Experimental authentication of Quantum Key Distribution with Post-Quantum Cryptography

Bhardwaj Aditya

Eötvös Loránd University

June 29, 2023

Content

- Quantum Supremacy
- Implications
- Quantum safe cryptography
- Authentication for QKD
- Solution: PKI + PQC
- Aigis-sig
- Asymmetric variants of LWE and SIS
- Module LWE/SIS
- Aigis-sig algorithms
- Results
- References

Quantum Supremacy - 2019

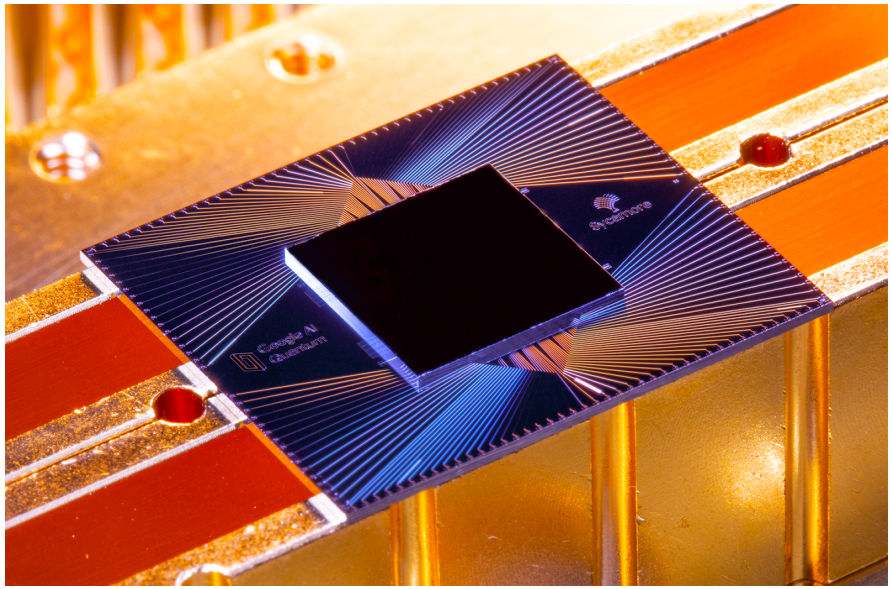


Figure: Sycamore processor

- 200 seconds vs 10,000 years (IBM supercomputer - Summit)
- 2022 Paper titled "Solving the Sampling Problem of the Sycamore Quantum Circuits [1]"

Implications in cryptography

Shor algorithm - 1994

- Can do factoring in polynomial time

Grover's search algorithm - 1996

- search for an element in $\sqrt{N}$ steps for N elements
- Could brute-force a 128-bit symmetric cryptographic key in roughly 2^{64} iterations, or a 256-bit key in roughly 2^{128} iterations

Quantum safe cryptography

Perfect secrecy

- Shannon one-time pad encryption
- Problem of key-distribution

Quantum key distribution can provide perfect security in **theory**

- Prone to errors and disturbances
- Improving hardware devices
- Detect eavesdropping

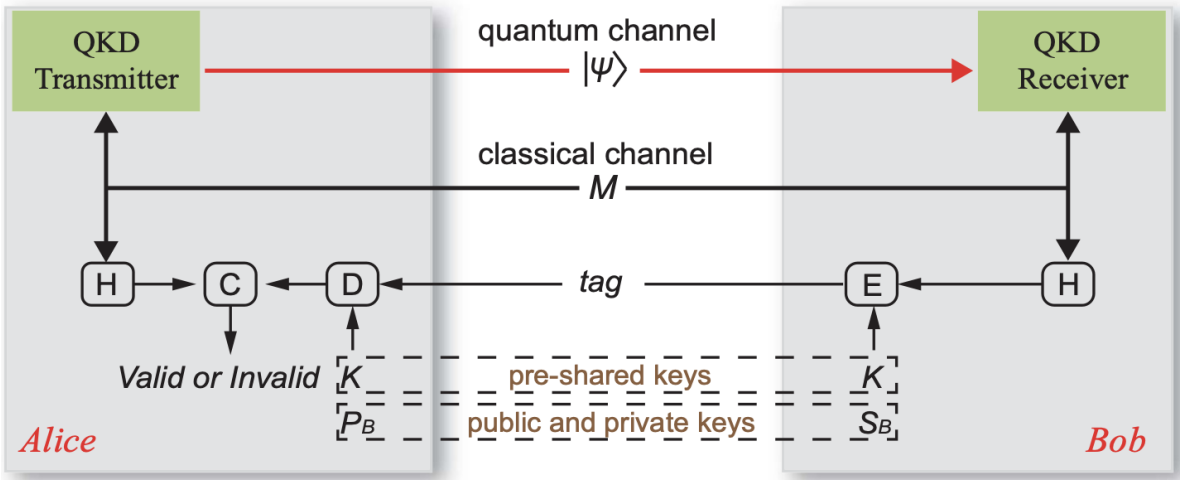
QKD requires an authenticated channel

Authentication for QKD

- Confidentiality not required
- Still needs authentication - MITM attack

Processes that need authentication: Basis sifting, error correction verification, random number transfer needed for privacy amplification, and final key verification

- Pre-share a small amount of symmetric seed keys



Solution: PKI + PQC

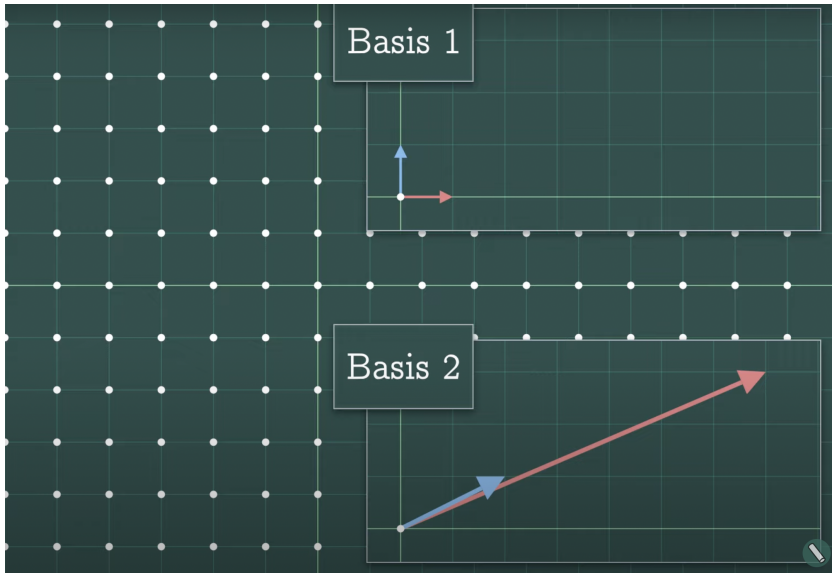
Pre-sharing keys between 2 users means $C_n^2 = n(n\text{-}1)/2$ pairs of keys.

For 100 users, **4950** pairs of keys

- PKI enables user to get one certificate each
- PQC algorithm will protect against quantum attacks
- Leverage short-term security

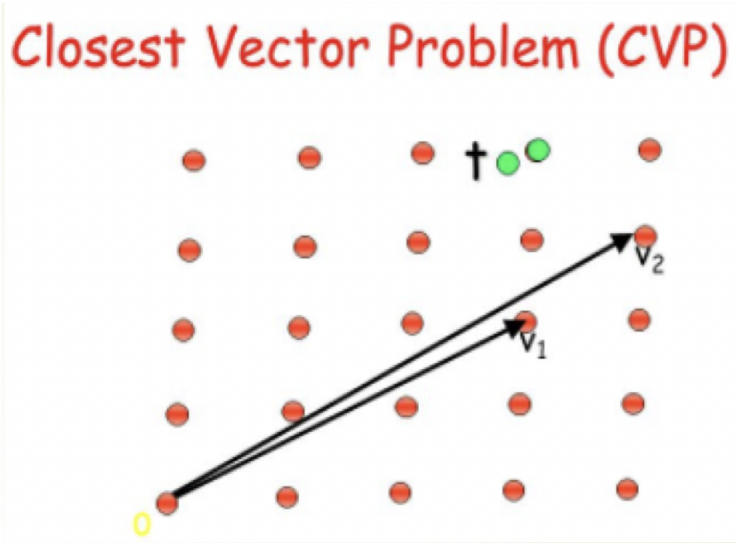
PQC in QKD authentication is safer and efficient

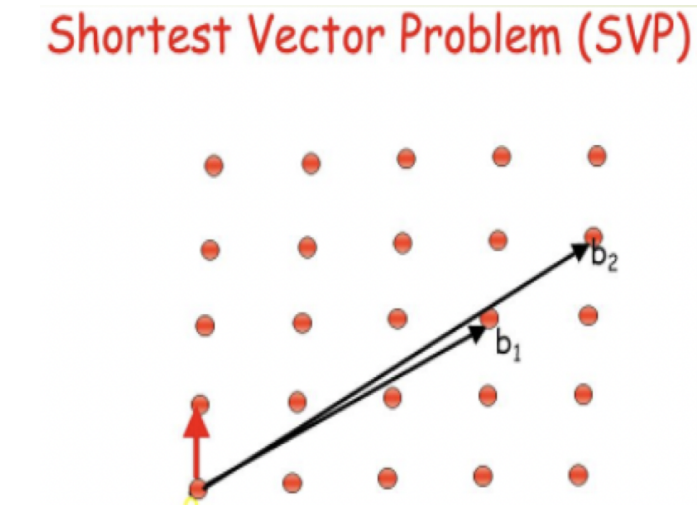
PQC Algorithm: Aigis-sig



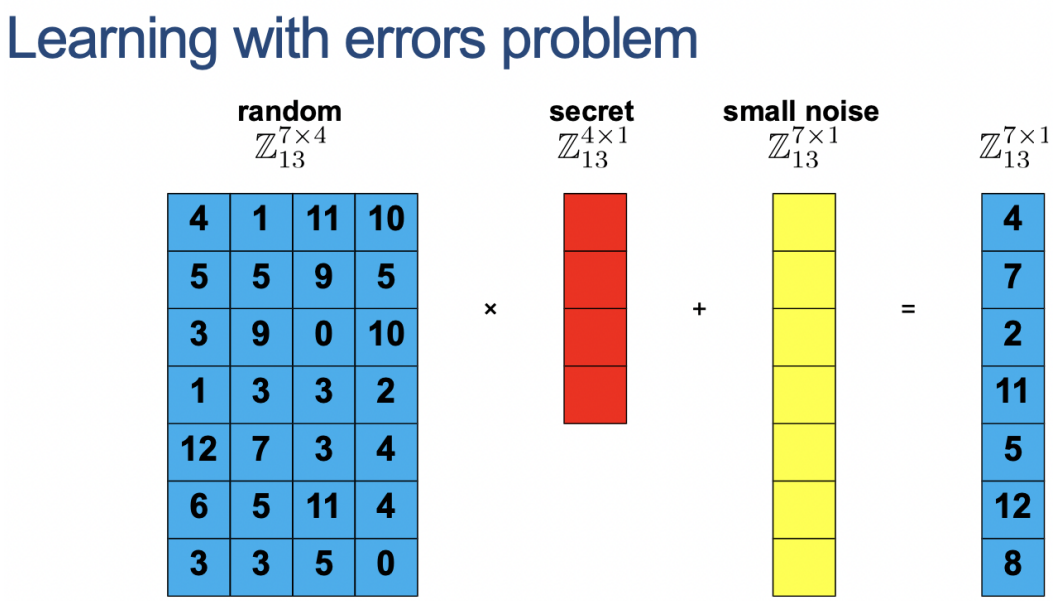
- Lattice based PQC algorithm
- Infinite set of points generated by addition,subtraction of vectors
- Same lattice can be generated by different 'basis'

Problems


Closest Vector Problem (CVP)
t
v_2
v_1
0


Shortest Vector Problem (SVP)
b_2
b_1

Aigis-sig uses LWE and SIS

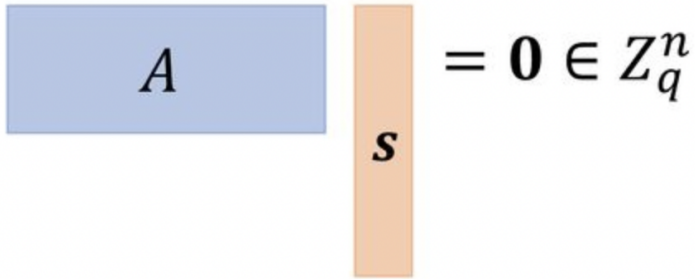


- In 2005, Regev [2] showed that the decision version of LWE is hard assuming quantum hardness of the lattice problem GapSVP.

Aigis-sig uses LWE and SIS

Short Integer Solution (SIS) Problem

- Input: Random matrix A $\in Z_q^{n \times m}$
- Goal: Find non-trivial $\boldsymbol{s} \in Z^m$ with $A\boldsymbol{s} = 0 \bmod q$ and $\|s\|_\infty < \beta$



- In 1996, Ajtai [3] showed that SIS is secure if SVP is hard for some specific value of the parameter β

Asymmetric variants: ALWE and ASIS

ALWE

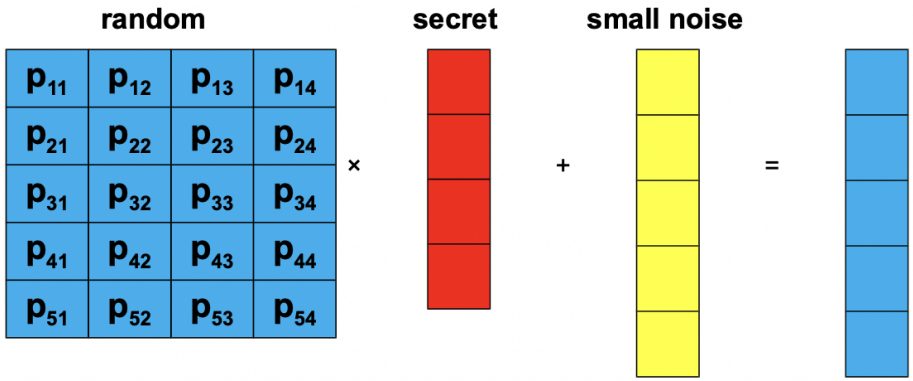
the ALWE problem $\mathrm{ALWE}_{n,m,q,\alpha_1,\alpha_2}$ asks to find out $\mathbf{s} \in \mathbb{Z}_q^n$ from samples $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}, \mathbf{s} \xleftarrow{\$} \chi_{\alpha_1}^n, \mathbf{e} \xleftarrow{\$} \chi_{\alpha_2}^m$.

ASIS

Informally, the ASIS problem $\mathrm{ASIS}^{\infty}_{n,m_1,m_2,q,\beta_1,\beta_2}$ refers to the problem that, given a random $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m_1+m_2)}$, find out a non-zero $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T)^T \in \mathbb{Z}^{m_1+m_2}$ satisfying $\mathbf{Ax} = \mathbf{0} \bmod q$, $\|\mathbf{x}_1\|_\infty \leq \beta_1$ and $\|\mathbf{x}_2\|_\infty \leq \beta_2$.

Aigis-sig: Module-LWE/SIS

Module learning with errors problem



every matrix entry is a polynomial in $\mathbb{Z}_q[x]/(x^n+1)$

Ring elements $r \in R_q = \mathbb{Z}_q[X]/(X^n+1)$:

- Coefficients integers modulo q
- Degree at most $n-1$ i.e.
  $r = r_0 + r_1 \cdot X + \cdots + r_{n-1} \cdot X^{n-1} \in \mathbb{Z}_q[X]/(X^n+1)$
- Coefficient Embedding $r = (r_0, \ldots, r_{n-1}) \in \mathbb{Z}_q^n$

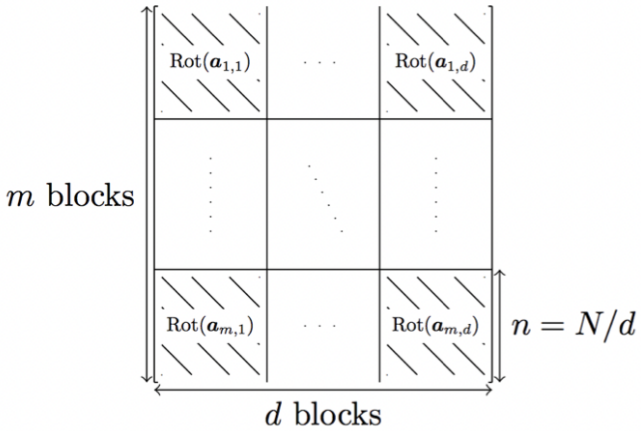
Understanding Module LWE/SIS

Ring-LWE versus Module-LWE

Ring-LWE

4	1	11	10
3	4	1	11
2	3	4	1
12	2	3	4
9	12	2	3
10	9	12	2
11	10	9	12

Module-LWE



Definition: Rotational shift operator on $\mathbb{R}^n (n \geq 2)$ is denoted by rot, and is defined as:

$$\forall \boldsymbol{x} = (x_1, \ldots, x_{n-1}, x_n) \in \mathbb{R}^n : \mathrm{rot}(x_1, \ldots, x_{n-1}, x_n) = (x_n, x_1, \ldots, x_{n-1})$$

Understanding Module LWE/SIS

Learning from example of Crystals-KYBER [4]

The special form of this matrix means that the matrix multiplication

$$\begin{pmatrix} c_0 & c_1 & c_2 & \dots & c_{255} \\ -c_{255} & c_0 & c_1 & \dots & c_{254} \\ -c_{254} & -c_{255} & c_0 & \dots & c_{253} \\ \vdots & & & \ddots & \vdots \\ -c_1 & -c_2 & -c_3 & \dots & c_0 \end{pmatrix} \begin{pmatrix} \ell_{255} \\ \ell_{254} \\ \vdots \\ \ell_0 \end{pmatrix} = \begin{pmatrix} r_{255} \\ r_{254} \\ \vdots \\ r_0 \end{pmatrix}.$$

Is consistent with the multiplication
$(c_0 + c_1X + \cdots c_{255}X^{255})(\ell_0 + \ell_1X + \cdots + \ell_{255}X^{255}) = r_0 + r_1X + \cdots + r_{255}X^{255}$
when the multiplication is performed modulo $X^{256} + 1$.

This means that in Kyber512, the expression $\mathbf{As} + \mathbf{e}$ can equally be thought to represent the following matrix equation over the ring $\mathbb{Z}[X]/\langle q, X^{256} + 1\rangle$

$$\begin{pmatrix} n_{1,1}(X) & n_{1,2}(X) \\ n_{2,1}(X) & n_{2,2}(X) \end{pmatrix} \begin{pmatrix} s_1(X) \\ s_2(X) \end{pmatrix} + \begin{pmatrix} e_1(X) \\ e_2(X) \end{pmatrix}$$

Aigis-sig - Key Generation

Algorithm 1: Key Generation Algorithm

Function KeyGen

$A \leftarrow R_q^{k \times l}$;
$s_1, s_2 \leftarrow S_\eta^l \times S_\eta^l$;
$t = As_1 + s_2$;
$pk = (A, t), sk = (s_1, s_2, pk)$;
return (pk, sk)

generates a $k \times l$ matrix A

each entry is a polynomial in $R_q = Z_q[x]/(x^n + 1)$

Coeff of s_1 & s_2 is an element of R_q of size at most η

secret key vectors

All operations are assumed to be over R_q

Aigis-sig - Signature

Algorithm 2: Signature Algorithm

Function Sign $sk=(s_1, s_2, pk), \mu$

  repeat

    $y \leftarrow S^{l+k}_{\gamma_1 - 1}$;

    $w = Ay$;

    $c = \text{Hash}(\text{HighBits}(w, 2\gamma_2)||\mu)$;

    $z = y + cs_1$;

  until $||z||_\infty < \gamma_1 - \beta$ *and* $\text{LowBits}(Ay - cs_2, 2\gamma_2) < \gamma_2 - \beta$;

  return $\sigma = (z, c)$

y → vector of polynomials, coeff < γ_1 SK

→ set strategically =(s_1, s_2

$w = Ay$ and $w = w_1 . 2\gamma_2 + w_0$ pk)

↳ can be expressed μ = message

c → challenge and sign $z = y + cs_1$

β is set to be the maximum possible coeff of cs_i

Aigis-sig - Verification

Algorithm 3: Verification Algorithm

Function Verify $pk{=}(A,t),\sigma = (z,c),\mu$
  if $||z||_\infty < \gamma_1 - \beta$ *and*
   $c = \text{Hash}(\text{HighBits}(Az - ct, 2\gamma_2)||\mu)$ **then**
    return *true ;*
  return *false ;*

$pK = (A,t) \quad \sigma = (z,c) \quad \mu$

The step in signature scheme helps to ensure

$\text{HighBits}(Az - ct, 2\gamma_2) = \text{HighBits}(Ay - cs_2, 2\gamma_2)$

Correctness

Correctness. Note that if $\|c\mathbf{t}_0\|_\infty < \gamma_2$, by Lemma 1 we have $\mathsf{UseHint}_q(\mathbf{h}, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$. Since $\mathbf{w} = \mathbf{A}\mathbf{y}$ and $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$, we have that

$$\mathbf{w} - c\mathbf{s}_2 = \mathbf{A}\mathbf{y} - c\mathbf{s}_2 = \mathbf{A}(\mathbf{z} - c\mathbf{s}_1) - c\mathbf{s}_2 = \mathbf{A}\mathbf{z} - c\mathbf{t},$$
$$\mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0 = \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d,$$

where $\mathbf{t} = \mathbf{t}_1 \cdot 2^d + t_0$. Therefore, the verification algorithm computes

$$\mathsf{UseHint}_q(\mathbf{h}, \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2).$$

As the signing algorithm checks that $\mathbf{r}_1 = \mathbf{w}_1$, this is equivalent to

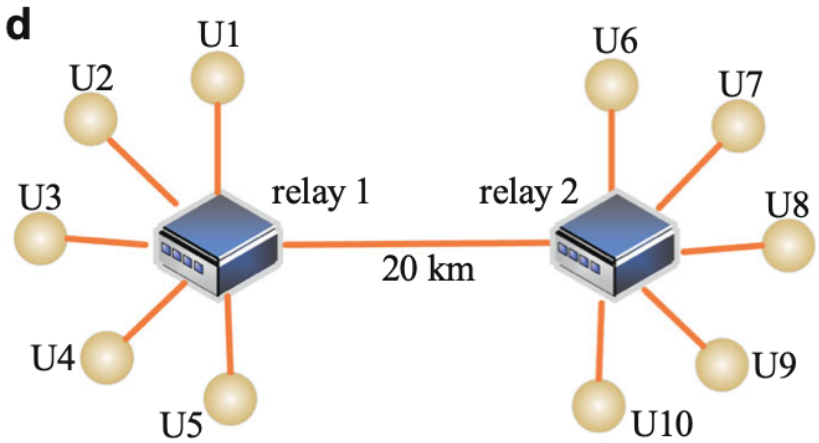
$$\mathsf{HighBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2).$$

Hence, the $\mathbf{w}_1$ computed by the verification algorithm is the same as that of the signing algorithm, and thus the verification algorithm will always return 1.

Paper: Tweaking the Asymmetry of Asymmetric-Key Cryptography on Lattices: KEMs and Signatures of Smaller Sizes [5]

Results

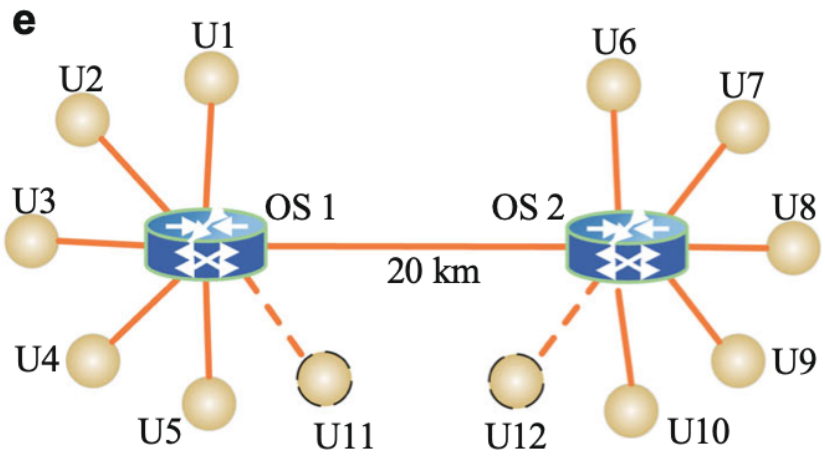
10-node QKD metropolitan area network



- Trusted relay is needed to manage pre-shared keys
- For 2 new users - Share 10 keys with previous users + 1 key for the new user - Total 21 keys

Results

10-node QKD network - replaced with Optical Switches



- Only need to trust CA
- For 2 new users - Only 2 new digital certificates instead of 21 pre-shared keys
- Better connectivity

References

1. Solving the sampling problem of the Sycamore quantum circuits - https://arxiv.org/abs/2111.03011
2. Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography - http://portal.acm.org/citation.cfm?id=1060590.1060603
3. Generating hard instances of lattice problems - https://dl.acm.org/doi/10.1145/237814.237838
4. https://crypto.stackexchange.com/questions/104392/how-is-mlwe-used-for-key-generation-in-kyber
5. Tweaking the Asymmetry of Asymmetric-Key Cryptography on Lattices: KEMs and Signatures of Smaller Sizes - https://link.springer.com/chapter/10.1007/978-3-030-45388-6_2