Quantum-safe lattice-based abstractions in functional programs

Daniel Barrero[1] **Valérie Gauthier**[1] **Nicolás Cardozo**[1]
[1]Universidad de los Andes
dr.barrero2562@uniandes.edu.co

Quantum-safe Cryptography

Shor's quantum algorithm, published in 1994, demonstrated that factorization-based cryptography is no longer safe in the face of large-scale quantum computers.

$$S_{i,j} = \begin{array}{c|cccc} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ |00\rangle & 1 & 0 & 0 & 0 \\ |01\rangle & 0 & 1 & 0 & 0 \\ |10\rangle & 0 & 0 & 1 & 0 \\ |11\rangle & 0 & 0 & 0 & e^{i\theta_{k-j}} \end{array}$$

In 2015, the NSA announced its transition to quantum-resistant algorithms, in recognition of threats such as (1) *Harvesting attacks:* store today's keys and cyphertexts to break later, (2) *Forging signatures for old keys*, (3) *Implementing new cryptography at scale takes a long time*.

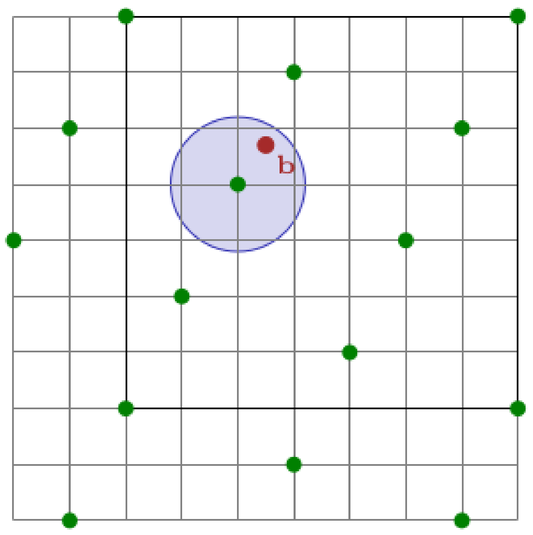
The following are the winners as of August 2024:

- CRYSTALS-kyber*
- CRYSTALS-dilithium*
- FALCON*
- SPHINCS+

Lattice-based cryptography

Lattices offer the possibility of grounding cryptography on operations as simple as matrix multiplication and vector addition, where a lattice is a periodic grid in space (of any dimension), defined as the $\mathbb{Z}$-span of a basis for $\mathbb{R}^n$:

$$L(B) = \{a_1\mathbf{b_1} + ... + a_n\mathbf{b_n} | a_i \in \mathbb{Z}, \mathbf{b_i} \in B\}.$$



Computationally hard problems arise when the dimension of the lattice becomes large. In the relation

$$\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t,$$

the public key is the random matrix $\mathbf{A}$ and the secret key is the vector $\mathbf{s}$. Recovering such key from the cyphertext $\mathbf{b}$ is as hard as solving the *shortest vector problem* (SVP, SVP_γ), which is known to be NP-hard.

Current Shortcomings

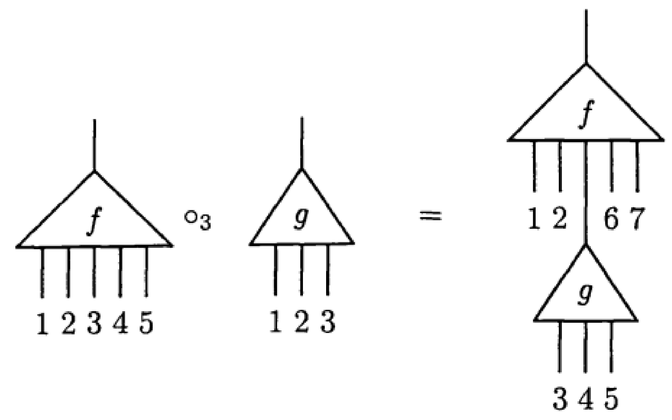
The best PQC algorithm so far is *CRYSTALS-kyber*, which in a chat encryption scenario is known to

- Increase runtime by a factor of 2.3
- Increase energy consumption by 3.1
- Have 70 times more data overhead

These complications are due to internal hashing and *discrete Fourier transforms* (DFTs), which occur iteratively in the encryption process.

Operad-based cryptography

We may therefore attempt to simplify such sequences of operations. For this we will use the theory of *operads*, whose objects represent *operation composition*.



The fact that the best standards rely on property-preserving transformations (a.k.a. homomorphisms) to simplify computational complexity encourages this approach even further.

A declarative language for quantum-safe cryptography

- The implementations of today's best protocols are written in languages such as C, which lack the expressiveness to express the operations naturally. This adds computational overhead and hides the intrinsic complexity of the scheme.
- Functional languages such as Haskell allow programmers to define category-theoretic abstractions in a way that is mathematically clear and natural.

```haskell
class (Graded f) => Operad (f :: Nat -> *) where
  ident :: f (S Z)
  compose :: f n -> Forest f m n -> f m
instance Operad MoveTree where
    ident = Leaf
    compose Leaf (Cons Leaf Nil) = Leaf
```

- In this project, we create the language SILVER, which will be **safe, expressive, and maintainable**. This will follow from the almost verbatim correspondence between mathematical definitions and type/typeclass declarations in our purely functional language (c.f. *Curry-Howard correspondence*), together with the type-check requirement for the execution of a program.