GT Crypto Reading Group

Post Quantum Cryptography

Name: Abrahim Ladha

1.1 Physics Notation

We describe a quantum state as a vector of the form $|\psi\rangle$. In a complex valued vector space. For any number of vectors in this state, we may superposition them as a linear combination of complex coefficients. Given some basis of our vector space, we may write any $|\psi\rangle = \sum_{i=1}^{n} c_i |e_i\rangle$. The dimension of our vector space in general is infinite (making it a Hillbert space). Any quantum state can be described by some ket in our Hillbert space.

For each $|\psi\rangle \in V$, there exists $\langle\psi| \in V^*$[1] such that $\langle\psi| = |\psi\rangle^\dagger$.[2]

The inner product is written naturally as $\langle\phi|\psi\rangle = |\phi\rangle^\dagger |\psi\rangle$. the sum of the pairwise components, and is a complex number.

There is also a less intuitive outer product defined as a matrix $|\phi\rangle \langle\psi|$. This is an operator less naturally. Suppose $A = |\phi\rangle \langle\psi|$, then $A|x\rangle = |\phi\rangle \langle\psi|x\rangle = c|\phi\rangle$. While The inner product is a complex value, since the outer product with a vector is another vector, it must be an operator.

1.2 Quantum Computer

1.2.1 Qubits

A *classical computer* is simply a Turing machine, or some model of equal power. There is some infinite tape where each cell may contain a 0 or 1. Instead of bits, a *quantum computer* has qubits, which can be a 0, 1 or any *superposition* of 0 or 1. More formally, instead of 0 or 1, we have a zero or one vector, denoted as $|0\rangle$ and $|1\rangle$. Our superposition is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. If you ever try to *measure* a qubit, you will only see that you will get $|0\rangle$ or $|1\rangle$. If you try to measure it, you will only receive $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$.[3] Given this construction, we can then represent a qubit in the form

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \tag{1.1}$$

Reparametrizing our qubit $|\psi\rangle$ in terms of θ, φ as angles, we get the analogy of a qubit being in a state as

[1] spoken as: "bra in the dual space"

[2] The † operator is the hermitian conjugate. It is the complex conjugate, and the transpose. The complex conjuage just replaces i with $-i$ component wise. If $|\psi\rangle$ is a column vector, then its transpose, $|\psi\rangle$ is a row vector

[3] You might think we have four degrees of freedom here with α, β being complex numbers with two real parts each, but we only have three because the constraint $|\alpha|^2 + |\beta|^2 = 1$

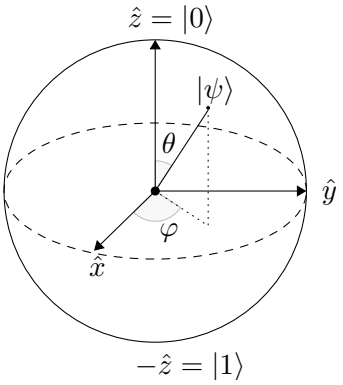

Figure 1.1: The Bloch sphere.

1.2.2 Multiple Qubits

We can have multi qubit systems.

1.2.3 Quantum Circuits

Hadamard gates, reversible, and diagramming.

1.2.4 No Cloning Theorem

It is impossible to copy or clone a qubit. Intuitively, you may think that the cloning operation requires the ability to take measurement. We can prove there is no arbitrary operations that can clone a qubit.

Assume to the contrary there is some unitary transform that can clone qubits. That is, given on input $|\psi\rangle |0\rangle$ it outputs $|\psi\rangle |\psi\rangle$. Written as operators, we have $|\psi\rangle |\psi\rangle = U |\psi\rangle |0\rangle$. Since this unitary transform works for all qubits, we have $|\varphi\rangle |\varphi\rangle = U |\varphi\rangle |0\rangle$, for any other qubit $|\varphi\rangle$. Lets inner product these two statements:

$$\langle\psi| \langle\psi| U^\dagger U |\varphi\rangle |\varphi\rangle = \langle 0| \langle\psi|\varphi\rangle |0\rangle \tag{1.2}$$

$$\langle\psi|\varphi\rangle^2 = \langle\psi|\varphi\rangle \tag{1.3}$$

This implies that $\langle\psi|\varphi\rangle$ is 0 or 1, which is not true to the assumption that $|\psi\rangle, |\varphi\rangle$ were any arbitrary states.

1.3 Shors Algorithm

1.3.1 Period finding

The period of a function f is some r such that for all x $f(x+r) = f(x)$. How can this be used for integer factorization? If $f(x) = a^x \pmod N$ then

$$f(x+r) = f(x) \tag{1.4}$$

$$a^{x+r} \equiv a^x \pmod N \tag{1.5}$$

$$a^r \equiv 1 \pmod N \tag{1.6}$$

Then for $a \in \mathbb{Z}_N$, We know $r|N$.

1.3.2 The classical portion

Shor's algorithm has two parts. A quantum portion to find the period in polynomial time, and some classical processing. For now, suppose that the quantum period finding portion is a polynomial time oracle. Suppose that we wish to factor $N = pq$. We can do processing to eliminate smaller factors before hand to eliminate the chance that we run Shor's algorithm to retrieve a factor we could have done classically anyway. For any composite N, it must contain a factor less than $\sqrt{N}$ so for p, q equal bitlength.[4], this is our ideal worst case. The algorithm is as follows:

1. $a \xleftarrow{\$} \mathbb{Z}_N$
2. Check if a, N are co-prime[5], If they are not then halt and return a
3. Query our oracle for $r \leftarrow QC(a, N)$
4. if r is odd, restart with a different a
5. if $a^{r/2} \equiv -1 \pmod{N}$ restart with a different a
6. Return one of $\gcd(a^{r/2} - 1, N)$ or $\gcd(a^{r/2} + 1, N)$

Why is this correct? Given that $a^r \equiv 1 \pmod{N}$, then $(a^{r/2} - 1)(a^{r/2} + 1) \equiv a^r - 1 \equiv 0 \pmod{N}$.

1.3.3 Quantum Portion

Eigenvectors intuition.

1.4 Violated Hardness Assumptions

As demonstrated, we have a quantum algorithm for integer factorization in polynomial time. CITE provides a list of nearly a hundred cryptographic hardness assumptions.

1.4.1 RSA Example

The RSA problem is given $N, e, M^e \pmod{N}$, can you efficiently compute M. You can break the RSA problem by breaking discrete log, but you can shortcut and break the RSA cryptosystem by breaking integer factorization. Given $N = pq, M^e \pmod{N}$, you can factor N efficiently to compute $\phi(N) = (p-1)(q-1)$, knowing this, and $ed \equiv 1 \pmod{\phi(N)}$, you can obtain d which lets you solve for $M = (M^e)^d$.

1.4.2 Discrete Log

Recall the discrete log problem. Given $y = g^x \pmod{N}$, solve for $x = DLOG_g(y)$ The current state of the art classical algorithm for solving the DLP is the Pollig Hellman algorithm CITE and it runs in $O(\sqrt{N})$ in the worst case. Solving DLP implies you can factor integers in polynomial time, but the reverse remains an open question. Shor's algorithm can be modified to immediately break discrete log as follows:

[4]I can factor a 256 bit number into its two factors, both around 127 bits in four minutes on my laptop. If $N = pq$ and $p >> q$, then $p >> \sqrt{N}$ and $q << \sqrt{N}$, so when we are trying to find any factor, we will find q much faster. For PGP, the weakest choice of an RSA modulus was 512 bits.

[5]Two numbers are a, b co-prime if they share no common factors. This can be tested by checking if $\gcd(a, b) = 1$. This is computed via the euclidean algorithm, which runs in polynomial time. If a, N are not co-prime, then $\gcd(a, N)$ is an factor of N not equal to 1.

1. Given $y \equiv g^x \pmod{N}, g, N$, the task is to solve for x. Suppose N is prime[6] and g is a generator
2. Construct the bivariate, periodic function $f(x_1, x_2) = g^{x_1} y^{x_2}$
3. Obtain the pair $(r_1, r_2) \leftarrow QC(g, y, N)$ from our quantum oracle such that $f(x_1, x_2) = f(x_1 + r_1, x_2 + r_2)$
4. return $x = -\frac{r_1}{r_2} \pmod{N}$

For N composite, this can be extended without knowing the factorization of N, which some messier cases. Why is this correct? $g_1^x y_2^x = g^{x_1+r_1} y^{x_2+r_2} \iff g^{r_1} y^{r_2} \equiv 1 \pmod{N}$, but then $g^{r_1} y^{r_2} \equiv g^{r_1} g^{xr_2} \equiv g^{r_1+xr_2} \equiv 1 \pmod{N}$. Since g is a generator, then $r_1 + xr_2 \equiv 0 \pmod{N}$. Solving for x we get $x \equiv -\frac{r_1}{r_2} \pmod{N}$.

1.5 Grovers Algorithm

Wave my hands really hard, intuition from parallelism. Talk about current work in attacks on AES(6,681)

1.6 NIST Contest

1.6.1 Background

Quantum Computers are developing far faster than people expected, so NIST decided to start early on replacing cryptosystems with quantum resistant ones. AES, SHA3 standardizations took many many years. quantum resistant systems are much less studied. The quantum resistant cryptoschemes do not appear to just be drop in replacements for how the world currently uses assymmetric crypto.

1.6.2 Unviolated Assumptions

There were around 69 submissions into round 1. Now 26 have made it into round 2, with 17 cryptosystems for public-key encryption, and 9 for digital signatures. Of the 17 for PKE, 9 are based on lattice hardness problems, 5 on coding, 1 on isogeny, and 2 on rank hardness problems[7] For the 9 digital signature submissions, 3 are based on lattices, 4 on multivariate, and 2 are classified as other

1.6.2.1 Lattice Based Cryptosystems

A lattice is a subgroup of $\mathbb{R}^n$ that is isomorphic to the additive group $\mathbb{Z}^n$, while also spanning the vector space $\mathbb{R}^n$. Anything in the lattice is a linear combination of some chosen basis of $\mathbb{R}^n$ with integer coefficients. The Shortest Vector Problem (SVP) is as follows. Given L a lattice, its basis B, $\lambda(L)$ the shortest nonzero vector in the lattice, and some real $\gamma > 0$, output a vector $v \in L$ such that $||v|| \leq \gamma\lambda(B)$.

There is no known quantum algorithm to solve lattice problems, such as SVP efficiently. The LLL algorithm can approximate solutions in polynomial time. For a while, people thought that

[6] It works if N is not prime just as well, but requires the chinese remainder theorem, and repeated applications.

[7] Rank hardness problems are in a subset of coding problems. They are separated because of the significant difference in the kinds of cryptographic attacks they are subject to CITE NIST STATUS

lattice problems might be efficient generally. In CITE, Ajtai shows how to generate hard instances of the lattice problem. These are used in the Pailier cryptosystem.

1.6.2.2 Learning with Errors

In 2006, the first hardness conjectures from statistical problems in machine learning were used to design cryptosystems. They earned the Godel prize in 2018 for their work. Many of the NIST lattice submissions are based on either Ring learning with Errors (RLWE) or Module Learning with Errors (MLWE). These problems naturally arise a homomorphic cryptosystem, more so than how security of Paillier is reduced to shortest vector problems.

DETAILS ON IT

1.6.2.3 Code Based Cryptosystems

Encryption in codebased cryptosystems is usually done by having the plaintext be some uncoded word, the ciphertext some pertubation of plaintext, and decryption is done by decoding. They tend to all use the same coding scheme (Goppa) and have larger keys. McEliece

1.6.2.4 Multivariate Cryptosystems

1.6.2.5 Isogeny

Traditional elliptic curve cryptography has its security based on the hardness assumption of the ECDLP. There is an immediate reduction from Shor's algorithm for the discrete log to the ECDLP.[8]

1.6.2.6 Others

Two of the NIST Submissions, Picnic and SPHINCS+ are based entirely on hash and blockcipher assumptions.

(These are Vlad's team, PICNIC, which only uses blockciphers and hash functions, and SPHINCS+ by DJB et. al which uses a hash function)

ML

References

[CW87] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.

@articlevercauteren2013final, title=Final report on main computational assumptions in cryptography, author=Vercauteren, F and others, journal=ECRPYT II, year=2013

https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[8] As a quantum circuit, Shor's algorithm for ECDLP is a little different than the one for DLP. It requires different subcircuits for elliptic curve operations