Security of VPN Infrastructures in Times of Upcoming Quantum Computer Threats

Summer Semester 2024

Fachgebiet Telematik/Rechnernetze

(Material prepared by Friedrich Altheide, David Schatz, Michael Rossberg, Günter Schäfer)

Overview

Network layer VPNs: Scenarios, requirements & current solutions

Challenge: How to defeat quantum computing attackers?

Emerging Quantum Key Distribution standards: Overview and reflection

Securing connections on different layers:

- Layer 1 / layer 2
- Scaling layer 1 / layer 2 networks
- Scaling beyond small networks → layer 3 (IPsec)

Securing large scale networks:

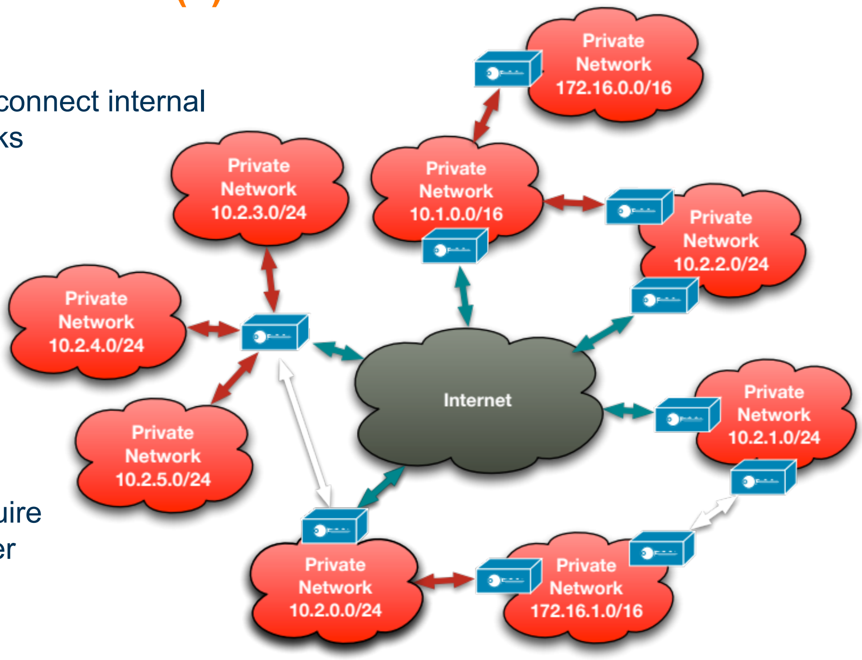
- Countermeasures against attackers with quantum computer
- Proposal for “combination” of various key exchange methods in times of uncertainty of Post Quantum Cryptography’s security

Summary & Outlook

Network Layer VPN Infrastructures (1)

Scenario:

- VPN gateways and mobile workers connect internal networks over untrustworthy networks
- Smartcards used as trust anchors
- Public & private IP address ranges (IPv4 or IPv6)
- Nested networks
- Multiple networks per gateway
- Multiple gateways per network
- Cycles in the network (required for robustness and handling load!)
- Some sites with many networks require advanced load balancing and failover mechanisms



⇒ High complexity!

Network Layer VPN Infrastructures (2)

Customer expectations are simple:

- BSI-compliant crypto-processing at line speed or at least at well-defined speeds
- Handling of appliances as good/bad as other networking equipment: Robustness, management, enrollment
- Behave as transparently as possible
- Important VPN properties: scalability, agility, robustness

Key enablers to implement secure, scalable and robust VPNs:

- Avoid centralized components
- Use as few security associations (SAs) as possible (SA establishment is expensive!)
  - VPN gateways implement an overlay network/graph (gateway = node, SA = link)
  - Use tunneled SAs to guarantee end-to-end security (some gateways might be compromised)
- Keep (overlay) topology knowledge local
- Automatic configuration as far as possible (by “control algorithm”)

Network Layer VPN Infrastructures (3)

Further required for scenarios with enhanced needs for protection (e.g., “GEHEIM”):

- Security hardening of components, e.g., regarding:
  - Side-channel attacks
  - Minimizing trusted computing base (TCB)
  - Tamper-proofing
- Approval according to protection profile(s)

How to Overcome “Quantum Threat” to Classical Asymmetric Cryptography? (1)

Three main directions for overcoming threat by Shor’s algorithm [Sho97], [PZ03]:

- Symmetric Cryptography
  - Grovers algorithm [Gro96] halves effective key length, which also is a lower bound [BBB+97]
    → AES-256 etc. should stay save
  - Symmetric key management either cumbersome or needs central trusted third party
    (→ single point of failure)
- Post Quantum Cryptography (PQC)
  - Requires: Longer keys, longer messages and more computation (→ smart cards?)
  - Still raises concerns regarding maturity of cryptanalysis (e.g., see Rainbow [BW22])
  - Currently not recommended to be used alone
    → use hybrid mode (PQC together with e.g. classical Elliptic Curve Cryptography (ECC))
- Quantum Key Distribution (QKD)
  - Can “physically” guarantee point-to-point confidentiality (“no cloning theorem”):
    - After out-of-band authentication!
    - If no implementation weakness exists and side-channel attacks are impossible!
  - Only provides point-to-point security (requires “direct” medium, limited reach, currently ~100 km)

How to Overcome "Quantum Threat" to Classical Asymmetric Cryptography? (2)

QKD requires concepts for networking QKD-enabled devices (point-to-point ➜ end-to-end)

Open challenges:

- How to do this without unnecessarily "reinventing wheels" (→ established VPN technology)?
- How to reduce efforts for hardening "QKD networking"-related software components?
- How can security be increased between red networks with no direct QKD link?
- How can security be increased for red networks with no QKD link at all?

Implications for QKD Integration

At first glance: None!

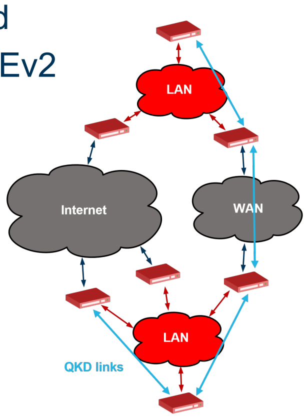
- QKD only affects confidentiality, integrity, availability properties of certain links
- "Buried" in layers below

At second glance: We need to use the keys for establishing SAs in the overlay

- Secure interface between QKD devices and VPN gateways required
- Preferably integration of QKD keys in established protocols, e.g., IKEv2 (instead of a complete redesign)

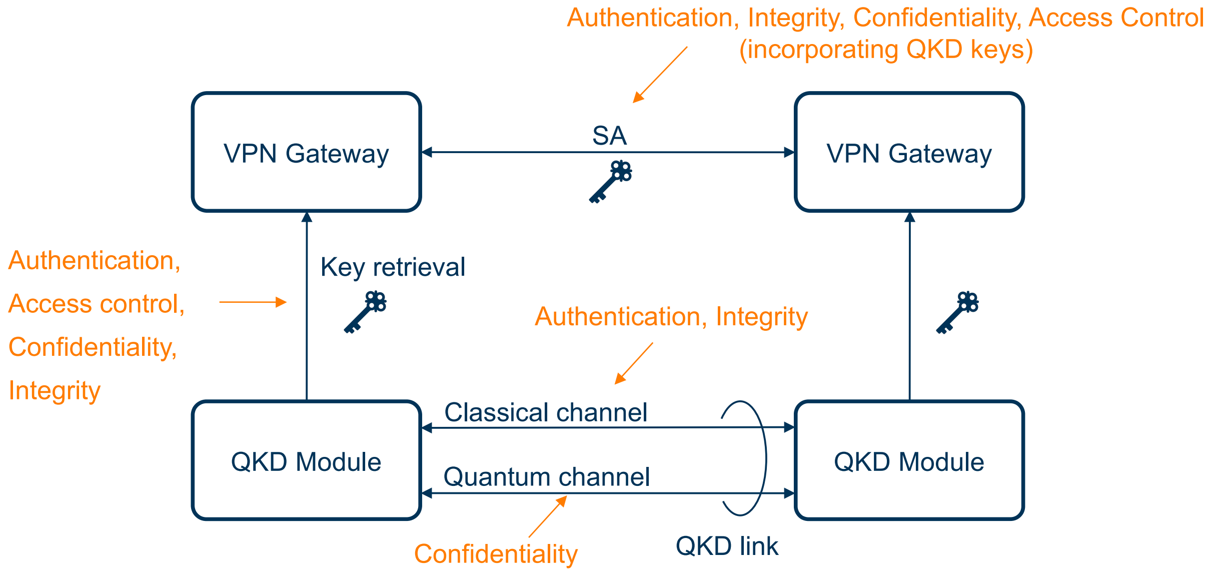
Broader view: Impact in heterogenous infrastructures?

- Only some links have QKD (due to limited reach, costs)
- Benefit of QKD to the overall security for arbitrary SAs?
  - How to quantify benefit?
  - How to maximize benefit?

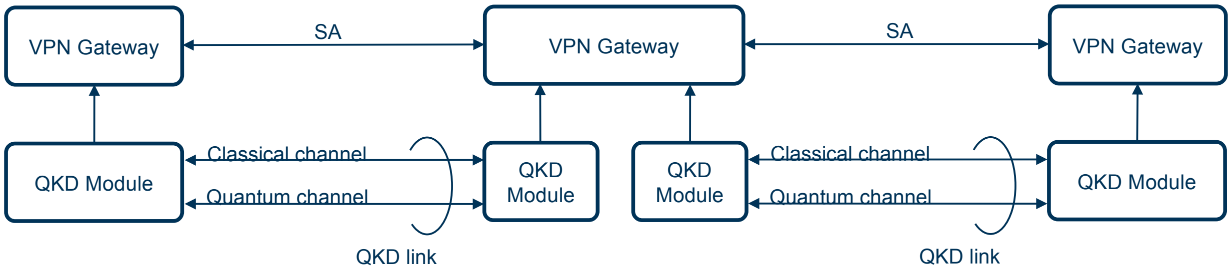


Required Security Services

Abstract, high-level view of QKD link integration:



QKD Network Security Considerations



Basic assumptions:

- Authentication needs to be realized with combination of PQC and classical cryptography
- Symmetric cryptography with sufficiently long keys (e.g., ≥ 256 bit) can not be broken
- It is impossible to eavesdrop on a “securely” authenticated QKD link
- It is rather easy to eavesdrop on individual classical links
- With growing network size, it gets harder to always eavesdrop on all classical links
- It is not impossible to compromise individual VPN gateways / QKD modules (but high effort!)
- The more complex a solution is, the easier it is to compromise

Emerging Standards: ITU-T Y.3800 – Y.3805 (1)

Scope: QKD networks

- Idea: Transparently extend the reach of QKD by relaying keys via “trusted” nodes
- Main contribution: Reference architecture(s)

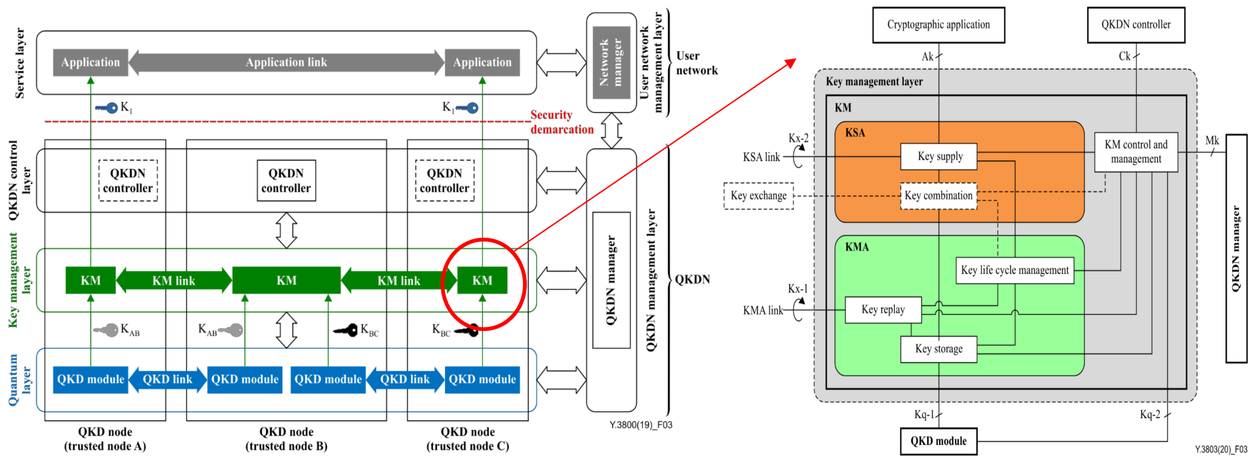


Figure source: [Y.3800, Y.3803]

Emerging Standards: ITU-T Y.3800 – Y.3805 (2)

Discussion:

- No specific protocols → No interoperability, implementation complexity “hidden”
- “Standard Writer’s Standard”?
  - ~36 Functional Requirements with 9 notes [ITU-T Y.3801]
  - ~32 Functional Elements, ~22 Reference Points [ITU-T Y.3802]
  - > 50 Functions [ITU-T Y.3804]

→ overly complicated?

- Security services: Identified, but very little information provided on what concrete security objectives need to be ensured and how this is supposed to be realized:
  - “[Security] [d]etails are outside the scope of this Recommendation” [ITU-T Y.3801, Y.3802, Y.3804, Y.3805]
  - “[…] security requirements described in [ITU-T X.1710], [ITU-T Y.3801] and [ITU-T Y.3802] and general network security requirements and mechanisms in IP-based networks described in [ITU-T Y.2701] and [ITU-T Y.3101] are recommended to be applied”

→ How to ensure secure implementations with these recommendations?

Gall's Law

John Gall (1925 –2014), pediatrician and author

- Most famous book: *"General Systemantics: An Essay On How Systems Work, And Especially How They Fail..."* (1975)
  (Third edition, entitled "The Systems Bible" published in 2002)
- *"A complex system that works is invariably found to have evolved from a simple system that worked.*
  A complex system designed from scratch never works and cannot be patched up to make it work.
  You have to start over with a working simple system."
  (1975, p. 71)
- In security, we are not only concerned with systems simply "working", but to ensure that they do not have unintended vulnerabilities
  - This is even harder to achieve!

Excursion: Software Vulnerabilities

Some examples:

- Heartbleed [CVE-2014-0160]: Memory leak in the `openssl` implementation of the TLS heartbeat extension → Potentially leaked many long-term secret keys
- Log4Shell [CVE-2021-44228]: Vulnerability in "harmless" dependency (logging framework)
  → Allowed remote code execution for nearly ten years
- And countless more

Implications:

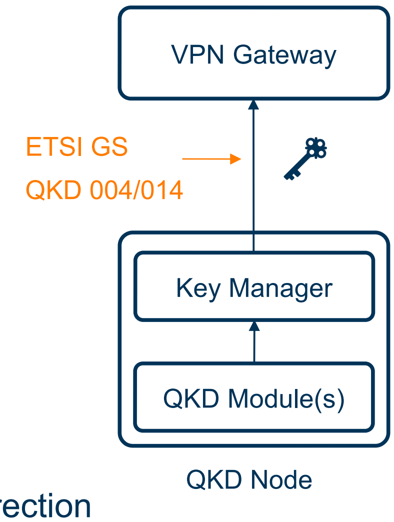
- Avoid (designing and) implementing complex protocols from scratch
- Keep TCB as small as possible

Emerging Standards: ETSI GS QKD 004, 014

Scope: Key retrieval in QKD networks

ETSI GS QKD 014

- State of the art in commercially available products
- REST-based HTTP API
- Security services implemented by PKI-based TLS
  - Does not match the security level of QKD
  - Overall huge TCB: ~500k lines of code dependencies for client and server each (using well established Rust libraries)



ETSI GS QKD 004

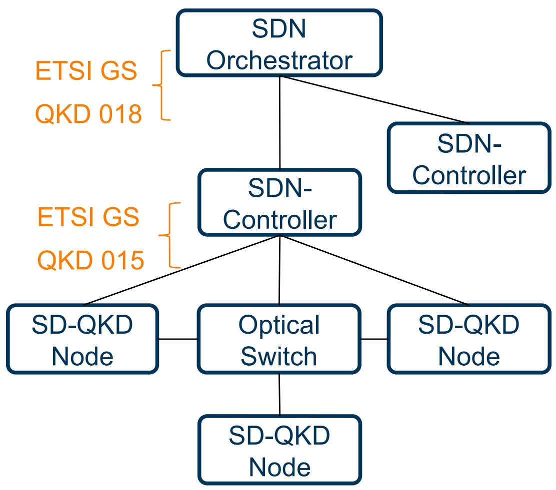
- Sleeker design compared to ETSI GS QKD 014 → Right direction
- But: Underspecified (e.g., encoding on wire) → Interoperability?

Emerging Standards: ETSI GS QKD 015, 018

Scope: Management & monitoring of QKD nodes

ETSI GS QKD 015

- Central management and on demand configuration of QKD nodes and "lightpaths" using SDN
- Dynamically configuring trusted nodes to increase reachability
  → Introduces central weak point (SDN controller)



ETSI GS QKD 018

- Introduces SDN orchestrator for multi-domain management and monitoring
  - But: What is a domain? How separated?

Emerging Standards: Reflection

Common conception/objective: “Standalone” QKD networks?

- Hope: Maximizes transparency for (generic) key consumers

Not optimally suited in the context of existing VPN infrastructures

- Routing, key management, authentication, and integrity implemented on two layers (QKD and VPN) → Increased complexity and larger TCB
- Lack of standardization for many interfaces and implementation of security services (e.g., authentication on classical channel of QKD links)
  - → Proprietary protocols and implementations
  - → Additional effort for hardening and approval of QKD nodes software components
- “Trusted” nodes not satisfying (or even prohibitive?) in VPNs with enhanced needs for protection

Integrated approach better suited?

How to maximize the benefit of QKD without solely relying on trusted nodes?

Intermezzo: Recommendation of Federal Authorities (1)

Source: Presentation of Manfred Lochter @CODE conference 2022 (Additional source: [NSA23])

What do other security agencies say?

National Cyber Security Centre

NCSC – Whitepaper: Quantum Security Technologies (2020)

*“Given the **specialised hardware requirements** of QKD over classical cryptographic key agreement mechanisms and the **requirement for authentication** in all use cases, the NCSC does not endorse the use of QKD for any government or military applications […].”*

ANSSI - Technical Position Paper: QKD (2020)

*“Security guarantees provided in principle by QKD come with **significant deployment constraints** which reduce the scope of the services offered and compromise in practice QKD security assurances, particularly in scenarios where communications travel through a network of interconnected QKD links.”*

NSA – Quantum Key Distribution (QKD) and Quantum Cryptography (QC)

*“NSA **does not recommend** the usage of quantum key distribution and quantum cryptography for securing the transmission of data in National Security Systems (NSS) unless the limitations […] are overcome.”*

Federal Office for Information Security

Intermezzo: Recommendation of Federal Authorities (2)

Source: Presentation of Manfred Lochter @CODE conference 2022

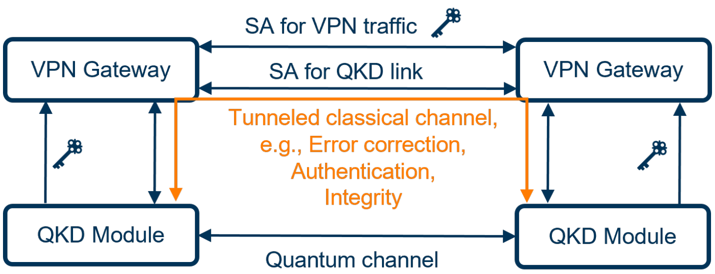
Key Points from BSI‘s recommendations

- QKD is feasible with technology available today and provides key agreement schemes whose security is based on quantum mechanical principles and which are expected to be information-theoretically secure at the protocol level.
- In addition to theoretical security, implementation security must also be considered.
- QKD is subject to some restrictions and is therefore only suitable for certain application scenarios.
- Standards, for example on protocols, and certified products are still lacking.
- QKD should only be used in hybrid mode with classical and post-quantum key agreement schemes.
- Using the one-time pad alone for encryption is not recommended.

Federal Office
for Information Security

Integrated Approach (1): Direct QKD Link

- Additional SA for each QKD link, established using PQC/pre-shared keys (PSKs)
- Tunnel classical channel (e.g., error correction) via VPN gateways and additional SA
- Options for security services between QKD module and VPN gateway: PQC, PSKs, “physical means”
- Include QKD keys when establishing SAs for “normal” VPN traffic (“include” in key derivation)
- Traffic secured by symmetric cryptography as usual (e.g., AES, …)

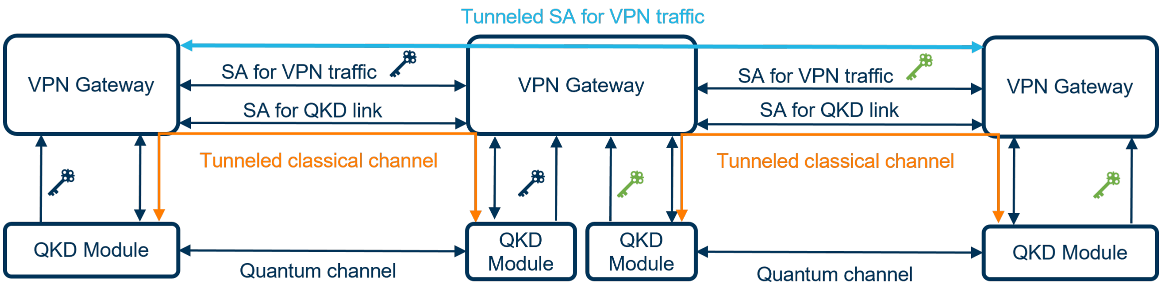


→ Reduced attack surface on QKD modules (no classical communication via public channels)
→ Reduced complexity of QKD modules (no authentication with other modules)

Integrated Approach (2): Multi-hop QKD

Approach:

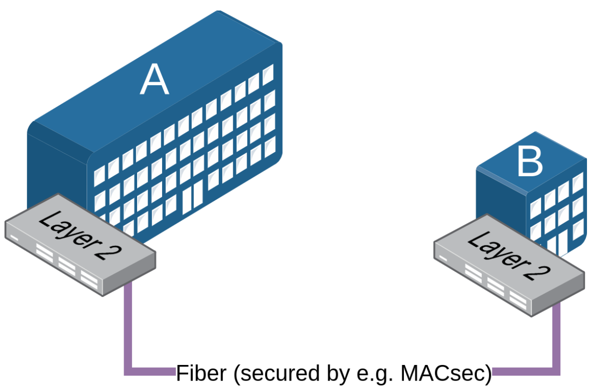
- Establish “tunneled” SA, hop-by-hop protected by existing SAs with direct access to QKD links
- End-to-end authentication and key exchange: PQC/classical cryptography
- Optimization: Re-route (shortcut) VPN traffic after successful authenticated key exchange



Discussion:

- Same (or better?) end-to-end security properties compared to QKD network with trusted nodes
- Reduced complexity and TCB (use established VPN technologies for multi-hop key management)

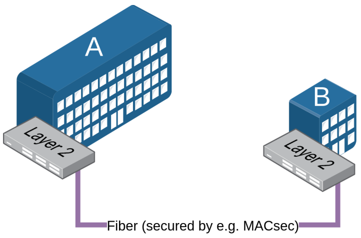
Securing Traffic Between Two Sites (Point to Point)

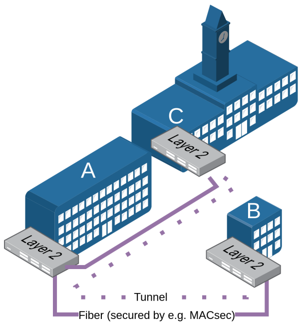


- Two sites can easily be connected securely
  - e.g. via layer 1 encryption, MACsec, L2-VPN
  - QKD can easily be integrated for perfect forward secrecy (PFS)
    - But, the classical channel initally has to be authenticated out of band
      → reliance on pre-shared key (PSK) for this
- Easy maintainability

→ How to secure traffic between e.g. three sites?

Securing Traffic Between Some Sites (Routing vs. Tunneling)





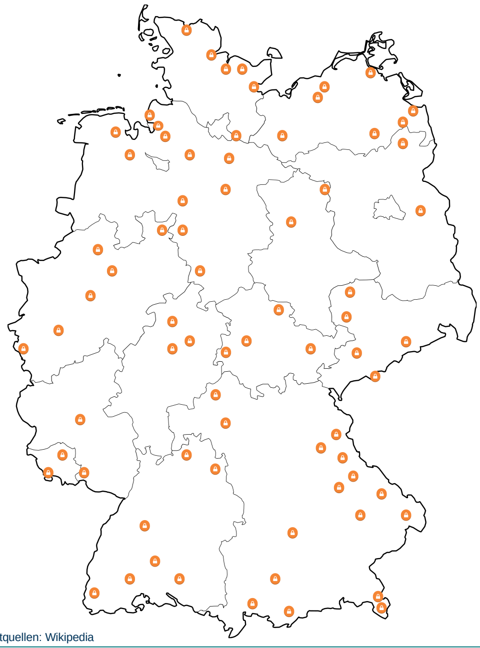
- Traffic secured Hop by Hop
- Traffic between B and C is not end-to-end secure
  - → A sees traffic in cleartext
  - → What if A is compromised?

→ Same principles apply to QKD

- Introduce cryptographic tunnel between B and C
  - e.g. virtual tunnel, secured by end-to-end MACsec
- If A is compromised, traffic between B and C still secure
- But scalability of L2 Tunnels and manual configuration very limited, we need something different

→ **Note**: QKD cannot be tunneled

Securing Traffic Between Many Sites (VPN)



Standortquellen: Wikipedia

- With many sites, many devices are expected → L3
- Manually exchanging PSKs between all pairs of sites cumbersome → PQC-based PKI?
- Tunnels between many sites, requires mechanisms to automatically establish on demand, without SPoF → SOLID
- But, what if PKI (PQC) may not proof to be secure?
  - Exchange PSKs once manually → cumbersome and key quality degrades (BSI concerns)
  - Exchanging PSKs manually regularly → just cumbersome
  - QKD → viable for only a few connections due to limited range, fiber requirements, keys exchanged via “Trusted Nodes” can not be trusted

→ **Idea:** Drastically increase attacking costs and lower attackers chances

- How to automate regular PSK exchanges, so that attacks get too expensive & attackers miss some exchanged keys?
- How to utilize the point to point security properties of QKD within the process?

Countermeasures: Make Attacks Very Expensive to Conduct & Coordinate

Desirable properties of secure PSK exchanges

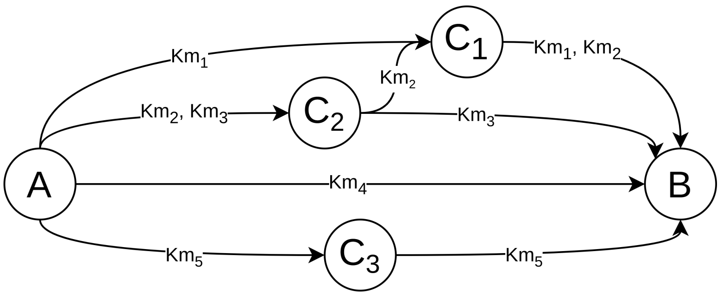
Property	Idea sketch
Force attacker to compromise every SA secured by PQC, classical asymmetric, PSKs, …	Exchange keys within the VPN, hidden in user traffic
Force attackers to constantly intercept	Exchange new PSKs frequently, e.g. every hour
Ensure Attacker have to infiltrate many locations / paths	Exchange keys via different paths (→ MKR)
Force attackers to compromise QKDs on first usage (PFS if the integrity of the classical channel was ensured)	Use keys exchanged using QKD to secure SAs if available
Exclude attackers that may have compromised some gateways or SAs and can not be excluded by utilizing single or even multiple paths	Exchange some keys out of band, e.g.: • Use business trips to automatically exchange PSKs using secure workstations • Exchange some PSKs on smartcards, e.g. via postal service or during on premise maintenance
Do not introduce single points of failure	Build a distributed system
Do not allow "downgrade" attacks	Build a resilient system without any fallbacks

- Combine all these properties to make attacks very expensive, while PSK exchanges are inexpensive
  → How to discover and route PSKs over different paths and securly combine all these PSKs?

Multipath Key Reinforcement (1)

Basic idea: Alice sends key material $Km_1, \ldots, Km_n$ to Bob via multiple paths

- All Km_j are combined via key derivation function (KDF) to a single PSK $s_i = KDF(Km_1 \,||\, \ldots \,||\, Km_n)$
- Corresponding s_i is secure if attacker Eve can not eavesdrop on all paths and obtain cleartext of all Km_j
- Eve can eavesdrop on a path if she
  - knows the TEK of a "link" or
  - compromised at least one involved gateway



→ Hope: An attacker can neither eavesdrop on every path nor compromise gateways on every path

→ Hence, the more paths we use over time the better, but: layer 1 path diversity will always be limited...

The SPIRIT of science TECHNISCHE UNIVERSITÄT ILMENAU

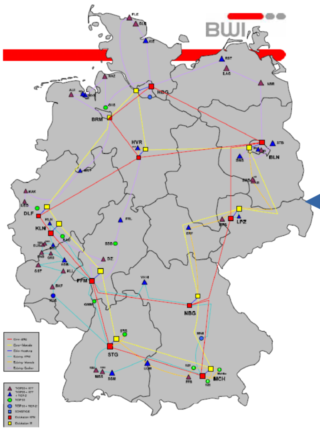
Multipath Key Reinforcement (2)

MKR on a physical network

- Very limited path diversity
- Hence, MKR security gain limited

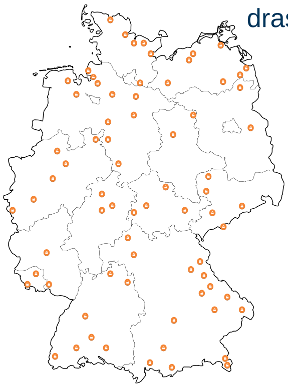
MKR inside an VPN overlay

- VPN introduces overlay view, independent from layer 1
- MKR inside an overlay will be at least as diverse as the underlay
- But, secure SAs can be seen as additional virtual "secure links". Hence, the "path diversity" in overlay drastically increases



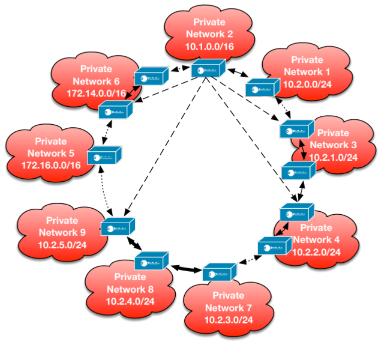
Layer 1

Underlay



Layer 3

Overlay



Bericht der Bundesregierung zur „Gesamtstrategie IT-Netze der öffentlichen Verwaltung" 2013

Standortquellen: Wikipedia

Multipath Key Reinforcement (3)

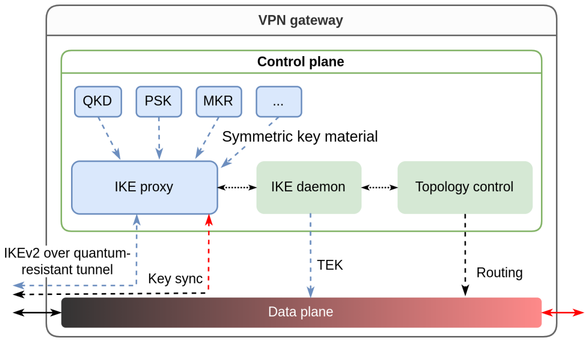
Current implementation and evaluation progress

- Prototypical implementation of MKR protocol (randomized path discovery and path selection)
- Standalone discrete event simulation of the same MKR protocol for "post-mortem" analyses
- First simulation results
  - VPN overlay topology: Chord ring with 256 gateways
  - Each node runs MKR every 10 minutes, IKE rekey happens every 20 minutes
  - Attack model: Static attacker that initially compromised all SAs except a spanning tree
    → Between every pair of nodes, there initially exists at least one secure path
  - Over time, MKR is able to find and use secure paths to secure previously insecure SAs
    → See video

Remaining question: How to securely combine MKR, QKD, business trip, postal key exchange?

→ Schatz, David; Altheide, Friedrich; Koerfgen, Hedwig; Rossberg, Michael; Schaefer, Guenter:
Virtual Private Networks in the Quantum Era: A Security in Depth Approach. SECRYPT, 2023. Accepted and in press. Preprint.

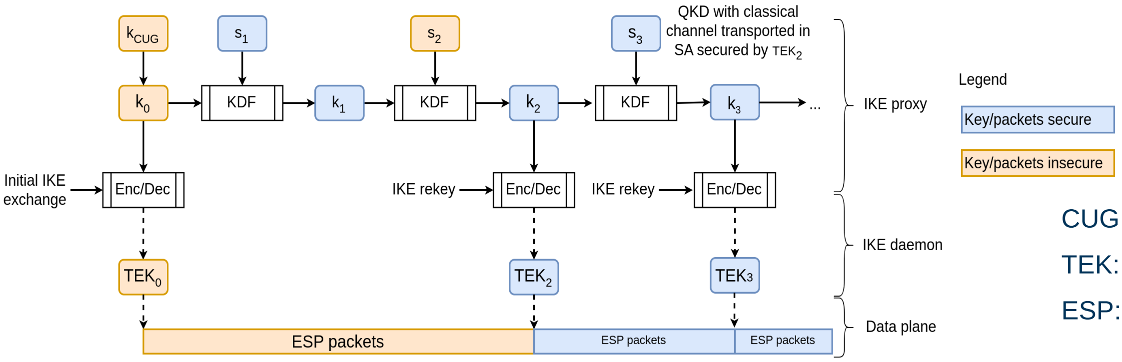
Combining MKR, QKD and More PSK Sources: IKE Proxy



- **Idea**: Use MKR, QKD, … PSKs to securely tunnel IKE key exchanges
  - As long MKR, QKD, ... PSKs are not compromised: No attacks on IKE possible
  - If IKE proxy “fails”: TEK still protected by IKE (PQC + classical in hybrid)
- Flexibility regarding sources of symmetric keys: QKD, Closed user group key (CUG), pairwise pre-shared keys (PSK), multipath key reinforcement (MKR), …
- Opportunity: Tunnel classical channel of QKD devices via VPN gateway → reduce QKD device attack surface!
- Next step: How to securely combine PSKs from different key sources?

Combining PSKs From Multiple Key Sources

Deriving and using pairwise (Alice and Bob) keys k_i inside IKE proxy



CUG: Closed User Group

TEK: Traffic Encryption Key

ESP: Encapsulated Security Payload (IPsec)

Note

- Assumption: PQC used by IKE does not hold, e.g., due to flawed implementation
- s_i can be from any symmetric key source (QKD, MKR, …)
- Simple key synchronization protocol ensures that k_i stay in sync at Alice and Bob, by using unique key identifiers for each s_i

Summary (of our Approach)

The IKE proxy implements a quantum-resistant tunnel for IKE by

- continuously combining symmetric keys from different sources to a pairwise master key for each remote proxy,
- keeping the master key in sync, and
- using the master key to protect every IKE packet (encryption and data integrity protection)

→ Data plane TEK secure if latest IKE proxy master key secure or IKE secure (e.g., PQC)

Orthogonal approach for key exchange with forward secrecy: MKR

- Over time, existing QKD paths (hop-by-hop secured by QKD) will be used (compare QKD networks)
- Supported by including additional PSKs exchanged via offline means (“business trips”, etc.)
- Enables to quickly create a quantum-attacker-secure overlay as soon as a spanning tree of secure SAs exists!

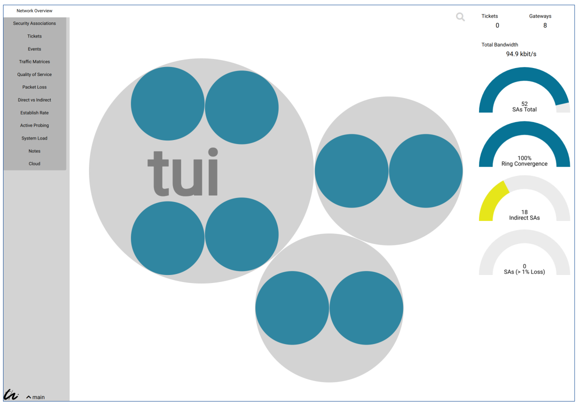
Intermediate Summary

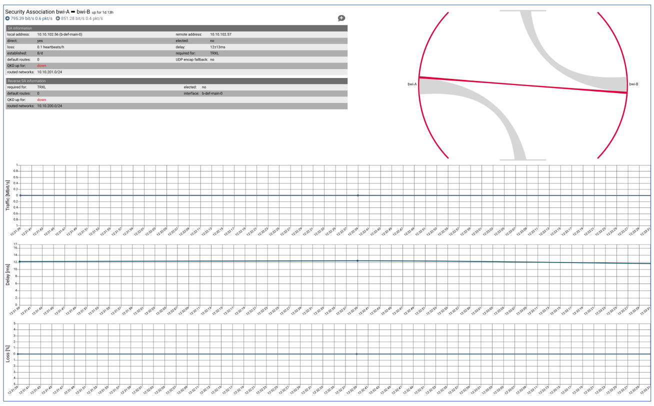
- **Secure every connection with:**
  - Classical asymmetric cryptography (IKEv2) → Attackers require quantum computers
  - PQC (IKEv2) → Attackers need to break PQC
  - CUG (overcrypt IKEv2) → Attackers need to compromise one VPN gateway to retrieve the CUG key
  - MKR inside VPN overlay → Attackers need to continously intercept every SA of the VPN ever established
- **Additionally secure some connections with:**
  - QKD → Attackers have to break the first authentication of QKDs classical channels, otherwise the keys will ensure perfect forward secrecy
  - Pairwise PSKs (at least required for QKD) → Attackers need to compromise the manual PSK exchange or each VPN gateway
  - Business trips → Attackers have to compromise all “traveling” workstations
  - Key exchanged via smart cards over postal service → Attackers have to compromise all exchanged smart cards

→ PQC, classical cryptography, MKR, QKD, … together will secure VPNs in times of quantum computers even from very powerful attackers (including “nation state”-type attackers)

Monitoring of Highly Scalable VPNs (StreamDB)

- We participate in a testbed which has a monitoring interface
  - State of VPN (direct / indirect SAs, throughput, network errors, …) https://telematik.prakinf.tu-ilmenau.de/solidmon/?vpn=main#network_overview
  - But also QKD (state and uptime of QKD links)
- Work in progress: QKD and MKR key rate, ...





Invitation

- **Please come visit us for in-depth discussions, we developed a comprehensive set of security protocols and implementations, parts of which are already being tested / deployed by some network operators**

- **SOLID:** Automated, distributed and highly robust VPN autoconfiguration for VPNs of any size (BSI approved)
  - State: Deployed on several thousand VPN gateways in Bavaria + some smaller deployments by other stakeholders
- **StreamDB:** Monitoring and ticketing system for large scale VPN
  - State: Deployed
- **HEAT:** Highspeed Encryption Acceleration Track (packet encryptor based on DPDK)
  - State: In productization and approved by BSI (July 2023)

- **SDN:** Distributed and resilient software defined networking SDN for large scale VPNs (State: Prototype)

Abbreviations

BSI: Bundesamt für Sicherheit in der Informationstechnik
CUG: Closed User Group
DPDK: Data Plane Development Kit
ESP: Encapsulating Security Payload
IKE: Internet Key Exchange
IPsec: Internet Protocol Security
KDF: Key Derivation Function
MACsec: Medium Access Conrol Security
MKR: Multipath Key Reinforcement
PKI: Public Key Infrastructure
PQC: Post Quantum Cryptography
PSK: Pre-Shared Key
SA: Security Association
SDN: Software Defined Networking
SOLID: Secure Overlay for IPsec Discovery
SPoF: Single Point of Failure
TEK: Traffic Encryption Key
VPN: Virtual Private Network

References (1)

[BBB+97]	Bennett, Charles H., et al. "Strengths and weaknesses of quantum computing." SIAM Journal on Computing. 1997.
[BW22]	Beullens, Ward. "Breaking rainbow takes a weekend on a laptop." Cryptology ePrint Archive. 2022.
[Gro96]	Grover, Lov K. "A fast quantum mechanical algorithm for database search." Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. 1996.
[NSA23]	NSA. "Quantum Key Distribution and Quantum Cryptography". https://www.nsa.gov/Cybersecurity/Quantum-Key-Distribution-QKD-and-Quantum-Cryptography-QC/ Accessed January 9, 2023.
[PZ03]	Proos, John and Zalka, Christof. "Shor's discrete logarithm quantum algorithm for elliptic curves". Quantum Information and Computing. 2003.
[Sho97]	Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM Journal on Computing. 1997.

References (2)

[ITU-T Y.3800]	ITU-T. “Overview on networks supporting quantum key distribution”. Version 1.1. 2020. https://www.itu.int/rec/T-REC-Y.3800/en
[ITU-T Y.3801]	ITU-T. “Functional requirements for quantum key distribution networks”. Version 1.0. 2020. https://www.itu.int/rec/T-REC-Y.3801/en
[ITU-T Y.3802]	ITU-T. “Quantum key distribution networks - Functional architecture”. Version 1.1. 2021. https://www.itu.int/rec/T-REC-Y.3802/en
[ITU-T Y.3803]	ITU-T. “Quantum key distribution networks - Key management”. Version 1.0. 2020. https://www.itu.int/rec/T-REC-Y.3803/en
[ITU-T Y.3804]	ITU-T. “Quantum key distribution networks - Control and management”. Version 1.0. 2020. https://www.itu.int/rec/T-REC-Y.3804/en
[ITU-T Y.3805]	ITU-T. “Quantum key distribution networks - Software defined networking control”. Version 1.0. 2021. https://www.itu.int/rec/T-REC-Y.3805/en

References (3)

[ETSI GS QKD 004]	ETSI. “Quantum key distribution (QKD); Application Interface”. Version 2.1.1. 2020. https://www.etsi.org/committee/1430-qkd
[ETSI GS QKD 014]	ETSI. “Quantum key distribution (QKD); Protocol and data format of REST-based key delivery API”. Version 1.1.1. 2019. https://www.etsi.org/committee/1430-qkd
[ETSI GS QKD 015]	ETSI. “Quantum key distribution (QKD); Control interface for software defined networks”. Version 2.1.1. 2022. https://www.etsi.org/committee/1430-qkd
[ETSI GS QKD 018]	ETSI. “Quantum key distribution (QKD); Orchestration interface for software defined networks”. Version 1.1.1. 2022. https://www.etsi.org/committee/1430-qkd

References (4)

[CVE-2014-0160]	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
[CVE-2021-44228]	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228