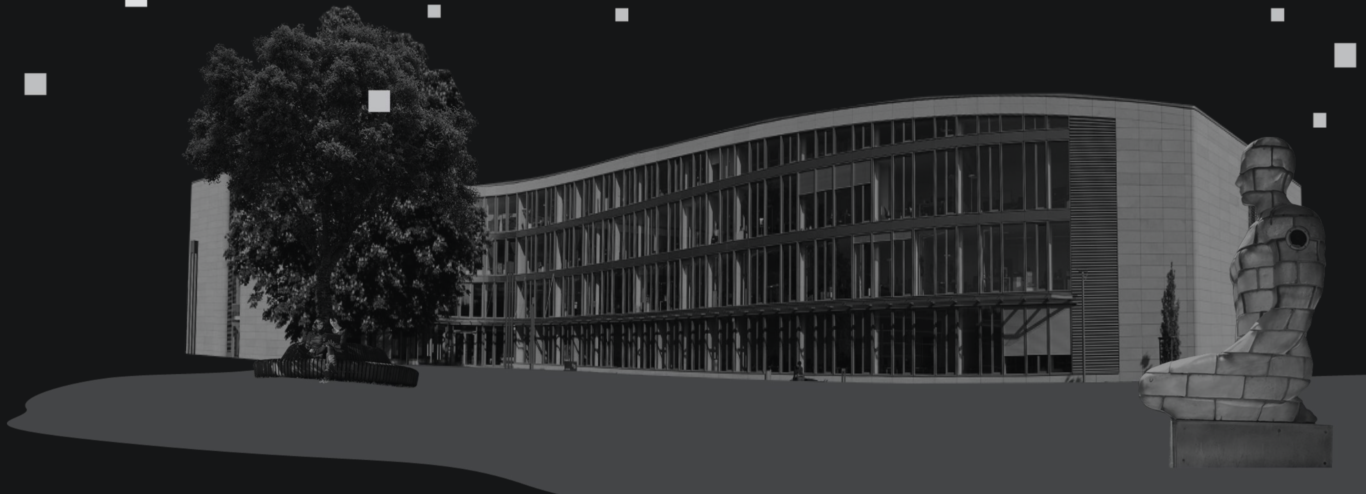
Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Quantum Key Distribution for Railway

Initial Presentation

Sebastian Staib



Design IT.
Create Knowledge.

www.hpi.de

Motivation

- Quantum Computers (QC) that can break currently used asymmetric crypto
  - are expected to be available in ~10 years. [1]
- Shor's algorithm enables prime factorization and the calculation of discrete logarithms (in cyclic groups) in polynomial [2]
  - Breaks the security of several public-key methods used today:
    - RSA [3]
    - Diffie–Hellman (ECDH, DSA/ECDSA) [4]
- Adversaries can record encrypted communications for encryption once QC become available ("*Harvest Now, Decrypt Later*") [5]

[1] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Quantencomputer/Entwicklungstand_QC_V_2_1.pdf?__blob=publicationFile&v=3

[2] https://epubs.siam.org/doi/10.1137/S0097539795293172

[3] https://doi.org/10.22331/q-2021-04-15-433

[4] https://doi.org/10.6028/NIST.IR.8547.ipd

[5] https://english.aivd.nl/binaries/aivd-en/documenten/publications/2022/01/18/prepare-for-the-threat-of-quantumcomputers/Prepare+for+the+threat+of+quantumcomputers.pdf

Two Approaches to Quantum-Secure Systems

HPI

- Post-Quantum Cryptography (PQC):
  - Based on mathematical problems
  - Resistant to both conventional and quantum cryptanalysis [6]
  - Standards are available since August 2024, by NIST: Kyber, Dilithium and SPHINCS+ [7]

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE

- Quantum Key Distribution (QKD):
  - Two parties establish a shared secret key by encoding information in quantum states [8]
  - Generates information-theoretically secure key material that can be used to generate perfectly secure ciphertexts, e.g. One-Time-Pad [9]

[6] https://doi.org/10.2824/92307
[7] https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards
[8] https://doi.org/https://doi.org/10.1016/j.tcs.2014.05.025
[9] https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

QKD Architecture

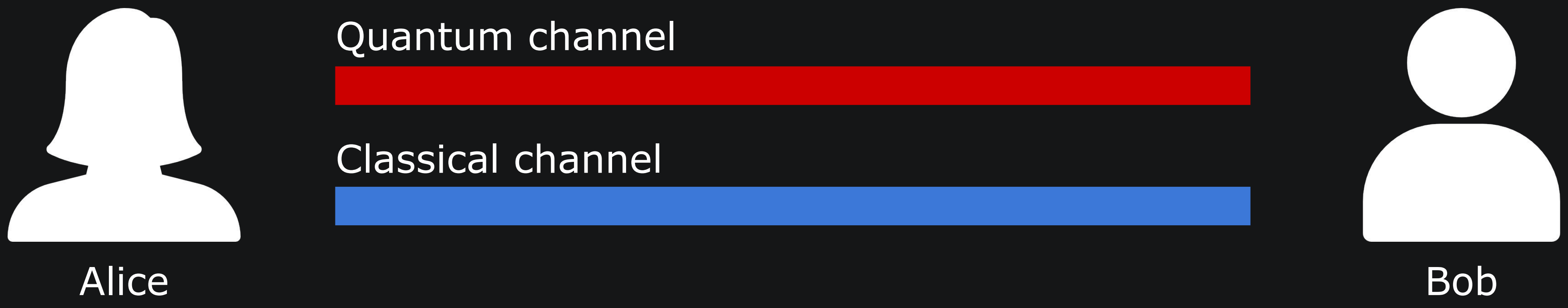
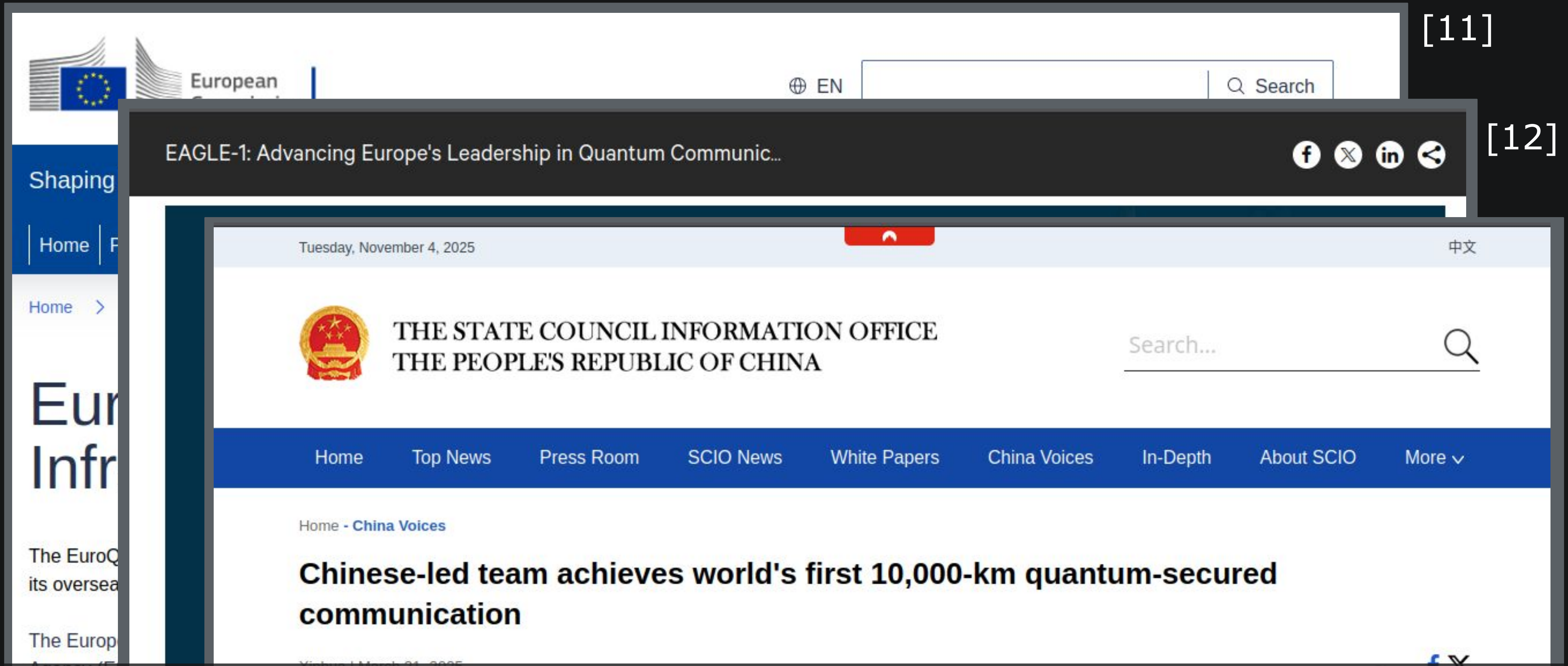


Fig 1. QKD setting. Alice and Bob are connected through a quantum channel and a classical channel [10]

[10] based on Fig 1 in https://doi.org/10.1103/RevModPhys.81.1301

Recent QKD Developments



[11]

[12]

[13]

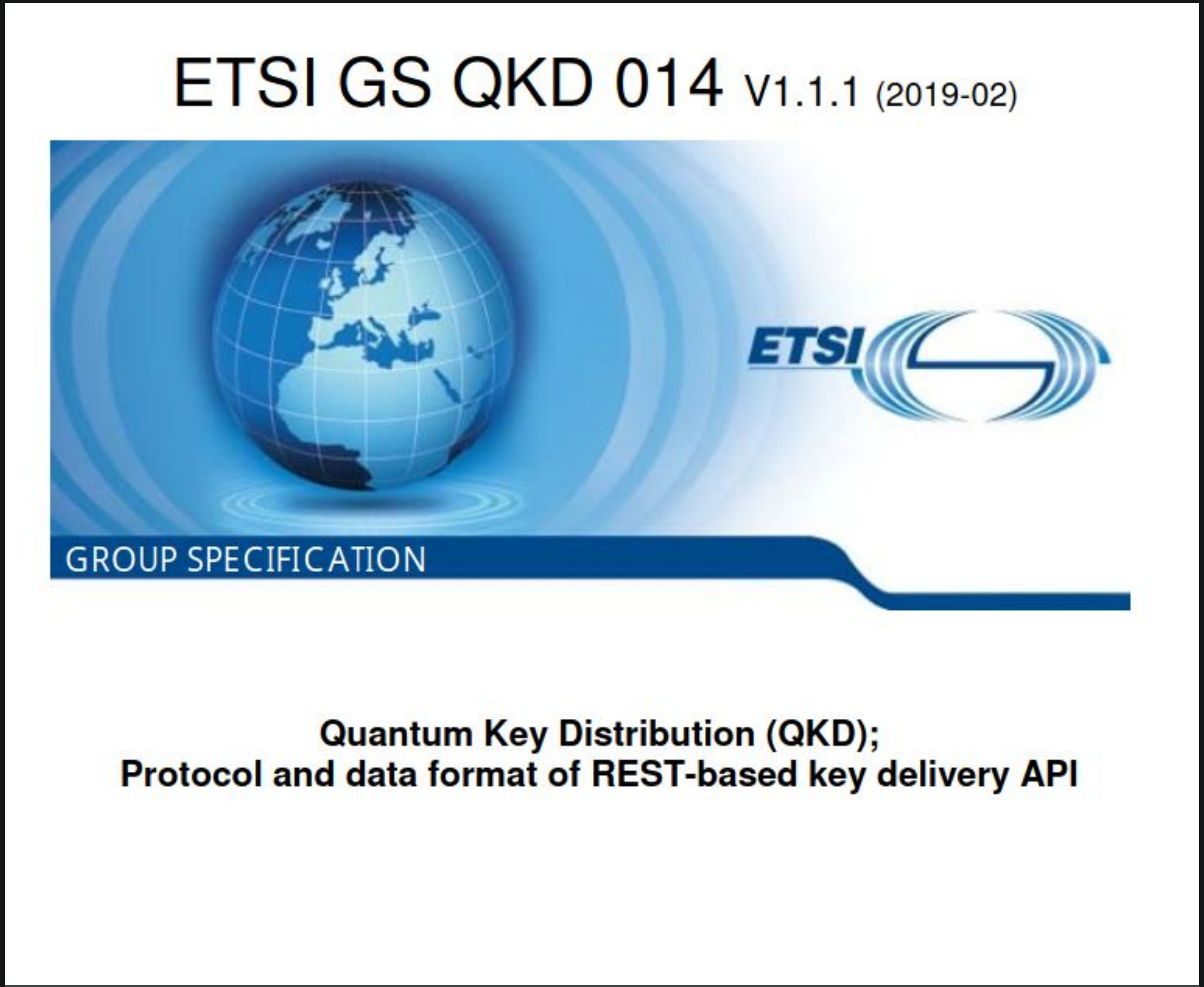
[11] https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci
[12] https://www.ses.com/newsroom/eagle-1-advancing-europes-leadership-quantum-communications
[13] http://english.scio.gov.cn/chinavoices/2025-03/21/content_117778711.html

Content of the Thesis

- Discuss the theoretical and practical foundations of QKD
- Introduce architecture of QKD Networks (QKDN)
- Develop a taxonomy of the various protocols and standards
  - QKD protocols: BB84, E91 and BBM92
  - Different types of QKD (CV-QKD vs. DV-QKD)
  - Standards: *ETSI GS QKD 014* [14] and *ITU-T Y.3800* [15]
- State of the art and related work
- Proof of concept and software architecture
  - Implementation for use case “Schlüsseltankstelle” using the above standards

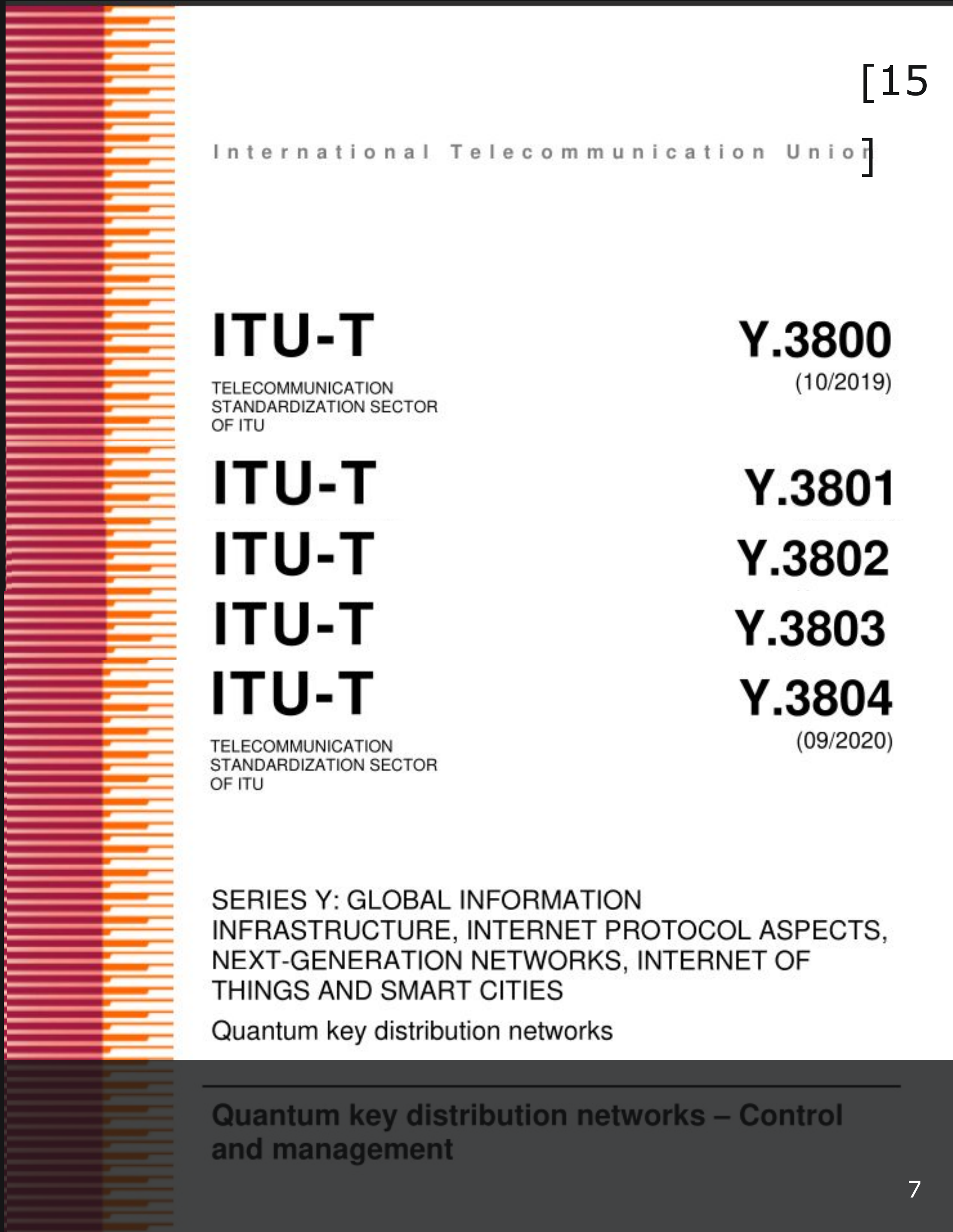
[14] https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pdf
[15] https://www.itu.int/rec/T-REC-Y.3800

Standards and Reference Architecture[14]


ETSI GS QKD 014 V1.1.1 (2019-02)

Quantum Key Distribution (QKD);
Protocol and data format of REST-based key delivery API

[15]


International Telecommunication Union

ITU-T **Y.3800**
TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU (10/2019)

ITU-T **Y.3801**

ITU-T **Y.3802**

ITU-T **Y.3803**

ITU-T **Y.3804**
TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU (09/2020)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Quantum key distribution networks

Quantum key distribution networks – Control and management

[14] https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pdf
[15] https://www.itu.int/rec/T-REC-Y.3800

Standards & reference architecture

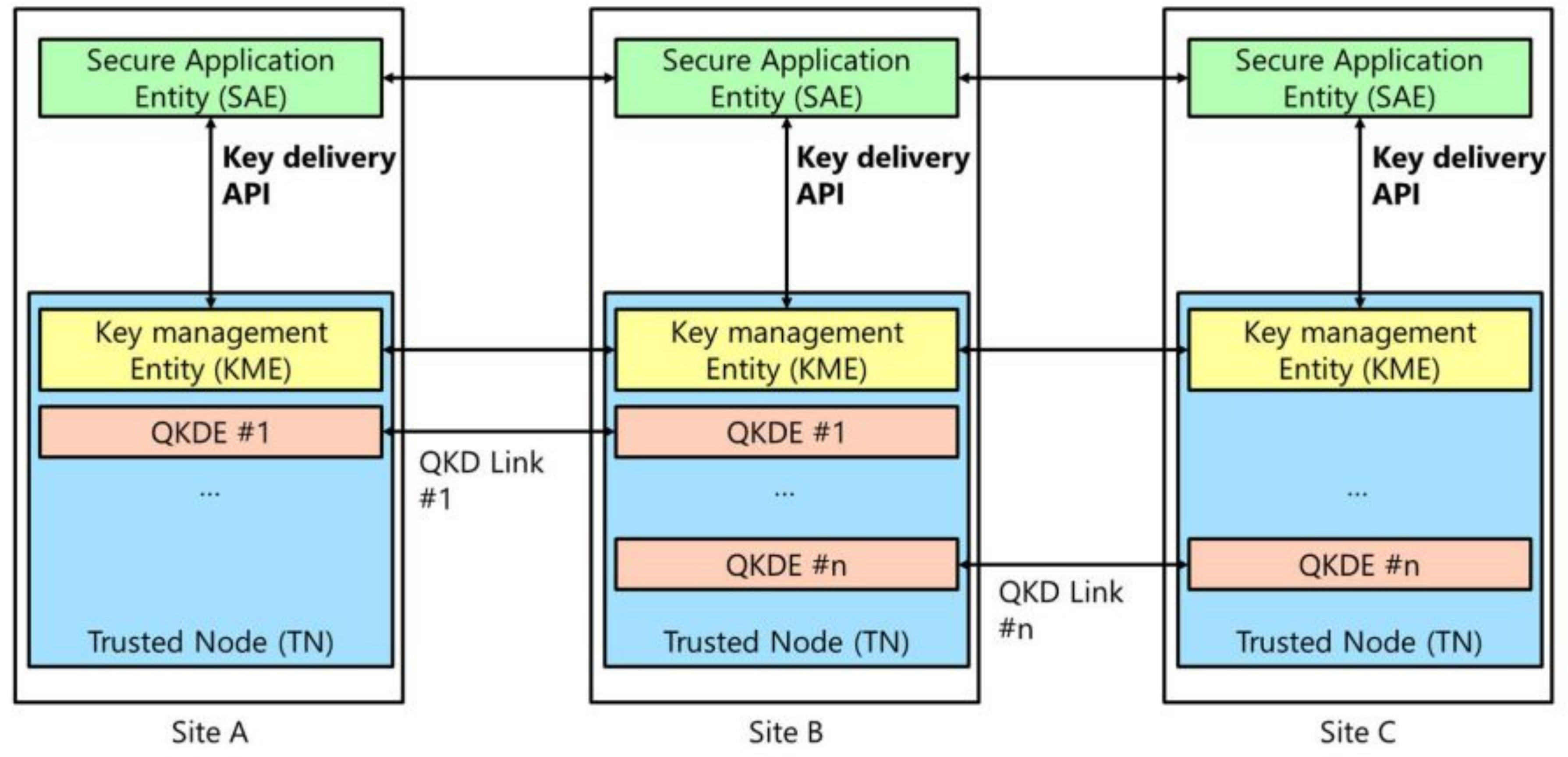


Figure 1: Example of QKD network

[14]

[14] https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pdf

Standards & reference architecture

[15]

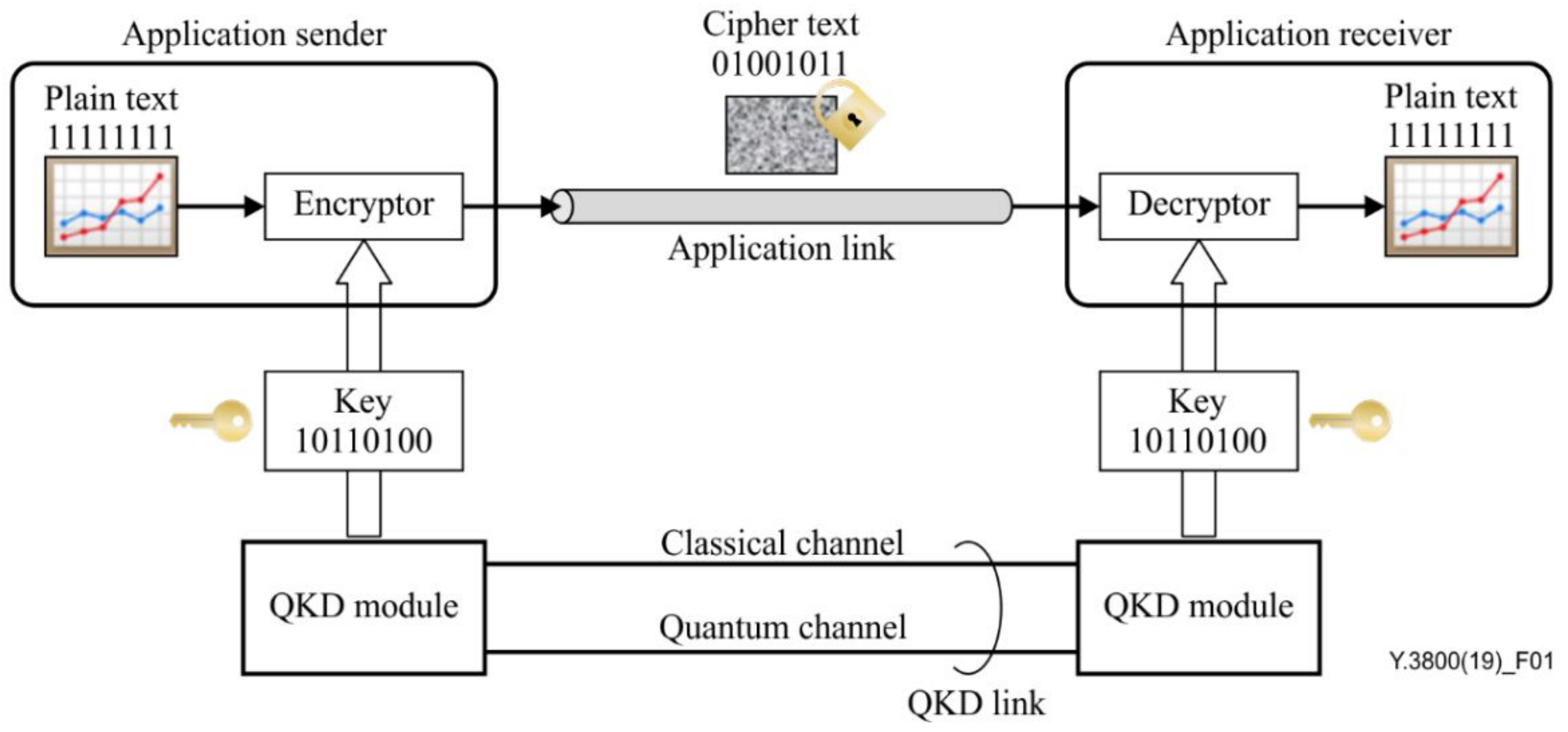
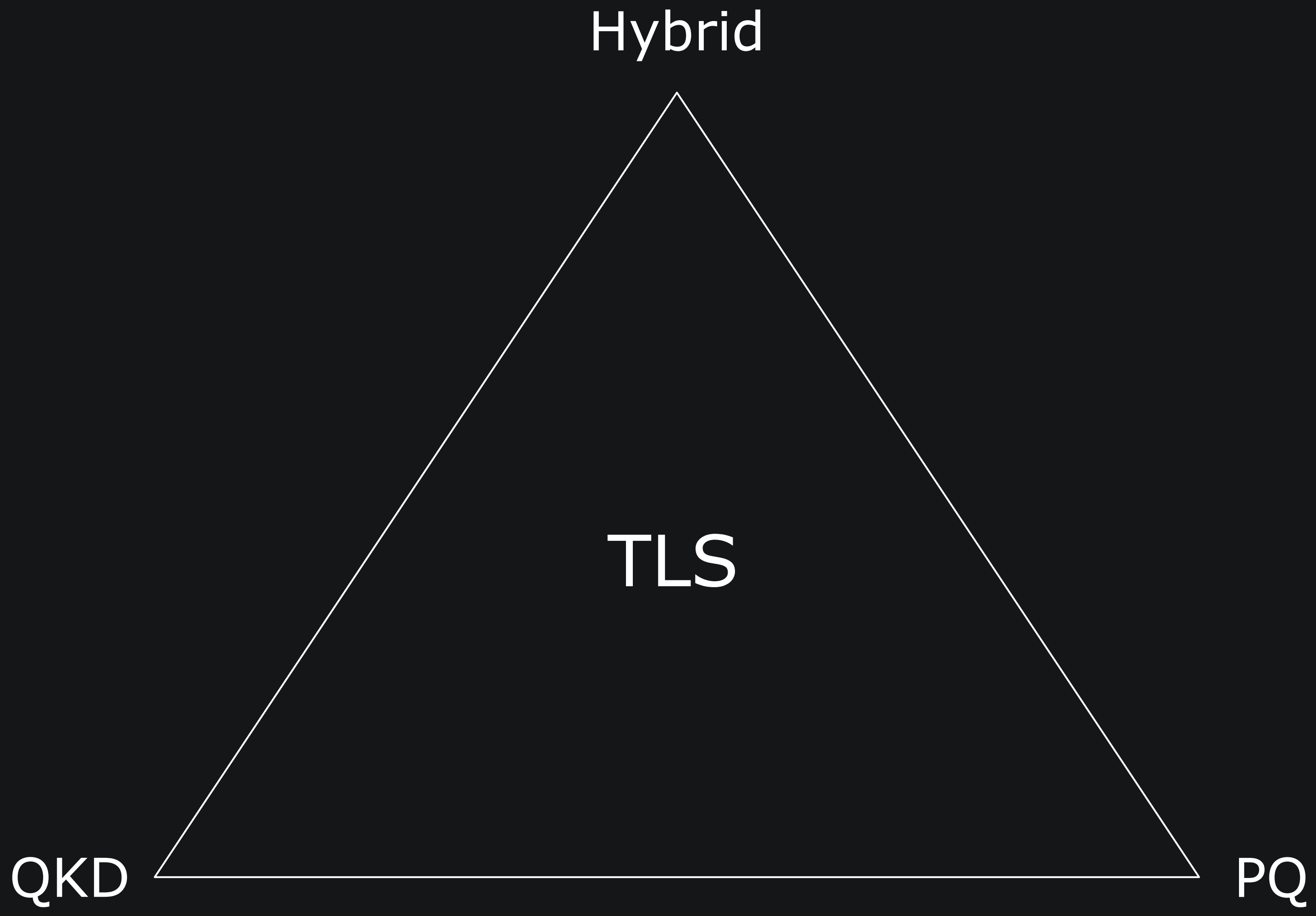


Figure 1 – Configuration example of QKD use for securing a P-to-P application link

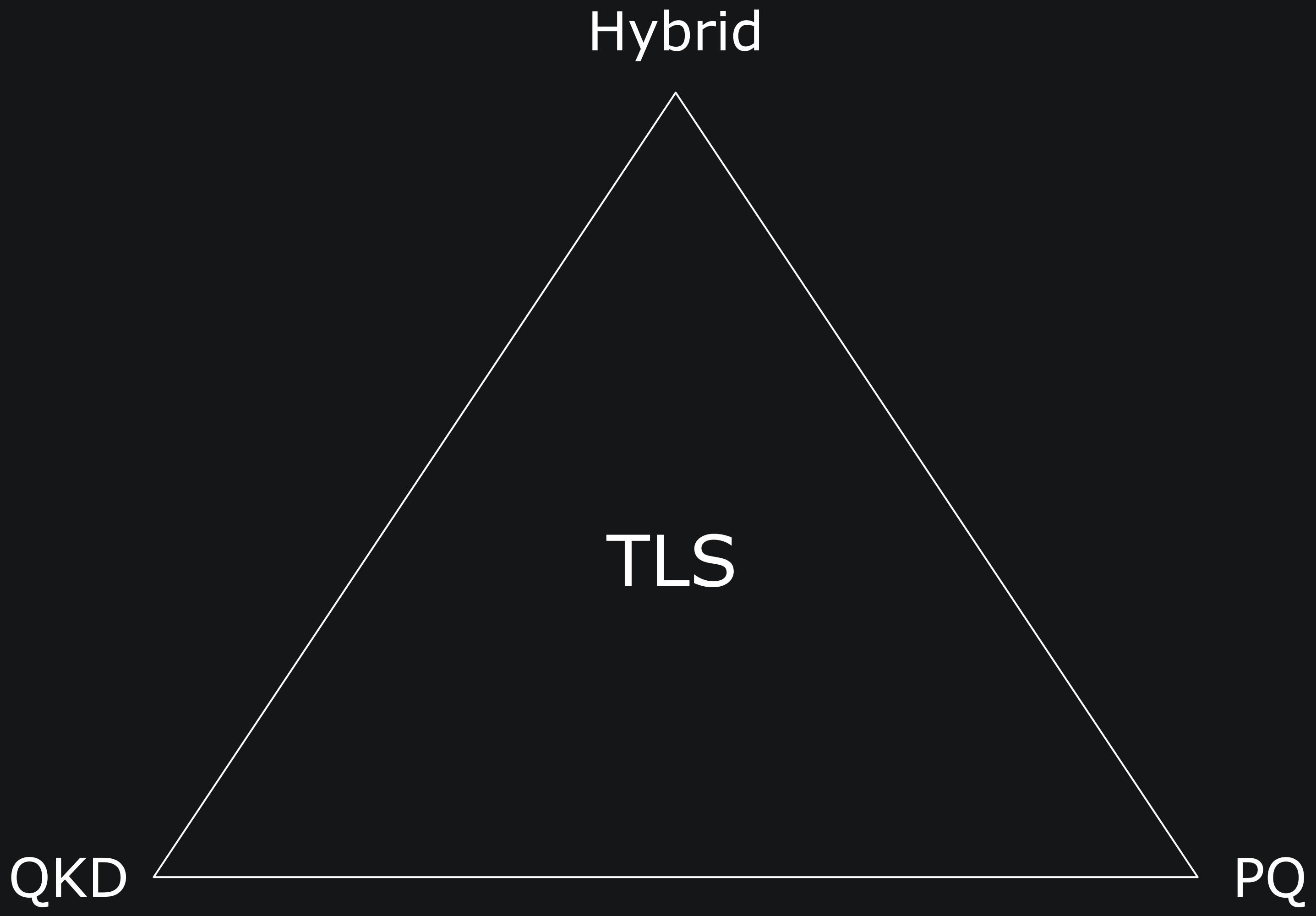
[15] https://www.itu.int/rec/T-REC-Y.3800

TLS Post-Quantum Approaches (1)



TLS Post-Quantum Approaches (5)

- Evaluation criteria:
  - Security model
  - QC-resistant
  - Handshake/latency behavior
    - mobile contexts
  - Deployability & maturity

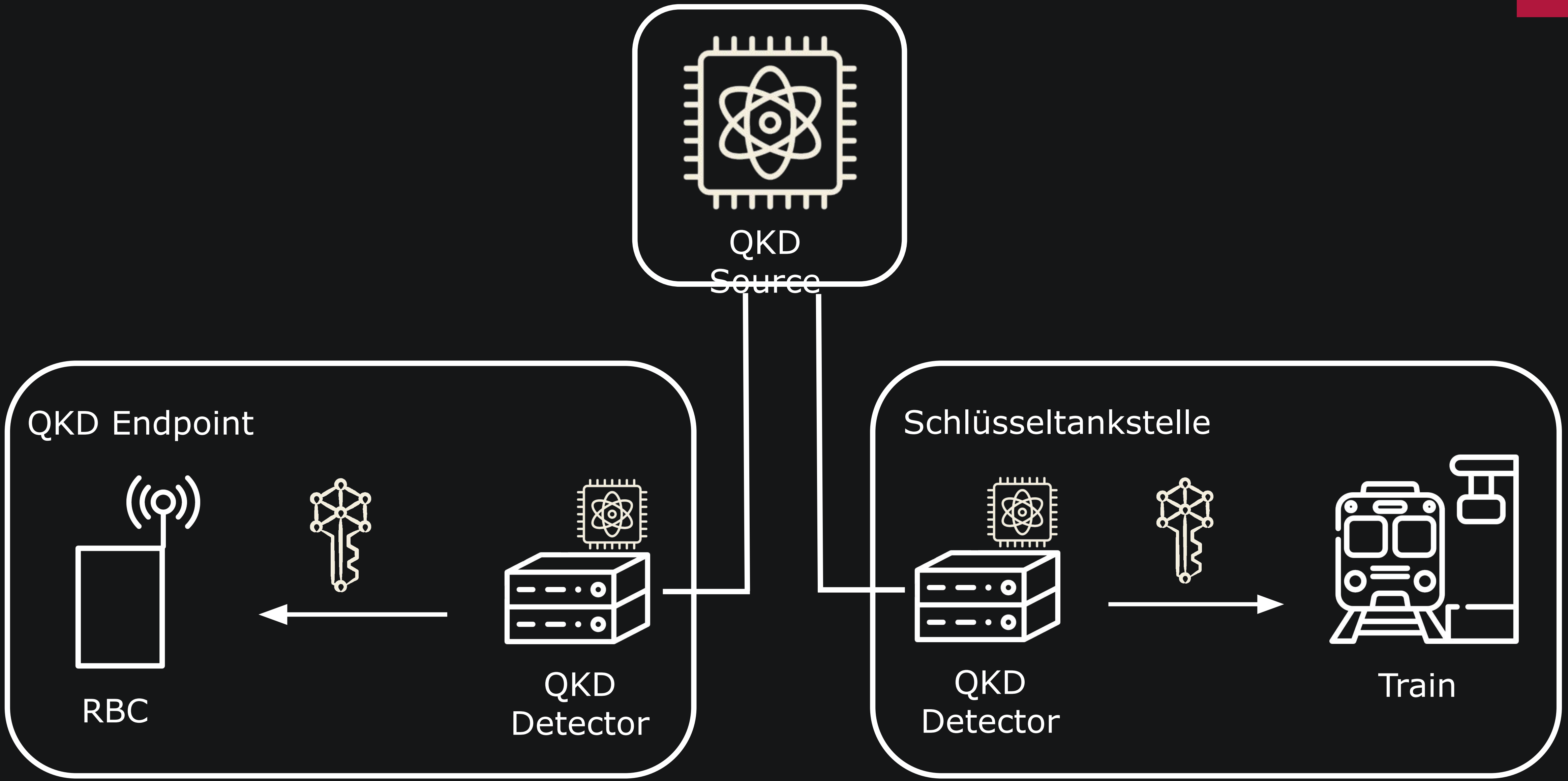


TLS Post-Quantum Approaches (6)

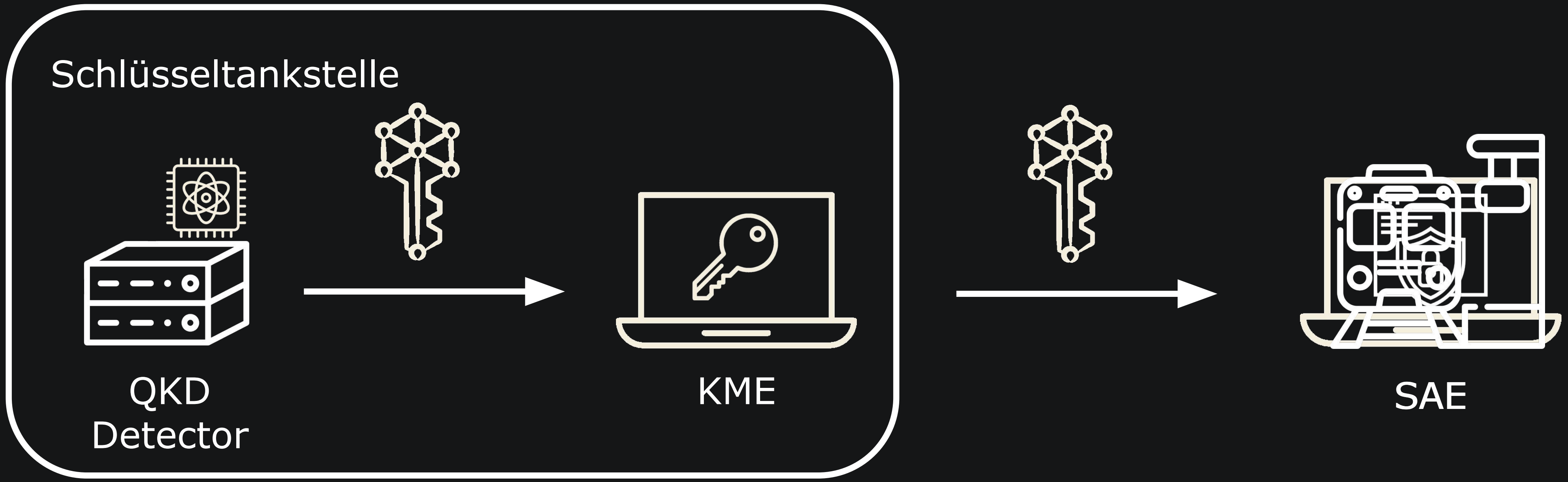
Table4

Scheme / Property	TLS 1.3 Classical approach as described in RFC 8446	TLS-QKD (as described in 2025_Prévost_An ETSI GS QKD compliant TLS)	KEMTLS (as described in 2022_Schwabe_Stebila_Wiggers_Post-quantum_TLS_without_handshake_signatures)	KEM-QKD (as discussed in 2025_Blanco_Romero_et_al._-_QKD-KEM_-_Hybrid_QKD_Integration_into_TLS_with_OpenSSL_Providers)
Summary	Integrity and Confidentialty can be reached, computational security server **authenticity** by default (client is optional), **confidentiality** of application data and the encrypted parts of the handshake. **Integrity** of both the handshake (via transcript-bound MAC/signatures) and application data (via AEAD record protection) [4] Secure channel over a reliable stream; server is always authenticated, client optional. [5]	Plug quantum keys from ETSI GS QKD 014 (KME/SAE architecture) into ordinary TLS so that endpoints can use QKD-provided entropy without breaking the web-PKI/TLS stack.	uses key-encapsulation mechanisms (KEMs) instead of signatures for server authentication PKI certifies KEM keys; same RTT; smaller/faster than PQ-signed TLS.	Mix a normal post-quantum KEM (e.g., ML-KEM/Kyber) with QKD material so that the TLS 1.3 handshake stays compatible with OpenSSL while the final handshake secret is protected by both PQC and QKD. If either one holds, the session key is safe
Repository	-	https://github.com/qursa-uc3m/qkd-kem-provider	https://github.com/thomwiggers/kemtls-experiment	QKD-KEM Provider (OpenSSL 3): https://github.co QKD ETSI API (C wrapper): https://github.com/qu QKD-KEM Benchmarking Suite: https://github.com
Confidentiality	Application data **confidentiality** is provided by AEAD ciphers at the record layer; the handshake is encrypted from ServerHello onward, so identities and key shares exchanged after that point are confidential. TLS does not hide message lengths [4]	feeds QKD-derived symmetric keys into an otherwise standard TLS 1.3 record layer that uses AEAD (e.g., AES-256-GCM), so the channel's confidentiality remains computational (not information-theoretic). AES in AEAD mode with a 256-bit key ("hardcoded" key size in the prototype)	Similar to classical TLS 1.3	Similar to classical TLS 1.3
Availabilty	Replay Attacks: 0-RTT replay, where an attacker **duplicates** early-data flights or exploits inconsistent server-state/blocked ServerHello to **make the same request run multiple times,** overloading rate-limiters and backends. Attack works by replaying ClientHello/early data at scale, polluting or bypassing replay caches if misconfigured, and abusing client retry behavior in multi-zone deployments. [4]	Same as classical TLS 1.3	No difference from classical TLS 1.3	No difference from classical TLS 1.3
Integrity	yes [4]	yes, from AEAD just like clasiscal TLS 1.3	Uses AEAD	Uses AEAD
Forward Secrecy	yes [4]	yes [22]	Provided by ephemeral KEM exchange	Provided by ephemeral KEM exchange
		Yes, Possesion of the same QKD key proven with		

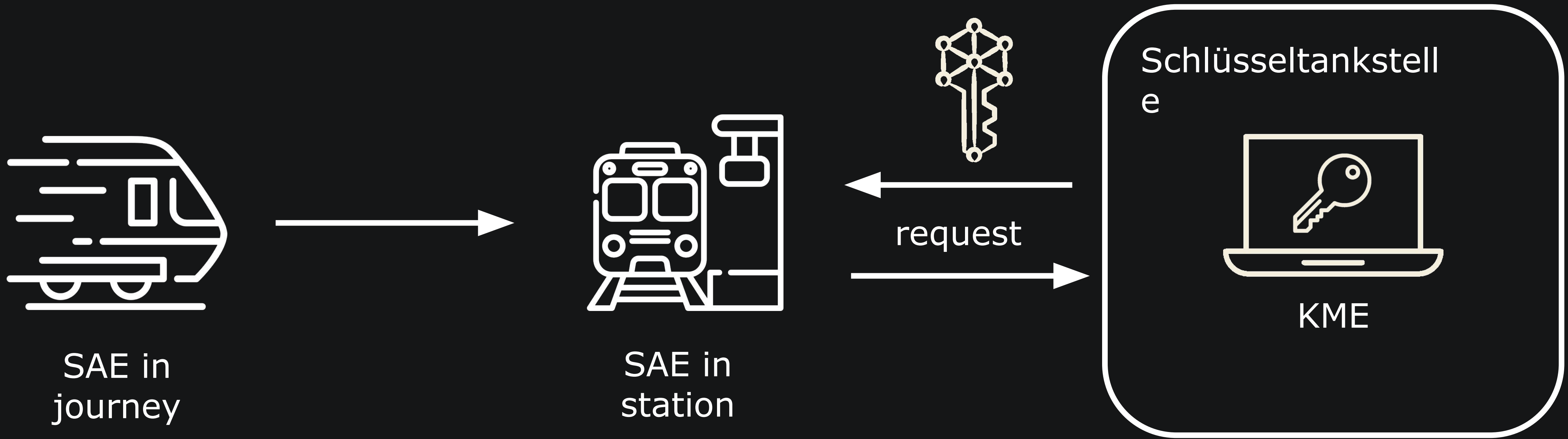
Use Case: “Schlüsseltankstelle” (1)



Use Case: “Schlüsseltankstelle” (2)

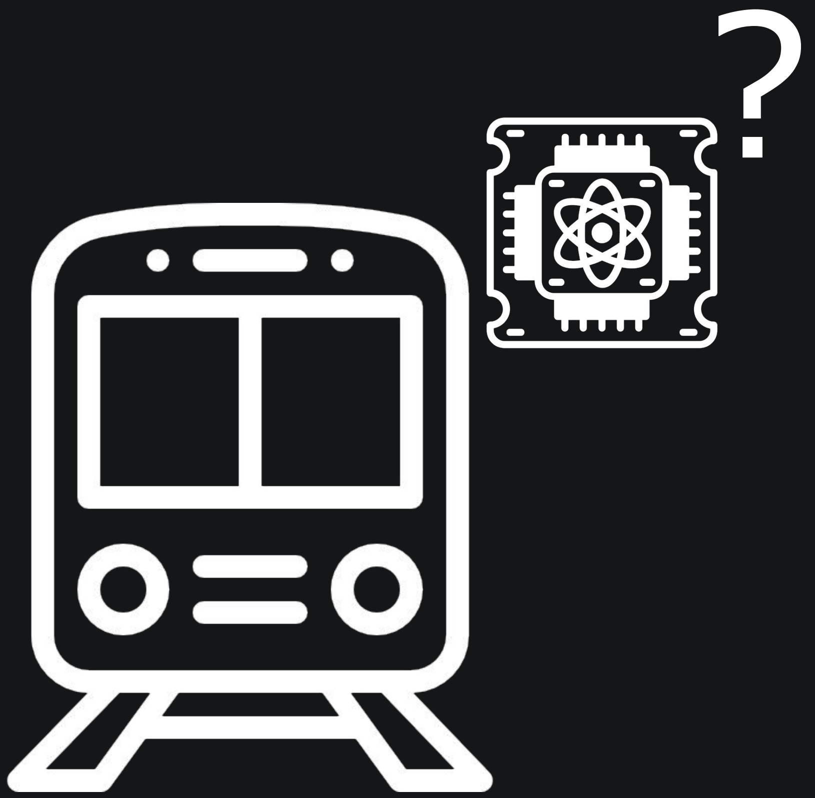


Use Case: “Schlüsseltankstelle” (3)



Use Case: “Schlüsseltankstelle” (4)

Open Questions



Quantum optics required on the train?

What about key storage capacity?

Source: Flaticon.com

Prototype Plan

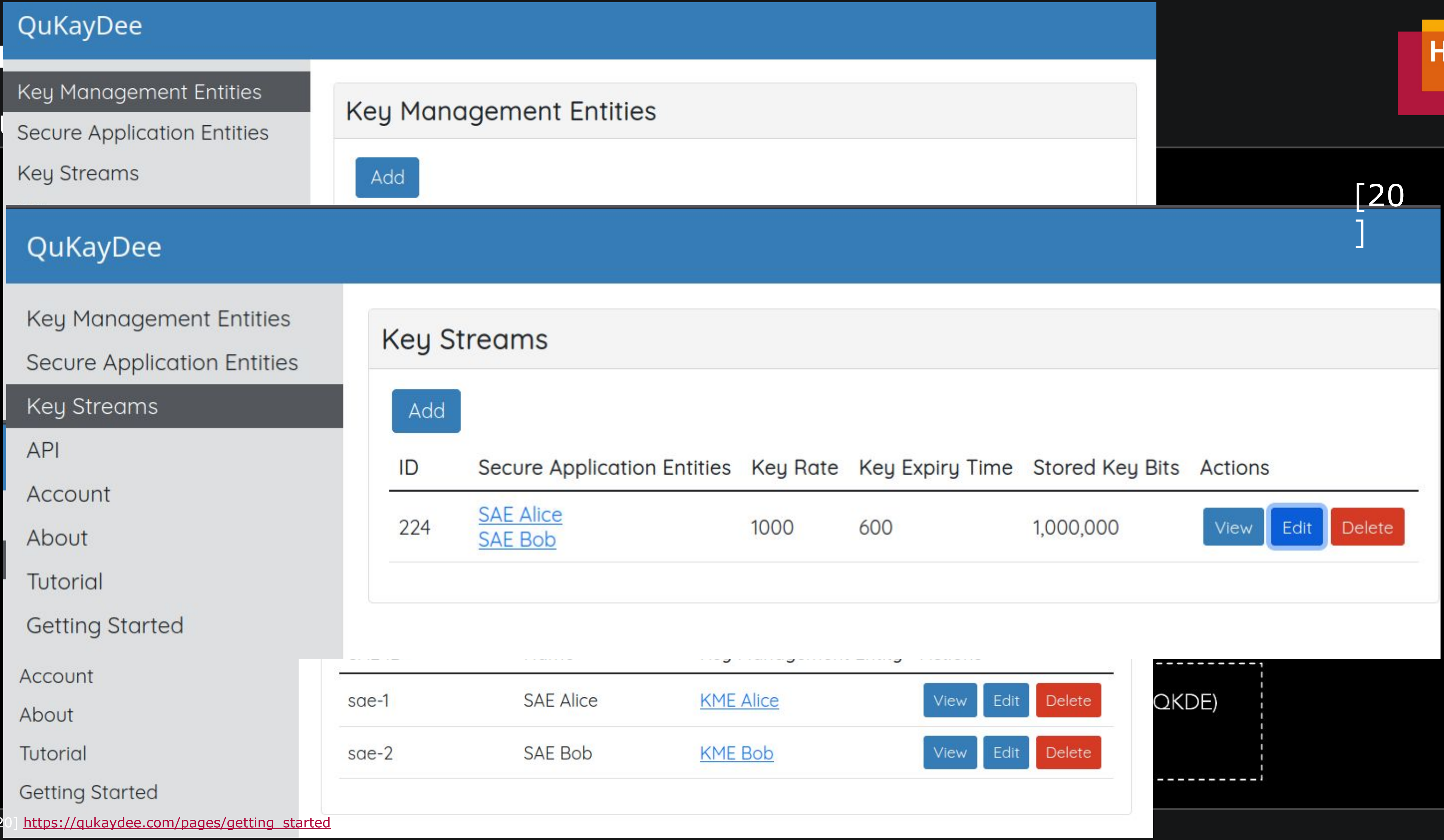
- Phase 1
  - API-level integration with “Hello World” and test harness using QKD key delivery interfaces (ETSI GS QKD 014 v1.1.1)
  - using Key Material from *QuKayDee* [19]
- Phase 2
  - Evalutiion phase of different TLS variants and decision based on evaluation criteria
- Phase 3
  - Prototype implementation with chosen variant for key management solution, specifically addressed to Mobile Use Case “Schlüsseltankstelle”.

[19] https://qukaydee.com/pages/about

Pr

Qu

[20]

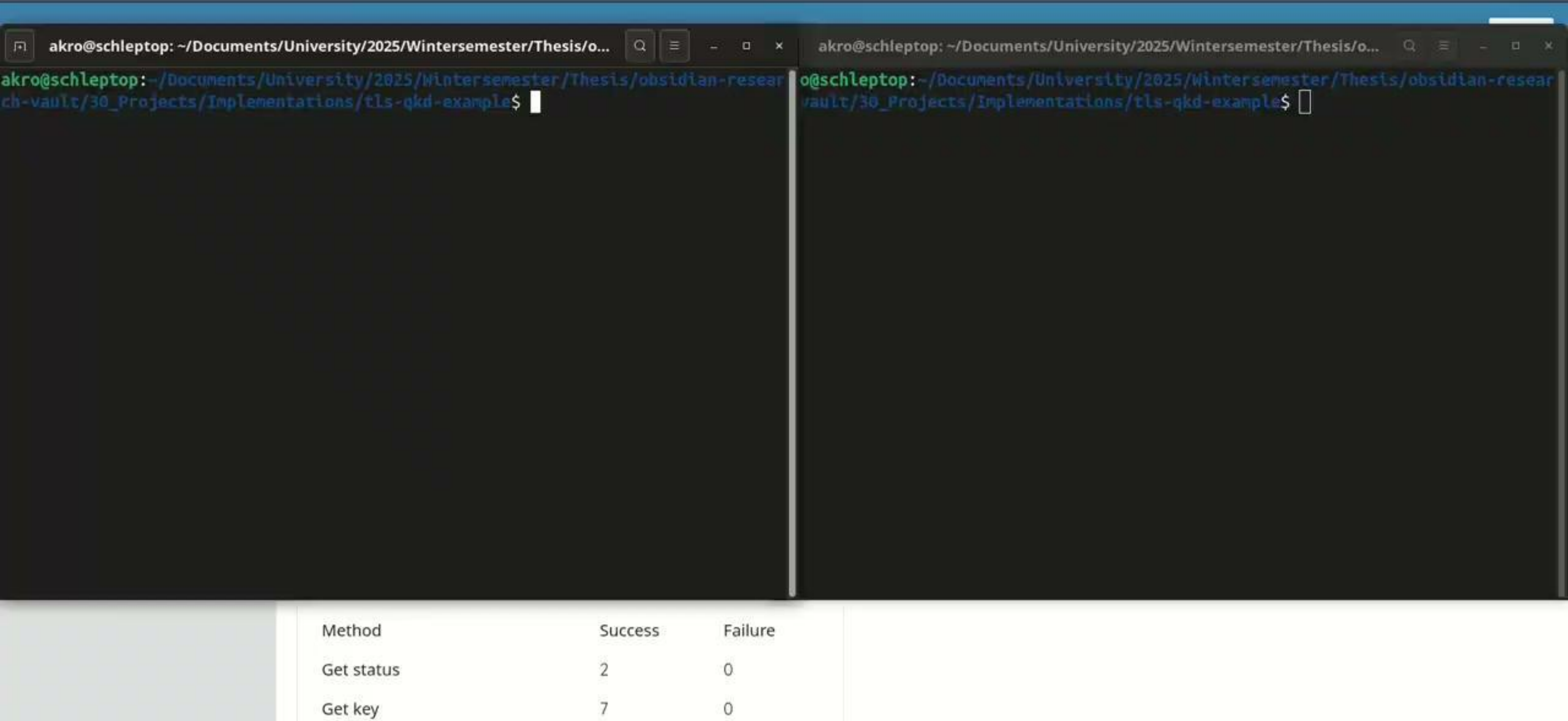


QKDE)

[20] https://qukaydee.com/pages/getting_started

Proof of Concept – Approach One (2)

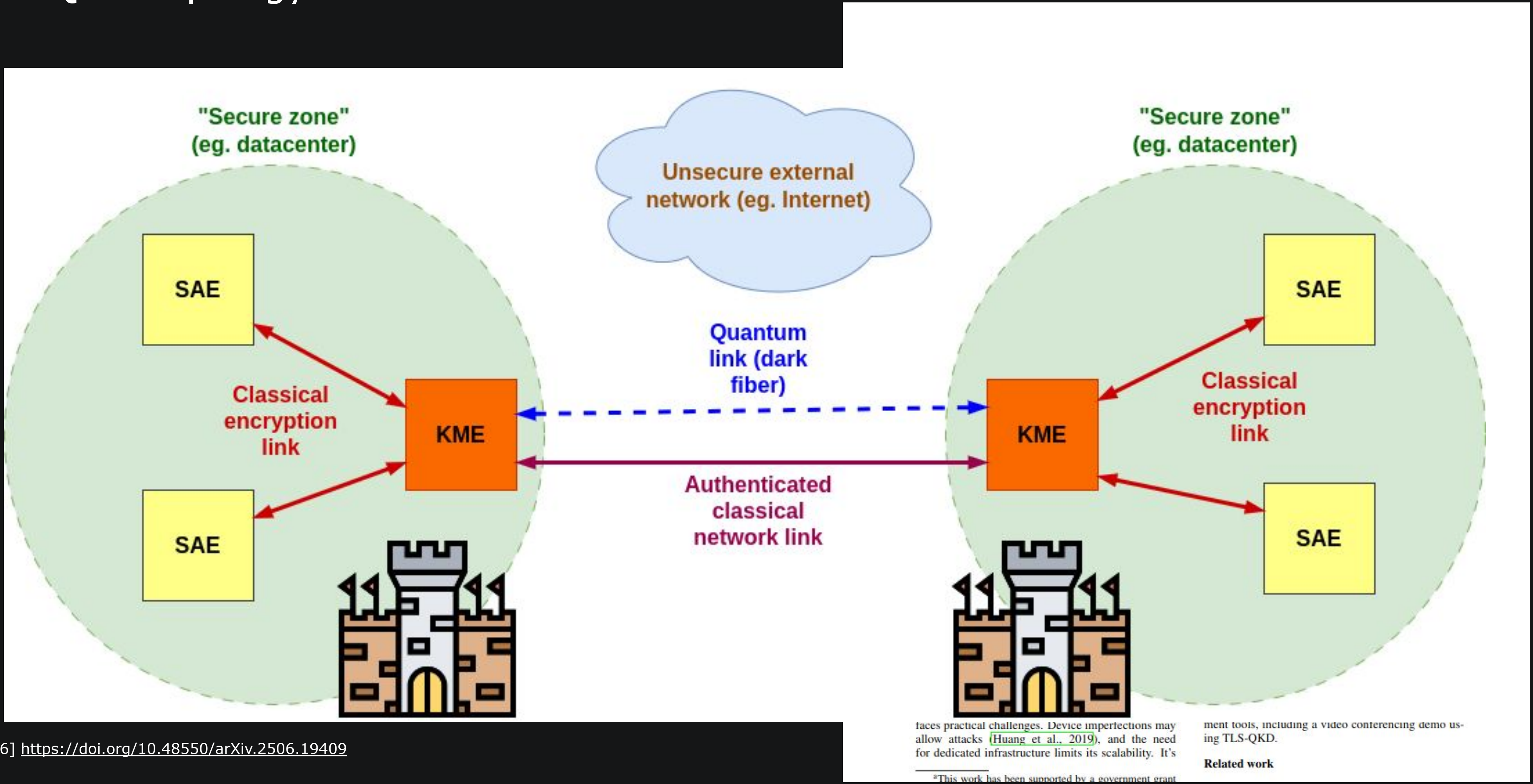
Video



Proof of Concept – Approach Two (1)

TLS-QKD Topology

[16]

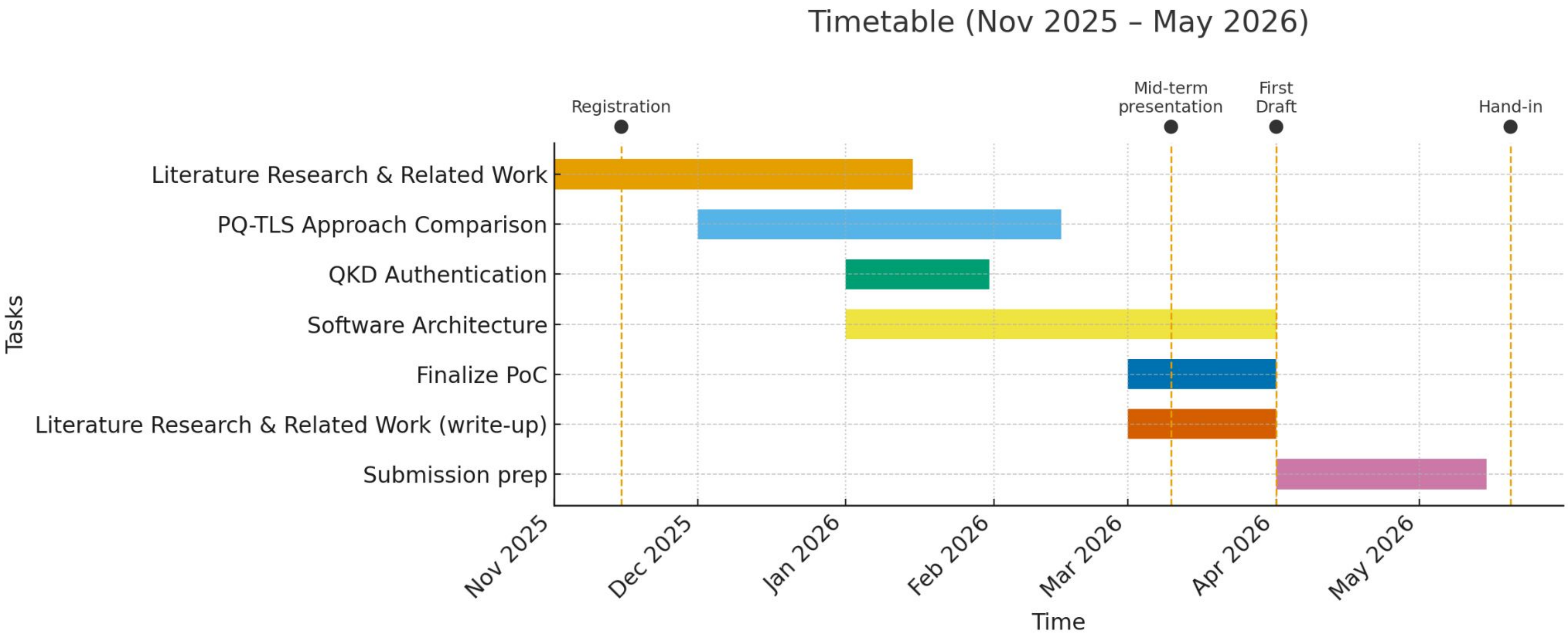


[16] https://doi.org/10.48550/arXiv.2506.19409

Proof of Concept – Approach Two (2)

Video

Timetable


Timetable (Nov 2025 – May 2026)
Registration
Mid-term presentation
First Draft
Hand-in
Literature Research & Related Work
PQ-TLS Approach Comparison
QKD Authentication
Software Architecture
Finalize PoC
Literature Research & Related Work (write-up)
Submission prep
Tasks
Nov 2025
Dec 2025
Jan 2026
Feb 2026
Mar 2026
Apr 2026
May 2026
Time

Working Packages

1. What are different post quantum secure alternatives for TLS? How do they perform compared to other approaches (e.g. KEMTLS)? *Compare on security, authentication, latency, deployment model, maturity level, hardware requirements, resistance to QC and the mobile use case "Schlüsseltankstelle".*

2. Trust Bootstrapping and Authentication: What is the abstract concept of authentica-tion in QKDN? *What approaches exist to establish trust between two parties? What are their differences? (PKI vs. PSK)*

3. Architecture mapping to the "Schlüsseltankstelle" Use Case: How can a high-level QKDN architecture be mapped to this use case? *Clarify roles, responsibilities, interfaces at a conceptual level, the policies that govern key use and distribution*

Backup Slides

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Design IT.
Create Knowledge.

www.hpi.de

Introduction

Master student: Cybersecurity

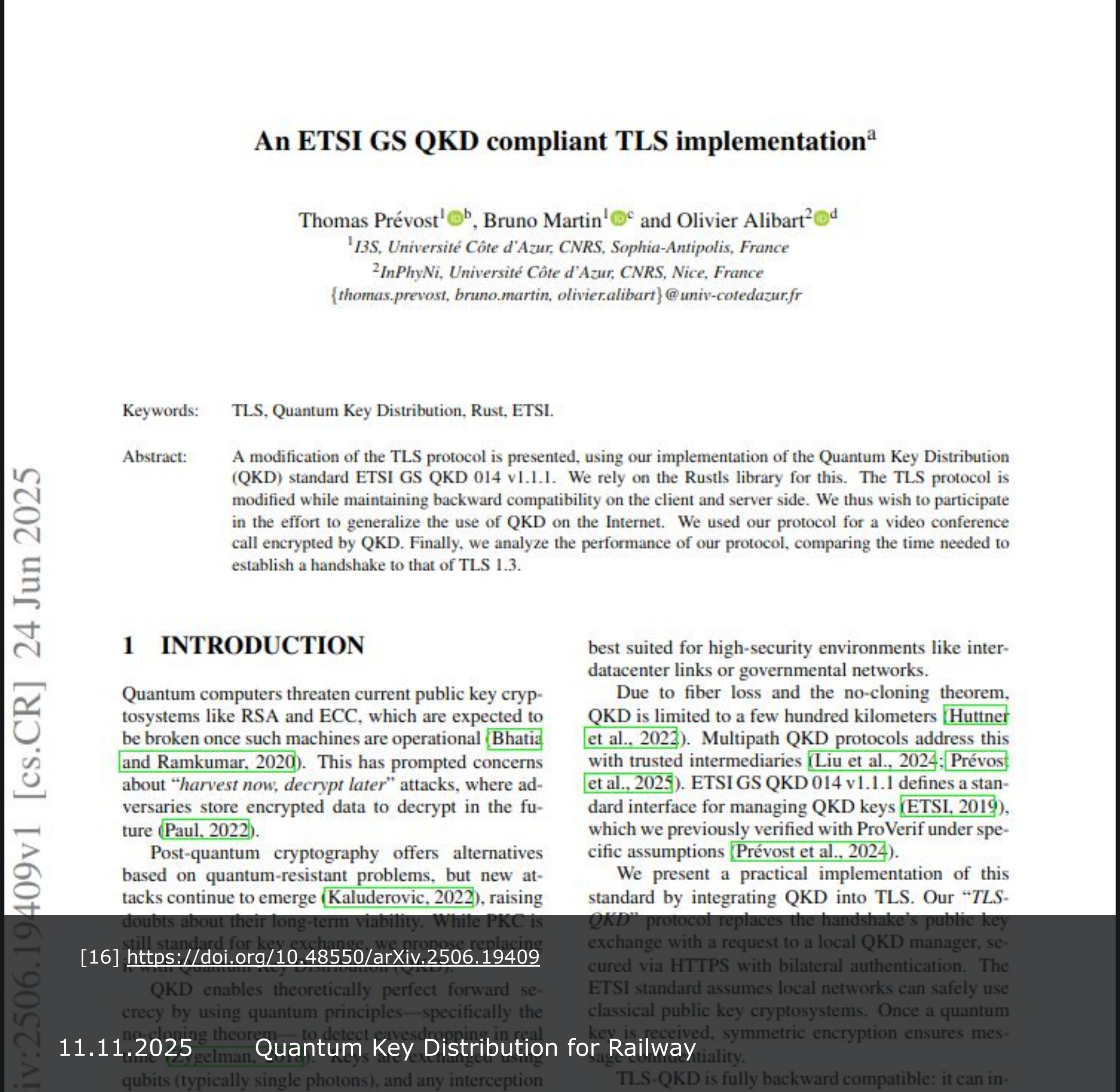
Work experience:

- Worked in IT-Consulting
  - “IT-Grundschutzpraktiker” (certified)
  - Incident response for ~1 year,
    Focus on crisis management

Academic Progress:

- Key areas of cybersecurity
  - Network and application security
- Focus on identity management and applied cryptography
- Thesis about Quantum Key Distribution

TLS Post-Quantum Approaches (2)



iv:2506.19409v1 [cs.CR] 24 Jun 2025

An ETSI GS QKD compliant TLS implementation[a]

Thomas Prévost[1][b], Bruno Martin[1][c] and Olivier Alibart[2][d]
[1]I3S, Université Côte d'Azur, CNRS, Sophia-Antipolis, France
[2]InPhyNi, Université Côte d'Azur, CNRS, Nice, France
{thomas.prevost, bruno.martin, olivier.alibart}@univ-cotedazur.fr

Keywords: TLS, Quantum Key Distribution, Rust, ETSI.

Abstract: A modification of the TLS protocol is presented, using our implementation of the Quantum Key Distribution (QKD) standard ETSI GS QKD 014 v1.1.1. We rely on the Rustls library for this. The TLS protocol is modified while maintaining backward compatibility on the client and server side. We thus wish to participate in the effort to generalize the use of QKD on the Internet. We used our protocol for a video conference call encrypted by QKD. Finally, we analyze the performance of our protocol, comparing the time needed to establish a handshake to that of TLS 1.3.

1 INTRODUCTION

Quantum computers threaten current public key cryptosystems like RSA and ECC, which are expected to be broken once such machines are operational (Bhatia and Ramkumar, 2020). This has prompted concerns about "*harvest now, decrypt later*" attacks, where adversaries store encrypted data to decrypt in the future (Paul, 2022).

Post-quantum cryptography offers alternatives based on quantum-resistant problems, but new attacks continue to emerge (Kaluderovic, 2022), raising doubts about their long-term viability. While PKC is still standard for key exchange, we propose replacing it with Quantum Key Distribution (QKD).

QKD enables theoretically perfect forward secrecy by using quantum principles—specifically the no-cloning theorem— to detect eavesdropping in real time (Zygelman, 2018). Keys are exchanged using qubits (typically single photons), and any interception

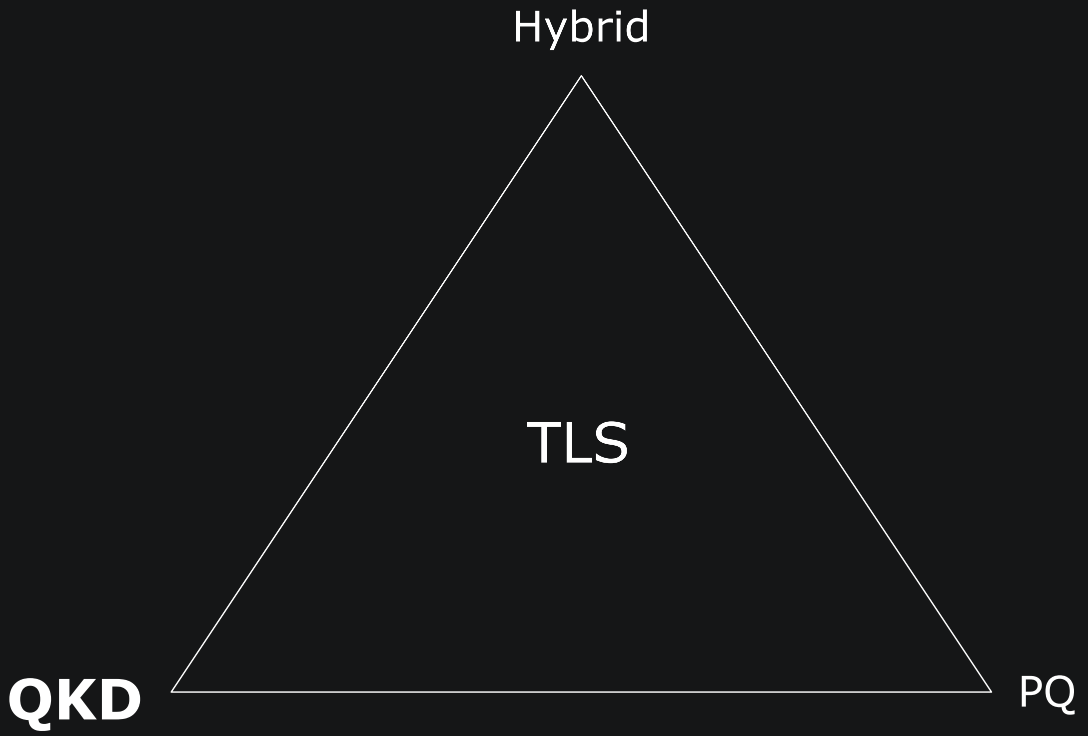
best suited for high-security environments like interdatacenter links or governmental networks.

Due to fiber loss and the no-cloning theorem, QKD is limited to a few hundred kilometers (Huttner et al., 2022). Multipath QKD protocols address this with trusted intermediaries (Liu et al., 2024; Prévost et al., 2025). ETSI GS QKD 014 v1.1.1 defines a standard interface for managing QKD keys (ETSI, 2019), which we previously verified with ProVerif under specific assumptions (Prévost et al., 2024).

We present a practical implementation of this standard by integrating QKD into TLS. Our "*TLS-QKD*" protocol replaces the handshake's public key exchange with a request to a local QKD manager, secured via HTTPS with bilateral authentication. The ETSI standard assumes local networks can safely use classical public key cryptosystems. Once a quantum key is received, symmetric encryption ensures message confidentiality.

TLS-QKD is fully backward compatible: it can in-

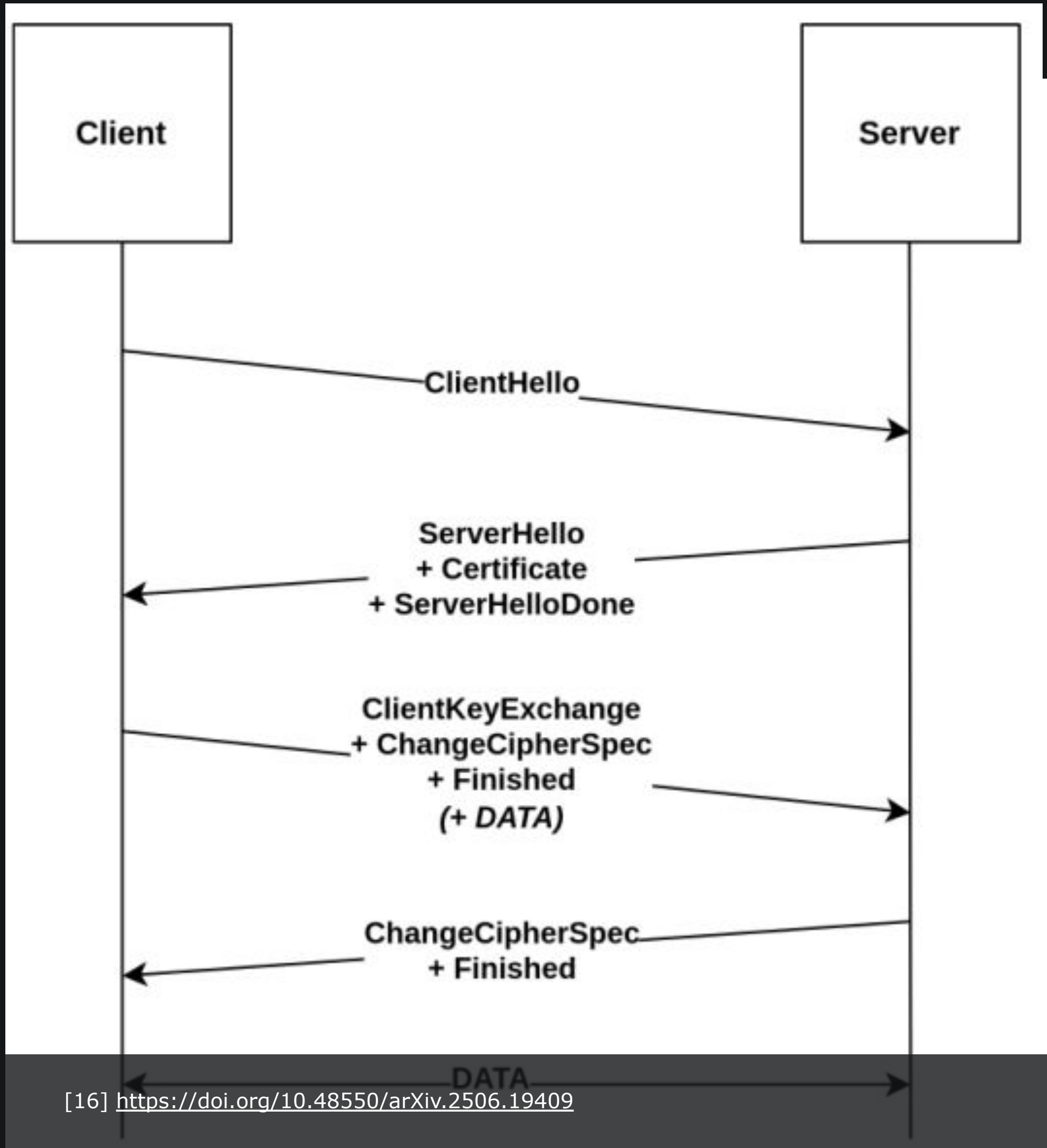
[16]



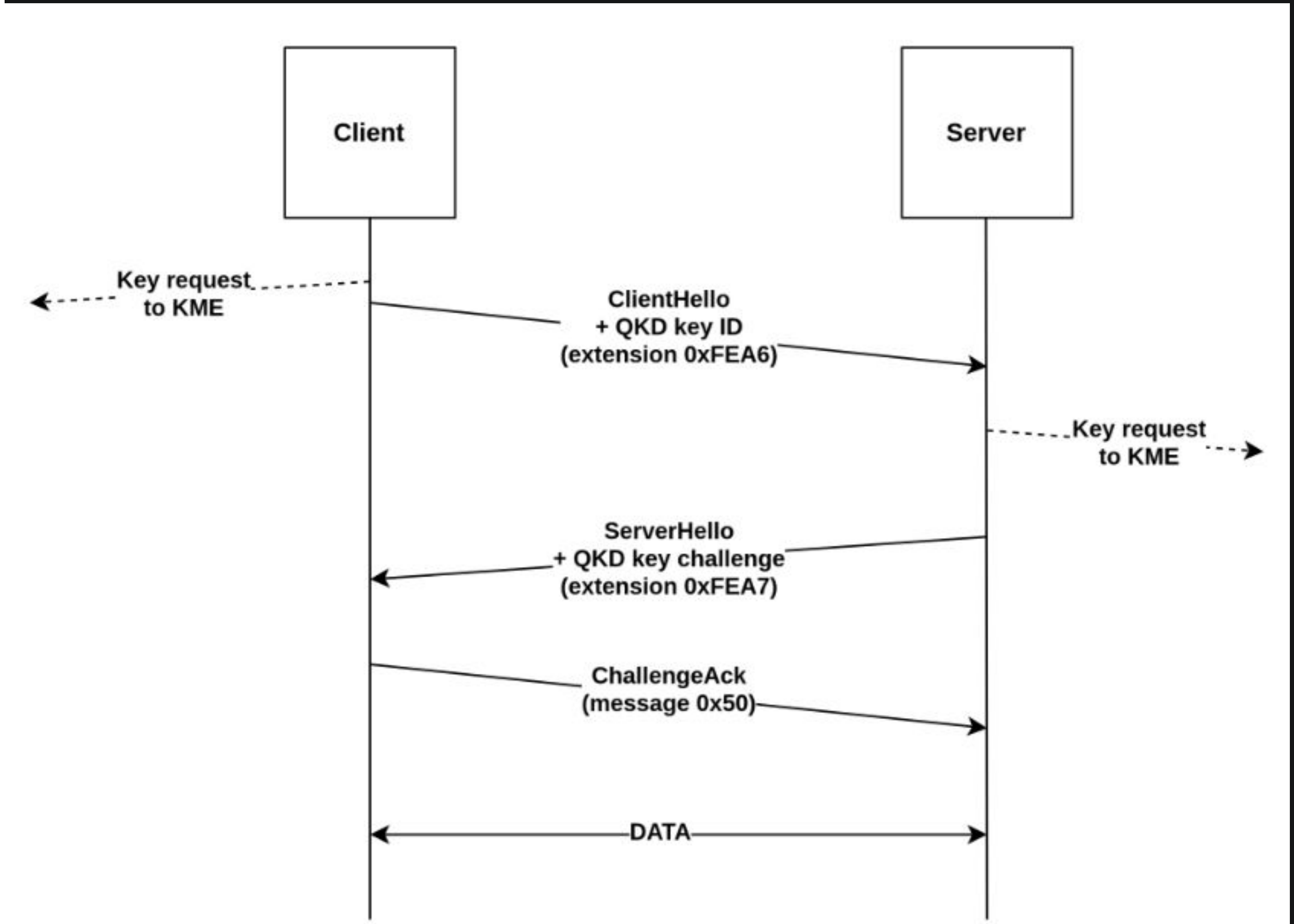
[16] https://doi.org/10.48550/arXiv.2506.19409

“Classical” TLS vs. TLS-QKD

[16]



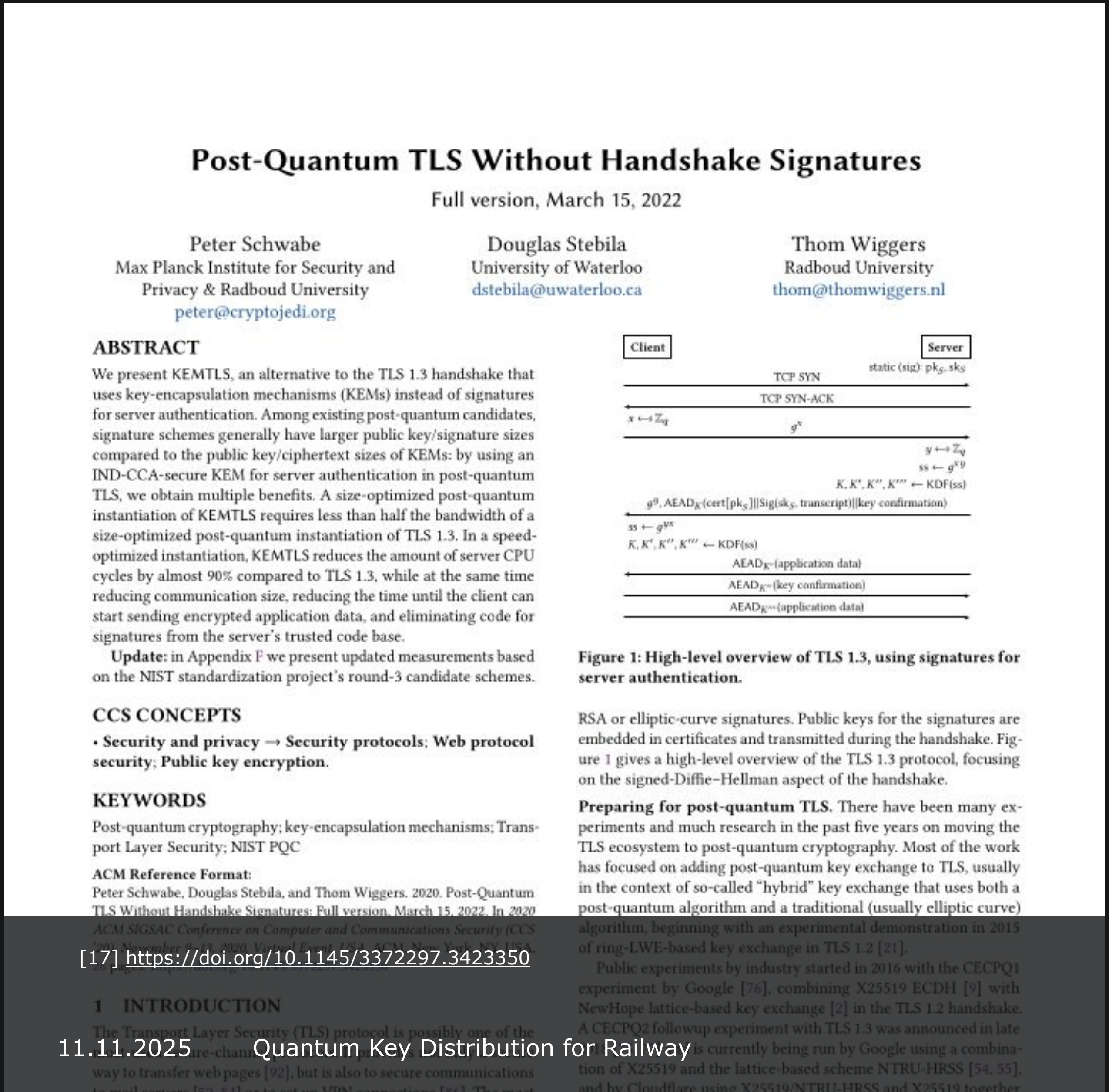
(b) Handshake on classic TLS 1.3.



(a) Handshake on TLS-QKD.

[16] https://doi.org/10.48550/arXiv.2506.19409

TLS Post-Quantum Approaches (3)



Post-Quantum TLS Without Handshake Signatures

Full version, March 15, 2022

Peter Schwabe
Max Planck Institute for Security and Privacy & Radboud University
peter@cryptojedi.org

Douglas Stebila
University of Waterloo
dstebila@uwaterloo.ca

Thom Wiggers
Radboud University
thom@thomwiggers.nl

ABSTRACT

We present KEMTLS, an alternative to the TLS 1.3 handshake that uses key-encapsulation mechanisms (KEMs) instead of signatures for server authentication. Among existing post-quantum candidates, signature schemes generally have larger public key/signature sizes compared to the public key/ciphertext sizes of KEMs: by using an IND-CCA-secure KEM for server authentication in post-quantum TLS, we obtain multiple benefits. A size-optimized post-quantum instantiation of KEMTLS requires less than half the bandwidth of a size-optimized post-quantum instantiation of TLS 1.3. In a speed-optimized instantiation, KEMTLS reduces the amount of server CPU cycles by almost 90% compared to TLS 1.3, while at the same time reducing communication size, reducing the time until the client can start sending encrypted application data, and eliminating code for signatures from the server's trusted code base.

Update: in Appendix F we present updated measurements based on the NIST standardization project's round-3 candidate schemes.

CCS CONCEPTS

• **Security and privacy → Security protocols; Web protocol security; Public key encryption.**

KEYWORDS

Post-quantum cryptography; key-encapsulation mechanisms; Transport Layer Security; NIST PQC

ACM Reference Format:
Peter Schwabe, Douglas Stebila, and Thom Wiggers. 2020. Post-Quantum TLS Without Handshake Signatures: Full version, March 15, 2022. In *2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20), November 9–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA,

1 INTRODUCTION

The Transport Layer Security (TLS) protocol is possibly one of the ... way to transfer web pages [92], but is also to secure communications

Client | Server
static (sig): pk_S, sk_S
TCP SYN
TCP SYN-ACK
$x \leftarrow_\$ \mathbb{Z}_q$
g^x
$y \leftarrow_\$ \mathbb{Z}_q$
$ss \leftarrow g^{xy}$
$K, K', K'', K''' \leftarrow \mathrm{KDF}(ss)$
g^y, AEAD$_K$(cert[pk_S]‖Sig(sk_S, transcript)‖key confirmation)
$ss \leftarrow g^{yx}$
$K, K', K'', K''' \leftarrow \mathrm{KDF}(ss)$
AEAD$_{K'}$(application data)
AEAD$_{K''}$(key confirmation)
AEAD$_{K'''}$(application data)

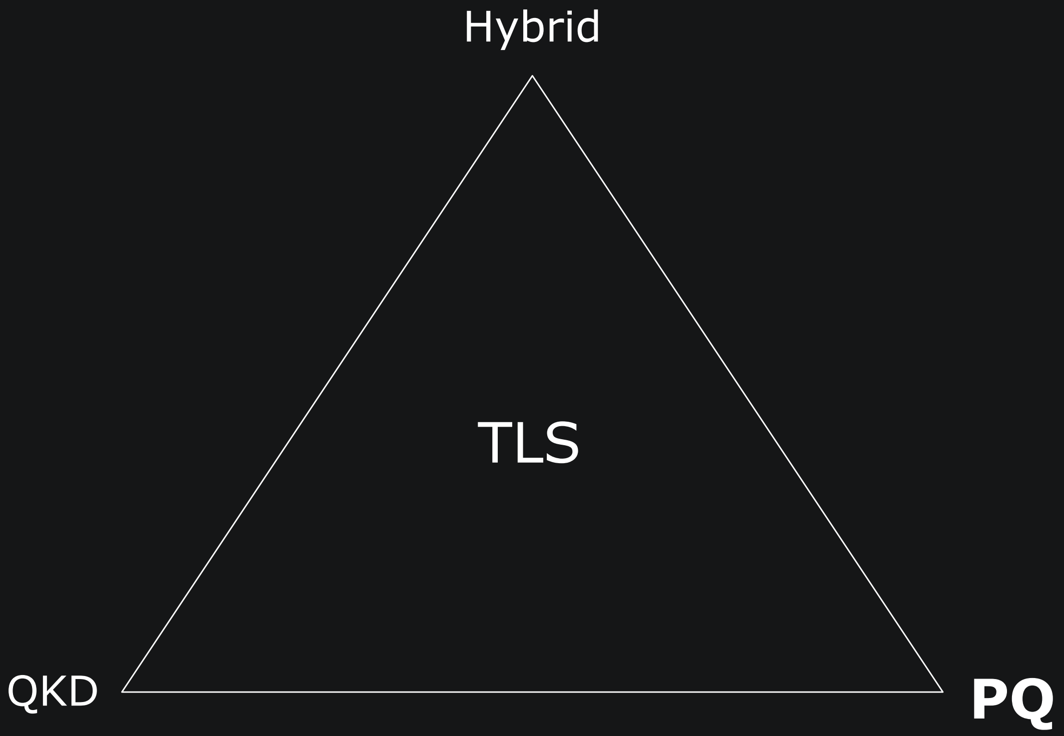
Figure 1: High-level overview of TLS 1.3, using signatures for server authentication.

RSA or elliptic-curve signatures. Public keys for the signatures are embedded in certificates and transmitted during the handshake. Figure 1 gives a high-level overview of the TLS 1.3 protocol, focusing on the signed-Diffie–Hellman aspect of the handshake.

Preparing for post-quantum TLS. There have been many experiments and much research in the past five years on moving the TLS ecosystem to post-quantum cryptography. Most of the work has focused on adding post-quantum key exchange to TLS, usually in the context of so-called "hybrid" key exchange that uses both a post-quantum algorithm and a traditional (usually elliptic curve) algorithm, beginning with an experimental demonstration in 2015 of ring-LWE-based key exchange in TLS 1.2 [21].

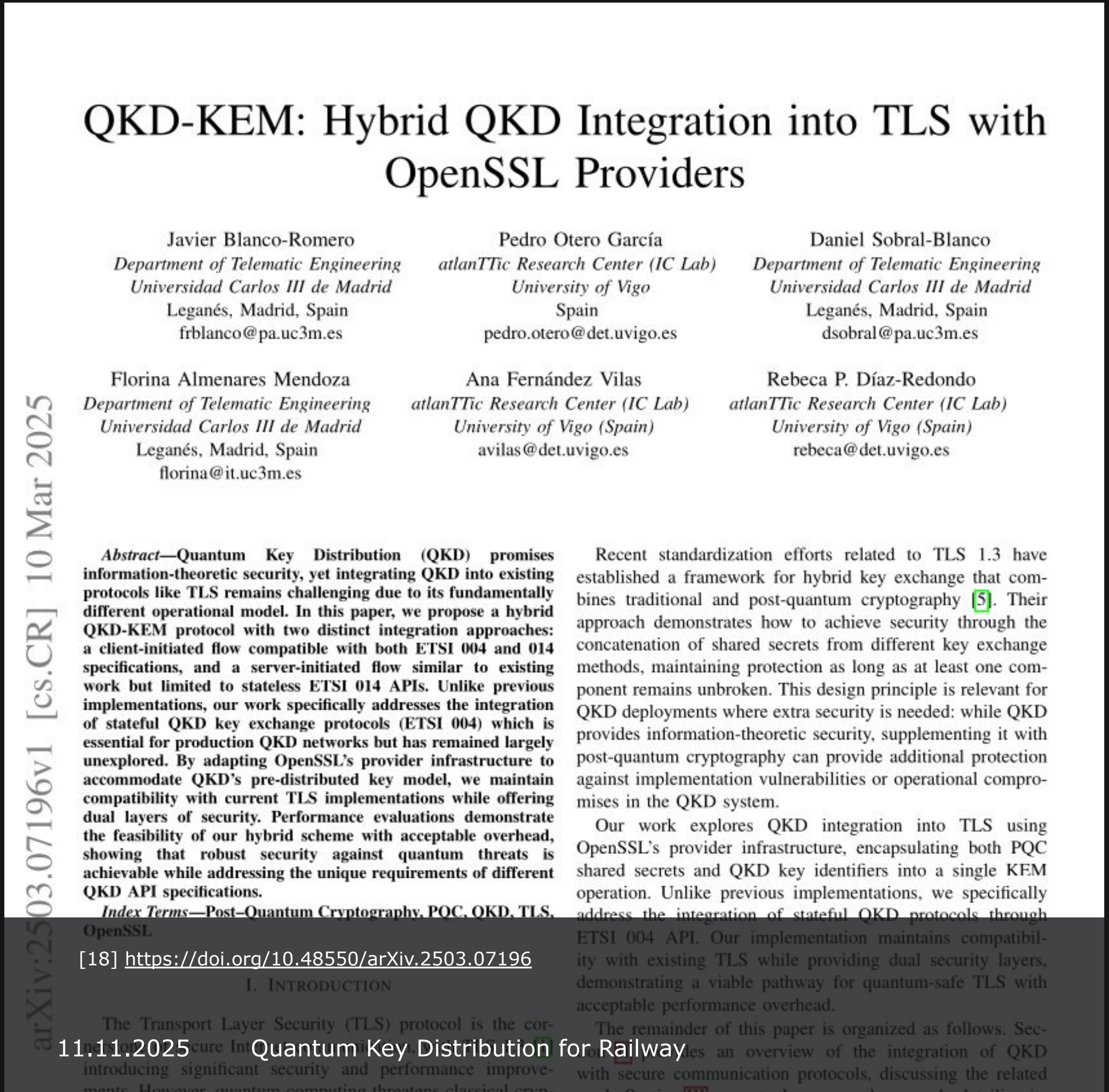
Public experiments by industry started in 2016 with the CECPQ1 experiment by Google [76], combining X25519 ECDH [9] with NewHope lattice-based key exchange [2] in the TLS 1.2 handshake. A CECPQ2 followup experiment with TLS 1.3 was announced in late ... is currently being run by Google using a combination of X25519 and the lattice-based scheme NTRU-HRSS [54, 55], and by Cloudflare using X25519/NTRU-HRSS and X25519 together

[17]



[17] https://doi.org/10.1145/3372297.3423350

TLS Post-Quantum Approaches (4)



arXiv:2503.07196v1 [cs.CR] 10 Mar 2025

QKD-KEM: Hybrid QKD Integration into TLS with OpenSSL Providers

Javier Blanco-Romero
Department of Telematic Engineering
Universidad Carlos III de Madrid
Leganés, Madrid, Spain
frblanco@pa.uc3m.es

Pedro Otero García
atlanTTic Research Center (IC Lab)
University of Vigo
Spain
pedro.otero@det.uvigo.es

Daniel Sobral-Blanco
Department of Telematic Engineering
Universidad Carlos III de Madrid
Leganés, Madrid, Spain
dsobral@pa.uc3m.es

Florina Almenares Mendoza
Department of Telematic Engineering
Universidad Carlos III de Madrid
Leganés, Madrid, Spain
florina@it.uc3m.es

Ana Fernández Vilas
atlanTTic Research Center (IC Lab)
University of Vigo (Spain)
avilas@det.uvigo.es

Rebeca P. Díaz-Redondo
atlanTTic Research Center (IC Lab)
University of Vigo (Spain)
rebeca@det.uvigo.es

***Abstract*—Quantum Key Distribution (QKD) promises information-theoretic security, yet integrating QKD into existing protocols like TLS remains challenging due to its fundamentally different operational model. In this paper, we propose a hybrid QKD-KEM protocol with two distinct integration approaches: a client-initiated flow compatible with both ETSI 004 and 014 specifications, and a server-initiated flow similar to existing work but limited to stateless ETSI 014 APIs. Unlike previous implementations, our work specifically addresses the integration of stateful QKD key exchange protocols (ETSI 004) which is essential for production QKD networks but has remained largely unexplored. By adapting OpenSSL's provider infrastructure to accommodate QKD's pre-distributed key model, we maintain compatibility with current TLS implementations while offering dual layers of security. Performance evaluations demonstrate the feasibility of our hybrid scheme with acceptable overhead, showing that robust security against quantum threats is achievable while addressing the unique requirements of different QKD API specifications.**

***Index Terms*—Post–Quantum Cryptography, PQC, QKD, TLS, OpenSSL**

I. INTRODUCTION

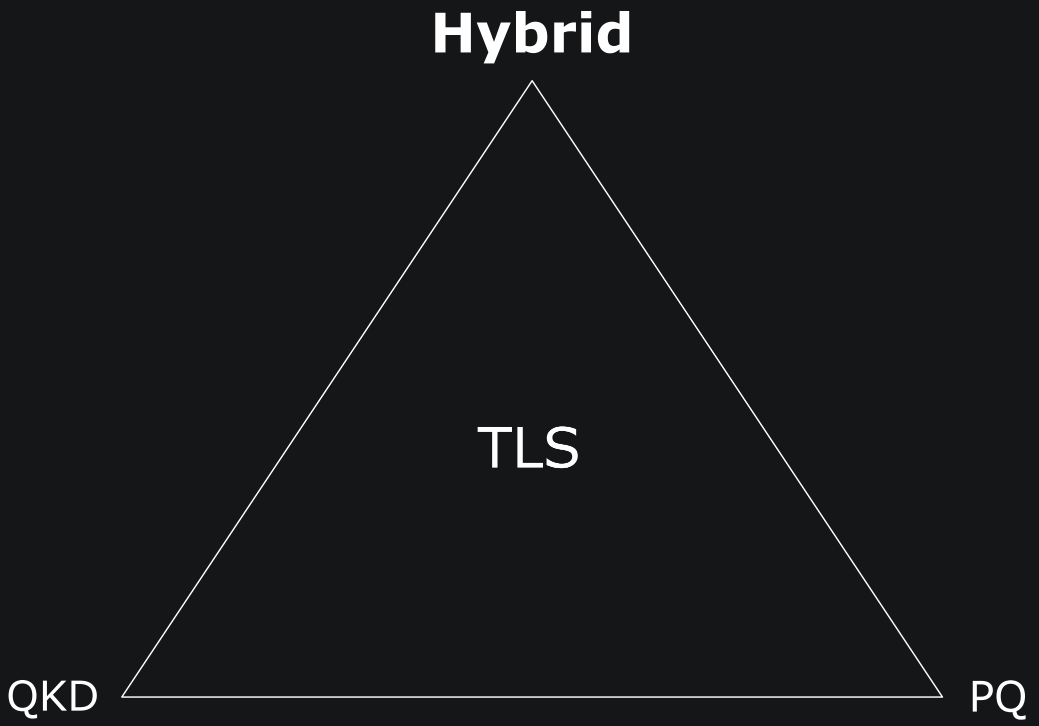
The Transport Layer Security (TLS) protocol is the cor-

Recent standardization efforts related to TLS 1.3 have established a framework for hybrid key exchange that combines traditional and post-quantum cryptography [5]. Their approach demonstrates how to achieve security through the concatenation of shared secrets from different key exchange methods, maintaining protection as long as at least one component remains unbroken. This design principle is relevant for QKD deployments where extra security is needed: while QKD provides information-theoretic security, supplementing it with post-quantum cryptography can provide additional protection against implementation vulnerabilities or operational compromises in the QKD system.

Our work explores QKD integration into TLS using OpenSSL's provider infrastructure, encapsulating both PQC shared secrets and QKD key identifiers into a single KEM operation. Unlike previous implementations, we specifically address the integration of stateful QKD protocols through ETSI 004 API. Our implementation maintains compatibility with existing TLS while providing dual security layers, demonstrating a viable pathway for quantum-safe TLS with acceptable performance overhead.

The remainder of this paper is organized as follows. Sec-

[18]



[18] https://doi.org/10.48550/arXiv.2503.07196

Authentication in the QKD context

- Two core needs
  - Entity authentication
  - Message/data authentication
- Options
  - PKI-based vs pre-shared symmetric approaches; MACs for data authentication.
- Bootstrapping and lifecycle key management are central design choices

ITU-T Y,380X standards

Table 2.1: Overview of *ITU-T Y.380X* QKD Standards

Standard	Content
Y.3800	Outlines concepts, capabilities, and design considerations for QKD and QKDN [14].
Y.3801	Defines functional requirements across the quantum, key management, control, and management layers [15].
Y.3802	Specifies the functional architecture, reference points, configurations, and base procedures [16].
Y.3803	Defines the key management architecture, interfaces, and security requirements for QKD networks [17].
Y.3804	Specifies functions and procedures for QKDN control and management via Fault, Configuration, Accounting, Performance, and Security (FCAPS) [18].

The ETSI GS QKD 014 v1.1.1

What are SAEs and KMEs?

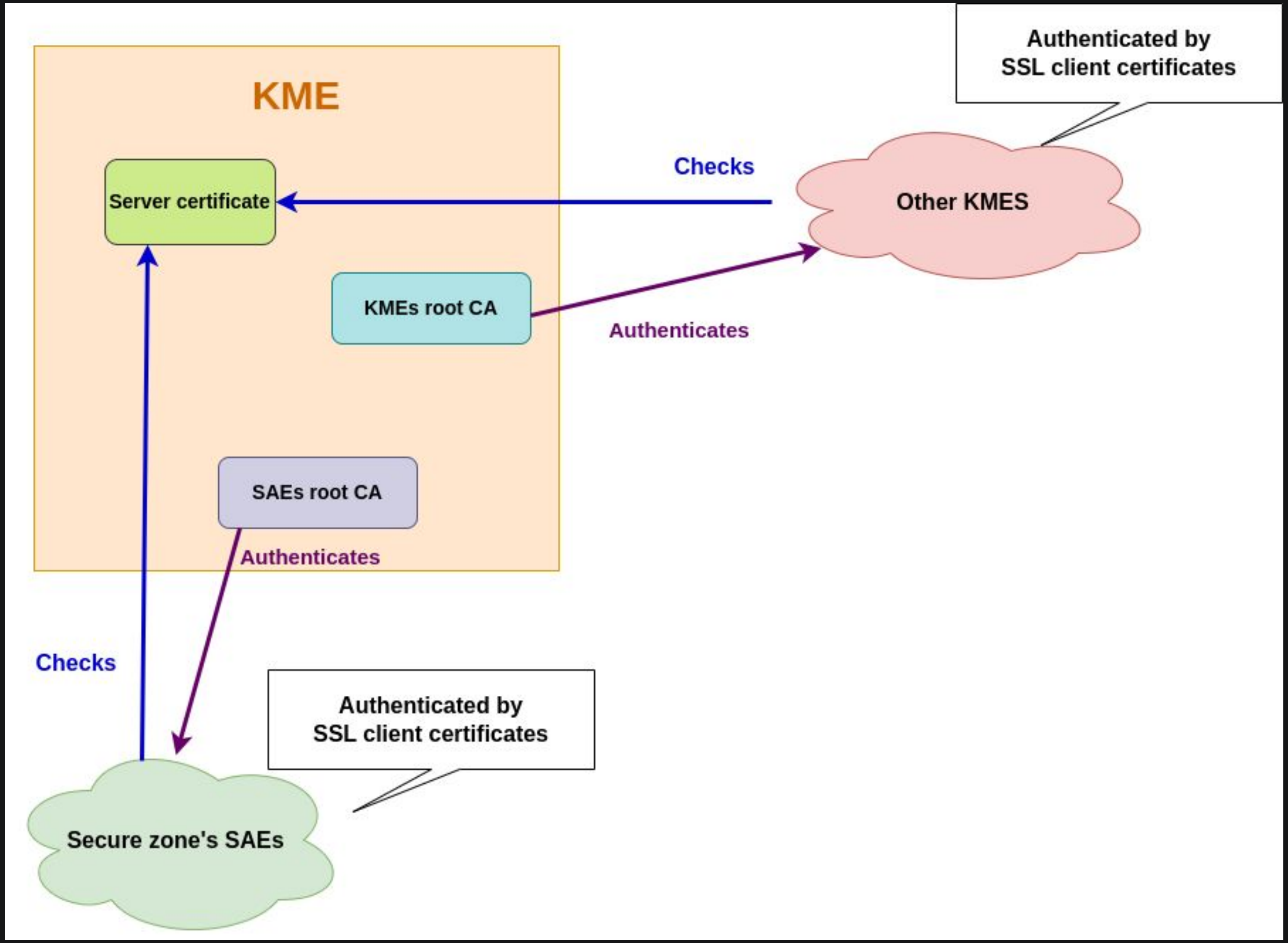
- SAE (Secure Application Entity): the application-side client (e.g., encryptors, optical switches, security management systems) that requests keys from its local KME over the ETSI QKD-014 REST/HTTPS API; it sits within the same security boundary/site as its KME, has a unique SAE ID, and can act as master/slave when identical keys must be delivered to multiple peers.

- KME (Key Management Entity): the key-management server in a trusted node that interfaces to QKD devices (QKDEs), cooperates with other KMEs across the QKD network, and delivers keys to SAEs via a web API; it authenticates SAEs, provides the REST/HTTPS key-delivery service, and has a unique KME ID.

Design Idea: Key Storage on Train that uses a “Key petrol station” with ETSI GS QKD 014 v1.1.1

Proof of Concept – Approach Two (3)

TLS-QKD Topology (2)



TLS variants comparison (3)

Scheme / Property	**TLS 1.3**	**TLS-QKD**	**KEMTLS**
Authentication	Signature-based certificate authentication (or Pre-Shared Key). Trust via PKI	possession of a pre shared QKD key proven via a challenge.	Replace signatures with KEMs: certificates contain long-term KEM keys
Hardware Requirements	No specific hardware requirements.	Special QKD system required.	Runs on standard CPUs; no special hardware required.
Resistance to quantum computers?	No, handshake relies on ECDHE for key establishment and ECDSA for authentication.	Yes. Keys have information-theoretic security due to quantum physical effects	Yes, relies only on symmetric primitives.

Design Idea: “Key Petrol Station” (“Schlüsseltankstelle”) (4)

ETSI GS QKD 014 v.1.1.1

<u>Phase A: Pre-trip preload</u>

0. Stationary QKD setup produces keys in secure location
1. Key Management Entity (KME, secure data center with QKD setup) and Secure Application Entity Master (SAE, e.g. the train) authenticate to ensure readiness `Get status`
2. SAE Master requests N slices of keys, KME returns key container `Get key`
    a. SAE loads stack of keys into its on-board Hardware Secure Module (HSM)
3. KME marks keys so that Slave SAE (e.g. Interlocking) can later call `Get key with keyIDs`

<u>Phase B — In journey: Use TLS-QKD</u>

<u>Phase C — Refill at stops</u>

1. Repeat Phase A when docked, replenish and purge expired or used keys.

ETSI GS QKD 014 v1.1.1 https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pd