Key Management und Kontrollsysteme für QKD

von Jasmin Neumann, Felix Trunk, Susanne Naegele-Jackson

Inhalt

Einführung

Quantum Key Distribution (QKD) ist eine Methode zur Erzeugung sicherer kryptographischer Schlüssel zwischen Netzwerkteilnehmern. Bisher sind einige QKD-Systeme gut erforscht und kommerziell verfügbar. Aber mit dem bloßen Erzeugen der Schlüssel ist nur der erste Schritt für eine kryptographische Applikation abgeschlossen; weitere Schritte sind die Schlüsselverwaltung und Schlüsselverteilung: Generierte Schlüssel müssen sicher und systematisch in einem Schlüsselspeicher mit entsprechender Puffergröße abgespeichert werden können, um im Bedarfsfall rechtzeitig zur Verfügung zu stehen. Dabei muss stets die Sicherheit in Form von Authentifizierung und Schlüssellebenszyklus-Management eingehalten werden. Key Management Systeme (KMSs) müssen darüber hinaus auch gewährleisten, dass die Schlüssel über geeignete Links im Netz so verteilt werden, dass sie zum richtigen Zeitpunkt an der Zielapplikation zur Verfügung stehen. Hierfür gibt es ähnlich zu einem klassischen Netzwerk Kontrollstrukturen, die das Schlüsselmanagement unterstützen. Außerdem bietet sich ebenfalls die Verknüpfung mit Software Defined Networking (SDN) an.

Im Folgenden wird zunächst das KMS und dessen Aufgaben ausführlich vorgestellt, sowie die Kontrollebene in einem QKD Netz (QKDN) näher beschrieben. Anschließend wird auf die offenen Fragestellungen beim derzeitigen Stand der Standardisierungen (sowohl bei ETSI, als auch bei ITU) bei den verschiedenen Schnittstellen von QKD-, KMS- & Kontroll-Systemen und deren benötigte Key Monitoring Parameter eingegangen.

1. Das Key Management System (KMS) im QKDN

In einem Netz zur Quantenschlüsselverteilung unterscheidet man mehrere Schichten (siehe ***Abbildung 1***): Auf unterster Ebene ist der Physical Layer (Quantum Forwarding Plane (QFP); im Bild blau) in dem die Quantenschlüssel erzeugt werden; der Key Management Layer (grün) darüber ist für die Verwaltung und das Forwarding der Schlüssel zuständig. Darüber liegt der QKDN-Control Layer (orange; wird im folgenden Kapitel ***Kontrollsysteme im (QKD)-Netz*** näher erläutert). Die oberste Schicht ist der Application Layer (grau) wo die Schlüssel von den Anwendungen verbraucht werden.

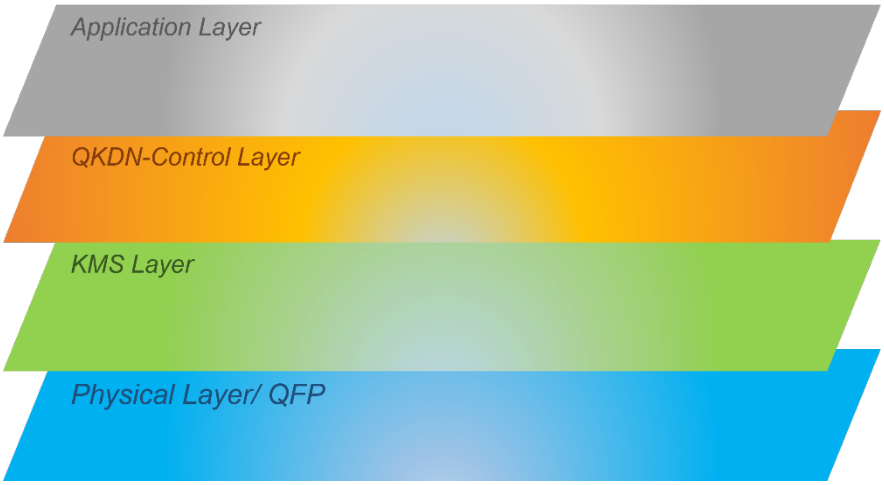

Abbildung 1: *Hierarchisches Schichtenmodell QKDN*

Ein KMS des KMS-Layers nimmt zum einen die generierten Schlüssel der QKD-Geräte (Physical Layer/QFP) entgegen und speichert sie in verschiedenen Schlüsselspeichern ab. Das KMS kann nicht nur die Schlüssel von der untersten Schicht anfordern, sondern sich auch mit den anderen KMSs an anderen Knoten bezüglich Schlüsselverteilung synchronisieren. Das KMS sorgt zum anderen dafür, dass die sicheren, mittels QKD erzeugten Schlüsselpaare, über die Strecken nach vorgegebenem Routing an die Applikationen verteilt werden und auch im korrekten Format zur Verfügung stehen, wenn die Anwendung sie benötigt.

Zu den grundlegenden Funktionen eines KMS gehören:

- das sichere Abspeichern von Schlüsseln in Puffern inkl. Formatierung der Schlüssel (Funktion 1)
- der Auf-/Abbau gemeinsamer Schlüsselspeicher, sowie der Schlüsselaustausch (Relaying) anhand vorgegebener Routen (Funktion 2)
- die rechtzeitige Schlüsselauslieferung an die Applikationen (Supply) nach vorgegebener Quality of Service (QoS) (Funktion 3)

Weitere Aufgaben eines KMS umfassen:

- die Synchronisation, die Authentifizierung, sowie die Verwaltung der Schlüsseldatenbanken
- die Überprüfung der Füllstände dieser Puffer, sowie die damit verbundene rechtzeitige Neuanforderung bei einem empirisch festzulegenden Schwellwert
- die Beteiligung an der Key Rotation (Ersetzen der alten Schlüssel, damit die Daten, die durch den derzeitigen Schlüssel verschlüsselt sind, begrenzt sind, falls dieser doch öffentlich wird).

Der ETSI Standard ***ETSI GS QKD 004*** unterscheidet beim KMS zwischen einem QKD Key Manager Peer (South-Bound KMS) pro QKD-Device, der die Schlüssel auf dem QKD Link Level entgegennimmt und einem QKD Key Server (North-Bound KMS) pro Knoten auf dem QKD Network Level (siehe ***Abbildung 2***).

Die Key Server (Funktionen 2/3) werden auch benötigt, um den Austausch über mehrere QKD-Devices hinweg zu ermöglichen. Die Key Server nutzen dieselbe Anwendungsschnittstelle für den Schlüsseltransport mit den Key Manager Peers (grüne Pfeile), wie auch mit den Applikationen (orange Pfeile).

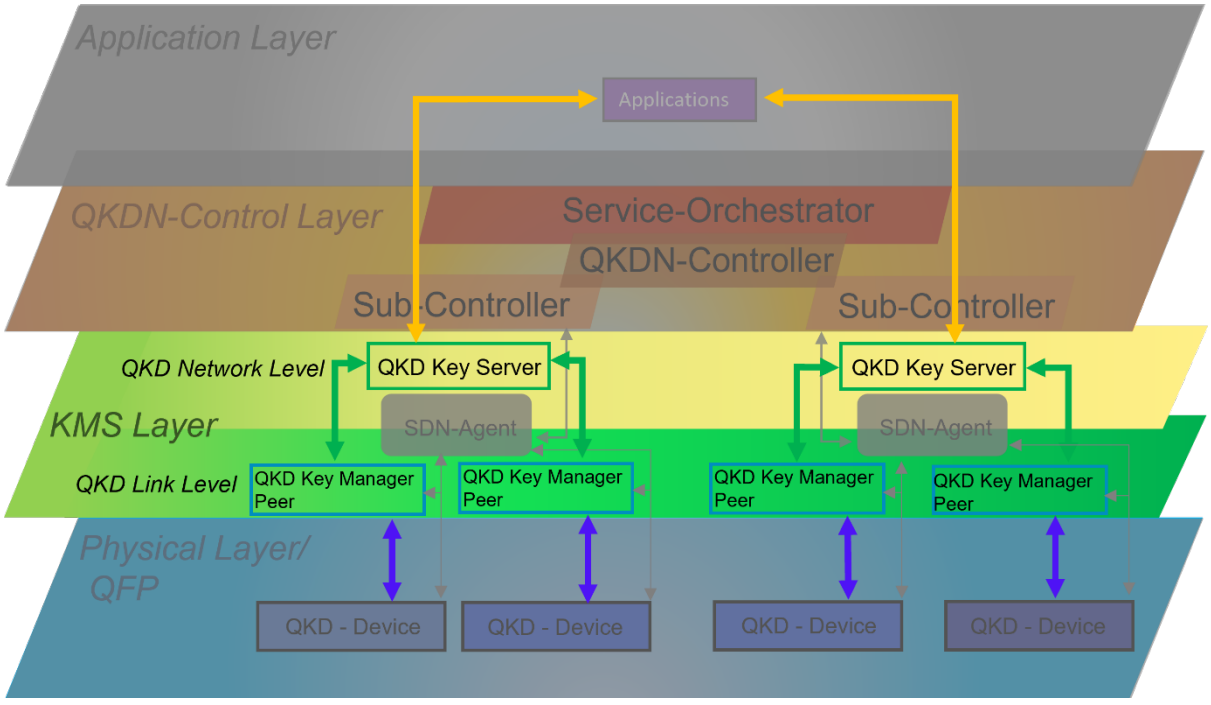

Abbildung 2: *KMS-Hierarchie nach ETSI*
(Dicke farbige Pfeile kennzeichnen den vertikalen Schlüsseltransport)

Die ITU-T Recommendation ***ITU-T Y.3803*** untergliedert ebenfalls das KMS in zwei Stufen: quantenorientierter Key Management Agent (KMA) und anwendungsorientierter Key Supply Agent (KSA) (siehe ***Abbildung 3***).

Der KMA kümmert sich um das Key Relay, also den Transport zwischen KMSs verschiedener vertrauenswürdiger Knoten (Konzept wird anschließend noch genauer erläutert), das Abspeichern und das Key Life Cycle Management, während der KSA mit der Bereitstellung der Schlüssel für die Applikation vertraut ist (und mit der Kombination anderer Schlüssel). Anstelle des in ETSI zur Kontrollschicht gehörenden SDN-Agenten, befindet sich hier außerdem ein spezielles Control- und Management Modul direkt im KMS-, QKD- & Control-Layer. Diese Management Module sind nicht nur untereinander verknüpft (über die Schnittstellen *Ck, Kq-2*), sondern auch jeweils mit einer gesonderten, allumfassenden Management-Schicht (z.B. über Schnittstelle *Mk*). Man erkennt hier ebenfalls, wie die Schlüssel vom QKD Layer zum KMS Layer über Schnittstelle *Kq-1* und von dort weiter über Schnittstelle *Ak* direkt mit der Applikation ausgetauscht werden, während auch hier ein Control Layer zwischengeschaltet ist. Über die Schnittstellen *Kx* soll ein Schlüsselaustausch zwischen KMSs ermöglicht werden.

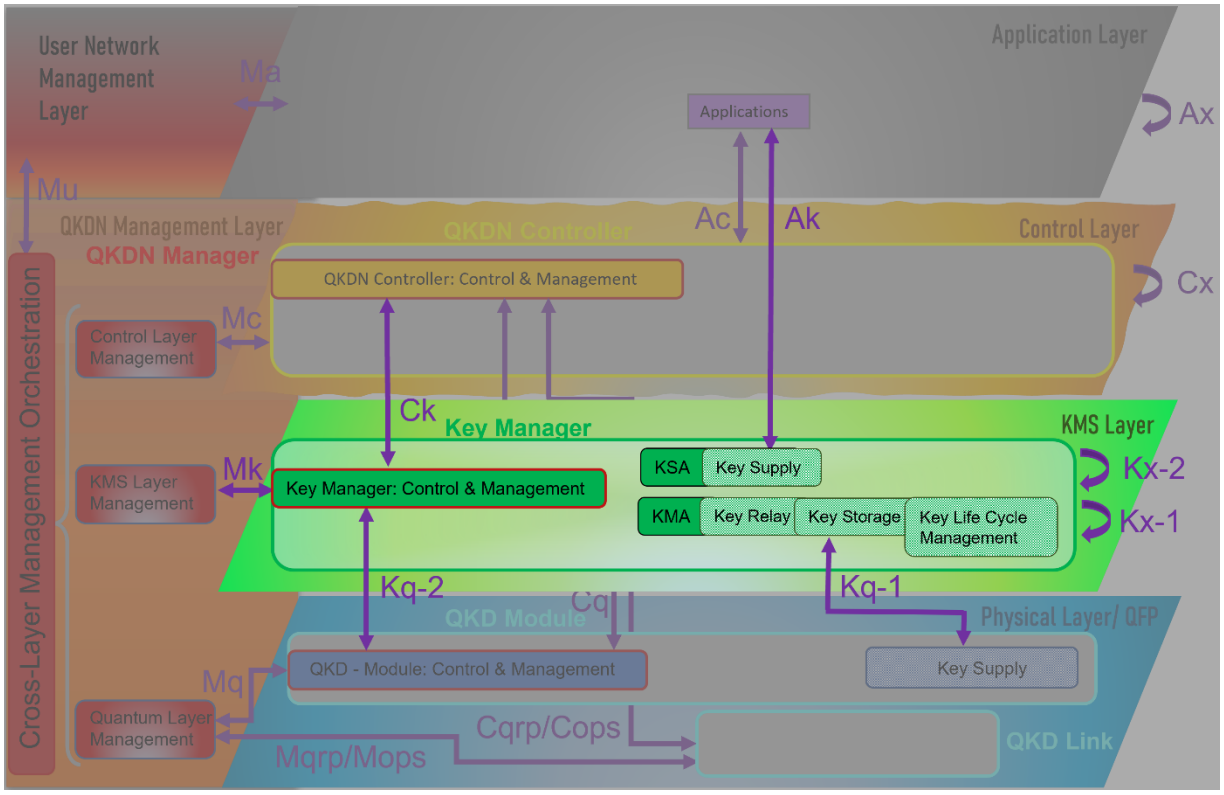

Abbildung 3: *KMS-Hierarchie nach ITU-T*

Nähere Erläuterungen zu ETSI und ITU-T Standards sind im Kapitel ***Interoperabilität & Standardisierungen*** zu finden.

Man unterscheidet nach [1] bei der Schlüsselausgabe an die Applikationen zwischen Einmalschlüssel- & Schlüsselstromsystemen. Unter ersteren versteht man die Anforderung eines Schlüssels für jeden einzelnen Vorgang mit einer neuen, unabhängigen Anfrage über eine Key-ID. Der Vorteil der sich hier bietet ist, dass keine Zustände über vorherige Schlüsselanforderungen zwischengespeichert werden müssen. Allerdings bedeutet dies auch, dass man keine automatische QoS (z.B. Mindestdurchsatz von Schlüsseln, Netzwerkpfade) einhalten kann, da man keine Vorhersage über den Schlüsselverbrauch treffen kann. Zudem ergibt sich durch die erneuten Anfragen eine gewisse Zeitverzögerung. Im Gegensatz dazu bringen Schlüsselstromsysteme ein Sitzungskonzept mit. Hierzu wird zwischen den beiden Applikationen ein Schlüsselstrom zur zeitgleichen Schlüsselverteilung über eine Schlüsselstrom-ID definiert, woraufhin sequentiell die Schlüssel übermittelt werden. Hierbei kann inhärent eine QoS umgesetzt werden, welche auch einfach auf neue QoS-Aspekte ausgedehnt werden kann. Dieses Verfahren eignet sich besonders für Echtzeitapplikationen mit verschiedenen Qualitätsanforderungen, wie man sie beispielsweise beim Streaming findet. Dafür sind die Schlüsselstromsysteme aus ETSI GS QKD 004 auch deutlich komplexer und benötigen mehr Ressourcen. Einmalschlüsselsysteme ohne QoS können dagegen vergleichsweise leicht mittels RESTful-APIs (z.B. ETSI GS QKD 014) implementiert werden.

Aktuelle Entwicklungen zeigen allerdings, dass es sich für eine Entwicklung im realen Umfeld anbietet, die beiden Standards entsprechend zu kombinieren. Es wird dabei häufig von „ETSI 004 +“ gesprochen. Dabei soll sich die reiche Parameterauswahl (QoS) von ETSI GS QKD 004 mit einer einfachen Implementierung vereint werden.

Der Prozess der Schlüsselanforderung von der Applikation an das KMS muss gewissen Kriterien entsprechen [1]:

1. Schlüsselerzeugung: Um die QKD-Geräte entlang des vorher festgelegten Pfades auffordern zu können das Schlüsselmaterial zwischen den paarweisen QKD-Devices zu produzieren, muss der QKDN-Controller mit den lokalen KMSs entlang des Pfades entsprechend interagieren und konkrete Pfadangaben zur Planung des Schlüsselbedarfs angeben. Somit nimmt das KMS - wie im Schichtenmodell bereits zu erkennen - eine Vermittlerposition zwischen der QKD-Schicht und der Kontroll-Schicht ein (vgl. ***Abbildung 1***).
2. Schlüsselklassifizierung: Schlüssel besitzen unterschiedliche Eigenschaften, welche für den Nutzer bzw. die Applikation von Bedeutung sind (z.B. die Länge) und verschiedene Netzwerkparameter (z.B. die Hops über vertrauenswürdige Knoten). Für die Einteilung in Klassen müssen die unterschiedlichen lokalen KMSs zusammenarbeiten. Bei der Schlüsselverteilung muss auch eine zweifelsfreie Identifikation sichergestellt werden.
3. Schlüsselverteilung: Nach erfolgter Klassifizierung können die Schlüssel entsprechend geroutet werden.
4. Schlüsselspeicherung: Um Schlüssel nicht instantan bei Applikationsanforderung erzeugen zu müssen, nutzen die KMS einen Puffer, um Bedarfsspitzen abzufangen und die Zeit der Schlüsselerstellung zu überbrücken. Diese Abspeicherung muss in den KMSs standardisiert, sicher und schematisch erfolgen.
5. Schlüsselzuweisung: Besonders wenn keine initiale Anfrage einer Applikation vorhergegangen ist, sondern die Schlüssel für die Füllung der Puffer auf Vorrat erstellt wurden, muss es Protokolle geben, die die existierenden Schlüssel auch bei mehreren gleichzeitigen Anfragen fair aufteilen, ggf. auch unter Berücksichtigung bestimmter QoS.

Um einen Schlüsselaustausch zwischen Netzteilnehmern – ohne eine direkte Verbindung über einen QKD-Link zweier gepaarter QKD-Devices – zu ermöglichen, findet sog. Key Relaying statt. Dabei wird die Strecke über einzelne Punkt-zu-Punkt (P2P) Verbindungen zwischen den Knoten mit einzelnen, direkten Schlüsseln überbrückt, sodass die gesamte Strecke über einen indirekten Schlüssel verbunden ist (siehe ***Abbildung 4***) [1]. Dieser kann z.B. durch XOR-Operationen zwischen den direkten Schlüsseln erzeugt werden [2]. Dort wird auch eine weitere Möglichkeit zum Key Relay mittels verschlüsselt ausgetauschten Zufallszahlen vorgeschlagen. Aus Sicherheitsaspekten ist es aber vorzuziehen, dass die verschlüsselten Schlüssel direkt zum Zielknoten/ zentralen Knoten geschickt werden und dort erst entschlüsselt werden.

In [3] wurde zum Beispiel eine effiziente Optimierung für einen Key Relay Algorithmus entwickelt, um die benötigten QKD-Schlüssel beim Übergang zwischen vertrauenswürdigen Knoten möglichst gering zu halten. Es werden dabei nur Knoten für die Route ausgewählt, wenn daraus ein möglichst geringer Schlüsselbedarf entsteht.

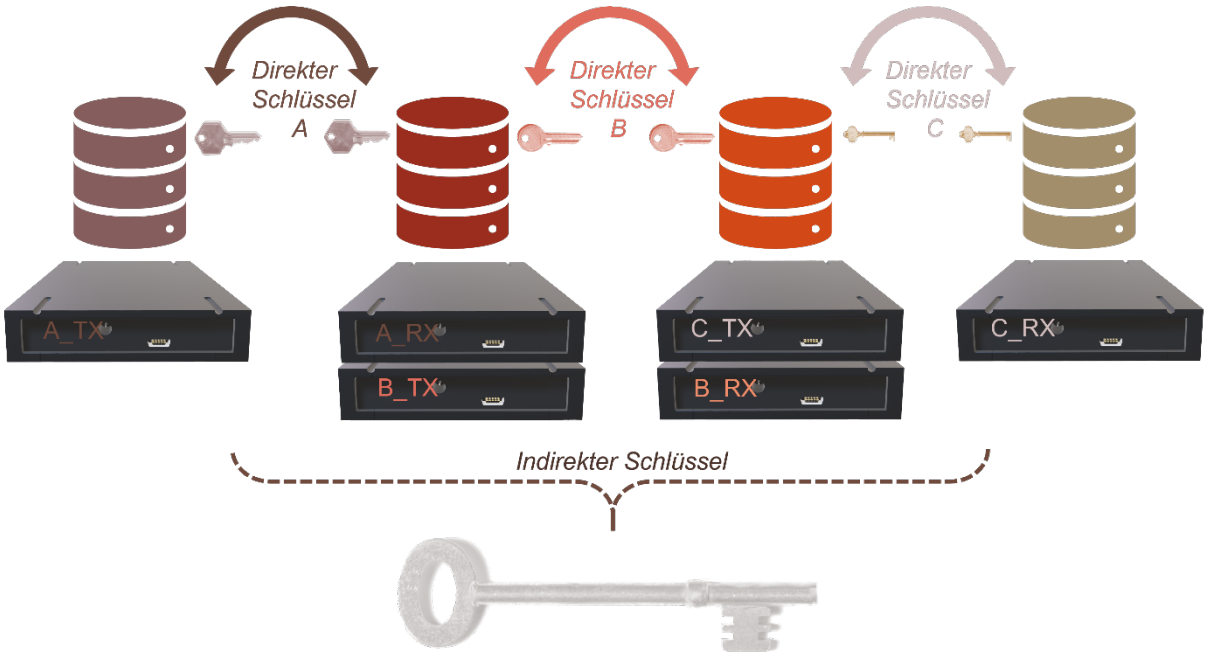

Abbildung 4: *Schlüsselaustausch über vier vertrauenswürdige Knoten nach* [1]

Dabei gehört das Routing nicht zu den Aufgaben des KMS [NIST SP 800, [4]], sondern der QKDN-Controller berechnet die optimale Route. Das Routing der Schlüssel kann zentral oder verteilt mit QKDN Controllern gesteuert werden und ist im Allgemeinen mit SDN verknüpft. Der Controller transportiert die Schlüssel aber nicht selbst, sondern das KMS versorgt die kryptographische Applikation mit Schlüsseln über die vom QKDN Controller berechnete Route [5]. Näheres zu Controllern im QKDN findet sich in Kapitel ***Kontrollsysteme im (QKD)-Netz***.

Abschließend werden die indirekten Schlüssel an die Applikation ausgegeben, damit diese die Nutzerdaten entsprechend verschlüsseln kann. Dazu muss noch der Ort der Abspeicherung der Schlüssel durch die entsprechenden Applikationen festgelegt werden. Man unterscheidet zwischen Peer-to-Peer (Schlüsselmaterial wird am Verbrauchsknoten/Clientrechner abgespeichert) und Service-to-Service (Schlüsselmaterial wird an einem Applikationsserver gespeichert). Im letzteren Fall wird der Schlüssel und die verschlüsselte Nachricht mit entsprechenden Userdaten abgespeichert, um diesen für die Anfrage bereit zu haben. Service-to-Service beinhaltet einen weiteren sicherheitskritischen Aspekt: die Daten zwischen den Clientrechnern und den Applikationsservern werden im Klartext übertragen [1]. Um die Sicherheit dennoch aufrecht zu erhalten, müssen diese Links mit Post Quantum Cryptography (PQC) verschlüsselt werden.

Zudem gibt es ein Schlüssellebenszyklus-Management, das je nach globaler Policy die ungenutzten Schlüssel in den KMS-Puffern nach Neuanforderung nach einer gewissen Zeitspanne wieder löscht [6]. Hierzu müssen Time-Constraints ähnlich wie die Key-ID bei der Schlüsselübermittlung mitgeführt werden. Auch die Key Rotation der alten verwendeten Schlüssel bei der Anwendung, sowie die Synchronisation zählen zum Lebenszyklus Management der Schlüssel.

Bei KMSs von besonderer Bedeutung ist natürlich der Sicherheitsaspekt, da der Schlüsselaustausch zwischen KMSs durch den Transport über das Netzwerk und auch insbesondere an den vertrauenswürdigen Knoten gegen potentielle Angreifer geschützt werden muss. Zur Absicherung des KMS gehört die Authentifikation, die Autorisierung und die Überwachung. Bei einer KMS Attacke wird das KMS selbst angegriffen, sodass dieses die Enkryptoren nicht mit Schlüsseln versorgen kann [7]. KMSs bieten die Schwachstellen, dass sie nicht mit vielen Anfragen von den Applikationen gleichzeitig umgehen können, weil die Schlüsselpuffer leerlaufen können, wenn nicht genug Schlüsselmaterial ausreichend schnell produziert werden kann [8]. Es muss also generell ein Ausfallszenario definiert werden, wobei sich auch hier PQC als Ersatz anbietet.

Doch obwohl jedes QKDN über eine entsprechende Schlüsselverwaltung verfügen muss und auch SDN immer häufiger eingesetzt wird, kommt es an dieser zentralen und bedeutungsvollen Stelle im QKDN häufig zu offenen Fragestellungen bei der Umsetzung bei der realen Interaktion zwischen den verschiedenen Schichten vor allem bei Geräten unterschiedlicher Hersteller. Um eine reibungsfreie Kommunikation zwischen allen Netzwerkbestandteilen zu ermöglichen, bedarf es insbesondere wohl definierter Schnittstellen, die in Kapitel ***Interoperabilität & Standardisierungen*** behandelt werden.

2. Kontrollsysteme im (QKD)-Netz

Ein modernes Netz, das auf automatischen Elementen und Software-Defined Networking (SDN) beruht, verwendet SDN zur Steuerung, d.h. mit Hilfe von Software wird die Datenebene von der Kontrollebene getrennt und es werden übergeordnete SDN-Instanzen verwendet, die die zentrale Logik der Kontrolle im Netz erhalten. Zudem wird die Abarbeitung einer Dienstleistung überwacht, sodass die Serviceanforderung des Kunden korrekt mit den dafür notwendigen Ressourcen umgesetzt werden kann. Im Folgenden werden die Bedeutung von SDN/QKDN-Controller, Netzwerkorchestrierung und -Management klar abgrenzt und auf Unterschiede bei den Begrifflichkeiten bezüglich ETSI und ITU-T hingewiesen.

Ein klassischer Netz-Controller ist für die Steuerung der Netzkomponenten zuständig und abstrahiert den Zugang zu den zugehörigen Ressourcen des (klassischen) Netzwerks für den Service-Orchestrator, der für die Umsetzung der gesamten Dienstleistung verantwortlich ist. Bietet ein Netz auch Quantenschlüsselverteilung, so wird zusätzlich für den Quantenbereich ein QKDN-Controller eingesetzt. Zur Kombination beider Netzwerke wird bei ETSI ein SDN-Orchestrator (~ Service-Orchestrator) eingesetzt, der dadurch auch den Übergang/ das Routing zwischen verschiedenen Domänen gewähren kann, siehe ***Abbildung 5***. Der SDN-Orchestrator kann automatisch die im Netz verfügbaren Ressourcen zuteilen und übernimmt ähnliche Aufgaben assistierend wie ein SDN-Controller nur domänenübergreifend.

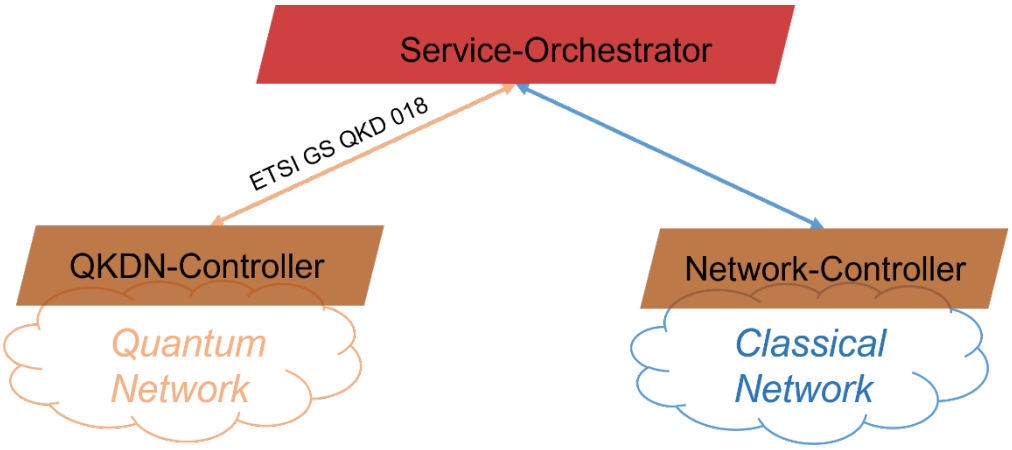

Abbildung 5: *Verbindung QKDN-Domäne mit klassischer Domäne bei ETSI 018*

Controller können sowohl dezentral am Knoten, als auch zentralisiert vorkommen. Je nach Netzwerkarchitektur kann der QKDN Controller auch mehrere Sub-Controller steuern. Beim Schlüsselaustausch berechnet der QKDN-Controller zwar die Route, transportiert die Schlüssel aber nicht selbst, sondern das KMS versorgt die kryptographische Applikation mit Schlüsseln, welches auch als erstes den Service Request erhält. Ein QKDN-Controller ist nach ITU-T für die Konfiguration des QKD Netzwerkes, das Routing, die Zugangskontrolle, sowie für die Policy-basierte Kontrolle und die Sitzungskontrolle verantwortlich [9]. Ein SDN-Controller virtualisiert das QKDN und kontrolliert alle programmierbaren Elemente unter sich und ist zudem direkt für die Applikationsregistrierung und die Topologie-Akquirierung zuständig [10]. Außerdem abstrahiert bei ETSI der Controller den zuständigen Teil des Netzwerks für den SDN-Orchestrator, damit dieser aus den relevanten Informationen das Netzwerk optimieren und Netzwerkressourcen zuweisen kann.

Laut ***ETSI GS QKD 015*** werden bei zentralisiertem SDN-Controller, zusätzlich SDN-Agenten am Knoten eingesetzt, die alle am Knoten befindlichen Geräte (QKD-Devices und KMS) nach den Vorgaben des SDN-Controllers steuern. Dazu gehört sowohl die Konfiguration der QKD-Module mit den erhaltenen abstrahierten Informationen, als auch der Abgleich der Zustände der Schlüssel in den KMS-Puffern mit dem Bedarf. In ETSI GS QKD 015 sollte zudem jeder SDN-fähige QKD-Knoten ein KMS enthalten, welches Schlüsselmaterial von verschiedenen Assoziationen sammelt und viele logische Schlüsselspeicher haben kann. Es sollen die Applikationen mit ihren QoS-Vorgaben vom KMS registriert und die Monitoring-Parameter bereitgestellt werden.

Abbildung 6 zeigt die hierarchischen Kontrollstrukturen in einem QKDN nach ETSI-Nomenklatur, die einen zentralen SDN-Orchestrator, mehrere SDN-Controller und mehrere SDN-Agenten kennt.

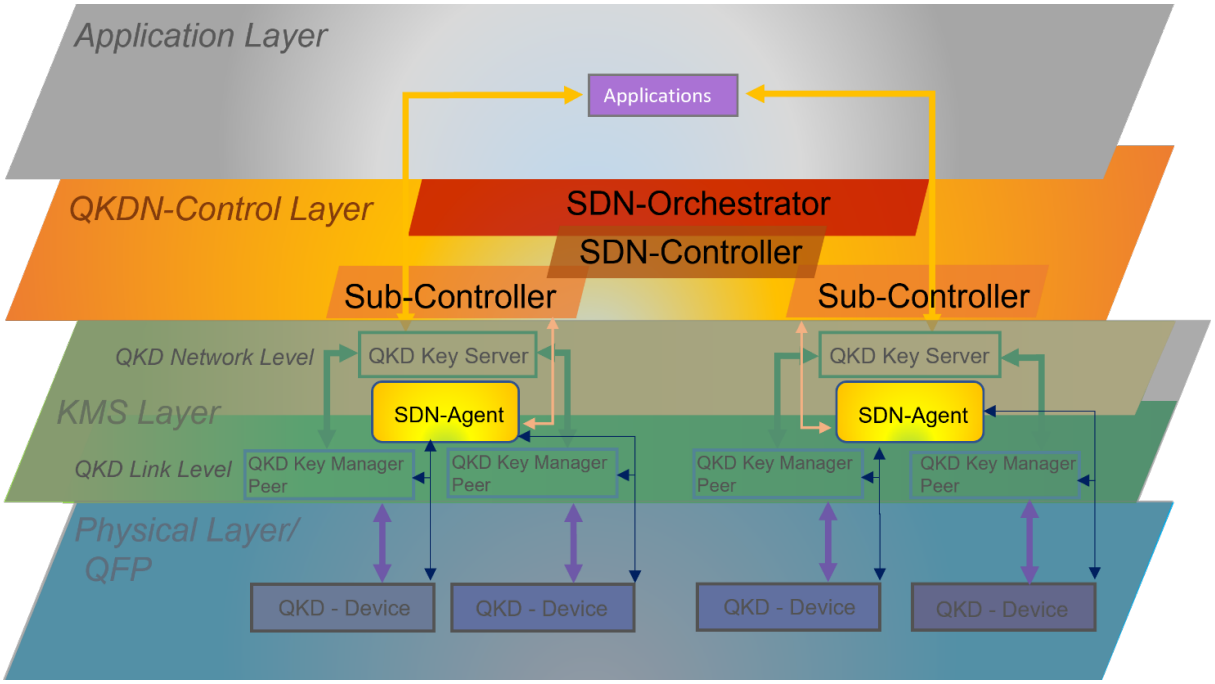

Abbildung 6: *Kontrollstrukturen bei ETSI*

SDN-Controller/Orchestratoren führen bei ETSI parallel auch Managementaufgaben aus, während bei ITU-T ein gesonderter QKDN Netzwerk Manager existiert, der speziell für das Fault / Configuration / Accounting / Performance / Security (FCAPS)-Management zuständig ist. Speziell für das QKD-Schlüsselmanagement relevant sind z.B. folgende Managementfunktionen: Key Supply Service Policy verwalten, mittels Log-Datenbank Schlüssellebenszyklus-Management zurückverfolgen oder Key Management Policies einhalten und an die SDN-Controller weiterleiten.

Die folgende ***Abbildung 7*** zeigt eine Übersicht der Kontrollstrukturen nach ITU-T. In der Abbildung ist zwar kein Hierarchiekonzept zu erkennen, aber es sind grundsätzlich mehrere Hierarchieebenen möglich, in der SDN-Controller verschiedene Sub-QKDNs miteinander verknüpfen. Der gesonderte Management Layer inkl. Cross-Layer Management kommuniziert über Control & Management-Submodulen mit allen Layern. Auch die Submodule untereinander können über die Schnittstellen *Ck/Cq/Kq-2* miteinander kommunizieren. Somit bilden sie ein Äquivalent zu den SDN-Agenten im jeweiligen Knoten von ETSI. Weitere Details zum Netzwerk Manager finden sich bei ITU-T Y.3804 und im Unterkapitel ***Vergleich mit ITU-T*** des folgenden Kapitels.

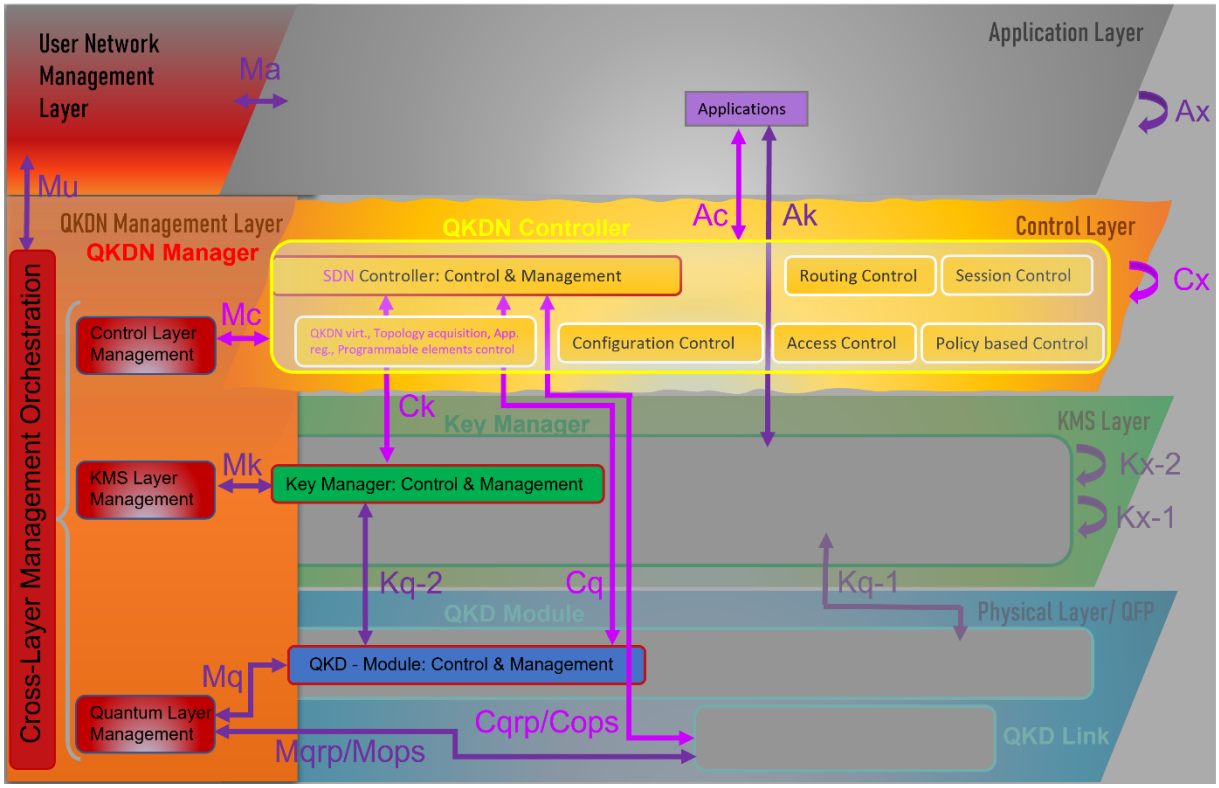

Abbildung 7: *Kontrollstrukturen mit SDN (pink) bei ITU-T Y.3805*

In der nachfolgenden ***Tabelle 1*** sind alle Kontrollsysteme mit ihren jeweiligen Aufgaben noch einmal zusammengestellt:

Key Management	Controller/Orchestrator	Netzwerk Manager
Schlüsselformatierung	Routingkontrolle zwischen Key Relays	Fehlermanagement
Schlüsselformat mit Metadaten erweitern (Key-ID, Datum, Länge, etc.)	Kontrolle der Kommunikation der Schichten bei Anfrage	Konfigurationsmanagement
Gewinnung von QKD Link Parametern (z.B. Durchsatz)	Kontrolle QKD & KMS (-Links)	Accountingmanagement
Abspeicherung in Puffern	Konfigurationskontrolle Rekonfiguration bei Scheitern	Performancemanagement
Key Relay zwischen KMSs	Authentifizierung/Autorisierung Zugriffskontrolle	Sicherheitsmanagement
Schlüsselsynchronisation	Sitzungskontrolle	Status Monitoring (QKD Module, QBER)
Schlüssellebenszyklus-Management	Policy-basierte Kontrolle	Unterstützt KMS beim Schlüssellebenszyklus-Management & Controller beim Routing
Schlüsselauslieferung an Applikation	FCAPS-Management QoS/Charging	Management Authentifizierung/Autorisierung
Schlüsselauthentifizierung mittels KMS Links	**Nur Controller:** Bereitstellung Netzwerktopologie/-parameter abstrahiert für Orchestrator	Management von QoS/Charging
	Nur Orchestrator: Koordination/Optimierung der SDN-Controller unterschiedlicher Domänen; Unterstützung Routing domänenübergreifend; Topologieerkundung/Service-bereitstellung domänenübergreifend; Monitoring Kontrollparameter	

Tabelle 1: *Übersicht Zuständigkeiten Netzwerkmanagementsysteme verallgemeinernd nach ITU-T Y.3800, ETSI 004/014/015/018*

3. Interoperabilität & Standardisierungen

Bei KMS-Systemen ist Interoperabilität ein wichtiges Thema, um verschiedene QKD-Komponenten in der Praxis mit vorhandener Hardware in einem Netzwerk kombinieren zu können.

Doch gerade im Bereich vom Netzwerkmanagement ist die Standardisierung noch nicht so weit fortgeschritten, wie beispielsweise bei der Schlüsselübergabe an die Applikation.

Auch wenn bereits einige Bemühungen laufen, die Schnittstellen zunehmend zu standardisieren, befinden sich noch viele in Aktualisierung (z.B. ETSI GS QKD 014) oder im Entwurf. Darunter fällt auch der Standard, der die Kommunikation verschiedener KMS-Hersteller untereinander an einem vertrauenswürdigen Knoten ermöglichen soll (ETSI GS QKD 020). Erst wenn alle laufenden Standardisierungen abgeschlossen sind, werden die Hersteller ihre Geräte entsprechend gestalten können und herstellerneutrale QKD-Netzwerke zunehmend einfacher aufzubauen sein. Dabei sind einige der großen Hersteller (z.B. IDQ, Toshiba an ETSI) auch an der Standardisierung selbst beteiligt.

Eine Übersicht der Definition verschiedener möglicher Interfaces in einem typischen QKD Netzwerk – inklusive der entsprechend existierenden ETSI-Standardisierungen und des geplanten ETSI GS QKD 020 – ist ***Abbildung 8*** zu entnehmen:

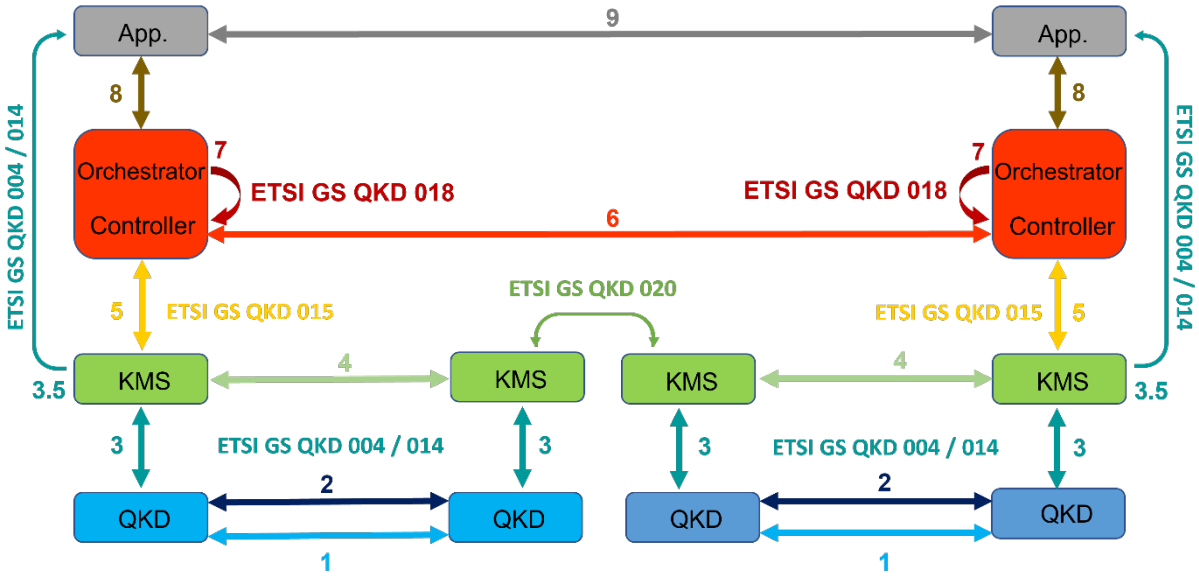

Abbildung 8: *Interfaces und ETSI-Standards nach* [1], [11]

Es gibt sowohl vertikale Schnittstellen (APIs zwischen höherer und niedriger Instanz), als auch horizontale zwischen sog. Peer-to-Peer-Entities auf gleicher Ebene.

In ***Tabelle 2*** werden die verschiedenen Interfaces und Protokolle von ***Abbildung 8*** kurz vorgestellt und erläutert [11]:

1	Quantum Level Communication Protokoll: Übertragung von QuBits auf Quantenkanal
2	QKD Post Processing Protokoll: Sifting, Fehlerschätzung & -korrektur, Privacy Amplification auf klassischem Kanal
3	QKD API zur Schlüsselabgabe an das KMS: ETSI GS QKD 004/014, proprietäre Protokolle (Cisco SKIP)
4	Austausch von Schlüsselmaterial zwischen verschiedenen KMSs beliebiger Hersteller: proprietäre Protokolle (Quantum Point-to-Point Protokoll aus SECOQC [12]), ETSI GS QKD 020
5	Kommunikation KMS mit SDN-Controller zur Verbindungsherstellung: ETSI GS QKD 015
6	Routing der Schlüssel zwischen den verschiedenen, nicht benachbarten KMSs
7	ETSI GS QKD 018 für den Austausch zwischen SDN-Controller & SDN-Orchestrator
8	API zwischen SDN-Orchestrator und Applikation zum Aufbau/Konfiguration der Ende-zu-Ende-Applikationsverbindungen durch das Netzwerk (z.B. QoS)
3.5	API Verbindung zwischen KMS und Applikation: ETSI GS QKD 004/014, proprietäre Protokolle (Cisco SKIP)
9	Applikationsspezifischer Austausch von Schlüsseln

Tabelle 2: *Schnittstellen SDN QKDN im Überblick*

ETSI-Standards

Es folgt eine Übersicht über die wichtigsten Standards für QKDN in Bezug auf KMS und SDN (vgl. [6]):

ETSI GS QKD 004 [13]

Hier wird mit dem *push/pull based operation mode* sowohl eine North-Bound API zwischen KMS und Applikation definiert, als auch ein South-Bound Interface, das dem KMS erlaubt mit der QKD-Schicht zu reden. Somit ergeben sich zwei vertikale Untergruppen innerhalb des KMS: das QKD Key Manager Peer und ein QKD Key Server (vgl. Kapitel ***Das Key Management System (KMS) im QKDN***). KMS ermöglichen den horizontalen Austausch zwischen Sende-/Empfangsknoten innerhalb der lokalen Sicherheitsgrenzen. Schlüsselanforderungen mit dem Parameter einer vordefinierten QoS oder die Anforderung eines Key-Stream-ID Parameters vom KMS sind möglich. Es soll eine Möglichkeit geben, Metadaten mitzuübertragen. Zudem hat jeder Key-Stream eine Time To Live (TTL).

ETSI GS QKD 014 [14]

Ähnlich zum Vorgänger wird eine vergleichsweise vereinfachte API zwischen KMS (hier: Key Management Entity (KME)) und Application Layer (hier: Secure Application Entity (SAE)) definiert, dem REST Architekturprinzip folgend. Die Kommunikation basiert auf dem HTTPS Protokoll mit TLS 1.2 oder höher. Hier werden ausschließlich Key-IDs und keine Key-Stream-IDs unterstützt (vgl. Kapitel ***Das Key Management System (KMS) im QKDN***).

ETSI GS QKD 015 [15]

Dieser Standard beschreibt die Schnittstelle zwischen einem SDN-Controller und dem KMS. Durch die Kommunikation des SDN-Controllers über einen SDN-Agenten im QKD-Knoten wird die kombinierte Ansteuerung des QKD-Moduls und des KMS durch den SDN-Controller ermöglicht (vgl. Kapitel ***Kontrollsysteme im (QKD)-Netz***). Der Agent soll Informationen zur Registrierung und Optimierung dem Controller bereitstellen, sowie Informationen über die Links zu anderen QKD-Systemen. Dabei soll die Information für den Controller möglichst abstrahiert sein. Dazu wird die Datenmodellierungssprache YANG verwendet. Die Bestandteile des QKD-Knoten sind hierfür in vier Gruppen unterteilt: Die Parameter für den QKD-Knoten, die QKD-Schnittstellen, den Link zur QKD-Schlüsselzuordnung (direkt/virtuell) und die QKD-Applikation (external/ internal). Unter externen Applikationen versteht man z.B. eine Endnutzer-Applikation und unter internen Applikationen Authentifizierungsschlüssel. Der SDN-Controller übernimmt außerdem das Applikationsmanagement (Registrierung/QoS, Peer-Knoten finden), damit die Applikation dies nicht verwalten muss.

ETSI GS QKD 018 [16]

Dies ist ein Standard speziell für die Orchestrierung mit SDN. Ein SDN-Orchestrator wird dazu verwendet, um über die unterschiedlichen von SDN-Controllern gesteuerten Domänen hinweg, das Netzwerk übergreifend zu steuern (vgl. Kapitel ***Kontrollsysteme im (QKD)-Netz***). Dazu muss die generelle Kommunikation zwischen SDN-Controller und SDN-Orchestrator definiert werden. Zunächst wird die Topologie eines SDN-kontrollierten Netzwerks mit all seinen Knoten vom SDN-Orchestrator entweder proaktiv bei jeder Pfadneuberechnung oder reaktiv bei Änderung durch die Controller bestimmt. Dabei wird keine Information über die Applikationen, die diese Schlüssel konsumieren, erhoben, weshalb noch Service-Links bestimmt werden müssen. Anschließend werden Monitoring-Parameter definiert, um die Knoten, die Links und damit den Netzwerkstatus zu beschreiben. So kann eine Ende-zu-Ende Servicebereitstellung durch den SDN-Orchestrator ermöglicht werden. Außerdem kann der Orchestrator von den Netzwerkkomponenten Benachrichtigungen (Ereignissen, Alarme) – einschließlich derer an den SDN-Controller – erhalten. Die Schnittstelle besteht immer aus dem Data Model und dem Transport Protocol. Ersteres beschreibt die Sprache mittels derer die Daten ausgetauscht werden, wofür auch hier YANG verwendet werden soll. Letzteres definiert die Kommunikationsregeln über Protokolle wie NETCONF und RESTCONF. REST-Protokolle basieren auf HTTP und unterstützen als eine Untermenge von NETCONF deren Funktionalität, welche die YANG Daten auslesen können.

ETSI GS QKD 020 (im Entwurf) [17]

ETSI GS QKD 020 wird die Interoperabilität verschiedener KMSs von unterschiedlichen Herstellern sicherstellen, damit vermehrt herstellerneutrale Geräte eingesetzt werden können. Er soll Ende des Jahres 2024 veröffentlicht werden. Allerdings soll ETSI GS QKD 020 nur für zwei KMSs innerhalb eines Knotens gelten. Somit fehlt immer noch eine Standardisierung eines KMS-Interfaces zwischen beliebigen Knoten.

Verwandt dazu wird es einen weiteren Standard ETSI GS QKD 021 geben, der das Orchestration-Interface speziell für diese interoperablen KMSs in Multi-Domain QKDNs ermöglicht.

Im spanischen Testbed MadQCI ([4], ***Abbildung 9***) wurde allerdings gezeigt, dass diese Funktionalität leicht mittels SDN-Modulen, die Informationen über alle Netzwerkgeräte (einschließlich dem Schlüsselbedarf der KMSs) verfügen, realisiert werden kann: Der Einsatz von SDN kommt in Verbindung mit einem lokalen *SDN-Agenten* zustande, der zusätzlich zum *lokalen KMS (LKMS)* am Knoten sitzt und mit dem der gesamte restliche Knoten gesteuert werden kann. Damit können Ende-zu-Ende-Schlüsseltransporte herstellerunabhängig zwischen beliebigen Knoten im Netzwerk realisiert werden. Zusätzlich wird ein extra *Forwarding-Module* in der untersten Schicht für das Routing und den damit verbundenen Schlüsseltransport eingesetzt. In das Routing können dadurch einfach QoS-Parameter integriert werden. Somit agiert das LKMS nur noch als verwaltender Schlüsselpuffer, der dem SDN Informationen über die Schlüssel liefert und anschließend die Schlüssel an die Applikation weitergibt.

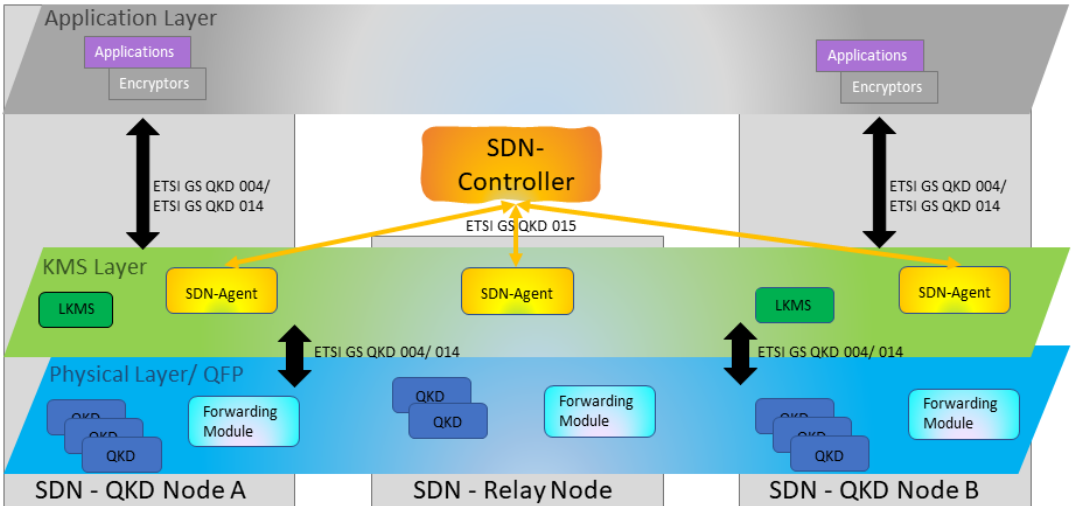

Abbildung 9: *Interfaces und Standards mit SDN-Controller bei MadQCI [4]*

Aus den bisher vorgestellten Komponenten des KMS und Control Layers von ETSI, ergibt sich entsprechend ein allgemeines Schichtenmodell, wie in ***Abbildung 10*** im Ganzen zu sehen ist:

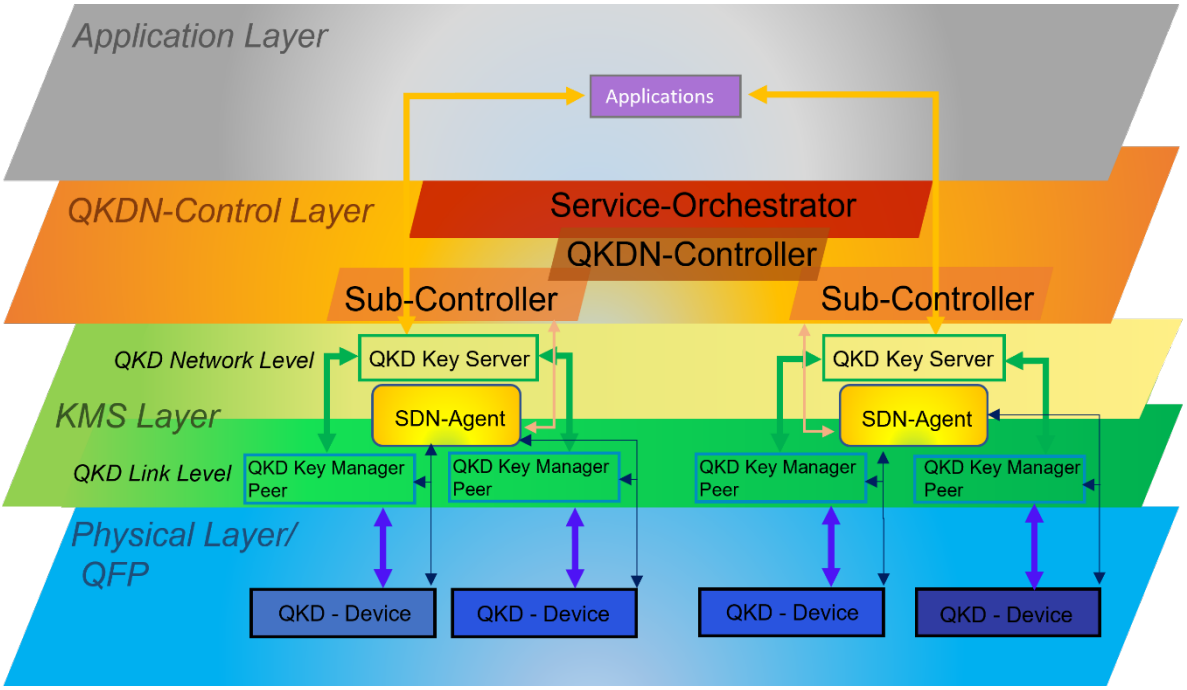

Abbildung 10: *QKDN gesamt nach ETSI*

Vergleich mit ITU-T

Auch ITU-T widmet sich in seiner Recommendation-Suite Y.3800 ff dem Thema der Quantenschlüsselverteilung und dem Management bei QKDN:

***ITU-T Y.3800*: Überblick QKDN** [18]

Hier werden die Grundlagen für ein QKDN-Netzwerk definiert. Ein QKDN muss u.a. die Fähigkeit besitzen, nicht nur Point-to-Point Schlüssel, sondern auch gemeinsame Multi-User/Point Schlüssel in einem bestimmten Format unter speziellen Sicherheitsanforderungen an die anfragende Applikation zu verteilen und diese unter Einhaltung von speziellen QoS-Anforderungen und von gewissen Privacy-Policies zu kontrollieren/verwalten. Auch hier sind die Links zwischen den Key-Management-Systemen, die sich lokal am selben Knoten befinden, häufig durch QKDN-Controller unterstützt und senden entsprechende Parameter an diese. Der Controller ist mit Tätigkeiten, wie dem Key-Relay-Routing und der Kontrolle von QKD & Key Management Links, sowie mit Authentifizierungskontrolle und QoS vertraut. Darüber hinaus gibt es einen QKDN-Manager, der für das FCAPS-Management zuständig ist (vgl. Kapitel ***Kontrollsysteme im (QKD)-Netz***). In den QKDN-Knoten herrschen strikte Key Management Policies für die Schlüssellebensdauern (vgl. Kapitel ***Das Key Management System (KMS) im QKDN***), sowie bzgl. der Unabhängigkeit von den Applikationen; d. h. die Schlüssel werden nur über Key-IDs unterschieden und es liegt eine gegenseitige Sicherheitsabgrenzung zwischen der Applikationsebene/ dem Nutzernetzwerk und dem QKDN vor, sodass die Applikationen keine Kenntnis über die darunterliegenden QKDN-Prozesse benötigen. Umgekehrt benötigt das QKDN auch keine Informationen darüber, wie die Schlüssel von der Applikation eingesetzt werden und beschränkt sich auf die Kenntnis der Schlüssellänge und der zugehörigen Applikations-ID.

***ITU-T Y.3803*: Key Management** [2]

Die ITU-T-Empfehlung für KMS differenziert zwischen Key Management Agent (KMA) und Key Supply Agent (KSA). Der KMA ist mit den QKD-nahen Tätigkeiten, wie Empfangen, Speichern, Weiterleiten, Managen und Verwerfen gemäß Policies während des Life Cycles der Schlüssel beauftragt; dagegen ist der KSA für die Key Requests, sowie die Übersendung der KSA-Key-(ID)s an die Applikationsschicht und der Metadaten an den QKDN-Manager zuständig (siehe ***Abbildung 11***, Schnittstelle *Ak*). Es ist eine gegenseitige Authentifikation zwischen KMA und KSA nötig, bevor ein Key Request an den KMA weitergegeben wird. Zudem werden verschiedene *Key-Relay* Ansätze vorgestellt: Falls es keinen direkten KMS-Link zwischen den KMSs gibt, wird eine Route zwischen den Relays vom QKDN-Controller angefordert und die Schlüssel entsprechend mittels XOR-Verknüpfungen an jedem Knoten über Relay-Knoten zum Zielknoten geroutet. (Vgl. Kapitel ***Das Key Management System (KMS) im QKDN***)

ITU-T Y.3804: **Kontrolle und Management von QKDN** [9]

In dieser Richtlinie werden die Kontrolle und das Management im QKDN-Kontext näher definiert. Während die grundsätzlichen Funktionen bereits in Y.3800 beschrieben sind, wird hier vertiefter auf Routing, Sitzungskontrolle, QoS und FCAPS-Managementfunktionen eingegangen. Zudem werden Referenzpunkte/Schnittstellen zwischen den Kontroll- & Managementeinheiten beschrieben, ebenso wie deren Orchestrierung in den verschiedenen Schichten und deren Zusammenarbeit mit externen Managementsystemen (z.B. User Network Management System). Jede Schicht weist eine eigene Kontroll- & Managementfunktion auf, die mit dem QKDN Management Layer korrespondiert. Der QKDN-Controller kontrolliert den Physical Layer/ Quantum Layer und den Key Management Layer (geht aber nicht mit den Schlüsseln selbst um). Zusätzlich unterstützt dieser den QKDN Management Layer und den Application/Service Layer. In ***Abbildung 11*** sind die zum Controller zugehörigen Referenzpunkte veranschaulicht*: Ck, Cq, Cqrp (Quantum Relay Point),* und *Cops (Optical Splittting)*, während *Cx* für die Kommunikation zwischen den verschiedenen Controllern zuständig ist. *Mc* dient als Schnittstelle zum QKDN-Manager und unterstützt somit FCAPS. Der QKDN Manager besteht aus Quantum-, Key Management-, Control- Layer Management und einem diese drei Module übergreifenden „Cross-Layer Management Orchestration“ Modul. Der QKDN Manager verwaltet das gesamte QKDN (FCAPS), hat komplette Kenntnis der QKDN-Topologie und regelt, z.B. bei Ausfall von QKD-Links, das Fault-Management. Über die *Mu*-Schnittstelle tauscht sich der QKDN Manager mit dem User Network Management aus. (Vgl. Kapitel ***Kontrollsysteme im (QKD)-Netz***)

ITU-T Y.3805: **SDN-basierte Kontrolle** [10]

In diesem Standard wird auf die hierarchische Strukturierung von SDN-Controllern eingegangen (vgl. Kapitel ***Kontrollsysteme im (QKD)-Netz***). Außerdem wird ein generelles Prozedere für die SDN Kontrolle in QKDNs einge-führt, bestehend aus verschiedenen Phasen:

- „Service Request System Initialization Phase“
- „Key Generation Phase“
- „Key Request, Relay and Supply Phase“
- „Management Monitor Phase“
- „QKDN Virtualization Phase“

Verglichen zu Y.3804 kommen hier speziell die Kontrolle programmierbarer Elemente, die Applikationsregistrierung, die Topologie-Akquirierung von untergeordneten Controllern und die Virtualisierung des QKDN hinzu; darüber hinaus muss auch die Kommunikation zwischen den Controllern ermöglicht werden. Die Schnittstelle *Ac* aus ***Abbildung 11*** wurde hier nachträglich hinzugefügt, um eine Kommunikation zwischen dem SDN-Controller und der Applikation zu ermöglichen.

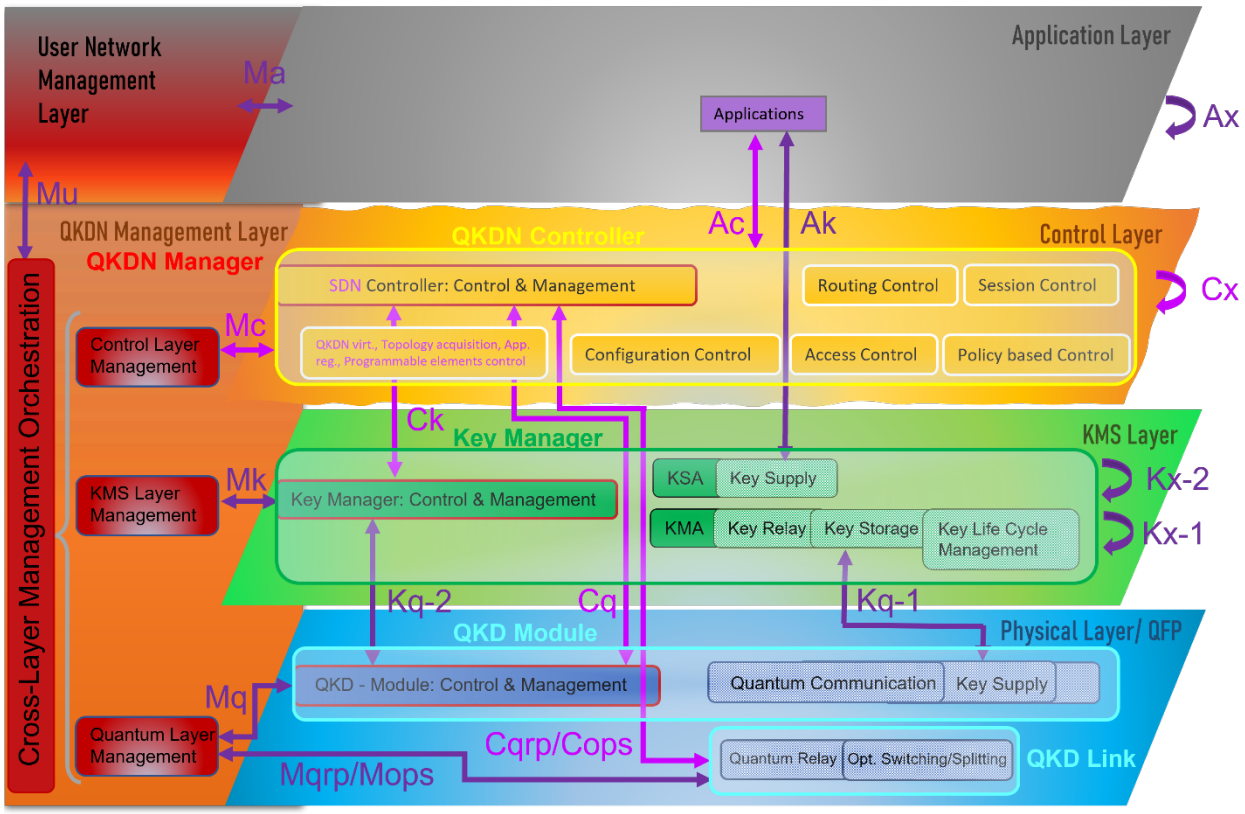

Abbildung 11: *Die verschiedenen Layer eines QKDN mit Schnittstellen und Funktionen bei ITU-T Y.3804/5*

Wesentliche Unterschiede in der Beschreibung eines QKDN bei ITU-T gegenüber ETSI sind:

- Die ITU sieht eine Cross-Layer Management Orchestration vor mit parallelen Managementfunktionen in den einzelnen Layern. ETSI beschreibt dagegen übergeordnete SDN-Controller/Orchestrator mit integrierter Monitoring-Managementfunktion.
- Der QKDN Manager bei ITU koordiniert auch direkt einzeln den QKD- & KMS- (& Control-) Layer, welche bei ETSI gemeinsam durch den SDN-Agenten gesteuert werden.
- Die direkte Schnittstelle zwischen SDN-Controller und Applikation (*Ac*) existiert bisher nur bei ITU.

4. Key Monitoring Parameter

Es gibt eine Vielzahl an Parametern in einem QKDN, die überwacht werden können bzw. müssen. An den verschiedenen Schnittstellen sollte eine Mitübertragung verschiedener Key Monitoring Parameter ermöglicht werden, um eine automatische Überwachung zur Sicherheit, Stabilität und Optimierung des Netzwerkes realisieren zu können. Darunter fallen im Bereich KMS Parameter die Füllstände der Schlüsselpuffer, die Füllraten, die durchschnittliche angeforderte Schlüsselrate (Secret Key Rate (SKR)) und das Key Recycling Intervall nach Ablauf der Time to Live (TTL) des Schlüssels [13], [19]. Weitere Parameter, die im Zusammenhang mit dem Einsatz von SDN in einem Testbed ermittelt werden können, sind die Latenz der Schlüsselverteilung, der Erfolg der Schlüsselverteilung und der durchschnittliche Schlüsselverbrauch, wie es z.B. in [20] gezeigt wurde.

Aktuell gibt es im Vorfeld häufig keine konkreten Herstellerinformationen darüber, welche Parameter von QKD-Geräten und KMSs ermittelt und wie sie bereitgestellt werden. Folgende Parameter im Zusammenhang mit KMS und SDN wären von besonderem Interesse in einem QKDN zu erfassen:

- Durchschnittlich benötigter Schlüsselvorrat in Bezug auf die Größe der Puffer unter Berücksichtigung der Füllraten:
    - Festlegung von Schwellwerten ab denen ein Alarm geworfen werden soll, wenn Stand zu niedrig -> Neuproduktion von Schlüsseln
    - Definition eines Ausfallszenarios, z.B. Bestimmen der unbedingt zu verschlüsselnden Daten oder Ausweichen auf andere Methoden zur Schlüsselgenerierung (z.B. PQC)
- Benötigte SKR je nach Bedarf im Netzwerk (Applikations- & zeitabhängig)
- Key Rotation Interval (TTL des Schlüssels bei der Applikation [13])

  In [21] wurden verschiedene Key Rotation Intervalle von 1, 5, 15 und 60 min mit TTL = 4h und Puffergröße = 1000 Schlüssel getestet. Im Intervall von 1 Minute beträgt die Erfolgsrate der Quantum Key Rotation 89%.
- Erfassung der Key Request Statistics (z.B. Anzahl und Länge der Schlüssel, Link Status) für die Optimierung durch das SDN:

  Nach ***ETSI GS QKD 015*** sollen die Knoten dem Controller Informationen bereitstellen; u.a. für die Registrierung und Optimierung des Schlüsselmanagements
- Zusätzliche nützliche Parameter:
    - Latenz, Zeitsynchronisation und Zeitbeschränkungen
    - Key-ID zur Identifikation nicht immer standardmäßig im Protokoll mitübertragbar (-> ***ETSI GS QKD 014*** oder ***ITU-T Y.3803*** unterstützen) [22]
    - Einführen weiterer vom Nutzer aktiv und individuell definierbarer QoS-Parameter (über ***ETSI GS QKD 004*** hinaus) nötig (z.B. nur Schlüssel von/über bestimmte KMS-Standorte bestimmter Hersteller [1]).

Zusammenfassung

In diesem Dokument sind die aktuelle Situation in Bezug auf QKD-Schlüsselverwaltung und die dafür notwendigen Kontrollsysteme in QKDNs dargestellt. Das Dokument hat zu diesem Zweck die Funktionszuweisungen der KMS- und SDN-Komponenten präsentiert und diese vereinheitlichend zusammengeführt, um zukünftig den zielführenden Einsatz der sicher-erzeugten QKD-Schlüssel im Netzwerk zu erleichtern.

Dafür wurden auch die bereits existierenden Standards von ETSI und ITU im Zusammenhang mit den beiden Komponenten vorgestellt. Allerdings fällt auf, dass die Ansätze von ETSI und ITU sich an diesem Verbindungsglied zwischen Schlüsselgenerierung und sicherem Schlüsseleinsatz unterscheiden und mitunter Fragen bezüglich der Realisierung eines funktionierenden Quantenkommunikations-netzwerkes offenlassen. Es müssen noch einige wichtige Funktionen in den Standards im Bereich der Netzwerkmanagementsysteme aber auch des KMS erweitert werden oder fehlen sogar, wie beispielsweise:

- ein KMS-zu-KMS Interface verschiedener Hersteller zwischen verschiedenen Knoten (ohne SDN)
- die Möglichkeit der direkten Kommunikation SDN-Controller mit Applikation (bei ETSI)
- klare Abgrenzung der Zuständigkeiten der Control- & Managementabläufe (zukünftig kombiniert in SDN)
- insbesondere muss geklärt werden, ob die angedachte Hybridisierung der verschiedenen Schlüssel (Verschlüsselung des QKD-Schlüssels mit PQC & Detektion des Leerlaufens durch eine Denial of Service-Attacke) Teil des KMSs ist [23].

Zudem ist es häufig schwierig, zusätzliche (Monitoring)-Parameter (z.B. Key-ID, Time-Constraints), die nicht in allen Standards vorgesehen sind und nicht bei allen Herstellern unterstützt werden, mit zu übertragen.

Es gilt daher, die weiteren Entwicklungen in diesem sehr aktiven Bereich genau zu verfolgen. Da insbesondere das Zusammenspiel der verschiedenen Komponenten schwierig theoretisch zu untersuchen ist, werden praktische Untersuchungen in Testbeds mit vielen verschiedenen Herstellern und Komponenten, wie beispielsweise bei MadQCI sehr wichtig sein, um die Schlüssel in einem QKD-Netzwerk nach der Erzeugung auch effizient und sicher nutzen zu können und den praktischen Einsatz von QKD Systemen somit zu ermöglichen.

Literaturverzeichnis

[1] „Quantenverschlüsselte Kommunikation ohne Grenzen“, Okt. 2021, Zugegriffen: 28. Juni 2024. [Online]. Verfügbar unter: https://www.stmd.bayern.de/wp-content/uploads/2022/03/Bayern-Oesterreich-Studie-Quantenverschluesselte-Kommunikation.pdf

[2] *Y.3803 : Quantum key distribution networks - Key management*. Zugegriffen: 27. Juni 2024. [Online]. Verfügbar unter: https://www.itu.int/rec/T-REC-Y.3803-202012-I/en

[3] C. Lee, Y. Kim, K. Shim, und W. Lee, „Key-count differential-based proactive key relay algorithm for scalable quantum-secured networking“, *J. Opt. Commun. Netw., JOCN*, Bd. 15, Nr. 5, S. 282–293, Mai 2023, doi: 10.1364/JOCN.478620.

[4] V. Martin *u. a.*, „MadQCI: a heterogeneous and scalable SDN QKD network deployed in production facilities.“, 2023, doi: https://doi.org/10.48550/arXiv.2311.12791.

[5] T. Choi, S. Yoon, T. Y. Kim, und H. Kim, „Design and Implementation of Quantum Key Distribution Network Control and Management“, in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, Okt. 2021, S. 724–727. doi: 10.1109/ICTC52510.2021.9621170.

[6] P. James, S. Laschet, S. Ramacher, und L. Torresetti, „Key Management Systems for Large-Scale Quantum Key Distribution Networks“, in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, Benevento Italy: ACM, Aug. 2023, S. 1–9. doi: 10.1145/3600160.3605050.

[7] M. Mehic, S. Rass, E. Dervisevic, und M. Voznak, „Tackling Denial of Service Attacks on Key Management in Software-Defined Quantum Key Distribution Networks“, *IEEE Access*, Bd. 10, S. 110512–110520, 2022, doi: 10.1109/ACCESS.2022.3214511.

[8] E. Dervisevic *u. a.*, „Simulations of Denial of Service Attacks in Quantum Key Distribution Networks“, in *2022 XXVIII International Conference on Information, Communication and Automation Technologies (ICAT)*, Juni 2022, S. 1–5. doi: 10.1109/ICAT54566.2022.9811238.

[9] *Y.3804 : Quantum key distribution networks - Control and management*. Zugegriffen: 22. Juli 2024. [Online]. Verfügbar unter: https://www.itu.int/rec/T-REC-Y.3804/en

[10] *Y.3805 : Quantum key distribution networks - Software-defined networking control*. Zugegriffen: 22. Juli 2024. [Online]. Verfügbar unter: https://www.itu.int/rec/T-REC-Y.3805/en

[11] O. Maurhart, T. Länger, A. Poppe, C. Pacher, M. Stierle, und H. Leopold, „Standardization and Certification of QKD-Devices and QKD Networks“, Zugegriffen: 27. Juni 2024. [Online]. Verfügbar unter: https://2020.qcrypt.net/posters/QCrypt2020Poster133Maurhart.pdf

[12] M. Peev *u. a.*, „The SECOQC quantum key distribution network in Vienna“, *New J. Phys.*, Bd. 11, Nr. 7, S. 075001, Juli 2009, doi: 10.1088/1367-2630/11/7/075001.

[13] *ETSI GS QKD 004 Quantum Key Distribution (QKD); Application Interface*, August 2020. [Online]. Verfügbar unter: https://www.etsi.org/deliver/etsi_gs/QKD/001_099/004/02.01.01_60/gs_QKD004v020101p.pdf

[14] *ETSI GS QKD 014 Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API*, Februar 2019. [Online]. Verfügbar unter: https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_QKD014v010101p.pdf

[15] *ETSI GS QKD 015 Quantum Key Distribution (QKD); Control Interface for Software Defined Networks*, April 2022. [Online]. Verfügbar unter: https://www.etsi.org/deliver/etsi_gs/QKD/001_099/015/02.01.01_60/gs_QKD015v020101p.pdf

[16] *ETSI GS QKD 018 Quantum Key Distribution (QKD); Orchestration Interface for Software Defined Networks*, April 2022. [Online]. Verfügbar unter: https://www.etsi.org/deliver/etsi_gs/QKD/001_099/018/01.01.01_60/gs_QKD018v010101p.pdfhttps://www.etsi.org/deliver/etsi_gs/QKD/001_099/018/01.01.01_60/gs_QKD018v010101p.pdf

[17] *ETSI DGS QKD 020 InteropKMS*, 1. Dezember 2024. [Online]. Verfügbar unter: https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=63115

[18] *Y.3800 : Overview on networks supporting quantum key distribution*. Zugegriffen: 22. Juli 2024. [Online]. Verfügbar unter: https://www.itu.int/rec/T-REC-Y.3800/en
[19] G. Pmo und P. Hasleham, „QKD Concepts and Considerations“, Zugegriffen: 28. Juni 2024. [Online]. Verfügbar unter: https://resources.geant.org/wp-content/uploads/2024/02/GN5-1_White-Paper_QKD-Concepts-and-Considerations.pdf
[20] Y. Wang, X. Yu, Z. Wang, Y. Cao, Y. Zhao, und J. Zhang, „Demonstration of Hierarchical SDN Orchestration for End-to-End Key Provisioning in Large-Scale Quantum Key Distribution Networks“, in *2023 21st International Conference on Optical Communications and Networks (ICOCN)*, Qufu, China: IEEE, Juli 2023, S. 1–4. doi: 10.1109/ICOCN59242.2023.10236303.
[21] N. Makris *u. a.*, „Field demonstration of a fully managed, L1 encrypted 3- node network with hybrid relayed-QKD and centralized symmetric classical key management“, [Online]. Verfügbar unter: https://arxiv.org/pdf/2403.08526
[22] „Validation of a Quantum Safe MACsec Implementation.pdf“. Zugegriffen: 4. Juni 2024. [Online]. Verfügbar unter: https://www.juniper.net/content/dam/www/assets/white-papers/us/en/2022/validation-of-quantum-safe-macsec-white-paper.pdf
[23] „EuroQCI ConOps (Concept of Operations) | Shaping Europe’s digital future“. Zugegriffen: 4. Dezember 2024. [Online]. Verfügbar unter: https://digital-strategy.ec.europa.eu/en/miscellaneous/euroqci-conops-concept-operations

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

Zusammenstellung der Referenzen zu QKDN-Standards von KMS & SDN

ETSI:

[A1]
https://www.etsi.org/deliver/etsi_gs/QKD/001_099/004/02.01.01_60/gs_QKD004v020101p.pdf
[A2]
https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_QKD014v010101p.pdf
[A3]
https://www.etsi.org/deliver/etsi_gs/QKD/001_099/015/02.01.01_60/gs_QKD015v020101p.pdf
[A4]
https://www.etsi.org/deliver/etsi_gs/QKD/001_099/018/01.01.01_60/gs_QKD018v010101p.pdf

ITU-T:

[A5]
https://www.itu.int/rec/T-REC-Y.3800/en
[A6]
https://www.itu.int/rec/T-REC-Y.3803-202012-I/en
[A7]
https://www.itu.int/rec/T-REC-Y.3804/en
[A8]
https://www.itu.int/rec/T-REC-Y.3805/en

Sekundärliteratur:

[B1] Key Management Systems for Large-Scale Quantum Key Distribution Networks
https://dl.acm.org/doi/10.1145/3600160.3605050
[B2] Interoperable KMS
https://www.mdpi.com/1099-4300/25/6/943
[B3] Standardization
https://2020.qcrypt.net/posters/QCrypt2020Poster133Maurhart.pdf
[B4] Standardization progress
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8596065