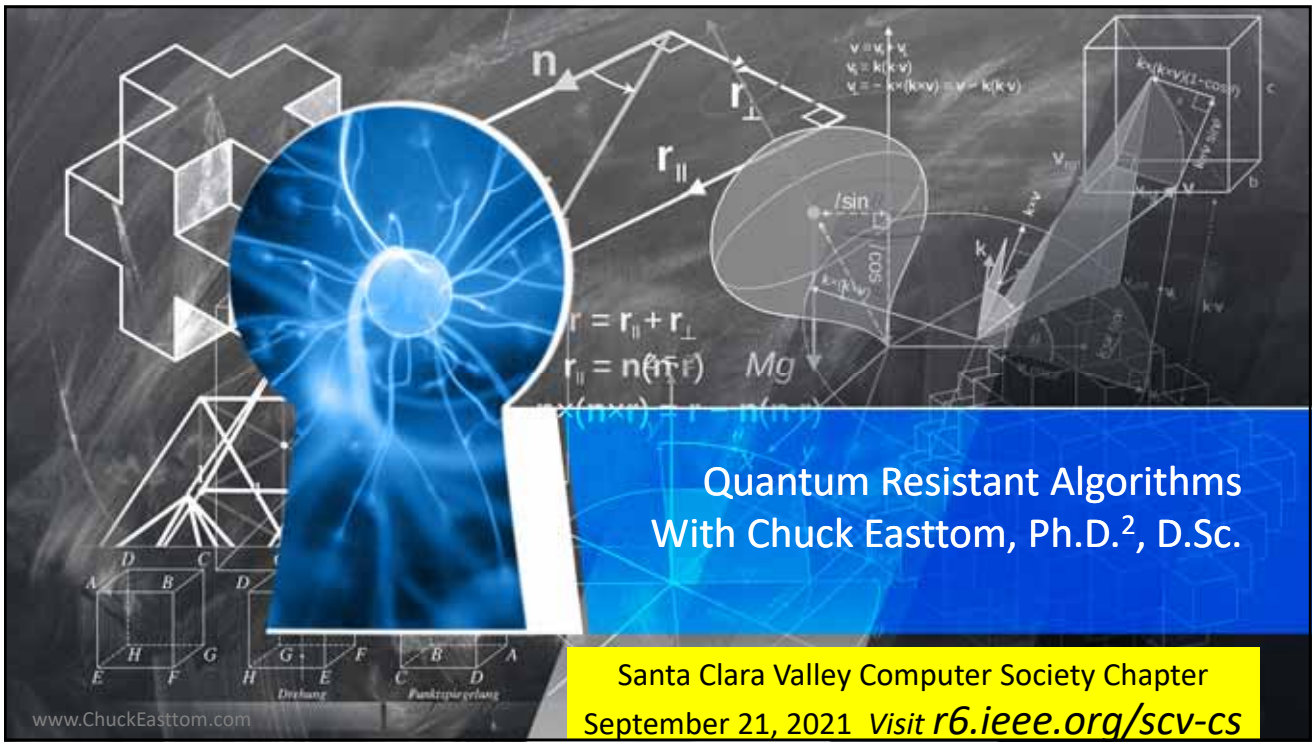
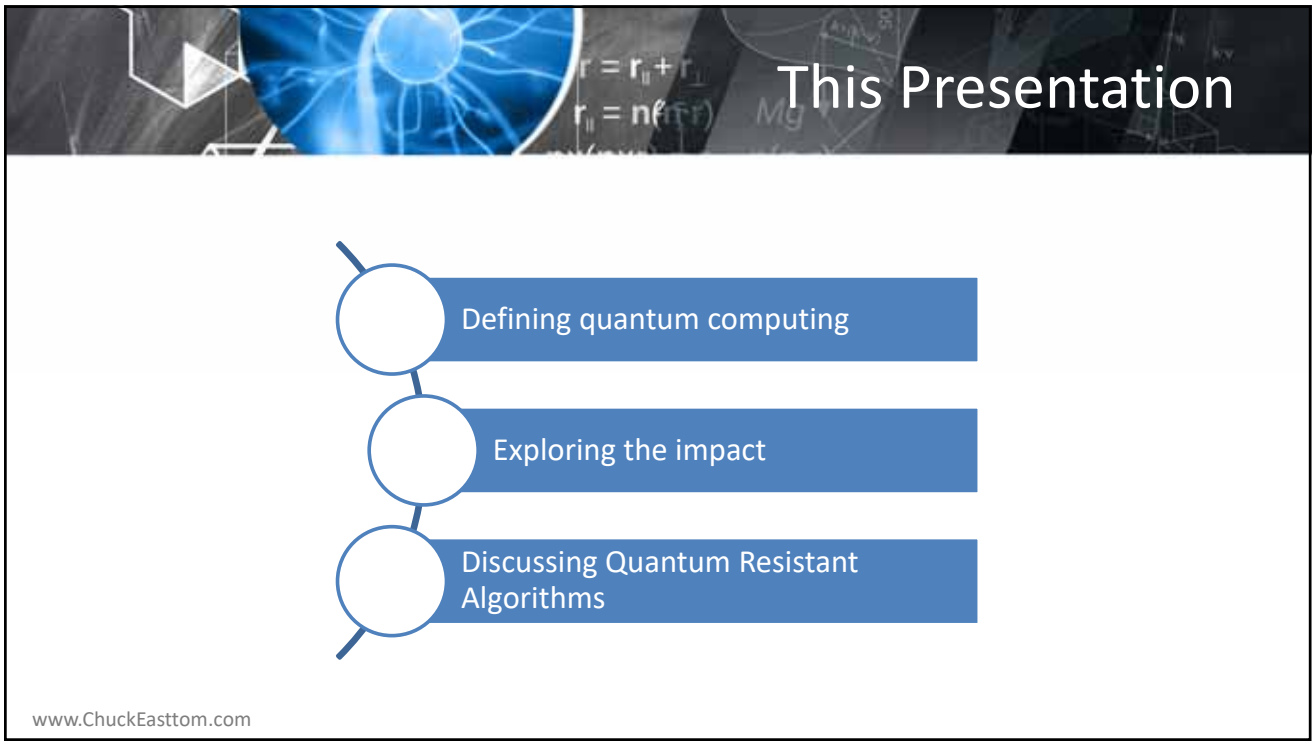
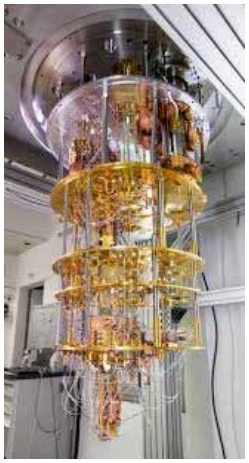

Quantum Resistant Algorithms
With Chuck Easttom, Ph.D.[2], D.Sc.
Santa Clara Valley Computer Society Chapter
September 21, 2021 Visit r6.ieee.org/scv-cs
www.ChuckEasttom.com


This Presentation
Defining quantum computing
Exploring the impact
Discussing Quantum Resistant Algorithms
www.ChuckEasttom.com

What is a quantum computer?

A quantum computer is a machine that performs calculations based on the laws of quantum mechanics, which is the behavior of particles at the sub-atomic level.



www.ChuckEasttom.com

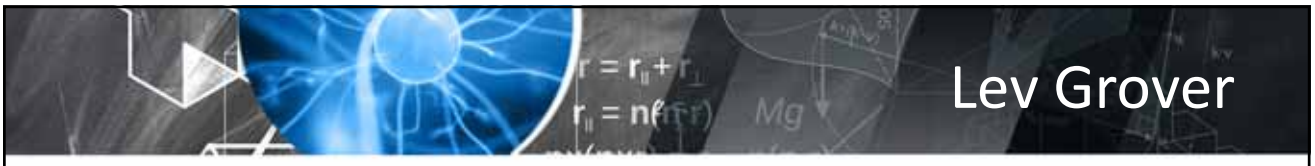
Peter shor

Peter Shor developed Shor's algorithm. On a quantum computer it can factor an integer N in polynomial time (actual time is log N). This is substantially faster than the most efficient known classical factoring algorithm (the general number field sieve) which works in sub-exponential time.

Peter Shor was awarded the Gödel Prize of the ACM and a MacArthur Foundation Fellowship in 1999

www.ChuckEasttom.com

4



Lev Grover

Grover's algorithm is essentially a search algorithm. It was developed by Lov Grover in 1996.

The efficacy of Grover's algorithm has been proven mathematically in multiple ways. It is one of the algorithms that confirm that power we will realize from quantum computers one decoherence is solved.

www.ChuckEasttom.com

But what do we need

These are rather general estimates. There are a lot of variables that effect these numbers:

- 100 qubits for quantum chemistry simulations
- 1000 qubits for effective machine learning
- 4000 qubits to factor 2048-bit RSA

Of course, how many qubits are needed, also depends on how fast one wishes to crack the algorithm in question. Researchers Graig Gidney of Google, and Martin Ekera from the Swedish Royal Institute of Technology demonstrated that 20 million qubits could crack 2048 bit RSA in 8 hours. You can find more about their work at https://cacm.acm.org/news/237303-how-quantum-computer-could-break-2048-bit-rsa-encryption-in-8-hours/fulltext This is actually far better than previous estimates. It had been estimated that 1 billion qubits would be needed to break 2048-bit RSA keys. For breaking RSA, it is all about factoring. As early as 2012 researchers used 4 qubits to factor the number 143.

An important fact we must address now is how many physical qubits are needed to implement a logical qubit? You might naturally suppose that it is a one-to-one relationship; however, that supposition would be inaccurate. There is not a specific correlation formula; however, it typically takes several *physical* qubits to implement one *logical* qubit. As an example, Shor's error correction code works by encoding a single logical qubit in nine physical qubits. The system is predicated on repetition codes in groups of 3 qubits. Equation 9.1 illustrates the general definitions of logical states

www.ChuckEasttom.com

NIST

Post quantum cryptography standards working group.

NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. Full details can be found in the Post-Quantum Cryptography Standardization page.

The submission deadline of November 30, 2017, has passed. Please see the Round 1 Submissions for the listing of complete and proper submissions.

- In recent years, there has been a substantial amount of research on quantum computers – machines that exploit quantum mechanical phenomena to solve mathematical problems that are difficult or intractable for conventional computers. If large-scale quantum computers are ever built, they will be able to break many of the public-key cryptosystems currently in use. This would seriously compromise the confidentiality and integrity of digital communications on the Internet and elsewhere. The goal of *post-quantum cryptography* (also called quantum-resistant cryptography) is to develop cryptographic systems that are secure against both quantum and classical computers and can interoperate with existing communications protocols and networks.
- The question of when a large-scale quantum computer will be built is a complicated one. While in the past it was less clear that large quantum computers are a physical possibility, many scientists now believe it to be merely a significant engineering challenge. Some engineers even predict that within the next twenty or so years sufficiently large quantum computers will be built to break essentially all public key schemes currently in use. Historically, it has taken almost two decades to deploy our modern public key cryptography infrastructure. Therefore, regardless of whether we can estimate the exact time of the arrival of the quantum computing era, we must begin now to prepare our information security systems to be able to resist quantum computing.

www.ChuckEasttom.com

Quantum Resistant Crypto

- Lattice Based
- Multi-variate
- Other

www.ChuckEasttom.com

Lattice Based Cryptography

A lattice can be defined as a set of points in some space having n-dimensions, which has a periodic structure. The basis vectors of the lattice are used to generate the lattice. A mathematical description of a lattice is

$$\mathcal{L}\ (b1, ...,bn) = \left\{ \sum_{I=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}$$

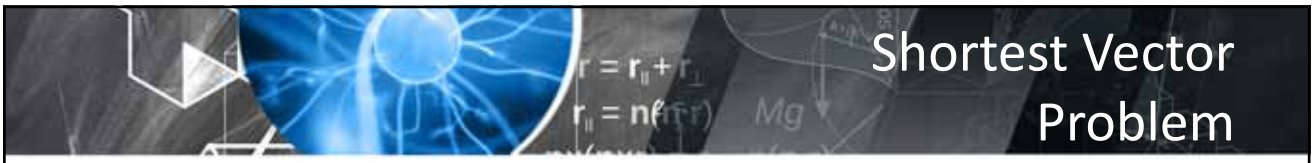
A lattice consists of vectors. Lattices use linearly independent vectors. A set of vectors is described as linearly dependent if any one of the vectors in the set can be defined as a linear combination of the other vectors. Conversely, if there is no vector in the set which can be defined in this manner, then the vectors are described as linearly independent. These vectors form the basis for a lattice.

www.ChuckEasttom.com

Shortest Integer Problem

The Short Integer problem (sometimes called short integer solution or SIS) is: Given an m x n lattice A which is comprised of *m* uniformly random vectors (which are integers), also stated as A ϵ z_q^{nxm}, find a nonzero short integer vector *v* satisfying such that Ax = 0 mod q. This problem forms the basis for Ajtai cryptographic primitive. There are other ways of stating this problem that you will find in some of the literature; however, this form is used here because it is the clearest.

www.ChuckEasttom.com



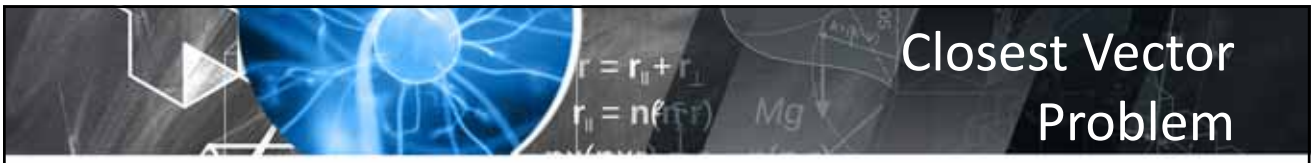
Shortest Vector Problem

The Shortest Vector Problem (SVP) is frequently used as a basis for lattice-based cryptographic systems. This problem is given a particular lattice, and a norm N for a lattice L, find the shortest non-zero vector in V as measured by N in L. Put more formally, the SVP problem is to find the shortest vector in the vector space V, as measured by a *norm*, N. Remember that a *norm* is a mathematical function which assigns a positive integer to each vector in a vector space, which is the vectors length or size. The shortest vector must be a non-zero vector.

Put even more formally, given lattice L, vector v and Norm n:

$$N(v) = \lambda L$$

www.ChuckEasttom.com



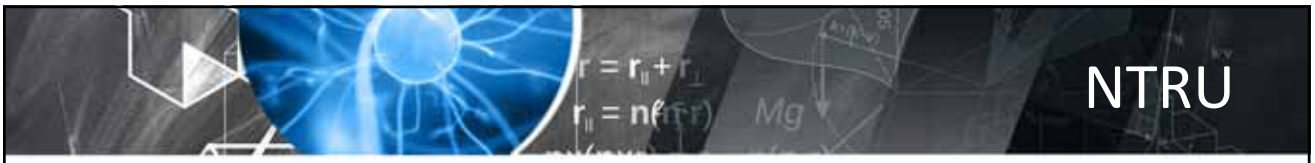
Closest Vector Problem

Another mathematical problem which is used in lattice-based cryptography is the Closest Vector Problem (CVP). This problem is given a particular vector space V, along with a metric M for a lattice L and a vector v which is in the vector space V, but not necessarily in the lattice L, how does one find the vector in the lattice L which is closest to the vector v. This problem is related to the previously discussed shortest vector problem and is also computationally hard to solve. In fact, the CVP is in fact a generalization of the SVP.

Given the relationship of CVP to SVP, you may assume there is a GapCVP much as there is a GapSVP. If you assume that, you are indeed correct. With GapSVP, the input consists of a lattice basis and a vector v and the algorithm must determine if one of the following is correct:

There is a lattice vector w such that the distance between w and v is at most 1.
Every vector in the lattice is at a distance greater than β from v.

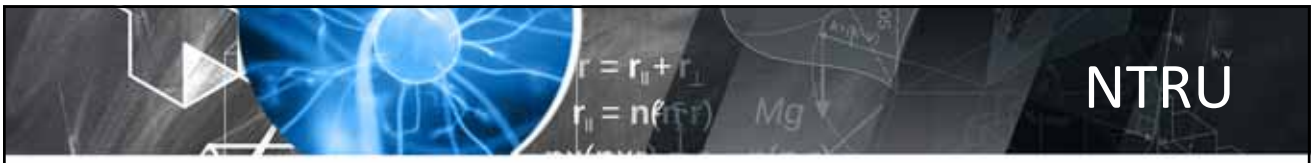
www.ChuckEasttom.com



NTRU was first publicly described in 1996 by Jeffery Hoffstien, Jill Pipher, and Joseph Silverman. There have also been additional variants of this algorithm developed since its initial publication. NTRU can best be defined as a group of related cryptographic algorithms. This group of algorithms has been one of the most studied lattice-based cryptosystems. It is a very important group of algorithms, particularly due to the fact that two variations of NTRU have made it past round 2 of the NIST project to find a quantum resistant cryptography standard.

NTRU is based on the shortest vector problem in a lattice. The security of NTRU is predicated on the computational difficulty of factoring certain polynomials in a given truncated polynomial ring. This requires that we briefly explore the concept of polynomial rings, for those readers who might not be familiar with them.

www.ChuckEasttom.com



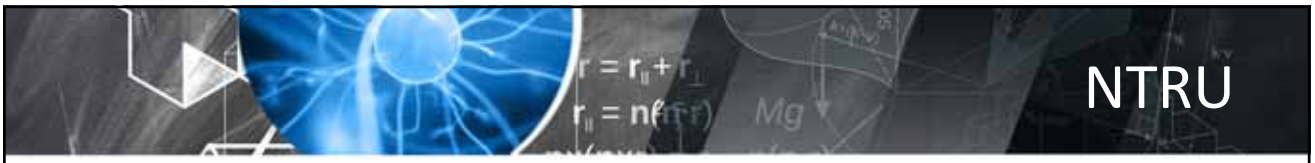
A polynomial ring is a ring formed from the set of polynomials in one or more indeterminates with coefficients in another ring. Recall that we discussed rings in chapter 1. A ring is a set with two operations, an identity element, two operations and the inverse operation of the first operation. A polynomial is an expression that consists of variables and coefficients. Another term for variable is indeterminate, as it was in the definition of a polynomial ring.

Now with that preliminary mathematical information covered, we can return to the NTRU algorithm. NTRU specifically utilizes the truncated polynomial ring shown here:

$R = z[x] \,/\, (x^N - 1)$

N is understood to be prime. The key generation process begins with the selection of N, followed by selection of p and q that must satisfy the criteria that gcd(p,q) = 1. Q is often some power of 2, and p is usually relatively small.

www.ChuckEasttom.com



N is understood to be prime. The key generation process begins with the selection of N, followed by selection of p and q that must satisfy the criteria that gcd(p,q) = 1. Q is often some power of 2, and p is usually relatively small.

The next step is the generation of two polynomials, usually labelled f and g, each having a degree at most N-1. The two polynomials have coefficients in {-1,0,1} (e.g. $-x^3$, $-x^2+x-1$). The polynomial f must also have inverses both modulo p and modulo q. If f does not have inverses both for modulo p and q, then a different f value must be selected. The next step is to actually calculate the invers of f mod p and f mod q. These are usually designated as f_p and f_q respectively. The public key is generated according to this equation.

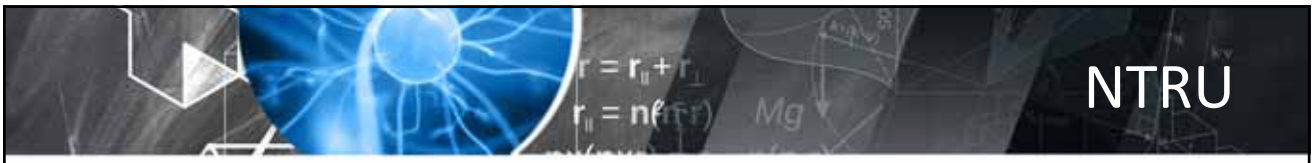
$h = pf_q g (mod\ q)$

The h is the public key. The polynomials f, fp and g are the private key. Now that we have a key, how do we apply it to encryption and decryption? We will use the traditional Alice and Bob for this discussion. Assume some message m that is in the form of a polynomial. We already have h and q. Alice wants to encrypt message m and send it to Bob. Alice now chooses some random polynomial r, usually with small coefficients. Now to encrypt Alice performs the following equation, shown in equation 3.

*e = r * h + m (mod q)*

When Bob receives this message, he will need to decrypt it. Bob takes the encrypted message e and uses equation 4 to decrypt:

*a = f * e (mod q)*



Note a, in equation 4, represents an intermediate step to the plaintext we wish to retrieve. Keep in mind that e is just r * h + m. And f, fp and g are private keys. This means we could re-write equation 4 as shown here:

a = f * (r * h + m) (mod q)

Remember the key generation process and that should tell you that h is really $pf_q g (mod\ q)$ so we can rewrite equation 7, to what you see here:

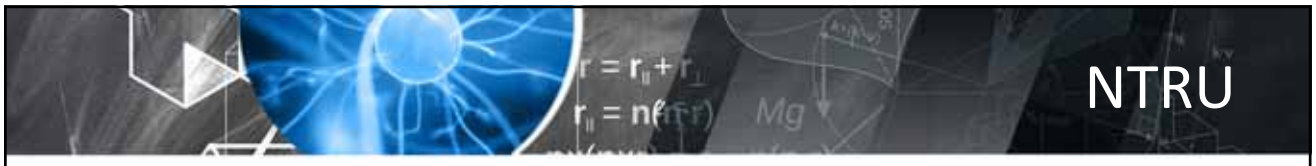
a = f * (r * pfq * g + m) (mod q)

You don't have to consider the permutation of the decryption equation. You can stay with the *a = f * e (mod q)* version. It was just important that you fully understand the process. Now Bob will need to calculate a polynomial, typically called b, that satisfies here:

b = a (mod p)

Recall Bob's secret key was f, fp and g, and we have not used all of those values yet, well now Bob will use them to get back the message m that Alice sent, using equation shown here:

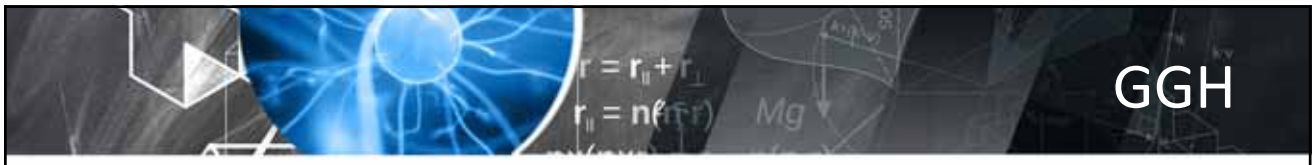
$m = f_p * b\ (mod\ p)$



Parameters

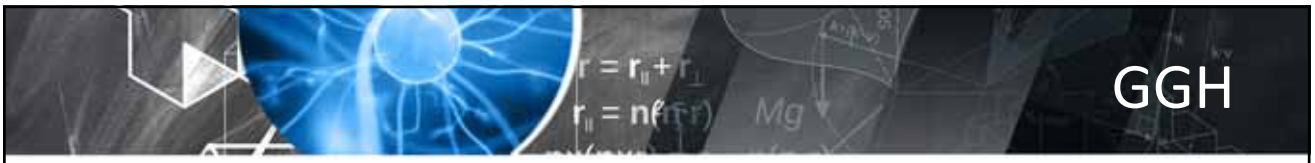
	N	q	p
Moderate Security	167	128	3
Standard Security	251	128	3
High Security	347	128	3
Highest Security	503	256	3

www.ChuckEasttom.com



The GGH algorithm, which is named after the surnames of its inventors Oded Glodreich, Shafi Goldwasser, and Shai Halevi, is a widely studied lattice-based cryptosystems. It is an asymmetric algorithm which has been demonstrated to be resistant to cryptanalysis. This algorithm was first publicly described in 1997. The algorithm was constructed using the closest vector problem (CVP). The private key is a basis vector B of the lattice L and a unimodular matrix U.

www.ChuckEasttom.com



This basis vector has certain properties such as the vectors are nearly orthogonal vectors and a matrix U which is unimodular. The public key is another basis of the same lattice of the form B' = UB. The message M is a message space which consists of the vector (m_1,..., m_n) in a range of $-M < m_i < M$.

The message is encrypted by multiplying the message vector by the public key B'. This is shown mathematically here:

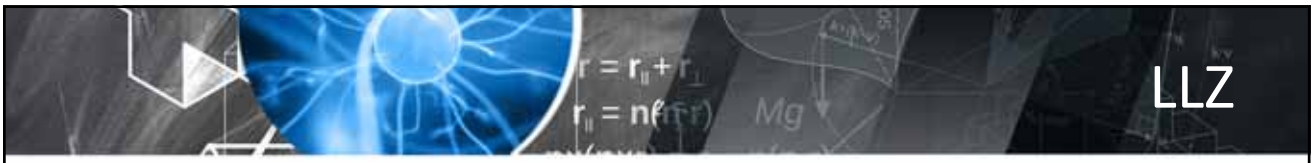
$$v = \sum m_i b'_i$$

Remember that m consists of integer values while B' is a lattice point. This means we can describe the cipher test as follows:

$c = v + e$

The e is an error correcting vector, (1, -1). To decrypt the message the cipher text, c is multiplied by the inverse of B, B^{-1}. Put mathematically:
$M = c\ B^{-1}$.

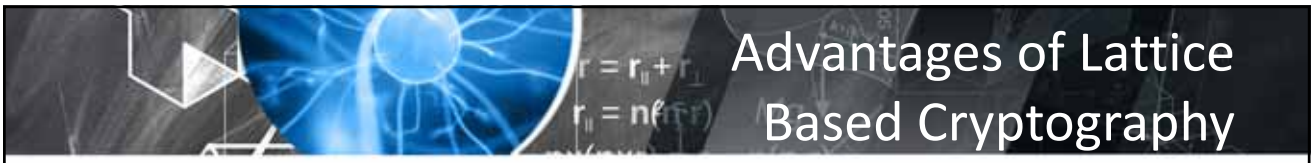
www.ChuckEasttom.com



This is a lattice reduction algorithm used to attack lattice-based algorithms

Lenstra, Lenstra and Lovasz , which finds an approximately short vector - guaranteed to be within a factor (2/3) n of the actual shortest - in polynomial time. LLL produces a "reduced" basis of a lattice, producing an approximately short basis vector as a result. The details of this algorithm will also be rather complex, but not such that most readers cannot follow the general outline. A few preliminaries. There is a Lattice L and a basis $B = \{b_1, b_2, \ldots b_d\}$ that has n-dimensional integer coordinates with $d \leq n$. The LLL algorithm calculates a reduced lattice basis.

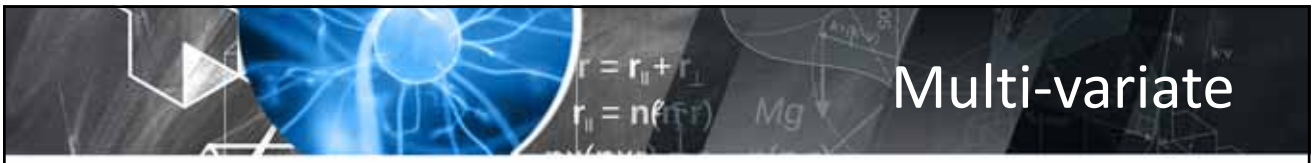
www.ChuckEasttom.com



Advantages of Lattice Based Cryptography

- Many well tested algorithms
- There are Lattice based hash function such as LASH
- Some Lattice Based algorithms are fully homomorphic (i.e. supports arbitrary computation on ciphertexts).
- Multiple Open Source implementations
- SSL/TLS library offering NTRU (WolfSSL).

www.ChuckEasttom.com

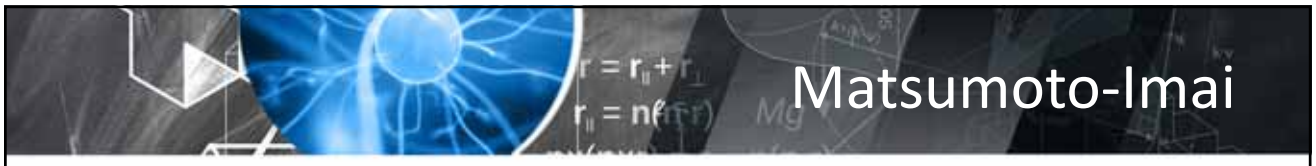


Multi-variate

Multivariate cryptography uses trapdoor one-way functions that take the form of a multivariate quadratic polynomial. The public key is given by a public set of quadratic polynomials:
P = (p1(w1…,wn),…pm(w1….,wn))
Each pi is a nonlinear polynomial in w = (w1,…,wn).

A trapdoor function is a function that is relatively easy to compute in one direction but computationally infeasible to compute in the other direction. Essentially, without knowing specifics about the inputs to the algorithm, it would take so much time and computing resources to solve the problem, that it is impractical to do so. Trapdoor functions are the key to many cryptographic algorithms.

www.ChuckEasttom.com



Matsumoto-Imai

This cryptographic algorithm was one of the first published multivariate systems. That makes it important to study. It was published by Tsutomo Matsumoto and Hideki Imai in 1988. This cryptographic system was later broken, so it is not currently used. However, its importance in the history of multivariate cryptography warrants a brief examination.

www.ChuckEasttom.com

Hidden Field Equations

The Hidden Field Equations (HFE) is a public key cryptography system that was invented by Jacques Patarin and first publicized at Eurocrypt in 1996. The cryptographic security is based on the difficulty of finding solutions to a system of multivariate quadratic equations, sometimes referred to as the MQ problem. This algorithm was created as an improvement of the Matsomoto-Imai system.

www.ChuckEasttom.com

McEliece

The McEliece cryptosystem is eponymously named after its inventor Robert McEliece. This algorithm was published in 1978. That makes it perhaps the oldest algorithm being proposed as quantum resistant. Despite its age, this algorithm has already been shown to be immune to Shor's algorithm.

The security of McEliece is based on the difficulty of decoding a general linear code. For those readers not familiar, a brief description of a general linear code is provided. A linear code is used for error correction. There is a linear combination of codes. Linear codes are frequently used in forward error correction. Hamming codes are a classic example of a linear code. Hamming codes can detect errors and correct them.

www.ChuckEasttom.com

Conclusions

www.ChuckEasttom.com

Resources - simulators

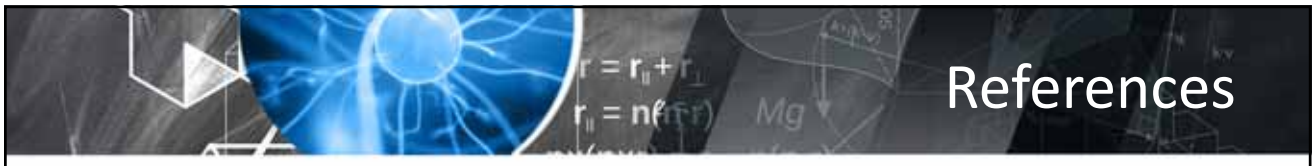
- http://www.quantumplayground.net/#/home
- https://algassert.com/quirk
- https://www.ibm.com/quantum-computing/technology/simulator/

www.ChuckEasttom.com

Resources – quantum computers

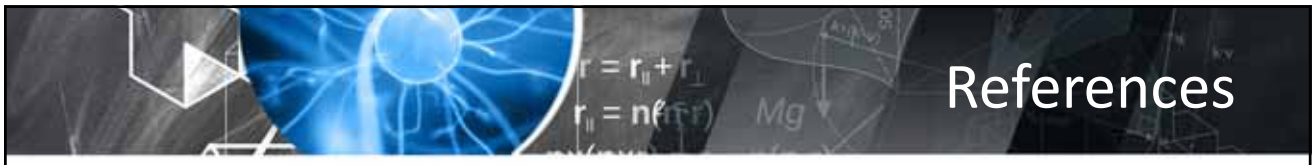
- D-Wave https://www.dwavesys.com/take-leap
- IBM https://www.ibm.com/quantum-computing/

www.ChuckEasttom.com



References

- Albash, T., Rønnow, T. F., Troyer, M., & Lidar, D. A. (2015). Reexamining classical and quantum models for the D-Wave One processor. The European Physical Journal Special Topics, 224(1), 111-129.
- Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. Nature, 549(7671), 188-194.
- Chen, L., Chen, L., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., ... & Smith-Tone, D. (2016). Report on post-quantum cryptography. US Department of Commerce, National Institute of Standards and Technology.
- Chi, D. P., Choi, J. W., San Kim, J., & Kim, T. (2015). Lattice based cryptography for beginners. IACR Cryptology ePrint Archive, 2015, 938
- Fano, G., Blinder, S. (2017). Twenty-First century quantum mechanics: Hilbert space to quantum computers: Mathematical methods and conceptual foundations. New York City, New York: Springer.
- Imre, S., & Balazs, F. (2013). Quantum computing and communications: An engineering approach. Hoboken, New Jersey: John Wiley & Sons.
- Kumar, R., Maurya, S. G., Chugh, R., & Manoj, P. V. (2014). Current refuge trends using classical and quantum cryptography. International Journal of Computer Science and Information Technologies, 5(3), 2974-77.
- Mariano, A., Laarhoven, T., Correia, F., Rodrigues, M., & Falcão, G. (2017). A practical view of the state-of-the-art of lattice-based cryptanalysis. IEEE Access, 5, 24184-24202
- Monteiro, R. T. (2016). Post-quantum cryptography: lattice-based cryptography and analysis of NTRU public-key cryptosystem (Doctoral dissertation). University of Lisbon, Portugal.



References

- Moret-Bonillo, V. (2017). Adventures in computer science: From classical bits to quantum bits. New York City, New York: Springer.
- Peikert, C. (2016). A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science, 10(4), 283-424.
- Raychev, N. (2015). Quantum computing models for algebraic applications. International Journal of Scientific & Engineering Research, 6(8), 1281-1289.
- Rieffel, E., Polak, W. (2011). Quantum computing: A Gentle introduction. Boston, Massachusetts: MIT Press.
- Shenoy-Hejamadi, A., Pathak, A., & Radhakrishna, S. (2017). Quantum cryptography: Key distribution and beyond. Quanta, 6(1), 1-47
- Stanescu, T. (2016). Introduction to quantum matter & quantum computation. Boca Raton, Florida: CRC Press.
- Trabesinger, A. (2017). Quantum computing: towards reality. Nature, 543(7646), S1-S1.
- Wang, D. S., Hill, C. D., & Hollenberg, L. C. (2017). Simulations of Shor's algorithm using matrix product states. Quantum Information Processing, 16(7), 176-183.
- Easttom, C. (2019). "An Analysis of Leading Lattice-Based Asymmetric Cryptographic Primitives". 2019 IEEE 9th Annual Computing and Communication Conference.

www.ChuckEasttom.com