Digital signatures and key exchange protocols are the two most important public key cryptographic primitives used in the electronic transmission of data. The goal of key exchange is to preserve the secrecy of the communication, while the goal of digital signatures is to guarantee the authenticity of the exchanged messages. Constructions of digital signature schemes based on classical mathematical assumptions appeared shortly following the invention of public key cryptography in the late 1970s. And just like with key exchange, the most efficient variants are based on number-theoretic problems that are believed to be (sub-)exponentially hard for classical machines, but are solved in polynomial time by Shor’s algorithm on a powerful-enough quantum computer. Also, like for key exchange, the most efficient constructions that we believe to be quantum safe are based on the presumed hardness of lattice problems over polynomial rings.

One interesting difference between key exchange and digital signatures is that key exchange appears to inherently require that some mathematical problem be computationally hard. Digital signatures, on the other hand, can be generically constructed from any one-way function to a digital signature (the ‘challenge’ $H(\mu)$ for the proof (where $H$ is a public function that maps $\mu$ to something ‘random looking’), if the function $f$ is one way then only someone in possession of a trapdoor should be able to invert it. An example of such a construction based on a ‘factoring-like’ assumption is the RSA signature scheme [5].

The high-level ideas for lattice-based signatures follow the above blueprints, but the technical details are significantly more involved. The main reason for the complications is the different algebraic structure of the hard one-way function underlying lattice cryptography. While the domain of the function in discrete log and RSA-based one-way functions are groups, the domains in lattice-based signatures are sets that are not closed under any operation—in particular, they are elements in a group that have small norms. This crucial small norm requirement precludes us from using uniformly random masking as in Schnorr signatures or having a trapdoor for a bijective one-way function as in RSA signatures. These barriers have, nevertheless, been overcome and the resulting digital signatures are quite practical.

In 2017, the US National Institute of Standards and Technology (NIST) began a ‘competition’ for a quantum-safe key exchange and digital signatures standard. At the time of this writing, this process is in the third round and there are two lattice-based signatures remaining—each following one of the above high-level designs. The CRYSTALS-Dilithium [6] scheme follows the Schnorr framework, but adds a crucial rejection-sampling step to keep the size of the coefficients small. The FALCON scheme [7] utilizes a randomized trapdoor sampling technique that uses a secret trapdoor for $f^{-1}$ to produce random pre-images from a particular distribution. Because there is
Table 1. Approximate parameter sizes (in bytes) for the CRYSTALS-Dilithium and FALCON digital signature schemes at approximately 128-bit security levels.

|                  | CRYSTALS-Dilithium | FALCON |
|------------------|--------------------|--------|
| Public key       | 1300               | 900    |
| Signature        | 2400               | 650    |

no longer a bijection, it is crucial to also have the property that the distribution of the outputted pre-images does not leak information about the trapdoor. Both schemes are relatively fast and their parameters (public key + signature size) are the shortest of all quantum-safe signature schemes.

While both schemes are based on lattices, they have rather different characteristics. FALCON has very short parameters (see Table 1), but entails a rather complicated procedure for signature generation. In particular, it uses (an optimized version of) the GPV sampler [8], which requires floating-point arithmetic with approximately 64 bits of precision. Requiring such high precision means that subtle implementation errors may not get detected even with rigorous testing. Dilithium, on the other hand, has larger parameters but a very simple implementation where all the sampling in the signing is done in a power-of-2 range; it is thus much less prone to implementation errors. It is quite possible that in the future both schemes will be used for different applications.

Conflict of interest statement. None declared.

Vadim Lyubashevsky
Zurich Research Laboratory, IBM Research, Switzerland
E-mail: vlad@zurich.ibm.com

REFERENCES

1. Lamport L. Constructing digital signatures from a one-way function. Technical report. SRI International Computer Science Laboratory, 1979.

2. Merkle RC. A certified digital signature. In: Brassard G (ed.). Advances in Cryptology – CRYPTO’89 Proceedings. New York: Springer, 1989, 218–38.

3. Bernstein DJ, Hülsing A and Kölbl S et al. The SPHINCS+ signature framework. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2019, 2129–46.

4. Schnorr CP. Efficient identification and signatures for smart cards. In: Brassard G (ed.). Advances in Cryptology – CRYPTO’89 Proceedings. New York: Springer, 1989, 239–52.

5. Rivest RL, Shamir A and Adleman L. Commun ACM 1978; 21:120–6.

6. Ducas L, Kiltz E and Lepoint T et al. IACR Trans Cryptographic Hardware Embedded Syst 2018; 2018:238–68.

7. Prest T, Fouque P-A and Hoffstein J et al. National Institute of Standards and Technology. FALCON. https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions (10 May 2021, date last accessed).

8. Gentry C, Peikert C and Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the Fortyeth Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2008, 197–206.

National Science Review
8: nwab077, 2021
https://doi.org/10.1093/nsr/nwab077
Advance access publication 8 May 2021