Topology-Hiding Path Validation for Large-Scale Quantum Key Distribution Networks

Stephan Krenn[1], Omid Mir[1], Thomas Lorünser[1,2], Sebastian Ramacher[1], and Florian Wohner[1]

[1] AIT Austrian Institute of Technology, Vienna, Austria
{firstname.lastname}@ait.ac.at
[2] Digital Factory Vorarlberg GmbH, Dornbirn, Austria

Abstract. Secure long-distance communication in quantum key distribution (QKD) networks depends on trusted repeater nodes along the entire transmission path. Consequently, these nodes will be subject to strict auditing and certification in future large-scale QKD deployments. However, trust must also extend to the network operator, who is responsible for fulfilling contractual obligations–such as ensuring certified devices are used and transmission paths remain disjoint where required. In this work, we present a path validation protocol specifically designed for QKD networks. It enables the receiver to verify compliance with agreed-upon policies. At the same time, the protocol preserves the operator's confidentiality by ensuring that no sensitive information about the network topology is revealed to users.
We provide a formal model and a provably secure generic construction of the protocol, along with a concrete instantiation. For long-distance communication involving 100 nodes, the protocol has a computational cost of $1-2.5$ s depending on the machine, and a communication overhead of less than 70 kB–demonstrating the efficiency of our approach.

Keywords: Path validation · Topology-hiding · QKD networks

1 Introduction

Quantum Key Distribution (QKD) [6] enables secure key exchange by utilizing quantum mechanics, specifically superposition and entanglement, to ensure that any eavesdropping attempt can be detected. Unlike classical cryptographic methods, which rely on the difficulty of certain mathematical problems, QKD is based on the laws of physics and achieves unconditional security, making it resistant against classical and quantum attacks. QKD is thus expected to play a crucial role in highly security-sensitive application domains in the future, and its practicability has been demonstrated in numerous testbeds, e.g., [10,20,26,32].

However, QKD faces limitations, primarily the distance over which quantum signals can travel due to photon loss in optical fibers or through free space. To overcome this limitation, trusted repeaters are employed to relay quantum information over long distances [28,39], typically placed at most every $100-200$ km,

leading to potentially tens of repeaters in a terrestrial long-distance connection. These repeaters allow the key exchange to continue securely, but they also introduce vulnerabilities, as a compromised repeater could potentially intercept or alter the key, thereby undermining the security guarantees of the entire network. Consequently, minimizing the necessary trust required for trusted repeaters remains an active area of research to ensure the scalability and robustness of QKD systems. One such approach is based on so-called *quantum repeaters*, which aim to enable secure long-distance quantum communication without the need for trusted intermediate nodes, leveraging quantum entanglement [3]. However, as of today, quantum repeaters are still in the early experimental stage with significant challenges remaining, including the development of efficient quantum memories and the integration of all required components into a stable and scalable system.

Another approach to reduce the necessary trust in the repeater nodes is to use multi-path QKD [36, 47], which distributes quantum key material simultaneously over multiple independent paths in a network, increasing resilience against node or link failures and reducing the reliance on any single trusted repeater. By combining the partial keys received via different paths–typically using information-theoretic techniques such as secret sharing [31, 46]–the communicating parties can reconstruct a final key with improved security guarantees. Specifically, when using k-out-of-t secret sharing for some $k < t$, one additionally obtains robustness guarantees: while the transferred data remains secure as long as at most $k - 1$ paths contain a corrupted node, a joint key can be successfully established as long as at no more than $t - k$ paths refuse to relay data.

However, although this approach mitigates the risk of a single compromised repeater, it has a fundamental limitation: it requires the QKD network operator to be trusted to faithfully transmit the partial keys over pairwise disjoint paths. This is because the security guarantees rely critically on the assumption that no individual node has access to more than one partial key. Given that in the long term it can be expected that QKD networks will be operated by private entities such as telecommunications providers, and the stringent security requirements in scenarios like inter- and intra-governmental communication, relying solely on contractual instruments such as service level agreements (SLAs) may pose unacceptable risks.

Therefore, technical and auditable measures are necessary to ensure that contractual agreements have been satisfied, e.g., verifying path disjointness, the certification level of deployed repeater nodes, or the host countries through which the communication path has been routed, without revealing sensitive business information about the topology of the underlying QKD network.

1.1 Our Contribution

In this paper, we propose a construction for proving that contractual agreements have been satisfied in a transmission over a QKD network. In particular, our construction is powerful enough to model the aforementioned use-cases. That is, it can be used to (i) ensure that the secret was transmitted over at least k

pairwise disjoint paths through the QKD network, and (ii) guarantee that only devices satisfying certain attributes (e.g., certification levels, manufacturers, etc.) were used during transmission, where (iii) the devices may have been certified by multiple authorities (differing, e.g., per country), (iv) without leaking sensitive information about the network topology to the user of the network.

More specifically, we present a modular extension applicable to arbitrary QKD networks, introducing only a small computational and communication overhead per repeater node and also at the receiver's end. Assuming that devices are certified before being deployed in practice–a necessary yet realistic assumption in high-security contexts which are subject to strict security controls–our construction can be used to audit that a QKD transmission has only passed through repeater nodes satisfying a previously defined policy; for concreteness, one might think of policies defining minimum certification levels for highly sensitive transactions, or trusted manufacturers or countries of origin of repeaters. At the same time, by using appropriate privacy-enhancing technologies (PETs), the receiving party only learns information about the number of repeaters on the path, which can anyhow be estimated based on the maximum distance of practical QKD networks. However, no further information about the topology is leaked. In particular, it remains hidden whether a specific node participated in different communication sessions, or to which other nodes it might be connected.

To show the soundness of our approach, we provide a formal definitional framework capturing the intended security guarantees, and provide rigorous security proofs. Furthermore, we underpin the practicability of our solution by a detailed efficiency evaluation and benchmarks.

1.2 Technical Overview

In the following, we briefly sketch the core technical ideas underlying our construction, omitting specific details which will be discussed in the full construction. While being conceptually easy to grasp, we want to stress that the benefit and added value of our contribution can significantly contribute to minimizing trust requirements in providers of QKD networks.

Before deployment, each repeater node generates a local key pair $(\mathsf{sk}_{\mathtt{N}}, \mathsf{pk}_{\mathtt{N}})$, and receives a certificate cred in the form of a digital signature on its attributes and its public key $\mathsf{pk}_{\mathtt{N}}$.

When sending a message over a single path, the sender defines a policy ϕ and chooses a unique session id $\mathtt{sid}$, which it transmits to the first node in the path. The node computes a non-interactive zero-knowledge proof of knowledge (NIZK) π_1 that it owns a credential cred satisfying the policy ϕ, and binds it to the session id $\mathtt{sid}$. It then forwards $(\pi_1, \mathtt{sid}, \phi)$ to the next node, which proceeds in a similar manner and sends $((\pi_1, \pi_2), \mathtt{sid}, \phi)$ to the next node. Eventually, the receiver verifies the correctness of all received NIZKs, thereby receiving guarantees that all nodes on the path satisfied the policy ϕ.

Now, in a multi-path setting, each node on each path additionally computes a pseudonym nym for $\mathtt{sid}$, using a scope-exclusive pseudonym scheme. It then

extends the NIZK to show that nym was derived from the secret key $\mathsf{sk}_\mathtt{N}$ underlying $\mathsf{pk}_\mathtt{N}$, which in turn was included in the credential cred, without disclosing $\mathsf{pk}_\mathtt{N}$ to the verifier. At the end of the transmission, the receiver now checks that all received pseudonyms are pairwise distinct, thus receiving guarantees that no node was involved in more than one path.

When aiming for trans-national QKD networks, nodes may be certified by different authorities (e.g., per country). As the verification of the NIZK requires access to the corresponding authority's public key $\mathsf{pk}_\mathtt{I}$, the receiver would learn information about the path. This can be overcome by leveraging the idea of issuer-hiding attribute-based credentials [8], which allow one to check that only accepted authorities issued the certificates while hiding the precise issuer.

1.3 Related Work

While QKD itself is a highly active research area and a significant body of work aims at overcoming trust assumptions using, e.g., multi-path communication [36, 47], or at outsourcing computationally expensive parts of the post-processing [37], only limited work has focused on cryptographically auditing the behavior of network providers and nodes. As an example, Franzoi et al. [19] recently presented a protocol allowing one to identify the inconsistent link in cases where the integrity of the transferred secret was broken, i.e., that Alice and Bob did not obtain identical secret due to misbehavior of some repeater node. Concurrent to our work, Cozzolino et al. [17] presented a protocol for showing the availability of QKD-links between two nodes in an inter-network scenario based on graph signatures, keeping the topology of the underlying networks secret. This is complementary to our effort in the sense that they prove availability of a (multi-path) connection, while we provide evidence that a used path satisfied certain constraints.

Regarding mechanisms for *path validation and path verification* for a next-generation Internet—e.g., to ensure that packages are routed via a superior (e.g., faster) network path as defined in the SLAs—have been researched, e.g, in [29, 35, 42, 45]. However, most importantly, these works all assume that the sender is aware of the entire path through the network, which is not applicable in our setting. Also, multi-path communication or complex policies ϕ are not considered in these works, while the former could be easily achieved by letting the sender define disjoint transmission paths. At a more detailed level, there are trade-offs compared to our work in terms of privacy: for instance, [45] does not reveal the index (i.e., position on the path) of a node to that node, but discloses the length of the entire path to all nodes. In contrast, while revealing the index, we hide the path length to all but the last node in the communication. Notably, [35] achieves a constant size proof, yet also relying on the knowledge of the entire path. While all the aforementioned work focuses on a single transmission path, Sengupta [44] lets senders define multiple valid paths, and the forwarding logic ensures that one of those paths was indeed followed. Note that this is different from our notion of multi-paths, which ensures that different parts of the data are sent over multiple, disjoint paths through the network. Summing up, while constituting a

large body of work, existing path validation mechanisms are insufficient for our application, mainly because they assume that the sender is aware of the paths in the communication network, which is considered sensitive information in our scenario.

Complementary to this, *topology-hiding computation and communication* were introduced as privacy notions for running distributed protocols over an incomplete network while revealing essentially nothing about the underlying communication graph. Moran, Orlov, and Richelson initiate topology-hiding computation and establish feasibility and limitations in early settings [41]. Their work was extended to a general tool for higher-level protocols [27], and extended to topology-hiding computation over all graph topologies [2]. Later works refine the model toward stronger adversaries and more realistic timing assumptions, by pushing boundaries beyond semi-honest security [33] and to networks with unknown delays [34]. Complementarily, Ball et al. investigate topology-hiding communication from the viewpoint of minimal setup and assumptions required to realize it [4]. While all these works are closely related to our ambition of hiding the network structure while still carrying out nontrivial distributed tasks correctly (and in some cases even robustly), they pursue a different goal than ours: they aim to realize generic topology-hiding communication or computation among protocol participants, and their guarantees are tied to the correctness and security of that interactive execution. They do not provide a post-hoc, transferable audit guarantee for the receiver, namely an externally verifiable proof that the specific realized route satisfied policy constraints such as certified node properties and disjointness, which is what our approach is designed to deliver.

In summary, our work can be viewed as bridging topology-hiding protocols with path validation/verification: we adopt the topology-hiding objective to protect the operator's internal knowledge while importing the path-validation goal to provide externally verifiable evidence on the realized route's properties.

1.4 Outline

This document is structured as follows. In Section 2 we introduce the basic notation used in this paper, and recap the cryptographic building blocks used later on. Then, in Section 3, we formalize the syntax and security requirements for an auditable QKD protocol. We then introduce a generic construction achieving these properties in Section 4, where we also provide rigorous security proofs. A specific instantiation is provided in Section 5, together with an efficiency analysis. Finally, we briefly conclude in Section 6.

2 Preliminaries

Throughout the paper, we will denote the main security parameter by λ. We write $s \leftarrow_{\$} S$ to denote that s was sampled uniformly at random from a set S. Similarly, $s \leftarrow_{\$} \mathsf{A}(x)$ denotes that s is the output of a potentially randomized algorithm A on input x. For an interactive protocol between two parties A and

B we write $(out_{\mathsf{A}}; out_{\mathsf{B}}) \leftarrow_{\$} \langle \mathsf{A}(in_{\mathsf{A}}), \mathsf{B}(in_{\mathsf{B}}) \rangle (in)$ to denote that A and B receive the corresponding outputs out_{A} and out_{B} upon executing the protocol on private inputs in_{A} and in_{B}, as well as common input in. For an integer n, the set $\{1, \ldots, n\}$ is denoted by $[n]$. A function negl is called negligible, if it vanishes faster than any inverse polynomial, i.e., for every integer j there exists an integer n_j such that $\mathsf{negl}(n) < n^{-j}$ for all $n > n_j$. We use $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, G, \hat{G})$ to denote a bilinear group for asymmetric type-3 bilinear groups, where p is a prime of bit length λ.

2.1 Structure-Preserving Signatures

Digital signature schemes can be used to ensure the integrity, authenticity, and non-repudiation of digital messages. They consist of a quadruple of algorithms $(\mathsf{ParGen}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, where $\mathsf{ParGen}(1^\lambda)$ outputs system parameters pp, $\mathsf{KeyGen}(pp)$ outputs a key pair consisting of a secret signing key sk as well as a public verification key pk, $\mathsf{Sign}(pp, \mathsf{sk}, \mathsf{msg})$ uses sk to derive signatures on messages msg, and $\mathsf{Verify}(pp, \mathsf{pk}, \sigma, \mathsf{msg})$, checks the validity of a signature on a message relative to pk.

Digital signature schemes must satisfy EUF-CMA (existential unforgeability under chosen-message attacks), first formalized by Goldwasser et al. [22]. This property requires that no PPT adversary, given only pk, can compute a valid signature on a new message, even after having obtained valid signatures on arbitrarily many signatures of its own choice from a signing oracle.

Structure-preserving signatures, first introduced by Abe et al. [1], are digital signatures where verification keys, messages, and signatures are elements of a bilinear group, and verification is done by checking a conjunction of pairing-product equations.

2.2 Pseudonym Systems

Pseudonym systems were first introduced by Chaum [15], and later formalized in a series of work, e.g., [12,38]. Such systems let users interact or authenticate using persistent aliases instead of their real identities, enhancing privacy and security. Specifically, in the case of scope-exclusive pseudonyms, pseudonyms are stable only within a given scope, while they cannot be linked across different scopes, even if derived from the same secret key.

Pseudonym systems consist of three algorithms $(\mathsf{ParGen}, \mathsf{KeyGen}, \mathsf{NymGen})$, where $\mathsf{ParGen}(1^\lambda)$ generates public parameters pp, $\mathsf{KeyGen}(pp)$ generates a user's secret key sk, and the deterministic algorithm $\mathsf{NymGen}(pp, \mathsf{sk}, \mathsf{scope})$ computes a pseudonym nym for a given sk and a given scope.

Pseudonym systems need to be collision resistant, meaning that for each scope, any two users will have different pseudonyms. Furthermore, they have to be unlinkable, intuitively meaning that no PPT adversary can decide to which user a specific nym belongs, even after having received arbitrarily many pseudonyms of users on scopes chosen by the adversary.

2.3 Zero-Knowledge Proofs of Knowledge

A zero-knowledge proof of knowledge (ZKP) is a protocol where a prover convinces a verifier that they know a piece of information without revealing the information about the secret beyond what is already revealed by the claim itself. Unlike a basic zero-knowledge proof, which only demonstrates the existence of a fact, a ZKPoK proves that the prover possesses the knowledge of a secret. A ZKP is called non-interactive (NIZK), if the protocol consists of a single message sent from the prover to the verifier.

A NIZK consists of a triple of algorithms $(\mathsf{ParGen}, \mathsf{NIZK}, \mathsf{Verify})$, where ParGen generates the necessary system parameters, NIZK generates a non-interactive zero-knowledge proof of knowledge that the prover knows a secret witness w such that $(x, w) \in \mathcal{R}$ for a binary relation $\mathcal{R}$ and a public value x, and Verify indicates whether to accept or to reject the proof.

A NIZK has to be complete, meaning that an honest prover knowing a valid witness can always convince the verifier. Furthermore, zero-knowledge means that not knowing w but knowing a simulation trapdoor, one can generate transcripts which are indistinguishable from real protocol transcripts. Finally, simulation sound extractability means that from any prover who can make the verifier accept, a valid witness can be extracted, even if the prover has seen simulated proofs for potentially false statements. For formal definitions, we refer, e.g., to Goldwasser et al. [21, 23].

Slightly overloading notation, we will use Camenisch-Stadler notation [13] to denote proof goals. We write:

$$\pi \leftarrow_{\$} \mathsf{NIZK}\left[(\alpha, \beta, \gamma) : Y = G^{\alpha} \cdot H^{\beta} \ \wedge \ Z = G^{\alpha} \cdot H^{\gamma} \ \wedge \ \gamma = \alpha \cdot \beta\right](\mathsf{ctx})$$

to prove knowledge of values α, β, γ such that the relation on the right hand side are satisfied; in particular, all values not in parentheses are assumed to be public. Furthermore, we often bind the proof to a context ctx in the sense of a signature of knowledge, for formal definition see, e.g., [5, 14].

3 Auditable QKD Framework

In this section, we present a novel, privacy-preserving, and auditable protocol framework tailored to the requirements of large-scale QKD networks. Here we first formalize the notation and define the required security properties for auditable QKD protocol extensions.

3.1 Syntax

Definition 1 (Auditable QKD). *An auditing extension for a QKD protocol consists of the following algorithms:*

$pp \leftarrow_{\$} \mathsf{ParGen}(1^{\lambda})$**.** On input the security parameter λ in unary representation, output the public parameters pp. These parameters are implicit input to all further algorithms, and might not be made explicit to ease presentation.

$(\mathsf{sk}_\mathtt{I}, \mathsf{pk}_\mathtt{I}) \leftarrow_\$ \mathsf{KeyGen}_\mathtt{I}(pp)$**.** This algorithm generates a secret key $\mathsf{sk}_\mathtt{I}$ as well as a public key $\mathsf{pk}_\mathtt{I}$ for an issuer.

$(\mathsf{sk}_\mathtt{N}, \mathsf{pk}_\mathtt{N}) \leftarrow_\$ \mathsf{KeyGen}_\mathtt{N}(pp)$**.** This algorithm generates a secret key $\mathsf{sk}_\mathtt{N}$ as well as a public key $\mathsf{pk}_\mathtt{N}$ for a repeater node.

$(\mathsf{cred}; \perp) \leftarrow_\$ \langle \mathsf{Register}_\mathtt{I}(\mathsf{sk}_\mathtt{I}, \mathsf{pk}_\mathtt{N}), \mathsf{Register}_\mathtt{N}(\mathsf{sk}_\mathtt{N}, \mathsf{pk}_\mathtt{I}) \rangle (\boldsymbol{a})$**.** In this interactive protocol between an issuer and a repeater node, each party takes as input its own secret key and the other entity's public key, as well as previously agreed attributes $\boldsymbol{a}$ of a node. At the end of the protocol, the node receives a certificate cred certifying its public key $\mathsf{pk}_\mathtt{N}$ and the attributes $\boldsymbol{a}$.

$(\perp; \{\perp\}; n') \leftarrow_\$ \langle \mathsf{Send}_\mathsf{A}(), \{(\mathsf{Send}_\mathtt{N}(\mathsf{sk}_{\mathtt{N}_{ij}}, \mathsf{cred}_{\mathtt{N}_{ij}}, \boldsymbol{a}_{\mathtt{N}_{ij}})_{j=1}^{m_i})\}_{i=1}^{n}, \mathsf{Send}_\mathsf{B}() \rangle (\mathsf{pk}_\mathtt{I}, \phi)$**.** This is an interactive protocol between a sender A, a receiver B, an a set of nodes along n paths between A and B, each containing m_i nodes. The parties take as inputs the issuer's key and the policy, and their respective secret inputs. At the end of the interaction, B receives as output an integer n' indicating the number of disjoint paths used for the communication, or an error symbol to reject the transmission, while all the other parties do not receive any output.

3.2 Security Overview and Adversarial Capacity

In the following we briefly explain the considered adversary model and the intuition underlying our security framework presented in Section 3.3.

As discussed earlier, our ambition is to balance the needs of giving strong guarantees on the selected path to the customer, while at the same time protecting the confidentiality of the operator.

On the one hand, we define *policy compliance*. This property requires that a malicious network operator cannot forge proofs that a chosen path satisfied certain requirements. That is, the operator's aim is to convince the communication partners A and B that a QKD-session was carried out over (one or more) communication links satisfying the agreed policy ϕ (specifying, e.g., a minimum certification level), while at least one of the involved repeaters did not fulfill the requirements. In terms of the adversary's capabilities, we consider a malicious network operator controlling the communication network. The operator has full control over the path used for a key exchange, and may install or remove arbitrary repeaters from the network. However, we assume that key material of repeaters certified by an authority cannot be manipulated by the network operator – a seemingly strong yet realistic assumption in the QKD context, which can be achieved, e.g., using secure hardware elements. Furthermore, as is the case in practice, each repeater is assumed to know its physical neighbors in the network.

On the other hand, we introduce the notion of *path-hiding*, which ensures that the operator's network topology remains hidden from the customer. This is important as a provider's exact QKD network topology may reveal information about the location of trusted nodes, key-management infrastructure, and physical links, which exposes operational capabilities, redundancy, and potential single points of failure. This information could be exploited by competitors for

strategic mapping and capacity inference, or by attackers to target the most valuable nodes and routes for disruption or surveillance. We thus consider an overly strong adversary, who has full information about the network topology, and who is allowed to request an arbitrary (polynomial) number of communication sessions. Despite this information, the adversary – controlling the communicating parties A and B – should not be able to learn anything about the selected path for a given communication session, except for the selected "entry" and "exit" nodes, which are necessarily revealed due to physical connectivity properties in QKD networks. This will ensure that in particular no relevant information about a communication path is revealed to a less powerful adversary having less information about the overall network topology.

3.3 Formal Security Definition

Besides completeness, and as discussed in the previous section, we require two main properties from auditable QKD protocols, which we will formally define in the following.

Path-Hiding. This property ensures that the topology of the underlying QKD network is not revealed to the sender or receiver of a communication session. This is modeled in an indistinguishability game in Fig.1, where we assume that the adversary knows the precise graph of the QKD network.

Let $\mathbf{G}$ be a description of the network graph, containing all nodes and edges. Now, in a first step, we generate all keys honestly, but let the adversary control the attributes of each node, which get certified honestly by the authority. We then give the adversary (controlling potentially multiple senders and receivers) access to two oracles:

- First, $\mathcal{O}$ allows the adversary to initiate arbitrary (multi-path) transmissions in the network using self-chosen policies over self-chosen paths.
- Second, the challenge oracle $\mathcal{O}_{LR}$, which may be called only once—enables the adversary to select two sets of paths. The oracle then initiates a transmission session using one of the two.

At the end of the experiment, the adversary needs to guess which paths were used for the transmission. In the oracles, we exclude trivial distinguishing features, like repeating session ids, paths of non-matching lengths, paths with loops, or inconsistent entry/exit nodes (note that A and B trivially know to/from whom they send/receive messages), or a policy that is not satisfied by (some of) the nodes on one of the paths.

A scheme is now said to be path hiding, if no adversary has more than negligible advantage in winning the described experiment compared to random guessing. We formally define this game as follows:

Definition 2 (Path hiding). *An auditable QKD protocol is* path hiding*, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* negl *such that for*

every network $\mathbf{G}$*:*

$$\left|\Pr\left[Exp^{\mathcal{A}}_{path\text{-}hiding}(\lambda)=1\right]-\frac{1}{2}\right| \leq \mathsf{negl}(\lambda),$$

where the experiment is as defined in Fig. 1.

Experiment $\mathrm{Exp}^{\mathcal{A}}_{\text{path-hiding}}(\lambda)$

$b \leftarrow_{\$} \{0,1\}$
$\mathcal{S} \leftarrow \emptyset$
$pp \leftarrow_{\$} \mathsf{ParGen}(1^{\lambda})$
$(\mathsf{sk}_{\mathtt{N}}, \mathsf{pk}_{\mathtt{N}}) \leftarrow_{\$} \mathsf{KeyGen}_{\mathtt{N}}(pp)$ for all $\mathtt{N} \in \mathbf{G}$
$(\mathsf{sk}_{\mathtt{I}}, \mathsf{pk}_{\mathtt{I}}) \leftarrow_{\$} \mathsf{KeyGen}_{\mathtt{I}}(pp)$
$(\mathsf{st}, (\boldsymbol{a}_{\mathtt{N}})_{\mathtt{N}\in\mathbf{G}}) \leftarrow \mathcal{A}(pp, \mathsf{pk}_{\mathtt{I}}, \{\mathsf{pk}_{\mathtt{N}}\}_{\mathtt{N}\in\mathbf{G}}, \mathbf{G})$
$(\mathsf{cred}; \perp) \leftarrow_{\$} \langle \mathsf{Register}_{\mathtt{I}}(\mathsf{sk}_{\mathtt{I}}, \mathsf{pk}_{\mathtt{N}}), \mathsf{Register}_{\mathtt{N}}(\mathsf{sk}_{\mathtt{N}}, \mathsf{pk}_{\mathtt{I}})\rangle(\boldsymbol{a}_{\mathtt{N}})$ for all $\mathtt{N} \in \mathbf{G}$
$b' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}(\cdot,\cdot,\cdot),\mathcal{O}_{LR}(\cdot,\cdot,\cdot,\cdot)}(\mathsf{st})$
  where $\mathcal{O}(\cdot,\cdot,\cdot)$ on input $(\mathtt{sid}, \phi, \{(\mathtt{N}_{ij})_{j=1}^{m_i}\}_{i=1}^{n})$:
    Aborts if $\mathtt{sid} \in \mathcal{S}$, otherwise sets $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathtt{sid}\}$, and engages in a transmission protocol acting as $\{(\mathtt{N}_{ij})_{j=1}^{m_i}\}_{i=1}^{n}$ for the given $\mathtt{sid}$ and ϕ
  where $\mathcal{O}_{LR}(\cdot,\cdot,\cdot,\cdot)$ on input $(\mathtt{sid}, \phi, \{(\mathtt{N}^0_{ij})_{j=1}^{m_i}\}_{i=1}^{n}, \{(\mathtt{N}^1_{ij})_{j=1}^{m_i}\}_{i=1}^{n})$:
    Aborts if it had been activated before, or if $\mathtt{sid} \in \mathcal{S}$, or if the provided input paths contain loops, are not pairwise disjoint, or do not have corresponding lengths or entry/exit nodes, or any node does not satisfy ϕ. Otherwise, it sets $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathtt{sid}\}$, and engages in a transmission protocol acting as $\{(\mathtt{N}^b_{ij})_{j=1}^{m_i}\}_{i=1}^{n}$ for the given $\mathtt{sid}$ and ϕ
Return $b = b'$

Fig. 1. Path hiding

Note that the above definition considers a single authority certifying all nodes in the network. In case, e.g., of cross-border communication, different authorities might be involved. If hiding the precise authority is important, the definition (as well as the construction presented below) can be adapted in a straightforward manner.

Policy compliance. Besides hiding the topology, our second main goal is to ensure that the nodes used for routing the message comply with the imposed policy and that the number of disjoint paths is correct.

In our security definition, we again generate all keys honestly, and let the adversary choose the attributes of all devices in the network $\mathbf{G}$. Subsequently, the nodes are honestly registered. The adversary then gets full access to the nodes' certificates, allowing for more targeted attacks than by solely giving oracle access to the presentation of certificates. The adversary then has to define a session id

$\mathtt{sid}$, a set of n routes through the network, and a policy ϕ of its choice. Based on these outputs, a transmission from A to B via the specified paths for the given session id and policy is initiated. The adversary now wins if B does not output an error symbol, and at least one of the following conditions is satisfied: (i) At least one node in the transmission did not satisfy the policy ϕ, or (ii) The different paths used for the transmission were not disjoint, or (iii) The number of disjoint paths identified by B differs from the number of actual disjoint paths in the transmission. A scheme is now said to be policy compliant if no adversary has more than negligible advantage in winning the described experiment.

Definition 3 (Policy compliance). *An auditable QKD protocol is* policy compliant*, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* negl *such that for every network* $\mathbf{G}$*:*

$$\Pr\left[Exp^{\mathcal{A}}_{policy\text{-}compliance}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda),$$

where the experiment is as defined in Fig. 2.

Experiment $\mathrm{Exp}^{\mathcal{A}}_{\text{policy-compliance}}(\lambda)$

$pp \leftarrow_{\$} \mathsf{ParGen}(1^{\lambda})$

$(\mathsf{sk}_{\mathtt{N}}, \mathsf{pk}_{\mathtt{N}}) \leftarrow_{\$} \mathsf{KeyGen}_{\mathtt{N}}(pp)$ for all $\mathtt{N} \in \mathbf{G}$

$(\mathsf{sk}_{\mathtt{I}}, \mathsf{pk}_{\mathtt{I}}) \leftarrow_{\$} \mathsf{KeyGen}_{\mathtt{I}}(pp)$

$(\mathsf{st}, (\boldsymbol{a}_{\mathtt{N}})_{\mathtt{N}\in\mathbf{G}}) \leftarrow \mathcal{A}(pp, \mathsf{pk}_{\mathtt{I}}, \{\mathsf{pk}_{\mathtt{N}}\}_{\mathtt{N}\in\mathbf{G}}, \mathbf{G})$

$(\mathsf{cred}; \bot) \leftarrow_{\$} \langle \mathsf{Register}_{\mathtt{I}}(\mathsf{sk}_{\mathtt{I}}, \mathsf{pk}_{\mathtt{N}}), \mathsf{Register}_{\mathtt{N}}(\mathsf{sk}_{\mathtt{N}}, \mathsf{pk}_{\mathtt{I}})\rangle(\boldsymbol{a}_{\mathtt{N}})$ for all $\mathtt{N} \in \mathbf{G}$

$(\mathtt{sid}, \{(\mathtt{N}_{ij})_{j=1}^{m_i}\}_{i=1}^{n}, \phi) \leftarrow_{\$} \mathcal{A}(\mathsf{st}, \{\mathsf{cred}_{\mathtt{N}}\}_{\mathtt{N}\in\mathbf{G}})$

$(\bot; \{\bot\}; n') \leftarrow_{\$} \langle \mathsf{Send}_{\mathsf{A}}(), \{(\mathsf{Send}_{\mathtt{N}}(\mathsf{sk}_{\mathtt{N}_{ij}}, \mathsf{cred}_{\mathtt{N}_{ij}}, \boldsymbol{a}_{\mathtt{N}_{ij}})_{j=1}^{m_i})\}_{i=1}^{n}, \mathsf{Send}_{\mathsf{B}}()\rangle(\mathsf{pk}_{\mathtt{I}}, \phi)$

Return 1 if $n' \neq \bot$ and:

- $\phi(\boldsymbol{a}_{\mathtt{N}_{ij}}) \neq 1$ for some $i \in [n]$ and $j \in [m_i]$, OR //policy violation
- $\{\mathtt{N}_{i_0 j}\}_{j=1}^{m_{i_0}} \cap \{\mathtt{N}_{i_1 j}\}_{j=1}^{m_{i_1}} \neq \emptyset$ for some $i_0 \neq i_1$, OR //non-disjoint paths
- $n \neq n'$ //wrong number of paths

Return 0

Fig. 2. Policy compliance

Again, similar as above, an extension of the definition to multiple issuing authorities is straightforward.

4 An Auditable QKD Protocol

In the following, we present our auditable QKD protocol. We first present a generic construction from well-known cryptographic building blocks, and show that it satisfies the requirements set out in Section 3.3. We then give a concrete instantiation of the generic construction to show its practical usability.

4.1 A Generic Construction

Our generic construction can be built from a signature scheme Σ, a pseudonym scheme Ψ, and a non-interactive zero-knowledge proof system NIZK, cf. Sections 2.1, 2.2 and 2.3.

In a first step, the public parameters are generated by simply generating the parameters needed for the respective building blocks. Then, the issuer (authority) generates a key pair for a signature scheme, and each node samples the secret key for a pseudonym scheme. Furthermore, each node computes its public key as a pseudonym for a distinguished scope `setup`. Now, for registration, the node first proves in zero-knowledge to the authority that it knows the secret key corresponding to its public key. The authority then uses its signing key to generate the certificate simply as a digital signature on the node's public key and the previously agreed attributes (note that validation of attribute values needs to happen out of band). These onboarding steps are formally depicted in Construction 1.

Parameter generation. The public parameters pp consist of the parameters pp_Σ of the signature scheme Σ, pp_Ψ of the pseudonym system Ψ, and pp_{NIZK} of the non-interactive zero-knowledge proof system NIZK.

Key generation. The issuer generates a key pair for a digital signature scheme as $(\mathsf{sk}_\mathtt{I}, \mathsf{pk}_\mathtt{I}) \leftarrow_\$ \Sigma.\mathsf{KeyGen}(1^\lambda)$.
Each node generates a key pair as $\mathsf{sk}_\mathtt{N} \leftarrow_\$ \Psi.\mathsf{KeyGen}(1^\lambda)$ and $\mathsf{pk}_\mathtt{N} \leftarrow \Psi.\mathsf{NymGen}(\mathsf{sk}_\mathtt{N}, \mathtt{setup})$.

Registration. To register a node $\mathtt{N}$ with public key $\mathsf{pk}_\mathtt{N}$ and attributes $\boldsymbol{a}_\mathtt{N}$, the node first proves possession of the corresponding secret key:

$$\pi \leftarrow_\$ \mathsf{NIZK}[(\mathsf{sk}_\mathtt{N}) : \Psi.\mathsf{NymGen}(\mathsf{sk}_\mathtt{N}, \mathtt{setup}) = \mathsf{pk}_\mathtt{N}](\mathtt{did})\,,$$

where `did` is a device registration id that is unique per registration process. The issuer then signs a credential $\mathsf{cred} \leftarrow \Sigma.\mathsf{Sign}((\mathsf{pk}_\mathtt{N}, \boldsymbol{a}_\mathtt{N}), \mathsf{sk}_\mathtt{I})$ for the node.

Construction 1: Generic construction: Initialization and registration.

Now, when A wants to transmit a message to B, they jointly agree on the number n of disjoint paths to be used, and the policy ϕ that has be satisfied by all nodes on the transmission paths. A then selects a unique session id `sid` and sends $(\mathtt{sid}, \phi)$ to the entry node of each path. Note here that re-using a session id might leak information about the network topology, which however can be avoided, e.g., by ensuring freshness through inclusion of a timestamp validated by the entry nodes. Note further that A learns only the entry nodes of each communication path, which is unavoidable due to the physical connectivity in QKD systems; however, as modeled in Definition 2, the remaining routing is oblivious to A and determined by the network operator.

Upon receiving an (initially empty) set of zero-knowledge proofs and pseudonyms (π, nym), as well as `sid` and ϕ, each node now computes its own pseudonym

nym for the given session id. Subsequently, it computes a zero-knowledge proof of knowledge $\pi_{\mathtt{N}}$ that $\mathsf{nym}_{\mathtt{N}}$ was indeed derived from the same secret key underlying the public key certified by the authority. Furthermore, the proof shows that the policy ϕ is satisfied. As an additional safeguard against a service provider trying to introduce non-certified nodes, each node additionally computes a second zero-knowledge proof σ showing that nym and its public key are indeed consistent, yet this time revealing its own public key. The node finally sends all previously received proofs π and pseudonyms nym, its own proof and pseudonyms $(\pi_{\mathtt{N}}, \mathsf{nym}_{\mathtt{N}})$, as well as σ, $\mathtt{sid}$, and ϕ to the next node on the path, who only accepts the inputs if σ is valid. See also Fig. 3 for a visualization of this message flow.

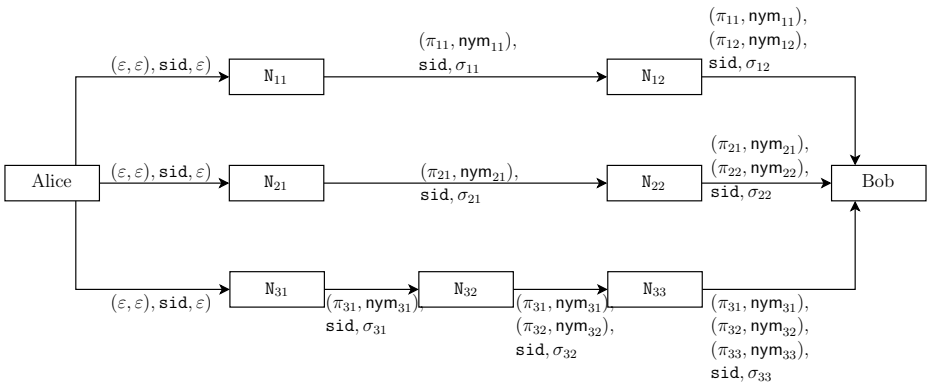

Fig. 3. Example message flow for three disjoint paths.

Eventually, B receives zero-knowledge proofs and pseudonyms from the exit nodes of all n paths. B now checks whether all zero-knowledge proofs are valid, thereby receiving guarantees that all nodes involved in the transmission actually satisfied the policy ϕ and that all received pseudonyms are authentic. Furthermore, B checks that all pseudonyms are pairwise distinct, thereby receiving guarantees that no node was involved twice in the transmission, and thus the paths were pairwise disjoint.

Note that the additional proof σ is not forwarded to B, but always just sent to the subsequent node, who verifies its correctness and aborts otherwise. Given that in a QKD networks each node knows its neighbors (and thus also their public keys), this allows the node to check that the last (π, nym)–sharing the same nym as σ–was indeed computed by the previous node in the path, as the public key underlying nym is revealed in σ. This prevents against service providers deploying non-certified nodes in the network, which would solely forward the received auditing data without appending their own information.

A formal description of the protocol is now provided in Construction 2.

Sending. Alice defines the number n of disjoint paths over which the secret has to be transmitted. Furthermore, Alice agrees with Bob on a policy ϕ to be satisfied by all nodes. She then selects a unique session id $\mathtt{sid}$, and sends $((\varepsilon,\varepsilon),\mathtt{sid},\phi,\varepsilon)$ to the entry node of each path previously received from the service provider, i.e., to $\mathtt{N}_{i1}$ for all i.

Forwarding. A node $\mathtt{N}_{ij}$, upon receiving $(\mathsf{msg} = (((\pi_{i1},\mathsf{nym}_{i1}),\ldots,(\pi_{i,j-1},\mathsf{nym}_{i,j-1})),\mathtt{sid}),\sigma)$, behaves as follows:

- If $j > 1$, it fetches the public key $\mathsf{pk}_{i,j-1}$ of the sending node, and verifies the validity of the received σ. The node then computes $\mathsf{nym}_{ij} \leftarrow_{\$} \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij},\mathtt{sid})$ as well as the following two zero-knowledge proofs:

$$\begin{aligned}
\pi_{ij} \leftarrow \mathsf{NIZK}[(\mathsf{sk}_{ij},\mathsf{pk}_{ij},\boldsymbol{a}_{ij},\mathsf{cred}_{ij}) :\ & \\
& \Sigma.\mathsf{Verify}((\mathsf{pk}_{ij},\boldsymbol{a}_{ij}),\mathsf{cred}_{ij},\mathsf{pk}_{\mathtt{I}}) = 1 \wedge && \text{//the credential is valid}\\
& \mathsf{nym}_{ij} = \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij},\mathtt{sid}) \wedge && \text{//}\mathsf{nym}_{ij}\text{ is well-formed and ...}\\
& \mathsf{pk}_{ij} = \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij},\mathtt{setup}) \wedge && \text{//... belongs to the right }\mathsf{sk}_{ij}\\
& \phi(\boldsymbol{a}) = 1](\mathsf{msg}) && \text{//the policy is satisfied}
\end{aligned}$$

$$\begin{aligned}
\sigma_{ij} \leftarrow \mathsf{NIZK}[(\mathsf{sk}_{ij}) : \mathsf{nym}_{ij} = \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij},\mathtt{sid}) \wedge & \\
\mathsf{pk}_{ij} = \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij},\mathtt{setup})](\mathsf{msg})\,. &
\end{aligned}$$

- The node then transfers $((\pi,(\pi_{ij},\mathsf{nym}_{ij})),\mathtt{sid},\phi,\sigma_{ij})$ to the next node of each path, i.e., to $\mathtt{N}_{ij+1}$ or Bob in the case that $j = m_i$.

Receiving. Having received messages from $\mathtt{N}_{im_i}$ for all $i \in [n]$, Bob verifies all proofs and signatures.
It furthermore validates that all received nym_{ij} are pairwise distinct. If all checks validate correctly, Bob outputs n, otherwise it aborts with $\perp$.

Construction 2: Generic construction: Message routing.

4.2 Security Analysis

We now formulate the main results of our work, and provide formal proofs that our generic construction indeed satisfies the properties defined in Section 3.3.

Theorem 1. *The auditable QKD protocol presented in Constructions 1 and 2 is path-hiding in the sense of Definition 2, if the deployed pseudonym system Ψ is unlinkable and the proof system* NIZK *is simulation sound extractable.*

Proof. The proof follows standard techniques for privacy-enhancing protocols, and can be seen using the following series of games which are all computationally indistinguishable.

Game 0: This is the original experiment as in Definition 2.

Game 1: We first modify the setup of the proof system NIZK by simulating the necessary parameters.

This is indistinguishable by definition of simulation-sound extractability.

Game 2: We now modify $\mathcal{O}_{LR}$ such that all zero-knowledge proofs π_{ij} and σ_{ij} generated by the different nodes $\mathtt{N}^b_{ij}$ for $i \in [n]$ and $j \in [m_i]$.

By the zero-knowledge property of NIZK this is indistinguishable from the adversary's point of view. Note here that the adversary (controlling A and B) only gets to see the π_{ij}, as well as the last σ_{ij} per path, i.e., for $j = m_i$. Note further that in particular, after this game, all proofs are independent of the nodes' certificates cred_{ij}, attributes $\boldsymbol{a}_{ij}$, and the secret keys sk_{ij}.

Game 3: We now change the computation of $\mathsf{nym}_{ij} = \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij}, \mathtt{sid})$ in $\mathcal{O}_{LR}$ to $\mathsf{nym}_{ij} = \Psi.\mathsf{NymGen}(r_{ij}, \mathtt{sid})$ for fresh and independently sampled $r_{ij} \leftarrow_{\$} \Psi.\mathsf{KeyGen}(1^\lambda)$.

Here, it is important to first note that `sid` is fresh, i.e., has not been used in any other transmission in the experiment, i.e., triggered through $\mathcal{O}$. The indistinguishability of this change then follows directly from the unlinkability of the pseudonym system, by which pseudonyms of the same and different users cannot be kept apart.

The adversary's view in the last game is now fully independent of the choice of b, and thus the claim follows. □

Theorem 2. *The auditable QKD protocol presented in Constructions 1 and 2 is policy-compliant in the sense of Definition 3, if the deployed signature scheme Σ is EUF-CMA secure, the pseudonym system Ψ is collision resistant, and the proof system* NIZK *is simulation sound extractable.*

Proof. We prove the statement using a series of games.

Game 0: This is the original experiment as in Definition 3.

Game 1: In this game, we modify the setup of the proof system NIZK to simulate the necessary parameters.

This change is indistinguishable by definition of simulation-sound extractability.

Game 2: In this game, we use the trapdoor from the previous game to extract the witnesses of all proofs π_{ij} and σ_{ij} received by B. That is, we obtain values cred_{ij}, $\boldsymbol{a}_{ij}, \mathsf{sk}_{ij}, \mathsf{pk}_{ij}$ satisfying the relations stated in π_{ij} and σ_{ij} as described in Construction 2, respectively. If the extraction of valid witnesses fails for either of the proofs, the experiment aborts and the adversary wins.

By the simulation sound extractability of the proof system, the difference between these two games is negligible.

Game 3: In this game, we additionally check that $\boldsymbol{a}_{ij}$ and pk_{ij} have indeed been signed by the issuing authority as part of valid registration processes. If this is not the case, we abort.

By the EUF-CMA security of the signature scheme, an abort only happens with negligible probability.

Game 4: In this game, we abort if two different nodes generated identical pseudonyms for the given `sid`.

By the collision resistance of the pseudonym system, this can only happen with negligible probability.

Overall, we are now at a point where we extracted from each node $\mathtt{N}_{ij}$ a set of attributes $\boldsymbol{a}_{ij}$ satisfying the required policy ϕ, and previously certified by the issuing authority together with a public key pk_{ij} which was also used to compute nym_{ij}. Furthermore, the pseudonyms were all honestly generated and pairwise different, proving the disjointness of the different paths. Finally, the number n' of paths obtained by B corresponds to the claimed number n by inspection of the construction (note here that in the definition we assume that the protocol is executed by honest nodes).

The claim of the statement now follows immediately. □

The proof strategy shows that the security proof remains valid even if the adversary controls all secret key material of all nodes before the protocol starts. In particular, the key pair $(\mathsf{sk}_{\mathtt{N}}, \mathsf{pk}_{\mathtt{N}})$ need not be honestly generated. However, this stronger model does not reflect realistic scenarios, as nodes are typically subject to thorough auditing. We therefore chose not to adopt this stronger model in Definition 3, to avoid ruling out constructions that align with real-world requirements.

4.3 Multiple issuers

In realistic deployments, it might happen that QKD links involving multiple network operators have to be established, e.g., when communicating across country borders. In this case, as already briefly mentioned in Section 3.3, a single issuing authority might not be realistic anymore, but rather one issuer, e.g., per EU member state might exist. Using the framework and construction presented so far, it would now be leaked to the receiver who certified the different repeaters, as the issuer's public key $\mathsf{pk}_{\mathtt{I}}$ is used when verifying the zero-knowledge proofs, thereby disclosing some minimum information about the path being used.

We deliberately excluded this scenario from our definitional framework and main construction to maintain clarity and comprehensibility. Yet, this limitation can be easily addressed if needed, using the concept of issuer-hiding attribute-based credentials, first introduced by Bobolz et al. [8], and later picked up also by, e.g., [9, 16, 40]. Such systems allow one to hide the precise issuer of a credential, while still guaranteeing that it was among a set of accepted issuers.

Following the approach of [8], this could be achieved by modifying the computation of π_{ij} as follows. Instead of proving knowledge of $\mathsf{pk}_{ij}, \boldsymbol{a}_{ij}, \mathsf{cred}_{ij}$ such that

$$\Sigma.\mathsf{Verify}((\mathsf{pk}_{ij}, \boldsymbol{a}_{ij}), \mathsf{cred}_{ij}, \mathsf{pk}_{\mathtt{I}}) = 1\,,$$

one would prove knowledge of additional values $\mathsf{pk}_{\mathtt{I}}$ and $\tau_{\mathsf{pk}_{\mathtt{I}}}$ such that

$$\Sigma.\mathsf{Verify}((\mathsf{pk}_{ij}, \boldsymbol{a}_{ij}), \mathsf{cred}_{ij}, \mathsf{pk}_{\mathtt{I}}) = 1 \;\wedge\; \Sigma'.\mathsf{Verify}(\mathsf{pk}_{\mathtt{I}}, \tau_{\mathsf{pk}_{\mathtt{I}}}, \mathsf{pk}') = 1\,.$$

Here, Σ' is an appropriate signature scheme, $\tau_{\mathsf{pk}_{\mathtt{I}}}$ is a signature on the public key of the issuer under some pk' trusted by A and B. Depending on the scenario, pk' can be a fresh key for which all acceptable issuers (e.g., those of EU27 member states) are signed for a single transmission. Alternatively, the signatures

$\tau_{\mathsf{pk}_\mathsf{I}}$ could also be computed and published by a trusted entity, and used across multiple sessions. For further details on this approach, we refer to the original work on issuer-hiding credentials.

5 Concrete Instantiation

In the following we now present a specific instantiation of our generic construction, in order to prove the practicability of our approach.

5.1 Building Blocks

We first detail the instantiations of the building blocks introduced in Sections 2.1, 2.2 and 2.3.

Structure-Preserving Signatures. In the following, we present a version of Groth's scheme [24] tailored to our needs, as we only need to sign a single group element. Similar to previous work [8, 11] we thereby consider a version Groth_1 signing elements of $\mathbb{G}_1$. More specifically, as our construction in the following only needs to sign a single element in $\mathbb{G}_1$, we also restrict the presentation in the following to such a case.

- $\mathsf{Groth}_1.\mathsf{ParGen}(1^\lambda)$ generates public parameters of a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, G, \hat{G})$ of prime order p, where $G \in \mathbb{G}_1$ and $\hat{G} \in \mathbb{G}_2$ are generators. It additionally outputs an element $Y \leftarrow_\$ \mathbb{G}_1$.
- $\mathsf{Groth}_1.\mathsf{KeyGen}(pp)$ samples $\mathsf{sk} \leftarrow \mathbb{Z}_p^*$ and sets $\mathsf{pk} \leftarrow \hat{G}^{\mathsf{sk}}$.
- $\mathsf{Groth}_1.\mathsf{Sign}(pp, \mathsf{sk}, \mathsf{msg})$ samples $r \leftarrow \mathbb{Z}_p^*$ and computes a signature $\sigma = (\hat{R}, S, T) = (\hat{G}^r, (Y \cdot G^{\mathsf{sk}})^{1/r}, (Y^{\mathsf{sk}} \cdot \mathsf{msg})^{1/r})$.
- $\mathsf{Groth}_1.\mathsf{Rand}(pp, \sigma)$ rerandomizes a valid signature σ by sampling $r' \leftarrow \mathbb{Z}_p^*$ and outputting a randomized signature $\sigma' = (\hat{R}', S', T') = (\hat{R}^{r'}, S^{1/r'}, T^{1/r'})$ on the same message.
- $\mathsf{Groth}_1.\mathsf{Verify}(pp, \mathsf{pk}, \sigma, \mathsf{msg})$ outputs 1 if and only if it holds that $e(S, \hat{R}) = e(Y, \hat{G}) \cdot e(G, \mathsf{pk})$ and $e(T, \hat{R}) = e(Y, \mathsf{pk}) \cdot e(\mathsf{msg}, \hat{G})$.

As shown by Groth [24], the scheme satisfies EUF-CMA in the generic group model. Furthermore, it is easy to see that outputs of $\mathsf{Groth}_1.\mathsf{Rand}$ follow the same distribution as fresh signatures on the same message.

In the following we will not make pp explicit as input to the algorithms. Furthermore, a dual scheme signing elements in $\mathbb{G}_2$ can easily be obtained by switching the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$.

Pseudonyms. The following description follows that of Camenisch et al. [12].

- $\mathsf{Nym}.\mathsf{ParGen}(1^\lambda)$ outputs a parameters pp specifying a group $\mathbb{G}$ of prime order p together with a generator G of $\mathbb{G}$. Furthermore, $\mathsf{H} : \{0,1\}^* \to \mathbb{G}$ specifies a cryptographic hash function.
- $\mathsf{Nym}.\mathsf{KeyGen}(pp)$ samples a user secret key as $\mathsf{sk} \leftarrow_\$ \mathbb{Z}_p^*$.

– $\mathsf{Nym.NymGen}(pp, \mathsf{sk}, \mathsf{scope})$ computes a scope-exclusive pseudonym as $\mathsf{nym} \leftarrow \mathsf{H}(\mathsf{scope})^{\mathsf{sk}}$.

Under the DDH assumption and in the random oracle model, the scheme can be shown to be unlinkable and collision resistant.

In the following we will not make *pp* explicit as input to the algorithms.

Zero-Knowledge Proofs. Depending on the specific proof goal, different instantiations can be used. All our proof goals correspond to proving the secret preimage of a homomorphism, such that the Schnorr protocol [43] can be used. The protocol can be made turned into a non-interactive, simulation sound extractable variant using the Fiat-Shamir heuristic [7, 18].

Alternatively, also Groth-Sahai proofs [25], which also satisfy simulation-sound extractability, could be deployed for all proof goals used in our construction.

5.2 Instantiating the Generic Construction

Now, putting all together, we obtain the construction presented in Constructions 3 and 4. Note that in the forwarding phase in Construction 4, the efficiency of the zero-knowledge proof was slightly increased by not proving explicitly that $\mathsf{pk} = \Psi.\mathsf{NymGen}(\mathsf{sk}_{ij}, \mathtt{setup})$. Rather, this is proven implicitly by not proving that pk_{ij} is contained in the credential cred but that the node knows the discrete logarithm of the signed message, i.e., that $\mathsf{H}(\mathtt{setup})^{\mathsf{sk}_{ij}}$ is signed. Importantly, this syntactical change does not modify the proof goal, such that the security proofs of the generic construction still holds for the instantiation.

Parameter generation. Generates public parameters of a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, G, \hat{G})$ of prime order p, where $G \in \mathbb{G}_1$ and $\hat{G} \in \mathbb{G}_2$ are generators. Sample $Y \leftarrow_\$ \mathbb{G}_1$. Define a cryptographically secure hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{G}_1$. Finally, sample $H_i \leftarrow_\$ \mathbb{G}_1$ for $i \in [\ell]$, where ℓ is the maximum number of supported attributes.
Output $pp \leftarrow (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, G, \hat{G}, Y, \mathsf{H}, (H_i)_{i=1}^{\ell})$.

Key generation. The issuer computes a Groth key pair as $\mathsf{sk}_\mathtt{I} \leftarrow_\$ \mathbb{Z}_p^*$ and $\mathsf{pk}_\mathtt{I} = \hat{G}^{\mathsf{sk}_\mathtt{I}}$.
Each node samples $\mathsf{sk}_\mathtt{N} \leftarrow_\$ \mathbb{Z}_p^*$ and sets $\mathsf{pk}_\mathtt{N} = \mathsf{H}(\mathtt{setup})^{\mathsf{sk}_\mathtt{N}}$.

Registration. To register, a node $\mathtt{N}$ with attributes $\boldsymbol{a}_\mathtt{N}$ computes:

$$\pi \leftarrow_\$ \mathsf{NIZK}[(\mathsf{sk}_\mathtt{N}) : \mathsf{pk}_\mathtt{N} = \mathsf{H}(\mathtt{setup})^{\mathsf{sk}_\mathtt{N}}](\mathtt{did}),$$

where $\mathtt{did}$ is a nonce chosen by the issuer. The issuer then computes msg as a Pedersen hash to $\mathsf{pk}_\mathtt{N}$ and $a_\mathtt{N}$ as $\mathsf{msg} = \mathsf{pk}_\mathtt{N} \cdot \prod_{i=1}^{\ell} H_i^{a_i}$. It then computes the credential as a Groth signature, i.e., it samples $r \leftarrow \mathbb{Z}_p^*$ and sets $\mathsf{cred} = (\hat{R}, S, T) = (\hat{G}^r, (Y \cdot G^{\mathsf{sk}_\mathtt{I}})^{1/r}, (Y^{\mathsf{sk}_\mathtt{I}} \cdot \mathsf{msg})^{1/r})$.

Construction 3: Concrete instantiation: Initialization and registration.

Given that the resulting proof goal still realizes the proof goal in the generic construction, it is important to note that the security analysis of the generic construction still holds true for the instantiation.

Sending. This step is identical to the generic construction.

Forwarding. A node $\mathtt{N}_{ij}$, upon receiving $(\mathsf{msg} = (((\pi_{i1}, \mathsf{nym}_{i1}), \ldots, (\pi_{i,j-1}, \mathsf{nym}_{i,j-1})), \mathtt{sid}), \sigma)$, behaves as follows:

- If $j > 1$, it fetches the public key $\mathsf{pk}_{i,j-1}$ of the sending node, and verifies the validity of the received σ.
- The node re-randomizes $\mathsf{cred} = (\hat{R}, S, T)$ by sampling $r' \leftarrow \mathbb{Z}_p^*$, obtaining $(\hat{R}', S', T') = (\hat{R}^{r'}, S^{1/r'}, T^{1/r'})$. It blinds the result by sampling $\alpha, \beta \leftarrow_\$ \mathbb{Z}_p^*$ and setting $(\hat{R}'', S'', T'') = (\hat{R}', S'^{1/\alpha}, T'^{1/\beta})$. The node computes $\mathsf{nym}_{ij} \leftarrow_\$ \mathsf{H}(\mathtt{sid})^{\mathsf{sk}_{ij}}$ as well as the following two Schnorr-like zero-knowledge proofs:

$$\begin{aligned}\pi'_{ij} \leftarrow \mathsf{NIZK}[(\mathsf{sk}_{ij}, \boldsymbol{a}_{ij}, \alpha, \beta) :\ & e(S'', \hat{R}'')^{\alpha} = e(Y, \hat{G}) \cdot e(G, \mathsf{pk}_{\mathtt{I}}) \ \wedge \\ & e(T'', \hat{R}'')^{\beta} = e(Y, \mathsf{pk}_{\mathtt{I}}) \cdot e\left(\mathsf{H}(\mathtt{setup})^{\mathsf{sk}_{ij}} \prod_{i=1}^{\ell} H_i^{a_i}, \hat{G}\right) \\ & \mathsf{nym}_{ij} = \mathsf{H}(\mathtt{sid})^{\mathsf{sk}_{ij}} \ \wedge \\ & \phi(\boldsymbol{a}) = 1](\mathsf{msg})\end{aligned}$$

$$\sigma'_{ij} \leftarrow \mathsf{NIZK}[(\mathsf{sk}_{ij}) : \mathsf{nym}_{ij} = \mathsf{H}(\mathtt{sid})^{\mathsf{sk}_{ij}} \ \wedge \ \mathsf{pk}_{ij} = \mathsf{H}(\mathtt{setup})^{\mathsf{sk}_{ij}}](\mathsf{msg})\,.$$

- The node finally sets $\pi_{ij} = (\pi'_{ij}, \hat{R}'', S'', T'')$ and transfers $((\pi, (\pi_{ij}, \mathsf{nym}_{ij})), \mathtt{sid}, \sigma_{ij})$ to the next node of each path, i.e., to $\mathtt{N}_{ij+1}$ or Bob in the case that $j = m_i$.

Receiving. This step is identical to the generic construction.

Construction 4: Concrete instantiation: Message routing.

5.3 Efficiency Analysis

In the following we estimate the actual complexity of our construction. As the key generation and registration protocols are only executed once per node, they are omitted in the analysis.

We thus first count the number of predominant operations (i.e., full-length exponentiations as well as pairing evaluations, yet omitting simple group operations or operations in $\mathbb{Z}_p$ as well as hash evaluations) for each node $\mathtt{N}_{ij}$ in the transmission as well as for the final receiver B, cf. Table 1, where for clarity we break down the costs for each step in the computation. Note here that the receiver only has to verify one σ'_{ij}, but all proofs π'_{ij} generated during the transmission, and thus the overall costs are linear in the number n of involved nodes.

		$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	$e(\cdot,\cdot)$
Repeater node	Verifying σ'_{ij}	4	–	–	–
	Computation of nym_{ij}	1	–	–	–
	Deriving $(\hat{R}'', S'', T'')$	4	1	–	–
	Computing π'_{ij}	$\ell + 2$	–	2	4
	Computing σ'_{ij}	2	–	–	–
	Total	$\ell + 13$	1	2	4
Receiver	Verifying final σ'_{ij}	4	–	–	–
	Verifying all π'_{ij}	$(\ell + 3) \cdot n$	–	$4 \cdot n$	$4 \cdot n$
	Total	$(\ell + 3) \cdot n + 4$	–	$4 \cdot n$	$4 \cdot n$

Table 1. Computational costs per node

Besides this theoretical analysis, we also provide actual benchmarks from a prototypical implementation of our application for different numbers of nodes ($n = 10, \ldots, 100$), attributes ($\ell = 10, 20$ assuming that half of them are disclosed), and for single and multi-path settings (in the single-path setting, no pseudonyms are required). The benchmarks were carried out on an Intel Core 2 Duo E8400 CPU with 3.00GHz and 4GB of RAM, running Ubuntu 24.04.2 LTS and Python 3.12.3, using the `mcl 3.00`[3] library for pairing-based crypto. To compensate for the relatively weak hardware of the setup, the figure also includes estimates for an AWS Linux 2 8-core Intel Xeon Platinum 8259CL CPU running at 2.50GHz using 32GB of RAM, obtained using the *zkalc estimator*[4] using the `gnark-crypto` implementation. As a pairing friendly curve, we opted for the BLS12-381 curve, offering about 120bit of security.

The benchmark results are now depicted in Fig. 4.[5] The figure only presents the computational costs of the receiver B. The computational costs of all intermediate nodes were well below 20 ms and thus in the area of network latency, such that they can be ignored in practice. As could be expected from Table 1, the receiver's runtime is linear in the number n of nodes, while the number ℓ or the costs of pseudonyms only play a secondary role.

As can be seen, even for a very high number of 100 repeater nodes (corresponding, e.g., to 3 disjoint paths over more than 3′300 km considering current distance limitations, thus exceeding the maximum distance between any two EU capitals in mainland Europe, i.e., Porto/Portugal and Helsiki/Finland), the computational overhead is in the area of less than 2.5 s and less than 1 s on more powerful hardware, both without applying any optimizations like batching or pre-processing. We consider this overhead acceptable for typical quantum-key distribution deployments, where key material is often generated ahead of use

[3] https://github.com/herumi/mcl

[4] https://zka.lc

[5] The source code of the benchmark implementation can be accessed at https://anonymous.4open.science/r/auditable-qkd-55EF/.

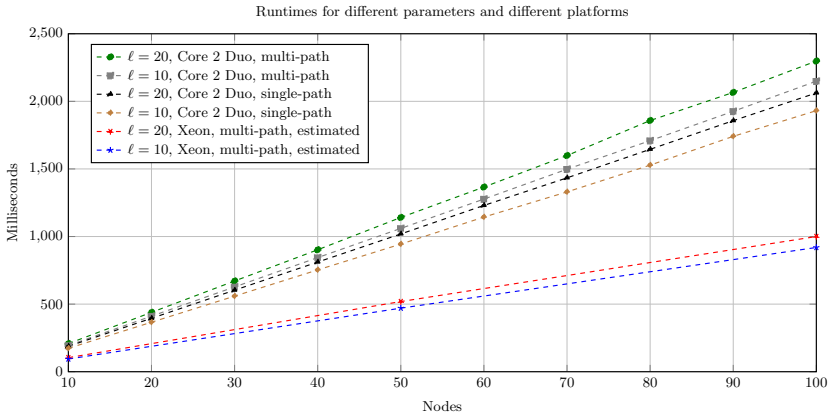

Fig. 4. Runtime evaluation for receiver B.

and buffered rather than produced strictly on-demand. In practice, QKD keys are commonly generated in batches during periods of link availability (for example, during connectivity windows in satellite or hybrid networks) and stored in a key store or HSM for subsequent consumption. As a result, the introduced verification delay generally impacts key availability only at session boundaries and does not translate into added data-path latency or reduced usability for most applications.

Finally, Table 2 shows the overall bandwidth overhead for the receiver B, i.e., the overall size of the received data. We omit the bandwidth requirements of the repeater nodes, as the size of transferred data grows linearly from the first node to the receiver, such that all intermediate nodes have less bandwidth requirements than B. Note that in the table, following the notation from the protocols, π_{ij} also includes the re-randomized signatures $(\hat{R}'', S'', T'') \in \mathbb{G}_2 \times \mathbb{G}_1^2$ per node. In the table, we further assume that the policy ϕ requires to disclose d of the ℓ attributes, while the other $\ell - d$ attributes remain undisclosed.

	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	$\mathbb{Z}_p$
Final σ_{ij}	–	–	–	2
Proofs π_{ij}	$2 \cdot n$	n	–	$5 \cdot n$
Pseudonyms nym_{ij}	n	–	–	–
Total	$3 \cdot n$	n	–	$(4 + d) \cdot n + 2$

Table 2. Bandwidth overhead for receiver B

When estimating the actual bandwidth requirements, we assume a size of 48 B for elements in $\mathbb{G}_1$, 96 B for elements in $\mathbb{G}_2$, and 32 B for elements in $\mathbb{Z}_p$,

which are appropriate values for BLS12-381. As before, we further assume that half of the attributes are disclosed and half of them remain private. This then results in an overall communication size for B of about 67 kB for 100 repeaters and 20 attributes. Given the substantial classical communication already required by standard QKD post-processing, this additional overhead can be considered negligible in practice. For example, in the original (unbiased) BB84 protocol [6], basis reconciliation alone requires exchanging basis information for each detection event, which corresponds to a few bits per detection/sifted bit, even before accounting for parameter estimation, error correction, and authentication.

Considering the above estimates, we thus consider the proposed protocol practical for many application scenarios.

6 Conclusions and Open Challenges

In this work, we presented a protocol to validate the paths over which a QKD transmission has been carried out. To do so, we formally defined the framework including the desired security properties, and provided a generic construction which we formally proved secure. Our concrete instantiation shows the practicality of approach, also for long-distance QKD transmissions with a high number of intermediate repeater nodes.

While our main focus is on QKD networks, the general idea of topology-hiding path validation may be useful also in other settings where an operator wants to keep internal routing and infrastructure confidential, but endpoints still require verifiable evidence that traffic complied with externally specified constraints. This is complementary to traditional approaches such as source-routing, where the sender fixes (and therefore learns) the end-to-end path; in many operational networks routing is hop-by-hop and policy-driven, so requiring the sender to specify, or even know, the route is often unrealistic.

Concretely, one could imagine applications in managed, compliance-driven networks such as carrier backbones (e.g., proving traversal only through "hardened" equipment from an approved vendor set) or critical-infrastructure interconnects (e.g., power-grid, rail, or emergency-services backhaul, where regulators or customers may require evidence that traffic avoided uncertified nodes or certain geographic regions). Related scenarios include confidential content delivery networks (CDNs) (e.g., for proving traffic was handled only by attested gateways or enclaves), and high-assurance multi-path delivery for resilience (e.g., proving that replicated deliveries used disjoint provider resources to reduce correlated failure risk). We emphasize that such uses make sense primarily when hop counts are moderate or sessions are long-lived (so linear proof growth can be amortized), and when deployments already support per-hop cryptographic identities or attestations; within those regimes, our approach offers a practical way to couple routing privacy with receiver-verifiable policy compliance.

Finally, interesting open challenges include achieving index-hiding, where a node does not learn its position in the path. Furthermore, overcoming the current linear growth of the audit proofs with the path length would be interesting

to broaden the applicability of our approach. One possible approach is to use recursive SNARKs [30], where each node proves in constant-size form that it verified the previous proof and correctly applied the prescribed local extension, yielding proof size and receiver verification cost that are independent of the number of hops. A key challenge is to retain practical efficiency under standard deployment constraints and, in particular, to support succinct proofs of multi-path properties such as node-disjointness without reintroducing linear overheads.

Acknowledgments. This work has received funding from the European Union's Horizon Europe research and innovation program under No. 101114043 (QSNP), from the Digital European Program under No. 101091588 (QUARTER) and No. 101091642 (QCI-CAT), and the National Foundation for Research, Technology and Development. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the funding institutions. Neither the funding institutions nor the granting authorities can be held responsible for them.

The authors would also like to thank the anonymous reviewers for detailed comments and suggestions on how to improve the presentation and positioning of the result.

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 209–236. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 2010). https://doi.org/10.1007/978-3-642-14623-7_12
2. Akavia, A., LaVigne, R., Moran, T.: Topology-hiding computation on all graphs. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017, Part I. Lecture Notes in Computer Science, vol. 10401, pp. 447–467. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63688-7_15
3. Azuma, K., Economou, S.E., Elkouss, D., Hilaire, P., Jiang, L., Lo, H.K., Tzitrin, I.: Quantum repeaters: From quantum networks to the quantum internet. Rev. Mod. Phys. **95**, 045006 (Dec 2023). https://doi.org/10.1103/RevModPhys.95.045006
4. Ball, M., Boyle, E., Cohen, R., Kohl, L., Malkin, T., Meyer, P., Moran, T.: Topology-hiding communication from minimal assumptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020: 18th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 12551, pp. 473–501. Springer, Cham, Switzerland, Durham, NC, USA (Nov 16–19, 2020). https://doi.org/10.1007/978-3-030-64378-2_17
5. Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 551–572. Springer Berlin Heidelberg, Germany, Kaoshiung, Taiwan, R.O.C. (Dec 7–11, 2014). https://doi.org/10.1007/978-3-662-45611-8_29

6. Bennett, C.H., Brassard, G.: An update on quantum cryptography (impromptu talk). In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology – CRYPTO'84. Lecture Notes in Computer Science, vol. 196, pp. 475–480. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 1984). https://doi.org/10.1007/3-540-39568-7_39
7. Bernhard, D., Pereira, O., Warinschi, B.: How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 626–643. Springer Berlin Heidelberg, Germany, Beijing, China (Dec 2–6, 2012). https://doi.org/10.1007/978-3-642-34961-4_38
8. Bobolz, J., Eidens, F., Krenn, S., Ramacher, S., Samelin, K.: Issuer-hiding attribute-based credentials. In: Conti, M., Stevens, M., Krenn, S. (eds.) CANS 21: 20th International Conference on Cryptology and Network Security. Lecture Notes in Computer Science, vol. 13099, pp. 158–178. Springer, Cham, Switzerland, Vienna, Austria (Dec 13–15, 2021). https://doi.org/10.1007/978-3-030-92548-2_9
9. Bosk, D., Frey, D., Gestin, M., Piolle, G.: Hidden issuer anonymous credential. Proceedings on Privacy Enhancing Technologies **2022**(4), 571–607 (Oct 2022). https://doi.org/10.56553/popets-2022-0123
10. Brauer, M., Vicente, R.J., Buruaga, J.S., Méndez, R.B., Braun, R., Geitz, M., Rydlichowski, P., Brunner, H.H., Fung, C.F., Peev, M., Pastor, A., López, D.R., Martin, V., Brito, J.P.: Linking QKD testbeds across europe. Entropy **26**(2), 123 (2024)
11. Camenisch, J., Drijvers, M., Dubovitskaya, M.: Practical UC-secure delegatable credentials with attributes and their application to blockchain. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security. pp. 683–699. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017). https://doi.org/10.1145/3133956.3134025
12. Camenisch, J., Krenn, S., Lehmann, A., Mikkelsen, G.L., Neven, G., Pedersen, M.Ø.: Formal treatment of privacy-enhancing credential systems. In: Dunkelman, O., Keliher, L. (eds.) SAC 2015: 22nd Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 9566, pp. 3–24. Springer, Cham, Switzerland, Sackville, NB, Canada (Aug 12–14, 2016). https://doi.org/10.1007/978-3-319-31301-6_1
13. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski, Jr., B.S. (ed.) Advances in Cryptology – CRYPTO'97. Lecture Notes in Computer Science, vol. 1294, pp. 410–424. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 1997). https://doi.org/10.1007/BFb0052252
14. Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: Dwork, C. (ed.) Advances in Cryptology – CRYPTO 2006. Lecture Notes in Computer Science, vol. 4117, pp. 78–96. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2006). https://doi.org/10.1007/11818175_5
15. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (1985). https://doi.org/10.1145/4372.4373
16. Connolly, A., Lafourcade, P., Perez-Kempner, O.: Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022: 25th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes

in Computer Science, vol. 13177, pp. 409–438. Springer, Cham, Switzerland, Virtual Event (Mar 8–11, 2022). https://doi.org/10.1007/978-3-030-97121-2_15
17. Cozzolino, M., Krenn, S., Lorünser, T.: Topology-hiding connectivity-assurance for qkd inter-networking. Tech. rep. (2025)
18. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO'86. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
19. Franzoi, N., Krenn, S., Lorünser, T., Ramacher, S.: Secure key forwarding for large-scale quantum key distribution networks. In: International Conference on Quantum Communications, Networking, and Computing – QCNC 2025. pp. 486–493. IEEE, Nara, Japan (Mar 31 – Apr 1, 2025)
20. Geitz, M., Döring, R., Braun, R.: Hybrid QKD & PQC protocols implemented in the berlin openqkd testbed. In: ICFSP. pp. 69–74. IEEE (2023)
21. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: 17th Annual ACM Symposium on Theory of Computing. pp. 291–304. ACM Press, Providence, RI, USA (May 6–8, 1985). https://doi.org/10.1145/22145.22178
22. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing **17**(2), 281–308 (Apr 1988)
23. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) Advances in Cryptology – ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 444–459. Springer Berlin Heidelberg, Germany, Shanghai, China (Dec 3–7, 2006). https://doi.org/10.1007/11935230_29
24. Groth, J.: Efficient fully structure-preserving signatures for large messages. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015, Part I. Lecture Notes in Computer Science, vol. 9452, pp. 239–259. Springer Berlin Heidelberg, Germany, Auckland, New Zealand (Nov 30 – Dec 3, 2015). https://doi.org/10.1007/978-3-662-48797-6_11
25. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) Advances in Cryptology – EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer Berlin Heidelberg, Germany, Istanbul, Turkey (Apr 13–17, 2008). https://doi.org/10.1007/978-3-540-78967-3_24
26. Henrich, J., Heinemann, A., Stiemerling, M., Seidl, F.: Crypto-agile design and testbed for qkd-networks. In: EICC. pp. 191–192. ACM (2023)
27. Hirt, M., Maurer, U., Tschudi, D., Zikas, V.: Network-hiding communication and applications to multi-party protocols. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 335–365. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53008-5_12
28. Huttner, B., Alléaume, R., Diamanti, E., Fröwis, F., Grangier, P., Hübel, H., Martín, V., Poppe, A., Slater, J.A., Spiller, T., Tittel, W., Tranier, B., Wonfor, A., Zbinden, H.: Long-range QKD without trusted nodes is not possible with current technology. CoRR **abs/2210.01636** (2022). https://doi.org/10.48550/ARXIV.2210.01636

29. Kim, T.H., Basescu, C., Jia, L., Lee, S.B., Hu, Y., Perrig, A.: Lightweight source authentication and path validation. In: Bustamante, F.E., Hu, Y.C., Krishnamurthy, A., Ratnasamy, S. (eds.) ACM SIGCOMM 2014 Conference, SIGCOMM'14, Chicago, IL, USA, August 17-22, 2014. pp. 271–282. ACM (2014). https://doi.org/10.1145/2619239.2626323
30. Kothapalli, A., Setty, S.T.V.: HyperNova: Recursive arguments for customizable constraint systems. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part X. Lecture Notes in Computer Science, vol. 14929, pp. 345–379. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68403-6_11
31. Krenn, S., Lorünser, T.: An Introduction to Secret Sharing - A Systematic Overview and Guide for Protocol Selection. Springer Briefs in Computer Science, Springer (2023). https://doi.org/10.1007/978-3-031-28161-7
32. Kutschera, F., Dervisevic, E., Behan, L., López, D.R., Mehic, M., Voznák, M., Hübel, H., Pastor, A., Cepeda, L.: Data acquisition and simulation tools for virtual QKD testbed access - examples from the OPENQD project. In: ECOC. pp. 1–4. IEEE (2021)
33. LaVigne, R., Zhang, C.D.L., Maurer, U., Moran, T., Mularczyk, M., Tschudi, D.: Topology-hiding computation beyond semi-honest adversaries. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018: 16th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 11240, pp. 3–35. Springer, Cham, Switzerland, Panaji, India (Nov 11–14, 2018). https://doi.org/10.1007/978-3-030-03810-6_1
34. LaVigne, R., Zhang, C.D.L., Maurer, U., Moran, T., Mularczyk, M., Tschudi, D.: Topology-hiding computation for networks with unknown delays. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 12111, pp. 215–245. Springer, Cham, Switzerland, Edinburgh, UK (May 4–7, 2020). https://doi.org/10.1007/978-3-030-45388-6_8
35. Li, J., Su, Y., Lu, R., Su, Z., Meng, W., Shen, M.: Stealthpath: Privacy-preserving path validation in the data plane of path-aware networks. IEEE Trans. Dependable Secur. Comput. **22**(1), 192–204 (2025). https://doi.org/10.1109/TDSC.2024.3392299
36. Liu, C., Che, X., Xie, J., Dong, Y.: A multi-path qkd algorithm with multiple segments. Journal of Cyber Security and Mobility **13**(02), 193–214 (Feb 2024). https://doi.org/10.13052/jcsm2245-1439.1321, https://journals.riverpublishers.com/index.php/JCSANDM/article/view/23347
37. Lorünser, T., Krenn, S., Pacher, C., Schrenk, B.: On the security of offloading post-processing for quantum key distribution. Entropy **25**(2), 226 (2023). https://doi.org/10.3390/E25020226
38. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999: 6th Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 1758, pp. 184–199. Springer Berlin Heidelberg, Germany, Kingston, Ontario, Canada (Aug 9–10, 1999). https://doi.org/10.1007/3-540-46513-8_14
39. Mehic, M., Niemiec, M., Rass, S., Ma, J., Peev, M., Aguado, A., Martin, V., Schauer, S., Poppe, A., Pacher, C., Voznák, M.: Quantum key distribution: A networking perspective. ACM Comput. Surv. **53**(5), 96:1–96:41 (2021)

40. Mir, O., Bauer, B., Griffy, S., Lysyanskaya, A., Slamanig, D.: Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials. In: Meng, W., Jensen, C.D., Cremers, C., Kirda, E. (eds.) ACM CCS 2023: 30th Conference on Computer and Communications Security. pp. 30–44. ACM Press, Copenhagen, Denmark (Nov 26–30, 2023). https://doi.org/10.1145/3576915.3623203
41. Moran, T., Orlov, I., Richelson, S.: Topology-hiding computation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015: 12th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 9014, pp. 159–181. Springer Berlin Heidelberg, Germany, Warsaw, Poland (Mar 23–25, 2015). https://doi.org/10.1007/978-3-662-46494-6_8
42. Naous, J., Walfish, M., Nicolosi, A., Mazières, D., Miller, M., Seehra, A.: Verifying and enforcing network paths with icing. In: Cho, K., Crovella, M. (eds.) Proceedings of the 2011 Conference on Emerging Networking Experiments and Technologies, Co-NEXT '11, Tokyo, Japan, December 6-9, 2011. p. 30. ACM (2011). https://doi.org/10.1145/2079296.2079326
43. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) Advances in Cryptology – CRYPTO'89. Lecture Notes in Computer Science, vol. 435, pp. 239–252. Springer, New York, USA, Santa Barbara, CA, USA (Aug 20–24, 1990). https://doi.org/10.1007/0-387-34805-0_22
44. Sengupta, B.: VALNET: privacy-preserving multi-path validation. Comput. Networks **204**, 108695 (2022). https://doi.org/10.1016/J.COMNET.2021.108695
45. Sengupta, B., Li, Y., Bu, K., Deng, R.H.: Privacy-preserving network path validation. ACM Trans. Internet Techn. **20**(1), 5:1–5:27 (2020). https://doi.org/10.1145/3372046
46. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176
47. Valbusa, F., Lorünser, T., Spini, G., Laschet, S.: Relaxing the single point of failure in quantum key distribution networks: An overview of multi-path approaches. In: Skopik, F., Naessens, V., Sutter, B.D. (eds.) Availability, Reliability and Security - ARES 2025 EU Projects Symposium Workshops, Ghent, Belgium, August 11-14, 2025, Proceedings, Part I. Lecture Notes in Computer Science, vol. 15998, pp. 183–200. Springer (2025). https://doi.org/10.1007/978-3-032-00642-4_11