Post-Quantum Cryptography

Johan Håstad

KTH Teknikvetenskap

CDIS Conference, May 24, 2022

Ask questions

I am just giving selected facts and opinions.

Do feel free to ask questions.

People involved




- Johan Håstad, project leader.
- Martin Ekerå, industrial PhD student from MUST.
- Joel Gärtner, CDIS PhD student.
- Douglas Wikström, assistant supervisor.

Basic question

Which systems for public key cryptography should we use?

- For systems that should be secure for 50 years.
- For systems that should not be broken in a year.
- For systems that should not be broken in real time.

All questions are changed by the possible construction of quantum computer.

Quantum computers

Computers taking advantage of quantum physics.

The computer can be in a superposition of exponentially many states.

You need these exponentially many computations to have positive interference. Very far from general parallelism.

Photo of Bristlecone

Photo removed for copyright reasons.

Shor's result

Integer factorization and discrete logarithms (also based on elliptic curves) are easy.

One exponentiation of an N digit number to an N digit exponent modulo an N digit number.

About the cost of a primality test. But of course on a quantum computer.

When, if ever, will quantum really factor?

Very hard question.

Billions spent on constructing such computers and reports of progress but an unclear road towards general quantum computers.

When, if ever, will quantum really factor?

Very hard question.

Billions spent on constructing such computers and reports of progress but an unclear road towards general quantum computers.

Martin Ekerå made several improvements to both design and analysis of Shor's algorithms.

When, if ever, will quantum really factor?

Very hard question.

Billions spent on constructing such computers and reports of progress but an unclear road towards general quantum computers.

Martin Ekerå made several improvements to both design and analysis of Shor's algorithms.

We need to be prepared and I think we should work with the fear (hope?) that we have functional quantum computers within 10-20 years.

Looking 50 years in to the future

The original RSA paper was published in 1977, 45 years ago. From their paper.

> *An 80-digit N provides moderate security against an attack using current technology; using 200 digits provides a margin of safety against future developments.*

Here N is a number to be factored.

Some factorization records

As reported in public.

- 100 digits, 1988.
- 155 digits, 1999.
- 200 digits, 2005.
- 250 digits, 2020.

Done by academia. All using multiple machines for “months”.

It is likely that the world record holder is NSA and they do not publish their results. One target size is 1024 bits which is 309 digits.

My conclusion

RSA authors were slightly wrong but not terribly so.

- Computers did get faster, but this was accounted for.
- The algorithm “Number field sieve” was not discovered in 1977, and this made the predictions wrong.
- Nobody was thinking “quantum algorithms” in 1977 and this might kill it completely.

My own recollection

At some point RSA had a research center in Sweden and I remember discussing what parameters to be used.

I remember (must have been early 1990ies) that I felt that 1024 bits was going to be safe for a long time, probably my life time.

The state of RSA

Right now I think.

- 1024 bits are fine for private secrets.
- 2048 bits works for minor bank transactions.
- 4096 bits is OK for national security, but not in the 50 year perspective.

Similar for discrete logarithm based and shorter keys for elliptic curves.

Post-Quantum

We need to find new basic algorithms. Sources

- The NIST competition. Will probably make a third round selection soon.
- Our own understanding.

“Vem i hela världen kan man lita på”.

The favorite basic hard problems

Algorithms relying on hardness of lattice problems.

Algorithms relying on problems related to error correcting codes.

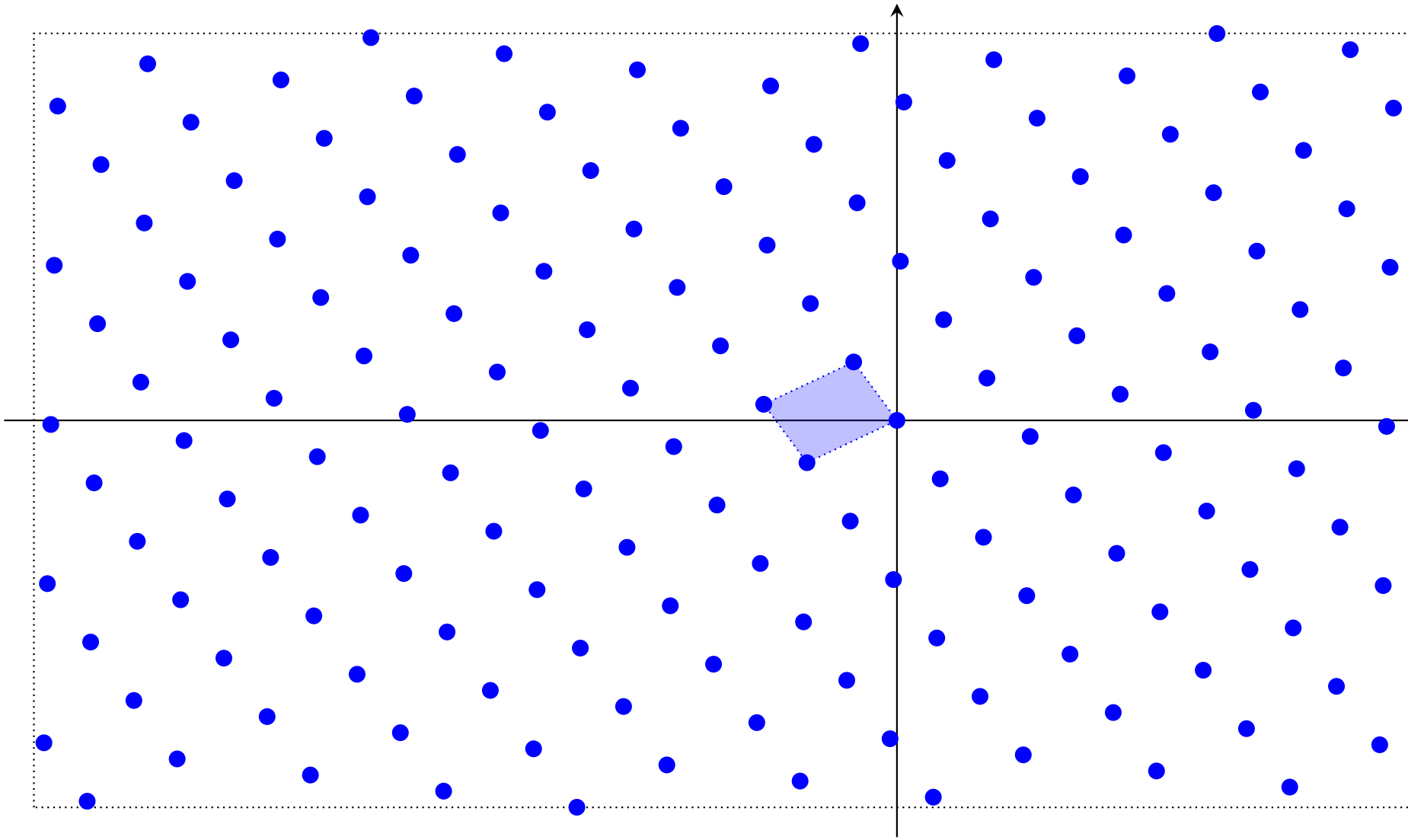
The most popular starting point, lattices

Integer lattices

$$L = \sum_{i=1}^{n} a_i \vec{b}_i$$

where $a_i \in \mathbb{Z}$ and $\vec{b}_i$ are linearly independent vectors in $\mathbb{R}^n$.

A lattice L



The basic computational problem, SVP

Given L by a basis $(\vec{b}_i)_{i=1}^n$ find the shortest non-zero vector in L.
Shortest Vector Problem.

Very easy in two dimensions but think 200-10000 dimensions.

Complexity of SVP, in broad terms

- (Almost) NP-hard to approximate within $2^{(\log n)^{1-\epsilon}}$.
- Polynomial time to approximate within c^n for any $c > 1$.
- Can be solved in time $2^{O(n)}$ exactly.
- There is a reduction between worst and average case.

The cryptographically interesting cases are neither easy nor known to be hard.

Algorithms for SVP

Needs to be investigated.

- What can be done on a classical computer?
- Algorithms for quantum computers.

We decided against implementing an algorithm. We have been following developments, which have been significant but not revolutionary.

Some time has been and will be devoted to thinking about efficient quantum algorithms. High risk, high gain.

Security of Lattice based crypto

For many crypto systems there is a natural lattice problem to solve to break the system.

A minimal requirement is to make sure that this is not feasible.

Key question: Are other attacks possible?

Two possible situations

- A more efficient algorithm for a basic lattice problem implies a more efficient attack on the crypto system.
- Any attack on the crypto system implies a more efficient algorithm for a basic lattice problem.

Typically we are here thinking SVP discussed above.

Main result so far

We (i.e Joel Gärtner) propose a (very slight) variant of one of the NIST candidates.

We prove that an efficient attack implies an algorithm for SVP assumed not to exist.

I would call this “High pain, solid gain” research.

Practical?

To get this (closer to) provable hardness we need rather large parameter cryptosystem.

Possible to use when speed is not essential.

Using better hardware

A tendency to put priority on getting minimal parameters for key sizes. Always top speed.

Why not use the increased speed to get increased security margins?

The eternal questions

That always are with us.

- Will there be a “Number field sieve” for lattices? A significant discovery causing a quicker increase of parameters.
- Will there be a “Quantum computer” of lattices? A complete change due to something really new.

That hardware is getting faster is no surprise and can be accounted for.

The End

Sometimes I worry about bugs in my proofs in old papers.

This is nothing compared to recommending a weak crypto to important applications.