The post-quantum world

Christelle Vincent

University of Vermont

March 1, 2019

Some basic facts

- Modern public-key cryptography is based on problems that **we believe** are time-consuming to solve, **not** impossible to solve.
- Quantum computers, by using certain physical quantum phenomenon, will give **faster** algorithms to solve **some** of these problems.

Some basic facts

- "Post-quantum cryptography" refers to algorithms that we think will be secure in a post-quantum world.
  In other words, the problems they rely on are believed to be time-consuming to solve on a classical **and** a quantum computer.

- In this paradigm, only the attacker needs a quantum computer to deploy a quantum attack. The encryption/decryption still takes place on a classical computer.

Organization of this lecture

1. NIST standards for cryptography
2. How Shor's algorithm changes the landscape
3. What now?

1. NIST standards for cryptography

How Shor's algorithm changes the landscape

What now?

Security levels

The **security level** of a cryptographic algorithm measures how "strong" the algorithm is.

We say that an algorithm *provides n-bits of security* if an attacker needs to perform 2^n operations to break it.

Security levels

Because of this, the security level of an algorithm depends on the current best attack on the algorithm; it is not fixed by the algorithm itself.

The Advanced Encryption Standard

The **Advanced Encryption Standard** (AES) is a symmetric-key algorithm that is believed to be particularly robust.

For example it is the only publicly accessible cipher approved by the NSA to encrypt top secret information.

The security of AES with an n-bit key is n-bits, and so often the "AES equivalent" is given as the security level of an algorithm.

Current security levels

Currently, NIST defines

- Level 1 security to provide 128-bits of security
- Level 3 security to provide 192-bits of security
- Level 5 security to provide 256-bits of security

(The even levels are for the security of hash functions.)

Current security levels

For the following algorithms, NIST's security recommendations are:

Level	AES	RSA	DLP	ECDLP
1	128	3072	3072	256
3	192	7680	7680	384
5	256	15360	15360	512

where

- for RSA this is the size of N in bits
- for DLP this is the size of p in bits (working over $\mathbb{Z}/p\mathbb{Z}$)
- for ECLP this is also the size of p in bits (using an elliptic curve defined over $\mathbb{F}_p$)

NIST standards for cryptography

1. How Shor's algorithm changes the landscape

What now?

Significance of Shor's algorithm

Assuming that quantum computers can be built at scale, Shor's algorithm allows us to solve some problems a lot faster.

This improves considerably the best-known attacks on certain algorithms, which in turns reduces their security level.

Algorithms affected

We saw that essentially, the quantum step of Shor's algorithm is a *period-finding algorithm*.

We saw explicitly how this allows one to factor a number N, which should affect the security level of **RSA**.

(We did this by finding the period of the function $f(x) = a^x \pmod{N}$, which is the multiplicative order r of a modulo N.)

Algorithms affected

It turns out that DLP and ECDLP can also be broken if we have a fast period-finding algorithm.

Suppose that g is a primitive root of p and $h = g^a$ for some unknown a.

Define $f(x, y) = g^x h^{-y} = g^{x-ay}$.

Then

$$f(x_1, y_1) = f(x_2, y_2)$$

if and only if

$$(x_2, y_2) = (x_1, y_1) + \lambda(a, 1).$$

So the period of f is $(a, 1)$ and if we can find the period of f we can find a, the secret key.

Algorithms affected

RSA, DLP and ECDLP are essentially the only public-key algorithms used currently.

So Shor's algorithm provides a polynomial-time attack on essentially all of the cryptography we use.

How bad is this? pqRSA

As a joke/thought experiment, Bernstein, Heninger, Lou and Valenta have prepared the article "Post-quantum RSA."

The goal of the article was to determine whether the parameters of RSA could be adjusted so that known quantum attacks were infeasible while encryption and decryption remained feasible.

pqRSA results

It turns out that pqRSA is not secure under the usual security definition, which requires security against polynomial-time adversaries.

However, by being very careful, the authors manage to give quadratic security against post-quantum attackers. (This is comparable to some original public-key cryptosystems.)

pqRSA costs

The cost, however, is prohibitive: \$1 of computer time per encryption or decryption.

This is because, to provide 100 bits of security, pqRSA requires a modulus N of size 1 TB!

This N is obtained by multiplying 2^{31} distinct 4096-bit primes.

At this scale, encryption takes about 10 hours and decryption had not been successfully done as of publication time.

“Conclusion”

RSA has enough flexibility to survive the advent of quantum computers – beaten, bruised, and limping, perhaps, but not dead.

NIST standards for cryptography

How Shor's algorithm changes the landscape

1. What now?

Next steps: New algorithms

In light of how impractical current algorithms would need to become to remain not even that secure, NIST has begun the process of searching for and standardizing new cryptographic systems that would be practical in the quantum paradigm.

NIST PQC

The algorithms that made it to round 2 were announced on January 30, 2019.

There are 17 public-key encryption/key encapsulation algorithms now under review. (pqRSA didn't make it.)

The rest of the semester

Our goal for the rest of the semester will be to study as many of these 17 algorithms as we can, in as much depth as we can.

The rest of the semester

The algorithms belong roughly to three "families" of algorithms:

- Code-based algorithms
- Lattice-based algorithms
- One elliptic curves-based algorithm

Signature schemes

The competition also has some signature schemes, some of which are part of the code-based and lattice-based families.

There is also one hash-based signature, a few multivariate signature algorithms, and one signature algorithm that does not belong to any family.

We probably will get to those too, especially the multivariate algorithms.

Thank you!