



Post-Quantum Ready: Integrating OpenSSL, Bouncy Castle, and QKD via KMIP for Future-Proof Key Management

- Eric Ye
- eric.ye@cryptsoft.com
- Cryptsoft



What is QKD & How QKD work with existing System

Why QKD?

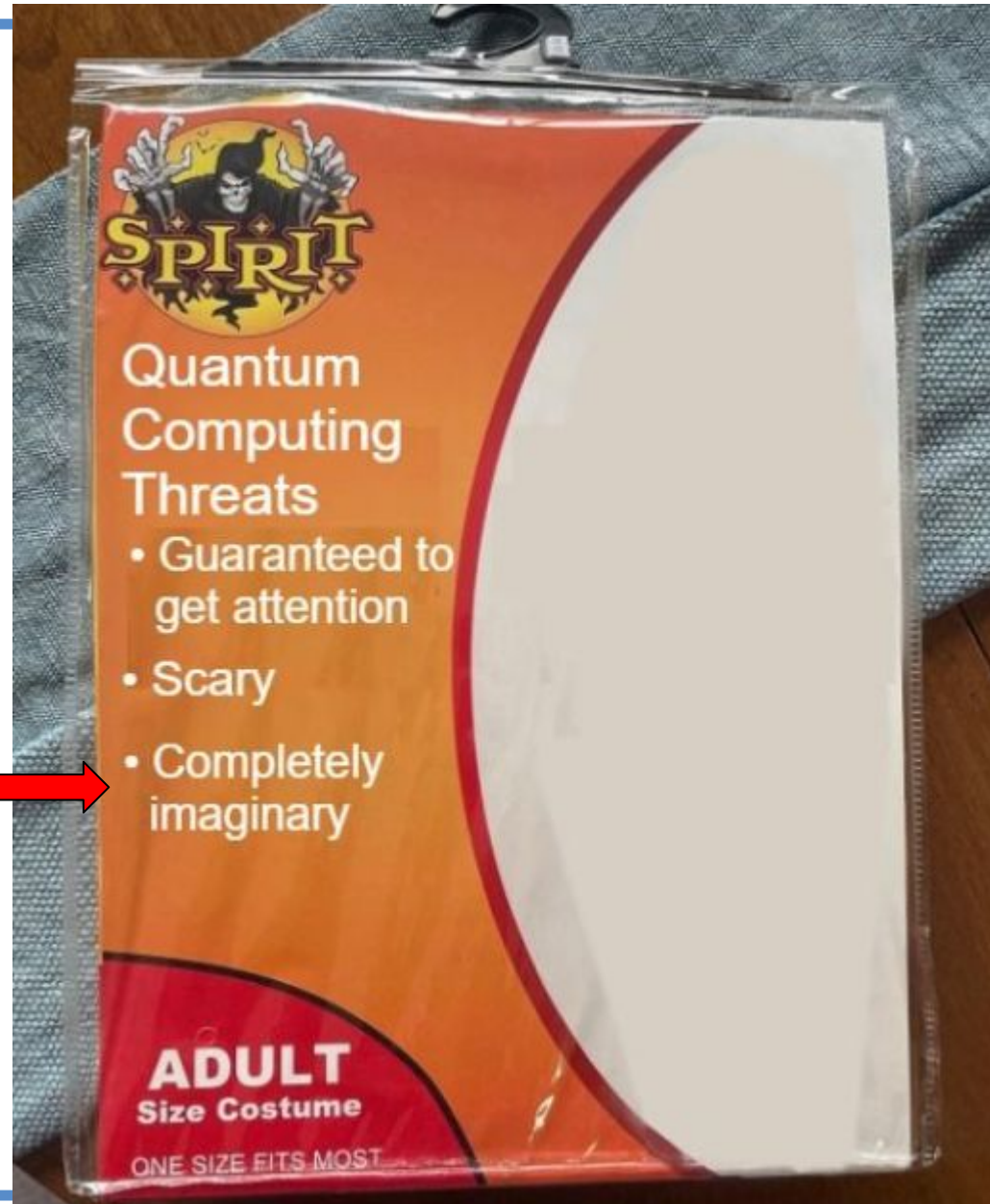
- By the end of 2027, PsiQuantum aims to launch the world's first utility-scale, fault-tolerant quantum computer.
- Shor's algorithm breaks RSA algorithm
- “harvest now, decrypt later” attack
- USD 1.8 billion in 2023; 5.27 billion by 2031 (verified market research 2024)
- Cisco, Nokia and Toshiba provide commercial QKD solution



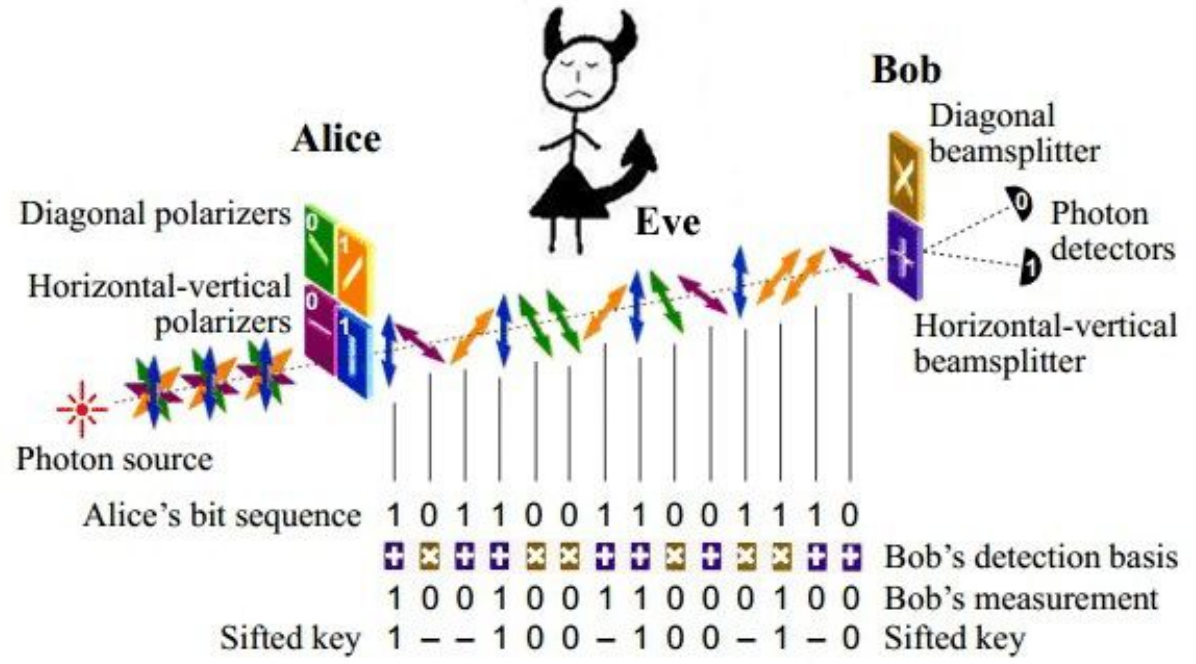
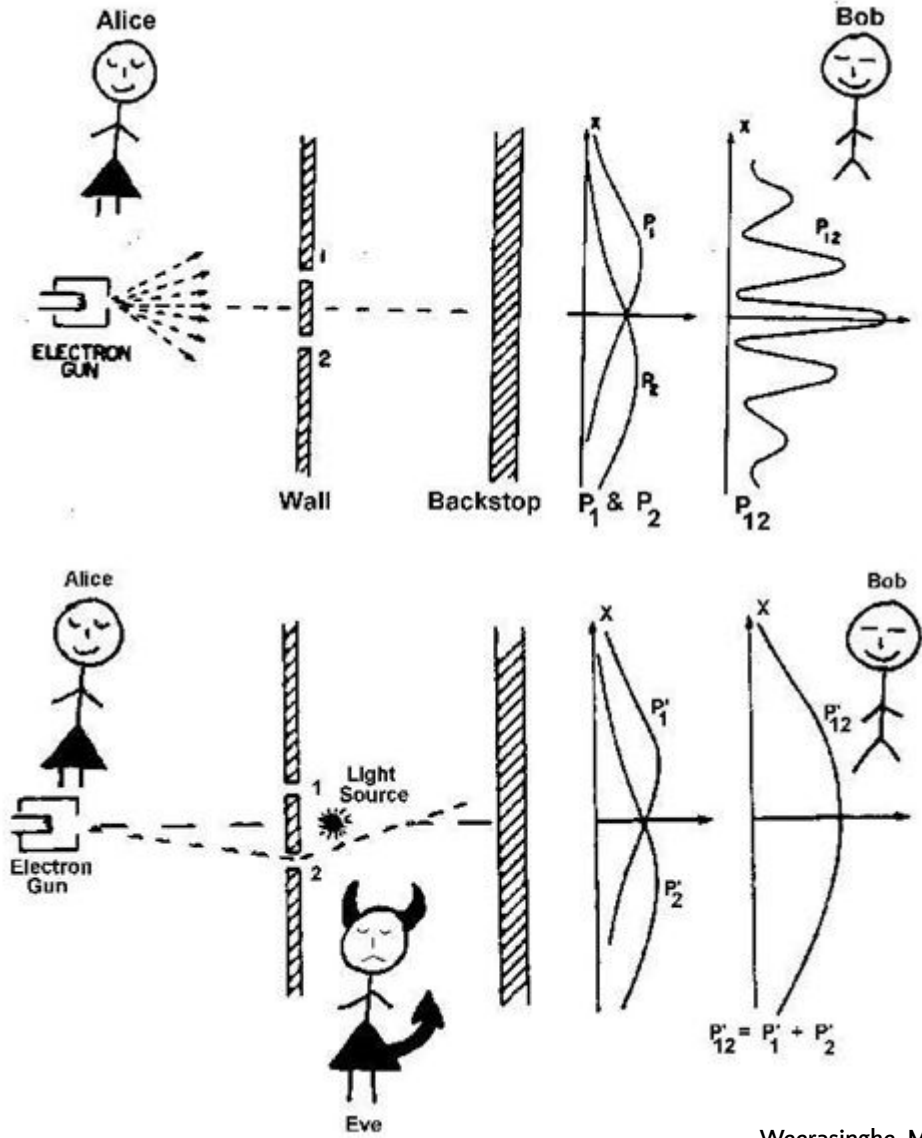
PsiQuantum's quantum computing site adjacent to Brisbane Airport.
Image courtesy of Lamar Johnson Collaborative.

Why QKD?

For now



What Is QKD ??



- Young's double slit experiment
- Change when Observed
- BB84 Protocol
- tamper evidence link

Weerasinghe, Maheshya. (2016). Quantum Cryptography. 10.13140/RC.2.1.4283.1603.

How To Do QKD ?? –Ground Based QKD



XG Series QKD machine – ID Quantique

■ QKD Infrastructure Cost

- small/point-to-point systems : ~ \$100,000
- Full-scale QKD systems: \$300,000 – >\$5 million
- fiber optics: \$5,000 – \$60,000 per mile

■ How it works:

- point to point Dark Fiber (~120KM)
- Trusted-node relays (intermediate stations every ~100KM)
- **two side get a continuous stream of symmetric key bits (“endless one-time pad”)**
- Loss in Optical fibre; Vulnerable to Interference and Noise



Satellite QKD – SpeQtral

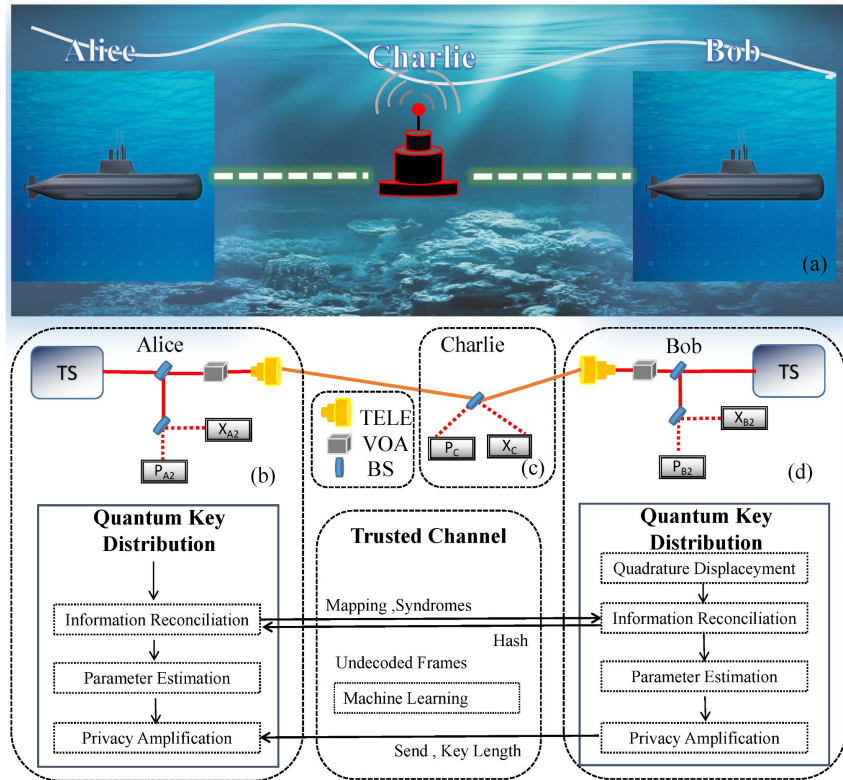
■ QKD Infrastructure Cost

- including the satellite and ground systems, ~130 million euros (– SES and European Space Agency project)

■ How it works:

- Satellite beams photons (laser) to ground stations
- up to 1,200 km (– QUESS 2016, Chinese Academy of Sciences)

How To Do QKD?? – Other Method in Research



Under Water QKD

Yi, J.; Wu, H.; Guo, Y. Passive Continuous Variable Measurement-Device-Independent Quantum Key Distribution Predictable with Machine Learning in Oceanic Turbulence. *Entropy* 2024, 26, 207. <https://doi.org/10.3390/e26030207>



Drone based QKD

Andrew Conrad¹, Samantha Isaac¹, Roderick Cochran², Daniel Sanchez-Rosales², Drone-based Quantum Key Distribution, *QCrypt 2021*, 1-Department of Physics, Illinois Quantum Information Science & Technology Center, University of Illinois Urbana-Champaign (UIUC); 2-Department of Physics, The Ohio State University (OSU)

How To Do QKD?? – Different Protocol / Technology

Discrete-variable QKD (DV-QKD)

Uses single photons and detectors (**BB84**, E91, etc.).

- **Advantage:** Well-studied, mature, widely tested in real-world networks.
- **Disadvantage:** Requires expensive single-photon detectors and limited distance (~100–200 km in fiber).

Continuous-variable QKD (CV-QKD)

Uses coherent light states and homodyne detection (more compatible with telecom).

- **Advantage:** Compatible with standard telecom components (lasers, coherent detectors), potentially lower cost.
- **Disadvantage:** More sensitive to channel loss and noise; requires precise phase and noise calibration.

Twin-field QKD (TF-QKD)

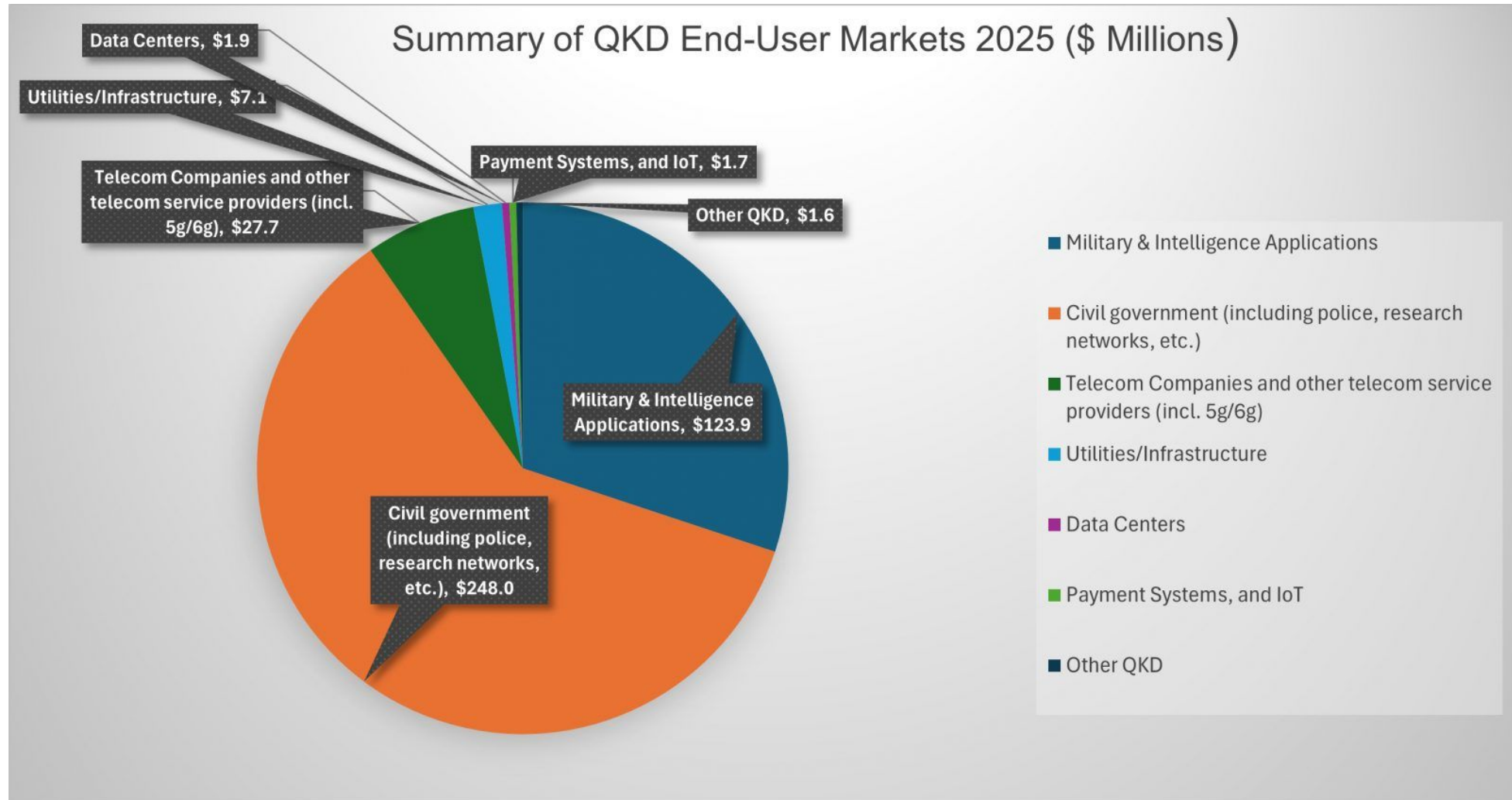
Extends distance limits of fiber QKD (~500–1000 km).

- **Advantage:** Extends distance limit to ~500–1000 km in fiber without quantum repeaters.
- **Disadvantage:** Technically complex — requires long-distance phase stabilization and synchronization.

Quantum repeater-based QKD (*future*)

Uses entanglement swapping and quantum memories to extend range.

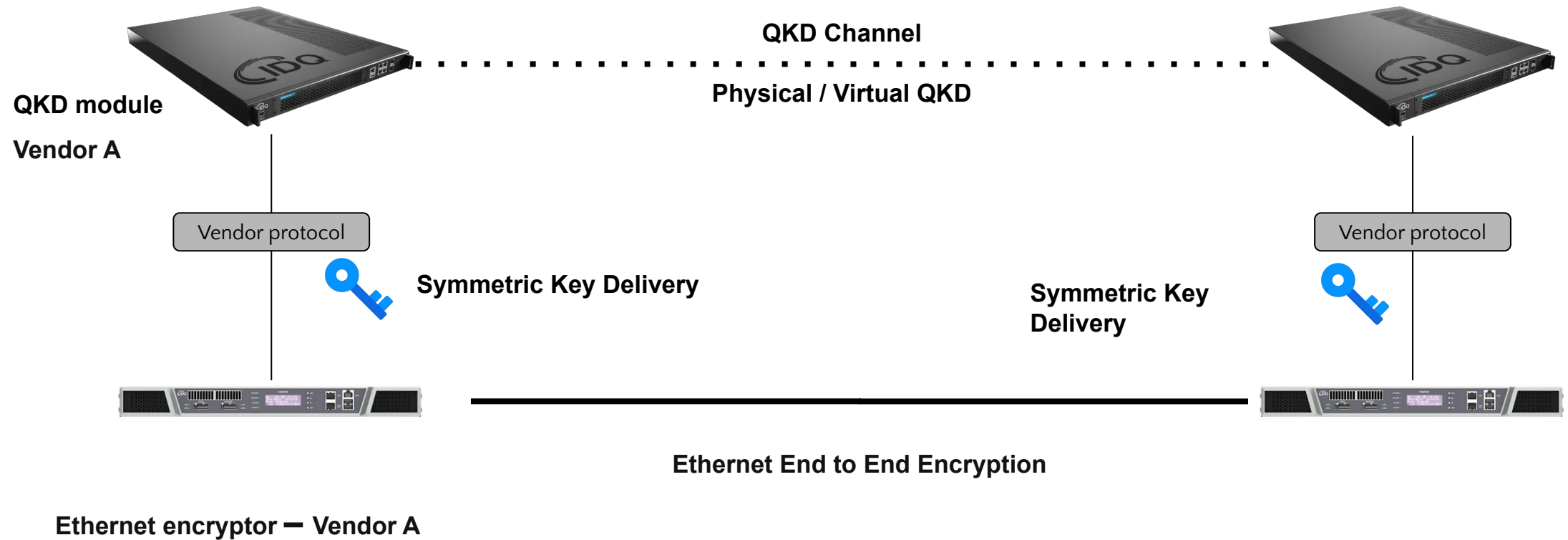
How To Do QKD?? – QKD Market



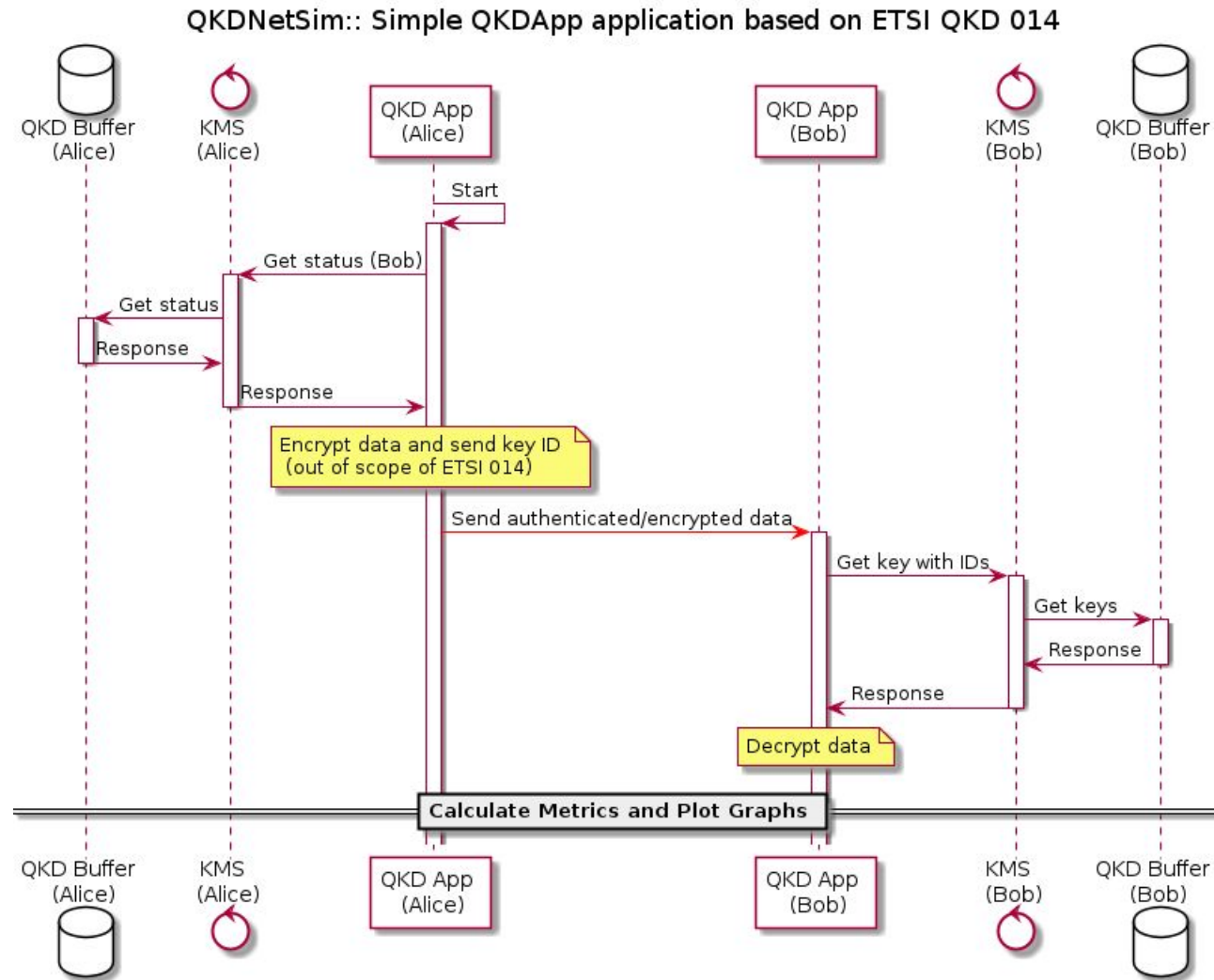
IQT RESEARCH posted 26 Jun 2024

www.insidequantumtechnology.com

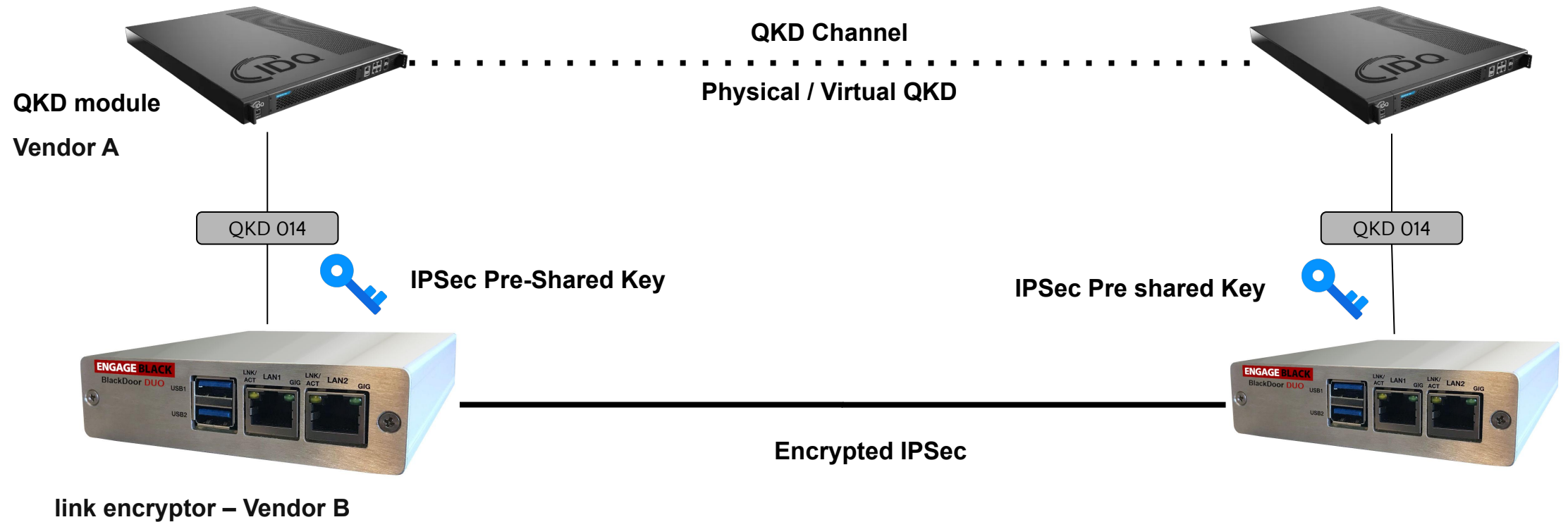
How To Do QKD ?? – QKD use case –



How To Do QKD? – ETSI QKD 014 protocol



How To Do QKD ?? – 3rd party link Encryptors



- **Software-based:** Uses mutual TLS with post-quantum algorithms (PQC).

Uses standard cryptographic protocols (e.g., TLS) with a “QKD-like” key delivery interface.

- No fiber optics. No QKD hardware.
- Equivalent protection **against quantum eavesdropping**.
- Simpler deployment using:
 - **OpenSSL 3.5+** (with hybrid KEM support).
 - **Bouncy Castle 1.81+** (Java stacks with PQC TLS).
- Lacks **tamper evidence** of physical links.

Existing Virtual QKD Network:

- QKDNetSim V2.0 -> open-qkd.eu
- qukaydee.com

Physical vs Virtual QKD

Feature	Physical QKD	Virtual QKD (TLS + PQC)
Tamper Evidence Link	✔ Yes	✘ No
Fiber optics /laser	✔ Yes (QKD devices)	✘ No
Quantum safe	✔ Yes	✔ Yes
Cost & Complexity	📦 High	💡 Low
vendor interop	✘ No	✘ No

QKD network -Three Layer architecture

➤ QKD Infrastructure (Quantum layer)

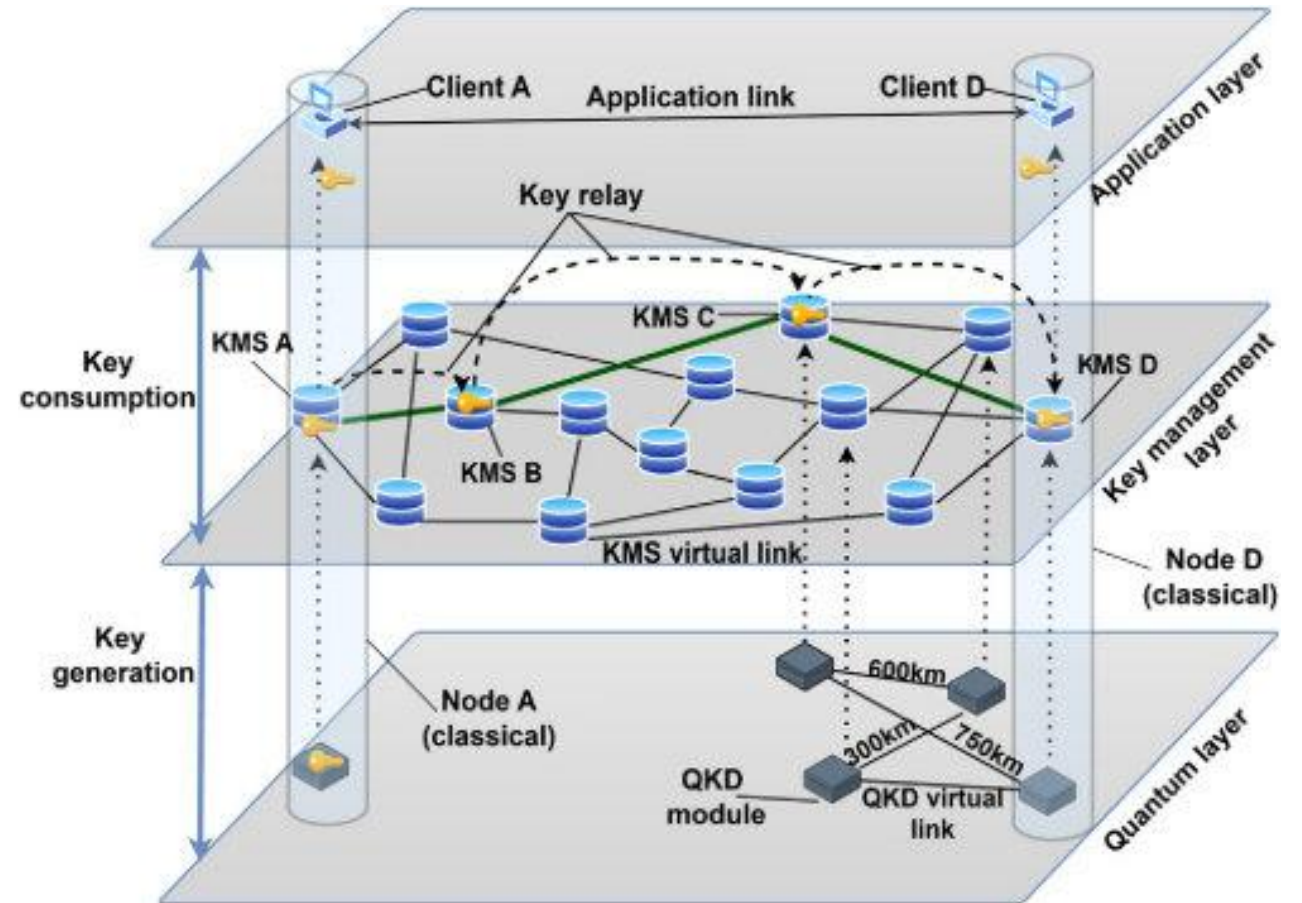
- limitation on distance
- in theory: radio, satellite
- in practise: point to point Dark Fibre

➤ KMS layer

- Not work with existing system
- No Open Standard
- key relay between

➤ Application layer (key use layer)

- IPsec ect..



Roa, Maria & Stan, Catalina & Verschoor, Sebastian & Tafur Monroy, Idelfonso & Rommel, Simon. (2025). Decentralized key distribution versus on-demand relaying for QKD networks. Journal of Optical Communications and Networking. 17. 732-742. 10.1364/JOCN.547793.

difference between Q-KMS & Classic KMS

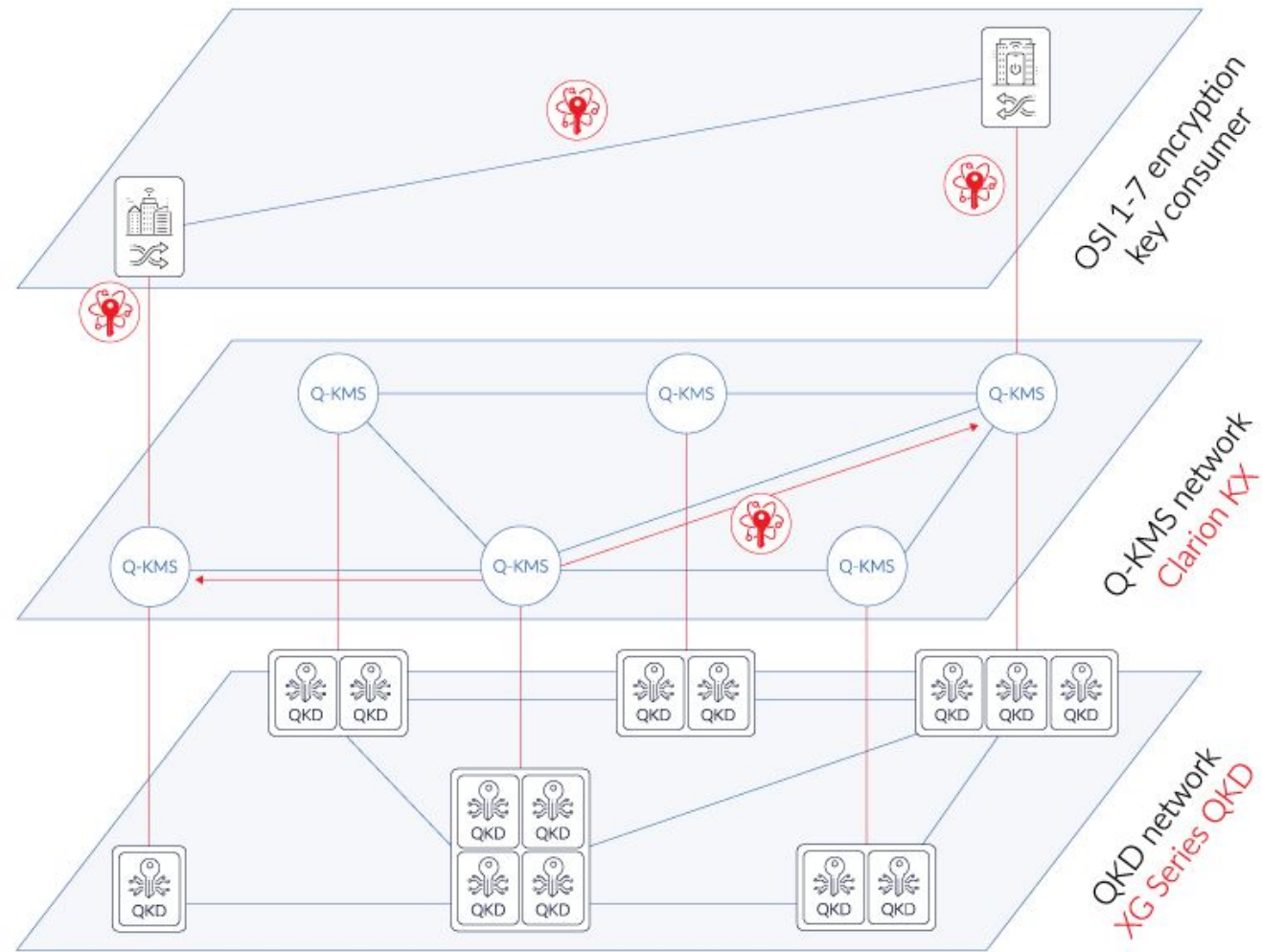
Classic KMS

- Centralized system
- Stores, distributes, and manages lifecycle of **asymmetric and symmetric keys**
- Built for conventional PKI and HSM environments
- **Open Standards Available for Interoperability: KMIP, PKCS#11**

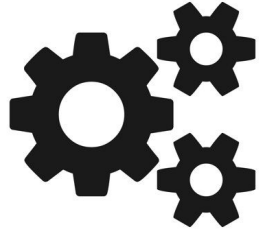
Quantum KMS (Q-KMS)

- Networking framework for **routing symmetric keys** generated by QKD
- Extends QKD from simple point-to-point links into **scalable multi-node networks**
- Provides **automatic key delivery, redundancy, and QoS management**
- Interconnects different QKD domains and integrates with telecom infrastructure (e.g., **SDN**)
- **Minimal Standards Available for Interoperability** (ETSI 014 is not a KM protocol)

QKD network- single vendor approach

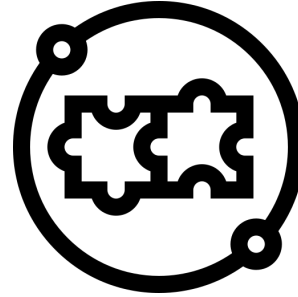


<https://www.idquantique.com/quantum-safe-security/key-exchange-service/>
id quantique 2025



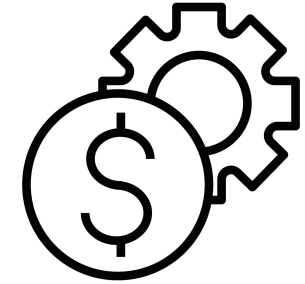
Transmission & Performance Limits

QKD systems are constrained by factors like distance, data rate, and **integration with existing networks & infrastructure.**



Standardization & Compatibility

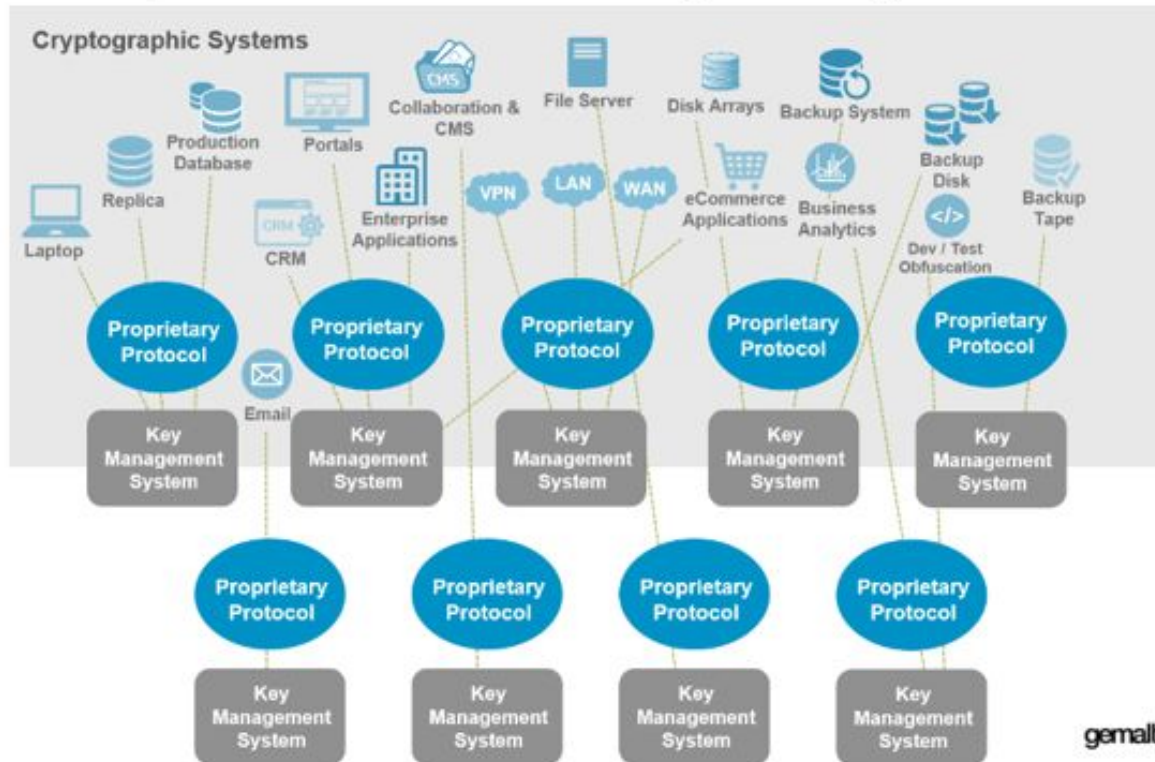
Lack of standardized protocols and interoperability between different QKD solutions slows broader adoption.



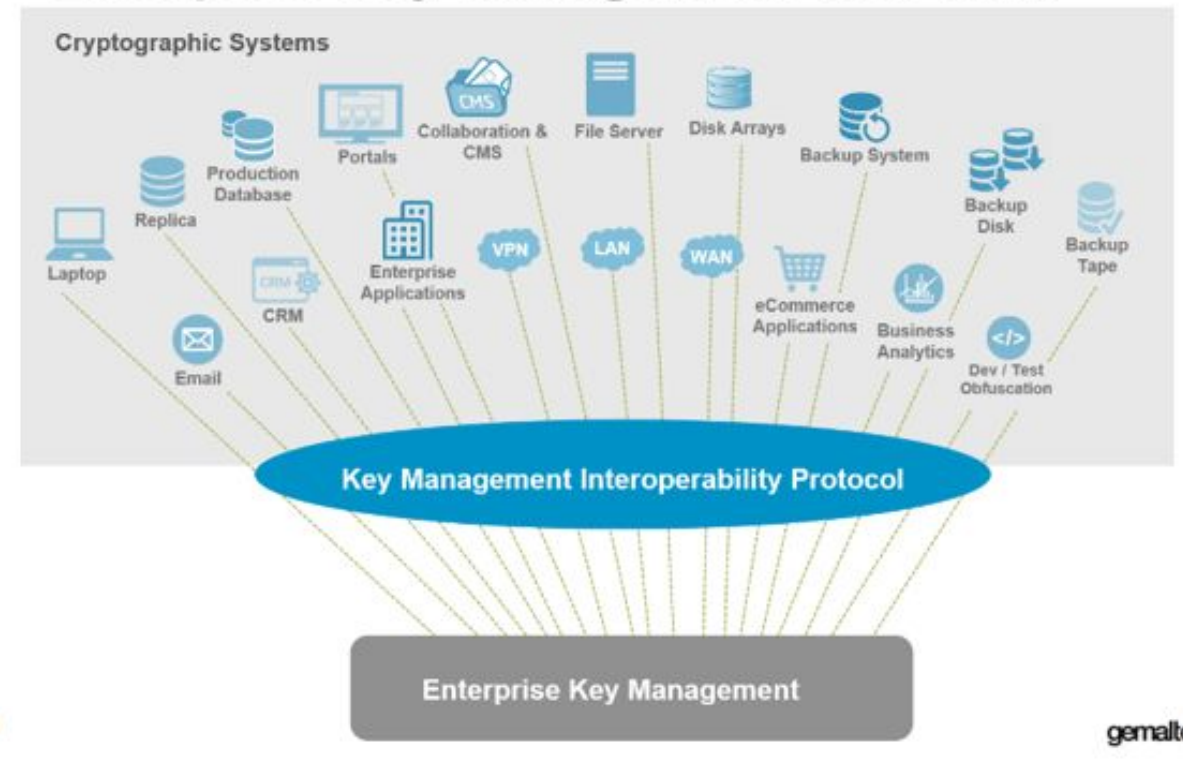
High Operational Costs:

Deploying QKD requires specialized hardware and infrastructure, making implementation expensive to maintain.

Market Challenge: Multiple Protocols for Key Management

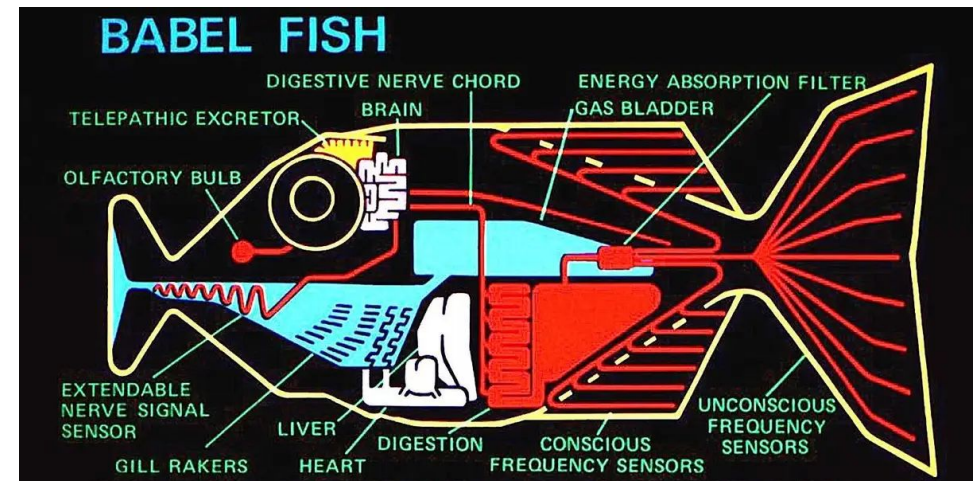
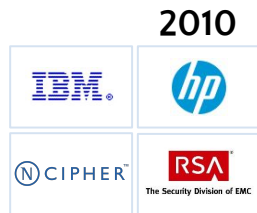


Market Solution: Enterprise Key Management with KMIP



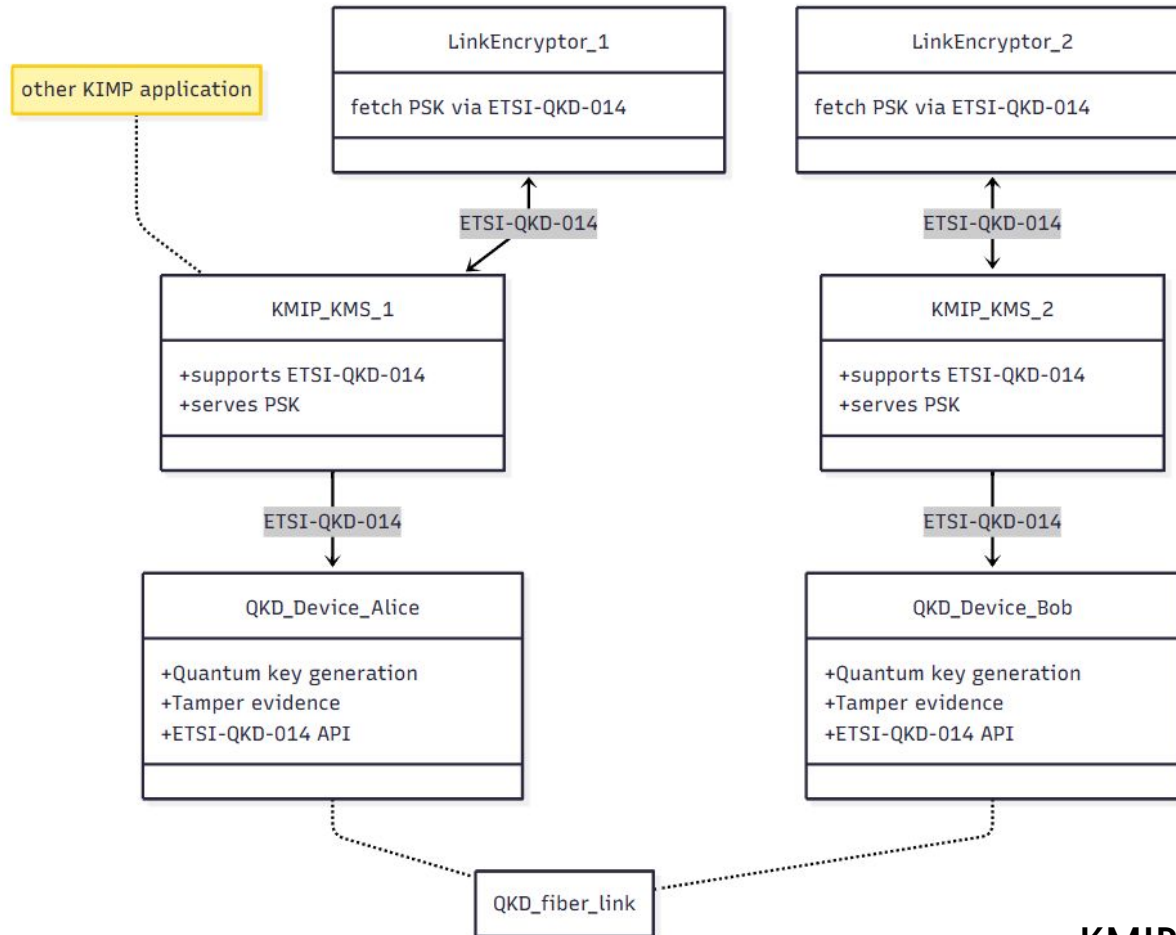
QKD with Standard adoption - KMIP

- From 2010 OASIS KMIP quickly become the most common standard for key management
- over 100+ KMIP product commercially available today.

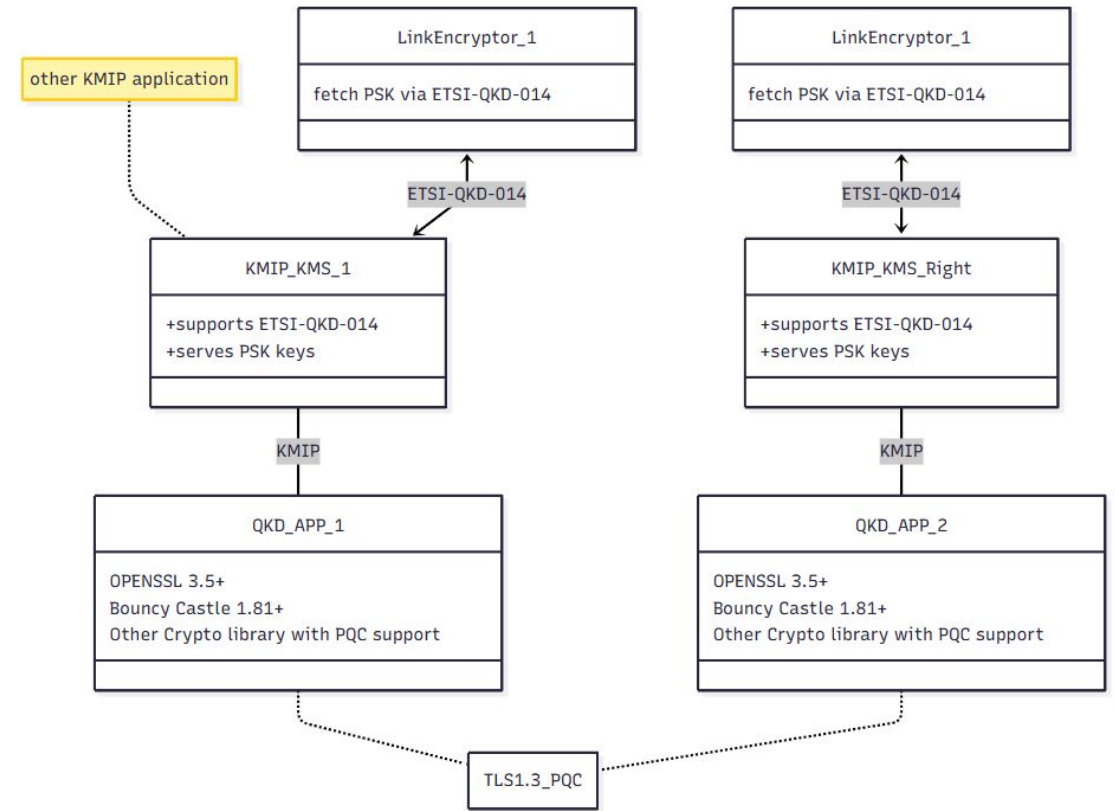


Ideal Architecture for Physical & Virtual QKD

Real QKD – Protection via Physics



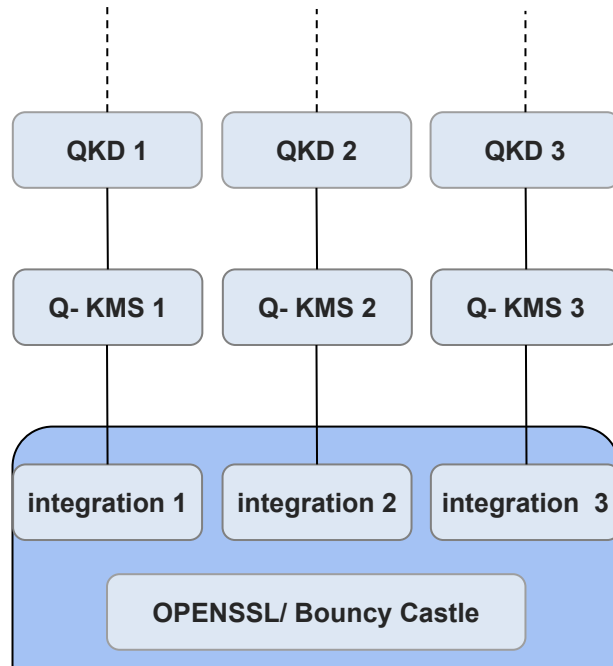
Virtual QKD with IPsec



- KMIP KMS with QKD 014 support
- easy integration with existing system

multi-vendor VS single-vendor approach

Single Node in QKD-Network

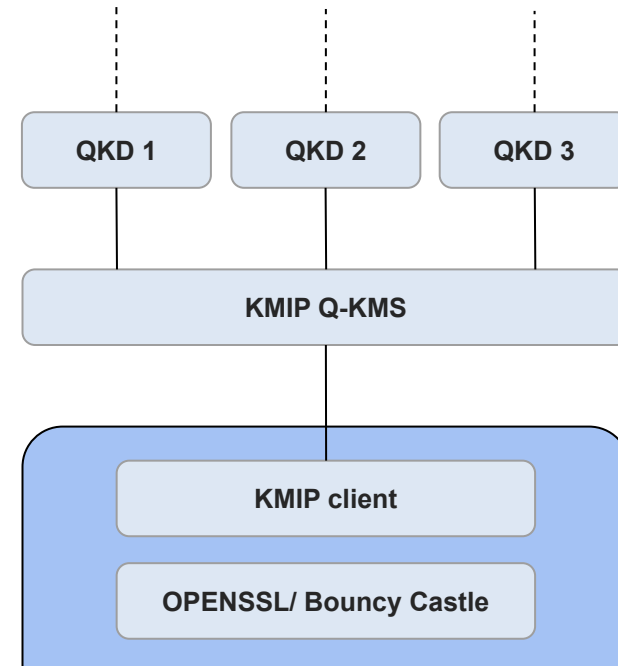


single-vendor approach

Quantum Layer

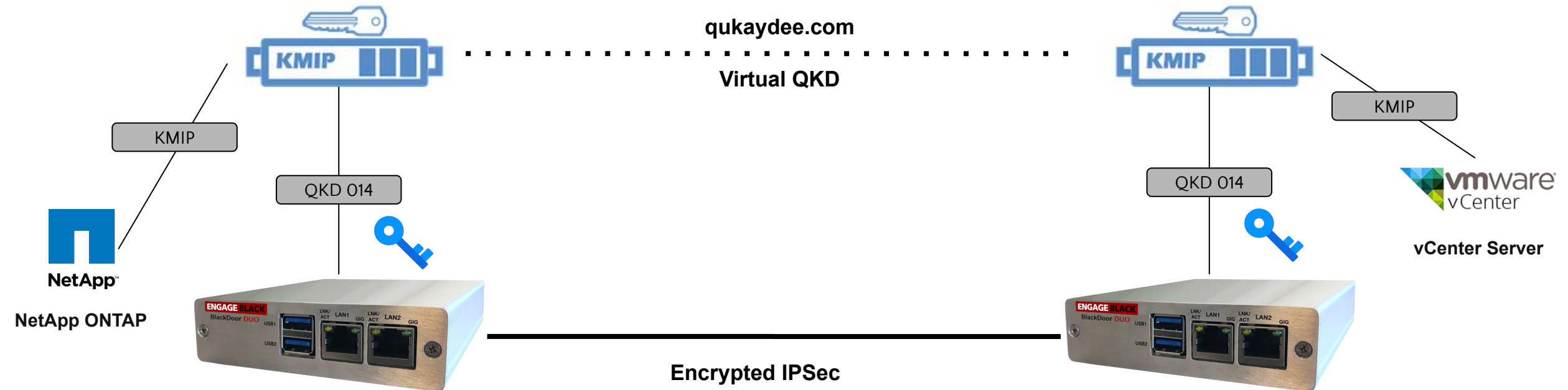
KMS Layer

Application Layer



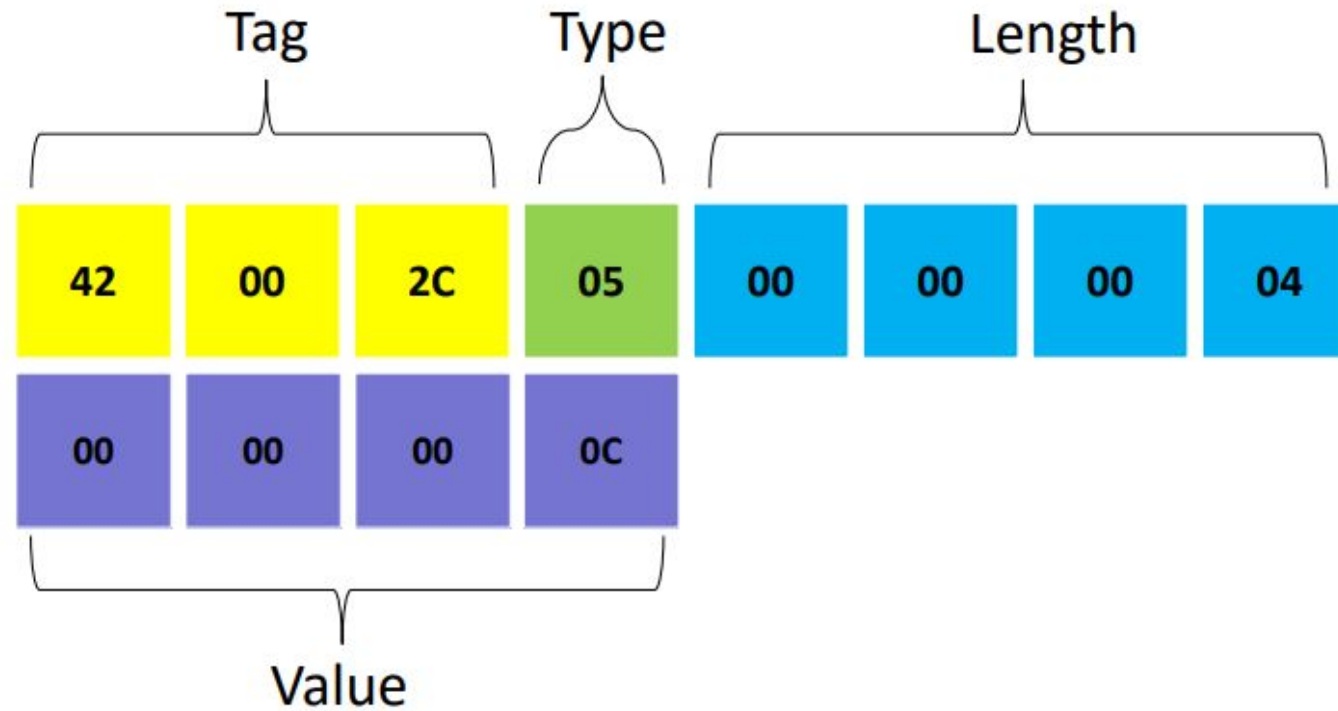
multi-vendor approach

Demonstrated multi-vendor interoperability with commercial products



KMIP Fundamentals

- Binary TTLV (Binary Tag-Type-Length-Value) Format
 - Optional JSON and XML encoding in KMIP



Cryptographic Usage Mask = Encrypt | Decrypt

KMIP Formats - HEX

```
42007801000001204200770100000038420069010000002042006a02000000040000000100000000
42006b020000000400000000000000042000d0200000004000000010000000042000f01000000d8
42005c0500000004000000010000000042007901000000c042005705000000040000000200000000
42009101000000a8420008010000003042000a070000001743727970746f6772617068696320416c
676f726974686d0042000b05000000040000000300000000420008010000003042000a0700000014
43727970746f67726170686963204c656e6774680000000042000b02000000040000000800000000
420008010000003042000a070000001843727970746f67726170686963205573616765204d61736b
42000b02000000040000000c00000000
```

KMIP Formats – TTLV

TAG

TYPE

LEN

VAL

PAD

OFFSET	DATA
00000000:	¹ 42 00 78 01 00 00 01 20 ² 42 00 77 01 00 00 00 38
00000010:	³ 42 00 69 01 00 00 00 20 ⁴ 42 00 6a 02 00 00 00 04
00000020:	00 00 00 01 00 00 00 00 ⁵ 42 00 6b 02 00 00 00 04
00000030:	00 00 00 00 00 00 00 00 ⁶ 42 00 0d 02 00 00 00 04
00000040:	00 00 00 01 00 00 00 00 ⁷ 42 00 0f 01 00 00 00 d8
00000050:	⁸ 42 00 5c 05 00 00 00 04 00 00 00 01 00 00 00 00
00000060:	⁹ 42 00 79 01 00 00 00 c0 ^A 42 00 57 05 00 00 00 04
00000070:	00 00 00 02 00 00 00 00 ^B 42 00 91 01 00 00 00 a8
00000080:	^C 42 00 08 01 00 00 00 30 ^D 42 00 0a 07 00 00 00 17
00000090:	43 72 79 70 74 6f 67 72 61 70 68 69 63 20 41 6c
000000a0:	67 6f 72 69 74 68 6d 00 ^E 42 00 0b 05 00 00 00 04
000000b0:	00 00 00 03 00 00 00 00 ^F 42 00 08 01 00 00 00 30
000000c0:	^G 42 00 0a 07 00 00 00 14 43 72 79 70 74 6f 67 72
000000d0:	61 70 68 69 63 20 4c 65 6e 67 74 68 00 00 00 00
000000e0:	^H 42 00 0b 02 00 00 00 04 00 00 00 80 00 00 00 00
000000f0:	^I 42 00 08 01 00 00 00 30 ^J 42 00 0a 07 00 00 00 18
00000100:	43 72 79 70 74 6f 67 72 61 70 68 69 63 20 55 73
00000110:	61 67 65 20 4d 61 73 6b ^K 42 00 0b 02 00 00 00 04
00000120:	00 00 00 0c 00 00 00 00

KMIP Formats – TTLV

TAG

TYPE

LEN

VAL

PAD

	INDEX	TAG	TYPE	LENGTH	VALUE	PAD
[1]	420078	01	00000120	REQUEST_MESSAGE	STRUCTURE	288
[2]	420077	01	00000038	REQUEST_HEADER	STRUCTURE	56
[3]	420069	01	00000020	PROTOCOL_VERSION	STRUCTURE	32
[4]	42006a	02	00000004	PROTOCOL_VERSION_MAJOR	INTEGER	4 0x00000001 00000000
[5]	42006b	02	00000004	PROTOCOL_VERSION_MINOR	INTEGER	4 0x00000000 00000000
[6]	42000d	02	00000004	BATCH_COUNT	INTEGER	4 0x00000001 00000000
[7]	42000f	01	000000d8	BATCH_ITEM	STRUCTURE	216
[8]	42005c	05	00000004	OPERATION	ENUMERATION	4 CREATE 00000000
[9]	420079	01	000000c0	REQUEST_PAYLOAD	STRUCTURE	192
[10]	420057	05	00000004	OBJECT_TYPE	ENUMERATION	4 SYMMETRIC_KEY 00000000
[11]	420091	01	000000a8	TEMPLATE_ATTRIBUTE	STRUCTURE	168

A very simple KMIP application with OPENSSL

- prepare the KMIP message in binary, sent over using `openssl s_client` function

```
eric-eng@eric-pc:~/work/kmipc$ cat request.bin | openssl s_client -connect 127.0.0.1:5696 -cert DEMO3.pem -key DEMO3
.pem -CAfile CA.pem -quiet > response.bin
Connecting to 127.0.0.1
Can't use SSL_get_servername
depth=1 C=AU, ST=Queensland, L=Brisbane, O=Cryptsoft Pty Ltd, CN=Cryptsoft KMIP Interop SHA2 CA
verify return:1
depth=0 C=AU, ST=Queensland, L=Brisbane, O=Cryptsoft Pty Ltd, OU=KMIP Interop Testing, CN=server, emailAddress=
verify return:1
```

```
eric-eng@eric-pc:~/work/kmipc$ xxd response.bin | head -50
00000000: 4200 7b01 0000 01e8 4200 7a01 0000 0070  B.{.....B.z....p
00000010: 4200 6901 0000 0020 4200 6a02 0000 0004  B.i.... B.j....
00000020: 0000 0003 0000 0000 4200 6b02 0000 0004  .....B.k....
00000030: 0000 0000 0000 0000 4200 9209 0000 0008  .....B.....
00000040: 0000 0000 68c7 8c7b 4201 0607 0000 002f  ....h..{B...../
00000050: 4538 4345 3530 3743 2d44 3732 442d 3445  E8CE507C-D72D-4E
00000060: 3434 2d38 3439 332d 4245 3442 3137 3139  44-8493-BE4B1719
00000070: 3930 3736 2d36 3843 3738 4337 422d 3600  9076-68C78C7B-6.
00000080: 4200 0f01 0000 0168 4200 5c05 0000 0004  B.....hB.\.....
00000090: 0000 001e 0000 0000 4200 7f05 0000 0004  .....B.....
000000a0: 0000 0000 0000 0000 4200 7c01 0000 0140  .....B.|....@
000000b0: 4200 6901 0000 0020 4200 6a02 0000 0004  B.i.... B.j....
```

A very simple KMIP application with Bouncy Castle

```
public static void main(String[] args) throws Exception {
    Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());

    String host = "127.0.0.1";
    int port = 5696;

    // TLS handshake
    Socket socket = new Socket(host, port);
    BcTlsCrypto crypto = new BcTlsCrypto();
    TlsClientProtocol protocol = new TlsClientProtocol(socket.getInputStream(), socket.getOutputStream());
    MyTlsClient client = new MyTlsClient(crypto, "keys/DEMO3.pem", "keys/DEMO3.pem", protocol);
    protocol.connect(client);

    // Send KMIP request
    byte[] request = java.nio.file.Files.readAllBytes(new File("request.bin").toPath());
    OutputStream out = protocol.getOutputStream();
    out.write(request);
    out.flush();

    // Read KMIP response
    InputStream in = protocol.getInputStream();
    FileOutputStream fos = new FileOutputStream("response.bin");
    byte[] buf = new byte[8192];
    int len;
    while ((len = in.read(buf)) > 0) {
        fos.write(buf, 0, len);
    }
    fos.close();
}
```

Future-Proofing Crypto Infrastructure

Past Milestones (2007–2019)

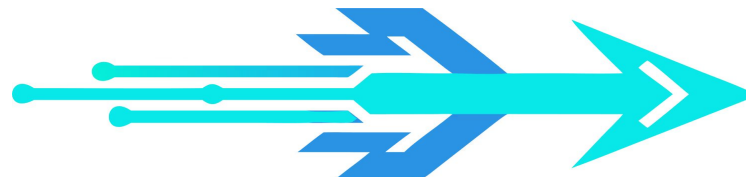
- **2007 – Proof of Feasibility:**
NIST & Los Alamos demonstrated the BB84 protocol over ~150 km optical fiber, proving QKD could move beyond lab settings.
- **2008 – Towards Higher Performance:**
Cambridge & Toshiba achieved a high-bit-rate QKD system, addressing scalability challenges in real-world telecom networks.
- **2017 – Quantum Entanglement at Scale:**
University of Science and Technology of China distributed entangled photons over 1,200 km, breaking distance records and proving QKD’s potential in satellite communications.
- **2018 – First Commercial Deployments:**
Quantum Xchange deployed a U.S. QKD network spanning 1,000 km of fiber from Boston to Washington, D.C., signaling the commercial readiness of QKD for national-scale secure communications.

Current Landscape (2020s)

- **Standardization in Progress:**
ETSI, and ISO are developing QKD and Q-KMS interoperability standards.
- **Integration with Existing Infrastructure:**
Telecom operators (BT, SK Telecom, China Mobile) have tried integrating QKD with classical IPsec and Ethernet encryptors.
- **QKD Technology Expansion:**
Experiments in Europe, China, and Canada are testing intercontinental quantum links via LEO and MEO satellites.

Future Outlook – 2030 & Beyond

- **Global Quantum Networks:**
Expansion from isolated testbeds to interconnected QKD networks, forming the backbone of a “quantum internet.”



Future-Proofing Crypto Infrastructure

Classical
Crypto

TLS1.3 with
PQC

QKD

Statistics from Cloudflare
<https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption>

Future-Proofing Crypto Infrastructure

17% are here
- 2024
Sep

0% are here
- 2024

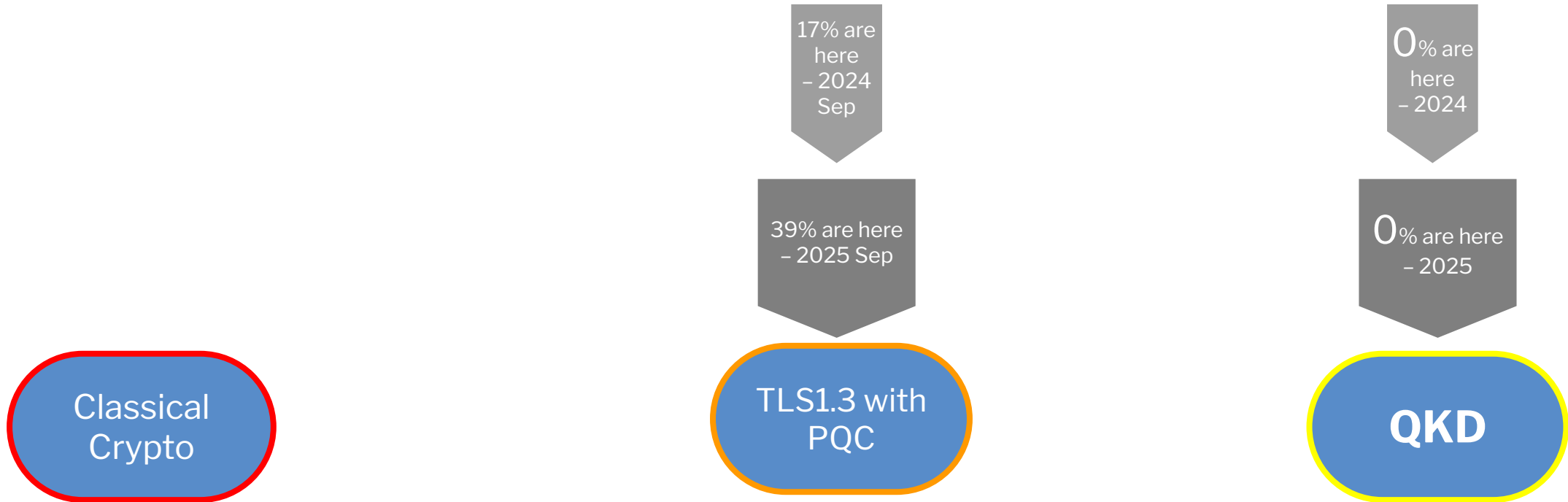
Classical
Crypto

TLS1.3 with
PQC

QKD

Statistics from Cloudflare
<https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption>

Future-Proofing Crypto Infrastructure



Statistics from Cloudflare
<https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption>

First time you read
about quantum mechanics



128th time you read
about quantum mechanics





CRYPTOSOFT®

FOUNDATION SECURITY TECHNOLOGIES
TRUSTED² | EMBEDDED | INTEROPERABLE

SALES@CRYPTOSOFT.COM

+61 7 3103 0321 | US +1 650 918 4362

WWW.CRYPTOSOFT.COM

 @CRYPTOSOFT

 CRYPTOSOFT-SECURITY-SPECIALISTS

 @CRYPTOSOFT