MUTLISS: a protocol for long-term secure distributed storage over multiple remote QKD networks*

Thomas Prévost[1], Olivier Alibart[2], Anne Marin[3], and Marc Kaplan[3]

[1] Université Côte d'Azur, CNRS, Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S), UMR 7271, Sophia-Antipolis 06900, France
{thomas.prevost}@univ-cotedazur.fr
[2] Université Côte d'Azur, CNRS, Institut de Physique de Nice (INPHYNI), UMR 7010, Nice 06200, France
olivier.alibart@univ-cotedazur.fr
[3] VeriQloud
{marin,kaplan}@veriqloud.fr

Abstract. We introduce MULTISS, a new distributed storage protocol over multiple remote Quantum Key Distribution (QKD) networks that ensures long-term data confidentiality. Our protocol extends LINCOS, a secure storage protocol that uses Shamir secret sharing to distribute data in a single QKD network. Instead MULTISS uses a hierarchical secret scheme that makes certain shares mandatory for the reconstruction of the original secret. We prove that MULTISS ensures that the stored data remain secure even if an eavesdropper (1) gets full access to all storage servers of some of the QKD networks or (2) stores and breaks later all the classical communication between the QKD networks. We demonstrate that this is strictly more secure than LINCOS which is broken as soon as one QKD network is broken.

Our protocol, like LINCOS, has a procedure to update the shares stored in each QKD network without reconstructing the original data. In addition, we provide a procedure to recover from a full compromission of one of the QKD network. In particular, we introduce a version of the protocol that can only be implemented over a restricted network topologies, but minimizes the communication required in the recovery procedure.

In practice, the MULTISS protocol is designed for the case of several QKD networks at the metropolitan scale connected to each other through channels secured by classical cryptography. Hence, MULTISS offers a secure distributed storage solution in a scenario that is compatible with the current deployment of quantum networks.

* This work has been conducted within the framework of the French government financial support managed by the Agence Nationale de la Recherche (ANR), within its Investments for the Future programme, under the Stratégie Nationale Quantique through the project of the PEPR-quantum QComtestbeds (ANR 22-PETQ-0011) and with fundings from the EUROPE HORIZON-FPA project QSNP and the EUROPE DIGITAL project FranceQCI.

Keywords: Decentralised · Quantum-proof · Secure storage · Information Theoretic Secrecy · Long-term · QKD · Birkhoff interpolation.

Introduction

Some information remain sensitive over several decades, thus requiring both long-term storage protection and availability. This is the case of medical data or certain industrial secrets, for example. Current encryption algorithms, however, are not suitable for protecting long-term secrets, since cryptanalysis and computation power evolves rapidly over time. We know, for example, that public key encryption algorithms used today, such as RSA or ECC, will be vulnerable to quantum computer attacks [6, 32]. We also can't exclude the fact that "post-quantum" public key encryption algorithms becomes vulnerable to cryptanalysis in the future [18]. This happened for example with the DES cipher, standard for symmetric cryptography until 2000, which has shown itself to be vulnerable to linear and differential cryptanalysis [7, 21]. It is therefore to be feared that malicious actors are listening to and recording confidential communications, in the hope of deciphering them, once they have acquired larger computing power or more efficient cryptanalysis. This type of attack is called "*harvest now, decrypt later*" (HNDL) [22].

Claude Shannon introduced the principle of *Information-Theoretic Security* (ITS) in 1949 [27]. This is a "perfect" encryption model, or "unconditionally secure", that is to say resistant regardless the computing power of the attacker. One-Time Pad (OTP) encryption is an ITS encryption method. The ITS property can be understood as follows: "it is as difficult for an attacker to find the encryption key as it is to randomly guess the original message". To ensure long-term security of our secret, we need a protocol that can provide ITS protection throughout its lifetime.

It is well known that ITS requires the encryption key to be as long as the message itself and perfectly random. This makes the key distribution difficult to scale in practice. The usual method used to transmit a key between two entities that have not physically met before is based on public key. However, these cryptographic primitives are not ITS. *Quantum Key Distribution* (QKD) [3] on the other hand proves to be an alternative for transmitting encryption keys between two parties in an ITS manner. Security is no longer guaranteed by the assumed computational hardness for an attacker to compute the key, but by the possibility of detecting any adversary trying to eavesdrop on the key exchange. Eventually, combined with OTP encryption, HNDL attacks are no longer possible over channels secured by QKD.

The LINCOS protocol [11] stores secret data distributed over nodes on a network whose communication is secured with QKD. The data are encrypted using Shamir's secret sharing [26] which splits them in several shares that individually contain no information about the original data. Reconstructing the data requires to recover sufficiently many shares, the exact number being a parameter

of the scheme called the *threshold*. Notice that unlike block cipher encryption for example, Shamir secret sharing is keyless.

In LINCOS, the confidentiality of the data is achieved by using Shamir Secret Sharing and distributing the shares with communication secured by QKD. Since Shamir's Secret Sharing itself is ITS [13], the protocol ensures that the secret remains ITS throughout the process. More precisely, since the communication links are protected with QKD, an eavesdropper that wants to learn the secret data needs to attack several storage. In the context of the development of QKD networks all over the world, the LINCOS protocol has been implemented in Europe [1] and Asia [15] on several QKD testbeds.

The LINCOS protocol is limited by the maximal distance of QKD. Due to fiber losses, a QKD link cannot exceed a few hundred kilometers. One practical approach is to make use of trusted nodes to route keys over longer distances, but this impacts the infrastructure cost and downgrades the security by revealing encryption keys to intermediate nodes, inducing weaknesses to the system. In practice, most QKD demonstrations in Europe by telecom operators remain at metropolitan area-scale [23, 25].

For this reason, previous deployments of LINCOS were performed at the metropolitan scale. We consider this as a weakness of the protocol. In this situation, the nodes might be operated by the same entity and for a hacker taking control of one of the servers in the network, it might be easier to extend its control to other ones. Additionally, a government could perform legal interception, seizing the data from all storage servers located in its country to recover the secret.

Our motivation is to strengthen the security of distributed storage using multiple remote QKD networks, hence increasing the distance and the number of nodes, while taking into account the practical limitations and non availability of QKD links over some of the sections. Some sections between those remote QKD networks are secured with classical cryptography only, which tends to indicate that those links are the weakest point of the overall security architecture.

We overcome those limitations with MULTISS. Our protocol replaces the Lagrange interpolation of Shamir with Birkhoff interpolation [8]. This results in a hierarchical secret sharing scheme. The main advantage is to preserve the ITS property of confidentiality even though the links between the QKD networks are secured by classical cryptography. Our protocol maintains the confidentiality against an attacker that either compromises some QKD networks, or stores all classical communication to perform Harvest Now Decrypt Later attacks. This is achieved by the specific combination of Shamir and Birkhoff that we devise for this quantum network structure.

In addition, we retain an important feature of LINCOS to renew the shares of each node periodically without reconstructing the initial secret. In LINCOS, this procedure allows to recover from the leakage of some of the nodes in the QKD network. Executing the share update procedure makes previously leaked shares obsolete. In practice, that forces an adversary to learn sufficiently many shares within the same update period.

MULTISS is using several QKD networks specifically to prevent an adversary to learn the original secret by fully corrupting a single QKD network. Furthermore, we introduce a procedure to recover from a full leakage of one of the QKD networks. This procedure updates the *value* stored in the network, and not only the shares stored in each server. This subnet value update procedure, however, requires some communication between the QKD networks. In order to minimize it, we introduce a special version of the MULTISS protocol called the *local mode*, which is valid only for specific network topologies. The local mode of MULTISS greatly simplifies the subnet value update by requiring only the communication between two of the networks whereas the standard mode requires communication between with all the subnets.

The paper is structured as follows. In Section 1, we introduce Quantum Key Distribution (QKD), Shamir's Secret Sharing (SSS) and its extension to hierarchical secret sharing. In Section 2, we give a high level description of LINCOS, followed, in Section 3 by the details of MUTLISS. We also specify our threat model and provide a security analysis. Finally, we conclude on future perspectives of our protocol in realistic settings.

1 Preliminaries

1.1 Quantum Key Distribution

No technical detail about quantum key distribution is required to understand our protocol. We nevertheless recall its main features, which will be useful for the security analysis of MULTISS.

In classical cryptography, the transmission of secrets between two remote entities is usually secured through a combination of public key cryptography and symmetric cryptography. In public key cryptography, there is not one but two keys, the public key to encrypt the data and the private key to decrypt it. Security then relies on the computational hardness, for an attacker who knows the public key, to find the private key, or to extract the clear text from the ciphers. In practice, public key cryptography is used to established a shared secret session key that is then used with symmetric cryptography.

Both public key and private key cryptography rely on computational assumptions, hence they are saif *computationally secure*. As computing power and cryptanalysis increase over time, it is very hard to guarantee the security of the data for a long-time based on such assumptions.

On the other hand, the security of quantum key distribution is based on a fundamental principle of quantum physics, the no-cloning theorem [33]. This theorem states that it is impossible to perfectly clone the state of an arbitrary qubit, the basic unit of representation of quantum information, without modifying its state.

QKD protocols work by sending qubits from one participant to another, either directly as in the protocol BB84 [4], or through an intermediate source which shares entangled qubits to both participants in the protocol E91 [14].When an

adversary attempts to intercept the qubits, it necessarily modifies their state, thus allowing honest participants to detect it, and then interrupt the key establishment protocol. Indeed, honest participants regularly send a sample of qubit measurement results, and can thus detect any abnormal modification of their states. Since the adversary remains with no information about the established key, QKD guarantees Information Theoretic Security as long as the legitimate parties share an authenticated channel. While this theoretical argument can be made formal, in practice, just as classical systems, QKD may be vulnerable to side-channel attacks which exploit the imperfections in the hardware [20, 2, 5].

Authenticating honest participants is necessary to prevent *Man In The Middle* (MiTM) attack. In the classical world, this is usually carried out with public key cryptography. ITS authentication is possible using pre-shared keys. Otherwise, QKD remains vulnerable against adversaries that can break the authentication during the execution of the key exchange protocol. This results in an interesting security model named *everlasting security* [30]: if the adversary is not able to break the protocol at the exact moment of its execution, it cannot extract any information later about the keys, so that the confidentiality of the secret is guaranteed forever. It is therefore possible to protect a secret with ITS security, by combining QKD, one-time-pad encryption and authentication based on pre-shared keys, but using public-key cryptography for authentication results in everlasting security, which is still more secure than the security achieved with classical cryptography only.

In the rest of the paper, we will use the expressions *QKD links* to refer to the quantum communication channels implementing the QKD protocol, and to *ITS links* to the combined quantum and classical channels that implement QKD, One-time-pad encryption and ITS authentication. Regarding the networks, we use ITS and QKD interchangeably. This means that a QKD network consists in both a quantum and a classical communication network implementing all together QKD, One-time-pad and ITS authentication.

The quantum state of a single photon is the usual support of flying qubits, used for quantum communication. This can be for example the polarization or the time bin (the exact moment at which the photon arrives being a probabilistic event in quantum physics). The majority of implementations use optical fibers to transmit qubits, with a range limited by fiber loss rate. Using dark fibers to limit losses and noise, QKD only reaches a few hundred kilometers only [10, 23–25]. These constraints currently restricts QKD to metropolitan distances without repeaters, who are notoriously hard to build for quantum information [17, 19].

1.2 Shamir's Secret Sharing Scheme

Secret sharing is a cryptographic primitive discovered independently by Adi Shamir [26] and George Blakley [9] in 1979. It allows a person, the *dealer*, to distribute a secret among n participants. The dealer defines a threshold k of participants who must pool their shares in order to find the initial secret.

We use the Shamir Secret Sharing (SSS) scheme, which works as follows: define $S \in \mathbb{N}$ as the initial secret, $n \in \mathbb{N}^*$ the number of participants and $k \in \mathbb{N}^*$ the decrypting *threshold*. In Shamir Secret Sharing, this value satisfies:

- Any set of k participants can recover the secret,
- A set of $k-1$ participants has no information about the secret.

The dealer starts by arbitrarily choosing a prime number p such that $p > S$. From now, all algebraic operations are supposed to be in $\mathbb{F}_p$. Then the dealer generates a random polynomial $P \in \mathbb{F}_p[X]$ of degree $d(P) = k-1$, so that $P(0) = S$, the initial secret. Finally, the dealer distributes to each of the n participants the evaluations of the polynomial $P(1), P(2), ..., P(n)$. It is possible for k participants among n to then pool their shares and thus find the initial secret $S = P(0)$ using Lagrange interpolation [31] whose details can be found in standard textbooks.

The confidentiality of the Shamir Secret Sharing is ITS [13]. This means that an attacker that learns at most $k-1$ shares cannot get any information about the secret even with an unbounded computational power.

The protocols we consider, LINCOS and MULTISS, are executed on real network architectures. The dealer is called the *document owner*, and the participants are called the *shareholders*, which in practice consist in storage servers. In practical implementations, the document owner might use a proxy to compute the shares and distributes them.

Consider a network N that consists of n servers $serv_1, \ldots, serv_n$. We say that the network N stores a value S using a local Shamir Secret Sharing with polynomial P when $serv_i$ is storing $P(i)$ for $i = 1, \ldots, n$. When there is not ambiguity, we simply write that the network N stores the value S.

In order to simplify the notations, we denote the decryption threshold $T(P) = d(P)+1$. This simplifies the expressions of thresholds of the MULTISS protocol.

In our protocol, we assume that the dealer is honest. However, there exists methods for participants to verify the integrity of the secret if they do not trust the dealer [12].

1.3 Hierarchical secret sharing

It is possible to extend Shamir's Secret Sharing Scheme to introduce a notion of participant hierarchy [28]. Consider a company whose CEO would like to share a secret among his employees. In addition to a threshold of employees for the decryption of the secret, he would want to impose the presence an employee of a certain category, for example a manager, among them to achieve the decryption. This can be achieved using hierarchical secret sharing.

Technically, this works as follows: let $n \in \mathbb{N}$ be the number of participants, $k \in \mathbb{N}^*$ the decryption threshold, $m < k$ the number of mandatory shares (the shares held by participants considered as managers), S the initial secret to protect. The dealer chooses a prime number $p > S$, and generates a random polynomial $P \in \mathbb{F}_p$ of degree $k-1$ such that $S = P(0)$. The dealer shares the

evaluations $P(1), \dots P(m)$ only to the managers. He then calculates the derivative polynomial P'. For

$$P(X) = a_{k-1}X^{k-1} + \ldots + a_1X + a_0 \tag{1}$$

with a_{k-1}, ... a_1, $a_0 \in \mathbb{F}_p$, it is

$$P'(X) = (k-1)a_{k-1}X^{k-2} + \ldots + a_1. \tag{2}$$

The dealer distributes to the non-manager employees the evaluations of the derivative polynomial $P'(m+1), \dots, P'(n)$. Thus, even if all non-manager employees pooled their shares, they would only be able to recover the derivative polynomial P', and thus could never compute $P(0)$. It is however possible, with the help of a manager who has an evaluation of the initial polynomial, to recover the secret $\mathrm{S} = \mathrm{P}(0)$ by Birkhoff interpolation [8]. As Shamir secret sharing, hierarchical secret sharing is ITS [29].

In the rest of the paper, we use this scheme with $m = 1$. In other words, there is only one share mandatory for reconstructing the secret.

2 LINCOS protocol

Since the MULTISS protocol is an extension of LINCOS [11], we propose here a short reminder of its main features. . The protocol is composed of two main procedures:

- The protocol guaranteeing authenticity and long-term integrity, *COPRIS*,
- The protocol guaranteeing long-term confidentiality.

Fig. 1 gives a diagram of the different parties involved in the LINCOS protocol. Notice that we extend the confidentiality guarantees of LINCOS, but we don't modify the COPRIS part for authenticity and integrity, which remain exactly the same.

LINCOS can be used to save secret documents over several storage servers using Shamir Secret Sharing. Each server receives a share, and the distribution uses ITS links. Since both QKD, OTP and SSS are ITS, the resulting combination of those remains ITS. This ensures the long-time security of distributed storage even against an adversary with unbounded computational power, including of course the use of quantum computers. We now present the protocol in details.

2.1 Long-time confidential secret storage

The confidentiality protocol requires the existence of ITS secure channels between the document owner and a network of nodes, the *Shareholders*. The ITS link proposed in the paper is a one-time pad with a key established by QKD. The document owner and the shareholder exchange keys κ as long as the message

m they want to exchange. The ciphertext c sent by the document owner to the shareholder is protected using a one-time pad with key κ:

$$c = \kappa \oplus m, \tag{3}$$

where $\oplus$ is the bitwise XOR of the messages considered as bitstrings. One-time pad guarantees ITS confidentiality provided that the key κ is as long as the message and perfectly random.

The confidentiality of LINCOS works as follows: the document owner starts by generating as many Shamir shares of his secret document S as there are shareholder nodes in the network, using a polynomial P. The decryption threshold $T(P) = d(P) + 1$ is chosen by the document owner, depending on the targeted tradeoff between security and server availability. A low threshold increases the risk of collusion between several malicious nodes (for example in case of hacking), while a higher threshold increases the risk of unavailability in case of failure of a part of the network. The document owner sends to each of the shareholders a share of the secret (i.e. $P(1), P(2)$ etc.) via ITS communication links (that is, communication links secured by QKD and One-time-pad).

Since Shamir Secret Sharing is ITS, the confidentiality protocol is itself ITS. The security proof of LINCOS can be found in the original paper [11].

Share update Being able to regularly renew shares is an important feature of the LINCOS protocol, because it is possible for an attacker to corrupt temporarily some of the nodes and learn some of the shares. In order to update the shares of the different shareholder nodes without reconstructing the initial secret, LINCOS proposes the *Reshare* procedure. The procedure is described in [16]. It is based on the fact that Shamir Secret Sharing is *proactive*: it is possible to change the access structure of the scheme without reconstructing the original secret. In LINCOS, it is only used to update the shares without reconstructing the original secret.

In LINCOS, the document owner starts by generating a new random polynomial Q of the same degree as the initial polynomial P, such that $\delta(0) = 0$. It then distributes to each of the shareholders an evaluation of the polynomial $\delta(1)$, ..., $\delta(n)$. The shareholders add the evaluation of the received polynomial to the share it already owns, so that for node i, $new\_share = P(i) + \delta(i)$. The shareholder node can then forget the old share, storing only new_share.

The shares have thus been renewed without altering the initial secret document, because $P(0)+Q(0) = P(0) = S$. Each time the document owner wants to renew the shares, it will ask the shareholders to add to their share the evaluation of a polynomial whose value at 0 is zero.

In practice, Reshare forces an eavesdropper that wants to learn S to break the security of at least k servers within the same update period. If the eavesdropper learns one of the shares, and those are then updated, then the share previously obtained becomes obsolete.

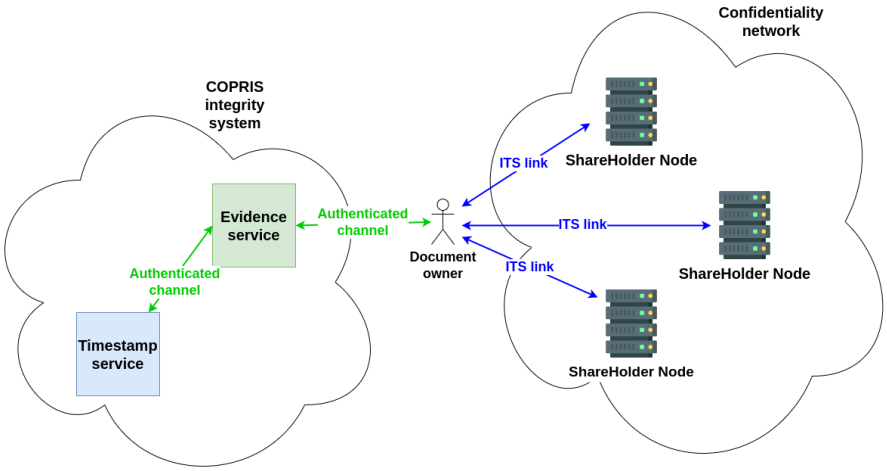

Fig. 1. Parties involved in the LINCOS protocol, on the left the COPRIS service for proof of integrity and authentication, and on the right the storage of the document in the form of Shamir shares between the different shareholder nodes. The MULTISS protocol only changes the second part.

2.2 COPRIS integrity and authenticity protocol

Although the MULTISS protocol doesn't add anything to LINCOS regarding integrity and authenticity, we briefly describe those here for completeness. The guarantees on integrity and authenticity are obtained with the COPRIS protocol. COPRIS allows the document owner to build a *Proof of Integrity* (PI) to prove that a document existed at time t, while keeping it secret.

As shown in 1, for this purpose, the document owner sends a commitment to an *Evidence Service*. The latter makes a timestamp request to a *Timestamp Service*. The evidence service then creates an evidence record E from the commitment c and the timestamp received.

The document owner regularly renews the commitment, i.e. it sends a new commitment to the evidence service. The latter then builds a new evidence record. In order to prove the integrity and authenticity of document S to the verifier, the document owner sends it document S as well as the pair (E, R), with E the latest evidence record and R the list of decommitments. The verifier can then ensure the existence of document S at time t.

3 The MULTISS protocol

The LINCOS protocol limits the sharing of secrets between shareholders of the same ITS network. In practice, a network whose nodes are linked by QKD is

often limited to a few hundred kilometers. For this reason, most current implementations of functional quantum networks are limited to metropolitan area size. This is the case of the Tokyo QKD [15] network on which the LINCOS protocol was first deployed. In Europe, the various EuroQCI networks are also considering, in their first stage, deployments at the metropolitan scale [23, 25].

Reaching long distance for QKD requires either using satellite or trusted nodes to route the keys. Both solutions are very costly. Moreover, trusted nodes decrease the overall security of the protocol, since these nodes have a clear view of the keys they are routing.

MULTISS distributes shares over several distant ITS networks, but the links between the ITS networks are not themselves ITS. This is the case when several metropolitan QKD networks are connected by a classic IP link, allowing at best only classical cryptography. This situation is compatible with most of current and short-term deployment of QKD networks in the world, which are planned a the metropolitan-area scale.

3.1 Assumptions and adversary model

Transmission of secrets to the MULTISS network The MULTISS protocol works over several distant ITS networks, linked together by classical communication links only. In practice, the ITS networks are QKD networks at the metropolitan-area scale, with a classical network secured by One-time-pad encryption and ITS authentication. The QKD networks could potentially be located on different continents, for example in Paris, Montreal and Tokyo. It is then considered that the document owner has only one ITS link to one of the subnetworks, as shown in Fig. 2. For example, he could have a direct QKD link with this network, have a pre-shared key of the same size as the document it wants to store, or travel personally with his secret document in a secure briefcase. The document owner can then only secure his communications with the other subnetworks using classical cryptography, that is, non-ITS.

Although classical links do not have ITS security, communications over these links are considered *perfectly authenticated*. In particular, we assume that the attacker does not break the classical cryptography to perform a MiTM attack during the execution of the protocol.

Compromission of an entire network by an adversary The fact that LINCOS distributes secret shares within nodes in the same area may introduce vulnerabilities against a sufficiently powerful adversary. For example, if the secret document is only shared between the nodes of the QKD network in the same jurisdiction, the secret can be fully recovered through legal interception. It the storage nodes are all operated by the same entity, they might not be fully independent. Therefore, a hacker that manages to penetrate one of the nodes of the QKD network will have less difficulty to penetrate the others.

We want to protect the secret against such an adversary that could take control of all the nodes of an ITS network. This implies the use of more nodes outside the ITS network considered in the LINCOS protocol.

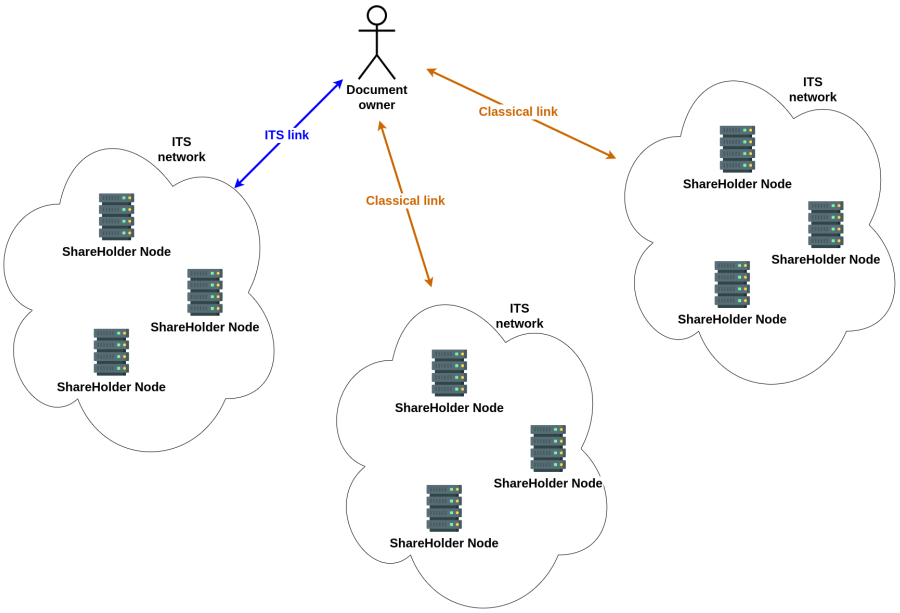

Fig. 2. In the MULTISS protocol, it is considered that the document owner can only transmit its secret document in an ITS manner to one of the ITS subnets. The link with the other subnets can only be secured with classical cryptography.

"Harvest now, decrypt later" (HNDL) adversary On the other hand, we want to guarantee the long-term confidentiality of the secret document. However, the transmission of information can only be ITS between the document owner and a single ITS network. The communication to other ITS networks can only be secured with classical cryptography, therefore non-ITS. If an adversary harvests all the classical communication, we want to ensure that the original secret remains secure in the future, even if the classical communication is decrypted by the adversary. In other words, the information transmitted by non-ITS channels must not be sufficient to recover the original secret.

Exclusivity of the two adversary models and subnet value renewal Our protocol is safe against an adversary that can learn the values stored in some of the QKD networks. To learn those values, the adversary needs to compromise sufficiently many storage servers since the communication is ITS. It is also safe against an adversary that can break the security of the classical links, for example by performing "Harvest now, decrypt later" attacks. We, however, exclude the case of an adversary that can perform both attacks. In order to break MULTISS, it suffices to learn the value stored in one of the QKD networks, and to break classical cryptography.

We believe that our security model remains relevant in practice. Even if the detection time for illegitimate network penetration is usually high, it remains much shorter than the expected time to break current classical cryptography. Similarly, if some shares are obtained through legal interception, it should be noticed by the server operator and after some time, by the document owner. In both cases, there is a gap of several years, or even decades, between the detection of the adversary and the decryption by this adversary of the secret document. As long as the adversary does not have cryptanalysis to break classical cryptography, he will not be able to discover the secret document.

We can exploit this gap to implement pro-active measures against the leakage of a value stored in a QKD network. The LINCOS protocol introduced a share update protocol that protects against some of the servers being compromised. It even recommends to apply this protocol periodically to pro-actively update the shares even if no leakages is detected. In MULTISS, we go further and propose a procedure to renew the values stored in a given ITS subnetwork if it is compromised. In standard mode, this essentially requires to restart the protocol and redistribute all shares over all subnets.

We also propose a local mode that minimizes the communication. It only requires communication between the mother network and the compromised subnet. This procedure can be executed when the document owner is notified that one subnetwork is compromised, but can also be performed pro-actively, to recover from a leakage even before it is discovered.

3.2 Description of the protocol

In the rest of this paper, we consider a secret S encoded as a digit. The set of polynomials it taken modulo p, with p a prime number greater than S. In practice, p is chosen by the document owner and publicly announced.

Network architecture The network architecture consists in ℓ subnetworks N_i. Each subnet contain a number of storage nodes that are connected one to each other with ITS links. The document holder also has an ITS link with one specific network called the "mother" subnetwork N_0. The communication with the "daughter" subnets $N_1, N_2, \dots N_\ell$ is purely classical and protected with classical cryptography. For subnetwork N_i, the number of nodes is denoted n_i.

The network architecture is symmetric, which means that the mother network can be different for different document owner. For example, if we operate three ITS subnets, in the cities of Paris, Tokyo, and Montreal, then a French user might use the Paris subnet as the mother subnet, and Tokyo and Montreal as the daughter subnets, while a Japanese user will use the Tokyo network as the mother subnet.

We give two versions of the MULTISS protocol. In the first version, called the *standard mode*, the distribution can be performed over multiple QKD networks with no constraint on the number of nodes. However, in the event where one of the QKD network is fully compromised, then the protocol must be fully restarted, with shares being redistributed over the whole network.

We introduce a variant of the protocol called the *local mode*. It is well defined only in the case where the number of daughter network is equal to the number of servers in the mother network, In this variant, if one of the subnetworks is compromised, it is possible to renew the value that it stores without changing the values in other subnetworks. This topology is shown in Figure 3. Notice that in order to make the number of subnetwork equal to the number of servers in the mother network, it is always possible to use only a subset of the subnets or a subset of the servers.

Thresholds In the LINCOS protocol, there is only one decryption threshold t which is, as in Shamir secret sharing, the number of shares needed to be pooled to recover the initial secret S. Conversely, t can also be interpreted as the compromission threshold: an adversary needs to compromise at least t shareholder nodes to recover the secret document S.

In the MUTLISS protocol, the secret document is distributed among several ITS networks using several polynomials, which induces a different definition of the threshold. In secret sharing, a threshold t means that any set of shares of size t suffices to recover the secret. This is the case in the LINCOS protocol, in which any number of shares allows to reconstruct the secret, given that this number is above the threshold.

In the local mode of MULTISS, the shares of a given daughter network are only useful when combined with one specific share of the mother network. For

this reason, the minimum number of shares required to reconstruct the secret is not a threshold as in secret sharing. Some sets of a given size allow to reconstruct the secret, whereas some other sets of the same size do not allow it. In particular, we notice that this makes it easier to perform fo example a denial-of-service attack, since erasing the share from one node of the mother network makes the shares of one of the subnetwork useless for the reconstruction.

We define the two following thresholds, t_{nodes} and $t_{networks}$.

- t_{nodes}: minimum number of nodes to learn in the whole network to recover the initial secret document.
- $t_{networks}$: minimum number of subnetworks to compromise to recover the secret document. We consider an ITS subnetwork "compromised" when the adversary can recover the value it stores.

As explained above, in local mode, the thresholds t_{nodes} and $t_{networks}$ are defined in the worst case. This means that the adversary chooses which nodes he compromises, with the maximum number of tolerable compromised nodes being defined by the thresholds. This is in contrast with the LINCOS protocol in which a threshold t indicates that *any* set of shares of size t is sufficient to obtain the secret.

Finally, we define the failure threshold t_{fail} as the minimum number of failed nodes so that the secret document S cannot be recovered. Again, t_{fail} is defined in the worst case, that is to say that an adversary who would like to prevent the document owner from recovering the secret document chooses the failed nodes, within the limit of t_{fail}.

Standard-mode MULTISS This mode uses two-levels of secret sharing. The owner starts by generating a polynomial P of degree $d(P)$ such that $P(0) = S$. Then the owner generates ℓ polynomials Q_i of degree $d(Q_i)$ such that:

- $Q_0(0) = P(1)$,
- $Q_i(0) = P'(i)$ for $i = 1, \ldots, \ell - 1$, where P' is the derivative of polynomial P.

Finally, each network N_i stores the values $Q_i(1), \ldots, Q_i(n_i)$ using a local Shamir secret sharing. In more details, the mother network applies a Shamir secret sharing to store the value $P(1)$, while the $\ell - 1$ daughter networks use a local Shamir secret sharing to store $P'(i)$ for $i = 1, \ldots \ell - 1$.

To reconstruct the original secret, the document owner needs to recover the shares from $T(P)$ networks, where the share from N_0 is mandatory. In the subnet N_i, the number of shares required to learn $P(i)$ is $T(Q_i)$.

We can fully analyse the threshold of the protocol. The number $t_{networks}$ of subnetworks to compromise is $T(P)$, so it is completely determined by the degree of P. The number t_{nodes} of nodes to compromise is

$$T(Q_0) \min\{\sum_{i \in I} (T(Q_i)) \,|T \subset \{1, \ldots, \ell - 1\}, \, card(I) = T(P) - 1\}.$$

The corresponding strategy is to recover the minimum values of P, that is, $T(P)$, by breaking the security of the ITS subnets N_i where the degrees of polynomials Q_i are the smallest.

In standard mode, the failure threshold is achieved when one of the following conditions is satisfied:

- $P(1)$ cannot be reconstructed, that is, $t_{f_0} = n_0 - T(Q_0) + 1$ shares are unavailable in the mother network N_0,
- less than $T(P) - 1$ shares of P are available, that is, $n_i - T(Q_i) + 1$ shares are lost in the networks N_i for $i \in I$ such that $card(I) = \ell - 1 - (T(P) - 1)$.

Denoting $t_{f_1} = \min\{\sum_{i\in I} n_i - T(Q_i) + 1 | T \subset \{1, \ldots, \ell - 1\}, card(I) = \ell - T(P)\}$, we get

$$t_{fail} = \min\{t_{f_0}, t_{f_1}\}.$$

Local-mode MUTLISS This mode uses three levels of secret sharing. The document owner first generates a random polynomial P whose degree $d(P)$ depending on the targeted values of t_{nodes} and $t_{networks}$, such that $P(0) = S$.

Then, for each node i in N_0, the document owner computes a polynomial Q_i of degree 1 such that $Q_i(0) = P(i)$, and distributes $Q_i(1)$ to each storage node in N_0.

The document owner then computes Q'_i, the derivative of each polynomial Q_i, and stores the value $Q'_i(1)$ in subnet N_i using a local Shamir scheme. For this, the document owner generates, for each polynomial Q'_i, a new polynomial R_i, such that $R_i(0) = Q'_i(2)$. The values $R_i(j)$ for $j = 1, \ldots, n_i$ are then distributed to the nodes of network N_i.

We can again analyse the threshold of the protocol. The number of subnetworks to compromise $t_{networks}$ is $T(P) + 1$, which corresponds to values stored in $T(P)$ subnets and the mother network. The smallest number of nodes to compromise is

$$t_{nodes} = \min\{T(P) + \sum_{i\in I}(T(R_i)) \,|\, card(I) = T(P)\}.$$

The corresponding strategy is to recover the minimum number of different values of P, that is, $T(P)$, by breaking the security of the ITS subnets N_i where the degrees of polynomials R_i are the smallest and the same number of nodes within the mother network.

In local mode, the failure threshold is achieved when $n_0 - T(P) + 1$ shares of P are unavailable. A share of P is unavailable in the following cases:

- a share is unavailable in the mother network N_0, which makes $Q_i(1)$ unavailable,
- $n_i - T(R_i) + 1$ nodes are unavailable in the network N_i, which makes $Q'_i(2)$ is unavailable.

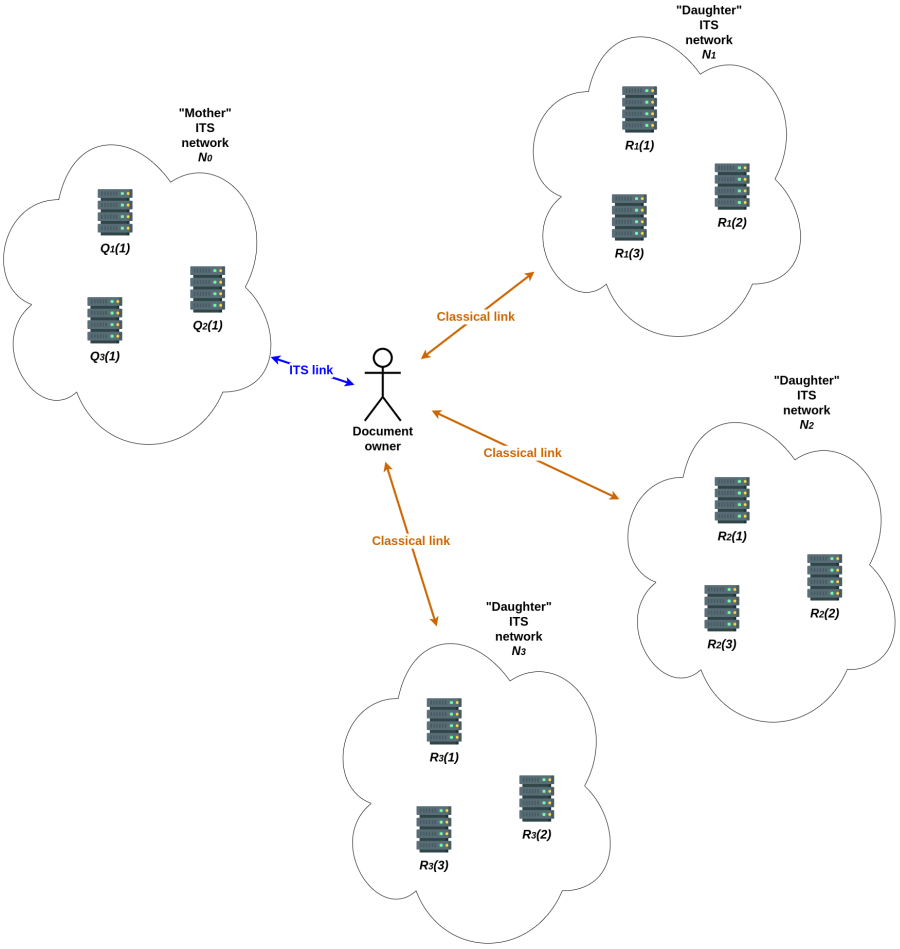

Fig. 3. Local-mode MULTISS: the number of daughter subnetworks is equal to the number of nodes in the mother subnet. We have here $t_{nodes} = 6$, $t_{networks} = 3$ and $t_{fail} = 2$.

Shares update The document owner can regenerate the shares of the secret document, in case one of the shareholder nodes is compromised and the share it is holding is leaked. The share update procedure aims at reseting the shares so that any share previously leaked cannot be combined with new ones to recover the secret. This operation can be done when there is a notice of leakage, but also periodically to pro-actively prevent the leakage of the original secret S.

We distinguish two cases: the share renewal of the daughter subnets, and of the mother subnet.

The share update within a daughter networks is straightforward both in standard and local mode. In both cases, the daughter subnets hold the evaluations of a polynomial using a local Shamir secret sharing as in the LINCOS protocol. The reshare procedure all is therefore similar to the LINCOS protocol. Moreover, each daughter network can be updated independently.

We proceed as described in Section 2. Assume that the subnet N_j stores some value using a local Shamir secret sharing scheme with polynomial R_j of degree $d(R_j)$. We generate a new polynomial δ_j of degree $d(R_j)$, such that $\delta_j(0) = 0$. Then, for each node i that stores the value $R_j(i)$, we add the value $\delta_j(i)$:

$$new\_R_j(k) = R_j(k) + \delta_j(k) \tag{4}$$

This satisfied $new\_R_j(0) = R_j(0)$.

The share update of the mother subnetwork is different in standard and local mode. In standard mode, the mother subnetwork stores the value $P(1)$ using a local Shamir scheme. It can be updated independently of the daughter networks. In local mode, on the other hand, we must restart the distribution process, since it is impossible to renew the shares of the mother subnetwork independently of the daughter subnetworks. The document owner must therefore retrieve the original document before regenerating the shares.

Subnet value update In the case one of the daughter subnet is fully compromised, and the attacker has learned the stored value, it is possible to initiate a *Subnet update* procedure. After this value has been updated, the information obtained by the attacker becomes obsolete.

The procedure differs significantly in standard and local mode. In local mode, the network N_i stores the value $Q'_i(2)$ using a local Shamir secret sharing scheme. This value is used with $Q_i(1)$ to reconstruct $Q_i(0) = P(i)$. If the attacker compromises the whole network N_i, he obtains the value $Q'_i(2)$. It is then possible to update the polynomial Q_i without changing the value $Q_i(0)$ as in the reshare procedure described above. This will change the value $Q_i(1)$ stored in the mother network, and all the shares stored in the daughter subnet N_i.

In global mode, the procedure is more involved. Assume that the daughter network N_i stores the value $Q_i(0) = P'(i)$ using a local Shamir secret sharing. If this value has leaked, it is required to change all the shares in all subnetworks. The document owner must therefore retrieve the original document before regenerating the shares and redistributing them to all the subnets

3.3 Security analysis

Proving the security of the MULTISS protocol is equivalent to proving the security of hierarchical secret sharing by Birkhoff interpolation. In section 3.1, we defined two types of adversaries: the adversary capable of compromising an entire ITS subnet, and the "Harvest now, decrypt later" adversary. We analyze the security of MUTLISS against these two adversaries independently. We also assume that classical cryptography is secure during the execution of the protocol, which means that the adversary cannot break the authentication to perform a Man-in-the-middle attack.

Subnet compromission We first consider an attacker who can take control of an entire ITS subnetwork, for example a hacker who has penetrated all the nodes of a subnetwork, or a state that has performed a legal interception in the computer equipment located under its jurisdiction.

We consider the case in which the adversary was able to compromise all the shareholder nodes of $t_{networks} - 1$ arbitrary subnetworks. Since the secret S cannot be recovered without the value stored in the mother network, we furthermore assume that it is fully compromised as well as all the nodes of $t_{networks} - 2$ arbitrary daughter subnetworks.

In standard mode, the adversary learns $P(1)$ and $t_{networks} - 2$ values $Q_i(0) = P'(i)$. Since $t_{networks} - 1 \leq d(P)$, this does not allow to reconstruct the polynomial P and $P(0) = S$ remains hidden to the adversary.

In local mode, the adversary can interpolate some polynomials R_j from the compromised daughter subnetworks while getting no information on the other polynomials R_j of the healthy daughter subnetworks. Even knowing all the evaluations of the polynomials $Q_i(1)$ from the mother subnet, the adversary lacks, for some polynomials Q_i, the evaluations of the polynomial Q'_i encrypted in shares R_j in the healthy daughter subnet. As in Shamir's secret sharing, the adversary lacks some evaluations of Q'_i, and it is impossible to interpolate $Q_i(0)$.

Since the adversary lacks the evaluation of some polynomials $Q_i(0)$, then the polynomial P remains undetermined, and the initial secret document S remains ITS, provided that the adversary does not compromise more than $t_{networks} - 2$ subnetworks in addition to the mother subnet.

"Harvest now, decrypt later" attacker An HNDL adversary listens to the encrypted communications, hoping to decrypt them when computational power and cryptanalysis are sufficient. In the worst possible scenario, this adversary is able to decrypt all communications transmitted by non-ITS channels. In other words, this amounts to an adversary that has all the information contained in all the daughter ITS subnets $N_1, N_2, \ldots$, whose values where transmitted using channels secured with classical cryptography. However, this adversary has no information on the mother subnet N_0, because the communication between the document owner and the mother subnet is ITS.

In standard mode, the adversary can use those values to interpolate P' but misses the value $P(1)$ stored in the mother network. The confidentiality of the value $P(0) = S$ thus remains ITS.

In local mode, the adversary has all the evaluations of the polynomials R_j of the daughter subnets, and will be able to interpolate those to learn the evaluations of the derived polynomials Q'_i. The adversary however has no information on the evaluations of the polynomials Q_i. Since the adversary does not know the evaluations of the polynomials $Q_i(0)$, the polynomial P remain undetermined, and therefore the confidentiality of the secret S remains ITS.

4 Discussion

As cryptanalysis and computing power are improving, the problem of long-term secure storage is becoming of prime importance. It is indeed likely that public key encryption algorithms will be broken in the long term. Symmetric encryption algorithms such as AES will also suffer from the evolution of cryptanalysis and computing power.

Quantum Key Distribution, preventing the adversary from eavesdropping key establishment, combined with One-time pad encryption, for which decrypting is as likely as guessing the secret, provides some answers. They can be used to guarantee the confidentiality of the secret even in the long term.

As we have seen, QKD is difficult to implement at a large scale, due to the distance limitations. An ITS protocol will therefore be more limited than a classical cryptography protocol, and will have to make stronger assumptions about the trust in the participants of the protocol. This directly impacts our protocol, which assumes that an adversary cannot take control of sufficiently many storage nodes to find the secret document.

The protocol we have introduced is very flexible and can be adapted to real-world constraints and opportunities that stem from QKD networks. We provide two examples here. Firstly, the whole ITS argumentation can easily be transposed to the case where QKD is combined with AES encryption. While this combination is not ITS, it still provides better security than classical public-key cryptography, while significantly increasing the throughput of the communication. AES is standardized and considered post-quantum, and when combined with QKD, provides *post-compromised security*, which means that compromising a session does not reveal the keys used in subsequent session. An adversary that wants to learn the keys used in each AES session needs to break QKD during each session.

All the arguments we made, showing that our construction is ITS remain mutatis mutandis when replacing one-time pad encryption with AES. Of course, the resulting construction is not ITS, not even everlasting, but "as secure as AES", which is in practice considered stronger than public key cryptography.

Another interesting extension is to consider subnets connected to each other using satellite QKD. While in theory satellite QKD provides the same security as fiber-based, the key rates are much lower. In other words, one may consider that

establishing keys within the same subnet is *cheap* while establishing keys between subnets is *expensive*. In this case, the local mode is particularly interesting, since the subnet value protocol does not require secure communication with all the subnets, but only between the mother network and the compromised subnet.

Conclusion

In this paper, we have introduced a new protocol, MULTISS, to extend the secure storage of the LINCOS protocol over several ITS networks. The LINCOS protocol allows to distribute a secret document between several nodes of an ITS network. In practice, these ITS networks are networks of nodes connected by a quantum communication links, and the ITS communication is implemented with Quantum Key Distribution, One-Time pad encryption and ITS authentication. The maximum distance that technically allows the transmission of qubits is in practice limited, so these ITS networks are often deployed at the metropolitan area scale. It is therefore possible for an attacker that can take control of the entire ITS network to find the secret document. This attacker, for example, can be a state that orders the seizure of computer equipment located in its jurisdiction, or a hacker who has infiltrated the entire infrastructure of the ITS network.

The MULTISS protocol distributes the shares of the secret document among the nodes of several ITS subnets, potentially very distant from each other. It forces an adversary to either break into several independent ITS networks, or to break the mother network and perform "Harvest Now, Decrypt Later" attacks on the classical links between the subnets. The ITS subnets can also be managed by different organizations, which makes hacker penetration more difficult. A state will have much more difficulty to obtain the data from computer equipment outside its jurisdiction. Although some shares transit through links protected by classic cryptography only, our secret document remains ITS in both cases.

Our protocol is intended to be particularly flexible, adapting to all network topologies. The document owner chooses how to distribute the shares, according to his need for security and availability of the secret document. To do this, he can adjust the distribution of shares according to the chosen t_{nodes}, $t_{networks}$ and t_{fail} thresholds.

The two versions of the protocol we have introduced satisfy different constraints. In standard mode, it is possible to update the shares in any subnetwork, but updating values requires communication between the mother network and all subnets. In local mode, its is not possible to update the shares of the mother network without updating all subnets, but updating a value in a subnet only requires one round of communication with the mother network. In further work, we will investigate if different secret sharing schemes [29] achieve better tradeoffs for communication in the subnet value update procedure.

Acknowledgements T. Prevost acknowledges PhD funding from UCA. The authors are grateful to Bruno Martin (`https://webusers.i3s.unice.fr/~bmartin/`) for theoretical support.

References

1. Medical data successfully protected by quantum cryptography in Graz (December 2020), `https://www.idquantique.com/openqkd-id-quantique-and-fragmentix-successfully-apply-qkd-for-a-medical-use-case/`
2. Ashkenazy, A., Idan, Y., Korn, D., Fixler, D., Dayan, B., Cohen, E.: Photon number splitting attack–proposal and analysis of an experimental scheme. Advanced Quantum Technologies (2024)
3. Bennett, C., Brassard, G.: Quantum cryptography: Public-key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing. pp. 175–179. IEEE Computer Society Press, Los Alamitos (1984)
4. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. Theoretical computer science (2014)
5. Bettelli, S., Lorünser, T., Peev, M., Querasser, E., Dušek, M., Bartůšková, L., Blauensteiner, B., Huebel, H., Poppe, A., Zeilinger, A.: Effect of double pair emission to entanglement based qkd. In: The European Conference on Lasers and Electro-Optics. Optica Publishing Group (2007)
6. Bhatia, V., Ramkumar, K.: An efficient quantum computing technique for cracking RSA using Shor's algorithm. In: 2020 IEEE 5th international conference on computing communication and automation (ICCCA). pp. 89–94. IEEE (2020). https://doi.org/10.1109/ICCCA49541.2020.9250806
7. Biham, E., Shamir, A.: Differential cryptanalysis of the full 16-round des. In: Advances in Cryptology—CRYPTO'92: 12th Annual International Cryptology Conference Santa Barbara, California. Springer (1993)
8. Birkhoff, G.D.: General mean value and remainder theorems with applications to mechanical differentiation and quadrature. Transactions of the American Mathematical Society (1906)
9. Blakley, G.R.: Safeguarding cryptographic keys. In: Managing requirements knowledge, international workshop on. IEEE Computer Society (1979)
10. Boaron, A., Boso, G., Rusca, D., Vulliez, C., Autebert, C., Caloz, M., Perrenoud, M., Gras, G., Bussières, F., Li, M.J., Nolan, D., Martin, A., Zbinden, H.: Secure quantum key distribution over 421 km of optical fiber. Phys. Rev. Lett. **121**, 190502 (Nov 2018). https://doi.org/10.1103/PhysRevLett.121.190502, `https://link.aps.org/doi/10.1103/PhysRevLett.121.190502`
11. Braun, J., Buchmann, J., Demirel, D., Geihs, M., Fujiwara, M., Moriai, S., Sasaki, M., Waseda, A.: Lincos: A storage system providing long-term integrity, authenticity, and confidentiality. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (2017)
12. Choc, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: Annual Symposium on Foundations of Computer Science (Proceedings) (1985)
13. Corniaux, C.L., Ghodosi, H.: An entropy-based demonstration of the security of Shamir's secret sharing scheme. In: 2014 International Conference on Information Science, Electronics and Electrical Engineering. IEEE (2014)

14. Ekert, A.K.: Quantum cryptography based on bell's theorem. Physical review letters (1991)
15. Fujiwara, M., Hashimoto, H., Doi, K., Kujiraoka, M., Tanizawa, Y., Ishida, Y., Sasaki, M., Nagasaki, M.: Secure secondary utilization system of genomic data using quantum secure cloud. Sci Rep **15**(18530) (2022)
16. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: Advances in Cryptology—CRYPT0'95: 15th Annual International Cryptology Conference Santa Barbara, California, USA, August 27–31, 1995 Proceedings 15. Springer (1995)
17. Honjo, T., Nam, S.W., Takesue, H., Zhang, Q., Kamada, H., Nishida, Y., Tadanaga, O., Asobe, M., Baek, B., Hadfield, R., et al.: Long-distance entanglement-based quantum key distribution over optical fiber. Optics Express (2008)
18. Kaluderovic, N.: Attacks on some post-quantum cryptographic protocols: The case of the Legendre PRF and SIKE. Tech. rep., EPFL (2022). https://doi.org/10.5075/epfl-thesis-8974
19. Li, B.H., Xie, Y.M., Li, Z., Weng, C.X., Li, C.L., Yin, H.L., Chen, Z.B.: Long-distance twin-field quantum key distribution with entangled sources. Optics Letters (2021)
20. Lydersen, L., Wiechers, C., Wittmann, C., Elser, D., Skaar, J., Makarov, V.: Thermal blinding of gated detectors in quantum cryptography. Optics express (2010)
21. Matsui, M.: Linear cryptanalysis method for des cipher. In: Workshop on the Theory and Application of of Cryptographic Techniques. Springer (1993)
22. Paul, S.: On the transition to post-quantum cryptography in the industrial Internet of things (2022)
23. Pelet, Y., Sauder, G., Cohen, M., Labonté, L., Alibart, O., Martin, A., Tanzilli, S.: Operational entanglement-based quantum key distribution over 50 km of field-deployed optical fibers. Phys. Rev. Appl. **20**, 044006 (Oct 2023). https://doi.org/10.1103/PhysRevApplied.20.044006, `https://link.aps.org/doi/10.1103/PhysRevApplied.20.044006`
24. Ribezzo, D., Zahidy, M., Lemmi, G., Petitjean, A., De Lazzari, C., Vagniluca, I., Conca, E., Tosi, A., Occhipinti, T., Oxenløwe, L.K., Xuereb, A., Bacco, D., Zavatta, A.: Quantum key distribution over 100 km of underwater optical fiber assisted by a fast-gated single-photon detector. Phys. Rev. Appl. **20**, 044052 (Oct 2023). https://doi.org/10.1103/PhysRevApplied.20.044052, `https://link.aps.org/doi/10.1103/PhysRevApplied.20.044052`
25. Ribezzo, D., Zahidy, M., Vagniluca, I., Biagi, N., Francesconi, S., Occhipinti, T., Oxenløwe, L.K., Lončarić, M., Cvitić, I., Stipčević, M., Pušavec, Ž., Kaltenbaek, R., Ramšak, A., Cesa, F., Giorgetti, G., Scazza, F., Bassi, A., De Natale, P., Cataliotti, F.S., Inguscio, M., Bacco, D., Zavatta, A.: Deploying an inter-european quantum network. Advanced Quantum Technologies **6**(2), 2200061 (2023). https://doi.org/https://doi.org/10.1002/qute.202200061, `https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.202200061`
26. Shamir, A.: How to share a secret. Communications of the ACM (1979)
27. Shannon, C.E.: Communication theory of secrecy systems. The Bell system technical journal (1949)
28. Tassa, T.: Hierarchical threshold secret sharing. In: Theory of Cryptography Conference. Springer (2004)
29. Traverso, G., Demirel, D., Buchmann, J.: Dynamic and verifiable hierarchical secret sharing. In: Information Theoretic Security: 9th International Conference, ICITS 2016, Tacoma, WA, USA, August 9-12, 2016, Revised Selected Papers 9. Springer (2016)

30. Unruh, D.: Everlasting multi-party computation. Journal of Cryptology **31**, 965 – 1011 (2018)
31. Waring, E.: Vii. problems concerning interpolations. Philosophical transactions of the royal society of London (1779)
32. Wohlwend, J.: Elliptic curve cryptography: Pre and post quantum. `http://math.mit.edu/\~apost/courses/18.204-2016/18.204\_Jeremy\_Wohlwend\_final\_paper.pdf` (2016)
33. Zygelman, B., Zygelman, B.: No-cloning theorem, quantum teleportation and spooky correlations. A First Introduction to Quantum Computing and Information (2018)