Post-Quantum Cryptography – Hardware and Software Implementations

Student: Jeremy Barthélemy
Advisor: Dr. Kris Gaj
George Mason University, Fall 2013

Abstract --Due to the advent of quantum computers, cryptographic encryption and key distribution schemes must continue to evolve if certain communication schemes are to remain secure. Although the security of symmetric cryptosystems is only slightly weakened by quantum computers, the security of widely popular public-key cryptosystems such as RSA and ECC is severely threatened. This paper presents a basic introduction to quantum computers as well as a brief history of the development of quantum computers.

Several candidate public-key cryptosystems are discussed and some of the many current implementations of these schemes in both hardware and software are analyzed in terms of efficiency and security.

Key words— Post-Quantum Cryptography; Shor's Algorithm; Lattice-Based Cryptography; Coding-Based Cryptography; MQ-Based Cryptography

1. Introduction

1.1 Overview

If quantum computers become widespread in years to come, they will have a large impact upon network communications such as the Internet due to the implications that arise from the ease of breaking a variety of cryptographic schemes.

One major concern is that, with quantum computing, public-key cryptography is broken, and only secret key cryptography will remain secure, albeit slightly weakened by quantum computers. This is not entirely true. As we will see, there are still schemes which should still theoretically be safe to use-even when and if the day of mass quantum computing arrives.

Table 1: Cryptosystems which are currently known to be vulnerable to quantum algorithms

Cryptosystem	**Broken?**
RSA	Broken
Diffie Hellman	Broken
ECC	Broken
Buchmann-Williams	Broken
Algebraically Homomorphic	Broken
McEliece	Not yet broken
NTRU	Not yet broken
Lattice-Based	Not yet broken

If this day arises in the future where quantum computers have a larger number of qubits and are more readily available to more people, many public-key cryptosystems which rely upon the difficulty to perform computations on a conventional computer will quickly and easily be broken due to the seemingly parallel processing of quantum computers (i.e. the ability to perform the same computation with different values at the same time).

There are many who still doubt that working quantum computers with enough computational power will ever be developed, or at least not for a very long time due to the difficulties of applying with quantum mechanics principles to the actual device.

Despite this possibility, it is still worthwhile to dedicate research to post-quantum cryptography schemes and to test current, less popular public-key cryptosystems for possible vulnerabilities as a safeguard against the chance of common quantum computer usage arising before such secure cryptosystems are in place (most experts are predicting another 20 years before the advent of quantum computing).

This is a particular danger, as, public-key cryptography schemes offer particular features which secret-key cryptographic schemes do not (at least not in the case without a trusted third-party). One major feature of public-key cryptography is the application of digital signature schemes. These schemes allow for sender authentication as well as non-repudiation. For example, a sender may sign a message by encrypting it with its private key and distributing it to various receivers. These receivers may then use the sender's public key to decrypt the message, thus verifying that the sender did indeed sign the message.

Public-key cryptographic schemes, despite being more computationally intensive than their symmetric partners, are also commonly used for key agreement. For example, RSA may be used first for key agreement between two end users before AES is used for encrypting the data sent between the two users. It would not make sense to use RSA for all of the encryption as it would be much slower, just as it would not make sense to send the shared secret key first across a secure channel.

A main challenge at hand for post-quantum cryptography though is not only to have a public-key scheme which is resistant against quantum computers, but also to be as efficient as currently existing public-key standards such as Elliptic-Curve Cryptography (ECC) or RSA. We will see in this paper that there are already currently existing high-throughput implementations of public-key cryptosystems which are resilient against quantum algorithms.

1.2 What is a Quantum Computer?

Definition: “a computer that makes use of the quantum states of subatomic particles to store information.” A quantum computer is a computer which is able to operate based upon the principles of quantum mechanics. The idea of quantum computers began with Yuri Manin in 1980 and Richard Feynman in 1982. While classical computers use bits for representation of information, quantum computers will use “qubits,” which have some key differences from normal bits, which allows for the improvements of quantum computers over classical computers.

There are some fundamental differences with regard to quantum information and classical information. The main difference between a classical bit and a quantum bit is that a classical bit must hold the value of a 0 or a 1, whereas the principle of superposition allows qubits to have both states at the same time, a fundamental concept of quantum computing. That is, a qubit can have a value of 0, 1, or both!

The improvement of a quantum computer over a normal computer is not in the speed of the steps of an operation, but in the total number of operations. Superposition and entanglement are used to perform operations upon data. There is a widespread interest in quantum computers from military to enterprise solutions due to the alleged ability of quantum computers to be able to solve specific problems much more quickly that would be solved by classical computers. Shor's algorithm, in particular, has been recognized as being one of the schemes which could be utilized by quantum computers to break many public key cryptosystems.

Public-key cryptography relies upon the difficulty in performing certain mathematical computations. Quantum computers will challenge this in that they will be able to easily perform many of these computational problems.

Aside from cryptographic use for quantum computers, they could also be very useful at searching large databases for information, and could do so in a tiny fraction of the time a conventional computer would take.

1.3 What is Post-Quantum Cryptography?

Post-quantum cryptography is a field of study for the goal of researching into new algorithms for public-key cryptosystems such that these new schemes will not be easily breakable by quantum computers, at least not much more than any classical computer.

Most modern public-key cryptosystems rely upon the difficulty of computation resulting from the integer factorization problem and from the discrete logarithm problem and so quantum computers will easily be able to solve these problems through the use of Shor's algorithm. The purpose of the study of post-quantum cryptography is to ensure that if and when quantum computers become powerful enough to attack the public-key cryptosystems in place, these will have already been replaced by new algorithms which will not be easily broken. Quantum computing would break most public key cryptosystems in use today because quantum computing would force the use of keys which would be so long that it is practically impractical to use them. The challenge is to develop new mathematical algorithms which will allow information security even in a world of quantum computers, yet still remain feasible in terms of efficiency.

The top candidates for quantum cryptography so far are: Coding-based schemes, Lattice-based cryptosystems, as well as MQ cryptosystems.

2. A Brief History of Quantum Computers

The original idea of quantum computing stems back to a speech given by Richard Feynman **in 1959**, where he spoke of the possibility of using the phenomena of quantum mechanics in the process of creating more powerful computers.

In 1985, David Deutsch came up with the idea of quantum logic gates, whereby his paper proved that any physical process can be simulated with a quantum computer.

In 1994, Peter Shor designed a quantum algorithm for using just 6 qubits (quantum bits) in order to perform factorizations of integers.

The first quantum computer ever was built **in 1998** with just 2 qubits.

It was able to perform some trivial computations, but would lose coherence in a very short time (just several nanoseconds).

In the year 2000, a 7-qubit quantum computer was developed by Los Alamos National Laboratory. These 7 qubits were all located within a single drop of liquid.

In 2001, IBM demonstrated Shor's Algorithm by showing that they were able to factor 15 into the two prime numbers 3 and 5 using a Nuclear Magnetic Resonance quantum computer with 7 qubits.

In 2005, the first qubyte was created by the Institute of Quantum Optics and Quantum Information of the University of Innsbruck. This was done by creating a series of 8 qubits, which were controlled using ion traps.

In 2006, scientists in Massachusetts establish methods for controlling a 12-qubit system.

In 2007, a Canadian startup known as D-Wave successfully demonstrated a 16-qubit quantum computer which was able to solve a Sudoku puzzle.

In 2011, D-Wave Systems claims their D-Wave One system uses a 128-qubit processor chipset.

In September 2011, researchers also proved that a quantum computer can be made with a Von Neumann architecture (separation of RAM).

I**n November 2011**, physicists at the University of Science and Technology of China in Hefei, China factorized 143 into prime factors 11 and 13 using just 4 qubits, which today is yet the largest number to be factored using a quantum algorithm.

In May 2013, it was announced by Google that they would be launching a "Quantum Artificial Intelligence Lab," which will hold a 512-qubit quantum computer developed by D-Wave Systems.

As of November of 2013, an international team of researchers led by Mike Thewalt of Simon Fraser University in Canada were able to maintain the superposition states of qubits for an entire 39 minutes, thus breaking all previous records by a longshot. They were able to do this by encoding information into the nuclei of phosphorous atoms which were held inside slices of pure silicon at -269°C. Magnetic field pulses were then used to modify the spin of the nuclei for the purpose of creating superposition states. These qubits were then brought to room temperature, where their superposition states began to decay. When they were left at cryogenic temperatures, however, the quantum memory was able to remain stable for up to three hours.

Today, quantum computers are still in their infancy. The most advanced quantum computers do not yet have enough qubits to make them actually useful for performing factorizations of very large numbers.

In the future, we will see quantum computers which are able to perform many computations much more quickly than in a classical computer. Although it does not yet seem likely that quantum computers will replace conventional computers entirely, they could be used for special-purpose uses where they perform better than classical computers and hybrids of classical and quantum computers could be made (e.g. the possibility of having clients with classical computers accessing a remote quantum server for performing fast computations when needed.

3. A (Qu)bit of Physics: How Quantum Computers Work

Conventional computers work by altering the states of bits between 0s and 1s in order to perform tasks. Whether it is the communication of information, sending an email, or watching a video online, computers represent information with just 0s and 1s. Quantum computers, on the other hand, are not limited to this. The way for encoding information is through the use of qubits, which is a portmanteau of “quantum bits.” These quantum bits are not limited to being just 0s and 1s. On the other hand, these qubits can also be in superposition, a characteristic typical of quantum mechanics.

Qubits can be electrons, which have control devices which work together to simulate computer memory and processors. They can contain multiple states in a single moment, and have the potential of being millions of times more powerful than even the most powerful classical computers used today.

Quantum superposition is the principle that, given an electron, it will exist partially in every theoretically possible state at the same time. This means that there is more than just two states for a single qubit, and as more qubits are added, the number of states possible will increase exponentially, more dramatically so than the increase of conventional bits would.

The catch is that, due to quantum decoherence, once the electron is observed, the result will decide upon only one of the possible configurations.

It is due to this characteristic of superposition that allows quantum computers a new level of parallelism, different from the likes of parallel-processing through multithreading and such. Physicist David Deutsch claims that a quantum computer should be able to work on a million computations at one single time, whereas a PC would be working on only one.

4. Shor’s Algorithm

Shor’s Algorithm was initially introduced in 1994 by mathematician Peter Shor. It is recognized as one of the key factors which led to research into the field of post-quantum cryptography.

This algorithm was designed to take advantage of the proposed computational capabilities of a quantum computer, if ever such a device is constructed.

There are two main parts to Shor’s algorithm. The second part must be done on a quantum computer but the first part may or may not be run on a classical computer.

If Shor's algorithm were to be implemented on a quantum computer with a sufficient number of qubits, it could theoretically be used to break RSA as well as a variety of other public-key cryptographic systems which rely upon the assumption that discovering the prime factors of large integers is computationally very difficult for classical computers. This is because, by its nature, Shor’s algorithm proves that factoring can be efficient on an ideal quantum computer.

In other words, given a very large integer N, Shor’s algorithm can efficiently find N’s prime factors.

The quantum part of Shor's algorithm relies upon "superposition," or, the nature of qubits being in multiple states at one moment in order to determine p and q given N.

Shor’s algorithm can also be modified for other applications which could help to break the security of other public-key systems which rely upon other computations which are assumed difficult on classical computers. Some of these applications have not been discovered however.

5. Candidates for Post-Quantum Cryptography

These schemes are three of the major candidates for post-quantum cryptography. They are candidates specifically because, at this point in time, there are no known quantum algorithms which allow for the solving of their problems with any performance significantly superior to those already existing for classical computers. There have been attempts in the past to extend Shor's algorithm to these schemes since 1994, but so far without any substantial success.

The question of being "hard" to solve lies on the basis of an assumption on computational hardness. An adversary's natural computational limitations allow for this assumption, as the harder a problem is to compute, the more secure the system can be assumed to be. Of course with a new algorithm, a computation could go from being very time-consuming to perform to taking minimal time, thus greatly reducing a system's security.

Different levels of complexity can be adjudged for cases where we have different inputs of size n to an algorithm. Some inputs of size n are naturally much faster to solve than others, and so computational complexities for an algorithm can range between: best-case, average-case, and worst-case complexities.

For the case of cryptography, a "bad" worst-case scenario is actually a good one. In other words, given inputs of size n, we want the computations to take as long as possible for an adversary to a cryptographic scheme. An optimal case would be to prove that the average case for an algorithm is no easier than the worst case.

5.1 Coding-Based Cryptography

5.1.1 Background

In recent years, coding-based schemes have been established as a possible alternative for public-key encryption schemes. In particular, these are of interest due to their resilience against quantum computers.

There is one major drawback to coding-based schemes in that they have very large key sizes compared with current popular asymmetric schemes.

The McEliece Cryptosystem was developed in 1978 by Robert McEliece. McEliece is based upon the assumption that decoding unknown linear, binary codes is NP Complete. The idea of a Niederreiter Cryptosystem was developed in 1986 by Harald Niederreiter.

It is a variation of the McEliece Cryptosystem, with equivalent security but with ten times the encryption speed of McEliece. Niederreiter can also be used for the construction of digital signatures.

It is an asymmetric encryption algorithm that uses randomization during the encryption process.

McEliece with Goppa codes and Niederreiter algorithms have resisted cryptanalysis so far.

Of the variety of attacks proposed against McEliece and Niederreiter cryptosystems, information set decoding appears to be the most effective, which have caused the original security parameters to require larger values than they were initially proposed to be.

RSA has significantly larger key sizes today than what was initially proposed. This is because of the rapid improvements of computation speeds of computers. This appears to be the case for McEliece cryptosystems as well, as Canteaut and Sendrier of Le Chesnay, France, and also researchers at Eindhoven University of Technology in the Netherlands have managed to point out that current security of McEliece is also not sufficient with the currently proposed parameters (similar to the case of RSA's originally proposed parameters).

In 2008, Bernstein, Lange, and Peters proposed a possible attack on the McEliece cryptosystem, which was able to be carried out in $2^{60.55}$ bit operations, which was based upon finding low-weight codewords.

Also in 2008, the Eindhoven University of Technology was able to prove their method of speeding up attacks against the McEliece cryptosystem by having a piece of software run on a distributed system of several dozens of computers in the Netherlands, France, Ireland, Taiwan, as well as the USA. After some time, a computer in Ireland was able to deduce the ciphertext. This proves that in order to McEliece to remain a leading candidate for post-quantum cryptography, there would need to be a scaling of the key sizes to a larger value to avoid the vulnerability which their attack took advantage of.

Canteaut and Sendrier claim that the McEliece and Niederreiter cryptosystems are still a valid alternative to RSA once the parameters are adjusted. They state that, given conservative modifications, Niederreiter cryptosystems can have a better cost of encryption per information bit that is 45 times lower than RSA-1024, and a better cost of decryption that is 70 times lower. They also mention, however, that given the parameters for this scenario, the public key will be larger than 88,000 bits, and that this may dissuade users from using this particular cipher as opposed to another alternative. [1]

5.1.2 Coding-Based Implementations

In 2009, Stefan Heyse, of Ruhr-University Bochum, implemented the McEliece encryption scheme on a Xilinx Spartan-3 FPGA board. Implementing this efficiently was seen as being very difficult considering the challenge of storing the very large keys. Using a Spartan-3 200-5 board, Heyse was able to achieve a throughput of 779,948 bits per second. This is particularly impressive when compared with an ECC-P160 implementation on a Spartan-3 1000-4 board with a throughput of 31,200 bits per second, and an RSA-1024 implementation on a Spartan-3E 1500-5 with a throughput of 20,275.
These parameters for RSA and ECC were selected as a rough approximation to have equivalent security as an 80-bit symmetric cipher.

Table 2:
Area for Encryption with n = 2048, k = 1751, t = 27 on a Spartan-3 FPGA after PAR

Resource	With Uart and Debug	Without Uart and DEBUG	Available
Slices	796 (41%)	694(36%)	1920
LUTs	1082(28%)	870(22%)	3840
FFs	1101(28%)	915(23%)	3840
BRAM	1(8%)	1(8%)	12
Flash	4644Kb	-	16896Kb

Table 3:
Area for Decryption with the same parameters

Resource	With Uart and Debug	Without Uart and DEBUG	Available
Slices	12977(63%)	12443(60%)	20480
LUTs	17974(43%)	18637(45%)	40960
FFs	9985(24%)	9894(23%)	40960
BRAMs	22(55%)	22(55%)	40
Flash	11218(100%)	4644Kb	16896Kb

A particular hardware implementation of interest is the QC-MDPC McEliece Implementation by Stefan Heyse, Ingon von Maurich, and Tim Güneysu of Horst Görtz Institute for IT-Security in Bochum, Germany. Their main goal was to tackle the issue of the large key-size which has been typically associated with McEliece and other code-based schemes.
They were able to achieve a microcontroller implementation which used only 4800 bits for the public key and 9600 bits for the secret key at an equivalent security of 80 bits for secret key security. The design goals of these implementations were mainly high throughput as well as low memory consumption. This effort illustrates the possibility of reducing the large key size for code-based schemes.

5.2 Lattice-Based Cryptography

5.2.1 Background

Lattices were first studied and introduced by famous mathematicians Joseph Louis Lagrange and Carl Friedrich Gauss. In recent years, lattices have been used in cryptanalysis as well as with homomorphic encryption (the latter of the two as announced in 2009 by IBM, having been developed by Craig Gentry).

Homomorphic encryption is a means by which a user is able to perform computations upon already encrypted data, without actually corrupting it. So a user Alice can pass some data to a remote server in encrypted form, have the server perform computations, pass back a text, decrypt it, and view the results of her computations on a new plaintext. Craig Gentry based the security on the hardness of worst-case problems over ideal lattices and upon the sparse subset sum problem.

Of particular interest in our case, however, are the possible impacts that lattice-based cryptography can have upon quantum cryptography. This is due to the simplicity of developing their implementations, as well as to their supposed security when dealing with modern attacks as well as quantum computers. At this point, there are only a handful of implementations with guaranteed security which are provable (worst-case hardness of lattice problems), but even fewer of these are efficient enough to be feasible for usage. There are other schemes which are much more efficient, but have yet to be proven to have guaranteed security.

5.2.2 The Lattice Problem

Lattice-based cryptography is resilient against quantum computers due to the lattice problem.

For example, there are the SVP and CVP mathematical lattice problems. SVP, short for Shortest Vector Problem, is the case where, given a basis of a lattice, the shortest vector in the lattice must be found. On the other hand CVP, short for Closest Vector Problem, is the case where, given a basis of a lattice as well as a vector which is not in the lattice, the lattice vector with the smallest distance to the first vector must be found.

Some lattice problems have been proven to be average-case hard, which is a property beneficial for cryptography use. There are, however, methods for lattice reduction which aim to convert an average basis for the algorithm to a good basis. A popular such algorithm is the LLL (Lenstra–Lenstra–Lovász) algorithm, which is an efficient scheme for giving an output of an almost reduced lattice basis in polynomial time. The LLL algorithm thus led many to believe that the lattice-problem could actually become an easy problem in practice.

Miklós Ajtai, a computer scientist at IBM, was able to cast some doubt upon this assumption with his research on SVP lattices. Ajtai then went on to develop a public-key cryptosystem with Cynthia Dwork with equivalent worst-case and average-case complexity in 1997. Today, despite the LLL algorithm, the lattice problem still seems a possible solution for the security weaknesses imposed by the advent of quantum computers upon many public-key cryptography schemes.

5.2.3 Lattice-Based Implementations

Two popular schemes based upon lattice-based cryptography are the GGH (Goldreich-Goldwasser-Halevi) and the NTRUEncrypt encryption schemes. The GGH scheme makes use of the closest vector problem (CVP), whereas NTRUEncrypt, on the other hand, makes use of the shortest vector problem (SVP). At the Crypto 1999 conference, Phong Q. Nguyen was able to illustrate that the GGH has a flaw in the design. Apparently every piece of ciphertext reveals some information about the plaintext. Due to this issue, decryption could be simplified into a special closest vector problem, which is much easier to solve than the general case. NTRUEncrypt is still considered secure, although admittedly it has not received as much attention as many other, more popular schemes, which introduces the danger of having some unknown security flaws.

NTRUEncrypt is implemented as part of "The NTRU Project," which is an open source software collaboration. At this point, NTRU is not known to be breakable even by quantum computers, and is also significantly faster than many other public-key cryptography systems.

Below are two illustrations of the efficiency of this implementation when running on an Intel i3 processor at 3.1GHz for 32 bits and 64 bits, respectively.

Figure 1: Operations per second compared between different cryptography schemes on different key sizes, on a 32-bit processor [12]

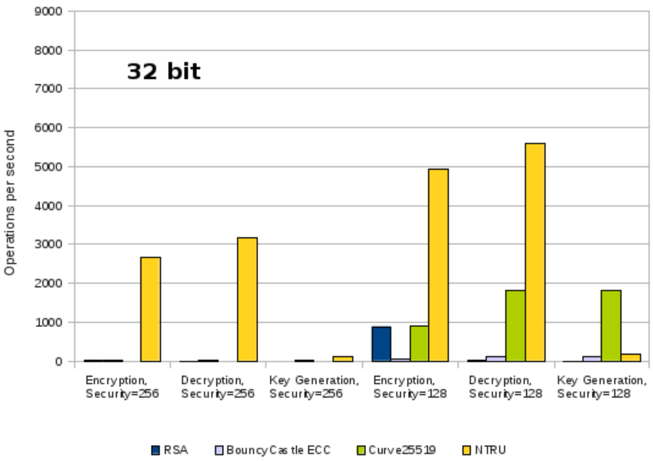
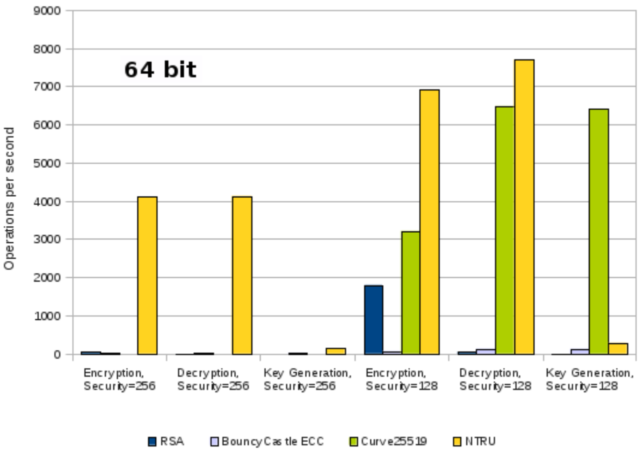


Figure 2: Operations per second compared between different cryptography schemes on different key sizes, on a 64-bit processor [12]



5.3 Multivariate Cryptography

5.3.1 Background

Multivariate-Quadratic Cryptography schemes range from the Matsumoto-Imai scheme, Hidden Field Equations, Unbalanced Oil and Vinegar schemes, and Step-wise Triangular Systems.

This branch of cryptosystems is based upon the difficulty of finding solutions to a system of multivariate quadratic equations. This is known as the MQ problem.
It uses private affine transformations to hide the extension field and the private polynomials.

5.3.2 Multivariate Public Key Implementations

One multivariate scheme was implemented by El-Hadedy, Gligoroski, and Knapskog for high performance on FPGA platforms. The implementation was of an MQQ, or Multivariate Quadratic Quasigroup, a special type of multivariate scheme. Using four Xilinx Virtex-5 boards running at 276.7 MHz clocks, they were able to have an encryption throughput of 44.27Gbps.
Based upon their implementation, they were able to achieve decryption rates of 399Mbps on a Xilinx Virtex-5 FPGA board which was running on a 249.4 MHz clock.

When these results are compared to traditional RSA implementations on Virtex-5, El-Hadedy et. al. had an encryption throughput 17,000 times faster than RSA and a decryption rate of 10,000 times faster than RSA.

Table 4: Synthesis Results for 160-bit MQQ with Virtex-5 [2]

	Slices	LUTs	Initial Delay	Cycles per op	Max Clk
Encryption	13137	25285	3.614	1	276.7
Decryption	1148	6993	4.01	100	249.4

Table 5: Hardware Performances of 1024-bit RSA, 160-bit MQQ, and 128-bit AES on Xilinx Virtex-5 FPGAs [2]

	1024-bit RSA	160-bit MQQ enc/dec	128-bit AES
Frequency	251 MHz	276.7/249.4 MHz	325 MHz
Throughput	40 Kbps	44.27 Gbps/ 399.04 Mbps	3.78 Gbps

The security of HFE and Multi-HFE schemes is analyzed by Luk Bettale, Jean-Charles Faugère, and Ludovic Perret in their paper, "Cryptanalysis of HFE, Multi-HFE, and Variants for Odd and Even Characteristic."
Through their analysis, they were able to practically break Multi-HFE with the originally proposed parameters in just a few days and with 256 bits security in only 9 days.
This indicates that for HFE (and very likely other MQ-based cryptosystems) will need to have its parameters modified so as to improve security.

Another algorithm to of MQ to analyze is the fast signature scheme known as the Matsumoto-Imai algorithm. This algorithm was a main proponent in the development of MQ post-quantum cryptography algorithms but had several faults.

In a paper [3] by Delsarte, Desmedt, Odlyzko, and Piret, the authors claim that it has apparent weaknesses and is easily broken. They state that the algorithm has appeal due to its high speed, but that it is totally insecure. Using their algorithm, they state that given different assumptions on performance of a computer along with the approximate required steps for their algorithm, that the cryptanalysis of this algorithm could take 8 hours with a fast computer (100ns per step). Using another algorithm based upon shift operations, they can do the same in just $2*10^6$ steps, which would result in about 0.2 seconds to attack. They go on to state that this algorithm could be improved by increasing security parameters m and r, but that this could lead to impractically large storage requirements.

6. Summary/Conclusions

In summary, we have found it to be vital to study post-quantum cryptography if we are to ensure the security of public key cryptography in the future. Although by no means will quantum computers destroy all of public key cryptography, new methods of public key cryptography will need to be developed to ensure maximum security.

The three main public key cryptosystems analyzed in this paper are not as popular as RSA or ECC, and thus have been tested and analyzed for vulnerabilities less, despite their appeal for use with quantum computers.

There still remains a great need for all of these cryptosystems to be more thoroughly analyzed in order to ensure that they can withstand attacks from classical computers as well as quantum computers, if and when they become powerful enough for such attacks.

References

[1] Anne Canteaut, Nicolas Sendrier, “Cryptanalysis of the Original McEliece Cryptosystem”, INRIA, 1998
[2] Mohamed El-Hadedy et. al, “High Performance Implementation of a Public Key Block Cipher - MQQ, for FPGA Platforms”, IACR, 2008
[3] Delsarte et. al, “Fast Cryptanalysis of the Matsumoto-Imai Public Key Scheme”, Proceedings of EUROCRYPT 84 A Workshop on the Theory and Application of Cryptographic Techniques Paris, France, April 9– 11, 1984
[4] Daniel J. Bernstein, "Grover vs. McEliece", Post-Quantum Cryptography, 2010
[5] Daniel J. Bernstein, “Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?”, SHARCS'09 Workshop Record (Proceedings 4th Workshop on Special-purpose Hardware for Attacking Cryptograhic Systems, Lausanne, Switserland, September 9-10, 2009)
[6] Johannes Buchmann et. al., “Post-Quantum Cryptography” Springer-Verlag Berlin Heidelberg, 2009

[7] Peter Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", AT&T Research, 2009
[8] IEEE Spectrum, “Q&A With Post-Quantum Computing Cryptography Researcher Jintai Ding”, 2008
[9] ScienceDaily, "Quantum Computers? Internet Security Code Of The Future Cracked", 2008
[10] Monica Heger, "Cryptographers Take on Quantum Computers", IEEE Spectrum, January 2009
[11] James Morgan, "Quantum Memory 'World Record' Smashed", BBC News, 14 November 2013
[12], Tim Buktu 2011, “http://tbuktu.github.io/ntru/”, The NTRU Project
[13] Daniele Micciancio and Oded Regev, "Lattice-Based Cryptography", New York University Courant Institute of Mathematical Sciences, 22 July 2008
[14] Daniele Micciancio, "The Geometry of Lattice Cryptography", UCSDCSE, 16 February 2012
[15] Stefan Heyse, "Code-Based Cryptography: Implementing the McEliece Scheme on Reconfigurable Hardware, Diploma Thesis, Ruhr-University Bochum, 31 May 2009
[16] Daniel J. Bernstein et. al, "McBits: Fast Constant-Time Code-Based Cryptography" 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings