MULTISS: Long-Term Secure Distributed Storage over Remote QKD Networks

Olivier Alibart[4][0000−0003−4404−4067], Marc Kaplan[5][0009−0002−5550−8859], Pascal Lafourcade[1,6][0000−0002−4459−511X], Anne Marin[5][0009−0007−4135−1172], Charles Olivier-Anclin[2][0000−0002−9365−3259], and Thomas Prévost[1,3][0009−0000−2224−8574]⋆

[1] LIMOS, Université Clermont-Auvergne, France
firstname.lastname@uca.fr
[2] LIMOS, Université Clermont-Auvergne, Clermont-Auvergne INP, France
firstname.lastname@uca.fr
[3] I3S, Université Côte d'Azur, France
firstname.lastname@uca.fr
[4] INPHYNI, Université Côte d'Azur, France
firstname.lastname@univ-cotedazur.fr
[5] Veriqloud, France
firstname.lastname@veriqloud.fr
[6] ASTEROIDE, Trust4Sign, La Monnerie-le-Montel, France

Abstract. We introduce MULTISS, a distributed storage protocol over multiple remote Quantum Key Distribution (QKD) networks that ensures long-term data confidentiality. MULTISS is designed to combine several QKD networks, as QKD links are inherently limited in range, and to connect these networks through classical cryptographic channels. It relies on a hierarchical secret sharing scheme that makes certain shares mandatory for the reconstruction of the original secret. Hence, MULTISS offers a secure distributed storage solution in a scenario that is compatible with the current deployment of quantum networks. We prove that MULTISS preserves the confidentiality of stored data even if an adversary (i) gains full access to all nodes of a subset of the QKD networks, or (ii) records all classical communications between distant QKD networks and compromises them at a later stage. We show that it provides strictly stronger security guarantees than existing QKD-storage protocols.
Our protocol also includes a procedure to update the shares without reconstructing the original data. In addition, we provide a recovery mechanism that tolerates the complete compromise of some QKD networks. We further introduce a variant of the protocol, called the local-mode, that minimizes the communication overhead of the recovery procedure when deployed over specific network topologies.

1 Introduction

The long-term protection of sensitive information remains a critical challenge in modern cryptography. Recent NIST guidance (NIST IR 8547) [31] highlights

⋆ Corresponding author, contact@thomasarmel.fr.

the need to account for the threat model commonly referred to as *Harvest Now, Decrypt Later* [35]. Indeed, it is reasonable to fear that adversaries may already be intercepting and storing encrypted communications with the intention of decrypting them once more powerful computational resources or improved cryptanalytic techniques become available. *Information-Theoretic Security* (ITS) [42] provides a notion of *perfect security*: confidentiality does not depend on any assumption regarding the adversary's computational power. To ensure long-term protection, protocols must therefore provide ITS guarantees throughout the entire lifetime of the protected data. However, ITS encryption comes with stringent requirements: the encryption key must be perfectly random, non-reused, and at least as long as the message. Consequently, key distribution becomes difficult to scale in practice, and protocols have to make stronger assumptions about the trust in the participants. In conventional communication systems, key establishment is typically achieved via public-key cryptography, relying on computational assumptions that may be broken and thus not ITS. Post-quantum primitives security assumptions may also be proved false at some point [44, 50, 40, 5].

Quantum Key Distribution (QKD) [3] offers a fundamentally different approach to key establishment. Rather than relying on the computational difficulty of solving mathematical problems, QKD derives its security from the laws of quantum mechanics, which let parties detect eavesdropping and thus establish shared keys with information-theoretic security. When combined with OTP encryption, the resulting communication channel provides end-to-end information-theoretic confidentiality, thereby eliminating the threat posed by HNDL attacks.

Several protocols [16, 9, 17] provide secure secret storage over QKD networks. The data are distributed using the ITS secure Shamir's secret sharing scheme [41, 10] which splits them into shares that individually contain no information about the original data. Recovery requires collecting a sufficient number of shares, specified by the scheme's *threshold*. The confidentiality is achieved by distributing the shares via QKD-secured communication, hence relying on trust assumptions *i.e.,* as long as the adversary cannot compromise enough nodes.

Despite successful implementations on several QKD testbeds in Europe [23] and Asia [18], these protocols remain limited by the transmission range of QKD. Due to attenuation in optical fibers, a QKD link is limited to a few hundred kilometers. As a result, existing deployments have been restricted to metropolitan-scale networks, reflecting the practical limitations of most current QKD infrastructures [36, 38]. We consider this geographical concentration to be a limitation of the protocol. At such a local scale, the participating nodes are likely to be operated by a single organization or by closely related entities, increasing the risk that the compromise of one server could facilitate the compromise of the others. Moreover, confining all nodes to a single jurisdiction may expose the protected secret to legal seizure or coercive measures by local authorities.

Our motivation is to strengthen the security of distributed storage systems against such an adversary by leveraging multiple remote QKD networks. This increases both the geographical coverage and the number of participating nodes. It also accounts for the practical limitations of QKD and the possible unavailability

Table 1: Security properties comparison between MULTISS and previous QKD network storage protocols.

Security property	[16]	[17]	[9]	**MULTISS**
ITS confidentiality	✓	✓	✓	✓
t_{fail}-availability	✓	✓	✓	✓
Proxyless	✓	✗	✓	✓
Authenticity	✗	✓	✓	✓
Integrity	✗	✓	✓	✓
Node post-compromise secrecy	✗	✓	✓	✓
Keyless	✗	✗	✓	✓
Subnet compromise secrecy	✗	✗	✗	✓
Subnet post-compromise secrecy	✗	✗	✗	✓

of QKD links over certain network segments. Some sections between those remote QKD networks are secured with classical cryptography only, making those links the weakest point of the overall security architecture. To our knowledge, no ITS-secure storage protocol on QKD network guarantees confidentiality against an adversary capable of compromising all nodes of a network.

Contributions. We overcome those limitations with MULTISS, a hybrid post-quantum secret storage protocol between QKD and classical transmission. Our protocol mixes classical and hierarchical [45] secret sharing schemes. The main advantage is to preserve the ITS security over 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 HNDL attacks. We propose a mechanism to renew the shares of each node periodically without reconstructing the initial secret, allowing to recover from the leakage of some of the nodes in the QKD network. Furthermore, we introduce a procedure to recover from a full leakage of some of the QKD networks. This recovery procedure still needs some classical communication between QKD networks. To minimize this overhead, we introduce a *local mode* for MULTISS: it simplifies subnet value updates by limiting classical coordination to just two networks, whereas the *standard mode* requires secret reconstruction and thus communication across all subnets and document owner. We provide formal security proofs showing that our protocol remains secure against a computationally unbounded adversary. Table 1 summarizes and compares the security properties of MULTISS and previous protocols for secret storage over QKD network.

Outline. The paper is structured as follows. Section 2 reviews related work on classical multi-cloud storage protocols and confidential storage over QKD networks. Section 3 defines the threat model and adversarial assumptions. Section 4 introduces a generic model for MULTISS schemes. Section 5 presents the technical background used in the paper. Section 6 details our protocol instantiations

in standard and local modes. Section 7 discusses security, and Section 8 presents the implementation and performance evaluation.

2 Related Work

Prior research [1, 34, 8, 32, 51, 27] proposed protocols for secure distributed storage over classical networks. These solutions target multi-cloud storage, assuming a threshold of uncompromised Cloud service providers (CSPs). Several secure storage protocols [16, 9, 17] utilize Quantum Key Distribution (QKD) networks to secure owner-server communications, achieving ITS security as long as sufficient nodes remain uncompromised.

Classical Multi-Cloud Storage Protocols. Horkos et al. [21] classified multi-cloud storage protocols by key usage (*key-leveraging* vs. *keyless*), by reliance on a trusted processor and distinguishing *proxyless* schemes from those using a *trusted proxy*. The Secure CloUD storage (SCUD) keyless protocol [32] leverages recursive Shamir Secret Sharing to ensure ITS confidentiality. It is proxyless and guarantees integrity from ECDSA signatures. The key-leveraging multi-cloud storage protocol CloudSeal [51] employs a Content Delivery Service as a semi-trusted proxy ("honest-but-curious"). It achieves computational security using secret sharing and proxy re-encryption for dynamic access policies with forward and backward secrecy. Lafourcade et al. [27] introduced KAPRE, a post-quantum keyless multi-cloud secret storage. The protocol relies on homomorphic proxy-re-encryption to transfer data shares to the CSPs via an untrusted intermediate proxy. The authors also introduced KAME (Keyless Archivage with Multikey Encryption), relying on homomorphic multikey encryption to achieve proxy-CSP collusion resistance, requiring an additional communication round. All the above achieve computational security only, meaning they are vulnerable against HNDL attacks.

Secret Storage on QKD Network. The first ITS secret storage protocol over a QKD network was proposed by [16]. They used four nodes of the Tokyo QKD network, with Wegman-Carter [49] authenticated BB84 key exchange [3]. The data owner sends password-derived Shamir shares to the nodes, receiving response shares that reconstruct the document only if the password was valid. The protocol does not allow shares renewal. In case data owner and end user that retrieves the document are different, Fujiwara et al. [17] propose an ITS storage protocol, implying the participation of a verifier assumed to be "honest but curious". Verifier receives timestamp commitments of the document, to attest that it was committed at a certain time. Authors employ ITS-preserving universal hash functions for commitments, but also propose to use computationally collision-resistant hash functions like SHA-512. This protocol needs a trusted proxy, the *Share Calculator*, that computes and distributes shares and commitments. Authors suggest relying on a Physical Unclonable Function [26] or secure hardware, like [11]. Their protocol ensures post-compromise security,

but the signatures of their decentralized share renewal rely on computational assumptions. They also deployed their protocol on the Tokyo QKD network. LINCOS ITS secret storage protocol [9] uses timestamp commitments as well to guarantee the long-term integrity of the stored document, and does not require a trusted proxy. The protocol was also implemented in Tokyo. We detail more the LINCOS protocol in section 5.4. Contrary to MULTISS, all of these protocols limit the sharing of secrets between shareholders of the same ITS network and thus have in common the vulnerability against an adversary powerful enough to compromise all the nodes of a QKD network.

3 Assumptions and Threat Model

Protocol Assumptions. In MULTISS, the document owner is assumed to be the final user who retrieves the document, so assumed to be honest. MULTISS is a proxyless protocol, the document owner is responsible for all the operations, including share generation and distribution, as well as share renewal. 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 classical IP link, allowing at best classical cryptography. This situation is compatible with most of current and short-term deployment of QKD networks, which are planned at the metropolitan-area scale. 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 is connected via ITS link to only one of the subnetworks*, as shown in Fig. 1. Specifically, the document owner establishes an ITS channel to each node in that subnetwork. The *qline* QKD protocol [12] enables the establishment of ITS channels between the document owner and each node along a single optical fiber, without requiring trust in intermediate nodes. The communications between document owner and other QKD subnets are secured using classical non-ITS cryptography. Nevertheless, we consider all the authentication cryptography *secure at the time of the exchange*, because it would evolve along with cryptographic standards, and thus assume that the attacker cannot perform a MiTM attack during the protocol execution.

Adversary Models. Distributing secret shares within nodes in the same area may introduce vulnerabilities against a sufficiently powerful adversary. If the secret is stored only within one jurisdiction, a legal seizure can recover it. If one entity runs all storage nodes, they are not independent, so compromising one node makes the others easier to breach, and the QKD network operator itself could turn malicious. We protect the secret against an adversary that can take control of all nodes of a QKD network, or of a small subset of the QKD networks. On the other hand, we want to guarantee the long-term confidentiality of the secret document. However, the transmission is ITS between the document owner and a single ITS network only. The secret must stay safe even if all recorded classical traffic is later decrypted, so non-ITS channel data must reveal nothing about it.

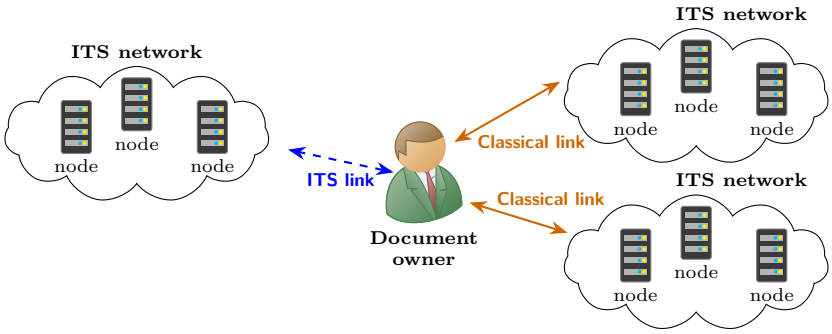


Fig. 1: In the MULTISS protocol, it is considered that the document owner can communicate in an ITS manner to the nodes of only one of the ITS subnets. The link with the other subnets can only be secured with classical cryptography.

Our protocol can be broken by an adversary able to perform both attacks within the same time frame: learn the value stored in one chosen QKD network, and break classical cryptography. We believe that our security model remains relevant in practice. First, because MULTISS proposes a share update procedure, as well as a procedure to renew the shares of a compromised subnet. In *standard mode*, compromised subnet renewal requires to restart the protocol and redistribute all shares over all subnets, while *local mode* minimizes the communication and does not require secret reconstruction. These measures force the adversary to perform the HNDL and subnet compromise attacks during the same renewal round, which can be difficult to achieve in practice. Second, because even if the detection time for illegitimate network penetration is usually high, it remains much shorter than the expected time to break classical cryptography used at the time of the attack. Similarly, if some shares are obtained through legal seizure, it should be noticed by the document owner. Thus, in the scenario where the adversary performs both attacks at the same share renewal round, there is a gap of several years, even decades, between the detection of the adversary and its decryption of the secret document, giving the document owner time to invalidate the secret document so it can no longer be used.

Thresholds. In the MULTISS protocol, the secret document is distributed among several ITS networks using multiple polynomials, which induces a different construction of the thresholds. Shares of a given daughter network are only useful when combined with mother shares, so the minimal shares set is not defined by a single threshold. Some sets of a given size allow to reconstruct the secret, whereas others do not. We define two thresholds for the adversary to recover the secret document:

- t_{nodes}: minimum number of nodes to compromise in the whole network.
- t_{nets}: minimum number of subnets to compromise. We consider an ITS subnet "compromised" when the adversary controls *at least* one of its nodes.

t_{nodes} and t_{nets} are defined in the worst case: the adversary chooses which nodes he compromises, within the limits of the thresholds. Finally, we define the failure threshold t_{fail} as the minimum number of failed nodes that prevents the secret document S from being recovered. t_{fail} is also defined in the worst case.

4 Protocol Model

We consider a secret S encoded as a digit. The set of possible outputs for any given input of algorithm `Alg` is denoted $[\texttt{Alg}(\cdot)]$. We denote the cardinality of a set $\mathcal{I}$ by $|\mathcal{I}|$.

Network Architecture. The network architecture consists in $\ell \geq 2$ QKD subnetworks $\mathcal{N}_i$. Each subnet contains n_i storage nodes that are connected one to each other with QKD ITS links. The document owner also shares an ITS link with one specific network called the "*mother*" subnetwork $\mathcal{N}_0$. The other subnets $\{\mathcal{N}_i\}_{i\geq 1}$ are called "*daughter*". Communications with the daughter subnets are protected with classical cryptography. $N = \sum_{i=0}^{\ell-1} n_i$ denotes the total nodes count in the whole MULTISS network. The network architecture is symmetric, so the mother subnet may vary across document owners: a French user might use the Paris subnet as mother, and Tokyo and Montreal as the daughter subnets, while a Japanese user chooses the Tokyo QKD network as mother.

Definition 1 (MULTISS). *A MULTISS scheme is a tuple of six probabilistic polynomial time algorithms* `MULTISS` *= (*`Setup`*,* `Distribute`*,* `Reconstruct`*,* `RenewGen`*,* `RenewGenNetwork`*,* `RenewUpdate`*):*

- `Setup`$(\{|\mathcal{N}_i|\}_{0\leq i<\ell}, t_{nodes}, t_{nets}, t_{fail}) \rightarrow$ `pp`*: takes as input the thresholds of the protocol and the network topology, and outputs protocol parameters.*
- `Distribute(pp,` S`)` $\rightarrow \{s_{i,j}\}_{0\leq i<\ell, 0\leq j<n_i}$*: takes as input the protocol parameters and the secret document S, and outputs the shares $\{s_{i,j}\}_{0\leq i<\ell,0\leq j<n_i}$ for distribution to the nodes j of subnets i.*
- `Reconstruct(pp,` $\{s_{i,j}\}_{0\leq i<\ell,0\leq j<n_i}$`)` $\rightarrow S$ `or` $\perp$*: takes as input the protocol parameters and a set of shares, and outputs the secret document S if the shares are sufficient to reconstruct it, and an error message otherwise.*
- `RenewGen(pp, i)` $\rightarrow \{\Delta s_{i,j}\}_{0\leq i<\ell,0\leq j<n_i}$ *: executed by the document owner (or a designated coordinator). Takes as input the protocol parameters and optionally a subnet index i, and generates a set of update shares $\{\Delta s_{i,j}\}$ for distribution to the nodes of subnet i or all the nodes of the network.*
- `RenewGenNetwork(pp, i)` $\rightarrow \{\Delta s_{i,j}\}_{0\leq i<\ell,0\leq j<n_i}$ *: procedure executed by the document owner. Takes as input the protocol parameters, optionally a subnet index i, and generates a set of update shares $\{\Delta s_{i,j}\}_{i,j}$ for distribution to the nodes of subnet i or all the nodes of the MULTISS network in case a whole subnet could have been compromised.*
- `RenewUpdate`$(s_{i,j}, \Delta s_{i,j}) \rightarrow s'_{i,j}$ *: executed locally by each storage node. Takes as input the node's current share $s_{i,j}$ and the received update share $\Delta s_{i,j}$, and outputs the renewed share $s'_{i,j}$. The new shares reconstruct the secret S.*

We define below the security properties that we want to guarantee for a MULTISS scheme, and the corresponding security games in Fig. 2.

Correctness. The composition of the algorithms of a MULTISS scheme must allow to recover the original data from the shares, even after shares renewal.

Definition 2 (Correctness). *A MULTISS scheme is* correct *if it verifies the following* $\forall S \in \{0,1\}^*$, $\forall \ell \geq 2$, $\forall \{|\mathcal{N}_i|\}_{0\leq i<\ell}$, $\forall t_{nodes} \leq \sum_{i=0}^{\ell-1} n_i$, $\forall t_{nets} \leq \ell$, $\forall$ $pp \in \mathtt{Setup}(\{|\mathcal{N}_i|\}_{0\leq i<\ell}, t_{nodes}, t_{nets}, t_{fail})$:
$\forall \{s_{i,j}\}_{0\leq i<\ell, 0\leq j<n_i} \in [\mathtt{Distribute}(pp, S)]$,
$\forall t \in \mathbb{N}$ *share renewal rounds count,*
$\forall \left(\{\Delta s_{i,j}^{(k)}\}_{0\leq i<\ell, 0\leq j<n_i} \in [\mathtt{RenewGen}(pp)] \cup [\mathtt{RenewGenNetwork}(pp)]\right)_{0\leq k\leq t}$,
$\forall \left(s_{i,j}^{(k)} \in [\mathtt{RenewUpdate}(s_{i,j}^{(k-1)}, \Delta s_{i,j}^{(k)})]\right)_{\substack{0\leq k\leq t \\ 0\leq i<\ell, 0\leq j<n_i}}$,
$\exists \mathcal{J}$ *set of shares such that* $\mathtt{Reconstruct}(pp, \{s_{i,j}^{(t)}\}_{(i,j)\in\mathcal{J}}) = S$.

Subnet Compromise Secrecy. We model the Subnet Compromise secrecy against an adversary that can take control of all the nodes of some subnets, in the limit of t_{nets}, using the $\mathbf{Exp}_{\mathcal{A}}^{\text{SC}}(\{|\mathcal{N}_i|\}, t_{\text{nodes}}, t_{\text{nets}}, t_{\text{fail}})$ security game defined in Fig. 2. This can be, for example, a state-level adversary that seizes the whole network of a country, or the organization operating the subnetworks that becomes malicious.

Definition 3 (Subnet Compromise Secrecy). *A MULTISS scheme is* subnetwork compromise secret *if for any network topology* $\{|\mathcal{N}_i|\}$*, for any* t_{nodes}*,* t_{nets}*,* t_{fail}*, for any adversary* $\mathcal{A}$*:*
$\boldsymbol{Adv}^{SC}(\mathcal{A}) := \left|Pr\left[\boldsymbol{Exp}_{\mathcal{A}}^{\text{SC}}(\{|\mathcal{N}_i|\}, t_{nodes}, t_{nets}, t_{fail}) = 1\right] - 1/2\right| = 0.$

HNDL Secrecy. We model the HNDL secrecy against an adversary that can eavesdrop and then decrypt later the classical communication between the document owner and the daughter subnets, using the security game defined in Fig. 2 $\mathbf{Exp}_{\mathcal{A}}^{\text{HNDL}}(\{|\mathcal{N}_i|\}, t_{\text{nodes}}, t_{\text{nets}}, t_{\text{fail}})$.

Definition 4 (HNDL Secrecy). *A MULTISS scheme is* HNDL secret *if for any network topology* $\{|\mathcal{N}_i|\}$*, for any* t_{nodes}*,* t_{nets}*,* t_{fail}*, any adversary* $\mathcal{A}$ *we have*
$\boldsymbol{Adv}^{HNDL}(\mathcal{A}) := \left|Pr\left[\boldsymbol{Exp}_{\mathcal{A}}^{\text{HNDL}}(\{|\mathcal{N}_i|\}, t_{nodes}, t_{nets}, t_{fail}) = 1\right] - 1/2\right| = 0.$

Remark. Neither *Subnet Compromise Secrecy* from Definition 3 nor *HNDL Secrecy* from Definition 4 property is contained in the other. HNDL secrecy ensures confidentiality against an adversary that can retrieve all shares except the mother network, while SC secrecy guarantees confidentiality against an adversary that compromised $t_{\text{nets}} - 1$ subnets, mother or daughters.

t_{fail}*-Availability.* This property ensures secret availability against an adversary that can make some shares unavailable.

$\mathbf{Exp}^{\mathrm{SC}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$

$b \xleftarrow{\$} \{0,1\}$
$\mathtt{pp} \leftarrow \mathtt{Setup}(\{|\mathcal{N}_i|\},\ t_{\mathrm{nodes}},\ t_{\mathrm{nets}},\ t_{\mathrm{fail}})$
$S^0, S^1, \mathcal{I} \leftarrow \mathcal{A}(\mathtt{pp})$
$\{s_{i,j}\}_{i,j} \leftarrow \mathtt{Distribute}(\mathtt{pp},\ S^b)$
$b' \leftarrow \mathcal{A}(\{s_{0,j}\}_{j<n_0}, \{s_{i,j}\}_{i\in\mathcal{I}, j<n_i})$
If $|\mathcal{I}| < t_{\mathrm{nets}} - 1$
  return $b = b'$
return b

$\mathbf{Exp}^{\mathrm{HNDL}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$

$b \xleftarrow{\$} \{0,1\}$
$\mathtt{pp} \leftarrow \mathtt{Setup}(\{|\mathcal{N}_i|\},\ t_{\mathrm{nodes}},\ t_{\mathrm{nets}},\ t_{\mathrm{fail}})$
$S^0, S^1 \leftarrow \mathcal{A}(\mathtt{pp})$
$\{s_{i,j}\}_{i,j} \leftarrow \mathtt{Distribute}(\mathtt{pp},\ S^b)$
$b' \leftarrow \mathcal{A}(\{s_{i,j}\}_{1\le i<\ell, j<n_i})$
return $b = b'$

$\mathbf{Exp}^{\mathrm{PCS}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$

$b \xleftarrow{\$} \{0,1\}$
$\mathtt{pp} \leftarrow \mathtt{Setup}(\{|\mathcal{N}_i|\},\ t_{\mathrm{nodes}},\ t_{\mathrm{nets}},\ t_{\mathrm{fail}})$
$S^0, S^1, q, \mathrm{state} \leftarrow \mathcal{A}(\mathtt{pp})$
$\{s^{(0)}_{i,j}\}_{i,j} \leftarrow \mathtt{Distribute}(\mathtt{pp},\ S^b)$
for $k = 1 \dots q$ **do**
  $\mathcal{I}^{(k)} \leftarrow \mathcal{A}(k, \mathrm{state})$
  $\{\Delta s^{(k)}_{i,j}\}_{i,j} \leftarrow \mathtt{RenewGenNetwork}(\mathtt{pp},\ i),\ \forall i \in [\![0, \ell-1]\!]$
  $s^{(k)}_{i,j} \leftarrow \mathtt{RenewUpdate}(s^{(k-1)}_{i,j}, \Delta s^{(k)}_{i,j})\ \forall i \in [\![0, \ell-1]\!], j < n_i$
  $\mathrm{state} \leftarrow \mathcal{A}(\{s^{(k)}_{i,j}\}_{i\in\mathcal{I}^{(k)}, j<n_i})$
$b' \leftarrow \mathcal{A}(\mathrm{state})$
If $\left(\forall k \le q, (|\mathcal{I}^{(k)}| < t_{\mathrm{nets}})\right) \vee \left(\mathcal{N}_0 \notin \mathcal{I}^{(k)}\right)$
  return $b = b'$
return b

Fig. 2: Security games for the three types of adversaries: whole subnets compromise, classical communication eavesdropping and mobile adversary. For $\mathbf{Exp}^{\mathrm{SC}}_{\mathcal{A}}$, we assume that the mother network $\mathcal{N}_0$ is among the compromised subnets. $\mathcal{I}$ denotes a set of compromised subnets chosen by the adversary.

Definition 5 (t_{fail}-Availability). *We say that a MULTISS scheme guarantees* t_{fail}-availability *if for any network topology* $\{|\mathcal{N}_i|\}$*, for any* t_{nodes}*,* t_{nets}*,* t_{fail}*, and any subset of up to* t_{fail} *unavailable nodes, the document owner can reconstruct the document* S *using the shares of the remaining nodes.*

Definition 6 (Mobile Adversary). *A* mobile adversary *is an adversary that can take control of whole subnets at different times, up to* $t_{nets} - 1$ *subnets at a time, and may additionally store all classical communications for later decryption. The adversary is adaptive and may move between subnets over time, while the document owner periodically refreshes the shares across the network.*

Post Compromise Secrecy. We model the post compromise secrecy against a *mobile adversary.* We assume that the document owner performs a network-

wide shares renewal after each compromise, using `RenewGenNetwork` algorithm, for all subnets. $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{PCS}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$ in Fig. 2 defines security game.

Definition 7 (Post Compromise Secrecy). *A MULTISS secure secret storage scheme guarantees* post compromise secrecy *if for any network topology* $\{|\mathcal{N}_i|\}$*, for any* t_{nodes}*,* t_{nets}*,* t_{fail}*, any adversary* $\mathcal{A}$ *we have*

$$\boldsymbol{Adv}^{PCS}(\mathcal{A}) := \left|Pr\left[\boldsymbol{Exp}_{\mathcal{A}}^{\mathrm{PCS}}(\{|\mathcal{N}_i|\}, t_{nodes}, t_{nets}, t_{fail}) = 1\right] - 1/2\right| = 0.$$

Remark. The `RenewGen` procedure aims to guarantee inner subnet post compromise secrecy. Because each subnet is itself ITS, the share renewal process can be done in a fully ITS manner, by delegating the task to a node in each subnet. However, since MULTISS aims to be as flexible as possible and thus allows to use subnetworks with only one node, we cannot guarantee that compromising one node would never be equivalent to compromising the whole subnet. Therefore, we do not provide formal guarantee of *inner subnet post compromise secrecy*.

5 MULTISS's Underlying Components

We present the underlying primitives and secure channels leveraged by MULTISS. We also detail the COPRIS [9] protocol, which is part of LINCOS [9]. COPRIS is the component for authenticity and integrity in MULTISS.

5.1 Quantum Key Distribution

Quantum Key Distribution (QKD) derives its security from the quantum no-cloning Theorem [52], which states that an arbitrary qubit cannot be copied without disturbing its state. QKD protocols transmit qubits between participants either directly [3] or via entanglement [14, 4]. Interception inevitably alters qubit states, enabling honest parties to detect eavesdropping by sampling measurement results, and abort the protocol. Provided an authenticated channel prevents MiTM attacks, QKD thus ensures ITS security. Authentication is typically achieved using public-key cryptography, but if that authentication is compromised during protocol execution, the security of QKD can still be undermined. This provides *everlasting security* [47]: an adversary unable to break the protocol during execution cannot extract any information later. It is possible to construct ITS channels by combining QKD, one-time-pad encryption [42] and authentication based on universal hash functions with pre-shared keys [49, 15]. QKD flying qubits are typically realized via a single photon's quantum state, such as its polarization or time bin (the exact arrival time). Most implementations transmit qubits through "*dark optical fibers*", with a range limited by fiber loss rate, usually a few hundred kilometers [7, 39]. Therefore QKD is currently restricted to metropolitan distances without quantum repeaters, which are notoriously hard to build [20, 29, 13, 2]. This is the case of the Tokyo QKD network [18] on which [16, 9, 17] protocols were first deployed. In Europe, EuroQCI networks are also considering for now deployments at the metropolitan scale [36, 38]. In the rest of the paper, we use the expressions *QKD links* to refer

to the quantum communication channels implementing QKD protocols, 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. QKD networks consist in both quantum and classical communication networks implementing all together QKD, One-time-pad and ITS authentication.

5.2 Shamir's Secret Sharing Scheme

Shamir's Secret Sharing Scheme [41] is a cryptographic primitive that allows a *dealer* to distribute a secret S among n participants. The dealer defines a threshold $k \leq n$ of participants who must pool their shares to retrieve the initial secret. He chooses a prime p such that $p > S$, then generates a random polynomial $P \in \mathbb{F}_p[X]$ of degree $\deg(P) = k - 1$, so that $P(0) = S$. Finally, he distributes to each of the n participants evaluations $P(1), P(2), \ldots, P(n)$ of the polynomial. k participants among n can then pool their shares and retrieve the initial secret $S = P(0)$ using Lagrange interpolation [48]. To simplify the notations, we denote the decryption threshold $T(P) = k = \deg(P) + 1$. Shamir Secret Sharing confidentiality is ITS [10]: a computationally unbounded attacker with at most $k - 1$ shares learns nothing about the secret. In real implementations, the secret S is usually split into several small blocks, and each block is shared using a separate polynomial. This avoids the use of too large prime numbers, computationally costly to handle. Another optimization technique is to use Mersenne primes, of the form $p = 2^m - 1, m \in \mathbb{N}^*$.

Definition 8 (Secret Sharing [41]). *A* (k, n) secret sharing *scheme* $\mathtt{SS}$ *is defined by the following algorithms:*

$\mathtt{SS.Split}(S, k, n) \rightarrow (s_1, \ldots, s_n)$*: on input a secret* S*, returns the shares* $(s_1, \ldots, s_n)$ *according to* n *and* k*.*

$\mathtt{SS.Rec}(k, s_1, \ldots, s_j) \rightarrow S$*: reconstructs the secret from the shares given that* $j \geq k$*, such that* $\mathtt{SS.Rec}(k, \mathtt{SS.Split}(S, k, n)) = S$*.*

It must achieve perfect secrecy.

Definition 9 (Secret Sharing Perfect Secrecy [25]). *A secret sharing scheme* $\mathtt{SS}$ *has* perfect secrecy *if for any* k, n *such that* $1 < k \leq n$*, any* $\mathcal{A}$ *in game* $\boldsymbol{Exp}^{\mathrm{SS}}_{\mathcal{A}}(n, k)$ *(Fig. 3) we have* $\boldsymbol{Adv}^{\mathrm{SS}}(\mathcal{A}) := \left| Pr\left[\boldsymbol{Exp}^{\mathrm{SS}}_{\mathcal{A}}(n, k) = 1\right] - 1/2 \right| = 0$*.*

5.3 Hierarchical Secret Sharing

Shamir's Secret Sharing can be extended to include participant hierarchy [45]. In this paper, we focus on the *trivial case* where decryption needs k participants *including at least one privileged participant*. Let n be the participants count, including $m \leq n$ privileged participants, and S be the initial secret. The dealer chooses a prime $p > S$, and generates a random polynomial $P \in \mathbb{F}_p[X]$ of degree $k - 1$ such that $S = P(0)$, then shares $P(1), \ldots P(m)$ to the privileged participants. He computes the derivative polynomial P': for $P(X) =$

$\mathbf{Exp}^{\mathrm{SS}}_{\mathcal{A}}(n,k)$	$\mathbf{Exp}^{\mathrm{HSS}}_{\mathcal{A}}(m,n,k)$
$1:\ b \xleftarrow{\$} \{0,1\}$	$1:\ b \xleftarrow{\$} \{0,1\}$
$2:\ S^0, S^1, \mathcal{I} \leftarrow \mathcal{A}(\cdot)$	$2:\ S^0, S^1, \mathcal{I}, \mathcal{I}' \leftarrow \mathcal{A}(\cdot)$
$3:\ \{s_i\}_{i=1}^{n} \leftarrow \texttt{SS.Split}(S^b,n,k)$	$3:\ \{s_i\}_{i=1}^{m} \leftarrow \texttt{HSS.SplitPriv}(S^b,m,k)$
$4:\ b' \leftarrow \mathcal{A}(\{s_i\}_{i\in\mathcal{I}})$	$4:\ \{s'_i\}_{i=1}^{n} \leftarrow \texttt{HSS.SplitUnpriv}(S^b,n,k)$
$5:$ **If** $\lvert\mathcal{I}\rvert < k$	$5:\ b' \leftarrow \mathcal{A}(\{s_i\}_{i\in\mathcal{I}}, \{s'_i\}_{i\in\mathcal{I}'})$
$6:$ **return** $b = b'$	$6:$ **If** $(\lvert\mathcal{I}\rvert + \lvert\mathcal{I}'\rvert < k) \vee (\lvert\mathcal{I}\rvert = 0)$
$7:$ **return** b	$7:$ **return** $b = b'$
	$8:$ **return** b

Fig. 3: Security games for secret sharing and trivial hierarchical secret sharing perfect secrecy. In $\mathbf{Exp}^{\mathrm{SS}}_{\mathcal{A}}$, $\mathcal{I}$ denotes a set of adversary-chosen shares. In $\mathbf{Exp}^{\mathrm{HSS}}_{\mathcal{A}}$, $\mathcal{I}$ is a set of privileged shares, and $\mathcal{I}'$ a set of unprivileged shares.

$a_{k-1}X^{k-1}+\ldots+a_1X+a_0 \in \mathbb{F}_p[X]$, it is $P'(X) = (k-1)a_{k-1}X^{k-2}+\ldots+a_1$. The dealer distributes $P'(m+1), \ldots, P'(n)$ to the non-privileged participants. Thus, even if all non-privileged participants pooled their shares, they would only be able to recover P', and could never compute $P(0)$. With at least one privileged participant, they can recover the secret $S = P(0)$ by Birkhoff interpolation [6]. As Shamir secret sharing, hierarchical secret sharing is ITS [46].

Definition 10 (Trivial Hierarchical Secret Sharing). *A* (k, m, n) trivial hierarchical secret sharing *scheme* `HSS` *is defined by:*

- `HSS.SplitPriv(S,k,m)` $\rightarrow$ $(s_1, \ldots, s_m)$*: on input a secret* S*, returns the privileged shares* $(s_1, \ldots, s_m)$ *according to* m *and* k*.*
- `HSS.SplitUnpriv(S,k,n)` $\rightarrow$ $(s'_1, \ldots, s'_n)$*: on input a secret* S*, returns the unprivileged shares* $(s'_1, \ldots, s'_n)$ *according to* n *and* k*.*
- `HSS.Rec`$(k, s_1, s'_1 \ldots, s'_{j-1})$ $\rightarrow$ S*: reconstructs the secret from the shares given that* $j \geq k$ *such that:*
  `HSS.Rec`$(k,$ `HSS.SplitPriv`$(S,k,m),$ `HSS.SplitUnpriv`$(S,k,n)) = S$.

It must achieve perfect secrecy.

Definition 11 (Trivial Hierarchical Secret Sharing Perfect Secrecy). *A trivial hierarchical secret sharing scheme* `HSS` *has* perfect secrecy *if for any* k, m, n*, any* $\mathcal{A}$ *in game* $\boldsymbol{Exp}^{\mathrm{HSS}}_{\mathcal{A}}(m,n,k)$ *(Fig. 3) we have:*

$\boldsymbol{Adv}^{\mathrm{HSS}}(\mathcal{A}) := \left|Pr\left[\boldsymbol{Exp}^{\mathrm{HSS}}_{\mathcal{A}}(m,n,k) = 1\right] - 1/2\right| = 0.$

5.4 COPRIS Protocol

In MULTISS, we rely on the COPRIS protocol for authenticity and integrity. COPRIS was introduced in the LINCOS original paper [9], as the service guaranteeing authenticity and long-term integrity of the secret, stored on a single QKD network. Fig. 4 gives a diagram of the different parties involved in the LINCOS protocol. In order to introduce COPRIS, we first briefly outline how the confidentiality protocol of LINCOS works.

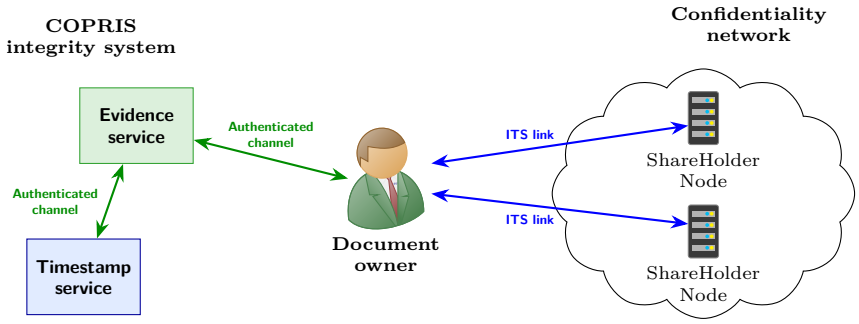


Fig. 4: Parties involved in the LINCOS protocol, on the left the COPRIS service for proof of integrity and authenticity, and on the right the storage of the document as Shamir shares between the different shareholder nodes. The MULTISS protocol uses COPRIS for integrity and authenticity.

Long-time Confidential Secret Storage. The confidentiality protocol of LINCOS requires ITS secure channels between a network of *Shareholders* nodes and the document owner. The latter generates as many Shamir shares of his secret document S as there are shareholder. He chooses the decryption threshold based on security versus availability tradeoff, and sends to each of the nodes a share via the ITS links. Since QKD-secured transfer and Shamir Secret Sharing guarantee ITS security, the resulting combination remains ITS. To update node shares without reconstructing the secret, LINCOS uses the `Reshare` procedure, described in [19]. The document owner generates a new random polynomial δ such that $\deg(\delta) = \deg(P)$ and $\delta(0) = 0$, then distributes $\delta(1), \ldots, \delta(n)$ to shareholders. For node i, new share is $P_{\text{new}}(i) = P(i) + \delta(i)$, and so $P(0) + \delta(0) = P(0) = S$. The `Reshare` procedure is run periodically so that an adversary must compromise at least k nodes within the same update period.

The COPRIS Integrity and Authenticity Protocol. COPRIS [9] provides integrity and authenticity guarantees by letting the document owner build a *Proof of Integrity* (PI) showing that a document existed at time t while keeping it secret. As shown in Fig. 4, 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. Because computing power of the adversary is assumed to be large, the document owner regularly renews commitments with the evidence service so the adversary cannot break the commitment algorithm before the next renewal. To prove integrity and authenticity of document S to the verifier, the document owner provides the pair (E, R), with E the latest evidence record and R the list of decommitments. The verifier can then ensure that document S existed at time t.

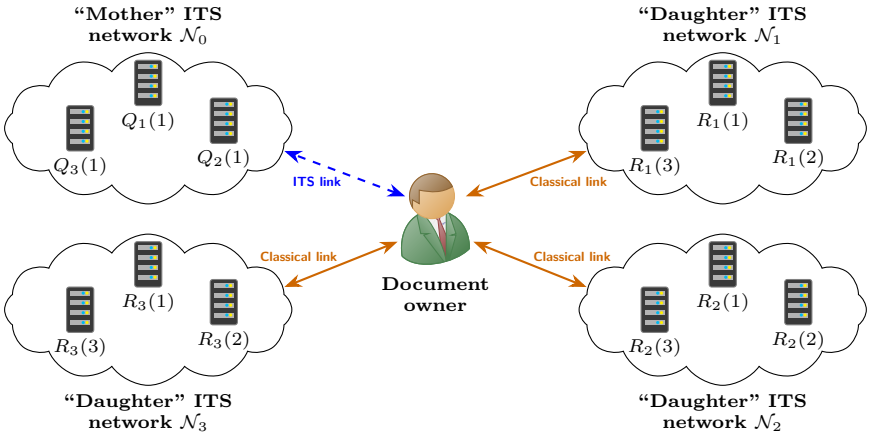


Fig. 5: Local-mode MULTISS: the number $\ell-1$ of daughter subnetworks is equal to the number n_0 of nodes in the mother subnet. Here we have $\ell-1=n_0=3$.

6 MULTISS Protocol

We give two versions of the MULTISS protocol, defined in Fig. 6. In the first version, called the *standard mode*, the distribution can be performed over multiple QKD networks with no constraint on the topology. However, in the event where one of the QKD network is fully compromised, then the protocol must be completely restarted, with all shares being redistributed. We introduce a variant of the protocol called the *local mode*. It is restricted to the case where *the number $\ell-1$ of daughter networks is equal to the number n_0 of nodes in the mother network*, as shown in Fig.5. In this variant, it is possible to renew the value that a compromised subnet stores without changing the values in other subnetworks. To satisfy $\ell-1=n_0$, document owner can leverage only a subset of the daughter subnets or a subset of the mother network nodes. In MULTISS, polynomials are generated modulo p prime chosen and announced by the document owner or at the system level. S is split into blocks S_b such that $S_b<p$. The shares $\{s_{i,j}\}_{i<\ell,j<n_i}$ denote the concatenation of the per-block shares of each block S_b.

Standard-mode MULTISS. This mode leverages two levels of secret sharing. For each block S_b, the document owner generates a polynomial P of chosen degree $\deg(P)$ such that $P(0)=S_b$, then generates ℓ polynomials Q_i of chosen degree $\deg(Q_i)$ such that $Q_0(0)=P(1)$ and $Q_i(0)=P'(i+1)$ for $i=1,\ldots,\ell-1$, where P' is the derivative of P. Finally, each network $\mathcal{N}_{i,0\leq i<\ell}$ stores the points $Q_i(1),\ldots,Q_i(n_i)$ in its respective nodes. To reconstruct the secret, the document owner retrieves shares from $T(P)=\deg(P)-1$ networks, where shares from $\mathcal{N}_0$ are mandatory. $T(Q_i)$ shares of subnet $\mathcal{N}_i$ are required to reconstruct $P(i)$.

Standard-mode MULTISS

Setup$(\{|\mathcal{N}_i|\}_{0\le i<\ell},\, t_{\text{nodes}}, t_{\text{nets}}, t_{\text{fail}})$:
1 : Choose prime p
2 : Define $T(P), \{T(Q_i)\}_{i<\ell}$
3 : **return** $\mathtt{pp} = (p, T(P), \{T(Q_i)\}_{i<\ell})$

Distribute$(\mathtt{pp},\, S_b)$:
1 : $P \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(P) = T(P) - 1$, $P(0) = S_b$
2 : $Q_0 \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(Q_0) = T(Q_0) - 1$, $Q_0(0) = P(1)$
3 : **for** $j \in [0, n_0 - 1]$: $s_{0,j} \leftarrow Q_0(j)$
4 : **for** $i \in [1, \ell - 1]$:
5 : $\quad Q_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(Q_i) = T(Q_i) - 1$, $Q_i(0) = P'(i+1)$
6 : $\quad$ **for** $j \in [0, n_i - 1]$: $s_{i,j} \leftarrow Q_i(j)$
7 : **return** $\{s_{i,j}\}_{0\le i<\ell, 0\le j<n_i}$

Reconstruct$(\mathtt{pp},\, \{s_{i,j}\})$:
1 : $P(1) \leftarrow$ Interp. $\{s_{0,j}\}_{j<n_0}$
2 : **for** each $\mathcal{N}_{i\ge 1}$:
3 : $\quad P'(i+1) \leftarrow$ Interp. $\{s_{i,j}\}_{j<n_i}$
4 : $P \leftarrow$ Interp. $P(1)$ and $\{P'(i+1)\}_{1<i<\ell}$
5 : **return** $P(0)$

RenewGen$(\mathtt{pp},\, i)$:
1 : $\delta_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(\delta_i) = T(Q_i) - 1$, $\delta_i(0) = 0$
2 : **for** $j \in [0, n_i - 1]$: $\Delta s_{i,j} \leftarrow \delta_i(j)$
3 : **return** $\{\Delta s_{i,j}\}_{j<n_i}$

RenewGenNetwork$(\mathtt{pp})$:
1 : $S_b \leftarrow$ **Reconstruct**$(\mathtt{pp}, \{s_{i,j}\})$
2 : **return Distribute**$(\mathtt{pp},\, S_b)$

RenewUpdate$(s_{i,j}, \Delta s_{i,j})$:
1 : $s'_{i,j} \leftarrow s_{i,j} + \Delta s_{i,j} \pmod p$
2 : **return** $s'_{i,j}$

Local-mode MULTISS

Setup$(\{|\mathcal{N}_i|\}_{0\le i<\ell},\, t_{\text{nodes}}, t_{\text{nets}}, t_{\text{fail}})$:
1 : Choose prime p
2 : Define $T(P), \{T(R_i)\}_{1\le i<\ell}$
3 : **return** $\mathtt{pp} = (p, T(P), \{T(R_i)\}_{1\le i<\ell})$

Distribute$(\mathtt{pp},\, S_b)$:
1 : $P \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(P) = T(P) - 1$, $P(0) = S_b$
2 : **for** $i \in [1, n_0]$:
3 : $\quad Q_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(Q_i) = 1$, $Q_i(0) = P(i)$
4 : $\quad s_{0,i} \leftarrow Q_i(1)$
5 : $\quad R_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(R_i) = T(R_i) - 1$, $R_i(0) = Q'_i(2)$
6 : $\quad$ **for** $j \in [0, n_i - 1]$: $s_{i,j} \leftarrow R_i(j)$
7 : **return** $\{s_{i,j}\}_{0\le i<\ell, 0\le j<n_i}$

Reconstruct$(\mathtt{pp},\, \{s_{i,j}\})$:
1 : **for** each $\mathcal{N}_{i\ge 1}$:
2 : $\quad Q'_i(2) \leftarrow$ Interp. $\{s_{i,j}\}_{j<n_i}$
3 : $\quad Q_i \leftarrow$ Interp. $s_{0,i}$ and $Q'_i(2)$
4 : $\quad P(i) \leftarrow Q_i(0)$
5 : $P \leftarrow$ Interp. $\{P(i)\}_{0<i<n_0}$
6 : **return** $P(0)$

RenewGen$(\mathtt{pp},\, i \ge 1)$:
1 : $\delta_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(\delta_i) = T(R_i) - 1$, $\delta_i(0) = 0$
2 : **for** $j \in [0, n_i - 1]$: $\Delta s_{i,j} \leftarrow \delta_i(j)$
3 : **return** $\{\Delta s_{i,j}\}_{j<n_i}$

RenewGenNetwork$(\mathtt{pp},\, i)$:
1 : **if** $i = 0$:
2 : $\quad S_b \leftarrow$ **Reconstruct**$(\mathtt{pp}, \{s_{i,j}\}_{0\le i<\ell})$
3 : $\quad$ **return Distribute**$(\mathtt{pp},\, S_b)$
4 : **else** :
5 : $\Delta Q_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(\Delta Q_i) = 1$, $\Delta Q_i(0) = 0$
6 : $\Delta s_{0,i} \leftarrow \Delta Q_i(1)$
7 : $R'_i \xleftarrow{\$} \mathbb{F}_p[X]$, $\deg(R'_i) = T(R_i) - 1$, $R'_i(0) = \Delta Q'_i(2)$
8 : **for** $j \in [0, n_i - 1]$: $\Delta s_{i,j} \leftarrow R'_i(j)$ - $s_{i,j}$
9 : **return** $\Delta s_{0,i}$, $\{R'_i(j)\}_{j<n_i}$

RenewUpdate$(s_{i,j}, \Delta s_{i,j})$:
1 : $s'_{i,j} \leftarrow s_{i,j} + \Delta s_{i,j} \pmod p$
2 : **return** $s'_{i,j}$

Fig. 6: Standard and local mode MULTISS algorithms instantiations. Similar steps are highlighted in gray.

We can fully analyze the thresholds of the protocol. The number t_{nets} of subnets to compromise is $T(P)$. The number t_{nodes} of nodes to compromise is

$$t_{\text{nodes}} = T(Q_0) + \min\{\sum_{i\in\mathcal{I}}(T(Q_i))\,|\mathcal{I} \subset \{0,\ldots,\ell-1\},\, |\mathcal{I}| = T(P)-1\}$$

The corresponding strategy for the attacker is to recover the minimum evaluations of P, that is, $T(P)$, by breaking the security of the ITS subnets $\mathcal{N}_i$ where the degrees of polynomials Q_i are the smallest. The failure threshold is achieved when either $P(1)$ cannot be reconstructed, that is, $t_{f_0} = n_0 - T(Q_0) + 1$ shares are unavailable in the mother network $\mathcal{N}_0$, or less than $T(P)-1$ shares of P are available, that is, $n_i - T(Q_i) + 1$ shares are lost in the networks $\mathcal{N}_i$ for $i \in \mathcal{I}$ such that $|\mathcal{I}| = \ell - T(P) + 1$. Thus we have $t_{\text{fail}} = \min\{t_{f_0}, t_{f_1}\}$, with

$$t_{f_1} = \min\{\sum_{i\in\mathcal{I}} n_i - T(Q_i) + 1 | \mathcal{I} \subset \{0,\ldots,\ell-1\},\, |\mathcal{I}| = \ell - T(P) + 1\}.$$

Local-mode MULTISS. This mode uses three levels of secret sharing. For each secret block S_b, the document owner first generates a random polynomial P such that $P(0) = S_b$, where $\deg(P)$ depends on the targeted values of t_{nodes} and t_{nets}. Next, for each node i in mother network $\mathcal{N}_0$, 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 $i = j + 1$ in $\mathcal{N}_0$. He then computes Q'_i and stores the value $Q'_i(2)$ in subnet $\mathcal{N}_i$ using a local Shamir scheme: for each daughter subnet i he generates a new polynomial R_i, with $R_i(0) = Q'_i(2)$, and distributes $R_i(j)$ for $j = 1,\ldots,n_i$ to all the nodes j of subnet $\mathcal{N}_{i\geq 1}$.

We can again analyze the thresholds of the protocol. The threshold t_{nets} 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

$$t_{\text{nodes}} = T(P) + \min\{\sum_{i\in\mathcal{I}}(T(R_i))|\mathcal{I} \subset \{0,\ldots,\ell-1\},\, |\mathcal{I}| = T(P)\}.$$

The corresponding strategy for the attacker is to recover $T(P)$ evaluations of P by breaking the security of the ITS subnets $\mathcal{N}_i$ where the degrees of polynomials R_i are the smallest and the associated nodes within the mother network. The failure threshold is achieved when $n_0 - T(P) + 1$ shares of P are unavailable. A share of P is unavailable when either a share is unavailable in the mother network $\mathcal{N}_0$, which makes $Q_i(1)$ unavailable, or $n_i - T(R_i) + 1$ nodes are unavailable in the network $\mathcal{N}_i$, which makes $Q'_i(2)$ unavailable. The best strategy for the attacker is to shut down nodes in the mother subnet, thus $t_{\text{fail}} = n_0 - T(P) + 1$.

For both standard and local mode MULTISS, we provide a `Setup` script to compute polynomials configuration, depending on network topology and defined t_{nodes}, t_{nets} and t_{fail} thresholds, in [37].

Shares update. As shown in Fig. 6, the `RenewGen` procedure distinguishes two cases: share renewal of daughter subnets, and of the mother subnet. The share

update within a daughter network 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, and the procedure is therefore the same. Moreover, each daughter network can be updated independently. Assume that the subnet $\mathcal{N}_i$ stores some value using a polynomial R_i. We generate a new polynomial δ_i of same degree, such that $\delta_i(0) = 0$. Then, for each node j that stores the value $R_i(j)$, we add $\delta_i(j)$: $R_{i,\text{new}}(k) = R_i(k) + \delta_i(k)$. This can be performed directly by a node inside that QKD subnet, to ensure full-ITS renewal.

The share update of the mother subnetwork is different in standard and local mode. In standard mode, the mother subnet 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, the document owner must reconstruct the document and restart the distribution process, since it is impossible to renew the shares of the mother subnet independently of the daughters.

Subnet value update. In case one of the daughter subnet is fully compromised, and the attacker learned the stored value, document owner must initiate a `RenewGenNetwork` procedure, so the information obtained by the attacker becomes obsolete. The procedure differs significantly in standard and local mode. In local mode, the network $\mathcal{N}_i$ stores the value $Q'_i(2)$ using a local Shamir's 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 $\mathcal{N}_i$, 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 changes the value $Q_i(1)$ stored in the mother network, and all the shares stored in the daughter subnet $\mathcal{N}_i$. In standard mode, the procedure is more involved. As the daughter network $\mathcal{N}_i$ stores the value $Q_i(0) = P'(i+1)$ using a local Shamir secret sharing, it is required to change all the shares in all subnetworks. The document owner must therefore retrieve the original document and restart a distribution process.

7 Security Analysis

In sections 3 and 4, we defined two types of adversaries: the adversary capable of compromising an entire ITS subnet, and the HNDL adversary. We analyze the security of MULTISS against these two adversaries independently, and also prove the post-compromise secrecy. We assume that authentication is secure during the execution of the protocol. To formally prove the security of MULTISS, we use a sequence of games approach [43]. Unlike computational security proofs where transitions bound the advantage by negligible functions, ours rely on perfect statistical indistinguishability.

7.1 Subnet Compromise

We first consider an attacker who can take control of all nodes of $t_{\text{nets}}-1$ arbitrary ITS subnetworks. Since the secret S cannot be recovered without the value stored

in the mother network, we assume that it is among the compromised subnets, along with $t_{\text{nets}} - 2$ daughter subnets. We assume that this attacker doesn't perform any HNDL attack against classical communications.

Theorem 1. *The standard-mode MULTISS protocol confidentiality against a "Subnet Compromise" adversary that can compromise all nodes of up to $t_{nets}-1$ subnetworks, including the mother subnet (meaning $t_{nets} \geq 2$), is ITS.*

Proof Sketch. We first prove that the shares of the compromised daughter subnetworks can be replaced by random values without affecting the adversary's advantage, using a sequence of reductions to $\mathbf{Exp}_{\mathcal{A}}^{\text{HSS}}$ for each daughter subnet. Then we prove that the shares of the compromised mother subnet can also be replaced by random values, with reductions to $\mathbf{Exp}_{\mathcal{A}}^{\text{SS}}$ for each mother subnet share. The full proof is available in Appendix A.1.

Theorem 2. *The local-mode MULTISS protocol confidentiality is ITS against a "Subnet Compromise" adversary that can compromise all nodes of up to $t_{nets}-1$ subnetworks, including the mother subnet.*

The full proof, analogous to the one of Theorem 1, appears in Appendix A.2.

7.2 HNDL attacker

An HNDL adversary listens to the encrypted communications, hoping to decrypt them when computational power and cryptanalysis are sufficient. In the worst scenario, this adversary decrypts all communications transmitted by non-ITS channels, and retrieves all the information contained in the daughter subnets $\mathcal{N}_1, \mathcal{N}_2, \ldots$. However, this adversary has no information about the shares stored on the mother subnet $\mathcal{N}_0$.

Theorem 3. *Both MULTISS versions are ITS against a HNDL adversary.*

Proof sketch. We show that the shares of the compromised daughter subnetworks can be replaced by random values without affecting the adversary's advantage, using a sequence of reductions to $\mathbf{Exp}_{\mathcal{A}}^{\text{HSS}}$ for each daughter subnet. The full proof is available in Appendix A.3.

7.3 Mobile Attacker

We now consider a *mobile adversary* that can take control of entire ITS subnetworks at different times, in the limit of $t_{\text{nets}} - 1$ subnets at a time, or all daughter subnets simultaneously (to simulate a mobile HNDL attack). We assume that the document owner performs a network-wide shares renewal after each compromise, using `RenewGenNetwork`.

Theorem 4. *Both versions of the MULTISS protocol guarantees ITS post compromise secrecy against a mobile adversary that can compromise all nodes of up to $t_{nets}-1$ subnetworks, including the mother subnet, or all the daughter subnets, at different times.*

Proof sketch. We show that each leaked renewed share can be replaced by a random value without affecting the adversary's advantage, using a sequence of reductions to $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{HSS}}$ for each leaking round, then we reuse the proof of Theorems 2 and 3 to show that the initial shares can also be replaced by random values. The full proof is available in Appendix A.4, as well as next corollary.

Corollary 1. *Secrecy against a mobile adversary implies both HNDL and subnet compromise secrecy.*

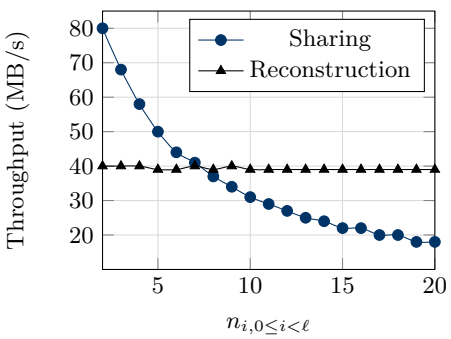
8 Implementation

We tested our protocol with a prototype implementation in Rust, that proves the correctness of our scheme. In the context of FranceQCI, the french project for a quantum communication infrastructure, MULTISS has been deployed on two QKD testbeds, each with two nodes. The implementation relies on the `malachite` and `rug` crates for big integer arithmetic. We employ the Mersenne prime $p = 2^{9689} - 1$, and the secret file is split into 9688-bit blocks. For performance optimization, we zero-pad each block to the next multiple of 64 bits, resulting in a share block size of 9728 bits. Thus the storage requirement is $|S| \cdot N \cdot \frac{9728}{9688}$, where N is the total number of nodes. We employ the Horner's polynomial evaluation method [22] to compute shares, whose complexity for a polynomial of degree d is $O(d)$. Birkhoff interpolation is $O(d^3)$, while Lagrange interpolation is $O(d^2)$. Implementation code is available in [28].

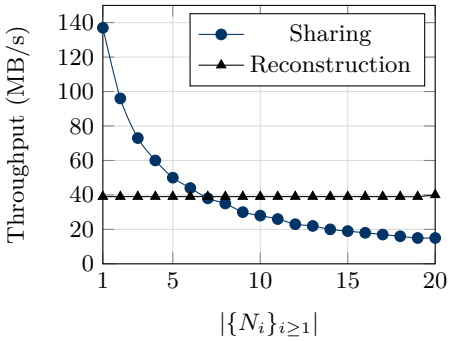
We conducted our experiments on a computer equipped with an Intel Core i5-9500 3.00GHz CPU and 16GB of RAM. For better performances we compile with the native CPU features flag `target-cpu=native`, and `-O3` optimization. We present the result throughput for standard-mode MULTISS, depending on the number of nodes in subnets, the number of subnets and the polynomials degrees, in Fig. 7. Local-mode throughput, depending on the number of daughter subnets and polynomial degrees, is presented in Appendix B. All throughput are relative to the input file size: a sharing throughput of 1 MB/s means that MULTISS will take one second to store a 1 MB file. As expected, sharing performance decreases as polynomial degrees, subnets count or nodes count per subnet increases. On the other hand, reconstruction time increases only with polynomial degrees.

9 Discussion

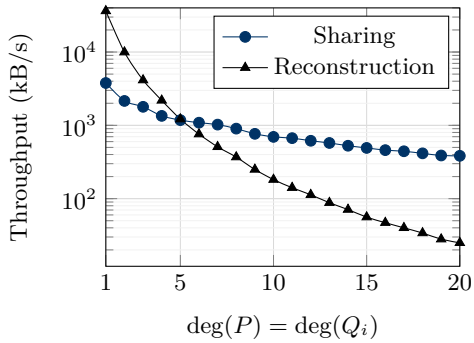
The protocol we introduced is flexible and can be adapted to real-world constraints and opportunities that stem from QKD networks. We provide two examples here. Firstly, the whole ITS argumentation can easily be transposed to the case where QKD is combined with AES encryption. While this combination 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 quantum-safe [33], and when combined with QKD, provides *post-compromise security*, which means that compromising a session



(a) Standard-mode MULTISS throughput depending on the number of nodes $n_{i,0\leq i<\ell}$ in each of the subnets. In our experiment, we simulated five daughter subnets ($\ell = 5+1 = 6$), with $\deg(P) = \deg(Q_i) = 1$.



(b) Standard-mode MULTISS throughput depending on daughter subnets count $|\{N_i\}_{i\geq 1}|$. In our experiment, we simulated $n_i = 5$ nodes for each subnet i, including the mother, with $\deg(P) = \deg(Q_i) = 1$.



(c) Standard-mode MULTISS throughput depending on degrees of polynomials P and Q_i. In our experiment we configured $\deg(P) = \deg(Q_i)$ for all i, and simulated 20 daughter subnets, with 21 nodes in each subnet (including the mother subnet).

Fig. 7: Standard-mode MULTISS sharing and reconstruction throughput, depending on network topology and polynomial degrees.

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. Another interesting extension is to consider subnets connected to each other using satellite QKD [30, 24]. 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 renewal does not require secure

communication with all the subnets, but only between the mother network and the compromised subnet.

10 Conclusion

We introduced a new protocol, MULTISS, to extend the secure storage over several remote QKD networks. 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 subnet and perform "*Harvest Now, Decrypt Later*" attacks on the classical links between the subnets, at the same time. Although some shares transit through links protected by classical cryptography only, our secret document thus remains ITS as long as the adversary does not break into the mother subnet.

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. He can adjust the distribution of shares according to the chosen t_{nodes}, t_{nets} and t_{fail} thresholds. The two versions of the protocol we introduced satisfy different constraints. In standard mode, it is possible to update the shares in any subnetwork, but requiring communication between the mother subnet and all daughter subnets. In local mode, it is not possible to update the shares of the mother subnet without updating all subnets, but updating a daughter subnet after full leakage only requires one round of communication with the mother subnet.

Acknowledgments. T. Prévost acknowledges PhD funding from UCA. The authors are grateful to Francesco Mazzoncini [7] and Bruno Martin [8] and for theoretical support.

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. This work has been partially supported by the European Astral Project ERASMUS-EDU-2024:101184483 and the French National Research Agency through the PRIvacy-preserving tools for VAlidation and Security of Queries (PRIVASIQ) project (ANR-23-CE39-0008).

References

1. Abu-Libdeh, H., Princehouse, L., Weatherspoon, H.: RACS: a case for cloud storage diversity. In: Proceedings of the 1st ACM symposium on Cloud computing (2010). https://doi.org/10.1145/1807128.1807165

[7] https://orcid.org/0000-0003-0589-400X
[8] https://orcid.org/0000-0002-0048-5197

2. Amirloo, J., Razavi, M., Majedi, A.H.: Quantum key distribution over probabilistic quantum repeaters. Physical Review A—Atomic, Molecular, and Optical Physics (2010). https://doi.org/10.1103/PhysRevA.82.032304
3. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. Theoretical computer science (1984). https://doi.org/10.1016/j.tcs.2014.05.025
4. Bennett, C.H., Brassard, G., Mermin, N.D.: Quantum cryptography without Bell's theorem. Physical review letters (1992). https://doi.org/10.1103/PhysRevLett.68.557
5. Beullens, W.: Breaking rainbow takes a weekend on a laptop. In: Annual International Cryptology Conference. Springer (2022). https://doi.org/10.1007/978-3-031-15979-4_16
6. Birkhoff, G.D.: General mean value and remainder theorems with applications to mechanical differentiation and quadrature. Transactions of the American Mathematical Society (1906). https://doi.org/10.2307/1986339
7. Boaron, A., et al.: Secure quantum key distribution over 421 km of optical fiber. Physical review letters (2018). https://doi.org/10.1103/PhysRevLett.121.190502
8. Bowers, K.D., Juels, A., Oprea, A.: HAIL: A high-availability and integrity layer for cloud storage. In: Proceedings of the 16th ACM conference on Computer and communications security (2009). https://doi.org/10.1145/1653662.1653686
9. Braun, J., et al.: 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). https://doi.org/10.1145/3052973.3053043
10. 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). https://doi.org/10.1109/InfoSEEE.2014.6948065
11. Costan, V., Devadas, S.: Intel sgx explained. Cryptology ePrint Archive (2016), `https://ia.cr/2016/086`
12. Doosti, M., Hanouz, L., Marin, A., Kashefi, E., Kaplan, M.: Establishing shared secret keys on quantum line networks: protocol and security. In: 2024 International Conference on Quantum Communications, Networking, and Computing (QCNC). IEEE (2024). https://doi.org/10.1109/QCNC62729.2024.00035
13. Duan, L.M., Lukin, M.D., Cirac, J.I., Zoller, P.: Long-distance quantum communication with atomic ensembles and linear optics. Nature (2001). https://doi.org/10.1038/35106500
14. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Physical review letters (1991). https://doi.org/10.1103/PhysRevLett.67.661
15. Fischlin, M., Günther, F., Muth, P.: Information-theoretic security of cryptographic channels. In: International Conference on Information and Communications Security. Springer (2020). https://doi.org/10.3929/ethz-b-000438741
16. Fujiwara, M., et al.: Unbreakable distributed storage with quantum key distribution network and password-authenticated secret sharing. Scientific reports (2016). https://doi.org/10.1038/srep28988
17. Fujiwara, M., et al.: Long-term secure distributed storage using quantum key distribution network with third-party verification. IEEE Transactions on Quantum Engineering (2021). https://doi.org/10.1109/TQE.2021.3135077
18. Fujiwara, M., et al.: Secure secondary utilization system of genomic data using quantum secure cloud. Scientific reports (2022). https://doi.org/10.1038/s41598-022-22804-x

19. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: annual international cryptology conference. Springer (1995). https://doi.org/10.1007/3-540-44750-4_27
20. Honjo, T., et al.: Long-distance entanglement-based quantum key distribution over optical fiber. Optics Express (2008). https://doi.org/10.1364/OE.16.019118
21. Horkos, D., Perret, L.: Sok: On cryptography for multi-cloud storage. Cryptology ePrint Archive (2026), https://ia.cr/2026/207
22. Horner, W.G.: A new method of solving numerical equations of all orders, by continuous approximation. Philosophical Transactions of the Royal Society of London (1819). https://doi.org/10.1098/rstl.1819.0023
23. IDQuantique: Medical data successfully protected by quantum cryptography in Graz. https://www.idquantique.com/openqkd-id-quantique-and-fragmentix-successfully-apply-qkd-for-a-medical-use-case/ (2020)
24. Islam, T., et al.: Finite-resource performance of small-satellite-based quantum-key-distribution missions. PRX Quantum (2024). https://doi.org/10.1103/PRXQuantum.5.030101
25. Karnin, E., Greene, J., Hellman, M.: On secret sharing systems. IEEE Transactions on Information Theory (1983). https://doi.org/10.1109/TIT.1983.1056621
26. Kursawe, K., et al.: Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage. In: 2009 IEEE International Workshop on Hardware-Oriented Security and Trust. IEEE (2009). https://doi.org/10.1109/HST.2009.5225058
27. Lafourcade, P., Mallordy, L.B., Olivier-Anclin, C., Robert, L.: Secure keyless multi-party storage scheme. In: European Symposium on Research in Computer Security. Springer (2024). https://doi.org/10.1007/978-3-031-70896-1_14
28. Lebrin, C., Prévost, T., Kaplan, M., Veriqloud: MULTISS arithmetics (2026). https://doi.org/10.5281/zenodo.22690361, also available at https://github.com/thomasarmel/multiss_arithmetics
29. Li, B.H., et al.: Long-distance twin-field quantum key distribution with entangled sources. Optics Letters (2021). https://doi.org/10.1364/OL.443099
30. Liao, S.K., et al.: Satellite-to-ground quantum key distribution. Nature (2017). https://doi.org/10.1038/nature23655
31. Moody, D., Perlner, R., Regenscheid, A., Robinson, A., Cooper, D.: Transition to post-quantum cryptography standards. Tech. rep., National Institute of Standards and Technology (2024). https://doi.org/10.6028/NIST.IR.8547.ipd
32. Niknia, A., Correia, M., Karimpour, J.: Secure cloud-of-clouds storage with space-efficient secret sharing. Journal of Information Security and Applications (2021). https://doi.org/10.1016/j.jisa.2021.102826
33. NIST: Security (Evaluation Criteria) - Post-Quantum Cryptography | CSRC | CSRC — csrc.nist.gov. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria) (2017), [Accessed 29-05-2026]
34. Papaioannou, T.G., Bonvin, N., Aberer, K.: Scalia: An adaptive scheme for efficient multi-cloud storage. In: SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE (2012). https://doi.org/10.1109/SC.2012.101
35. Paul, S.: On the Transition to Post-Quantum Cryptography in the Industrial Internet of Things. Ph.D. thesis, Technische Universität Darmstadt (2022). https://doi.org/10.26083/tuprints-00021368

36. Pelet, Y., et al.: Operational entanglement-based quantum key distribution over 50 km of field-deployed optical fibers. Physical Review Applied (2023). https://doi.org/10.1103/PhysRevApplied.20.044006
37. Prévost, T.: MULTISS topology calculator (2026). https://doi.org/10.5281/zenodo.22690206, also available at `https://github.com/thomasarmel/multiss_topology_calculator/`
38. Ribezzo, D., et al.: Deploying an inter-european quantum network. Advanced Quantum Technologies (2023). https://doi.org/10.1002/qute.202200061
39. Ribezzo, D., et al.: Quantum key distribution over 100 km of underwater optical fiber assisted by a fast-gated single-photon detector. Physical Review Applied (2023). https://doi.org/10.1103/PhysRevApplied.20.044052
40. Robert, D.: Breaking SIDH in polynomial time. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer (2023). https://doi.org/10.1007/978-3-031-30589-4_17
41. Shamir, A.: How to share a secret. Communications of the ACM (1979). https://doi.org/10.1145/359168.359176
42. Shannon, C.E.: Communication theory of secrecy systems. The Bell system technical journal (1949). https://doi.org/10.1002/j.1538-7305.1949.tb00928.x
43. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. cryptology eprint archive (2004), `https://ia.cr/2004/332`
44. Straznickas, Z., Weis, S.A.: HAWK-n Key Recovery Reduces to SVP in Dimension $n/2+1$. Cryptology ePrint Archive (2026), `https://ia.cr/2026/1593`
45. Tassa, T.: Hierarchical threshold secret sharing. In: Theory of Cryptography Conference. Springer (2004). https://doi.org/10.1007/978-3-540-24638-1_26
46. 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). https://doi.org/10.1007/978-3-319-49175-2_2
47. Unruh, D.: Everlasting multi-party computation. In: Annual Cryptology Conference. Springer (2013). https://doi.org/10.1007/978-3-642-40084-1_22
48. Waring, E.: Vii. problems concerning interpolations. Philosophical transactions of the royal society of London (1779). https://doi.org/10.1098/rstl.1779.0008
49. Wegman, M.N., Carter, J.L.: New hash functions and their use in authentication and set equality. Journal of computer and system sciences (1981). https://doi.org/10.1016/0022-0000(81)90033-7
50. Xagawa, K., Ito, A., Ueno, R., Takahashi, J., Homma, N.: Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer (2021). https://doi.org/10.1007/978-3-030-92075-3_2
51. Xiong, H., Zhang, X., Zhu, W., Yao, D.: Cloudseal: End-to-end content protection in cloud-based storage and delivery services. In: International Conference on Security and Privacy in Communication Systems. Springer (2011). https://doi.org/10.1007/978-3-642-31909-9_30
52. Zygelman, B., Zygelman, B.: No-cloning theorem, quantum teleportation and spooky correlations. A First Introduction to Quantum Computing and Information (2018). https://doi.org/10.1007/978-3-031-66425-0_6

A Proofs of Theorems

A.1 Proof of Theorem 1

We prove this theorem using a sequence of games. We denote by $\mathcal{A}$ the computationally unbounded "Subnet Compromise" adversary, and we consider the security game $\mathbf{Exp}^{\mathrm{SC}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$ defined in Fig.2. Furthermore we consider the case where the mother network is among the compromised networks. Otherwise, we show in Theorem 3 that the secret remains ITS if the mother network is not compromised, even if all the daughter networks are compromised.

Game G_0: The original game $\mathbf{Exp}^{\mathrm{SC}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$ instantiated with algorithms of standard-mode MULTISS.

Game G_1: The same game as G_0, except that the shares $\{s_{i,j}\}_{i\geq 1, i\in\mathcal{I}}$ of the compromised daughter subnets are replaced by random values drawn uniformly from $\mathbb{F}_p$.

Game G_2: The same game as G_1, except that the compromised mother subnet shares $\{s_{0,j}\}$ are replaced by random values drawn uniformly from $\mathbb{F}_p$.

Claim. We claim that G_0 and G_1 are perfectly indistinguishable.

In Game G_0, the adversary $\mathcal{A}$ compromises at most $t_{\mathrm{nets}} - 1$ subnetworks. Since by assumption the mother network $\mathcal{N}_0$ is among the compromised ones, the set of compromised daughter networks has a size of at most $t_{\mathrm{nets}} - 2$. We create a sequence of $t_{\mathrm{nets}} - 2$ reductions $\mathcal{R}^1_1, \ldots, \mathcal{R}^{t_{\mathrm{nets}}-2}_1$. The first reduction $\mathcal{R}^1_1$ takes random shares for the first compromised daughter subnet $\mathcal{N}_1$ instead of evaluations of Q_1. Each reduction $\mathcal{R}^i_1$ for $i \in [\![1, t_{\mathrm{nets}} - 2]\!]$ takes random shares for the first i compromised daughter subnets $\mathcal{N}_1, \ldots, \mathcal{N}_i$ instead of evaluations of $Q_1, \ldots, Q_i$. The last reduction $\mathcal{R}^{t_{\mathrm{nets}}-2}_1$ takes random shares for all the compromised daughter subnets. Because for all $i \in [\![1, t_{\mathrm{nets}} - 3]\!]$, $|\mathbf{Adv}(\mathcal{R}^i_1(\mathcal{A})) - \mathbf{Adv}(\mathcal{R}^{i+1}_1(\mathcal{A}))| \leq \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A})$ and $\mathcal{R}^{t_{\mathrm{nets}}-2}_1$ simulates G_1, perfectly, we have $|\mathbf{Adv}_0(\mathcal{A}) - \mathbf{Adv}_1(\mathcal{A})| \leq (t_{\mathrm{nets}} - 2) \cdot \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A}) = 0$.

Claim. We claim that G_1 and G_2 are perfectly indistinguishable.

In Game G_1, the shares $\{s_{i,j}\}_{i\geq 1, i\in\mathcal{I}}$ from the compromised daughter networks have been entirely replaced by uniformly random values. Consequently, the only information correlated to the initial polynomial P is located in the mother network $\mathcal{N}_0$. We create a sequence of n_0 reductions $\mathcal{R}^0_2, \ldots, \mathcal{R}^{n_0-1}_2$. The first reduction $\mathcal{R}^0_2$ takes random shares for the first node of the mother subnet $\mathcal{N}_0$ instead of evaluations of $Q_0(1)$. Each reduction $\mathcal{R}^j_2$ for $j \in [\![0, n_0 - 1]\!]$ takes random shares for the first j nodes of the mother subnet instead of evaluations of $Q_0(j+1)$. The last reduction $\mathcal{R}^{n_0-1}_2$ takes random shares for all the nodes of the mother subnet. Because for all $j \in [\![0, n_0-2]\!]$, $|\mathbf{Adv}(\mathcal{R}^j_2(\mathcal{A})) - \mathbf{Adv}(\mathcal{R}^{j+1}_2(\mathcal{A}))| \leq \mathbf{Adv}^{\mathrm{SS}}(\mathcal{A})$ and $\mathcal{R}^{n_0-1}_2$ simulates G_2, perfectly, we have $|\mathbf{Adv}_1(\mathcal{A}) - \mathbf{Adv}_2(\mathcal{A})| \leq n_0 \cdot \mathbf{Adv}^{\mathrm{SS}}(\mathcal{A}) = 0$. □

A.2 Proof of Theorem 2

We prove this theorem using a sequence of games. We denote by $\mathcal{A}$ the computationally unbounded "Subnet Compromise" adversary, and we consider the security game defined in Fig.2. Furthermore we consider the case where the mother network is among the compromised networks. Otherwise, we show in section 7.2 that the secret remains ITS if the mother network is not compromised, even if all the daughter networks are compromised.

Game G_0: The original game $\mathbf{Exp}^{\mathrm{SC}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$ instantiated with algorithms of local-mode MULTISS.

Game G_1: The same game as G_0, except that the shares $\{s_{i,j}\}_{i\geq 1, i\in\mathcal{I}}$ of the compromised daughter subnets are replaced by random values drawn uniformly from $\mathbb{F}_p$.

Game G_2: The same game as G_1, except that the shares $\{s_{0,j}\}$ of the compromised mother subnet are replaced by random $\mathbb{F}_p$ values drawn uniformly.

Claim. We claim that G_0 and G_1 are perfectly indistinguishable.

In Game G_0, the adversary $\mathcal{A}$ compromises at most $t_{\mathrm{nets}} - 1$, including $|\mathcal{I}| \leq t_{\mathrm{nets}} - 2$ daughter networks. We create a sequence of $t_{\mathrm{nets}} - 2$ reductions $\mathcal{R}_1^1, \ldots, \mathcal{R}_1^{t_{\mathrm{nets}}-2}$. The first reduction $\mathcal{R}_1^1$ takes random shares for the first compromised daughter subnet $\mathcal{N}_1$ instead of sub-shares of $Q'_1(2)$. Each reduction $\mathcal{R}_1^i$ for $i \in [\![1, t_{\mathrm{nets}} - 2]\!]$ takes random shares for the first i compromised daughter subnets $\mathcal{N}_1, \ldots, \mathcal{N}_i$ instead of sub-shares of $Q'_1(2), \ldots, Q'_i(2)$. The last reduction $\mathcal{R}_1^{t_{\mathrm{nets}}-2}$ takes random shares for all the compromised daughter subnets. Because for all $i \in [\![1, t_{\mathrm{nets}} - 3]\!]$, $|\mathbf{Adv}(\mathcal{R}_1^i(\mathcal{A})) - \mathbf{Adv}(\mathcal{R}_1^{i+1}(\mathcal{A}))| \leq \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A})$ and $\mathcal{R}_1^{t_{\mathrm{nets}}-2}$ simulates G_1, perfectly, we have:
$|\mathbf{Adv}_0(\mathcal{A}) - \mathbf{Adv}_1(\mathcal{A})| \leq (t_{\mathrm{nets}} - 2) \cdot \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A}) = 0.$

Claim. We claim that G_1 and G_2 are perfectly indistinguishable.

In Game G_1, the shares $\{s_{i,j}\}_{i\geq 1, i\in\mathcal{I}}$ from the compromised daughter networks have been entirely replaced by uniformly random values over $\mathbb{F}_p$. We create a sequence of n_0 reductions $\mathcal{R}_2^0, \ldots, \mathcal{R}_2^{n_0-1}$. The first reduction $\mathcal{R}_2^0$ takes random shares for the first node of the mother subnet $\mathcal{N}_0$ instead of $Q_1(1)$. Each reduction $\mathcal{R}_2^j$ for $j \in [\![0, n_0 - 1]\!]$ takes random shares for the first j nodes of the mother subnet instead of $Q_j(1)$. The last reduction $\mathcal{R}_2^{n_0-1}$ takes random shares for all the nodes of the mother subnet. Because for all $j \in [\![0, n_0 - 2]\!]$, $|\mathbf{Adv}(\mathcal{R}_2^j(\mathcal{A})) - \mathbf{Adv}(\mathcal{R}_2^{j+1}(\mathcal{A}))| \leq \mathbf{Adv}^{\mathrm{SS}}(\mathcal{A})$ and $\mathcal{R}_2^{n_0-1}$ simulates G_2, perfectly, we have $|\mathbf{Adv}_1(\mathcal{A}) - \mathbf{Adv}_2(\mathcal{A})| \leq n_0 \cdot \mathbf{Adv}^{\mathrm{SS}}(\mathcal{A}) = 0$. □

A.3 Proof of Theorem 3

We prove this theorem using a sequence of games. We denote by $\mathcal{A}$ the computationally unbounded HNDL adversary, and we consider the security game $\mathbf{Exp}^{\mathrm{HNDL}}_{\mathcal{A}}$ defined in Fig.2.

Game G_0: The original security game $\mathbf{Exp}^{\mathrm{HNDL}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}},\ t_{\mathrm{nets}},\ t_{\mathrm{fail}})$ instantiated with MULTISS's algorithms.

Game G_1: The same game as G_0, except that the shares $\{s_{i,j}\}_{i\geq 1}$ of the daughter subnets are replaced by random values drawn uniformly from $\mathbb{F}_p$.

Claim. We claim that G_0 and G_1 are perfectly indistinguishable.

We create a sequence a $\ell - 1$ reductions $\mathcal{R}_1^1, \ldots, \mathcal{R}_1^{\ell-1}$. The first reduction $\mathcal{R}_1^1$ takes random shares for the first daughter subnet $\mathcal{N}_1$ instead of evaluations of Q_1. Each reduction $\mathcal{R}_1^i$ for $i \in [\![1, \ell - 1]\!]$ takes random shares for the first i daughter subnets $\mathcal{N}_1, \ldots, \mathcal{N}_i$ instead of evaluations of $Q_1, \ldots, Q_i$. The last reduction $\mathcal{R}_1^{\ell-1}$ takes random shares for all the daughter subnets. Because for all $i \in [\![1, \ell-2]\!]$, $|\mathbf{Adv}(\mathcal{R}_1^i(\mathcal{A})) - \mathbf{Adv}(\mathcal{R}_1^{i+1}(\mathcal{A}))| \leq \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A})$ and $\mathcal{R}_1^{\ell-1}$ simulates G_1, perfectly, we have $|\mathbf{Adv}_0(\mathcal{A}) - \mathbf{Adv}_1(\mathcal{A})| \leq (\ell - 1)\mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A}) = 0$. □

A.4 Proof of Theorem 4

For standard-mode MULTISS, the proof is straightforward. After each compromise, all the shares are renewed by the `RenewGenNetwork` procedure. This procedure reconstructs the secret and completely redistributes it using new, independent random polynomials. Since the new shares are completely unlinked from the compromised shares, an attacker gains no advantage by combining shares gathered at different times. Therefore, the security remains exactly the same as independent instances of Shamir secret sharing, fully preserving the ITS security against mobile adversaries. For local-mode MULTISS, we formalize the security with the following game hops:

Game G_0: The original $\mathbf{Exp}^{\mathrm{PCS}}_{\mathcal{A}}(\{|\mathcal{N}_i|\}, t_{\mathrm{nodes}}, t_{\mathrm{nets}}, t_{\mathrm{fail}})$ Post-Compromise game instantiated with the algorithms of local-mode MULTISS.

Game G_1: The same game as G_0, except that the update shares $\{s_{i,j}^{(k)}\}_{k\leq q}$ of compromised subnets generated during the `RenewUpdate`(`pp`, i) procedures are entirely replaced by uniform random values drawn from $\mathbb{F}_p$.

Game G_2: The same game as G_1, except that the initial compromised shares $\{s_{i,j}^{(0)}\}$ are replaced by random values drawn uniformly from $\mathbb{F}_p$.

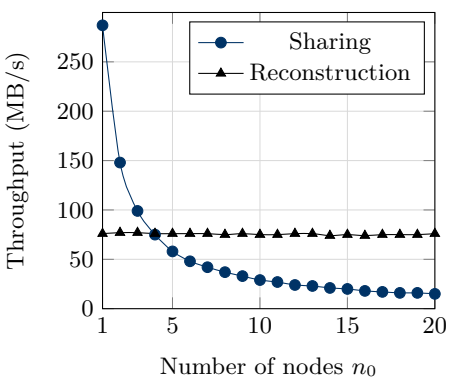
Claim. We claim that G_0 and G_1 are perfectly indistinguishable.

In Game G_0, at each of the q compromise instances the adversary $\mathcal{A}$ may either compromise at most $t_{\mathrm{nets}} - 1$ subnetworks or compromise all $N - 1$ daughter subnets, and the adversary's choice may vary between instances. We create a sequence of q reductions $\mathcal{R}_1^1, \ldots, \mathcal{R}_1^q$. The first reduction $\mathcal{R}_1^1$ takes random updated shares for the first renewal round instead of the genuine updated shares $\{s'_{i,j}\}$. Each reduction $\mathcal{R}_1^k$ for $i \in [\![1, q]\!]$ takes random update shares for the first k renewed shares instead of the genuine update shares generated by the `RenewUpdate` algorithm. The last reduction $\mathcal{R}_1^q$ takes random update shares for all the compromised renewing shares. Because, as shown by Theorem 2, for all $i \in [\![1, q-1]\!]$, $|\mathbf{Adv}(\mathcal{R}_1^i(\mathcal{A})) - \mathbf{Adv}(\mathcal{R}_1^{i+1}(\mathcal{A}))| \leq \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A})$ and $\mathcal{R}_1^q$ simulates G_1, perfectly, we have $|\mathbf{Adv}_0(\mathcal{A}) - \mathbf{Adv}_1(\mathcal{A})| \leq q \cdot \mathbf{Adv}^{\mathrm{HSS}}(\mathcal{A}) = 0$.

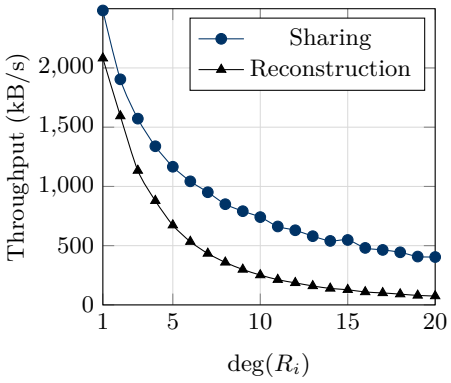
Claim. We claim that G_0 and G_1 are perfectly indistinguishable.
This is proved by Theorems 2 and 3. □

B Local-Mode Protocol Performance

In this section, we present local-mode MULTISS's throughput during our experiments, for different configurations. All the throughput values are computed relative to the input file size.



(a) Throughput depending on the number n_0 of nodes in the mother subnet. Recall that in local-mode, the number $\ell - 1$ of daughter subnets is equal to the number of nodes n_0 in the mother subnet. During our experiment we simulated $n_{i\geq 1} = 2$ nodes in each daughter subnet, and configured $\deg(P) = 0$, $\deg(R_i) = 1$ for all i.



(b) Throughput depending on the degree $\deg(R_i)$ of each polynomial R_i. In our experiment we simulated a mother subnet with 20 nodes (therefore 20 daughter subnets), 21 nodes per daughter subnet, and configured $\deg(P) = 10$.