Towards a Post Quantum OTS Scheme using QC-LDPC Codes

Giovanni Tognolini
(joint work with Alessio Meneghetti and Christian Picozzi)

University of Trento

1 December, 2022

In a nutshell

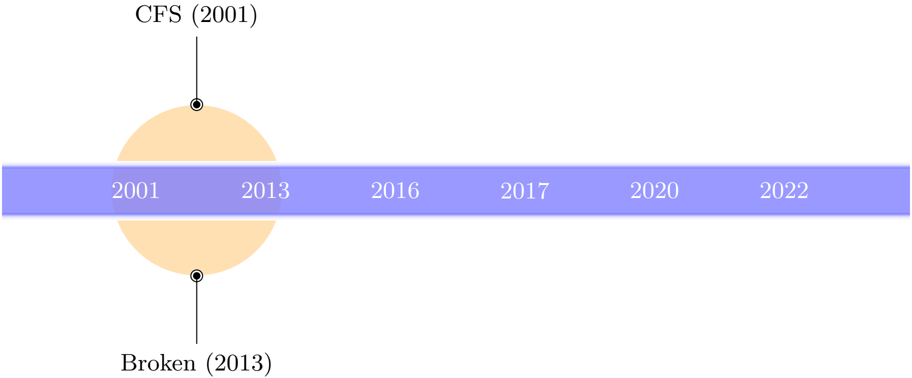
What will we see?

The Framework
The Scheme
Security
Future Directions

The framework

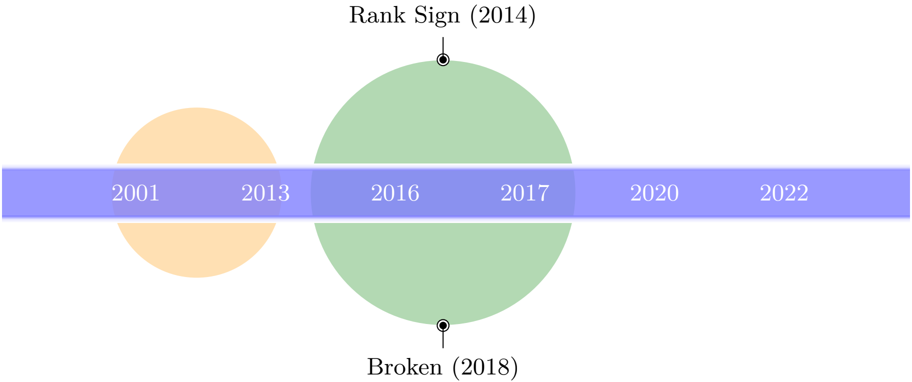
2001	2013	2016	2017	2020	2022

The framework



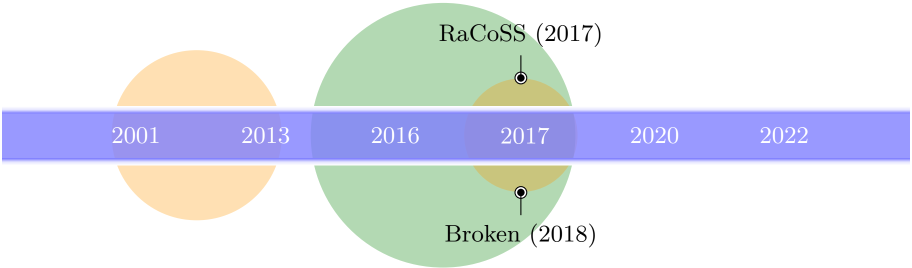
- Idea: use high rated goppa codes (a non-negligible fraction of the syndromes can be decoded to the nearest codeword);
- Assumption: indistinguishability;
- Problem: there exists a distinguisher.

The framework



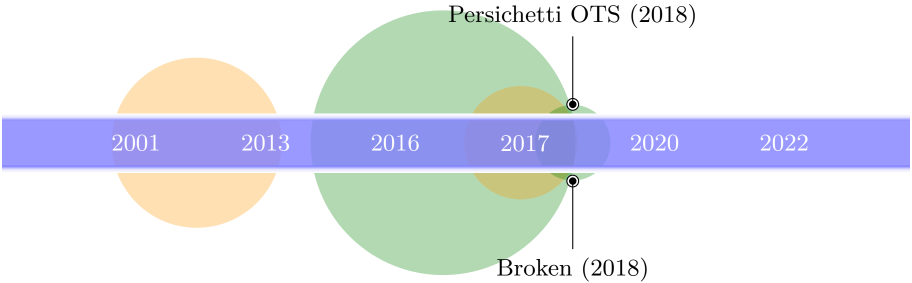
- Idea: change metric (work with the rank metric setting);
- Assumption: indistinguishability;
- Problem: structural attack.

The framework



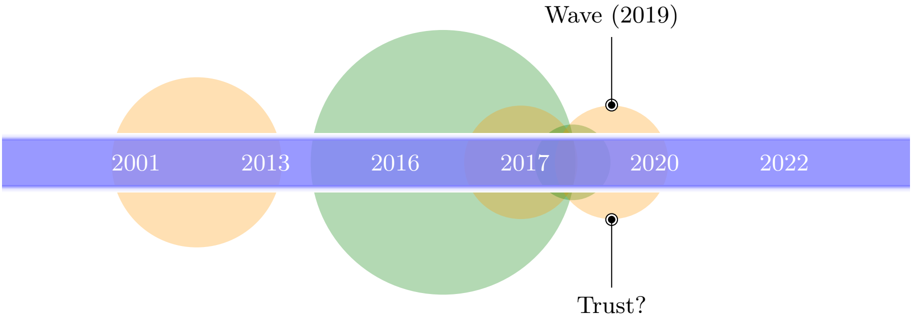
- In a nutshell: submitted to NIST PQC, attacked two days after, patched and then attacked again in 2018;
- Problem: the weight of a valid signature is large, and it intersects with the easy range of the SDP.

The framework



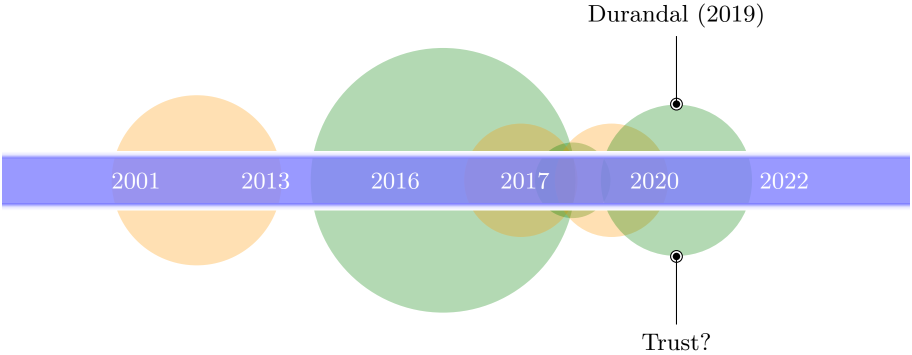
- Idea: exploit QC-codes;
- Problem: private key exposed through a LDPC matrix.

The framework



- Idea: hash&sign framework + use structured codes;
- Assumption: indistinguishability;

The framework



- Idea: change metric (work with the rank metric setting);
- Assumption: totally new problem ($PSSI^+$).

This task seems a hard goal

We tried to come up with a new proposal

The main idea

Some PKE/KEM from NIST PQC
(LEDACrypt, BIKE, HQC)

$\downarrow$

The result

- Post Quantum code-based digital signature;
- QC-codes (for compact key-size);
- LDPC-codes (for good performance).

Setup Phase

We don't really care about it for now...

Describe each parameter as soon as it is involved in the scheme.

$\downarrow$

We will denote in blue the parameters coming from the setup phase;

KeyGen Phase

Randomly generate the following elements:

- $x, y \in R := \mathbb{F}_2[X]/(X^n - 1)$, with $\mathrm{w}(x) = \mathrm{w}(y) = w$;
- $p, q \in R$, with $\mathrm{w}(p) = \mathrm{w}(q) = w_{pq}$.

Define the polynomials:

- $h := pq^{-1}$;
- $s := x + hy$.

With this notation the private and public keys are given by

$$\begin{cases} sk = (x, y, p, q) \\ pk = (h, s) \end{cases}.$$

Sufficient conditions for q
(to be invertible)

Known in literature

If we take

- n prime;
- 2 is a primitive root modulo n;
- w_{pq} odd.

then it works.

Observation (Why?)

No deails:

- Same idea behind BIKE and LEDACrypt;
- interesting;
- not our focus now.

Signature Phase

Take as input a message m to be signed and the secret key sk. Generate:

$$r := \mathcal{H}_{w_r}(m \,||\, pk \,||\, \texttt{nonce})$$

$$t \in R \text{ such that } \mathrm{w}(t) \in I_t$$

$$\begin{cases} \alpha := qt + ry \text{ and } \mathrm{w}(\alpha) \in I \\ \beta := \alpha h + sr \text{ and } \mathrm{w}(\beta) \in I \end{cases}$$

With this notation the signature is given by

$$(\alpha, \texttt{nonce}).$$

How to contruct I ?

A genuine signer must be able to sign efficiently

$$\alpha = qt + ry$$

Observation (from HQC...)

- q, t of given weights;
- $z := qt$.

Then z is distributed as a binomial r.v. of *known* parameter $\tilde{p}$.

$$\downarrow$$

The public parameters determine the probability distribution of α.

$$\downarrow$$

We can find an interval I such that, if the scheme is executed honestly, the failure probability (for α) is negligible.

Why $\mathrm{w}(\beta) \in I$?

$$\begin{aligned}\beta &= h\alpha + sr \\ &= h(qt + ry) + (x + hy)r \\ &= hqt + hry + xr + hry \\ &= pt + xr\end{aligned}$$

$$\downarrow$$

Same distribution as α:

$$\begin{cases}\alpha = qt + ry \\ \beta = pt + rx\end{cases}$$

Conclusion

A genuine signer is able to generate a pair $(\alpha, \texttt{nonce})$ such that $\mathrm{w}(\alpha), \mathrm{w}(\beta) \in I$.

How to do this in practise?

How do we choose parameters?

One possible way:

Variable Name	Weight Name	Size
p, q	w_{pq}	$\approx \sqrt{n}$
t	I_t	$\left[\lceil \frac{\sqrt{n}}{2} \rceil, \ldots, \lceil \frac{2\sqrt{n}}{3} \rceil\right]$
r	w_r	$\approx 3 \log n$
x, y	w	$\approx \sqrt{n}$

Example

If we take the parameters' choice for 128-security bits

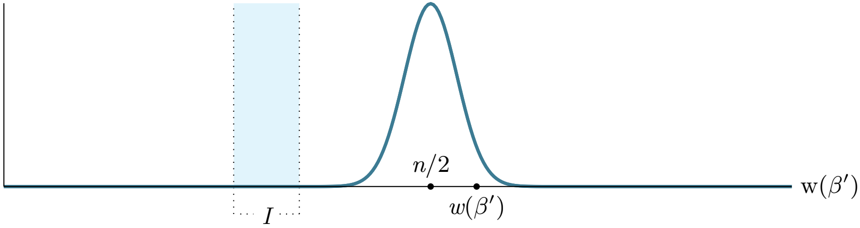
n	w_{pq}	w	I_t	Range of p_α, p_β	I
14627	121	121	$[61, 81]$	$[0.39102, 0.42220]$	$[5366, 6535]$

then the probability that the weight of α or β results outside the range I is $\approx 10^{-10}$.

What's the idea behind?

$n = 1427, w_r = 22, w_{pq} = 121, w = 121, I_t = [61, 81], I = [5366, 6535]$

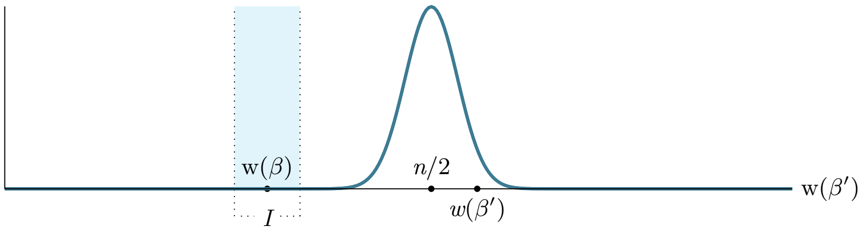
Let's just guess (α', `nonce'`) and compute $\beta' := \alpha' h + s r'$.



What's the idea behind?

$n = 1427, w_r = 22, w_{pq} = 121, w = 121, I_t = [61, 81], I = [5366, 6535]$

Choose α honestly.



A genuine signer is the only one who can *efficiently* produce a signature.

Verification Phase

Take as input the signed message $(m, (\alpha, \texttt{nonce}))$.
Compute $r := \mathcal{H}_{w_r}(m \,||\, pk \,||\, \texttt{nonce})$ and $\beta := h \cdot \alpha + s \cdot r$ and check that

$$\begin{cases} \mathrm{w}(\alpha) \in I \\ \mathrm{w}(\beta) \in I \end{cases} .$$

If these conditions are satisfied the verifier accepts the signature, otherwise it rejects.

Security

some considerations about the hardness of:

1. Attacks using the public key;
2. Attacks using a valid signature;
3. Attacks using multiple signatures.

Before doing that...

Well known:

$$\begin{array}{ccc} \mathbb{F}_2[X]/(X^n-1) & \longleftrightarrow & (\mathbb{F}_2)^n \\ a := a_0 + a_1 X + \ldots + a_{n-1}X^{n-1} & \longleftrightarrow & (a_0, a_1, \ldots, a_{n-1})^\top =: \bar{p} \end{array}$$

We can express the product $a \cdot b$ as

$$a \cdot b = \underbrace{\begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}}_{\mathrm{circ}(a)} \cdot \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

What does this representation allows?

We can relate our scheme to some lattice and coding problems.

(1) - Attacks using the public key

We discuss the hardness of

(1.a) - Recovering (p, q);
(1.b) - Recovering (x, y);
(1.c) - Forging a signature.

(1.a) - Hardness of recovering (p, q)

Public key: (h, s)
where $h = pq^{-1}$

$\downarrow$

Observation (from NTRU)

If we take $w_{pq} \approx 3\ln(n)$, then
$(q, p) = (q_0, q_1, \ldots, q_{n-1}, p_0, p_1, \ldots, p_{n-1})$
is very likely the shortest vector of the lattice

$$\mathcal{L}_h := \left\{ X \cdot M_h \mid X \in \mathbb{F}_2^{2n} \right\}, \text{ where}$$

$$M_h := \begin{pmatrix} I_n & \mathrm{circ}(h) \\ 0 & 2I_n \end{pmatrix}$$

Seems difficult to retrieve (p, q).

To be honest...

There is a better way to try to recover p, q from h.

- If we define $G := \left[\mathrm{circ}(h)^\top \mid \mathbb{1}\right]$, then $qG = (p \mid q)$;
- This means that $(p \mid q)$ belongs to the code generated by G, and so it is a solution of the CFP with parameters (H, w_{pq}, w_{pq}), where $H = \left[\mathbb{1} \mid \mathrm{circ}(h)\right]$.

Observation

We would have solved an instance of CPF.

Observation

The Codeword Finding Problem is known to be difficult.

To be more precise

these are particular instances of CFP

Observation

CFP is NP-complete in the general case (random matrices), but in our case the matrix has a particular structure, given by

$$\left[\mathbb{1} \mid \mathrm{circ}(h)\right].$$

In this case the problem is known as 2-QCCFP, and it is believed that there are no weaknesses linked to this particular structure.

$$\downarrow$$

Seems difficult to retrieve (p, q).

(1.b) - Hardness of recovering (x, y)

Public key: (h, s)
where: $s = x + hy$ and $\mathrm{w}(x), \mathrm{w}(y) = w$

Said otherwise...

$$\begin{cases} \bar{s} = \left[\mathbb{1} \mid \mathrm{circ}(h)\right] \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \\ \mathrm{w}(x), \mathrm{w}(y) = w \end{cases}$$

Observation (As before)

We would have solved an instance of SDP (in particular, 2-QCSDP), which is considered difficult.

$\rightarrow$ It seems difficult to retrieve (x, y).

(1.c) - Hardness of forging a signature

An adversary has to create a pair $(\alpha, \texttt{nonce})$ such that:

$$\begin{cases} \mathrm{w}(\alpha h + sr) \in I \\ \mathrm{w}(\alpha) \in I \end{cases}$$

Observation

If we were able to forge a single message, we would be able to solve a particular instance of SDP.

Indeed, if we fix a nonce

$$\begin{cases} \mathrm{w}(\alpha h + sr) \leq t_1 \\ \mathrm{w}(\alpha) \leq t_2 \end{cases} \iff \begin{cases} \beta := \alpha h + sr \\ \mathrm{w}(\beta) \leq t_1 \\ \mathrm{w}(\alpha) \leq t_2 \end{cases} \implies \begin{cases} sr = \left(\mathbb{1} \mid\mid \mathrm{circ}(h)\right) \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \\ \mathrm{w}(\beta \mid\mid \alpha) \leq t_1 + t_2 \end{cases} .$$

(2) - Attacks using a valid signature

Some considerations about the hardness of

Recovering the private key.

Key-Recovery using a valid signature

Suppose an attacker can access a valid signature $(\alpha, \texttt{nonce})$ of a message m.

Information available to the attacker

- $m, \alpha, \texttt{nonce}, r = \mathcal{H}_{w_r}(m \mid\mid pk \mid\mid \texttt{nonce})$;
- $\alpha = qt + ry = \underbrace{\left[\mathbb{1} \mid \operatorname{circ}(r)\right]}_{H} \begin{pmatrix} \bar{q}t \\ \bar{y} \end{pmatrix}$.

H is a low density matrix !
(efficient algorithms for decoding)

What's the problem?

If an attacker finds (qt, y) it could be able
to perform a forgery.

Observation (indeed...)

- Suppose (qt, y) is known to the attacker
- In order to forge a signature, an adversary could:
  - compute r';
  - compute $\alpha' = qt + r'y$;
  - compute $\beta' = \alpha' h + sr'$;
  - check the verifying condition (very likely to be satisfied).

How to avoid this attack?

With our parameter's choice, there are just too many solutions to

$$\alpha = \left[\mathbb{1} \mid \mathrm{circ}(r)\right] \begin{pmatrix} \bar{q}t \\ \bar{y} \end{pmatrix}$$

to hope to find the real one.

Just to be more precise

- We can estimate the weight distribution of (qt, y);
- We can compute a value $(w_{\max})$ for which $\mathbb{P}(\mathrm{w}((qt, y) > w_{\max}))$ is negligible;
- We can count how many elements have sydrome α and weight $\leq w_{\max}$:

$$\frac{1}{2^n} \cdot \sum_{i=0}^{w_{\max}} \binom{2n}{i}.$$

Example

Instantiation of the previous problem for a fixed parameter's choice.

n	w_{pq}	$\mathrm{w}(t)$	w	p	$w_{\max}$	# solutions
14627	121	61	121	0.3201	5013	$\approx 10^{1414}$

(3) - Attacks using multiple signatures

We discuss the hardness of

Recovering the private key.

Key-Recovery using multiple signatures

Suppose an attacker can access a list $\{(\alpha_i, \texttt{nonce}_i)\}_{i=1}^{l}$ of signatures.

Information available to the attacker

- $\{m_i, (\alpha_i, \texttt{nonce}_i), r_i\}_{i=1}^{l}$;
- $$\begin{cases} \alpha_1 = qt_1 + r_1 y \\ \alpha_2 = qt_2 + r_2 y \\ \vdots \\ \alpha_l = qt_l + r_l y \end{cases} \iff \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_l \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbb{1}_n & 0_n & \cdots & 0_n & \mathrm{circ}(r_1) \\ 0_n & \mathbb{1}_n & \cdots & 0_n & \mathrm{circ}(r_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_n & 0_n & \cdots & \mathbb{1}_n & \mathrm{circ}(r_l) \end{pmatrix}}_{H} \cdot \begin{pmatrix} qt_1 \\ qt_2 \\ \vdots \\ y \end{pmatrix}.$$

As before

H is a low density matrix !

- Efficient algorithms for decoding;
- If an attacker finds the real solution it could be able to perform a forgery.

The problem now is even worse...

Before

Number of possible solutions:

$$\frac{1}{2^n} \cdot \sum_{i=0}^{w_{\max}} \binom{2n}{i}.$$

Now

Number of possible solutions:

$$\frac{1}{2^{nl}} \cdot \sum_{i=0}^{w_{\max}} \binom{n(l+1)}{i}.$$

- If l increases, the whole expression decreases;
- If l increases, we expect to have a unique solution.

We could run xBF and find the solution.

Conclusions

We can't use the scheme to sign *multiple* times.

We are confident it is a good OTS.

Future directions

Security proof for OTS
(1-EUF-CMA)
Derivation from IP

Thanks