Quantensichere VPN-Infrastrukturen

David Schatz[1], Friedrich Altheide[2],
Prof. Dr.-Ing. Günter Schäfer[1], Dr. Kai Martius[2]

Kurzfassung:

Für die in den kommenden Jahren erforderliche Umstellung zahlreicher „Virtuelle Private Netz (VPN)"-Infrastrukturen auf quantensichere kryptografische Verfahren stehen grundsätzlich drei alternative technologische Ansätze zur Verfügung: Post Quantum Cryptography (PQC), Quantum Key Distribution (QKD) und klassische symmetrische Kryptografie mit ausreichend langen Schlüsseln (≥ 256 Bit). Da es im Gegensatz zur symmetrischen Kryptografie bei den beiden erstgenannten Verfahrenskategorien aktuell noch nicht klar absehbar ist, welche der verfügbaren Kandidaten auch langfristig ein hinreichendes Sicherheitsniveau garantieren können, stellen wir in diesem Beitrag einen integrierten Ansatz zur sicheren Authentisierung und Schlüsselverwaltung in großen VPN-Infrastrukturen vor, der einen komplementären Einsatz aller drei Verfahrenskategorien ermöglicht und dabei Sicherheit in der Tiefe realisiert.

Stichworte: IKE, MKR, PQC, QKD, Quantensicherheit, VPN

1 Einleitung und Motivation

Seit den Veröffentlichungen von Shor [1] in der Mitte der 1990er Jahre sowie Proos und Zalka [2] wenige Jahre später ist bekannt, dass alle klassischen asymmetrischen Authentisierungs- und Schlüsselaustauschverfahren bei Verfügbarkeit praktisch einsetzbarer Quantencomputer unsicher werden. Aufgrund der Bedrohung, dass ein Angreifer bereits heute großflächig Daten abhören und speichern könnte, um sie in einigen Jahren zu entschlüsseln („store now, decrypt later"), besteht allerdings in zahlreichen Anwendungsszenarien bereits heute akuter Handlungsbedarf, die bestehenden asymmetrischen Verfahren zu ersetzen. Bei symmetrischen Verfahren konnte hingegen gezeigt werden [3], dass diese, Stand heute, ab einer Schlüssellänge von 256 Bit auch Angriffen mittels Quantencomputer standhalten, soweit keine anderweitigen Schwächen bekannt werden. Allerdings sind bei ihrem Einsatz für Authentisierung und Schlüsselverwaltung in der Praxis eine Reihe von Herausforderungen zu meistern. Diese umfassen insbesondere die sichere initiale Verteilung paarweiser Schlüssel, sowie ihre regelmäßige sichere Erneuerung.

[1] Technische Universität Ilmenau, Ilmenau
[2] secunet Security Networks AG, Essen

Dementsprechend wird seit Mitte der 1990er Jahre mit großer Intensität an neuartigen asymmetrischen Verfahren geforscht, welche ebenfalls resistent gegenüber Angriffen mittels Quantencomputern sind. Diese Verfahren werden unter dem Begriff Post Quantum Cryptography (PQC) zusammengefasst (siehe [4] für einen Überblick über den aktuellen Stand der Technologie). Allerdings sind die bereits länger bekannten Verfahren (etwa Classic McEliece) aufgrund des erforderlichen Rechenaufwandes und der Länge der Schlüssel sowie der Chiffren/Signaturen für praktische Anwendungen wenig geeignet. Bei den jüngeren und effizienteren Ansätzen gibt es hingegen noch Vorbehalte/Unsicherheit bezüglich ihrer langfristigen Sicherheitseigenschaften. So wurde beispielsweise das Verfahren Rainbow – immerhin Finalist im Standardisierungsverfahren des NIST für PQC-Verfahren – im Jahr 2022 erfolgreich auf einem Laptop gebrochen [5]. Auch wenn der sofortige Einsatz von PQC-Verfahren zwingend nötig ist, wird dementsprechend von nationalen und einigen internationalen Sicherheitsagenturen für den Moment ein „hybrider" Einsatz von PQC in Kombination mit klassischer asymmetrischer Kryptografie empfohlen (vergleiche auch [4]).

Als Alternative zu PQC wird zunehmend die beweisbar sichere Schlüsselverteilung unter Nutzung quantenmechanischer Effekte (QKD) [6] in Betracht gezogen. In der Praxis stellen sich jedoch eine ganze Reihe von Herausforderungen, die mit der aktuell begrenzten Reichweite (ca. 100 km per Glasfaser) sowie offenen Fragen zur Authentisierung und zur Sicherheitsevaluierung zusammenhängen. Dies führt dazu, dass diverse Sicherheitsagenturen QKD aktuell, wenn überhaupt, höchstens als ergänzenden Schutz *zusätzlich* zu PQC empfehlen [7].

Insgesamt ist derzeit allerdings nicht absehbar, unter welchen Voraussetzungen die von der NIST standardisierten PQC-Verfahren sicher eingesetzt werden können. Deshalb kann nicht ausgeschlossen werden, dass zukünftig die jeweils verwendeten PQC-Algorithmen kurzfristig, etwa nach Bekanntwerden von konzeptionellen Schwachstellen, während des Betriebs ausgetauscht werden müssen. Abhängig von der Verwendung der Algorithmen (etwa in Smartcards) oder von den gefundenen konzeptionellen Schwachstellen, wird ein Wechsel möglicherweise nicht immer sofort möglich sein.

Es stellt sich somit insgesamt die Frage, wie „Virtuelle Private Netz (VPN)"-Infrastrukturen während einer solchen Migration beziehungsweise am besten auch über die Migration hinaus, zusätzlich abgesichert werden können. Insbesondere wäre es wünschenswert, wenn durch diese zusätzliche Absicherung der Angreifer gezwungen werden würde, langfristig zu planen, zu agieren, und hierbei große Mengen an Ressourcen aufzuwenden. Diese Frage

wird im weiteren Verlauf am Beispiel der weitverbreiteten VPN-Protokollfamilie IPsec diskutiert. Die dargelegten Ideen können jedoch auch für andere VPN-Technologien angepasst werden. Im Folgenden werden zunächst Anforderungen an quantensichere VPN-Infrastrukturen aus praktischer und implementierungstechnischer Sicht erläutert und auf dieser Grundlage eine Strategie für ihre Realisierung hergeleitet. Diese zielt darauf ab, den erforderlichen Aufwand für einen großflächig agierenden Angreifer erheblich zu steigern und damit die „store now, decrypt later" Problematik im gesamten VPN nachhaltig zu lösen.

2 Anforderungen an quantensichere VPN-Infrastrukturen

Neben den traditionellen Anforderungen an VPN-Infrastrukturen, wie etwa eine Ende-zu-Ende-Sicherung zwischen Sicherheitsendpunkten, Skalierbarkeit mit der Netzgröße und Robustheit gegenüber Denial of Service (DoS)-Angriffen, hat eine Migration zu quantensicheren VPN-Infrastrukturen vor allem folgende Anforderungen:

- **Kryptoagilität**: Die eingesetzten kryptografischen Verfahren müssen während des Betriebs leicht austauschbar sein, um schnell auf Entwicklungen in der Kryptanalyse reagieren zu können.

- **Schlüsselquellenagilität**: Die Implementierung sollte möglichst einfach um weitere Quellen von symmetrischem Schlüsselmaterial (etwa QKD oder manuell ausgetauschte Schlüssel) erweitert werden können. Insbesondere sollte es möglich sein, Quellen zur Laufzeit dynamisch hinzu oder, etwa wegen Wartungsarbeiten, abschalten zu können.

- **Implementierungssicherheit**: Die Integration muss so erfolgen, dass die Implementierungssicherheit des Gesamtsystems gewährleistet ist. Weiterhin sollte eine Implementierung möglichst einfach gehalten werden (kleine Trusted Computing Base, kurz TCB).

3 Sicherheit in der Tiefe für das IKE-Protokoll

Ein konkreter und weit verbreiteter Anwendungsfall für VPN, welche mittels IPsec realisiert sind, ist es, mehrere Standorte über nicht vertrauenswürdige Netze (zum Beispiel das Internet) sicher zu verbinden (siehe Abbildung 1). Dabei tunneln VPN-Gateways sämtlichen Verkehr von Clients zwischen Standorten und sichern diesen kryptografisch (Verschlüsselung, Schutz der

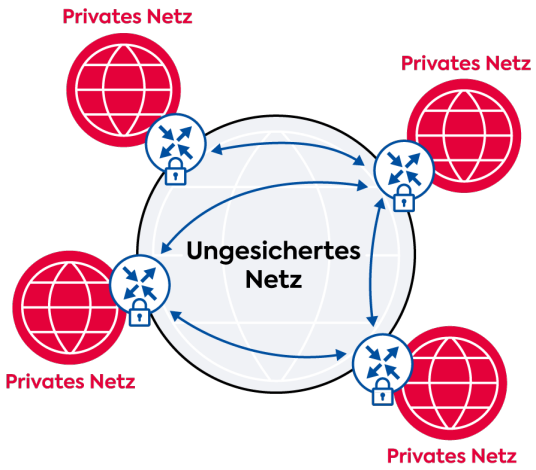

Abbildung 1: Einfache VPN-Infrastruktur mit einem VPN-Gateway und einem privaten Netz pro Standort

Integrität). Die Clients eines Standorts befinden sich dabei etwa in einem gemeinsamen, vertrauenswürdigen lokalen Netzwerk. Der authentisierte Austausch eines symmetrischen Sitzungsschlüssels zwischen zwei Gateways erfolgte bisher durch klassische asymmetrische Kryptografie. Dieser Schlüsselaustausch wird durch das IKE-Protokoll (Internet Key Exchange) durchgeführt. Nach dem erfolgten Austausch kann auf Basis des Sitzungsschlüssels sämtlicher Client-Verkehr zwischen den Gateways abgesichert werden.

Aufgrund ihrer konzeptionellen Ähnlichkeit zu klassischen Verfahren, konnten PQC-Verfahren bereits relativ einfach in die aktuelle Version des IKE-Protokolls (IKEv2) integriert werden, um in Kombination mit etablierten Verfahren verwendet zu werden. Beispielsweise kann durch die Erweiterung IKE_INTERMEDIATE [8] ein hybrider Schlüsselaustausch umgesetzt werden. Hierdurch sind ausgehandelte Sitzungsschlüssel und damit der Client-Verkehr so lange sicher, wie mindestens eines der eingesetzten asymmetrischen Verfahren sicher ist. Allerdings wird das „store now, decrypt later“ Problem nur gelöst, sofern sich die eingesetzten PQC-Verfahren als langfristig sicher herausstellen.

Eine orthogonale Idee ist es deshalb, den von IKE abgeleiteten Sitzungsschlüssel zusätzlich durch „out of band“ ausgetauschtes symmetrisches Schlüsselmaterial zu „verstärken“. Mögliche Quellen für solch zusätzliches Material sind unter anderem:

- **Pre-shared Keys (PSKs)**: VPN-Gateways könnten „per Hand“ mit zusätzlichem symmetrischem Schlüsselmaterial ausgestattet werden. Beispielsweise könnte auf jedem Gateway derselbe Gruppenschlüssel hinterlegt werden, stellenweise auch als „CUG-Schlüssel“ bekannt

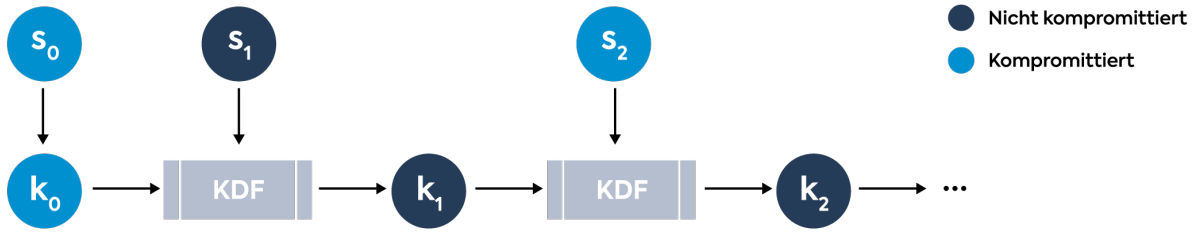

Abbildung 2: Schlüsselkette zur Kombination aller verfügbaren paarweiser symmetrischen Schlüssel s_i zu verschiedenen Master-Schlüsseln k_i über die Zeit. Im gezeigten Beispiel wird angenommen, dass der initial verwendete Schlüssel $k_0 = s_0$ dem Angreifer bekannt ist (kompromittiert). Die Synchronisation des sicheren Schlüssels s_1 sorgt dafür, dass der neue Master-Schlüssel k_1 dem Angreifer nicht länger bekannt ist. Dies ändert sich auch nicht durch die Synchronisation eines kompromittierten Schlüssels s_2.

(Closed User Group). Allerdings bietet ein Gruppenschlüssel keine Ende-zu-Ende-Sicherheit. Ziel sollte es daher sein, den Gateways *paarweise* symmetrische Schlüssel zur Verfügung zu stellen. Dies könnte etwa durch Mitarbeiter auf Dienstreisen oder durch den Versand von Smartcards mit Schlüsselmaterial auf dem Postweg realisiert werden.

- **„Business Trip Key Exchange“**: Anstatt händisch PSKs zu verteilen, könnten (abgesicherte) Laptops von Mitarbeitern verwendet werden, um bei Dienstreisen vollständig automatisiert Schlüssel zwischen den VPN-Gateways der unterschiedlichen Standorte zu verteilen.
- **QKD-Schlüssel**: Sofern einzelne QKD-Strecken sinnvoll/wirtschaftlich betrieben werden können, könnten die Gateways Schlüssel von den entsprechenden QKD-Endpunkten anfragen und einsetzen.

Die hier vorgestellten Schlüsselquellen eignen sich alle, um direkte Punkt-zu-Punkt Verbindungen abzusichern. Vor allem in großen VPNs kann damit jedoch noch nicht gewährleistet werden, dass zwischen allen Kombinationen von VPN-Gateways regelmäßig paarweise Schlüssel ausgetauscht werden. Ein alternatives Konzept, wie dennoch Schlüssel sicher indirekt in großen VPNs ausgetauscht werden können, wird später in Abschnitt 3.4 vorgestellt.

Eine weitere Problematik ist, dass es Angreifern möglicherweise zeitweise gelingen wird, einzelne Schlüssel aus den genannten Schlüsselquellen zu kompromittieren. Deshalb sollten so viele Schlüssel wie möglich und aus unterschiedlichen Quellen kombiniert werden, um den Aufwand für Angreifer dauerhaft zu steigern. Insbesondere sollte neues Schlüsselmaterial nicht einfach das bisherige Schlüsselmaterial ersetzen, welches dem Angreifer mög-

licherweise bereits unbekannt war. Deshalb schlagen wir vor, neues Schlüsselmaterial s_i immer mit dem bestehenden „Master-Schlüssel“ k_i, wie in Abbildung 2 veranschaulicht zu einem neuen Master-Schlüssel k_{i+1} zu kombinieren. Eine solche *Schlüsselkette* wird paarweise für jede Verbindung im VPN gepflegt und sollte mittels einer sicheren Key Derivation Function (KDF) realisiert werden. Stand heute ist die Sicherheit von existierenden, etwa Hash-basierten, KDFs nicht von Quantencomputern bedroht, sofern die einzelnen Ein- und Ausgaben lang genug gewählt sind (≥ 256 Bit pro Einzelschlüssel).

Da neue Schlüssel zu unterschiedlichen Zeitpunkten und in verschiedener Reihenfolge bei den beteiligten VPN-Gateways verfügbar werden können, wird ein *Synchronisationsprotokoll* zwischen den Gateways benötigt. Eine Möglichkeit dies zu realisieren, wäre eine Integration in das IKEv2-Protokoll. Allerdings würde dies die Protokoll- und Implementierungskomplexität weiter erhöhen und wäre aufgrund der Standardisierungsprozesse für IKE nicht kurzfristig umsetzbar. Stattdessen schlagen wir vor, dieses Synchronisationsprotokoll unabhängig durch eine kleine separate Software-Komponente durchführen zu lassen.

Weiterhin stellt sich die Frage, wie der aktuelle Master-Schlüssel am besten eingesetzt wird, um Client-Verkehr zusätzlich abzusichern. Auch hier wäre eine Integration in das IKEv2-Protokoll eine Möglichkeit, wie etwa in [9] und [10] vorgeschlagen. Damit werden Teile der IKE-Pakete durch den Master-Schlüssel geschützt, und der Schlüssel fließt in die Ableitung des finalen Sitzungsschlüssels mit ein. Allerdings können die ersten IKE-Pakete dabei nicht geschützt werden, da die Erweiterungen zunächst eine Einigung auf den richtigen Master-Schlüssel vorsehen. Während dies zumindest im zweiten Vorschlag [10] in der Theorie keinen Nachteil für die Sicherheit darstellt, bietet es dennoch eine gewisse Angriffsoberfläche auf die Implementierung und somit den gesamten IKE-Daemon. Deshalb präferieren wir stattdessen, sämtliche IKE-Pakete über eine separate Komponente –genannt IKE-Proxy – umzuleiten. Dort wird der Master-Schlüssel eingesetzt, um alle Pakete zu überschlüsseln sowie ihre Integrität zu sichern und somit sämtliche mittels IKE etablierten Sitzungsschlüssel implizit abzusichern. Das benötigte Synchronisationsprotokoll kann zudem direkt in den IKE-Proxy integriert werden.

Insgesamt bietet die Überschlüsselung der IKE-Pakete Sicherheit in der Tiefe: Solange der jeweils eingesetzte paarweise Master-Schlüssel k_i dem Angreifer nicht bekannt ist, sind Angriffe auf das IKE-Protokoll (sowohl die

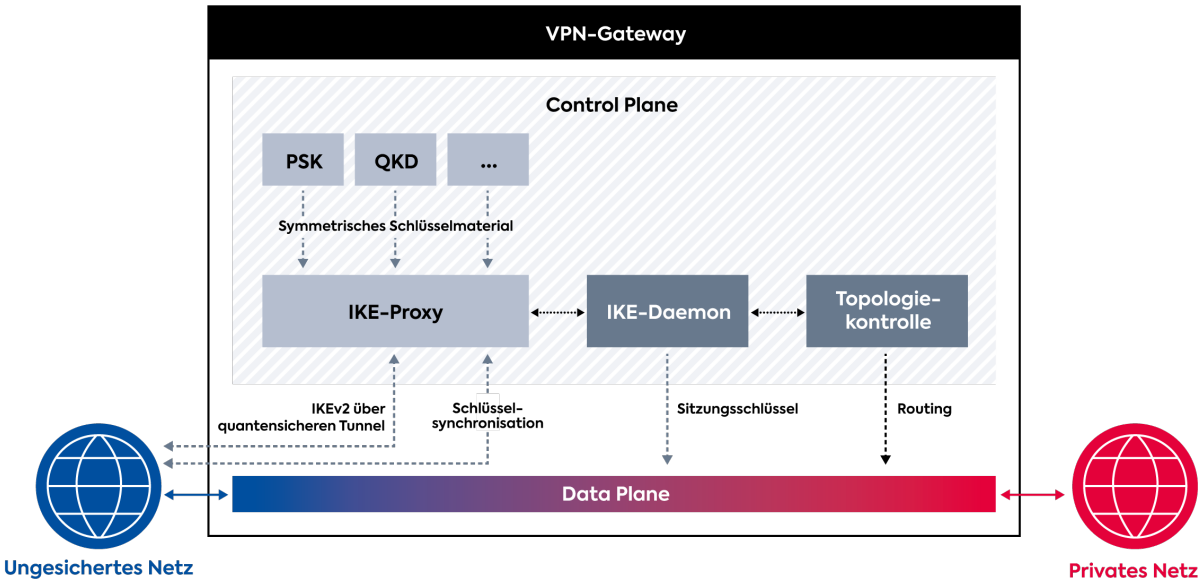

Abbildung 3: Integration des IKE-Proxy in eine VPN-Gateway-Architektur

Implementierung selbst als auch auf die eingesetzten asymmetrischen Kryptografieverfahren) beweisbar ausgeschlossen, selbst wenn zukünftig Schwächen in eingesetzten PQC-Verfahren entdeckt werden. Die Implementierungssicherheit des IKE-Proxys selbst kann zudem aufgrund seiner geringen Komplexität einfacher gewährleistet werden. Des Weiteren ist keinerlei Modifikation des IKE-Protokolls sowie seiner Implementierung nötig, sodass deren Sicherheitseigenschaften beweisbar nicht abgeschwächt werden. Die folgenden Abschnitte geben einen Überblick über die Integration und Funktionsweise des IKE-Proxys und diskutieren die Integration von QKD in das Gesamtkonzept, sowie die Ende-zu-Ende Absicherung von Verbindungen ohne direkte paarweise Schlüsselquellen.

3.1 Integration des IKE-Proxys

Der IKE-Proxy integriert sich in eine typische Architektur eines VPN-Gateways wie in Abbildung 3 gezeigt: Zum einen dient er als Schnittstelle zu allen vorhandenen Quellen für symmetrisches Schlüsselmaterial wie PSKs und QKD. Des Weiteren führt er mit allen VPN-Gateways, zu denen Schlüsselmaterial vorliegt, ein Synchronisationsprotokoll durch, um sich auf einen paarweisen Master-Schlüssel zu einigen, welcher alle bis zu diesem Zeitpunkt jemals beiden Seiten bekannten Schlüssel kombiniert. Dieser Schlüssel wird schließlich benutzt, um alle Pakete, welche vom IKE-Daemon gesendet werden, zusätzlich zu sichern (Verschlüsselung, Integritätsschutz) beziehungsweise eingehende Pakete zu überprüfen und zu entschlüsseln, bevor sie an den IKE-Daemon ausgeliefert werden. Um flexibel auf zukünftige Anforde-

rungen an symmetrische Verfahren reagieren zu können, bietet das verwendete Protokoll eine einfache Möglichkeit der Migration auf neue symmetrische Algorithmen.

3.2 Implementierungsdetails des IKE-Proxys

Dieser Abschnitt fasst einige Implementierungsdetails des IKE-Proxys zusammen, welche für seine Funktionalität und Sicherheitseigenschaften essenziell sind. Weitere Details sind in unserer vorangegangenen Veröffentlichung [11] zu finden.

3.2.1 Synchronisation von Schlüsseln

Die korrekte Synchronisation neuer Schlüssel s_i zu einem neuen Master-Schlüssel ist essenziell für die Verfügbarkeit des VPN: Kann ein IKE-Proxy eingehende Pakete nicht entschlüsseln, weil der Sender einen unbekannten oder falsch synchronisierten Master-Schlüssel verwendet, können keine weiteren IKE-Pakete ausgetauscht werden. Somit würden alle Rekeys sowie der Aufbau neuer Sicherheitsbeziehungen zwischen den betroffenen Gateways fehlschlagen.

Um mögliche Race-Conditions bei der Synchronisation neuer Schlüssel möglichst einfach zu handhaben, schlagen wir vor, zwischen zwei Gateways jeweils zwei Master-Schlüssel zu verwenden, einen in jede Richtung. Zu beachten ist dabei, dass neue Schlüssel s_i stets analog zu Abbildung 2 in beide Master-Schlüssel synchronisiert werden, aber unter Umständen zu unterschiedlichen Zeitpunkten. Ein einfaches Synchronisationsprotokoll sieht vor, dass jeder IKE-Proxy nur die Synchronisation neuer Schlüssel in den Master-Schlüssel für *ausgehende* Pakete initiiert, in einer von ihm festgelegten Reihenfolge. Dies geschieht mit einer einfachen Anfrage, welche eine eindeutige Schlüssel-ID des zu synchronisierenden Schlüssels s_i enthält. Nur wenn die Gegenstelle eine Bestätigung sendet, benutzt der IKE-Proxy den neuen Master-Schlüssel in diese Richtung und löscht den vorherigen. Um sicherzustellen, dass eine fehlerhafte Schlüsselquelle nicht die Verfügbarkeit beeinträchtigen kann, enthält die Bestätigung einen kryptografischen Nachweis über den Besitz von s_i. Dieser kann etwa mittels einer kryptografischen Hash-Funktion h aus s_i berechnet werden: $h(s_i)$. Durch diesen Nachweis können beide Gateways sicherstellen, dass sie denselben Schlüssel von der jeweiligen Quelle erhalten haben, bevor dieser im neuen Master-Schlüssel verwendet wird.

3.2.2 Zuordnung von Master-Schlüsseln

Ein Problem, welches sich aus der strikten Trennung von IKE-Daemon und IKE-Proxy ergibt, ist, dass der IKE-Proxy lediglich die IP-Adresse und Port-

Nummer der jeweiligen Gegenstelle sieht. Im Falle einer dynamischen Änderung, etwa in NAT-Szenarien, muss der Proxy aber dennoch eine statische ID der Gegenstelle ermitteln können, um den richtigen Master-Schlüssel auswählen und verwenden zu können. Dies kann durch ein simples Protokoll gelöst werden, indem der Proxy bei Paketen mit einer unbekannten Kombination aus IP-Adresse und Port-Nummer zunächst eine Anfrage nach der statischen ID der Gegenstelle sendet. Dabei wird sichergestellt, dass ausgetauschte IDs stets mindestens mit einem Gruppenschlüssel abgesichert sind.

3.2.3 Sichere persistente Speicherung von Recovery-Schlüsseln

VPN-Gateways können jederzeit neu gestartet werden, etwa zur Wartung oder nach einem Stromausfall. Damit nach einem solchen Neustart nicht neue paarweise Schlüsselketten verwendet werden müssen, welche beispielsweise nur einen initialen CUG-Schlüssel beinhalten, wird ein Mechanismus benötigt, um die Master-Schlüssel sicher zu persistieren. Eine einfache Methode wäre, alle Master-Schlüssel verschlüsselt auf der Festplatte abzuspeichern. Jedoch widerspricht dieser Ansatz dem Grundsatz der IT-Sicherheit, temporäre symmetrische Sitzungsschlüssel niemals persistent abzuspeichern.

Anstatt also die aktuellen Master-Schlüssel selbst zu persistieren, sehen wir für jeden Master-Schlüssel k_i vor, nur einen verschlüsselten, kryptografischen Hash-Wert $\mathrm{enc}_{\mathrm{kek}}\big(h(k_i)\big)$ zu persistieren („Recovery-Schlüssel“). Der zugehörige „Key Encryption Key“ (KEK) ist nur dem Gateway selbst bekannt und etwa auf einer Smartcard persistent gespeichert. Im Falle eines Neustarts eines der Gateways wird der gespeicherte Hash-Wert verwendet, um innerhalb der Smartcard einen neuen Master-Schlüssel k_{i+1} abzuleiten. In diese Ableitung gehen neben dem Hash-Wert zwei von den Gateways bereitgestellte Nonces n_1, n_2, ihre beiden statischen IDs i_1, i_2, sowie ein Gruppenschlüssel g ein:

$$k_{i+1} = kdf\left(h(k_i), n_1, n_2, i_1, i_2, g\right)$$

Die statische ID des jeweiligen Gateways sowie der Gruppenschlüssel werden hierbei direkt in der Smartcard gespeichert. Letzterer kann so hinterlegt werden, dass sich dieser im Regelfall nicht auslesen lässt.

Diese Art der Persistierung gewährleistet, dass selbst von einem Angreifer mit physikalischem Zugriff auf die Festplatte und die Smartcard, vorherige verwendete Master-Schüssel des angegriffenen Gateways nicht rekonstruiert werden können. Weiterhin kann ein Angreifer mit Zugriff auf eine Smartcard nur gültige (wiederhergestellte) Master-Schlüssel für dieses eine zugehörige Gateway erzeugen, nicht für beliebige andere Gateways.

3.3 Integration von QKD

Getrieben durch aktuelle Standards der ITU und ETSI, etwa [12], sowie durch kommerziell verfügbare Produkte, wird die QKD-Technologie aktuell vor allem mit sogenannten QKD- bzw. „Key Management"-Netzen in Verbindung gebracht. Dort ist die Idee, die eingeschränkte Reichweite von QKD zu überwinden, indem Schlüsselmaterial über mehrere sogenannte „Trusted Nodes" weitergeleitet wird, welche jeweils über eine QKD-Strecke verbunden sind. Dadurch sollen Schlüsselnutzern (etwa VPN-Gateways) transparent paarweise Schlüssel auch ohne direkte QKD-Verbindung bereitgestellt werden. Wesentliche Probleme dieses Vorgehens im Kontext eines VPN sind allerdings mögliche Schwächen im Falle kompromittierter „Trusted Nodes" und der Umstand, dass innerhalb des QKD-Netzwerks viele Funktionalitäten eines VPN erneut implementiert, evaluiert und gegebenenfalls zugelassen werden müssen. Hierzu zählen unter anderem die Authentisierung zwischen den QKD-Geräten zur Realisierung des klassischen Kanals, Routing-Verfahren, sowie Netzwerkmanagement und -monitoring.

Um die Gesamtkomplexität deutlich zu verringern, schlagen wir stattdessen vor, QKD-Strecken jeweils nur Punkt-zu-Punkt zu betreiben und lokal zu konfigurieren. Weiterhin sollte der klassische Kanal zweier QKD-Geräte über die zugehörigen VPN-Gateways mit einer eigenen Sicherheitsbeziehung getunnelt und abgesichert werden. Dadurch kann der klassische Kanal von den bereits etablierten Sicherheitsmechanismen und der gehärteten Implementierung der VPN-Gateways profitieren. Des Weiteren könnte je nach Einsatzbedingungen die Implementierung eigener Authentisierungsprotokolle in den QKD-Geräten entfallen beziehungsweise aus dem Zulassungsprozess genommen werden, um Komplexität einzusparen. Nach einem erfolgreichen QKD-Schlüsselaustausch kann die Sicherheitsbeziehung im VPN durch die Ableitung eines neuen Master-Schlüssels für den IKE-Proxy weiter abgesichert werden. Weitere Synergieeffekte würden sich durch eine Integration von Monitoring-Daten der QKD-Strecke (etwa Schlüsselrate oder Authentisierungsfehler) in bestehende Monitoring-Systeme für VPN-Infrastrukturen ergeben.

3.4 Transitive Ende-zu-Ende Sicherheit

Die bisher vorgestellten Schlüsselquellen sind jeweils darauf ausgelegt, Punkt-zu-Punkt-Verbindungen abzusichern, und skalieren nicht, um in großen VPN-Infrastrukturen paarweise individuelle Schlüssel für alle möglichen Verbindungen bereitzustellen und regelmäßig zu erneuern. Um nun auch

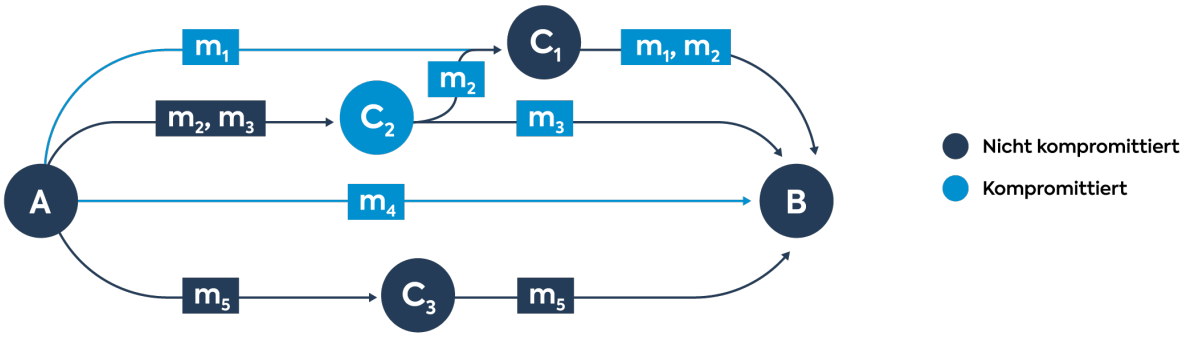

Abbildung 4: MKR-Schlüsselaustausch zwischen Gateways A und B mit insgesamt fünf Pfaden (fünf Teilschlüssel, jeweils Hop-by-Hop über zuvor etablierte Sicherheitsassoziationen übertragen). In diesem Beispiel gelingt es einem Angreifer durch die Kompromittierung einzelner Kanten und Knoten nur vier der fünf Teilschlüssel zu kompromittieren (abzuhören), jedoch nicht den Teilschlüssel m_5. Anzumerken ist noch, dass wir in diesem Beispiel davon ausgehen, dass zwar der Knoten C_2 kompromittiert ist, der Angreifer aber Pakete auf dem Transportweg von und zu C_2 nicht abhört (dunkelblaue Kanten). Dementsprechend sind die Teilschlüssel m_2 und m_3 erst als kompromittiert anzusehen, nachdem sie von C_2 weitergeleitet wurden.

Verbindungen zwischen zwei Gateways absichern zu können, die noch keinen paarweisen Schlüssel besitzen, müsste zwischen diesen ein Pfad existieren auf dem aktuell keine Sicherheitsbeziehung und auch kein VPN-Gateway kompromittiert ist. Existiert ein solcher Pfad, könnte entsprechend über diesen ein sicherer Schlüssel ausgetauscht werden.

In der Realität ist jedoch nicht bekannt, welche VPN-Gateways oder welche Sicherheitsbeziehung durch einen Angreifer zum jeweiligen Zeitpunkt kompromittiert sind. Um mit dieser Problematik umgehen zu können, gibt es den Ansatz, Schlüssel über möglichst viele unterschiedliche Pfade innerhalb des VPN zu verteilen und hierdurch den Aufwand und somit die Kosten für einen Angriff deutlich zu erhöhen. Dieser Ansatz ist auch unter dem Begriff „Multipath Key Reinforcement“ (MKR) bekannt, und wurde unter anderem bereits im Kontext von Wireless-Sensor-Netzen [13] sowie reinen QKD-Netzen [14] untersucht. In Abbildung 4 ist ein solcher Schlüsselaustausch zwischen zwei Gateways A und B dargestellt. Beide Knoten sind über mehrere Pfade miteinander verbunden. Da ihnen nicht bekannt ist, welche Pfade aktuell sicher sind, tauschen sie über jeden der verfügbaren Pfade einen unabhängigen Schlüssel aus. Diese Pfadschlüssel werden anschließend mittels einer KDF zu einem einzelnen MKR-Schlüssel kombiniert. Somit führt bereits ein einzelner sicherer Pfad zu einem sicheren MKR-Schlüssel.

Grundsätzlich lässt sich das beschriebene Verfahren direkt in kleineren und statischen VPNs einsetzen. In größeren VPNs und insbesondere in dynamisch verwalteten VPNs ergeben sich allerdings zwei Grundprobleme:

1. Die Topologie großer VPNs ist nicht statisch (Ausfall von Gateways, dynamischer Auf- und Abbau von Sicherheitsbeziehungen), sodass sich die verfügbaren Pfade jederzeit ändern können.

2. In größeren VPNs existieren gegebenenfalls sehr viele unterschiedliche Pfade zwischen zwei VPN-Gateways. Bei jedem MKR-Austausch alle verfügbaren Pfade zu verwenden, würde die Skalierbarkeit beeinträchtigten. Zur Wahrung der Skalierbarkeit muss deshalb die Anzahl der verwendeten Pfade pro Austausch limitiert werden. Dennoch sollte mittelfristig dieselbe Sicherheit gewährleistet werden, wie wenn immer alle Pfade verwendet würden.

Beide Probleme können leicht gelöst werden. Zwei VPN-Gateways können etwa mittels randomisierten und begrenzten Flutens Pfade zwischen einander suchen, ohne die Skalierbarkeit des VPNs zu beeinflussen. Ebenfalls kann diese Suche vollständig verteilt in den Gateways erfolgen, sodass keine zentrale Instanz benötigt wird. Abhängig von der Anzahl der gefundenen Pfade können die Gateways anschließend entweder alle oder nur eine Teilmenge der Pfade für den eigentlichen MKR-Schlüsselaustausch randomisiert auswählen.

4 Diskussion und Zusammenfassung

Bezüglich der in Kapitel 2 aufgeführten Anforderungen Krypto- und Schlüsselquellenagilität sowie Implementierungssicherheit lässt sich Folgendes festhalten:

Der hier vorgestellte Ansatz des IKE-Proxys bietet den Vorteil, symmetrisches Schlüsselmaterial aus unterschiedlichen Quellen einbeziehen zu können. Diese Eigenschaft ist von Vorteil, wenn die Sicherheit der verwendeten Schlüsselquellen nicht durch einen verständlichen Sicherheitsbeweis nachgewiesen werden kann. So wurde zum Beispiel kürzlich durch das BSI in einem neuen Positionspapier [7] die große Ungewissheit bezüglich der Sicherheit von QKD beziehungsweise aktueller QKD-Implementierungen deutlich. Würde der IKE-Proxy, so wie aktuell vorgeschlagene QKD-Netze, nur auf QKD-Schlüssel setzen, wären alle Vorteile durch diese Ungewissheit gefährdet. Durch die Kombination mit anderen Schlüsselquellen können hingegen selbst großflächige und weitreichende Sicherheitsprobleme, beispielsweise

in einer QKD-Implementierung, automatisch und transparent mitigiert werden.

Durch den konzeptionellen Ansatz, IKE-Pakete vor dem Versand über das Internet immer zu überschlüsseln, wird eine sogenannte „second line of defense“ aufgebaut. Durch diese Maßnahme muss ein Angreifer entweder den aktuellen Master-Schlüssel entwenden können, um direkt mit dem IKE-Daemon interagieren zu können, oder eine gravierende Sicherheitslücke in der Implementierung des IKE-Proxys identifizieren, durch die der gesamte Schutz des IKE-Proxy umgangen werden kann. Aufgrund der geringen Komplexität des Konzepts gehen wir allerdings von einer einfachen und nachvollziehbaren Implementierung aus. Sollte es dem Angreifer dennoch gelingen, Zugriff auf die IKE-Pakete im Klartext zu erlangen, muss er weiterhin die dort implementierten asymmetrischen kryptografischen Mechanismen brechen, um Client-Verkehr entschlüsseln zu können. Der allgemeine Nachteil einer „second line of defense“ ist allerdings, dass diese die Verfügbarkeit und Robustheit des abgesicherten Dienstes negativ beeinflussen kann. Konzeptionell wird diese Problematik durch das Schlüsselsynchronisationsprotokoll sowie die persistente Speicherung von Recovery-Schlüsseln adressiert, welche zusammen die Konsistenz der paarweisen Schlüsselketten sicherstellen. Zudem kann das Risiko eines Fehlerzustandes durch rigoroses Testen minimiert werden. Weiterhin können paarweise Schlüsselketten mit einem Gruppenschlüssel initialisiert werden, sodass auch eine Kommunikation zwischen VPN-Endpunkten stattfinden kann, die bisher keinen einzigen gemeinsamen paarweisen Schlüssel kennen.

Wird ein VPN zusätzlich mit dem IKE-Proxy und den in diesem Artikel vorgestellten Schlüsselquellen abgesichert, erhöhen sich die Kosten für einen Angreifer drastisch. Dies ist damit zu begründen, dass es nicht mehr ausreicht einen einzelnen Sicherheitsmechanismus anzugreifen. Stattdessen müssen alle im Folgenden aufgeführten Sicherheitsmechanismen umgangen werden, um Zugriff auf Client-Verkehr im Klartext zu erhalten:

- **Klassische asymmetrische Kryptografie**: Der Angreifer benötigt einen Quantencomputer, andernfalls kann er die eingesetzte asymmetrische Kryptografie nicht brechen.

- **PQC**: Der Angreifer benötigt Wissen über Schwachstellen in PQC-Algorithmen oder deren Implementierung, andernfalls kann ein sicherer Sitzungsschlüssel mit PQC ausgetauscht werden.

- **QKD**: Der Angreifer muss eine QKD-Verbindung ab der ersten Benutzung durchgängig kompromittieren, oder gravierende Implementierungsfehler identifizieren. Ist ihm dies nicht durchgängig möglich, wird sicheres Schlüsselmaterial mittels QKD ausgetauscht werden, welches anschließend in die paarweise Schlüsselkette im IKE-Proxy einfließt.

- **PSKs**: Der Angreifer muss jeden jemals ausgetauschten PSK, welcher in eine paarweise Schlüsselkette eingeflossen ist, abfangen oder nachträglich in Erfahrung bringen. Sofern ihm ein einzelner PSK fehlt, kann er die damit abgesicherte Verbindung nicht mehr abhören. Wird sichergestellt, dass PSKs nach der Verwendung auf sichere Weise gelöscht werden, kann der Angriff sogar nur noch während des manuellen Schlüsselaustausches erfolgen.

- **„Business Trip Key Exchange“**: Der Angreifer muss jeden mittels eines abgesicherten Laptops automatisch ausgetauschten Schlüssel, welcher in eine paarweise Schlüsselkette eingeflossen ist, in Erfahrung bringen. Hierzu muss jeder jemals verwendete Laptop kompromittiert werden. Wird sichergestellt, dass die Schlüssel nach dem Austausch auf den verwendeten Laptops auf sichere Weise gelöscht werden, muss ein Angriff während den Dienstreisen erfolgen. Demzufolge müssen alle Laptops vom Angreifer durchgängig kompromittiert sein.

- **MKR**: Der Angreifer muss im VPN entweder jede Verbindung durchgängig abhören oder jedes Gateway durchgängig kompromittieren. Ist dem Angreifer dies auch nur kurzzeitig nicht möglich, kann MKR durch seine transitive Eigenschaft das gesamte VPN schrittweise absichern. Demzufolge muss ein Angreifer darauf achten, dass ihm beim Abhören des VPN keine Pakete, etwa durch eine Überlast auf den Verbindungen zu seinem Rechenzentrum, verloren gehen.

Insgesamt sorgt die Kombination dieser Verfahren dafür, dass, sofern ein Angriff überhaupt noch möglich ist, dieser mit erheblichen logistischen und finanziellen Aufwänden für den Angreifer einhergeht.

Danksagung

Diese Veröffentlichung wurde durch dtec.bw – Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr gefördert (Projekt MuQuaNet). dtec.bw wird von der Europäischen Union – NextGenerationEU finanziert.

Literaturhinweise

[1] P. W. Shor, „Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer“, *SIAM J. Comput.*, Bd. 26, Nr. 5, S. 1484–1509, Okt. 1997, DOI: 10.1137/S0097539795293172.

[2] J. Proos und C. Zalka, „Shor’s Discrete Logarithm Quantum Algorithm for Elliptic Curves“, *Quantum Information and Computation*, Bd. 3, Nr. 4, S. 317–344, 2003.

[3] C. H. Bennett, E. Bernstein, G. Brassard, und U. Vazirani, „Strengths and Weaknesses of Quantum Computing“, *SIAM J. Comput.*, Bd. 26, Nr. 5, S. 1510–1523, Okt. 1997, DOI: 10.1137/S0097539796300933.

[4] European Union Agency for Cybersecurity, „Post-Quantum Cryptography: Current State and Quantum Mitigation.“, Publications Office of the European Union, 2021. doi: 10.2824/92307.

[5] W. Beullens, „Breaking Rainbow Takes a Weekend on a Laptop“, in *Advances in Cryptology – CRYPTO 2022*, Bd. 13508, Cham, 2022, S. 464–479. DOI: 10.1007/978-3-031-15979-4_16.

[6] C. H. Bennett und G. Brassard, „Quantum Cryptography: Public Key Distribution and Coin Tossing“, *Theoretical Computer Science*, Bd. 560, S. 7–11, Dez. 2014, DOI: 10.1016/j.tcs.2014.05.025.

[7] French Cybersecurity Agency, Federal Office for Information Security, Netherlands National Communications Security Agency, und Swedish National Communications Security Authority, Swedish Armed Forces, „Position Paper on Quantum Key Distribution“. Zugegriffen: 22. Februar 2024. [Online]. Verfügbar unter: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Quantum_Positionspapier.html

[8] V. Smyslov, „Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2)“, Mai 2022. Zugegriffen: 22. Februar 2024. [Online]. Verfügbar unter: https://www.rfc-editor.org/rfc/rfc9242.html

[9] S. Fluhrer, P. Kampanakis, D. McGrew, und V. Smyslov, „Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security“, Juni 2020. Zugegriffen: 22. Februar 2024. [Online]. Verfügbar unter: https://www.rfc-editor.org/rfc/rfc8784

[10] V. Smyslov, „Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security“, Nov. 2022. Zugegriffen: 22. Februar 2024. [Online]. Verfügbar unter: https://datatracker.ietf.org/doc/draft-smyslov-ipsecme-ikev2-qr-alt/

[11] D. Schatz, F. Altheide, H. Koerfgen, M. Rossberg, und G. Schaefer, „Virtual Private Networks in the Quantum Era: A Security in Depth Approach“, in *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT)*, 2023, S. 486–494. DOI: 10.5220/0012121800003555.

[12] ITU-T, „Overview on Networks Supporting Quantum Key Distribution“, Recommendation ITU-T Y.3800, 2019.

[13] J. Deng und Y. S. Han, „Multipath Key Establishment for Wireless Sensor Networks Using Just-Enough Redundancy Transmission“, *IEEE Trans. Dependable and Secure Comput.*, Bd. 5, Nr. 3, S. 177–190, Juli 2008, DOI: 10.1109/TDSC.2007.70233.

[14] S. Rass und S. König, „Indirect Eavesdropping in Quantum Networks“, in *Proceedings of the 5th ICQNM*, 2011, S. 83–88.