3 \text{ and } 4 \text{ do not click}) \times p(1 \text{ and } 2 \text{ click}).
\]
Then, from our detector model and the poissonian statistics of coherent states, we have that
\[
p(3 \text{ and 4 do not click}) = (1 - p_d)^2 e^{-\left(|\xi_u^{i,j}|^2 + |\xi_v^{i,j}|^2\right)},
\]
\[
p(1 \text{ and 2 click}) = \left(1 - e^{-|\xi_u^{i,j}|^2}\right) \left(1 - e^{-|\xi_v^{i,j}|^2}\right) + p_d \left[\left(1 - e^{-|\xi_u^{i,j}|^2}\right) e^{-|\xi_v^{i,j}|^2} + \left(1 - e^{-|\xi_v^{i,j}|^2}\right) e^{-|\xi_u^{i,j}|^2}\right] + p_d^2 e^{-\left(|\xi_u^{i,j}|^2 + |\xi_v^{i,j}|^2\right)}. \tag{E46}
\]
Putting both factors together and generalising the expression to an arbitrary successful event \((u,v)\), one obtains
\[
P^{i,j}_{(u,v)} = (1 - p_d)^2 \exp\left(- \sum_{w \neq u,v} |\xi_w^{i,j}|^2\right) \left[1 - (1 - p_d) \left(e^{-|\xi_u^{i,j}|^2} + e^{-|\xi_v^{i,j}|^2}\right) + (1 - p_d)^2 e^{-\left(|\xi_u^{i,j}|^2 + |\xi_v^{i,j}|^2\right)}\right], \tag{E47}
\]
or, more conveniently,
\[
P^{i,j}_{(u,v)} \left(1 - p_d\right)^2 = \exp\left(- \sum_{w \neq u,v} |\xi_w^{i,j}|^2\right) \left(1 - p_d\right) \left[\exp\left(- \sum_{w \neq u} |\xi_w^{i,j}|^2\right) + \exp\left(- \sum_{w \neq v} |\xi_w^{i,j}|^2\right)\right] \left(1 - p_d\right)^2 \exp\left(- \sum_{w} |\xi_w^{i,j}|^2\right). \tag{E48}
\]
Recalling that \(P^{i,j}_{(u,v)}\) was computed assuming pure coherent states, one needs to average over phase values in order to derive the resulting probability for PR-WCPs, which we denote by \(P^{i,j}_{(u,v),\alpha,\beta,\theta_A,\theta_B} = \frac{1}{2\pi} \int_0^{2\pi} P^{i,j}_{(u,v)} d\gamma\). For convenience, this notation explicitly shows that, in any round of the protocol, the probability of a successful detection event \((u,v) \in \Omega\) depends on all the protocol settings. The explicit calculation of this integral yields
\[
P^{i,j}_{(u,v),\alpha,\beta,\theta_A,\theta_B} \left(1 - p_d\right)^2 = \exp\left(- \frac{n}{2} \sum_{w \neq u,v} (|\alpha|^2 \Theta_{A,i,k_w}^2 + |\beta|^2 \Theta_{B,j,k_w}^2)\right) I_{0,\text{sym}} \left(\eta|\alpha||\beta| \sum_{w \neq u,v} (-1)^{s_w+1} \Theta_{A,i,k_w} \Theta_{B,j,k_w}\right)
- \left(1 - p_d\right) \exp\left(- \frac{n}{2} \sum_{w \neq u} (|\alpha|^2 \Theta_{A,i,k_w}^2 + |\beta|^2 \Theta_{B,j,k_w}^2)\right) I_{0,\text{sym}} \left(\eta|\alpha||\beta| \sum_{w \neq u} (-1)^{s_w+1} \Theta_{A,i,k_w} \Theta_{B,j,k_w}\right)
- \left(1 - p_d\right) \exp\left(- \frac{n}{2} \sum_{w \neq v} (|\alpha|^2 \Theta_{A,i,k_w}^2 + |\beta|^2 \Theta_{B,j,k_w}^2)\right) I_{0,\text{sym}} \left(\eta|\alpha||\beta| \sum_{w \neq v} (-1)^{s_w+1} \Theta_{A,i,k_w} \Theta_{B,j,k_w}\right)
+ (1 - p_d)^2 \exp\left(- \frac{n}{2} \sum_{w} (|\alpha|^2 \Theta_{A,i,k_w}^2 + |\beta|^2 \Theta_{B,j,k_w}^2)\right) I_{0,\text{sym}} \left(\eta|\alpha||\beta| \sum_{w} (-1)^{s_w+1} \Theta_{A,i,k_w} \Theta_{B,j,k_w}\right). \tag{E49}
\]
where we have introduced the function \(I_{0,\text{sym}}(x) = \left[I_0(x) + I_0(-x)\right]/2 = \frac{1}{2\pi} \int_0^{2\pi} e^{x \cos \gamma} d\gamma\), \(I_0\) being the modified Bessel function of the first kind.

From Eq. (E49), one can compute the probability \(Q^{\alpha,\beta}_{\theta_A,\theta_B}\), that any given round yields a successful BSM at the central node when Alice (Bob) uses basis \(\theta_A\) (\(\theta_B\)) and intensity \(|\alpha|^2 (|\beta|^2),
\]
\[
Q^{\alpha,\beta}_{\theta_A,\theta_B} = \frac{1}{4} \sum_{(u,v) \in \Omega} \left[P^{1,1}_{(u,v),\alpha,\beta,\theta_A,\theta_B} + P^{1,2}_{(u,v),\alpha,\beta,\theta_A,\theta_B} + P^{2,1}_{(u,v),\alpha,\beta,\theta_A,\theta_B} + P^{2,2}_{(u,v),\alpha,\beta,\theta_A,\theta_B}\right], \quad \theta_A, \theta_B \in \{0, \pi/4\}. \tag{E50}
\]
Similarly, one can compute the bit error rates, \(E^{\alpha,\beta}_{0,0}\) and \(E^{\alpha,\beta}_{\frac{\pi}{4}, \frac{\pi}{4}}\), for the basis coincidences, given by
\[
Q^{\alpha,\beta}_{0,0} E^{\alpha,\beta}_{0,0} = \frac{1}{4} \sum_{(u,v) \in \Omega} \left[P^{1,1}_{(u,v),\alpha,\beta,0,0} + P^{2,2}_{(u,v),\alpha,\beta,0,0}\right]
\]
and
\[
Q^{\alpha,\beta}_{\frac{\pi}{4}, \frac{\pi}{4}} E^{\alpha,\beta}_{\frac{\pi}{4}, \frac{\pi}{4}} = \frac{1}{4} \left\{ \sum_{(u,v) \in \Omega_1} \left[P^{1,1}_{(u,v),\alpha,\beta,\frac{\pi}{4}, \frac{\pi}{4}} + P^{2,2}_{(u,v),\alpha,\beta,\frac{\pi}{4}, \frac{\pi}{4}}\right] + \sum_{(u,v) \in \Omega_2} \left[P^{1,2}_{(u,v),\alpha,\beta,\frac{\pi}{4}, \frac{\pi}{4}} + P^{2,1}_{(u,v),\alpha,\beta,\frac{\pi}{4}, \frac{\pi}{4}}\right] \right\}, \tag{E51}
\]
where \(\Omega_1 = \{(1,4),(2,3)\}\) and \(\Omega_2 = \{(1,2),(3,4)\}\). Note that only the polarization settings \(h,h\) and \(v,v\) \((i.e., \ i = j = 1\) and \(i = j = 2\)) contribute to the basis \(Z\) bit error rate. This is so because, for these rounds, Bob flips his bit
irrespectively of the successful outcome of the BSM. On the contrary, for the basis X bit error rate, the post-selection of $|\psi^-\rangle$ in the BSM (events (1, 4) and (2, 3)) entails a bit flip, while the post-selection of $|\psi^+\rangle$ (events (1, 2) and (3, 4)) does not, thus leading to the definition of the bit error rate given in Eq. (E51).

Finally, we write down the expected values of the observables required for the simulations, according to the channel model above. For this purpose, we introduce the quantities $G_{Z,Z}^{\lambda,\lambda} = q_Z^2 Q_{0,0,\sqrt{\pi},\sqrt{\pi}}$, $G_{X,X}^{a,b} = q_X^2 p_a p_b Q_{0,0,\sqrt{\pi},\sqrt{\pi}}$, $G_{Z,X}^{\lambda,\lambda} = q_Z q_X p_a p_b Q_{0,0,\sqrt{\pi},\sqrt{\pi}}$, $E_{Z,Z}^{\lambda,\lambda} = E_{0,0,\sqrt{\pi},\sqrt{\pi}}$ and $E_{X,X}^{a,b} = E_{0,0,\sqrt{\pi},\sqrt{\pi}}$, with $a, b \in A = \{\mu, \nu, \omega\}$. From these quantities, it follows that

$$E[|Z_j|] = G_{Z,Z}^{\lambda,\lambda} N,$$
$$E[E_j] = E_{Z,Z}^{\lambda,\lambda},$$
$$E[|\lambda_j^{a,b}|] = G_{X,X}^{a,b} N,$$
$$E[\varepsilon_j^{a,b}] = E_{X,X} G_{X,X}^{a,b} N,$$
$$E[|a^j|_{\ell_j}] = \left(\sum_{a,b \in A} G_{X,X}^{a,b} + \sum_{b \in A} G_{Z,X}^{b,\lambda} + \sum_{a \in A} G_{Z,Z}^{\lambda,\lambda}\right) 2N \text{ bits},$$
$$E[|r^j_{A}|_{\ell_j,X}] = \left(\sum_{a,b \in A} G_{X,X}^{a,b} + \sum_{a \in A} G_{X,X}^{a,\lambda}\right) N \text{ bits and}$$
$$E[|r^j_{A}|_{\ell_j,Z}] = \left(G_{Z,Z}^{\lambda,\lambda} + \sum_{b \in A} G_{Z,Z}^{b,\lambda}\right) N \text{ bits.} \quad (E52)$$

We recall that $N$ is the number of signals transmitted per module in each QKD pair.

Appendix F: Decoy-state BB84

To further illustrate the applicability of the results in the main text, we combine Protocol with the standard decoy-state BB84 scheme \[37, 47–49\] with three common decoy intensities per basis.

1. QKD protocol

Again, although the protocol description below assumes that the corrupted devices do not deviate from the protocol, the security against actively misbehaving corrupted devices is established in the main text. As for the MDI-QKD protocol, we define the sets $\sigma_A^R = \{\text{CP}_{A_i}\}_i^R$ and $\sigma_B^R = \{\text{CP}_{B_i}\}_i^R$, where we recall that $R$ is given in Table 1 of the main text for each corruption model of the CP units. For instance, $R = 2t_c + 1$ in the AC corruption model.

For $j = 1, \ldots, n_q$, QKD$_{A_j}$ creates a trio of strings $(r_j^A, k_j^A, a_j^j)$. The string $r_j^A \in \{0,1\}^N$ is fully random (polarization bits string). For all $i = 1, \ldots, N$, $k_j^A \in \{Z,X\}^N$ verifies $P[k_j^A = \zeta] = q_\zeta$ with $\zeta \in \{Z,X\}$ (basis string), and $a_j^j \in \{\mu, \nu, \omega\}^N$ (intensities string) verifies $P[a_j^j = a] = p_a$ for $a \in A$, with $A = \{\mu, \nu, \omega\}$. Similarly, QKD$_{B_j}$ creates its basis string $k_j^B \in \{Z,X\}^N$ verifying $P[k_j^B = \zeta] = q_\zeta$ with $\zeta \in \{Z,X\}$ and $i = 1, \ldots, N$.

Let us now focus on a single QKD pair, say, the $j$-th one. For $i$ ranging from 1 to $N$, steps (i) to (iii) are repeated.

(i) **State preparation** QKD$_{A_j}$ prepares a PR-WCP with intensity $a_j^j$ in the BB84 state defined by $k_j^A$ and $r_j^A$.

(ii) **Transmission** QKD$_{A_j}$ sends the state to QKD$_{B_j}$ via the quantum channel.

(iii) **Measurement** QKD$_{B_j}$ performs a measurement in basis $k_j^B$ and stores the outcome in a classical value $r_j^{B_i} \in \{0, 1, \emptyset\}$, where $\emptyset$ is the symbol produced when no signal is detected. If a multiple click takes place, Bob assigns a random bit to this event.
After the above quantum communication phase, the distributed QKD post-processing starts. Again, we focus on a single QKD pair, say, the $j$-th one.

1. **Distribution of data.** Let $r_A^j|X$ ($r_A^j|Z$) be the sub-string of $r_A^j$ where QKD$_{A_j}$ uses basis $X$ ($Z$) for the encoding. QKD$_{A_j}$ communicates the trio $(k_A^j, a^j, r_A^j|X)$ to every CP$_{A_i} \in \sigma_A^j$ and uses the Share protocol of a conditional VSS scheme to distribute shares of $r_A^j|Z$ among them. Let $c^j$ be the string of detector clicks held by QKD$_{B_j}$, such that $c^j = 0$ if $r_B^j = 0$ and $c^j = 1$ otherwise, and let $k_B^j|c_j$ be the restriction of the basis string $k_B^j$ to the non-zero entries of $c^j$. Also, let $r_B^j|c_j, X$ ($r_B^j|c_j, Z$) be the restriction of $r_B^j$ to the non-zero entries of $c^j$ where QKD$_{B_j}$ uses basis $X$ ($Z$) for the measurements. QKD$_{B_j}$ communicates the trio $(c^j, k_B^j|c_j, r_B^j|c_j, X)$ to every CP$_{B'_i} \in \sigma_B^j$ and uses the Share protocol of a conditional VSS scheme to distribute shares of $r_B^j|c_j, Z$ among them. All the CP$_{A_i} \in \sigma_A^j$ (CP$_{B_i} \in \sigma_B^j$) perform a consistency test on $(k_A^j, a^j, r_A^j|X)$ ($(c^j, k_B^j|c_j, r_B^j|c_j, X)$).

2. **Sifting.** Every CP$_{A_i} \in \sigma_A^j$ sends $(k_A^j, a^j, r_A^j|X)$ to every CP$_{B'_i} \in \sigma_B^j$, which individually apply MV. Then, each CP$_{B'_i} \in \sigma_B^j$ builds the index sets

$$Z_j = \left\{ i | c^j_i = 1, k_A^j = k_B^j, z^j = Z \right\} \quad \text{and} \quad \lambda^a_j = \left\{ i | c^j_i = 1, k_A^j = k_B^j = X, a^j = a \right\},$$

for all $a \in A$, and checks if the sifting condition $|Z_j| \geq M$ is met, for a pre-established threshold value $M$. If it is not met, the CP$_{B'_i} \in \sigma_B^j$ abort the protocol. In case of not aborting, the CP$_{B'_i} \in \sigma_B^j$ units forward $Z_j$ to the rest of the units, which apply MV. All together the CP$_{B'_i}$ perform a RBS protocol to select a random subset $Z'_j \subseteq Z_j$, of size $M$. Then, Bob’s units locally perform the sifting. Precisely, every CP$_{B'_i}$ builds its shares of the sifted key $s_B^j = r_B^j|Z'_j$ from those of $r_B^j|c_j, Z$ (discarding the data external to $Z'_j$).

3. **Parameter estimation.** For each $a \in A$, every CP$_{B'_i} \in \sigma_B^j$ unit builds the PE strings $r_B^j|X_a^j$ and $r_A^j|X_a^j$ from the respective strings $r_B^j|c_j, X$ and $r_A^j|X$, discarding the data external to $\lambda^a_j$. Then, each of them computes the numbers of bit errors

$$e_a^j = \sum_{k=1}^{n} r_A^j_k | X_a^j \oplus r_B^j_k | X_a^j,$$

for $a \in A$, where $r_A^j_k | X_a^j$ ($r_B^j_k | X_a^j$) denotes the $k$-th bit of the corresponding string. Using $|Z_j|$ and the different $|\lambda^a_j|$ and $e_a^j$ ($a \in A$), every CP$_{B'_i} \in \sigma_B^j$ computes a lower bound on the number $n_Z^j$ of single-photon successes in $Z'_j$ and an upper bound on the single-photon phase-error rate $\phi_{i,j}$ associated to the single-photon successes in $Z'_j$.

Although the rest of the post-processing is identical to that of the MDI-QKD protocol (except from the fact that no bit flips are required to correlate $s_A$ and $s_B$), we include it here for completeness. The above steps 1 to 3 are performed for all $j = 1, \ldots, n_q$. At this stage, every CP$_{B'_i} \in \sigma_B^j$ derives a lower bound $l$ (given in the next section) on the secret key length that can be extracted from the concatenated sifted key $s_B = s_B^1 \ldots s_B^n$ via PA. If a CP$_{B'_i} \in \sigma_B^j$ finds $l \leq 0$, it aborts the protocol.

4. **RBS generation.** If the protocol does not abort, every CP$_{B'_i} \in \sigma_B^j$ forwards $l$ to the rest of Bob’s units, which apply MV. All CP$_{B'_i}$ perform a RBS generation protocol to randomly select two 2-universal hash functions $h_{EV}$ and $h_{PA}$, respectively devoted to error verification (EV) and PA. Following [35], if Toeplitz matrices are used for this purpose, $2[\log_2(2/\epsilon_{cor})]$ $(Mn_q + l - 1)$ bits are required to specify $h_{EV}$ ($h_{PA}$).

5. **Information reconciliation** Every CP$_{B'_i}$ computes its shares of (1) the concatenated syndromes string $s_B^j = s_B^j(s_B^1) \ldots s_B^j(s_B^n)$ and (2) the EV tag $h_{EV,B} = h_{EU}(s_B)$. All together, the CP$_{B'_i}$ reconstruct $s_B$ and $h_{EV,B}$ via the Reconstruct protocol of a conditional VSS scheme (see the Methods section in the main text). Each CP$_{B'_i} \in \sigma_B^j$ sends the following items to every CP$_{A_i} \in \sigma_A^j$:

1. The string $s_{Z'_j} = s_{Z'_j}^{s_B} \ldots s_{Z'_j}^{s_B^n}$, where $s_{Z'_j}$ specifies, say, the positions in $r_A^j|Z$ that contribute to $Z'_j$.

2. The syndrome information $s_y(s_B)$, together with the description of $h_{EV}$ and the EV tag $h_{EV}(s_B)$.

3. The description of $h_{PA}$. 
Each $\text{CP}_{i} \in \sigma_{A}^{j}$ decides on all three items via MV and communicate $s_{z'}$, $h_{\text{EV}}$ and $h_{\text{PA}}$ to the rest of Alice’s units, which apply MV too. Then, they proceed as follows. Using $s_{z'}$, all $\text{CP}_{i}$ shrink their shares of $r_{j}|z = r_{j}^{1}|z \ldots r_{j}^{n_{q}}|z$ into shares of $s_{A} = s_{A}^{1} \ldots s_{A}^{n_{q}}$, where $s_{A}^{j} = r_{j}^{j}|z_{j}$. All the $\text{CP}_{i}$ compute shares of $sy(s_{A})$ from those of $s_{A}$ and then perform the Reconstruct protocol of a conditional VSS scheme to agree on $sy(s_{A})$. Coming next, the $\text{CP}_{i} \in \sigma_{A}^{j}$ compute the error pattern $\hat{e}$ from $sy(s_{B})$ and $sy(s_{A})$ and update the first share of $s_{A}$ XOR-ing it with $\hat{e}$ (i.e., key reconciliation is achieved by acting on a single share). We denote the corrected key by $\hat{s}_{A} = s_{A} \oplus \hat{e}$. Using $h_{\text{EV}}$, all the $\text{CP}_{i}$ compute their shares of $h_{\text{EV}}(\hat{s}_{A})$ and reconstruct it via the Reconstruct protocol of a conditional VSS scheme. Then, each $\text{CP}_{i} \in \sigma_{A}^{j}$ checks that $h_{\text{EV}}(\hat{s}_{A}) = h_{\text{EV}}(s_{B})$. Otherwise, it aborts the protocol.

6. **Privacy amplification.** In case of not aborting, all the $\text{CP}_{i}$ compute their shares of Alice’s final key $S_{A} = h_{\text{PA}}(\hat{s}_{A})$. Similarly, if no abortion is notified, all the $\text{CP}_{B,i}$ compute their shares of Bob’s final key $S_{B} = h_{\text{PA}}(\hat{s}_{B})$.

### 2. Secret key length formula

#### a. AC, AN and PC corruption models for the QKD modules

Here, we particularise the secret key length for the decoy-state BB84 protocol of Sec. F 1. The formula is tight within the AC, AN and PC corruption models for the QKD modules and we maintain the asterisks to emphasize that we refer to well-defined/unique quantities consistently held by honest (and/or passively corrupted) units. The analysis is identical to the one for the MDI-QKD protocol, given in Sec. E 2, and we omit it here for simplicity. Precisely, the extractable key length is

$$l^{*} = \min \left\{ n_{1,2}^{j,\text{L}} \left[ 1 - h(\phi_{1,2}^{j,\text{U}}) \right] - \left| sy^{*}(s_{B}^{j,\text{U}}) \right| \right\} - \log_{2} \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{\text{PA}} \delta} \right), \tag{F3}$$

where $n_{1,2}^{j,\text{L}}$ ($\phi_{1,2}^{j,\text{U}}$) stands for a lower (upper) bound on $n_{1,2}^{j,\text{U}}$ ($\phi_{1,2}^{j,\text{U}}$), $h(\cdot)$ is the binary entropy function, $\left| sy^{*}(s_{B}^{j,\text{U}}) \right|$ is the size of the $j$-th EC syndrome, $\epsilon_{\text{cor}}$ is the correctness parameter, $\epsilon_{\text{PA}}$ is the error probability of the privacy amplification, and $\delta > 0$. Also, as shown in Sec. A, the above key length is $\epsilon_{\text{sec}}$-secret for all

$$\epsilon_{\text{sec}} = \hat{\epsilon}_{\text{sec}} + \epsilon_{\text{AU}} \tag{F4}$$

with $\hat{\epsilon}_{\text{sec}} \geq 2\epsilon + \delta + \epsilon_{\text{PA}}$, where $\epsilon$ is upper-bounded by the sum of the error probabilities of the estimates of $n_{1,2}^{j,\text{L}}$ and $\phi_{1,2}^{j,\text{U}}$, and $\epsilon_{\text{AU}}$ is the pre-agreed total error probability of the authentication, which depends on the corruption model of the CP units (see the Results section in the main text). Explicit expressions of $n_{1,2}^{j,\text{L}}$ and $\phi_{1,2}^{j,\text{U}}$ in terms of the observables of the protocol are given in the next section, together with an upper bound on the smooth-parameter $\epsilon$.

#### b. PN corruption model for the QKD modules

Within the PC corruption model, the following tighter key length formula holds,

$$l \leq \min \left\{ n_{1,2}^{j,\text{L}} \left[ 1 - h(\phi_{1,2}^{j,\text{U}}) \right] - \left| sy(\hat{s}_{B}) \right| \right\} - \log_{2} \left( \frac{1}{\epsilon_{\text{cor}} \epsilon_{\text{PA}} \delta n_{q} - 1} \right), \tag{F5}$$

where $\epsilon_{\text{sec}} = \hat{\epsilon}_{\text{sec}} + \epsilon_{\text{AU}}$ with $\hat{\epsilon}_{\text{sec}} \geq (n_{q} - 1)(2\epsilon + \delta) + \epsilon_{\text{PA}}$, and $\epsilon_{\text{AU}}$ is again pre-agreed by Alice and Bob and depends on the corruption model of the CP units. The analysis is again identical to that of Sec. E 2.

### 3. Parameter estimation

Here, we give the analytical bounds $n_{1,2}^{j,\text{L}}$ and $\phi_{1,2}^{j,\text{U}}$ that enter Eq. (F3). These bounds were originally presented in [37] and we include them here for completeness. Since the analysis is common for all $j = 1, \ldots, n_{q}$, we drop the QKD pair index $j$ and refer to any of the QKD pairs. In addition, we drop the asterisks for readability.
The decoy-state bounds below require $\mu > \nu + \omega$ and $\nu > \omega \geq 0$, where we recall that $A = \{ \mu, \nu, \omega \}$ is the set of intensity settings. In addition, let us introduce the decomposition $Z' = \cup_{a \in A} Z'^a$, where $Z'^a = \{ i \in Z' | a_i = a \}$. The observed sizes of the sets $Z'^a$ determine $n_{1,Z}^a$. Precisely, we have that for all $\epsilon_H \in (0, 1)$, $P(n_{1,Z} < n_{1,Z}^L) < 3\epsilon_H$ holds for

$$n_{1,Z}^L = \left\lfloor \frac{\mu \tau_1}{\mu (\nu - \omega) - (\nu^2 - \omega^2)} \left\{ \frac{e}{p_\nu} \left[ Z'^\nu - \delta_M, \epsilon_H \right] - \frac{e}{p_\omega} \left[ Z'^\omega - \delta(M, \epsilon_H) \right] - \frac{\nu^2 - \omega^2}{\mu^2} \frac{e}{p_\mu} \left[ Z'^\mu + \delta(M, \epsilon_H) \right] \right\} \right\rfloor,$$

where $\tau_1 = \mu e^{-\nu} p_\mu + \nu e^{-\nu} p_\nu + \omega e^{-\omega} p_\omega$ and $\delta(x, y) = \sqrt{(x/2) \ln y^{-1}}$ is the deviation term that follows from the use of Hoeffding’s inequality [50]. Such inequality is used three times in Eq. (F6) (with a common error probability, $\epsilon_H$) to obtain adequate one-sided bounds on the expected values of $Z'^\mu$, $Z'^\nu$ and $Z'^\omega$, respectively, given their realisations. Note that, for this task, one could also apply the inverse Chernoff-bound given in Sec. G.

Similarly, $|\mathcal{X}^\mu|$, $|\mathcal{X}^\nu|$ and $|\mathcal{X}^\omega|$ determine a lower bound on the number $S_{1,X}^L$ of rounds in $\mathcal{X} = \cup_{a \in A} \mathcal{X}^a$ where Alice sent single photons. Precisely, $P(S_{1,X} < S_{1,X}^L) < 3\epsilon_H$ holds for

$$S_{1,X}^L = \left\lfloor \frac{\mu \tau_1}{\mu (\nu - \omega) - (\nu^2 - \omega^2)} \right\rfloor \left\{ \frac{e}{p_\nu} \left[ |\mathcal{X}^\nu| - \delta (|\mathcal{X}|, \epsilon_H) \right] - \frac{e}{p_\omega} \left[ |\mathcal{X}^\omega| + \delta (|\mathcal{X}|, \epsilon_H) \right] - \frac{\nu^2 - \omega^2}{\mu^2} \frac{e}{p_\mu} \left[ |\mathcal{X}^\mu| + \delta (|\mathcal{X}|, \epsilon_H) \right] \right\} \right\rfloor,$$

where we assumed a common error probability, $\epsilon_H$, for each usage of Hoeffding’s inequality [50] again.

Regarding the number $E_{1,X}$ of single-photon errors in $\mathcal{X}$, it turns out that $P(E_{1,X} > E_{1,X}^U) < 2\epsilon_H$ holds for

$$E_{1,X}^U = \left\lfloor \frac{\tau_1}{\nu - \omega} \left\{ \frac{e}{p_\nu} \left[ e + \delta (e, \epsilon_H) \right] - \frac{e}{p_\omega} \left[ e - \delta (e, \epsilon_H) \right] \right\} \right\rfloor,$$

where we recall that $e_a$ is the observed number of errors in $\mathcal{X}^a$ ($a \in A$) and we defined $e = \sum_a e_a$. Also, the error probability $2\epsilon_H$ follows from the composition of two usages of Hoeffding’s inequality [50]. Finally, as we did for the parameter estimation in the MDI-QKD protocol, we use Serfling’s inequality [45] to relate the number $e_{1,Z}$ of single-photon errors in $Z'$ with the number $E_{1,X}$ of single-photon errors in $\mathcal{X}$. To be precise, it follows that $P(e_{1,Z} > e_{1,Z}^U) < 8\epsilon_H + \epsilon_S$ holds for

$$e_{1,Z}^U = \min \left\{ \left\lfloor n_{1,Z}^L \left( \frac{E_{1,X}^U}{S_{1,X}^U} \right) + (S_{1,X}^U + n_{1,Z}^L) \times \Upsilon (n_{1,Z}^L, S_{1,X}^L, \epsilon_S) \right\rfloor, n_{1,Z}^L \right\},$$

where the deviation function $\Upsilon (x, y, z)$ is given by Eq. (E23) and $\epsilon_S$ is the error probability of Serfling’s inequality [45]. Equivalently, the single-photon phase error rate $\phi_{1,Z}$ verifies $P(\phi_{1,Z} \geq \phi_{1,Z}^U) \leq 8\epsilon_H + \epsilon_S$ for

$$\phi_{1,Z}^U = \frac{\epsilon_{1,Z}^U}{n_{1,Z}^L},$$

where $n_{1,Z}^L$ is given in Eq. (F6) and $\epsilon_{1,Z}^U$ is given in Eq. (F9).

From the above PE procedure it follows that the smooth parameter $\varepsilon$ (see Sec. F 2) is upper-bounded as

$$\varepsilon \leq 8\epsilon_H + \epsilon_S,$$

(11)

4. Authentication cost

Here, we estimate the secret key bits consumed for authentication purposes in the BB84 protocol of Sec. F 1. The procedure is identical to that of Sec. E 4, so we skip the details.

On the one hand, the pre-agreed total error probability of the authentication, $\epsilon_{AU}$, is related to the individual error probability, $\gamma_{AU}$, via $\epsilon_{AU} \geq (\epsilon + 1)^2 (n_a + 1) \gamma_{AU}$ for active corruption of the CP units, and via $\epsilon_{AU} \geq (n_a + 1) \gamma_{AU}$ for active corruption of the CP units.
for passive corruption, where we remind the reader that $R$ is specified in Table 1 of the main text for every corruption model, and $t_c (n_q)$ is the threshold number of corrupted units per party (total number of QKD pairs).

On the other hand, the overall authentication cost, $l_{AU}$, is also determined by the corruption model of the CP units via $R$. Precisely, $l_{AU} = R^2 \times |k|$ secret bits, where $|k|$ stands for the size of the key pool pre-shared by each pair (CP$_{Ai}$, CP$_{Bi}$) with CP$_{Ai} \in \sigma_1^A$ and CP$_{Bi} \in \sigma_1^B$. A lower bound on $|k|$ is given in Eq. (E33), determined by the lengths $\{|m^A_A|\}_{j=1}^{n_A}$ and $|m_B|$ of the messages respectively sent by Alice and Bob in the BB84 protocol of Sec. F 1. Following the protocol description of Sec. F 1, we have

\[
|m^A_A| = |k^A| + |a^j| + |r^A_A|, \\
|m_B| = |s_Z| + |s_y(s_B)| + |h_{EV}(s_B)| + |h_{EV} description| + |h_{PA description}|. 
\] (F12)

In the previous equation, $k^A$ is a string of N bits, $a^j$ is a string of N trits (that can be accomodated with $2N$ bits), $r^j_A|\overline{A}$ is a string with an expected size of $E[|r^j_A|\overline{A}] = q_X N$ bits and $s_Z = \sum_{j=1}^{n_A} r^j_A|z|$, where $E[|r^j_A|z] = q_z N$ bits for all $j = 1, \ldots, n_A$. The size of the syndrome $|s_y(s_B)|$ depends on the EC protocol (and a typical model is given in the Discussion of the main text), $|h_{EV}(s_B)| = \log_2(2/\bar{\epsilon})$ bits, $|h_{EV} description| = 2\log_2(2/\bar{\epsilon})$ bits and $|h_{PA description}| = M n_q + l - 1$ bits, $l$ denoting the extractable key length, given by Eq. (F3) (Eq. (F5)) within the AC, AN and PC corruption models (PN corruption model) for the QKD modules.

5. Calculation of $N$ and $E_{tol}$ for the simulations

Here, we give adequate values for the number $N$ of signals transmitted per QKD$_{A_j}, j = 1, \ldots, n_q$, and for the threshold bit error rate of the EC protocol, $E_{tol}$, based on respective restrictions on the abortion probabilities of the sifting step and the error verification step. The analysis relies on a typical channel model described in Sec. F 6.

a. $N$

Let us impose a common abortion probability $\gamma_{sift}/n_q$ for each sifting step ($n_q$ of them in total). That is, we demand that $P(|Z_j| < M) \leq \gamma_{sift}/n_q$ for all $j = 1, \ldots, n_q$, where $|Z_j|$ is the set of detection events in which both parties use basis $Z$ and $M$ is the pre-specified size of the sifted keys (i.e., the block size). Using the Chernoff’s inequality [46], this condition is met if we set the number of signals transmitted per QKD$_{A_j} to N = \zeta(M; \sum a G^2_{Z,Z}; \gamma_{sift}/n_q)$, where $\zeta(x, y, z)$ is defined in Eq. (E33) and $\sum_a G^2_{Z,Z}$ is the probability that any given round contributes to $Z_j$ (see Sec. F 6).

b. $E_{tol}$

Following the BB84 protocol of Sec. F 1, EC is applied separately on each pair of sifted keys $(s^j_A, s^j_B)$. Assuming, for simplicity, that the EC protocol certainly corrects up to a fraction $E_{tol}$ of bit errors (and no more), either $E_j \leq E_{tol}$ for all $j$ or the (single) EV step aborts, where $E_j$ denotes the actual error rate between $s^j_A$ and $s^j_B$. Thus, applying Chernoff’s inequality [46], $P$ (EV aborts) $\leq \gamma_{EC}$ holds for any $\gamma_{EC} \in (0, 1)$ if

\[
E_{tol} = \min \left\{ 1, E_Z + \frac{\Delta(U(E_Z M, \gamma_{EC}/n_q))}{M} \right\} \tag{F13}
\]

where $E_Z$ is the expected bit error rate for the basis $Z$, i.e., $E_Z = E[E_j]$, and the deviation function $\Delta(U(x, y))$ is defined in Eq. (E25). An expression of $E_Z$ for a typical channel model is given in Sec. F 6.

6. Channel model

For the simulations, we adapt the typical channel model presented in Sec. E 6 to the decoy-state BB84 setup illustrated in Fig. 5.

Most of the notation is common with Sec. E 6: $n_{det}$ denotes the detector efficiency and $n_{ch} = 10^{-\alpha_{att} L}$ denotes the transmission efficiency, $\alpha_{att}$ (dB/km) being the attenuation coefficient and $L$ (km) being the transmission distance between Alice and Bob. Similarly, $p_{dc}$ stands for the dark count probability of the photo-detectors and $\delta_\gamma$ stands for the polarization misalignment occurring in the channel. In this setup, the relevant experimental parameters are the
yields basis-independent in the considered channel model. In particular, explicit calculation of \( G \) introduce the quantities fact that multiple clicks are randomly assigned to a specific detection outcome (see the caption of Fig. 5 for more

where \( \eta \) is the detector efficiency (set to a common value for both photo-detectors) and \( \eta_{\text{ch}} = 10^{-\alpha L/10} \) is the transmission efficiency, \( \alpha \) (dB/km) being the attenuation coefficient of the channel and \( L \) (km) being the transmission length. The angle \( \delta_A \geq 0 \) denotes the polarization misalignment occurring in the channel. On the other hand, Bob holds a detection system that consists of a polarization modulator (POL), a polarizing beam-splitter (PBS) denoted by the symbol “\( \oplus \)”, and two single-photon detectors. POL selects Bob’s measurement setting, \( \theta_B \in \{0, \pi/4\} \), and the corresponding outcome is recorded in \( j \in \{1, 2, \emptyset\} \). Precisely, Bob sets \( j = 1 \) (\( j = 2 \)) if a click is observed in the detector that detects the horizontal (vertical) component of the incoming pulse and \( j = \emptyset \) if no click is observed. If both detectors click, the outcome is randomly assigned to \( j = 1 \) or \( j = 2 \).

detection probability and the probability of having a bit error with amplitude \( |\alpha| \), given that both parties selected the same measurement setting (i.e., Z or X). We denote these parameters by \( Q^\alpha \) and \( E^\alpha \), respectively, and they are basis-independent in the considered channel model. In particular, explicit calculation of \( Q^\alpha \) and \( E^\alpha \) using this model yields

\[
Q^\alpha = 1 - (1 - p_d)^2 e^{-\eta |\alpha|^2}, \\
Q^\alpha E^\alpha = \frac{p_d^2}{2} + p_d(1 - p_d)(1 + h_{\eta,\alpha,\delta_A}) + (1 - p_d)^2 \left( \frac{1}{2} + h_{\eta,\alpha,\delta_A} - \frac{1}{2} e^{-\eta |\alpha|^2} \right)
\]

where \( \eta = \eta_{\text{det}}\eta_{\text{ch}} \) and we defined \( h_{\eta,\alpha,\delta_A} = (e^{-\eta |\alpha|^2} \cos^2(\delta_A) - e^{-\eta |\alpha|^2} \sin^2(\delta_A)) / 2 \). These expressions account for the fact that multiple clicks are randomly assigned to a specific detection outcome (see the caption of Fig. 5 for more details).

Finally, we write down the expected values of the observables required for the simulations. For this purpose, we introduce the quantities \( G^a_{Z,Z} = q_2^2 p_a Q^{\sqrt{2}}, \quad G^a_{X,X} = q_2^2 p_a Q^{\sqrt{2}}, \quad E^a = E^{\sqrt{2}} \) where \( a \in A \) and \( p_a \) is the probability that Alice uses the intensity setting \( a \). From these quantities, it follows that

\[
E[|Z^a_j|] = \frac{G^a_{Z,Z}}{\sum_a G^a_{Z,Z}} M, \\
E[|X^a_j|] = G^a_{X,X} N, \quad \text{and} \quad E[e^a_j] = \hat{E}^a G^a_{X,X} N,
\]

where we recall that \( N \) is the number of signals transmitted per QKD pair and \( M \) is the size of each sifted key. Also note that, for each \( j = 1, \ldots, n_q \) all three sets \( Z^a_j \) contribute to the \( j \)-th sifted key. Thus, averaging over all three intensity settings, the expected QBER in the basis \( Z \) is

\[
E_Z = \frac{\sum_a E^a G^a_{Z,Z}}{\sum_a G^a_{Z,Z}}.
\]

Remarkably, the formula above corresponds to the \( a \) priori expected bit error rate between any pair of sifted keys, i.e., the expected error rate without using the knowledge of the actual set sizes \( |Z^a_j| \). The knowledge of the set sizes indeed provides slightly more accurate values of the expected bit error rates, but these would be different for each \( j \). Thus, for simplicity, we use the common \( a \) priori expected bit error rate for all \( j \).

7. Performance evaluation

In Fig. 6, we plot the secret key rate that one can extract combining Protocol with the decoy-state QKD scheme presented in Sec. F1 of this Supplementary Information, as a function of the channel loss between Alice and Bob.
Both the security and the experimental parameters are set following the criteria described in Sec. III of the main text. That is, they are common with the simulations of MDI-QKD presented in Fig. 2 of the main text. As in that figure, two different block sizes are considered, (a) $M = 10^5$ and (b) $M = 10^6$, and various distinct adversarial scenarios are included. The reader is referred to Sec. III of the main text for a discussion of the results presented in Fig. 6 (such discussion is common with that of Fig. 2 in the main text).

![Graph](image)

**FIG. 6:** Secret key rate, $K$, that results from the decoy-state BB84 protocol with redundant devices presented in Sec. F 1 of this Supplementary Information. Two distinct block-sizes are considered, (a) $M = 10^5$ and (b) $M = 10^6$. In each case, $K$ is plotted as a function of the channel loss between Alice and Bob for various adversarial scenarios with malicious devices. In both figures, the purple line is the secret key rate in the standard scenario where each party holds a trusted QKD module and a trusted classical post-processing (CP) unit. On the contrary, green lines are used for different adversarial scenarios. Precisely, the dashed-dotted phosphorescent line is the secret key rate assuming that the corrupted devices are passive and non-collaborative, which requires the use of two QKD pairs and two CP units per lab (all of them being possibly malicious) to provide security. Meanwhile, the solid non-phosphorescent green lines assume active and collaborative corrupted devices. These latter lines further assume the same number, say $t$, of malicious QKD pairs and malicious CP units per lab, which requires the use of at least $n_q = t + 1$ QKD pairs and $n_c = 3t + 1$ CP units per party to provide security. Specifically, the dark (light) green line corresponds to $t = 3$ ($t = 5$).

**Appendix G: Inverse Chernoff bound**

Here, we rephrase the statement of the inverse Chernoff-bound presented in [51, 52].

Let $X_1, \ldots, X_N$ be independent Bernouilli random variables such that $P[X_i = 1] = p_i$, and let $X = \sum_{i=1}^{N} X_i$ and $\mu = E[X] = \sum_{i=1}^{N} p_i$, where $E[\cdot]$ denotes the expected value. Let $x$ be the observed outcome of $X$ for a given trial (that is, $x \in \mathbb{N}$). Then, $x$ satisfies

$$x = \mu + \delta$$  \hspace{1cm} (G1)

except with probability at most $\epsilon_L + \epsilon_U$, where the parameter $\delta \in [-\Delta(x, \epsilon_L), \hat{\Delta}(x, \epsilon_U)]$ and

$$\hat{\Delta}(x, y) = x \left[ W_0(\text{e}^{-c_{x,y}}) + 1 \right]$$

$$\Delta(x, y) = \begin{cases}  
-x \left[ W_{-1}(\text{e}^{-c_{x,y}}) + 1 \right] & \text{if } x \neq 0, \\
\ln y^{-1} & \text{if } x = 0.
\end{cases}$$  \hspace{1cm} (G2)

Here, $W_j$ stands for the $j$-th branch of the $W$ Lambert function and $c_{x,y}$ is defined as $c_{x,y} = 1 + \ln y^{-1}/x$. Also, $\epsilon_L$ and $\epsilon_U$ are the one-sided error probabilities.
Appendix H: Generalised chain rule for conditional smooth min-entropies

The first chain rule for conditional smooth min-entropies given in [40] can be restated as follows: for all $\epsilon_2, \epsilon_1 \geq 0$, $\epsilon_2 > 2\epsilon_2' + \epsilon_1$,

$$H_{\min}^{\epsilon_2'}(Z_2Z_1|E) \geq H_{\min}^{\epsilon_2'}(Z_2|Z_1) + H_{\min}^{\epsilon_1'}(Z_1|E) - \log_2 \left( \frac{1}{\epsilon_2 - 2\epsilon_2' - \epsilon_1} \right). \quad \text{(H1)}$$

Using mathematical induction, the previous claim is easily generalised to

$$H_{\min}^n(Z_n \ldots Z_1|E) \geq \sum_{j=2}^{n} \left[ H_{\min}^{\epsilon_j'}(Z_j|Z_{j-1} \ldots Z_1) - \log_2 \left( \frac{1}{\epsilon_j - 2\epsilon_j' - \epsilon_{j-1}} \right) \right] + H_{\min}^{\epsilon_1'}(Z_1|E), \quad \text{(H2)}$$

for all $\epsilon_j, \epsilon_j'$ such that

$$\epsilon_1 \geq 0, \{\epsilon_j \geq 0, \epsilon_j > 2\epsilon_j' + \epsilon_{j-1}\}_{j=2}^{n} \quad \text{(H3)}$$

and $n \geq 2$. We prove it in what follows. First, the case $n = 2$ trivially holds, as it reduces to Eq. (H1). Let us now assume that the proposition holds for a specific $n = m$ larger than two and consider the case $n = m + 1$. Again, from Eq. (H1)

$$H_{\min}^{\epsilon_{m+1}}(Z_{m+1} \ldots Z_1|E) \geq H_{\min}^{\epsilon_{m+1}}(Z_{m+1}|Z_m \ldots Z_1) + H_{\min}^{\epsilon_m}(Z_m \ldots Z_1|E) - \log_2 \left( \frac{1}{\epsilon_{m+1} - 2\epsilon_{m+1}' + \epsilon_{m}} \right), \quad \text{(H4)}$$

for all $\epsilon_{m+1}', \epsilon_m \geq 0, \epsilon_{m+1} > 2\epsilon_{m+1}' + \epsilon_{m}$. Note that the above equation is simply a recasting of Eq. (H1). Precisely, $Z_2$ and $Z_1$ in the left-hand side of Eq. (H1) are respectively replaced by $Z_{m+1}$ and the multivariable $Z_m \ldots Z_1$.

The second term in the right-hand side of Eq. (H4) can be lower-bounded using the induction hypothesis and the proposition for $n = m + 1$ follows, which completes the proof.

In what follows, we deduce a restricted version of Eq. (H2) that can be applied in the derivation of the extractable key length with passive non-collaborative eavesdroppers. Precisely, let us assume the following simplifications:

1. For $j = 2, \ldots, n$, $H_{\min}^{\epsilon_j'}(Z_j|Z_{j-1} \ldots Z_1) = H_{\min}^{\epsilon_j'}(Z_j|E)$ for all $\epsilon_j'$.
2. $H_{\min}^{\epsilon_1'}(Z_1|E) = 0$ for all $\epsilon_1$.
3. $\epsilon_1 = 0$ and $\epsilon_j = \epsilon_{j-1} + 2\epsilon_j' + \delta$ for $j = 2, \ldots, n$ and $\delta > 0$.
4. $\epsilon_j' = \epsilon$ for $j = 2, \ldots, n$ and $\epsilon > 0$.

Note that assumptions 3 and 4 are such that the conditions given in Eq. (H3) trivially hold and $\epsilon_n = (n-1)(2\epsilon + \delta)$. From 1 to 4, Eq. (H2) easily reduces to

$$H_{\min}^{(n-1)(2\epsilon + \delta)}(Z_n \ldots Z_1|E) \geq \sum_{j=2}^{n} H_{\min}^{\epsilon_j'}(Z_j|E) - \log_2 \left( \frac{1}{\delta(n-1)} \right), \quad \text{(H5)}$$

with $\epsilon, \delta > 0$ and $n \geq 2$.

[1] Bennett, C. H. & Brassard, G. Quantum cryptography: public key distribution and coin tossing. In Proc. IEEE International Conference on Computers, Systems & Signal Processing, 175–179 (IEEE, NY, Bangalore, India, 1984).
[2] Scarani, V., Bechmann-Pasquinucci, H., Cerf, N. J., Dušek, M., Lütkenhaus, N. & Peev, M. The security of practical quantum key distribution. Reviews of Modern Physics 81, 1301 (2009).
[3] Lo, H.-K., Curty, M. & Tamaki, K. Secure quantum key distribution. Nature Photonics 8, 595 (2014).
[4] Xu F., Ma X., Zhang Q., Lo, H.-K., Pan J.-W. Secure
quantum key distribution with realistic devices. Reviews of Modern Physics 92, 025002 (2020).
[5] Diffie, W. & Hellman, M. New directions in cryptography. IEEE Transactions on Information Theory 22, 644-654 (1976).
[6] Rivest, R. L., Shamir, A. & Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120-126 (1978).
[7] Gligor, V. D. A guide to understanding covert channel analysis of trusted systems. Vol. 30 (National Computer Security Center, 1994).
[8] Zander, S., Armitage, G. & Branch, P. A survey of covert channels and countermeasures in computer network protocols. IEEE Communications Surveys & Tutorials 9, 44-57 (2007).
[9] Prevelakis, V. & Spinellis, D. The Athens affair. IEEE Spectrum 4, 26-33 (2007).
[10] Yang, K., Hicks, M., Dong, Q., Austin, T. & Sylvester, D. A2: Analog malicious hardware. In IEEE symposium on security and privacy 18-37 (IEEE, 2016).
[11] Robertson, J. & Riley, M. The big hack: how China used a tiny chip to infiltrate US companies. Bloomberg Businessweek 34-39 (2008).
[12] Adee, S. The hunt for the kill switch. IEEE Spectrum 45, 34-39 (2008).
[13] Becker, G. T., Regazzoni, F., Paar, C. & Burleson, W. P. Stealthy dopant-level hardware trojans. In International Workshop on Cryptographic Hardware and Embedded Systems, 197-214 (Springer, Berlin, Heidelberg, 2013).
[14] Li, W. Experimental quantum key distribution with imperfect apparatus. In Proceedings of the 29th Annual Symposium on Foundations of Computer Science, 503-509 (1998).
[15] Acín, A., Brunner, N., Gisin, N., Massar, S., Pironio, S. & Scarani, V. Device-independent security of quantum cryptography against collective attacks. Physical Review Letters 98, 230501 (2007).
[16] Vazirani, U. & Vidick, T. Fully device-independent quantum key distribution. Physical Review Letters 113, 140501 (2014).
[17] Arnon-Friedman, R., Dupuis, F., Fawzi, O., Renner, R. & Vidick, T. Practical device-independent quantum cryptography via entropy accumulation. Nature Communications 9, 459 (2018).
[18] Miller, C. A. & Shi, Y. Robust protocols for securely expanding randomness and distributing keys using untrusted quantum devices. Journal of the ACM 63, 33 (2016).
[19] Barrett, J., Colbeck, R. & Kent, A. Memory attacks on device-independent quantum cryptography. Physical Review Letters 110, 010503 (2013).
[20] Curty, M. & Lo, H.-K. Foiling covert channels and malicious classical post-processing units in quantum key distribution. npj Quantum Information 5, 14 (2019).
[21] Chor, B., Goldwasser, S., Micali, S. & Awerbuch, B. Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proc. of the 26th Annual Symposium on Foundations of Computer Science (FOCS’85), 383-395 (IEEE Computer Society, Los Alamitos, California, 1985).
[22] Cramer, R., Damgård, I. B. & Nielsen, J. B. Secure Multi-party Computation and Secret Sharing (Cambridge University Press, New York, USA, 2015).
[23] Ben-Or, M., Goldwasser, S. & Wigderson, A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In Proceedings of the twentieth annual ACM symposium on Theory of computing, 1-10 (1988).
[24] Chaum, D., Crépeau, C. & Damgard, I. Multiparty unconditionally secure protocols. In Proceedings of the twentieth annual ACM symposium on Theory of computing, 11-19 (1988).
[25] Maurer, U. Secure multi-party computation made simple. Discrete Applied Mathematics 154, 370-381 (2006).
[26] Shamir, A. How to share a secret. Communications of the ACM 22, 612-613 (1979).
[27] Blakley, G. R. Safeguarding cryptographic keys. In Proc. of the AFIPS 1979 National Computer Conference (NCC’79), 313-317 (AFIPS Press, New Jersey, 1979).
[28] Li, W. & Zbinden, H. Concise security bounds for practical decoy-state quantum key distribution. Nature Communications 4, 893 (2013).
[29] Vitanov, A., Dupuis, F., Tomamichel, M. & Renner, R. Leftover hashing against quantum side information. Physical Review A 81, 022307 (2014).
[30] Zhou, Y. H., Yu, Z. W. & Wang, X.-B. Making the Tokyo QKD Network in Vienna. Optics Express 19, 10387-10409 (2011).
[31] Fung, C. H. F., Ma, X. & Chau, H.-F. Practical issues in quantum-key-distribution postprocessing. Physical Review A 81, 012318 (2010).
[32] Ben-Or, M., Goldwasser, S. & Wigderson, A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In Proceedings of the twentieth annual ACM symposium on Theory of computing, 1-10 (1988).
[33] Chaum, D., Crépeau, C. & Damgard, I. Multiparty unconditionally secure protocols. In Proceedings of the twentieth annual ACM symposium on Theory of computing, 11-19 (1988).
[34] Maurer, U. Secure multi-party computation made simple. Discrete Applied Mathematics 154, 370-381 (2006).
[35] Shamir, A. How to share a secret. Communications of the ACM 22, 612-613 (1979).
[36] Blakley, G. R. Safeguarding cryptographic keys. In Proc. of the AFIPS 1979 National Computer Conference (NCC’79), 313-317 (AFIPS Press, New Jersey, 1979).
[37] Li, W. & Zbinden, H. Concise security bounds for practical decoy-state quantum key distribution. Nature Communications 4, 893 (2013).
[38] Vitanov, A., Dupuis, F., Tomamichel, M. & Renner, R. Leftover hashing against quantum side information. Physical Review A 81, 022307 (2014).
[39] Zhou, Y. H., Yu, Z. W. & Wang, X.-B. Making the Tokyo QKD Network in Vienna. Optics Express 19, 10387-10409 (2011).
[40] Fung, C. H. F., Ma, X. & Chau, H.-F. Practical issues in quantum-key-distribution postprocessing. Physical Review A 81, 012318 (2010).
[41] Lim, C. C. W., Curty, M., Walenta, N., Xu, F. & Zbinden, H. Concise security bounds for practical decoy-state quantum key distribution. Physical Review A 89, 022307 (2014).
[42] Yin, H. L. et al. Measurement-device-independent quantum key distribution over a 404 km optical fiber. Physical Review Letters 117, 190501 (2016).
[43] Renner, R. Security of quantum key distribution. International Journal of Quantum Information 6, 1-127 (2008).
[44] Vitanov, A., Dupuis, F., Tomamichel, M. & Renner, R. Chain rules for smooth min-and max-entropies. IEEE Transactions on Information Theory 59, 2603-2612 (2013).
[45] Krawczyk, H. LFSR-based hashing and authentication. In Advances in Cryptology - CRYPTO’94, Lecture Notes in Computer Science (Springer-Verlag, 1994) 893, 129-139.
[46] Lo, H.-K., Curty, M. & Qi, B. Measurement-device-independent quantum key distribution. Physical Review Letters 108, 130503 (2012).
[43] Curty, M., Xu, F., Cui, W., Lim, C. C. W., Tamaki, K. & Lo, H.-K. Finite-key analysis for measurement-device-independent quantum key distribution. *Nature Communications* **5**, 1-7 (2014).

[44] Tomamichel, M., Lim, C. C. W., Gisin, N. & Renner, R. Tight finite-key analysis for quantum cryptography. *Nature Communications* **3**, 634 (2012).

[45] Serfling, R. J. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 39-48 (1974).

[46] Mitzenmacher, M. & Upfal, E. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis* (Cambridge University Press, 2017).

[47] Hwang, W. Y. Quantum key distribution with high loss: toward global secure communication. *Physical Review Letters* **91**, 057901 (2003).

[48] Lo, H.-K., Ma, X. & Chen, K. Decoy state quantum key distribution. *Physical Review Letters* **94**, 230504 (2005).

[49] Wang, X. B. Beating the photon-number-splitting attack in practical quantum cryptography. *Physical Review Letters* **94**, 230503 (2005).

[50] Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* **58**, 13–30 (1963).

[51] Zhang, Z., Zhao, Q., Razavi, M. & Ma, X. Improved key-rate bounds for practical decoy-state quantum-key-distribution systems. *Physical Review A* **95**, 012333 (2017).

[52] Bahrani, S., Elmabrok, O., Lorenzo, G. C. & Razavi, M. Wavelength assignment in quantum access networks with hybrid wireless-fiber links. *Journal of the Optical Society of America B* **36**, 99-108 (2019).