Doctoral Program in Computer and Control Engineering (35th cycle)

Cybersecurity and Quantum Computing: friends or foes?

Doctoral Examination Committee:
Prof. Sokratis Katsikas
David Arroyo, Ph.D.
Prof. Stefano Pirandola
Marco Gramegna, Ph.D.
Prof. Bartolomeo Montrucchio

Ph.D. candidate **Ignazio Pedone**
Supervisor **Prof. Antonio Lioy**

Outline

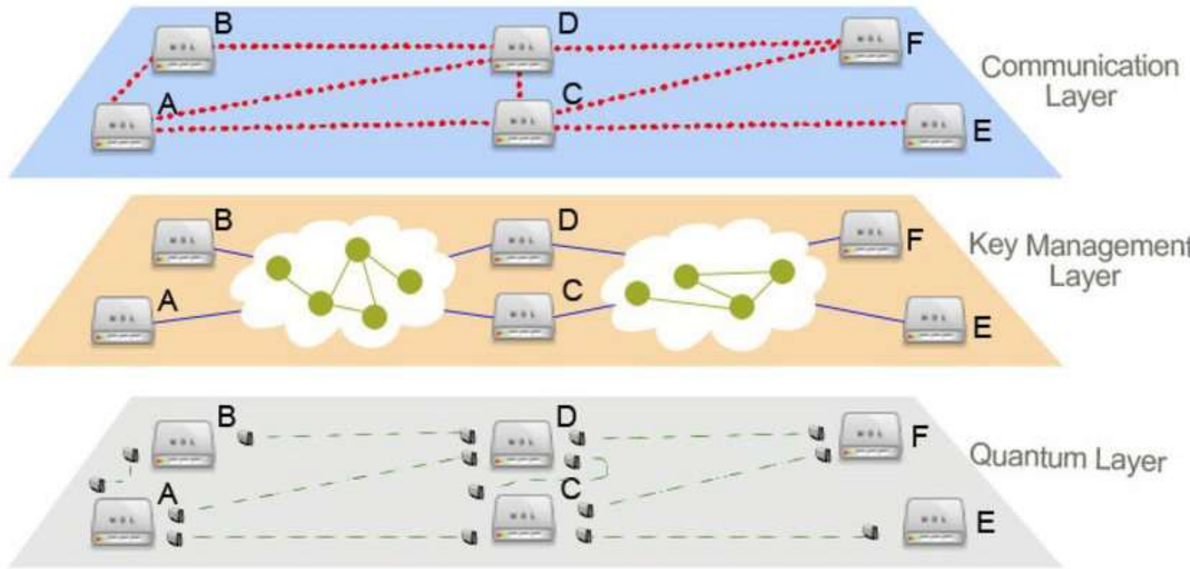
- Background and motivation
  - Quantum Technologies, Cybersecurity and Software-Defined Infrastructures (SDIs)
  - Research questions
- The Quantum Software Stack (QSS)
  - Quantum Key Distribution in SDIs
  - QKD simulator
- Virtual Network Function Embedding Problems (VNFEPs)
  - Quantum Annealing applied to optimisation problems
- “Quantum Offensive Security”
  - the impact of **Shor’s algorithm** on modern classical cryptography

Quantum Technologies

- Foundation
  - Quantum Mechanics, Quantum Information Theory
  - **qubit** as the basic unit of quantum information
- National Quantum Initiative Act (US, $1.2 billion USD)
- Quantum Computing
  - circuit model (e.g., IBM Q)
  - Quantum Annealing (e.g., D-Wave)
  - decoherence, faulty gates
  - Quantum Error Correction (QEC)
- Quantum Communication / Cryptography
  - Quantum Key Distribution (QKD)
  - Quantum Networks and the Quantum Internet
  - Quantum Cryptography (beyond QKD)

Quantum Key Distribution (QKD)

- Information-Theoretic Security (ITS)
- classes
  - Discrete Variable QKD (DV-QKD)
  - Continuous Variable QKD (CV-QKD)
- constraints
  - Point-to-Point (PTP) - exchange range
  - Multi-Hop (MH) - trusted relays
- QKD networks
- Standardisation (ETSI GS QKD)
- advances in the field
  - Device-Independent QKD (DI-QKD)
  - Measurement Device-Independent QKD (MDI-QKD)
  - Twin-Field QKD
  - quantum repeaters



(M. Mehic et al., Quantum Key Distribution: A Networking Perspective, ACM, 2020)

Cybersecurity

- global annual cost of cybercrime is growing fast (eSentire, 2023)
  - $8 Trilion USD in 2023
  - $10.5 Trilion USD by 2025

- *“Security is not a product, but a process.”* (cit. Bruce Schneier)

- cryptography as a pivotal building block

- security is more than a key exchange (protocols, standardisation)

Software-Defined Infrastructures

- cloud computing
  - fog and edge computing, Internet-of-Things (IoT), Network Functions Virtualisation (NFV)
  - “cloud-native” applications (software broken down into several distributed components)
- SDI “ingredients”
  - hardware resources, hypervisors, virtualisation technologies, operating systems capabilities
- main goal
  - resources and applications managed by software with minimum or no human intervention
- properties
  - high flexibility and scalability (on-demand)
  - network and software security are both critical

QTs vs Cybersecurity

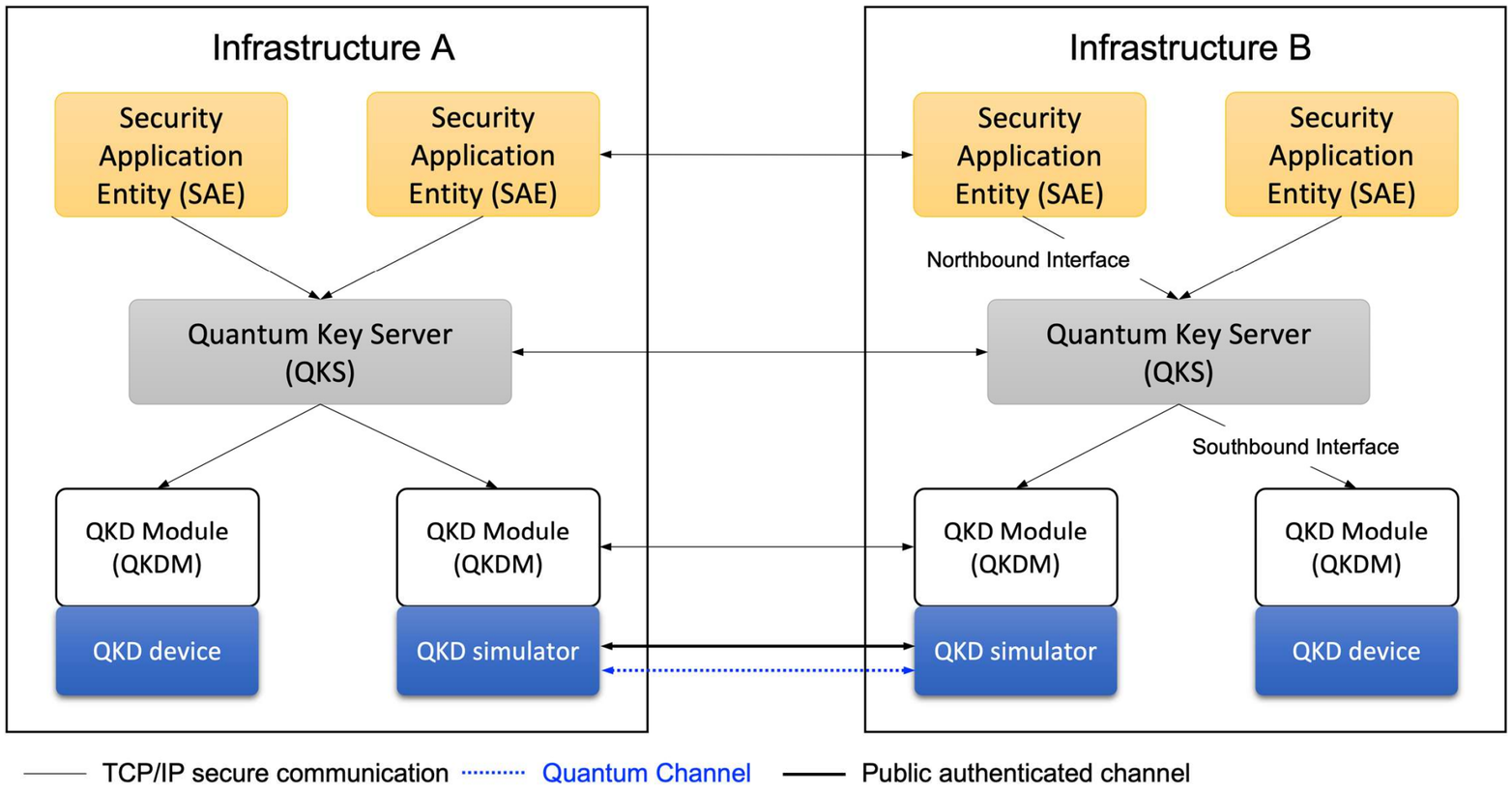
- Quantum Threat (Q-Day)
  - Shor's algorithm threatens current Public-Key Cryptography (PKC)
  - Grover's algorithm potentially threatens symmetric cryptography (e.g., AES) and hash functions (e.g., SHA-2)
- Quantum Cryptography
  - QKD, Quantum Digital Signature, Quantum Physical Unclonable Functions
- Quantum Annealing
  - applied to cybersecurity optimisation problems
- Quantum Machine learning (QML)
  - Intrusion Detection Systems to prevent Denial of Service attacks

Research Questions

- How to ease the adoption of QKD for SDIs?
  - high-level of flexibility of SDIs should meet the constraints of QKD
  - several key managers and crypto engines are available
  - QKD devices and infrastructures are expensive and require simulation
- What are short-to-medium-term applications of Quantum Computing to Cybersecurity?
  - current limitations of quantum hardware must be considered
- What is the impact of the Quantum Threat on current cryptography?
  - resource estimation for Shor's algorithm and variants
  - clear metrics to understand the maturity of the available implementations

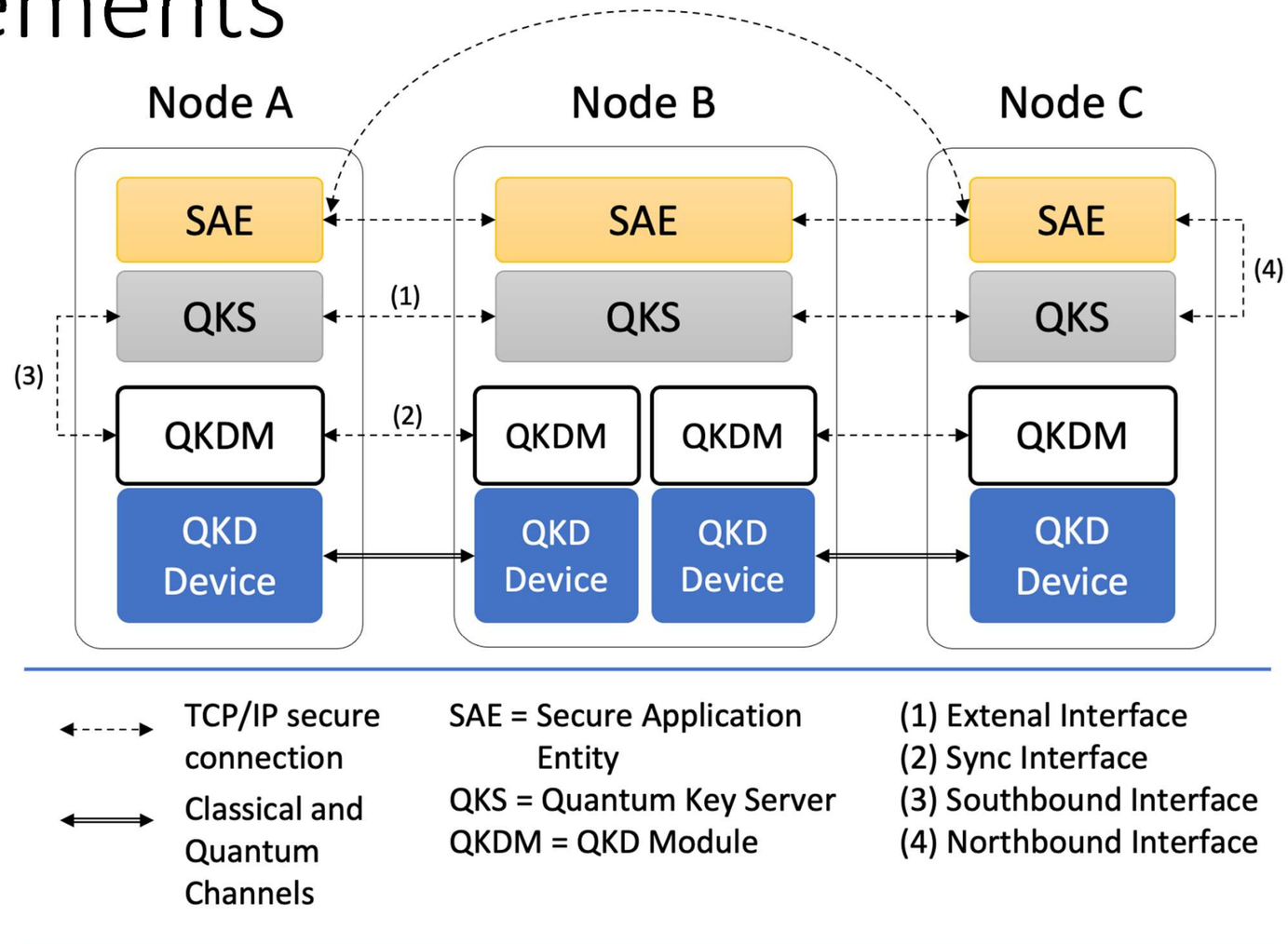
Quantum Software Stack (QSS)

- ETSI GS QKD 014 v1.1.1 compliant software stack (four layers – two primary interfaces)
- SAE is the security function which queries the QKS for fresh key material
- QKS manages the quantum key exchange process across the infrastructure sites, provides routing capability for the SAEs (Trusted Repeaters are necessary), and manages the registered QKDMs.
- QKDM is an abstraction of the QKD device and provides a common interface to quantum devices
- QKD Device/Simulator represents either the quantum device or the QKD simulator



QSS 2.0: improvements

- Asynchronous I/O
  - asyncio
- Multi-processing
  - Quart
- simplified QKDM
- routing capabilities
- trusted repeaters (MH exchange)
- Kubernetes operator
  - deploy
  - cluster-to-cluster exchange
  - target cloud-native infrastructures

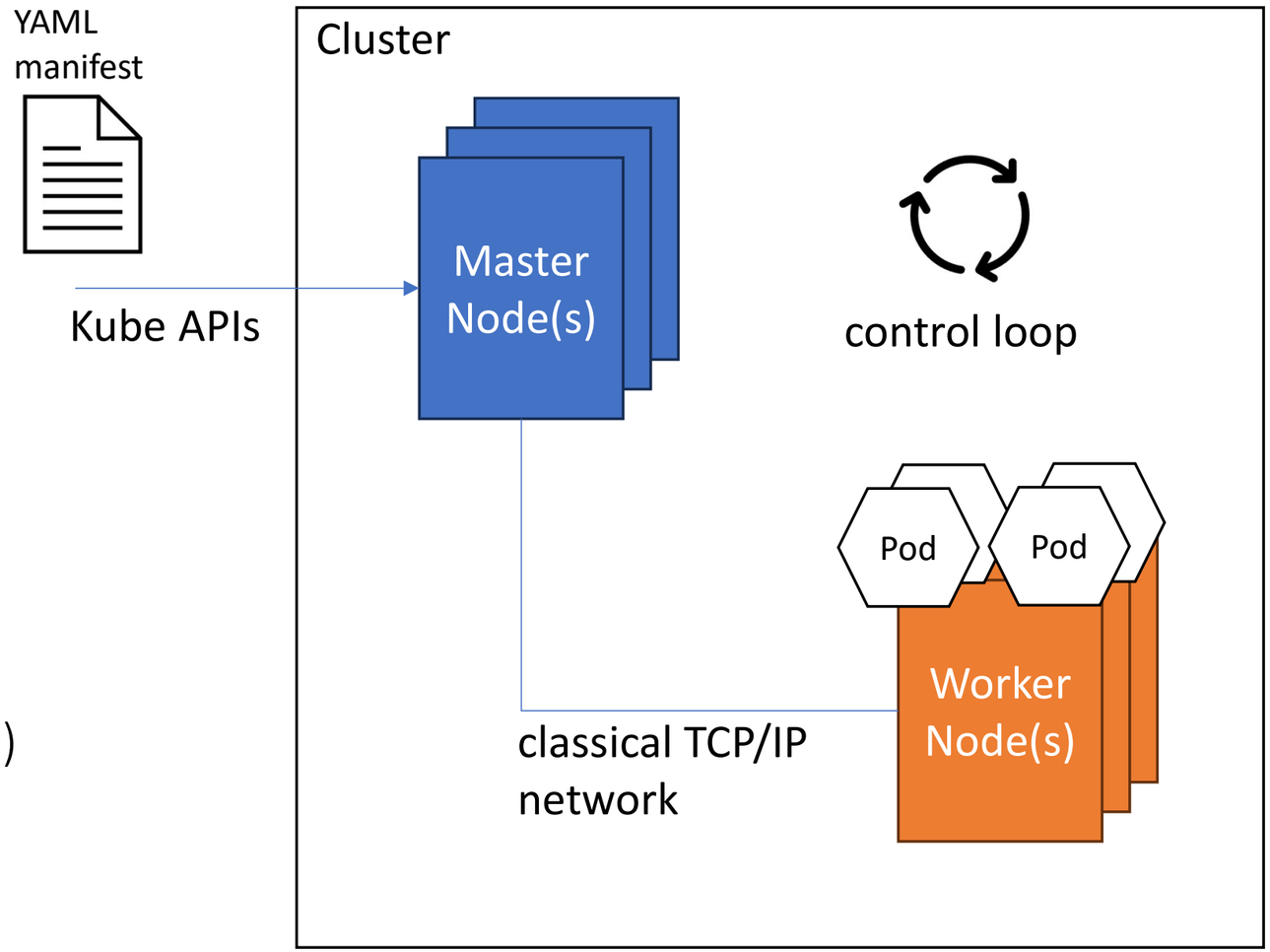


Kubernetes

- standard de facto of container orchestration
- master vs worker nodes
- cloud-native applications run in pods
- **declarative approach** (YAML files)

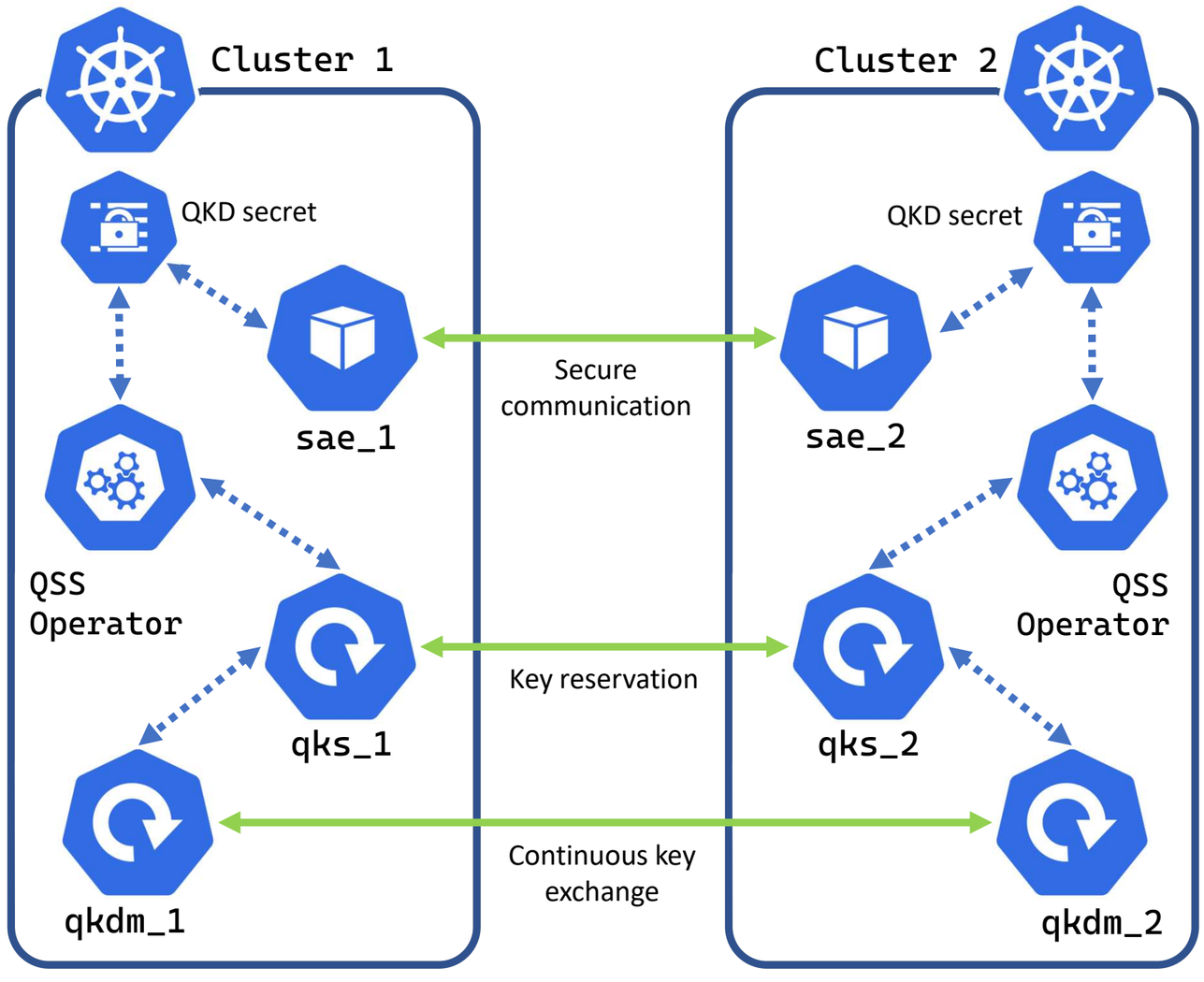
- **control-loop strategy**
  - Controllers
  - Custom Resources

- **Operators** (SDK for Go, Python, Ansible)
  - deploy, monitor, manage, and maintain applications



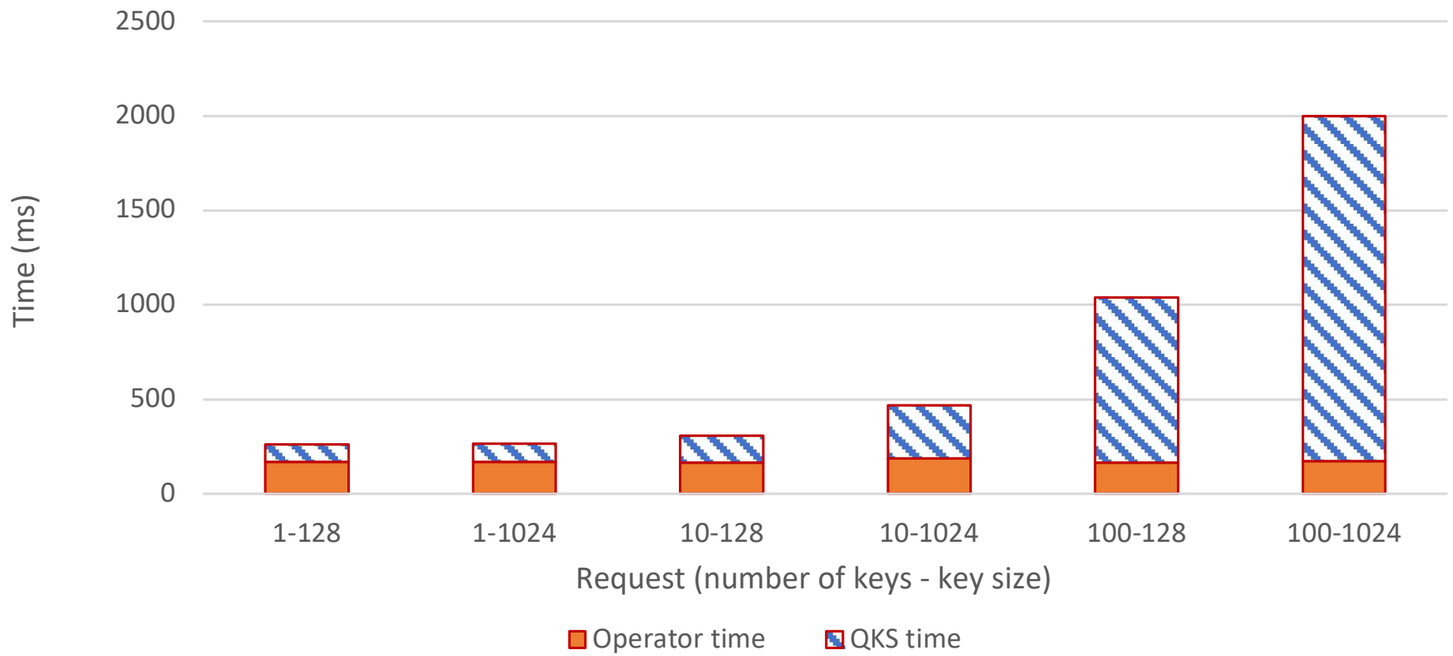
QKD in Kubernetes clusters

- a cluster can be seen as a single QKD node
- master nodes retain secret resources
- QSS Operator
  - developed in Python (SDK)
  - custom controller deploys QSS 2.0
- Workflow
  1. QSS Operator setup
  2. continuous key exchange
  3. SAE registration
  4. Key Request from SAE
  5. Key Reservation Process (KRP)
  6. QKD secret creation (in the specific SAE namespace)



QSS 2.0: PTP exchange

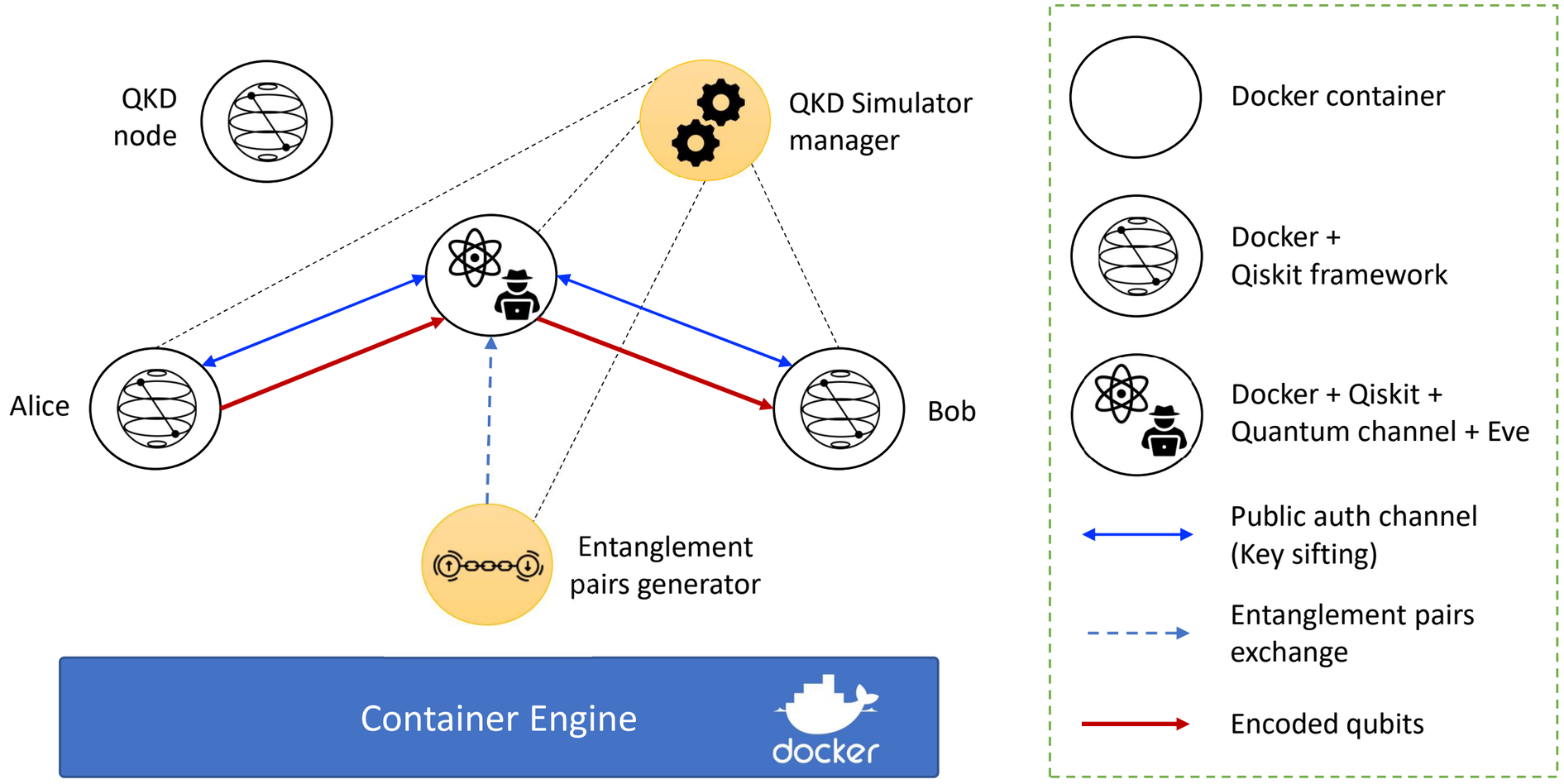
- **x2 physical nodes** (i5-5300U CPU @ 2.30 GHz, 16GB of RAM, Ubuntu 20.04 LTS)
- Kubernetes cluster, QSS Operator, QKS 2.0
- the system quickly adapts to changes and notify errors to the upper layers
- **22.8 Kbps** (max Key Rate)
- **the secret engine** is a bottleneck
- improvements:
  - larger key size (>128 bit)
  - horizontal scaling
  - different backend



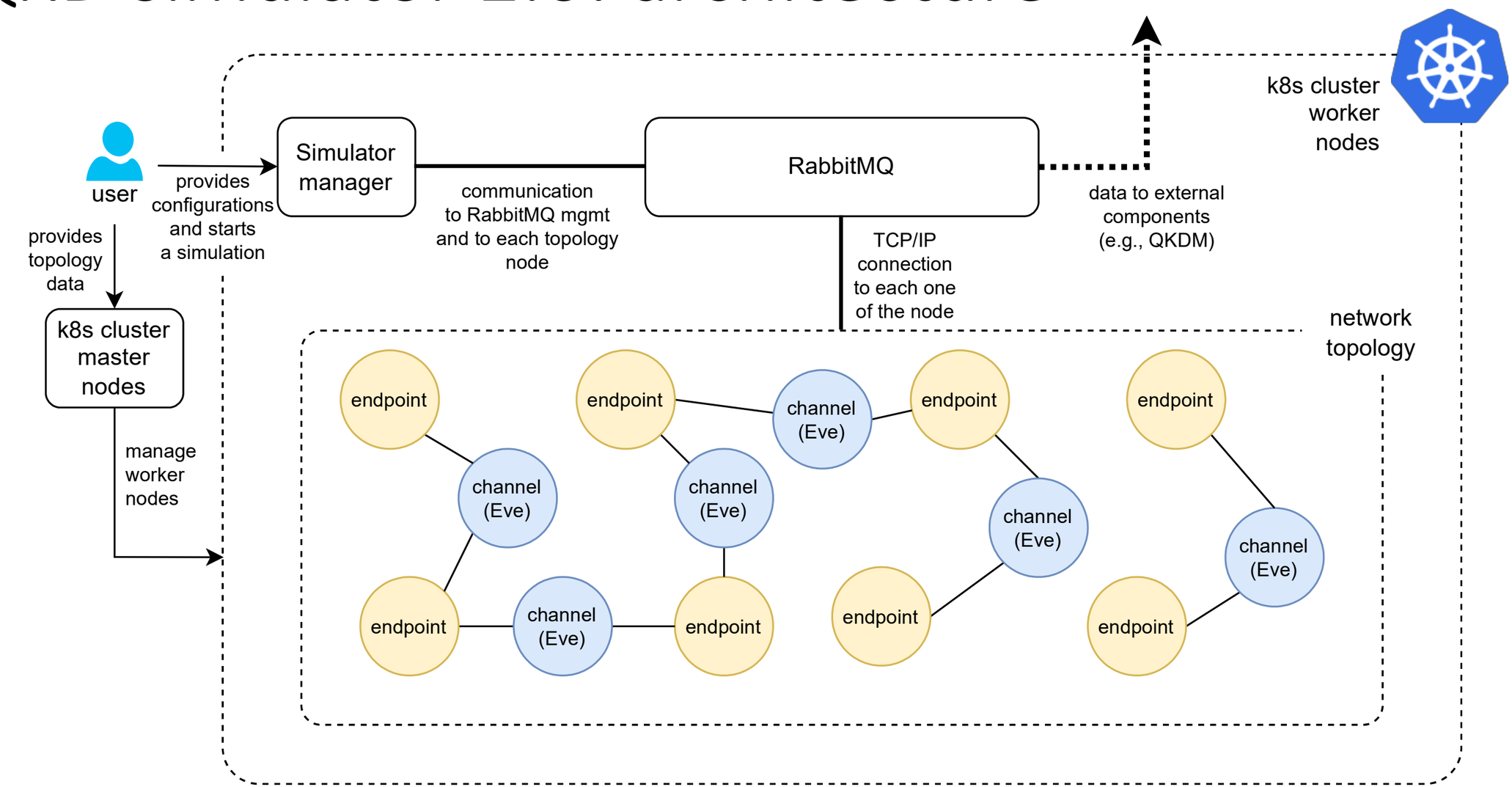
QKD simulator: motivation

- several QKD/Quantum Networks simulators
  - discrete-event simulation
  - key rate calculation
  - NetSquid, SimulaQron, QuISP, SeQUeNCe
- none of them offer a reliable:
  - distributed and scalable simulation
  - interface to a QKD network software stack
  - transparent support for different backends…
  - …and protocols beyond QKD
- create a generic distributed simulator that:
  - addresses endpoints, quantum channel (noise), and eavesdropper separately
  - supports classical post-processing
  - can be extended to protocol beyond QKD
  - supports quantum circuit-based simulation

QKD simulator: architecture

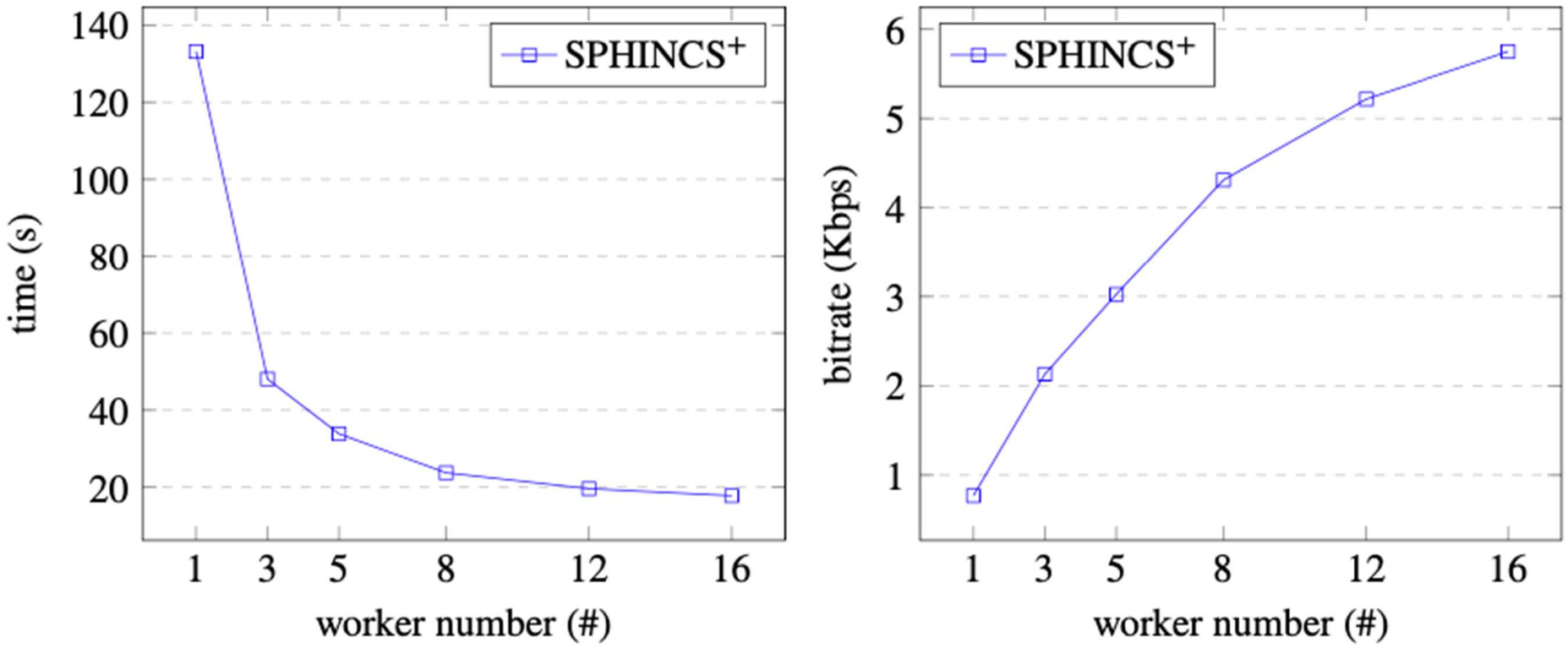


QKD simulator 2.0: architecture



QKD simulator 2.0: PTP performance

- x1 VM, 16 vCPU (Intel Xeon @ 2.30 GHz), 40GB of RAM, Ubuntu 20.04 LTS
- OLD VERSION -> 450 bps max throughput (5-qubit Qiskit register)



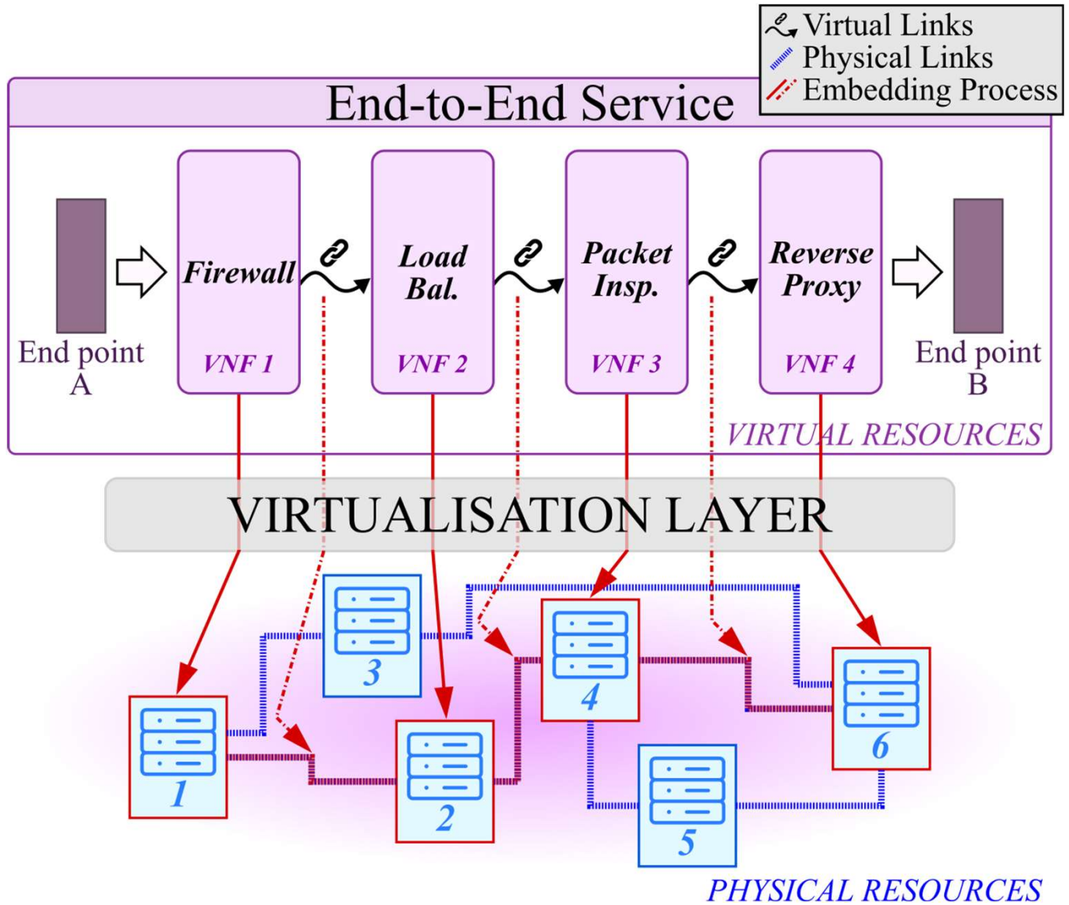
VNF Embedding Problem (VNFEP)

- Virtual Network Functions (VNFs)
  - NFV paradigm
  - decoupling hardware and software

- VNFEP (Substrate Network, virtualisation)

- Virtual Network Security Functions
  - Service Function Chains (SFCs)
  - Security-as-a-Service (SECaaS)

- VNFEPs are typically modelled as ILP o MILP (can be NP-hard)



VNFEP using QA: Methodology (I)

- Quantum Annealing and D-Wave system
  - **energy** of a specific solution (the lower the better)
  - **Quadratic Unconstrained Binary Optimisation (QUBO) formulation**
  - embed the problem into the Quantum Process Unit (QPU) of the D-Wave annealer

- **QUBO formulation of a generic VNFEP problem**
  - the number of QUBO variables depends on the number of VNF per chain, SFCs, nodes and links
  - **cost function**
    - allocation of a VNF on a specific server
    - costs can be chosen to reflect power consumption, monetary expenses of specific servers/regions
  - **constraints** modelled as penalty
    - allocation
    - continuity
    - resources

$$E_{problem} = E_{cost} + \sum_{v \in V} \lambda_v \cdot P_v$$

VNFEP using QA: Methodology (II)

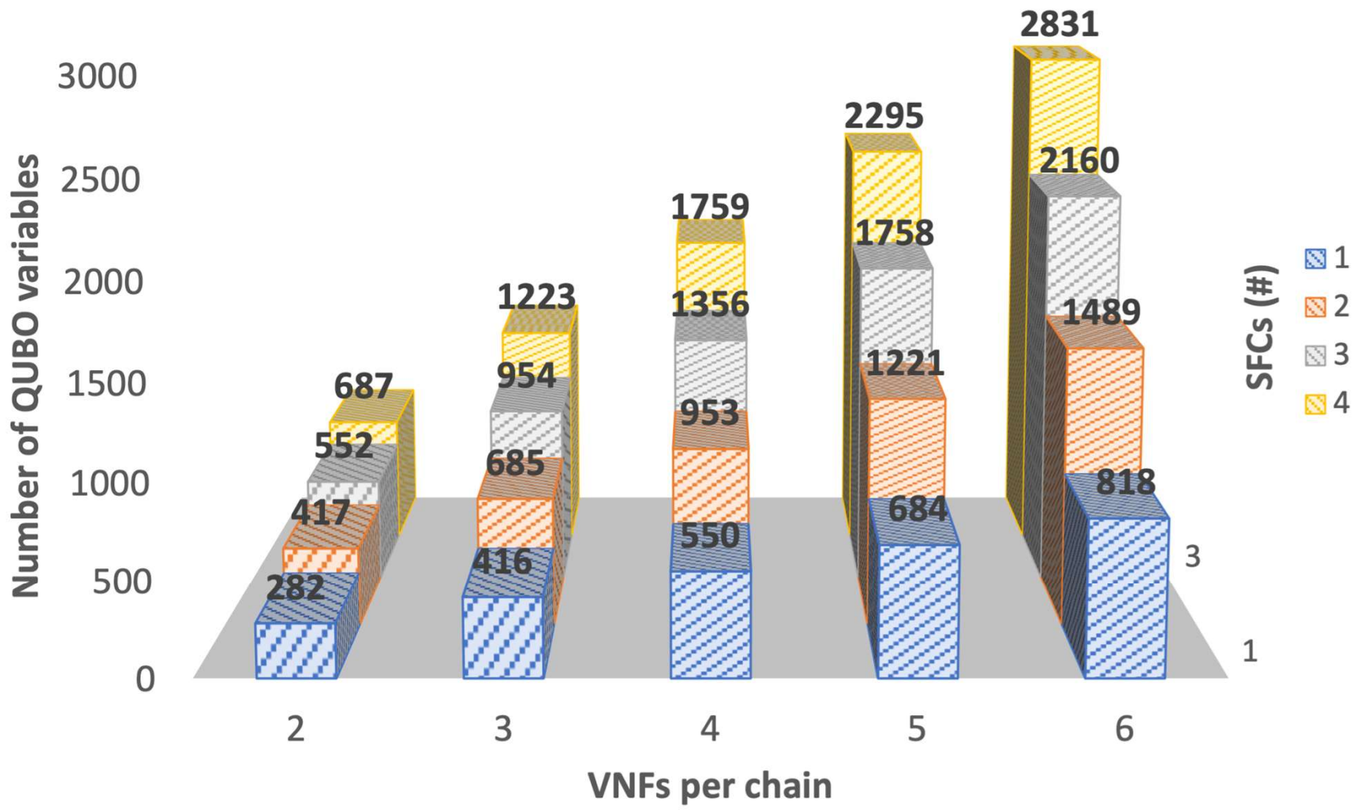
- Three solvers to compare classical and quantum results
  - Tabu search
  - Simulated Annealing
  - Quantum Annealing (QPU solver)

- Validation of the QUBO formulation
  - random generated networks
  - using classical solvers to find a solution
  - constraints verified setting the costs to zero

- Solving the VNFEP using the QPU solver
  - tuning of the solver parameters (e.g., chain strength, QA time)
  - minor-embedding problem
  - comparing results with classical solvers
    - enabling/disabling different types of constraints checks

QUBO variables for a topology of 20 nodes

- classical solvers can work up to around 2000 QUBO variables
- QPU can work up to around 150 variables
- QPU embedding is an issue
  - QUBO vars vs target qubits
- QPU embedding problem example
  - x2 SFCs, 2/3 VNFs
  - 10 nodes topology (17 links)
  - QUBO var/ target qubits
    - 48/163 (partial constraints)
    - 138/2194 (all constraints)



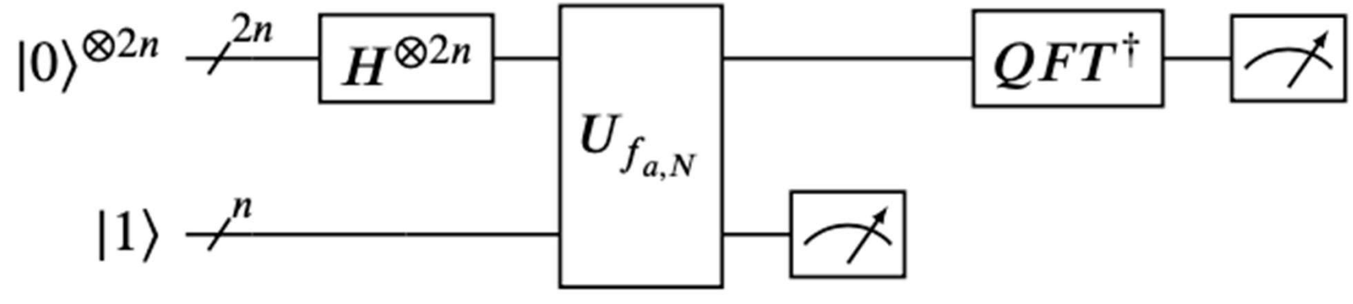
VNFEP using QA: results

- Topology: 7 nodes / 12 links
- Required allocation: 3x SFCs (x2 VNFs)

Problem	Solver	QUBO variables	Qubits	Solver time (s)	QUBO time (s)	Chain Strength	Lowest Energy	QA time (s)	Reads
Allocation and Continuity Constraints Only	SA	72	-	0,1821	0,0021	-	0	-	10^2
	Tabu	72	-	2,1947	0,0021	-	0	-	10^2
	QA	72	250 →	0,0333	0,0021	50	→ 0	50	10^2
Allocation and Continuity Constraints with Costs	SA	72	-	0,1862	0.0047	-	27	-	10^2
	Tabu	72	-	2,1319	0.0047	-	27	-	10^2
	QA	72	252 →	0,0299	0.0047	50	→ 27	50	10^2
Full VNFEP Formulation	SA	128	-	0,3918	0.0355	-	27	-	10^2
	Tabu	128	- →	2,1600	0.0355	-	→ 27	-	10^2
	QA	128	1335 →	2,7550	0.0355	50	→ 75	50	10^4

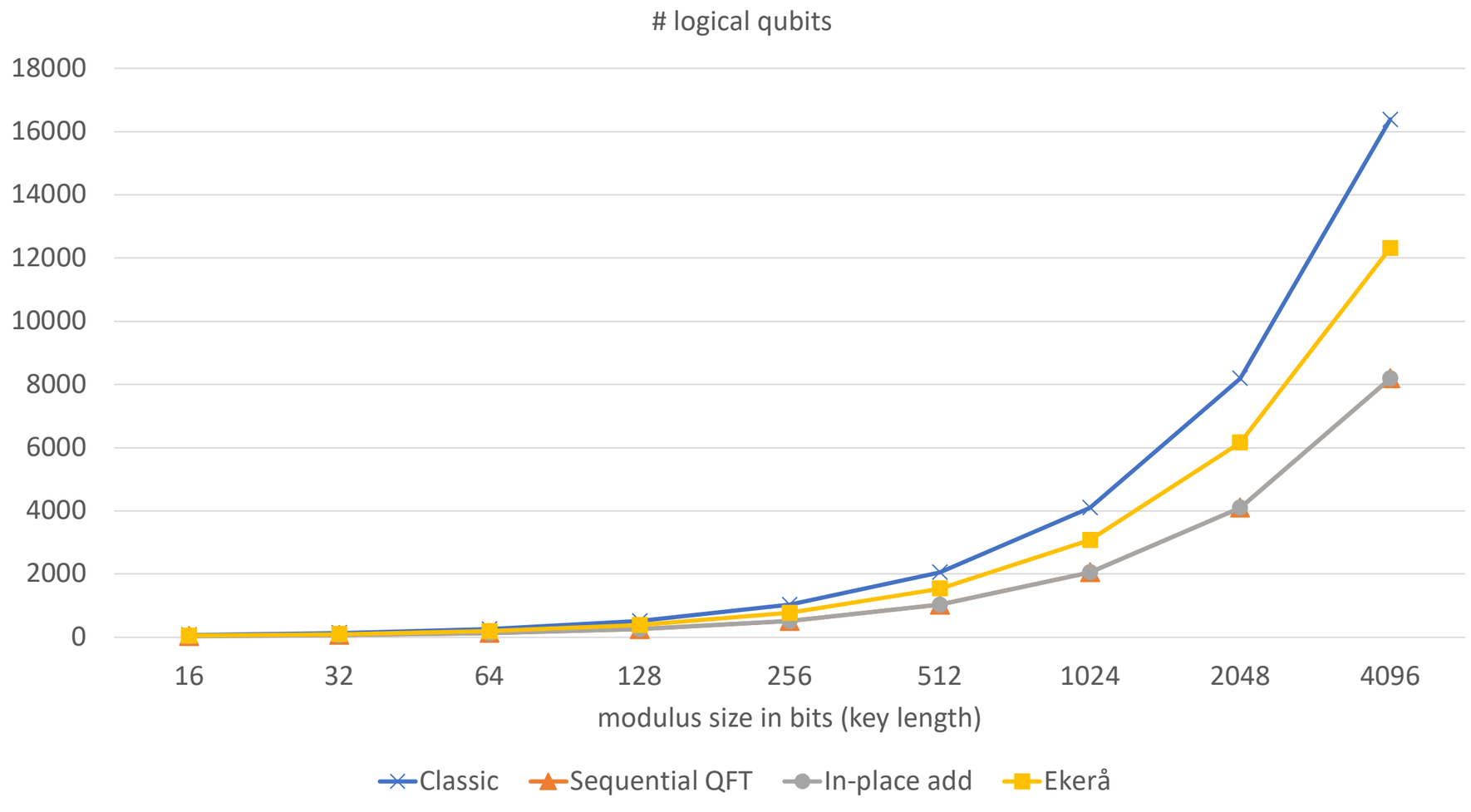
Shor’s algorithm analysis

- Prime factoring
- period r of the function $a^x (mod\ N)$
- Quantum Circuit
  - Quantum Phase Estimation (QPE)
  - Quantum Fourier Transform (QFT)
  - number of qubits / depth

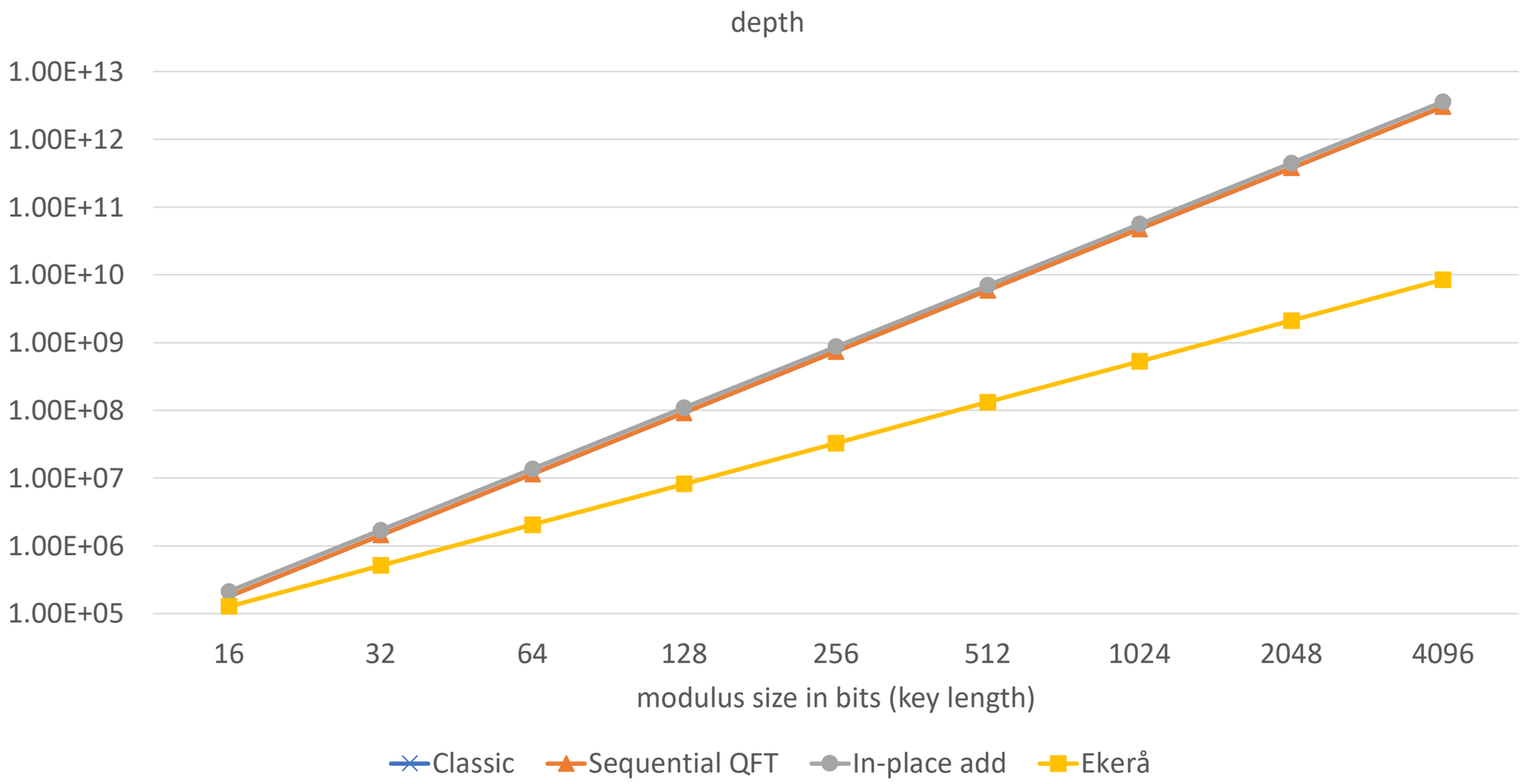


- Methodology
  - analysis of the first Shor’s proposal (Prime Factoring, RSA)
  - analysis of the improvements and specific implementation available in the literature
    - sequential QFT
    - in-place addition
    - Ekerå
  - implementation and testing with the Qiskit framework
  - analysis of the application of Shor’s for ECDLP
  - implementation of the quantum circuit components (Qiskit framework)
  - test of the subcomponents

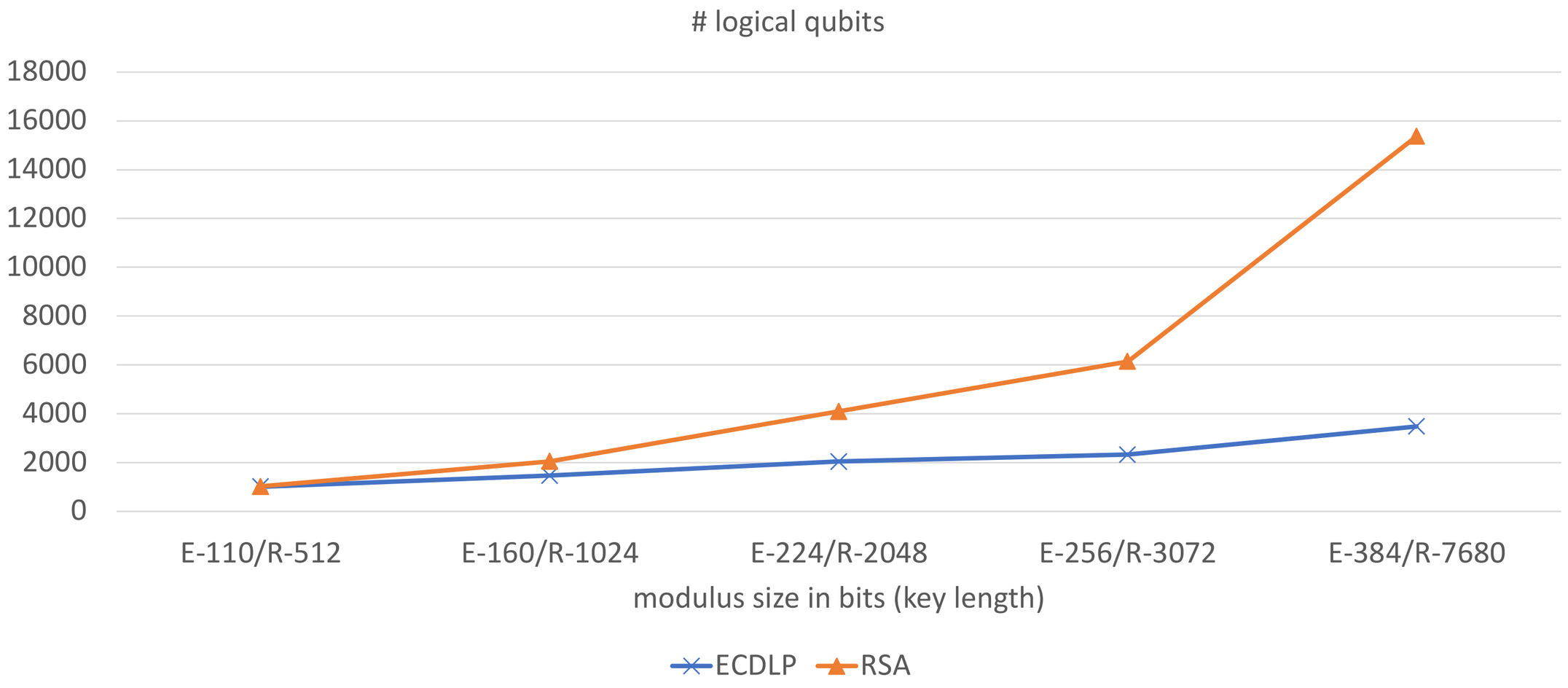
Resource estimation (RSA) - # qubits



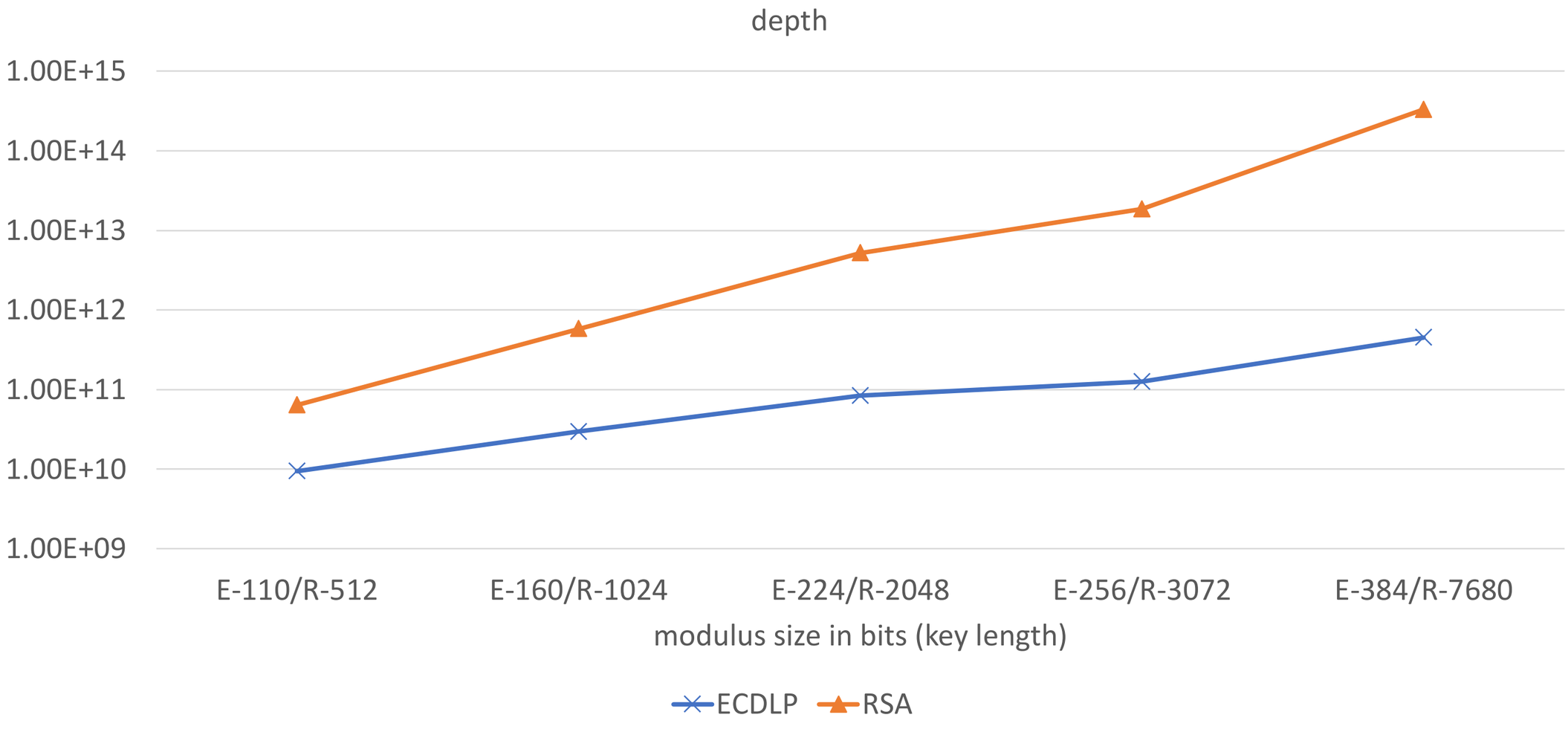
Resource estimation (RSA) - depth



Resource estimation (RSA vs ECDLP) - # qubits



Resource estimation (RSA vs ECDLP) - depth



Final remarks

- QKD can be integrated into SDIs and widely adopted platforms
  - test on large-scale networks required
- Quantum Annealing for solving optimisation problems
  - is promising in the cybersecurity domain
- Quantum Threat
  - is consistent but requires remarkable hardware progress to be effective
  - cybersecurity needs to react in time regardless of this conclusion

Acknowledgments:
Prof. Antonio Lioy
TORSEC research group
collegues from DAUIN and DET

Thank you all for attending this viva!

Any questions?