PQC summary

zaterdag 7 januari 2023 15:49

Grover's algorithm: reduces brute-force attack by square root.

Introducing new cryptographic systems and making them widely-in-use takes a very long time (e.g. switching to ECC from RSA took about 30 years).

Shor's algorithm for period finding: breaks RSA, ECC, DH

Categories of public-key post-quantum systems

Code-based encryption: based on hardness of decoding error-correcting codes
Hash-based signatures: based on hardness of finding second pre-images to hash functions
Isogeny-based encryption: based on hardness of finding isogenies between elliptic curves over finite fields
Lattice-based encryption & signatures: relies on hardness of finding short vectors in some (typically special) lattice
Multivariate-quadratic signatures: relies on hardness of solving systems of multivariate equations over finite fields

Schemes from NIST competition

Name	Function	Hardness assumption
Kyber	KEM	Structured lattices
Dilithium	Signature scheme	Structured lattices
Falcon	Signature scheme	Structured lattices
SPHINCS+	Signature scheme	Hash functions
BIKE	KEM	Codes
Classic McEliece	KEM	Codes
HQC	KEM	Codes
SIKE	KEM	Isogenies

Legend
Winners
Advanced to round 4
Advanced to round 4, now broken

One-time signatures

Lamport's 1-time signature scheme: sign by releasing part of a secret, which is the preimage of the public key (i.e. $publickey = H(secret)$)

Each signature has 2*256 hash outputs (32 bytes each) as public key and the signature has 256*32 bytes

Winternitz 1-time signature scheme: $publickey = H^{16}(secret)$. Reveal $s = H^m(sk)$ as signature, which can be checked using $pk = H^{16-m}(s)$. To make a secure system, run two hash chains in opposite directions.

Merkle's multi-use signature scheme: merge one-time public keys into a single public keys by combining them through a Merkle tree; the public key in one node is the hash of the combined keys in the children of that node. Only the top node needs to be released as public key for this to work.

Usually requires storing 2^n one-time secret keys, but this can be optimized by generating them from a (short) seed.

The scheme could be made stateless by using it to sign other trees to expand the key space. (based on ideas by Goldreich & Levin)

Winternitz 1-time signature system

- Define parameter w. Each chain will run for 2^w steps.
- For signing a 256-bit hash this needs $t_1 = \lceil 256/w \rceil$ chains. Write m in base 2^w (integers of w bits):
$$m = (m_{t_1-1}, \ldots, m_1, m_0)$$
(zero–padding if necessary).
- Put
$$c = \sum_{i=0}^{t_1-1} (2^w - m_i)$$
Note that $c \leq t_1 2^w$.
- The checksum c gets larger if m_i is smaller.
- Write c in base 2^w. This takes $t_2 = 1 + \lceil \lfloor (\log_2 t_1 \rfloor + 1)/w \rceil$ w-bit integers
$$c = (c_{t_2-1}, \ldots, c_1, c_0).$$
- Publish $t_1 + t_2$ public keys, sign with chains of lengths
$$m_{t_1-1}, \ldots, m_1, m_0, c_{t_2-1}, \ldots, c_1, c_0.$$