THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

POST-QUANTUM CRYPTOGRAPHY

JEFFREY LI, WILLIAM LOUCKS, YI ZHAI, TIM ZHONG

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
77 MASSACHUSETTS AVENUE, CAMBRIDGE, MASSACHUSETTS 02139, UNITED STATES

Abstract. Modern cryptography relies on the intractability of certain problems in mathematics, such as those related to factoring, discrete logarithms, and elliptic curves. Cryptographers have exploited the hardness of such problems to construct the world's most widely used cryptosystems, which have become critical in matters of electronic privacy, banking, national security, and others. These systems, resting on inefficiencies in classical computing, will likely become obsolete in the presence of a machine with superior computing ability. We assess that the introduction of quantum computing will force cryptographers to reexamine the schemes they once thought to be robust against computationally bounded adversaries. In 2017, 69 research groups from around the world submitted proposals to the United States' National Institute of Standards and Technology (NIST), suggesting new quantum resistant public-key infrastructures and signature schemes for standardization. We have examined the quantum resistant public-key systems presented to NIST, grouped them based on security assumption, and recommend the most viable option, considering resistance to known quantum attacks, efficiency, and fast implementation under existing classical architecture.

Contents:

1. Introduction

WE STAND TODAY on the brink of a revolution in computing [1]. In the near future, we assess that the wide use of quantum computers is unlikely, based on a high cost of production and a lack of need for enhanced computing for the general population. However, the use of a quantum computer by an advanced actor, or state, may be constructed by 2026, according to an article published by the *Physical Review* [2]. If one actor were to have a functional quantum computer, using known algorithms which can collapse to solutions for classically hard problems relatively quickly, it may be possible for the actor to acquire sensitive information protected by contemporary means. For instance, JP Morgan, an institution that the National Security Agency deemed critical to national infrastructure after the Manhattan bank was hacked in 2014, spent half a billion dollars on cyber security in 2016 [3,4]. Moreover, JP Morgan uses an RSA product called SecurID as a multi-factor authentication technique for customers to verify their identities before accessing account information. However, this product, and others, rely on the computational hardness of factoring large numbers, a challenge within the purview of a quantum computer. Such a quantum computer would not only compromise the security of institutions keen on protecting their information, such as JP Morgan, but also many other large corporations that are critical to national infrastructure and security.

We must note that classical cryptography can be used to deceive quantum computers, and preemptive implementation may be the best way to protect our expectation of digital security and to combat potential quantum supremacy. In 2017, the National Institute of Standards and Technology (NIST) called for proposals for post-quantum public-key cryptosystems and signature schemes. However, this paper will only consider the almost 50 public-key schemes that were officially accepted as proposals. Nevertheless, one of these schemes will likely be the preferred method of encryption after the introduction of a single quantum actor. When considering AES, NIST discussed the possibility of choosing a backup standard, or selecting multiple candidates. Eventually, the judges only selected one, and hardware designed specifically for AES helped to make it the optimal choice for many practitioners. Similarly, if we expect encrypting information over a public channel in the presence of an advanced adversary to require enhanced space complexity, we likely expect hardware optimization to enter the market, further embedding the standard as the world's most preferred public-key cryptosystem.

We first detail our standards for selecting a platform to be standardized. We describe the criteria that we expect NIST to consider and highlight our approach to examining the proposals. Then we provide an overview of public-key cryptography and key encapsulation abstracted from a specific scheme to the general algorithmic level. After that, we discuss the quantum algorithms that will degrade the security of our current public-key infrastructure, and we present an implementation of Grover's algorithm, a result that may compel practitioners to modify modern AES. We then categorize the NIST public-key proposals into three categories: lattice-based, code-based, and other schemes. We discuss each from a general perspective, noting the underlying security assumptions and highlighting benefits and drawbacks of the general systems. We then assess the individual proposals in each category, issuing a recommendation for the scheme that most adheres to our criteria for standardization.

2. Standardizing a Cryptosystem

In 1997, when NIST called for a new Advanced Encryption Standard (AES) to replace the existing Digital Encryption Standard (DES), NIST itemized three criteria for its selection process: security, cost, and algorithm and implementation characteristics [6]. Candidates began submitting proposals in the Fall of 1997, and in August of 1998, NIST announced 15 finalists for public review. Later, in August of 1999, NIST evaluated the work published by the international community relating to the proposed systems, subsequently announcing five finalists. The five finalists, MARS, RC6, Rijndael, Serpent, and Twofish, were then investigated further during a third round of consideration. Each of the five remaining systems were block ciphers incorporating key schedules, four of the systems involved substitution tables, and three of them incorporated some version of a Feistal structure [6]. NIST likely deemed key scheduling block ciphers to meet the criteria for security and also preferred the ciphers' hardware implementation which allows encryption rounds to be performed efficiently.

While satisfiable implementation is relatively easy to assess, it is probably more difficult to evaluate the relative quality of the systems when there are no known attacks. To make a judgment, NIST examined attacks on systems that are similar to the proposed systems. For instance, some attacks had been published on schemes that simply use less encryption rounds. NIST then considered the idea that attacks for larger rounds usually build on known attacks for protocols that execute fewer rounds [6].

From this, we assess that the best path to recommending a post-quantum public-key scheme is to consider three primary criteria: security against quantum and classical attacks, efficiency, and easy implementation on existing classical hardware. To examine a scheme's resistance to quantum and classical attacks, we referenced publications from the international cryptanalysis community, and compare schemes for which no known attacks exist to schemes that have been broken. We arrived at a judgment which considered how likely the community is able to break the system in the near future, based on the scheme's similarity to others with published attacks.

Second, the public-key system must be efficient. In other words, we don't want the of speed encryption to significantly depend on the latency associated with the computation of a shared secret key. Our aim is to maintain or improve our current level of security against new adversaries without generating new temporal constraints on transmitting data. Even though we have to increase the key size of AES to obtain quantum resistance, Intel produces microprocessors designed to efficiently execute AES for all key sizes, including the desired 256 bit key [7]. As a result, any significant adjustment in temporal expectation for encryption would likely rest in the public-key protocol. Thus, the standardized system must ensure that there is no significant change in the latency involved in data transmission and processing due to encryption.

Lastly, the standardized system must be feasibly implemented on existing hardware. As mentioned, the private key architecture required to rebuff a quantum attack using AES already exists. However, we require that the hardware to implement the new public-key protocol also already exist and be used used in practice. We assess that it is impractical to assume an entire hardware adjustment after the introduction of a quantum actor. Thus, we aim to find a protocol that is quantum resistant under existing technology.

3. Public-Key Cryptography and Key Encapsulation

In 1976, Whitefield Diffie and Martin Hellman published the first method to share a secret over an insecure channel, removing the need for parties to meet in advance and allowing for dynamic alteration of secret keys [1]. The key-exchange protocol allows two parties, each equipped with a public-private key pair, to compute a shared secret that an eavesdropping adversary can not feasibly compute in the average case. This shared secret can be used as a private, symmetric key to further encrypt information. In our context, we assume the channel of communication to be authentic, meaning that an adversary cannot malleate the public key information exchanged between the two parties.

While Diffie-Hellam is only key exchange protocol, not a cryptosystem, modern public-key systems rely on the same intractability of inverting certain functions. For instance, El-Gamal, like Diffie-Hellman, uses the assumption that discrete logarithms are hard to compute and distinguish from randomly generated strings. Moreover, RSA and Goldwasser-Micali take advantage of the hardness assumption for factoring large composites. Later, we will show that both assumptions are no longer reliable under Shor's algorithm.

In a public-key cryptosystem, there are three algorithms: (Gen, Enc, Dec). The algorithms are as follows:

- $Gen(1^\lambda)$: The key generation takes as input the security parameter and produces the public-private key pair (pk, sk) over a designated finite field
- $Enc_{pk}(m)$: The encryption algorithm takes in the public key and a message m in the message space and generates a ciphertext c.
- $Dec_{sk}(c)$: The decryption algorithm takes in the private key and decrypts the ciphertext, yielding the intended plaintext

The secrets shared using public-key systems are then used as private keys in larger schemes such as AES. Public key protocols are computationally expensive, even in the forward direction, and are not as efficient as ciphers with hardware constructed specifically for encryption and decryption. In a post-quantum environment, we may not require drastically different secret key systems, but the assumptions on which public-key schemes rely will lose their security. However, using a quantum resistant public-key protocol in tandem with 256-bit AES, would protect information in the presence of a quantum adversary, by the same standards of security we have today.

To accomplish public-key encryption of a secret key, and encryption of a message using a secret key scheme, we use what is known as a key encapsulation mechanism (KEM). KEM is defined by the three following algorithms $(Gen, Encaps, Decaps)$ [8]:

- $Gen(1^\lambda)$: The key generation algorithm takes in the security parameter λ and outputs the public-private key pair (pk, sk), where sk is the private (secret) key.
- $Encaps_{pk}(1^\lambda)$: The key encapsulation algorithm takes in the public key and security parameter and generates a ciphertext c and key k
- $Decaps_{sk}(c)$: The decapsulation algorithm takes in the private key and ciphertext and outputs the key k or $\perp$ indicating a mistake

The scheme is correct if Decaps outputs the desired k. Now, the hybrid scheme, combining public and private cryptography through key encapsulation, $(Gen^{hy}, Enc^{hy}, Dec^{hy})$ is as follows [8]:

- Gen^{hy} : Run $Gen(1^\lambda)$ for security parameter λ and output the public-private key pair (pk, sk)
- Enc^{hy} : Using the public key pk and a message m in the message space:
    1. Compute $(c, k) \leftarrow Encaps_{pk}(1^\lambda)$
    2. Compute $c' \leftarrow Enc_k(m)$
    3. Output ciphertext (c, c')
- Dec^{hy} : Using the private key sk and ciphertext (c, c'):
    1. Compute $k = Decaps_{sk}(c)$
    2. Output $m = Dec_k(c')$

The result is a cryptosystem whose security reduces to the security of the public-key system which comprises it, and a system that is as efficient as the secret key cryptosystem. Note the efficiency of the hybrid system per bit of plaintext approaches the efficiency of Enc as the size of the message increases. In the future, NIST likely plans to implement a KEM that consists of one of the public-key systems proposed. We must now consider the algorithms which necessitate this new approach to security.

4. Quantum Algorithms

While quantum algorithms can be used to speedup computation for a suite of NP problems, we will focus on two relevant algorithms, namely Grover's search algorithm and Shor's factoring algorithm. Moreover, while there isn't a consistent reduction in run-time for every problem that is executed through quantum computation, relative to classical computing, the use of quantum gates allows for a new paradigm of expected run-time. An actor in possession of a quantum computer will be able to collapse on solutions to classically hard problems more efficiently than an actor wielding a classical computer. As a result, theoretical computer scientists and cryptographers have been attempting to design circuits that can hone in on an aspect of a difficult problem and converge on solutions that make certain cryptosystems obsolete. We choose to cover the two aforementioned algorithms due to their relevance in breaking existing cryptosystems and to provide insight into NIST's decision to host a contest for a standardized public-key protocol.

Search algorithms are used in a range of problems where one wants to find the correct solution among an exponential number of possibilities. For instance, to factorize a composite number N, one can brute force search from 1 to $\sqrt{N}$ to find one of the factors. While there may exist more clever algorithms, which we will see with Shor's, searching provides one avenue to a solution.

Suppose, for instance, that there is a function f, such that f(x)=1 when x=x_0 and f(x)=0 o.w. The classical search method would be to evaluate f at random point and the worst case would be to try N-1 points before the correct answer pops up. Grover's algorithm creates a superposition state for each qubit. Then every possible input in the range is represented in the circuit. Then

the algorithm uses an oracle circuit to flip the amplitude of the state that represents x_0. Then we use a Grover diffusion Operator to flip every amplitude around the mean. Then all the other amplitude remain the same except that the amplitude for x_0 is increased by $\frac{2}{\sqrt{N}}$. We can apply this amplification several times and eventually the amplitude of x_0 will approach 1. Then we do the measurement and we can find x_0 with almost 100% probability. The number of times that we need to do the amplification is on the order of $O(\sqrt{N})$

Intuitively, Grover's algorithm is executing a superposition of computations where certain paths of computation interfere with one another, effectively canceling one another out and reducing the time it takes the algorithm to collapse on a solution. Proceeding with the computation too quickly sacrifices precision, but proceeding too slowly sacrifices speed. As a result, quadratic speedup seems to be the optimal zone for fast reliable computation. Another way to understand the runtime is to consider the relationship between quantum amplitudes and probabilities. If each possibility has a uniform chance, $1/N$, to be the correct answer, each possibility has amplitude $1/\sqrt{N}$. Thus, we only need $\sqrt{N}$ steps to obtain the correct solution with high probability. Note that because Grover's algorithm provides quadratic speedup, to achieve approximately the same security as 128-bit AES, we must use 256-bit AES. This follows since a classical adversary will take at most 2^{128} queries to find the secret key in 128-bit AES, but a quantum adversary using Grover's algorithm will asymptotically only require 2^{64} queries. Thus, to achieve $O(2^{128})$ security using AES, we likely will have to incorporate a 256-bit key.

Shor's algorithm, on the other hand, exhibits an exponential decrease in runtime, relative to the best classical algorithms for factoring. Shor's algorithm takes advantage of a series of reductions. First, factoring is reducible to order finding. In other words, if we're attempting to factor composite N, if we can find the order of an integer a, where $a \in \{1, ..., N-2\}$ and $GCD(a, N) = 1$, in polynomial time, we can factor N in polynomial time. Assuming there exists an algorithm to compute the order of an integer in polynomial time, our reduction algorithm could first sample an integer $a \leftarrow \{2, ..., N-2\}$ and then test if $GCD(a, N) = 1$ through the Extended Euclidean Algorithm. Note that if $GCD(a, N) \neq 1$, we have found a factor of N. Otherwise, we run the algorithm to compute the order of a as a subroutine and observe that the order of N will be even with probability $\frac{1}{2}$. If a is even, let $b = a^{r/2} \bmod N$, which implies $b^2 - 1 = 0 \bmod N$. As a result, N must divide $(b+1)(b-1)$, so we test if $GCD(a^{r/2} - 1 \bmod N, N)$ is a nontrivial factor of N. Since only a constant number of a values will need to be sampled, the overall runtime of the algorithm is polynomial time, due to the algorithm which finds the order. At its heart, Shor's algorithm uses a quantum approach to solve the order finding program for an integer that is relatively prime to the composite, allowing one to factor the composite [9].

Grover's algorithm will change how we think about implementing secret key cryptosystems, and Shor's algorithm will largely eliminate the public-key cryptosystems in use today. Since the discrete logarithm problem and and the quadratic residuosity assumption can be reduced to solving the factoring problem, most public-key cryptosystems and some commitment schemes, hash functions, and other cryptographic devices will be rendered obsolete. The harness assumptions we could once make about discrete logarithms, factoring, and elliptic curves no longer apply. The solution is probably a bolstered secret key infrastructure, such as 256-bit AES, coupled with a novel public-key protocol that is resistant to Shor's algorithm. To demonstrate the capabilities of the aformentioned algorithms, we implemented a small version of Grover's.

5. Implementation of Grover's algorithm

We use IBM Q to implement the Grover's algorithm. We will illustrate implementations which use two and three quibits. Note that Figure 2 shows different Oracle circuits used and Figure 3 shows the results for different Oracles.

5.1. 2 qubits. Figure 1 shows the implementation of 2 Qubits circuit.
The circuit is devided into four parts.

(1) Create Quantum Superposition
(2) Oracle Circuit

(3) Grover's diffusion operator
(4) measurement

Observe that step 2 and 3 will be sufficiently repeated to have a satisfying result.

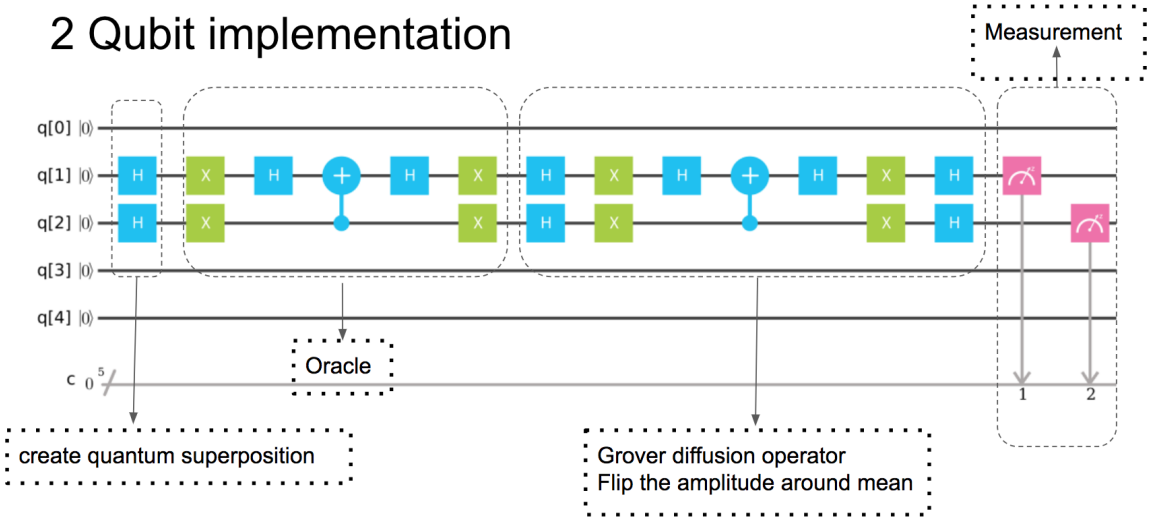
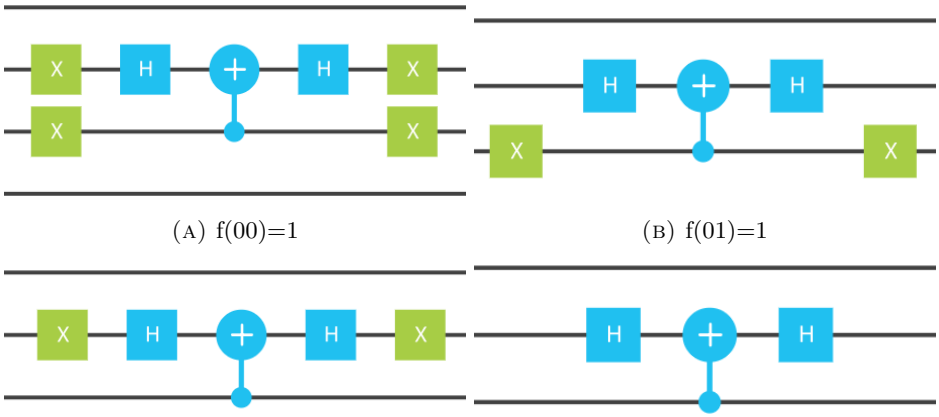


FIGURE 1. 2 Qubit circuit

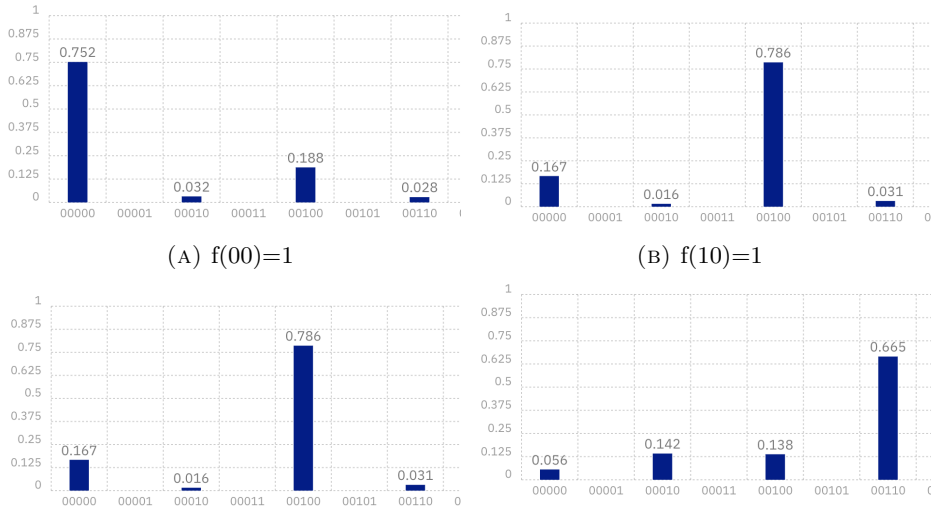


(A) f(00)=1 (B) f(01)=1

(C) f(10)=1 (D) f(11)=1

FIGURE 2. Different Oracle Circuit

5.2. 3 qubits. Figure 4 shows the results after 1 amplification. The probability of getting the right answer is very high. Figure 5 shows the results after 2 amplification, the correctness is even further boosted to almost 1. Figure 6 shows the results after 3 amplifications. However the correctness dropped. Because in the Grover's diffusion operator, the mean becomes negative and it drags down the amplitude of the target basis instead of increasing it.



(A) f(00)=1 (B) f(10)=1

(C) f(10)=1 (D) f(11)=1

FIGURE 3. Results for Different Oracle

Results for 3 Qubits, 1 Amplification

	000	001	010	011	100	101	110	111
000	817	32	26	34	28	34	25	28
001	31	802	34	33	27	34	31	32
010	32	27	787	40	30	35	31	42
011	17	39	35	793	38	27	31	44
100	40	32	33	31	805	30	28	25
101	31	30	31	29	32	813	21	37
110	26	37	29	24	24	26	834	24
111	27	34	31	31	36	33	27	805

FIGURE 4. 3 Qubits 1 amplification

Results for 3 Qubits, 2 Amplifications

	000	001	010	011	100	101	110	111
000	961	8	6	5	15	8	9	12
001	8	953	12	14	6	8	10	13
010	7	7	966	8	10	11	9	6
011	7	5	8	963	14	7	12	8
100	9	8	13	6	964	10	4	10
101	5	7	9	13	3	972	7	8
110	6	12	5	13	8	4	967	9
111	12	8	5	12	8	11	8	960

FIGURE 5. 3 Qubits 2 amplifications

Results for 3 Qubits, 3 Amplifications

	000	001	010	011	100	101	110	111
000	343	99	88	106	107	98	89	94
001	103	345	84	96	84	97	106	109
010	90	93	332	106	89	97	122	95
011	93	85	112	330	82	113	105	104
100	104	84	103	106	361	94	86	86
101	90	93	103	112	98	331	98	99
110	92	105	107	86	113	85	337	99
111	103	103	113	80	98	92	97	338

FIGURE 6. 3 Qubits 3 amplifications

6. Post-Quantum Cryptography

As mentioned, post-quantum public key protocols must be resistant to Shor's algorithm, as well as other classical and quantum attacks, and the research groups who submitted proposals hope to have invented such schemes. We have isolated the suggested public-key infrastructures into three groups: lattice-based schemes, code-based schemes, and other schemes [10]. We provide an overview of each category, and assess how well each approach caters to our criteria of quantum resistance,

efficiency, and fast implementation under hardware already in use. In the end, we present the proposal in each category that we assess to most effectively meet the mentioned criteria.

6.1. Lattice-Based Cryptography.

6.1.1. *Introduction.* Lattice-based cryptography refers to cryptographic primitives that involve lattices either in their construction or in their security proof. The integer constraints leave problems that have been analyzed and shown to be hard, which can form the basis of a scheme that is difficult to crack (similar to how the hardness of factoring a number is the basis of RSA). The first such scheme was first introduced in 1996 by Miklos Atjai and is based on the assumed hardness of known lattice-based problems–in this case on the short integer solutions problem, the task of finding the shortest nontrivial solution to $Ax = 0$ given A, a matrix of n-dimensional vectors. A similar problem is the shortest vector problem, where given a lattice basis of n-dimensional vectors, the goal is to find the shortest vector contained in that basis. When n is 2, the problem is trivial–however, when n is large, this problem becomes hard to solve. Other problems include the closest vector problem, bounded distance decoding and covering radius problem. These problems (and others just like it) now are being considered crucial candidates for post-quantum cryptography standardization because there is no algorithm (classic or quantum) that solves these problems in polynomial time. [11] This characteristic has led many people, including IBM research, to state that lattice-based schemes are the future of post-quantum cryptography and that it will change the world. [12]

There are several quantum-resistant lattice-based schemes that currently exist, including NTRU, GGH, and schemes based off of LWE. The NTRU scheme was developed in 1996 and features an encryption scheme whose hardness stems from the shortest vector problem. [13] The Goldreich-Goldwasser-Micali (GGM) scheme was developed in 1997 and bases the hardness of its encryption algorithm on the closest vector problem. [14] In 2005, Oded Regev introduced the Learning with Errors (LWE) problem–the problem of taking many samples of (x, y) and trying to create a function f such that $y = f(x)$. What's special is that in this paper he also showed that the LWE problem was as hard to solve as several hard lattice problems. [15] Thus, it has been used as a hardness assumption to create several encryption schemes. It has inspired the creation of several variants, including Ring-LWE and Module-LWE, which still hold many of the basic characteristics of the LWE problem. The impact of the LWE problem have been so important that the paper won the 2018 Godel award, awarded each year to one paper for outstanding contributions in theoretical computer science.

The variants of LWE have only come around fairly recently, with Ring-LWE being described in 2012 by Oded Regev (the same one who first introduced LWE) and Chris Peikert. [16] One of the biggest aspects of Ring-LWE is that its nature allows it to achieve the same security as other LWE based schemes with a smaller key size. A Ring-LWE scheme could achieve the same security as an LWE scheme with key size of length n with only a key size of $\sqrt{n}$. Module LWE is another variant that was described later in 2012 by Langlois and Stehle and was described as more generally secure than Ring-LWE for working solely with module lattices. [17] These variants have served as the crutch for many of the submissions to NIST's post-quantum cryptography standard–many of the proposals attempt to extend on these variants in their own subtle ways. Some of the reasons for this is that they are simple, strongly secure, and offer relatively efficient performance with their implementation.

Although lattice-based crypto-schemes have the potential to be secure from quantum computing, they still require several conditions be met by the lattice. Thus, it is still possible for lattice-based schemes to be susceptible to certain attacks. The most common attack is the Lenstra-Lenstra-Lovasz (LLL) lattice reduction attack–a polynomial time algorithm where the n-dimensional linearly independent vectors that cover a lattice basis are replaced with another group of vectors (which cover the same basis) that are orthogonal and smaller in magnitude (and are thus easier to work with to solve these hard problems). [18] In fact, NTRU was shown to be vulnerable to LLL attacks, and had to be modified in order to accommodate this. In general, most schemes are versatile enough to be able to shift parameters/lattice conditions to make them more secure against attacks like LLL. However, the existence of such attacks has led to the breaking of several schemes, including several submissions for NIST's Post Quantum Cryptography standard, such as RVB and FrodoKEM.

One aspect of LWE that's also problematic is the use of error/noise in the schemes based off of it. This adds a factor of unreliability to the act of decryption, and has opened up research into reconciliation methods for dealing with this noise. Most recently, Chris Peikert developed a low-bandwith reconciliation technique that allows two parties who "approximately agree" on a secret value to reach an *exact* agreement. This has proven easy to augment onto these schemes, and also has an added benefit of allowing the scheme to use a smaller ciphertext size. [19]

6.1.2. *Accordance to Selection Criteria.* Next, we will consider Lattice-based schemes in the context of the NIST Selection Criteria used in standardizing AES: Security, Implementability, and Performance/Cost. Schemes based on lattice problems are still safe from quantum attacks due to the verified hardness of such problems. These problems have also been analyzed and studied enough, so the hardness of such problems will not be in question for the near future. The aspects that then impact security are parameters within the scheme itself (such as key size), as well as specification for the lattice basis. These can be seen as weaknesses in implementation, as the scheme requires special cases of lattice basis to be used. However, when controlled, this is an issue that is easy to circumvent.

So far, the main issue with security of lattice-based implementations lies in security against lattice reduction attacks (such as LLL). This has already been shown to break some implementations. However, as mentioned before, measures can be taken to prevent that from being a problem, such as increasing the dimensionality of the basis, or increasing the key size.

We see several schemes moving towards Ring-LWE because they "...offer some of the best performance and key size characteristics among quantum-resistant candidates". [20] These schemes have also proven to be relatively easy to implement, as most of the hardware/implementation footprint stems from the ring arithmetic. In fact, with **New Hope**, a scheme that is based on Ring-LWE with a reconciliation method [21], a basic implementation can be done with only a few lines of code on a tool like SageMath.

6.1.3. *Recommendation.* Within the first round of proposals, many of the lattice-based schemes belong to the Ring-LWE or Module LWE family. As mentioned before, this mere fact alone speaks to the prominence that LWE and its variants have taken on, as well as the potential that many see in lattice-based cryptography. As discussed before, both are fairly secure, with Module LWE benefitting from more generalization of lattice characteristics, while Ring-LWE boasts better key size optimizations. Ring-LWE has been shown to be strongly secure if several conditions hold true on the lattice (this, however, may change with further cryptanalysis on Ring-LWE). Since these conditions are easy to verify, the benefit of more generalization to lattice characteristics is not crucial. Thus, between these two variants, the smaller key size of Ring-LWE is a feature that will make implementation simpler.

Within Ring-LWE schemes, the issue of unreliability of decryption that springs up with the use of error/noise in encryption is an inconvenience. Thus, introducing a reconciliation method to the scheme, such as Peikert's reconciliation, will allow the two parties who approximately agree to reach an exact agreement. This is an important implementation characteristic that we see as an important added benefit to any scheme.

Thus, within the Lattice-based cryptography family, we propose a Ring-LWE scheme with a reconciliation method such as Peikert's reconciliation as the best type of scheme with respect to the NIST selection criteria. Schemes within the first round of NIST submissions that satisfy these conditions are **New Hope** and **HILA-5**.

6.2. Code-Based Cryptography.

6.2.1. *Introduction.* Beyond Lattice-based schemes, a majority of the NIST proposals for post-quantum cryptography are Code-based schemes for public key encapsulation. Code-based schemes are fundamentally based on the McEliece Cryptosystem framework, an asymmetric encryption algorithm introduced in 1978 by Robert McEliece [22]. At the time, the McEliece scheme was quite novel in introducing randomization in the encryption process, but never gained wide acceptance in the cryptographic community due to cost issues, and was instead overshadowed by the RSA Cryptosystem. However, the McEliece Cryptosystem is more relevant in the context of post-quantum

cryptography than number-theoretic schemes such as those based on the RSA cryptosystem, El Gamal or Elliptic Curves because it is immune to attacks using Shor's algorithm–the aforementioned polynomial-time quantum integer factorization algorithm proposed in 1994.

The McEliece Cryptosystem framework is based on the known NP-hard problem of decoding a general linear code. The definition of a linear code of length n and rank k is a linear subspace C with dimension k of the vector space $\mathbb{F}_q^n$ where $\mathbb{F}_q$ is the finite field with q elements. In the case that $q = 2$, the code is known as a binary code. McEliece's original algorithm uses binary Goppa codes, which provide the advantages of having a fast polynomial time decoding algorithm, and being easy to generate but difficult to find [23].

There are several reasons why Goppa codes are the primary choice for the McEliece cryptosystem: Firstly, Goppa codes have a fast polynomial time decoding algorithm due to an efficient algorithm by Patterson [24]. Secondly, the reason Goppa codes are considered "easy to generate but hard to find" is that any irreducible polynomial over a finite field $\mathbb{F}_{2^m}$ can be used to create a Goppa code, but the generator matrices of Goppa codes are nearly random [25]. In this scheme, the public key is derived from the private key by disguising the selected code as a general linear code.

There are several variants of this McEeliece-based scheme that use different types of codes to replace the binary Goppa codes used in the original. However, most of them were proven less secure than McEliece with binary Goppa codes. So far, the McEliece Crypostytem with binary Goppa codes has resisted cryptanalysis.

In fact, the original version of the McEliece Cryptosystem, based on binary Goppa codes with irreducible generator polynomials, is actually faster than the widespread RSA Cryptosystem. However, it has two major drawbacks: large keys and low transmission rate, the latter being coincident with the code rate. The McEliece cryptosystem uses generator matrices and encodes the messages into codewords of the public code [24].

Closely related to the McEliece Cryptosystem is a variant called the Niederreiter Cryptosystem. The basic idea of the Niederreiter cryptosystem is to replace the generator matrix G with the parity-check matrix H. "A substantial difference between the McEliece and the Niederreiter cryptosystems is that the latter requires shorter public keys" [25]. It has been shown that Niederreiter and McEliece cryptosystems offer the same level of security when used with binary Goppa codes.

6.2.2. *Accordance to Selection Criteria.* Next, we will frame Code-based schemes in the context of the NIST Selection Criteria used in standardizing AES: Security, Implementability, and Performance/Cost. The McEliece cryptosystem with binary Goppa codes is still secure against quantum-computers and their ability to use Shor's algorithm to factor integers in polynomial-time. "Despite many attack attempts, the McEliece cryptosystem is still unbroken, in the sense that no algorithm able to realize a total break in polynomial time has been presented up to now" [25]. However, in practice, one of the main disadvantages of McEliece is that the private and public keys are large matrices. For a standard selection of parameters, the public key is 512 kilobits long. This is why the algorithm is rarely used in practice. Thus, the McEliece cryptosystem scores poorly in regards to implementability as most computers and machines will most likely be unable or unwilling to dedicated the resources necessary to support McEliece with binary Goppa codes. Hence, without significant hardware improvements across all devices in the near future, the cost of doing so is too great to move forward with the original McEliece Cryptosystem as the post-quantum cryptographic standard.

However, the most effective way to overcome the drawbacks of the McEliece cryptosystem is to replace Goppa codes with other families of codes, yielding a more compact representation of their characteristic matrices, and permitting to increase the code rate. Unfortunately, although several families of codes with such characteristics exist, it is very difficult to replace Goppa codes with other codes without incurring into serious security flaws, as occurred, for example, with Gabidulin codes and GRS subcodes [24].

We see this in all of the code-based schemes proposed to NIST as they make efforts to optimize the implementability and hence performance and cost of the McEliece cryptosystem. Some proposals such as **Classic McEliece** and **Bike-2** move towards a Niederreiter cryptosystem, which proposes a code-based cryptosystem using the parity-check matrix and Goppa codes. The main

advantage of Niederreiter's variant, which encodes the messages into syndrome vectors, is to achieve a significant reduction in the number of operations for encryption, though this is paid with a moderate increase in the number of operations for decryption. Others attempt to instead replace binary Goppa codes with other types of codes, which can be divided between Algebraic Codes and Sparse Matrices. For example, **Dags** proposes using algebraic Quasi-Dyadic Generalized Srivastava Codes, while **Big Quake** uses Quasi-Cyclic Goppa codes, and **RLCE-KEM** uses Random Linear Codes. Additionally, **MCNIE** uses Quasi-Cyclic Low Rank Parity Check, while **Bike-1** and **QC-MDPC KEM** use Quasi-Cyclic Medium Density Parity Check. All of these papers claim to achieve performance increases over the original McEliece scheme with binary Goppa codes, and have yet to be formally broken via cryptanalysis attacks. However, it is unsafe to say whether these schemes are truly secure and reasonable in post-quantum cryptography as related schemes are steadily being successfully attacked. For example, **EDON-K** proposed a scheme that was inspired by the McEliece scheme, but is based on a family of codes defined over $\mathbb{F}_{2^{128}}$ instead of $\mathbb{F}_2$, and is not based on the Hamming metric–two optimizations that allow for significantly shorter public keys than the original McEliece scheme with binary Goppa codes did. However, a polynomial-time attack to reconstruct the encapsulated secret was published and the authors of **EDON-K** have since withdrawn their proposal.

6.2.3. *Recommendation.* Among the most recent proposals, Quasi-Cyclic (QC) [15], Quasi-Dyadic (QD) [26] and Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) codes [27] have been considered for possible inclusion in the McEliece cryptosystem. However, the QC and QD solutions have recently been successfully attacked [24]. As a result, updated variants of the QD algorithm were proposed that should be more secure; however, due to the complexity of the schemes and the subsequent complexity of the attacks, it is unclear weather there exists an attack for the binary QD cases yet. However, what can be seen is attacks on the aforementioned Algebraic division of codes exploit an "algebraic approach, based on a system of bi-homogeneous polynomial equations, which holds for the whole class of alternate codes. Hence, such attack concerns all cryptosystems using codes in this family" [24].

Turning away from Algebraic Codes and to Sparse Matricies, we find that LDPC (Low Density Parity Check) codes are "state-of-art error correcting codes, first introduced by Gallager in the sixties, and more recently rediscovered. While random-based LDPC codes are able to approach the channel capacity, structured LDPC codes have the advantage of an easier implementation of the encoding and decoding operations, and benefit from reduced storage requirements" [28]. This analysis supports LDPC codes in McEliece Cryptosystems as a more implementable cryptosystem with fewer resource costs. "QC-LDPC codes are one of the most important examples of structured LDPC codes, and they have also been proved to achieve very good performance. The existence of efficient iterative decoding algorithms for LDPC codes is the distinguishing feature of this class of codes. The rationale of these algorithms is an iterated updating and exchange of messages along a bipartite graph, also known as Tanner graph, which represents the code parity-check matrix. Very good decoding performance is achieved as long as the code Tanner graph is free of short cycles, that is, closed loops starting and ending at one node" [24]. Thus, we found that QC-LDPC codes to be a promising avenue to construct a more implementable and low-cost Code-based Cryptosystem than the original McEliece Cryptosystem with binary Goppa codes.

Hence, within the family of Code-based schemes for post-quantum cryptography, we propose QC-LDPC schemes as the best type of scheme with respect to the NIST selection criteria. Schemes proposed to NIST that employe QC-LDPC codes include **MCNIE** and **Big Quake**.

6.3. Other Schemes Proposed.

6.3.1. *Introduction.* While many of the public-key schemes proposed to NIST rely on the same assumptions, there are a few schemes that do not cleanly fall into the two mentioned categories. While there are advantages to using schemes that rely on well studied assumptions, the security of the following schemes may also be robust against quantum actors and we examine how they compare to the systems above.

6.3.2. *Accordance to Selection Criteria.* Two proposals rely on the assumed hardness of solving nonlinear multivariate equations, while the other three were fairly distinct. One of the multivariate schemes was **CFPKM**, whose security assumption relied on computing solutions to noisy nonlinear polynomials. The encryption algorithm adds a small error to each equation in the system, similar to LWE, but to nonlinear equations. Moreover, if an adversary were to efficiently solve the system of equations, he could use his algorithm to solve the Decision Posso with Noise problem, a problem that is NP-hard. While the security of the system appears to be roughly equivalent to other schemes related to LWE, an 81 byte secret only requires a 729 byte ciphertext. Moving forward, security scales with the number of equations to solve, and efficiency scales inversely to the number of equations, so a middle-ground must be achieved [29]. In the end, at this time, we are unable to find a notable difference between this scheme and many lattice schemes, outside of a slight improvement on efficiency. There is no indication that hardware will be able to augment the scheme.

The other proposed scheme that incorporates multivariate equations is **Giophantus**. **Giophantus** also involves nonlinear equations, but the primary security assumption is that it is difficult to solve a system of indeterminate equations. In other words, since there are more equations than variables publicly available, it is difficult to guess the true values of the unknown variables. Initially the scheme likely did not include the previously mentioned noise, but after an attack was proposed, the researchers added a noise component to recreate security. As a result, the scheme is secure under a variant of the LWE assumption called the Indeterminate Equation Learning with Errors (IE-LWE) assumption, which essentially states that the small-coefficient polynomials generated in the scheme are indistinguishable from random, noisy polynomials. We must also not that this scheme has an additive and multiplicative homomorphism, so one can compute on secret information. However, the scheme does little to address efficiency and fast implementation concerns [30].

The **Guess Again** scheme used a novel approach where the sender of a bit attempts to guess the decryption key of the receiver. Assuming we are transmitting one bit, this requires that an adversary not guess what the bit that will be transmitted, but attempt to identify what bit the sender wants to transmit. A consequence of protocol is the possibility for errors during decryption. One way to avoid the errors is to amplify the number of rounds of transmission, but there is an efficiency tradeoff, which is made worse because we only transmit one bit at a time. First, we note that the security of the scheme considers a computationally unbounded adversary model. Moreover, the authors illustrate that the scheme is IND-CCA2 secure since the view of the receiver and an eavesdropper are the same, and an adversary who can adaptively choose plaintexts to encrypt obtains negligible advantage. The difference is in what the two parties are looking for, and the differences in message space accounts for the higher than uniform likelihood that the receiver will obtain the correct bit over a number of rounds. As a result, an eavesdropper has probability 1/2 of guessing the correct bit, and the authors showed that the receiver has a 0.55 chance of obtaining the correct bit. Amplified over 1000 rounds, the receiver obtains the wrong bit with negligible probability. However, considering cost and efficiency, a one bit message can corresponds to 18,000 bits of cipertext. The researchers were able to reduce some of the latency by prepossessing some of the computations and storing it with the parties (only about 15MB). In the end, it takes about 16 ms to encrypt one bit on a standard PC. While there is nothing else quite like the scheme among the batch of proposals, the space complexity might make it infeasible for encryption of large key without some type of modification or optimization [31].

A scheme titled **Mersenne-756839** involved a combination of Mersenne primes and Hamming Weight. Mersenne primes were chosen because working in a field modulo a Mersenne prime does not change the Hamming weight of a string. The scheme relies on computing linear equations while using strings with low Hamming weight. The security assumption is similar to that of LWE in that we assume two tuples are indistinguishable, but one contains the product of a string with another string of low hamming weight which is then added to another string. The security of the scheme thus rests on the Hamming weight, in fact the authors suggest that attacks likely take $O(2^h)$ computations, where h is the Hamming weight. A very similar scheme was broken for encrypting multiple bits. However, this scheme is not computationally prohibitive, and doesn't require the space that many others require [32].

The last scheme we mention in is the **SIKE** proposal, which uses a special type of elliptic curve called a Montgomery curve. The security of the scheme relies on the hardness of computing isogenies between fields. The researchers describe an isogeny as a non-constant map between elliptic curves that are homomorphic to one another over a finite field. Notably, the scheme already has existing hardware optimizations, avoids assumptions that can be attacked with Shor's algorithm, and the authors provide a detailed analysis of the number of quantum gates that would be required to break the scheme, a number that can be modified depending on the sophistication of quantum search algorithms. As a whole, the authors did a particularly careful job to explain the foundation of the security assumption and present possible attacks. Moreover, this scheme already has existing hardware that can be used to increase the efficiency of its performance in practice. In addition, security practitioners have experience implementing elliptic curve schemes, which may alleviate worries that quantum resistant schemes which haven't yet been reduced to practice may be riddled with side channel attacks [33].

6.3.3. *Recommendation.* Outside of lattice-based schemes and code-based schemes, we assess that the most well polished scheme was the **SIKE** proposal. Submitted by 14 authors in four different countries, the security of **SIKE** relies on the isogeny walk problem, stating that appears to be hard to find a path of isogenies of small degrees from one curve to another. The known quantum attacks rely on search algorithms, such as Grover's, but they require a large number of gates with respect to the key size. Moreover, practitioners with experience in elliptic curve cryptography may find **SIKE** more easy to implement, relative to other schemes which will require a new approach in practice. This experience, with more sound implementation, may also translate to a decrease in possible side channel attacks. Assuming that **SIKE** is as quantum resistant as any scheme presented, **SIKE** allows for the highest amount of hardware support. The researchers who developed **SIKE** demonstrated their scheme on Intel x64 microprocessors, which likely protects against timing and cache attacks. Moreover, the hardware is optimized for encryption, and is one of the most efficient proposals submitted. In the end, **SIKE**, relative to the other proposals, highly satisfies our criteria of efficiency and implementation under classical hardware, and its security against quantum actors has yet to be robustly demonstrated. None-the-less, we assess that of the schemes that are neither lattice-based nor code-based, it is the most qualified protocol to be standardized.

7. Conclusion

We assess that the National Institute of Standards and Technology will issue a standardized public-key system in the near future. The judges are likely to consider the proposed schemes' resistance to quantum adversaries, as well as classical security that concerns cache, timing, side-channel, and other types of attacks. We recommend that the criteria for consideration not only concern security, but also efficiency and fast implementation under existing hardware. We estimate that it would be financially prohibitive for everyone to purchase devices to secure information, or new computing devices entirely to reach an adequate level of security. To sufficiently meet the needs of users, the scheme must also have a low latency, and we hope that the scheme will leave open the possibility for optimized hardware to bolster efficiency, if such hardware doesn't already exist. In the end, we assess that much more research and investigation into the schemes is necessary to deliver a conclusion on which proposal to standardize. However, we see large potential in the **SIKE** scheme, which meets our presented criteria more than any of the other schemes. In the future, the public is likely to be receptive to unnoticed changes that improve computer and network security, and such a post quantum key encapsulation method achieves just that.

References

[1] https://ee.stanford.edu/ hellman/publications/24.pdf

[2] https://journals.aps.org/prx/pdf/10.1103/PhysRevX.6.031045

[3] https://dealbook.nytimes.com/2014/12/22/entry-point-of-jpmorgan-data-breach-is-identified/

[4] https://www.forbes.com/sites/stevemorgan/2016/01/30/why-j-p-morgan-chase-co-is-spending-a-half-billion-dollars-on-cybersecurity/3cf9c45f2599

[5] https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions

[6] https://nvlpubs.nist.gov/nistpubs/jres/106/3/j63nec.pdf

[7] https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni

[8] J. Katz, Y. Lindell. *Introduction to Modern Cryptography.* Chapman and Hall/CRC. 2015.

[9] P. Kaye, R. Laflamme, M. Mosca. *An Introduction to Quantum Computing.* Oxford University Press. 2010.

[10] https://en.wikipedia.org/wiki/Post-Quantum_Cryptography_Standardization

[11] https://eprint.iacr.org/2015/938.pdf

[12] https://www.research.ibm.com/5-in-5/lattice-cryptography/

[13] https://assets.onboardsecurity.com/static/downloads/NTRU/resources/NTRUTech006.pdf

[14] https://www.math.auckland.ac.nz/ sgal018/crypto-book/ch19a.pdf

[15] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6):34, 2009. Preliminary version in STOC'05.

[16] https://eprint.iacr.org/2012/230.pdf

[17] https://eprint.iacr.org/2012/090.pdf

[18] https://math.mit.edu/ apost/courses/18.204-2016/18.204-Xinyue-Deng-final-paper.pdf

[19] https://web.eecs.umich.edu/ cpeikert/pubs/suite.pdf

[20] M. Saarinen, HILA5 Key Encapsulation Mechanism (KEM) and Public Key Encryption Algorithm"

[21] https://newhopecrypto.org/data/NewHope-2017-12-21.pdf

[22] http://www.math.unl.edu/ s-jeverso2/McElieceProject.pdf

[23] P. Loidreau. Strengthening McEliece Cryptosystem, Springer-Verlag Berlin Heidelberg, 2000.

[24] M. Baldi, Security and complexity of the McEliece cryptosystem based on QC-LDPC codes. Dipartimento di Ingegneria dell'Informanzione, Università Politecnica delle Marche.

[25] The McEliece and Niederreiter Cryptosystems - https://link.springer.com/content/pdf/10.10072F978-3-319-02556-8-5.pdf

[26] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key length of the McEliece cryptosystem," in Progress in Cryptology - AFRICACRYPT 2009, ser. Lecture Notes in Computer Science. Springer Verlag, 2009, vol. 5580, pp. 77–97.

[27] R. Misoczki and P. S. L. M. Barreto, “Compact McEliece keys from Goppa codes,” in Selected Areas in Cryptography, ser. Lecture Notes in Computer Science. Springer Verlag, 2009, vol. 5867, pp. 376–392.

[28] M. Baldi, LDPC codes in the McEliece cryptosystem: attacks and countermeasures, ser. NATO Science for Peace and Security Series - D: Information and Communication Security. IOS Press, 2009, vol. 23, pp. 160–174.

[29] O. Chakraborty, J.C. Faugere, L. Perret. ”CFPKM : A Key Encapsulation Mechanism based on Solving System of non-linear multivariate Polynomials”. Universite Pierre et Marie Curie.

[30] K. Akiyama. ”Indeterminate Equation Public-key Cryptosystem (Giophantus)”. Toshiba Corporation. 2017.

[31] V. Shpilrain. ”Guess Again: Unconditionally Secure Public-Key Encryption (With Possible Decryption Errors)”. The City College of New York.

[32] D. Aggarwal, A. Joux, A. Prakash, M. Santha. ”A New Public-Key Cryptosystem via Mersenne Numbers”. 2017.

[33] D. Jao. ”Supersingular Isogeny Key Encapsulation”. University of Waterloo. 2017.