Standards for Post-Quantum Cryptography

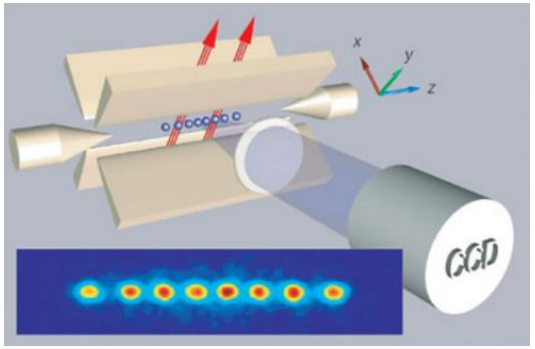
Yi-Kai Liu / NIST PQC team

http://indianajones.wikia.com/wiki/Raiders_of_the_Lost_Ark

Quantum Computers

- Quantum mechanics
  - Behavior of small objects: atoms, electrons, photons
  - Quantum superpositions: $|\psi_{cat}\rangle = |alive\rangle + |dead\rangle$, $|\psi_{qubit}\rangle = |0\rangle + |1\rangle$
  - Interference: combine $|0\rangle + |1\rangle$ with $|0\rangle - |1\rangle$, get $|0\rangle$

  - When an object is observed, the quantum superposition collapses
  - This is why large objects do not behave quantumly
  - Major challenge in building a quantum computer



R.Blatt & D. Wineland, Nature 453, 1008-1015 (19 June 2008)

Quantum Computers

- Potentially much more powerful than classical computers
  - Conjecture: A classical computer needs **exponential time** to simulate a quantum computer (in the general case)
- Exponential speedups for some interesting problems
  - Simulating the dynamics of molecules, superconductors, photosynthesis…?
  - **Factoring large integers (Shor's algorithm)**
  - **Discrete logarithms in any abelian group (Shor's algorithm)**
- And some polynomial speedups
  - **Unstructured search (Grover's alg.), collision finding**

Who Cares?

- Quantum computers would break most of our public-key crypto
  - RSA, Diffie-Hellman key exchange, elliptic curve crypto
  - TLS, digital certificates, IPSec
- Symmetric crypto would be affected, but not broken
  - “Keep using AES, but double the key length”
  - (Actually, it’s more complicated than that)

Who Cares?

- Fortunately, large quantum computers don't exist yet
  - Small ones do exist, but can they scale up?
  - Michele Mosca (http://eprint.iacr.org/2015/1075): "1/2 chance of breaking RSA-2048 by 2031"

- Unfortunately, 2031 is not that far away
  - How long does today's data need to remain secure? 5-10 years?
  - How long does it take to deploy new crypto software? 5-10 years?

Post-Quantum Cryptography

Cryptosystems	Hard problem	Trapdoor
Lattice-based	Finding short vectors in a high-dimensional lattice	Nice basis for the lattice (short, almost-orthogonal vectors)
Code-based	Decoding a random binary linear code	Linear trans-formations that reveal structure of the code
Multivariate	Solving a random system of multivariate quadratic equations over a finite field	Linear trans-formations that reveal structure of the equations

Post-Quantum Cryptography

- **Hash-based signatures**
  - Simple: uses only a hash function, doesn't need a trapdoor
  - Caveat: signing algorithm has to update an internal data structure every time it signs a message
- **Isogenies of supersingular elliptic curves**
  - Useful for key exchange?
- **Quantum key distribution**
  - Information-theoretic security
  - Requires optical fiber, distance limited to ~200 km

Post-Quantum Cryptography

- **How do we know a cryptosystem is secure?**
  - Cryptanalysis: what are the best known attacks?
  - Security proofs: based on some hardness assumption?

- **How well do these cryptosystems work in practice?**
  - Size of keys, time needed for each operation
  - Ease of implementation, how to set the parameters
  - Does it fit nicely with TLS, other higher-level protocols?
  - Vulnerabilities to side channel attacks?

- **There's a conference about this:**

Lattice-Based Cryptography

Lattice-Based Encryption Schemes

- **NTRUEncrypt**
  - Developed circa 1996 by Hofstein, Pipher and Silverman, commercially available
- **Regev's encryption scheme**
  - Based on LWE problem ("learning with errors") (2005)
    - Solving a noisy system of linear equations modulo p
  - Theoretical security guarantees
    - Solving average-case instances of LWE is at least as hard as solving worst-case instances of SIVP ("lattice short independent vectors problem")
  - When instantiated with ideal lattices, this looks sort of like NTRUEncrypt
    - Ideal lattice: an ideal in a ring, for example, $Z[X] / (X^n+1)$
    - This gives smaller key sizes, without compromising security?

LWE Problem ("learning with errors")

- Secret s in $(Z_q)^n$
  - q = poly(n)
- Given samples (a,b) in $(Z_q)^n \times Z_q$
  - a is uniformly random
  - $b = a^T s + e$, where e is Gaussian distributed, w/ std dev q/poly(n)
- Can we determine s?
  - "Decoding a random linear code over Z_q"

- **Claim: samples (a,b) look pseudorandom!**

Regev's Encryption Scheme

- Private key: s in $(Z_q)^n$
- Public key: LWE samples (a_i, b_i) in $(Z_q)^n \times Z_q$ (for i = 1,…,m)
  - Where we let m ~ n log n
  - Recall $b_i = a_i^T s + e_i$

- Encryption: Given a single bit x in {0,1}
  - Choose a random subset S of {1,…,m}
  - Output $a = \Sigma_{i \text{ in } S} a_i$ and $b = (0.5)(q-1)x + \Sigma_{i \text{ in } S} b_i$

- Decryption: Given (a,b)
  - Compute $b - a^T s = (0.5)(q-1)x + \Sigma_{i \text{ in } S} e_i$
  - Round this to either 0 or (0.5)(q-1), mod q
  - Output either x = 0 or x = 1, accordingly

Lattice-Based Signatures

- **"Hash-then-sign" approach (GGH '97)**
- Lattice L
- Public key: A "hard" basis B
- Private key: A "good" basis T (the "trapdoor")

- Signing: Given message m,
  - Hash it to a point x in R^n
  - Find the lattice vector v in L that lies closest to x
  - Output (x,v)

- Verification: Given (m,x,v),
  - Check that m hashes to x, v is in L, and v is close to x

Lattice-Based Signatures

- **NTRUSign**
  - Developed circa 2003
  - Broken by Nguyen and Regev in 2006 ("learning a parallelipiped")
    – each signature leaks some information about the secret key
  - Patched by adding "perturbations" to the signatures

- **GPV signatures**
  - Uses "Gaussian sampling" (Gentry, Peikert, Vaikuntanathan, 2007)
    - Provably secure variant of NTRUSign, but less efficient
    - Based on SIS problem ("small integer solutions") –
      random subset sum with vectors modulo p
    - Has worst-case to average-case reduction from lattice problems

Lattice-Based Signatures

- **Signatures using Fiat-Shamir heuristic**
  - More efficient than GPV approach
  - Provably secure based on hardness of SIS problem, in random oracle model
  - Lyubashevsky (2011), and several follow-on works…

Cryptanalysis

- Lattice basis reduction (in polynomial time)
  - Try to find a basis consisting of short, nearly-orthogonal vectors
  - LLL algorithm: finds a $2^{O(n)}$-approximation to the shortest vector in the lattice
  - Block-KZ reduction, follow-on work by Schnorr, Nguyen…
- Sieving, enumeration (in exponential time)
  - Find the shortest vector in the lattice
  - Extreme pruning (Gama, Nguyen, Regev, 2010)
- Algorithms for LWE and SIS problems
  - List merging (Lyubashevsky, 2004)
  - Linearization (Arora, Ge, 2011)

Quantum Cryptanalysis?

- Quantum algorithms for problems in number fields
  - Unit group, class group, principal ideal problem
  - Running time is polynomial in the degree
  - (Eisentrager, Hallgren, Kitaev, Song, 2014; Biasse, Song, 2016)
- Quantum attack on the Soliloquy cryptosystem
  - (Campbell, Groves, Shepherd, 2014)
    - Commentary: http://web.eecs.umich.edu/~cpeikert/soliloquy.html
- Quantum speed-ups of classical lattice algorithms
  - (Laarhoven, Mosca, van de Pol, 2013)

Issues and Open Questions

- Are ideal lattices just as hard as general lattices?
  - Clearly there is some additional structure there…
  - In the security proofs, we assume these problems are hard
- How hard are the LWE and SIS problems, for the parameters we use in practice?
  - Parameters are chosen based on experimental cryptanalysis
  - Worst-case to average-case reduction doesn't say anything meaningful in this regime
- How to implement Gaussian samplers?
  - Need good entropy, how to test this, what about discretization errors, need constant-time implementations to resist side-channel attacks…