International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-12 Issue-5, June 2023

Quantum Computers and Algorithms: A Threat to Classical Cryptographic Systems

Joshua J. Tom, Nlerum P. Anebo, Bukola A. Onyekwelu, Adigwe Wilfred, Richard E. Eyo

***Abstract*: Contemporary cryptographic algorithms are resistant to the strongest threats to cybersecurity and high-profile cyber-attacks. In recent times, information security scientists and researchers had developed various cryptographic schemes that defeated attacks using the most sophisticated (in terms of processor speed) classical computer. However, this resistance will soon erode with the arrival of quantum computers. In this paper, we profiled quantum computers and quantum algorithms based on their widely believed threat against currently secure cryptographic primitives. We found that Grover's and Shor's quantum-based algorithms actually pose a threat to the continued security of symmetric cryptosystems (e.g. 128-bit AES) and asymmetric (public key) cryptosystems (e.g. RSA, Elgamal, elliptic curve Diffie Hellman (ECDH), etc.) respectively. We discovered that the source of the algorithms' cryptanalytic power against the current systems, stems from the fact that they (Grover and Shor) both equipped their respective algorithms with a quantum circuit component that can execute the oracle in parallel by applying a single circuit to all possible states of an n-qubit input. With this exponential level of processing characteristic of quantum computers and quantum-based algorithms, it is easy for the current cryptosystems to be broken since the algorithms can existentially solve the underlying mathematical problems such as integer factorization, discrete logarithm problem and elliptic curve problem, which formed the basis of the security of the affected cryptosystems. Based on this realization and as part of our readiness for a post quantum era, we explored other mathematical structures (lattices, hashes, codes, isogenies, high entropy-based symmetric key resistance, and multivariate quadratic problems) whose hardness could surpass the cryptanalytic nightmare posed by quantum computers and quantum-based algorithms. Our contribution is that, based on the findings of this research work, we can confidently assert that all hope is not lost for organizations heavily relying on protocols and applications like HTTPS, TLS, PGP, Bitcoin, etc., which derived their security from the endangered cryptosystems.*

***Keywords*: Quantum Algorithm, Post Quantum Security, Classical Computers, Functional Cryptosystems, Multivariate, Supersingular Elliptic Curve Isogenies.**

I. INTRODUCTION

Contemporary cryptographic algorithms are resistant to the strongest threats to cybersecurity and high profile cyber-attacks. In recent times, information security scientists and researchers had developed various cryptographic schemes that defeated attacks using the most sophisticated (in terms of processor speed) classical computer. However, this resistance will soon erode with the arrival of quantum computers. Before it is too late, there is the urgent need to start exploring ways to develop cryptographic algorithms that will stand against the huge computing powers of a quantum computer. Quantum computing promises to be a threat to cybersecurity in the nearest future as the strongest cryptographic algorithms will fall under the processing power of quantum computers. Most of the widely used public key cryptographic algorithms today depend on the hardness of solving mathematical problems such as factoring, discrete logarithm, elliptic curve, etc. Schemes such as RSA, Elgamal, Diffie-Hellman key exchange, ECDSA, and ECDH, are built from the difficulty property of the aforementioned mathematical problems. These cryptographic schemes in turn are used as the building blocks of higher-level security solutions like SSL, TLS, PGP, Bitcoin, HTTPS used by most organizations that have embrace the Internet and cloud computing as integral tool in achieving the set business objectives.

There are security proofs mathematically backing up each of these schemes' security strength. Such proofs serve as formal mathematical evidence that breaking the security of each scheme depends on the difficulty of solving underlying mathematical problem. These security proofs provide the assurance and necessary confidence in deploying these basic security primitives and building blocks for higher level security applications used in most digital infrastructure. In other words, the security of these digital infrastructures depend on the airtight security of the building blocks used in such security solution. However strong a cryptographic scheme may be in the face of classical computers, with respect to difficulty in solving hard mathematical problems and, whatever security proof and level of confidence is assured by such proof(s), the crypto schemes are bound to collapse under a quantum computer enabled cryptanalytic attack. A quantum computer can seamlessly solve the so called hard mathematical problem in few minutes.

Manuscript received on 06 May 2023 | Revised Manuscript received on 13 May 2023 | Manuscript Accepted on 15 June 2023 | Manuscript published on 30 June 2023.

*Correspondence Author(s)
Dr. Joshua J. Tom*, Department of Cyber Security, Elizade University, Ilara Mokin, Nigeria. Email: joshua.tom@elizadeuniversity.edu.ng, ORCID ID: https://orcid.org/0000-0001-9644-6805
Dr. Nlerum P. Anebo, Department of Computer Science, Federal University Otuoke, Nigeria, Email: nlerumpa@fuotuoke.edu.ng, ORCID ID: https://orcid.org/0000-0002-0437-8161.
Dr. Bukola A. Onyekwelu, Department of Cyber Security, Elizade University, Ilara Mokin, Nigeria. Email: bukola.onyekwelu@elizadeuniversity.edu.ng, ORCID ID: https://orcid.org/0000-0001-5040-5143
Adigwe Wilfred, Department of Computer Science, Delta State University of Science and Technology, Ozoro, Nigeria. Email: wilfred7k@yahoo.com, ORCID ID: https://orcid.org/0009-0007-2893-9342
Richard E. Eyo, IT/Support Specialist, Riverstrong, Texas, USA. Email: richard.e.eyo@gmail.com, ORCID ID: https://orcid.org/0009-0004-4136-4350

The problem we are faced with now is that even though quantum computing is still in its theoretical stage of realization, quantum computers pose a great threat to future cybersecurity. This fear is borne out of the fact that most of the number theoretic centred mathematical problems, which have been used as the basis for security for the past four decades can be easily solved by a quantum computer. For example, using Snor's Algorithm and a variant of Grover's Algorithm, a quantum computer can conveniently solve factorization problem and a number of mathematical problems based on number theory respectively. It is clear that with the power of quantum computers, the security proofs of the present day cryptographic schemes become inexpressive since the mathematical problems, which underlie their security are now solvable. Here, it is established how easy it is for a quantum computer to break the security of a cryptographic scheme. Therefore, no security proof can be adequate in this case since no scheme is secure against a quantum computer attack.

Based on the foregoing, is it important to begin a search for new mathematical problems hard enough to beat a quantum computer and develop new corresponding cryptographic schemes that are resistant to cryptanalysis using quantum computers. In this paper, we explore various mathematical structures that could point to hard-enough mathematical problems that cannot be solved by a quantum computer. Interesting structures that are good candidates for a post-quantum security assurance include lattices, multivariate quadratic equations, hashes, etc. Cryptographic systems based on the above are secure against both classical and quantum-computers enabled cryptanalytic attacks.

II. MATHEMATICAL BASIS OF SECURITY OF FUNCTIONAL CRYPTOSYSTEMS

Mathematical techniques forms the basis of Cryptography, a branch of cryptology concerned with the use of some mathematical problems considered intractable or whose solution is impractically difficult to determine with considerable time and resources. Existing cryptographic algorithms are based on particularly hard mathematical problems. These algorithms are required to be virtually unbreakable to continue to provide security services for which they are designed such as confidentiality, integrity, availability, and nonrepudiation. The mathematical problems considered hard enough to form the basis of the engineering of today's cryptographic systems include integer factorization, discrete logarithm, and the elliptic curve problems. In this paper, we analyze the intractability of these mathematical problems, by classical means and, why and how they are surmountable by quantum means. This will enable a clear understanding of the limitation of the current cryptographic primitives built from the so-called hard problem in the face of quantum computing and emergence of quantum algorithms. We explain the hard mathematical problems underpinning the security of the current cryptographic systems.

A. Integer Factorization Problem

Many cryptographic primitives and schemes derive their security from the intractability of the integer factorization problem. Protocols leveraging the hardness of the hardness of this problem includes include but not limited to RSA encryption, RSA signature, and the Rabin public-key encryption schemes. This section provides an understanding of the integer factorization problem while summarizing the current knowledge on some integer factorization algorithms. Factoring integers into prime factors is an extraordinarily difficult problem. It would not be wrong to think that many people had made much effort in solving this problem for centuries; hence, the chances of an efficient algorithm for factoring integers are negligible. This difficulty is good for cryptography for cryptosystems as they rely on the difficulty of this mathematical problem. Although many researchers and mathematicians worldwide have tried unsuccessfully to find efficient algorithms that can solve the integer factorization problem in polynomial time, there is no reason to think that factoring integer efficiently is not possible, but because one has yet found a polynomial time classical algorithm for factoring integers. Some of the different integer factorization algorithms developed for factoring integers, include trial division, Fermat's, and Pollard rho methods. However, these methods are the most fundamental methods that are easy to understand and implement, but none solves this problem in polynomial time. Another non-polynomial time, integer factorization method worth mentioning here is the quadratic sieve method. This method gave birth to the "general number field sieve" algorithm, the fastest integer factorization algorithm for very large integers. We define integer factorization (prime decomposition) problem as, given a positive composite integer k, the problem is to find the positive integers p and q for p, q > 1, such that k = p * q. Note that we are particularly interested in a case where p and q are prime numbers. There is a wide assumption that factoring an integer is a hard problem, but to this present day, there is no established and published proof of its hardness. Despite this lack of proof, with sufficiently large integer, there is no efficient classical (non-quantum) algorithm to solve factoring problem in polynomial time. The hardness of factoring integers is not for all composites, but for composites believed to be difficult easily generated. This is important for "RSA public-key encryption and the RSA digital signature" [8]. It is necessary to determine the degree of difficulty in finding the prime factors of large numbers. The time it takes a classical computer to compute prime factors of a given integer depends on both the processing power and the value of prime factors. As the product increases in size, so the numbers we use to check are bigger, and each check takes more time on average. It is therefore easy to understand here that factoring the product gets much and much harder as we add more digits onto the prime number. For this reason, RSA is believed to be secure, giving credence to the security of RSA public-key encryption and the RSA digital signature scheme [9, 10]. Semiprimes, the product of two primes, are the most difficult to factor using integer factorization methods among the b-bit numbers. The largest integer ever factored was an RSA-250 made up of 250 decimal digits (829-bits number) in 2020 with the computation time of nearly "2700 core-years of computing using Intel Xeon Gold 6130 at 2.1 GHz" to implement the General Number Field Sieve integer factorization algorithm [11]. Since there are no proof for the existence or nonexistence of algorithm but we only assume that such algorithm that could efficiently factor prime integers does not exist,

Retrieval Number:100.1/ijeat.E41530612523
DOI: 10.35940/ijeat.E4153.0612523
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

we can rightly say that the problem of factoring integer is not in class P [13, 14, 15]. However, it is an assumption that this problem is a hard problem until date. Hence, the problem to be NP though suspected not to be NP-Complete [12].

B. Discrete Logarithm Problem

It is important, first, to understand what discrete logarithm is all about before delving into relating to its use in cryptography. Put simply, "discrete logarithms are logarithms defined with regard to multiplicative cyclic groups". Now given a multiplicative cyclic group, G and its generator, g, then from the definition of cyclic groups and $\langle g \rangle$ a cyclic subgroup generated by g, we can have $h = g^x$ for some x where h ∈ G. The value x is called the discrete logarithm of h in the group G to the base g. We can represent this as $x = \log_g h$, which is only determined modulo the order of g. The discrete logarithm problem is therefore the problem of finding x given h and g formally defined as:

Given a group G, a generator g of the group and h ∈ $\langle g \rangle$, find an integer x such that $g^x = h$. While it easy to compute $h = g^x$, it is nontrivial to determine x.

This gives the discrete logarithm problem an important property, a one-way function. Generally, a one-way function is a function $f: X \to Y$ for which it is computationally feasible to determine $f(x)$ for all x ∈ X but with y ∈ Y, it is difficult to compute x from $f(x) = y$ [13]. The problem described above is computationally intractable especially if we give adequate attention to the choice of the group, G since the hardness of this problem depends on the group. For instance, most discrete logarithm-based cryptographic algorithms make use of the group of prime integers denoted by $\mathbb{Z}_p^*$ where p is prime. Even though there no efficient classical algorithm is known to solve this problem, Pohlig–Hellman algorithm can very efficiently solve the discrete logarithm problem if $p - 1$ has factors that are small primes. For this reason, p is always required to be a "safe prime" when using $\mathbb{Z}_p^*$ as the basis of discrete logarithm based cryptographic algorithm. The safe prime is constituted by another large prime number, q such that p = 2q + 1. There are many sophisticated algorithms for solving this problem inspired by similar methods for integer factorization. However, none of these algorithms runs in polynomial time. These include "baby-step giant-step", "function field sieve", "index calculus", "Pohlig-Hellman", and "Pollard's Rho" algorithms.

C. Elliptic Curves Method

The equation, $\boldsymbol{y^2 = (x^3 + a * x + b) \bmod p}$ is used to represent and elliptic curve which forms the basis of Elliptic Curve cryptography. By the nature of the above elliptic curve equation, there are obviously the x and y coordinate and from $\boldsymbol{y^2}$, it is possible to have two different values of $\boldsymbol{y}$, hence the curve's symmetric characteristic on the X-axis. Operating in modulo $\boldsymbol{p}$ constraints the $\boldsymbol{y}$ values to satisfy the membership range of the finite field. However, we are interested in those values that have a "perfect square" since $\boldsymbol{y}$ must be integers. Obviously, the set of values of $\boldsymbol{y}$ represents the number points N, on the curve that are perfect squares, where 0 < N < p. As stated earlier, each x produces a pair of $\boldsymbol{y}$ values, one being a symmetrical image of the other. The negative values are not of interest revealing that only N/2 possible 'x' coordinates are valid points on the curve. This gives a finite field whose properties satisfies the finiteness requirement of values used in cryptography. This is due to the integer calculations and the modulus operation on the curve. We consider two important operations on points of an elliptic curve that explains the strength of ECC. These operations are point addition and point multiplication. Two points on an elliptic curve can produce another point on the curve by a simple operation called "Point Addition". For example, adding one point P to another point Q leading to a new point *R* entails drawing a line from P to Q to intersect the curve at a third point R which symmetrically equals to the point P+Q on the curve as illustrated in Figure 1.

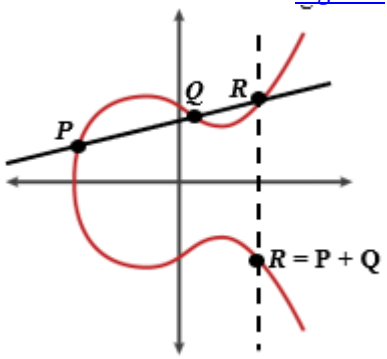


Fig. 1. Point Addition on an Elliptic Curve

Point multiplication is another important operation on an elliptic curve as far as cryptography is concerned. It is the repeated addition of a point to itself. As shown in Fig. 2, we draw a tangent from point *P* to intersect the curve at the second point *R* and the symmetric point to *R* is 2*P*, which is the addition of point *P* to itself resulting in point multiplication. Again, to multiply 2P by P, we draw a line from 2P to meet at P and it intersects the curve at *Q*. We then take the point *Z*, which is symmetrical to *Q*, and is equal to 3P, and so on.

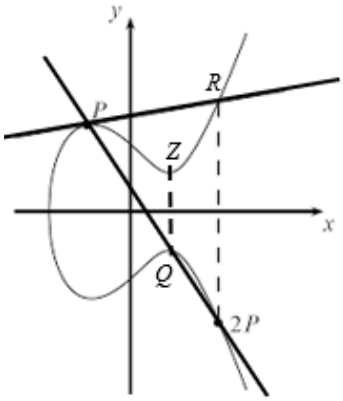


Fig. 2. Point Multiplication on an Elliptic Curve

Point multiplication is of great significance because given a point R = k*P, where R and P are known, it is nontrivial to determine value of k. This is because there no provision for subtracting or dividing points in elliptic curve mathematics hence $k = {}^{R}/_{P}$ cannot be resolved. In addition, with repeated additions of points, we only obtain a new point on the curve, and there is no way of determining "how" the current point on the curve resulted through backtracking since the addition operation is irreversible. Significantly, the multiplicand, k multiplied with the initial point P yielding another point on the curve, cannot be efficiently determine by a classical computer.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

This forms the basis of the security behind any ECC-based algorithm, a principle referred to as trap door function.

III. FUNCTIONAL CRYPTOGRAPHIC SYSTEMS

The security of existing cryptographic systems such as Rivest-Shamir-Adleman (RSA), Elgamal, Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic Curve Diffie–Hellman Key Exchange (ECDH) and Digital Signature Algorithm (DSA) cryptosystems typically depend on hardness of mathematical problems including integer factorization problem, discrete logarithmic problem, and points manipulation on elliptic curves. Classical computers' computational capability is not adequate to break the above-mentioned cyptos in polynomial time, as they deal with 2-bit processing. However, quantum computers can process q-bits at a time, a property that renders all the conventional cryptographic systems an endangered cryptosystems. It is clear that a quantum computer leveraging the quantum mechanics properties of superposition and entanglement could carry out quite complex computations in a matter of hours, which is not possible for classical computers in years. Thus, with quantum computers and algorithms, businesses and organizations, using the internet as a business component stand the risk of insecure transactions because the aforementioned schemes are the building blocks of security protocols such as Transport Layer Security (TLS), SSL, Pretty Good Privacy (PGP), HTTPS, and cryptocurrency used by businesses and organization for secure communication as mentioned earlier. It is important to do a review of the existing cryptographic systems in order to understand how they are bound to crumble under quantum computer and quantum algorithm enabled attacks.

A. Rivest-Shamir-Adleman Algorithm

Rivest-Shamir-Adleman abbreviated RSA is a public key encryption cryptographic system invented by the three-some Ronald Rivest, Adi Shamir, and Leonard Adleman in 1978. The security of RSA cryptosystem is based on the nontriviality in finding two large primes and nontrivially factoring the product of the two large prime numbers. The security of RSA therefore depends on the difficulty of solving the integer factorization problem.

With RSA, we select two large prime numbers p and q and compute $N = pq$. Then we work out the PHI value as $PHI = (p\text{-}1)\ (q\text{-}1)$. The attacker can only know the *PHI* value by finding p and q through integer factorization, a nontrivial task for classical computers provided there is no common factor of p-1 and q-1. The encryption key, e is derived in a way that $1< e < (p\text{-}1)(q\text{-}1)$ and e does not share value with Phi, that is e and Phi are not a coprime. The decryption key, d is calculated as d = e-1 mod (*Phi*). The encryption and decryption algorithm in RSA are given as:

Encryption: $c = m^e \pmod N$

Decryption: $m = c^d \ (mod\ N)$

From the above, we can see that the attacker could only decrypt the ciphertext, c if he can get access to the decryption key, *d*. He can only know *d* if knows *Phi*, which can only be determined by factorizing *N* into *p* and *q*. Factorizing a the product of two large prime numbers to get the individual prime numbers can take a classical computer years to accomplish.

B. Elgamal Algorithm

In 1984, Taher ElGamal invented a digital signature scheme called Elgamal named after him. Elgamal algorithm is one of the digital signature algorithms [1]. It is a public key cryptographic scheme whose security is based on the hardness of discrete logarithm problem (DLP) in a cyclic group. The algorithm proceeds in three phases: key generation, encryption and decryption phases. Our concern here is to consider the parameters in the key generation phase and show how difficult it is for a classical computer to compromise the cryptosystem. If Bob is to communicate securely with Alice, the key generation goes thus: Bob selects picks a prime number p, a primitive root, r of p and an integer a kept as secret such that $0 \leq a \leq p-1$. He then calculates the least nonnegative residue $b \equiv r^a \ mod\ p$. The public key is now given as the tuple (p, r, b). To decrypt the plaintext, an adversary needs to know a, which entails taking the discrete logarithm of b using the primitive root, r. even if the adversary has knowledge of r and arguably tires many instances of a until $r^a \equiv b \ mod\ p$, if p is a large prime number this is very impractical. The ElGamal cryptosystem is based on the assumption that while exponentiation is easy in modular arithmetic, it is difficult to take (discrete) logarithms. Mathematically, we can agree that $a \equiv \log_r b \ mod\ p-1$ is difficult to compute by an adversary using a classical computer.

C. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA is based on Elliptic Curve Cryptography (ECC). ECC is an alternative to RSA, which uses two points on a given elliptic curve instead of two large primes, as is the case in RSA. ECC derives its strength from the algebraic structuring of elliptic curves over finite field, which makes it mathematically difficult to subvert ECC encryption and decryption keys. Because of ECC's relatively small key size, it is one of the most widely used technique for implementation of digital signatures. Bitcoin and Ethereum are two cryptocurrencies deploying ECDSA to sign transactions. ECC is also been chosen as a standard for encryption by most mobile and web applications due to it shorter key length. Cryptographically, ECC is based on the hardness of discrete logarithm problem. In 1985, Koblitz [2] and Miller [3] invented the idea of using elliptic curves to build cryptosystems. Elliptic curve-based cryptosystems are the elliptic curve versions of discrete logarithm-based cryptosystems where the group of points on an elliptic curve over a finite field replace the subgroup of prime integers, $\mathbb{Z}_p^*$. The security of the elliptic curve-based schemes is the intractability of elliptic curve discrete logarithm problem (ECDLP) [4]. ECDSA is the elliptic curve version of the Digital Signature Algorithm (DSA) originally proposed by Vanstone in 1992 [5] because of the request by NIST for proposal for a new "Digital Signature Standard".

D. Elliptic Curve Diffie–Hellman Key Exchange (ECDH)

The ECDH is an anonymous key exchange scheme in which two parties in a secure communication wish to exchange a shared secret over an insecure channel.

Each parties must use one of a pair of an elliptic-curve based public and private key. ECDH is similar to DHKE (Diffie–Hellman Key Exchange) algorithm, the only difference being in the group used to generate the shared secret. DHKE employs a multiplicative cyclic group made up of integers whereas ECDH replaces the integers with points on an elliptic curve. Unlike DHKE, which uses "modular exponentiation", ECC uses point multiplication making it a better option for implementation on constrained devices. We discuss property of elliptic curve points as it relates to the shared key generation:

Given two secret numbers, *a* and *b*, denoting two private keys belonging to two parties engaged in communication over an insure channel. Also, given an elliptic curve with generator point *G*, the two parties can exchange the values of their public keys and then compute a shared secret, ***secret*** = (***a*** * **G**) * ***b*** = (***b*** * **G**) * ***a*** as depicted in the following protocol which proceeds in two steps:

Key Generation Step

1. Party_1 generates a random ECC key pair: {Party_1_PrivKey, Party_1_PubKey = Party_1_PrivKey * G}
2. Party_2 generates a random ECC key pair: { Party_2_PrivKey, Party_2_PubKey = Party_2_PrivKey * G}
3. Party_1 and Party_2 exchange their public keys through the insecure channel (e.g. over Internet)

Secret Key Calculation Step

4. Party_1 calculates sharedKey = Party_2_PubKey * Party_1_PrivKey
5. Party_2 calculates sharedKey = Party_1_PubKey * Party_2_PrivKey
6. Now both Party_1 and Party_2 have the same sharedKey == Party_2_PubKey * Party_1_PrivKey == Party_1_PubKey * Party_2_PrivKey

As explained in the Elliptic Curve Method section, the point multiplication operation is a very important contributor in the security of the ECDH Key Exchange protocol. Point multiplication is evident in the public key computation process where the point, *G* on the elliptic curve is repeatedly added to itself on the generator point *G* a number of times as determined by the randomly selected secret key value, where the secret value cannot be determined in polynomial time. This difficulty in determining the secret key plays into the computation of the shared key, as it is a function of the public key.

E. Digital Signature Algorithm (DSA)

The DSA is a fundamental algorithm employed in designing secure digital signature solutions for data transmission. It is a "Federal Information Processing Standard" (FIPS) for digital signatures based on modular exponentiation and discrete logarithm. A "Digital Signature" is a cryptographic primitive, which is fundamental in providing security services such as authentication, authorization and nonrepudiation [6]. Digital signatures has been deployed in different applications as a security mechanism since the protocol is considered tough to defend against the active attacks and passive attacks. The areas include but not limited to key agreement, contract signing, chip level programming, fault tolerant technique, assessment systems, identity based authentication, etc. Key agreement protocol is used to establish secure communication between two parties who need to communicate with each other securely. This requires both parties to agree on key information secretly over a distributed medium [7]. Security services provided by the digital signature schemes are authentication, integrity and non-repudiation. A better understanding of DSA requires looking at the algorithm as a protocol. The algorithm is made up of private/public keys generation phases and the signature generation/verification phase.

Private/Public Keys Generation

1. Select q, a prime number.
2. Select a second prime number p, (s.t. p-1 mod q = 0)
3. Select an integer g, such that 1 < g < p, $g^q \bmod p = 1$ and $g = h^{((p-1)/q)} \bmod p$. Where q is g's multiplicative order modulo p.
4. Select an integer *x*, such that 0 < x < q.
5. Evaluate $y = g^x \bmod p$.
6. the tuple {p,q,g,y} gives the public key
7. the tuple {p,q,g,x} gives the private key

Signature Generation/Verification

1. generate the message digest h
2. select a random number k, such that 0 < k < q.
3. Compute $r = (g^k \bmod p) \bmod q$. If r = 0, select a different k.
4. Compute i, such that $k * i \bmod q = 1$.
5. Compute $s = i * (h + r * x) \bmod q$. If s = 0, select a different k.
6. the tuple {r,s} is the digital signature
7. The sender sends the message, h, and the digital signature to the receiver

To verify a signature, the receiver proceeds as follows:

1. Generate the message digest h, using the same hash algorithm.
2. Compute w, such that s*w mod q = 1, where w is the modular multiplicative inverse of s modulo q.
3. Compute u_1 = h*w mod q.
4. Compute u_2 = r*w mod q.
5. Compute $v = (((g^{u_1}) * (y^{u_2})) \bmod p) \bmod q$.
6. If v == r, the digital signature is valid.

Considering the DSA algorithm above, it is nontrivial for a signature to be forged or compromise in either active attack or passive attack using a classical computer since it is based on discrete logarithm. In the key generation phase where the private-public pair is generated, it is nontrivial for an attacker to determine *x* given *y*, in step 5 of the key generation phase, using classical computer. This is because it is a difficult and impractical to compute the discrete logarithm, $x \equiv \log_g y \bmod p$. The same challenge is encountered in the signature generation phase where if the attacker must know the value of *r* and possibly *s* in other to compromise the digital signature, he must be able to obtain the discrete logarithm $k \equiv \log_g r \bmod p$. The values *r* and *s* can only be determine if *k* is known.

IV. QUANTUM COMPUTERS AND ALGORITHMS

Having carefully provided explanations on the hard mathematical problems including integer factorization, discrete logarithm, and elliptic curve problems,

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

we now introduce quantum computers and quantum algorithms that have the computational capabilities to break any cryptographic system designed or developed using cryptographic primitives that are based on any of the above-mentioned presumably hard mathematical problems.

In this section, we investigate and present clarification on how quantum algorithms and quantum computers are able to solve the long perceived hard problems in polynomial time as against the widely believed notion that no classical algorithms running on classical computers exist that can solve the existing presumably hard mathematical problems in the acceptable time and space. This paper is basically aimed at uncovering the points at which the so-called hard mathematical problems crumble in the face of quantum computing algorithms so as to provide a direction on our search for harder mathematical problems that can withstand quantum-based cryptanalysis. The power of quantum computers and the capability of quantum algorithms to solve hard mathematical problems generally used as the basis of design and development of current crypto systems have posed serious threats to continued assurance of secure communications. Unlike classical computers, which rely on transistors with the ability to represent either "high" or "low", "1" or "0" bit of data at a time, quantum computers are characterized by the use of quantum bits or qubits. A qubit is the basic unit of information processing of a quantum computer. With qubits a quantum computer can represent both states simultaneously, a phenomenon known as superposition (see fig. 3), which enables it to perform computations in parallel and thus in a large scale. Another interesting property of a quantum computer is entanglement, where a change in one qubit drastically affects the other irrespective of the distance.

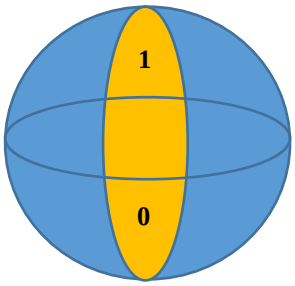


Fig. 3. Qubit superposition phenomenon representing 0 and 1 simultaneously

Quantum algorithms leverage the superposition property of quantum computers to process information with great speed even incomparable to the fastest classical systems. With this computing power, the magnitude of information that can be represented with, say, 500 qubits would not be possible to represent with even more than 2^{500} classical bits. Because of this limitation in terms of low computing power, it would take a classical computer a great number of years to factorize a 2,048-bit number whereas a quantum-based processor could do the computation in a couple of minutes. This is why the cryptographic systems relying on the hardness of integer factorization, discrete logarithm, and elliptic curve problems cannot survive quantum computer based attacks.

V. THE POWER OF QUANTUM COMPUTERS

The power of a quantum computer is due to its two important properties, superposition and entanglement. The capability of qubits to become entangled makes a quantum computer more powerful than a classical computer. Superposition allows a quantum computer to be able to represent both states of a qubit, "1" and "0" at the same time while entanglement allows for representation of every possible combination of the individual qubits. With large information capable of being stored through qubits superposition, some problems can be solved exponentially faster. With entanglement, qubits are configured such that the probabilities of each qubit are affected by all other qubits. This makes it possible for the quantum computer to consider all possible arrangement of the qubits in parallel whereby one desired candidate result is adopted as an intermediate or final result. For instance, a quantum computer with two entangled qubits processes four possible different combinations of the 2-qubits, i.e. 00, 01, 10, and 11 are represented at once. With three entangled qubits, a quantum computer is able to represent eight possible different combination at once, i.e. 000, 001, 010, 011, 100, 101, 110, and 111. The processing power of a quantum computer and hence the amount of information a qubit system can represent grow exponentially with each additional qubits that are entangled together making it more capable than its classical counterpart does. We look deeply into these two important quantum phenomena.

A. Superposition of Qubit States

There are basically two levels of direct current (DC) voltage. Classical computing implements a processed bit using either of these two DC voltage levels. When switching between these two levels, a forbidden zone must be crossed as fast as possible because the switch or change in the electrical voltage from one level to another cannot take place instantaneously. This means that there is only one possible outcome for the measurement of a "bit" of information represented with either "low" for "0" or "high" for "1". A qubit, used in quantum computing, has two possible outcomes for its measurement having the value "0" and "1". According to quantum mechanics, the basic state of a qubit is derived from the superposition of "0" and "1" as contrasted with a classical bit [14]. The state of a qubit is represented by a superposition of two basis states orthonormal to each other and denoted by $|0\rangle = \begin{matrix}1\\0\end{matrix}$ and $|1\rangle = \begin{matrix}0\\1\end{matrix}$. These states $\{|0\rangle, |1\rangle\}$ together is referred to as "computational basis" spanning the 2-dimensional vector space of the qubit. For example, a 4-dimensional linear vector is can represent two qubits in as follows:

$$|00\rangle = \begin{matrix}1\\0\\0\\0\end{matrix},\ |01\rangle = \begin{matrix}0\\1\\0\\0\end{matrix},\ |10\rangle = \begin{matrix}0\\0\\1\\0\end{matrix}, \text{and } |11\rangle = \begin{matrix}0\\0\\0\\1\end{matrix}$$

In general, a quantum register with n-qubits can be represented in a 2^n-dimensional linear vector space [25]. The basis states can coherently be superposed resulting in a "pure qubit" state. This simply means that given a qubit, ψ and probability amplitudes of the qubit's basis states, α and β, we define a linear combination of ket $|0\rangle$ and ket $|1\rangle$ as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

However, based on Born rule, it is important to note that when measuring qubit, ψ, the probability of outcome $|0\rangle$ and $|1\rangle$ is $|\alpha|^2$ and $|\beta|^2$ respectively.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

If we consider the absolute squares of the amplitudes as being probabilities, then we must constrain α and β, based on the "second axiom of probability theory" [16] given by the equation:

$$|\alpha|^2 + |\beta|^2 = 1$$

Worthy of note, also, is the fact that α and β do not only encode the probabilities of the outcome of a measurement, they also hold information about the relative phase between α and β, which is responsible for another quantum phenomenon known as quantum interference. Now, we can visualize the possible states for a 1-qubit using Bloch sphere in fig. 4.

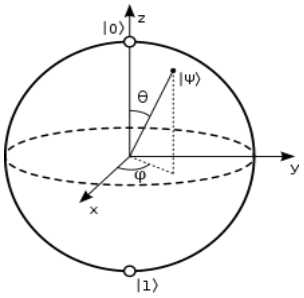


Fig. 4. A single qubit visualization using Bloch sphere

A classical bit can be on either the north or south pole of the sphere where $|0\rangle$ and $|1\rangle$ respectively are located on the diagram above with the North-South poles arbitrarily chosen. However, a pure qubit state can access any point on the surface of the sphere.

B. Entanglement of Quantum Qubits

Entanglement, one of the properties of a quantum computer, allows more than one qubits to be "entangled" to form a single system. A special connection exists between two qubits when they are entangled. All qubits within an entanglement are described as one unit. Entangled qubits share a state of superposition and are hence, dependent of each other. Entangling qubits offers a quantum computer more processing power. As earlier stated in this paper, every addition of a qubit in an entanglement increases the processing power of a quantum computer exponentially. Applying an operation to one qubit in an entangled state affects other entangled qubit(s) even when they are billions of miles away from each other. This allows the express higher correlation which is not achievable in classical systems. To understand this correlation, we explain further using fig. 5.

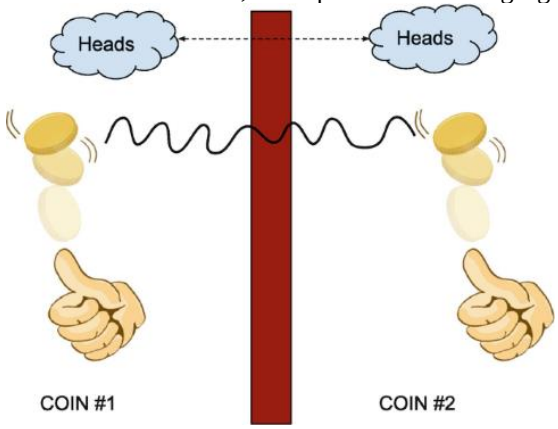


Fig. 5. Two entangled coins with no means of communication produce correlated results.

For instance, given two entangled coins, c_1 and c_2 separated by a great distance and c_2 is flipped and measured. Assuming c_2 produces the outcome heads then we expect that c_1 would produce the heads. In the same manner, if c_2 produced tails as the outcome, we automatically expect c_1 to produce tails. This can easily suggest that the coins, c_1 and c_2 are capable of instantaneous transmission of information between themselves with very high speed, see fig. 5. The experimented coins, flipped simultaneously with millions of miles away, somehow produce the same result with practically no classical communication between them. On a curious note, what is the basis of this seemingly instantaneous transfer of information between the two coins? However it may appear, studies have shown that there is no information transfer from coin to coin, hence no information is transferred. Rather, we understand that the coins shared information during entanglement observed in the "measurement process [17]. This correlation between entangled qubits is the property, which bequeaths speed to quantum computers in performing certain computations much faster than classical computers, which makes a quantum computer a threat to current cryptographic systems.

VI. POST QUANTUM CRYPTOGRAPHY (PQC)

Having critically examined the cryptanalytic power of quantum computers when it comes to reality, current cryptographic systems will lose their ability to provide the level of secure protocols and application currently enjoyed by businesses and government departments and agencies. The current cryptographic systems use the hardness of integer factorization, discrete logarithm, and elliptic curve problems to provide security. This has led to the cryptographic primitives based on the hardness of these problems becoming cryptographically irrelevant with quantum computers. The sudden erosion of the hardness of the aforementioned mathematical problems is attributed to the threat posed by the magnitude of the processing power of a quantum computer. The solution to this is to explore more difficult mathematical problems than the previous, which the quantum computer cannot find solution to in polynomial time. In preparation for the post quantum era, researchers in the field of cryptography focus attention six areas of advanced mathematics. These areas of advanced mathematics gave rise to lattices, multivariate, hash, code, super singular elliptic curve isogeny, and symmetric key quantum resistance based cryptography. In the preceding sections, we will discuss these quantum resistant approaches and see if they can efficiently provide replacements for the classical approaches found to be tractable by a quantum computer. In our related works section, we will also review the works of authors who had based their research on these areas of post-quantum cryptography.

A. Lattice-based Cryptography

One of the areas of advanced mathematics that promises to be of use in preparing for a post quantum era is lattices. Lattice based cryptography a quantum resistant approach aimed at increasing the security of future cryptographic systems that will help to prevent against quantum computer-based cryptanalysis.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The confidence that a lattice-based cryptographic system is secure against quantum computer and quantum related algorithm precipitates on the hardness of the lattice problem. Many lattice-based cryptosystems or primitives are secure based on the assumption that certain well-investigated computational lattice problems are intractable. In this section, we analyze the computational complexity of the lattice problem to ascertain the suitability of the hard lattice based problem for use in designing cryptosystems that can stand against a quantum-enabled cryptanalysis.

A lattice is a set of regularly spaced points arranged infinitely on any vector space [18]. It is an "infinite discrete structure" generated by a finite set of vectors called basis, represented as $(b_0, \dots, b_{j-1})$ [19]. More formally, a lattice can be defined as a discrete subgroup of $\mathbb{R}^j$ or $\mathcal{L}(b_0, \dots, b_{j-1})$. Mathematically, Let B = $(b_0, \dots, b_{j-1}) \in \mathbb{R}^{jxk}$ be the basis with linearly independent vector in $\mathbb{R}^j$, B will generate the lattice given by the set of all integer linear combination of the columns in B as follows:

$$\mathcal{L}(B) = \{Bx: x \in \mathbb{Z}^k\} = \left\{\sum_{i=1}^{k} x_i b_i : x_i \in \mathbb{Z}\right\}$$

where B is the basis of the lattice $\mathcal{L}(B)$ and j, k are the dimension and rank of the lattice respectively. In figure 6 the basis (x_1, x_2) is said to be a bad basis because x_1 and x_2 are nonorthogonal vectors. However, (b_1, b_2) id a good basis since the vectors b_1 and b_2 are orthogonal. Note that the two bases (x_1, x_2) and (b_1, b_2) will generate completely different lattices.

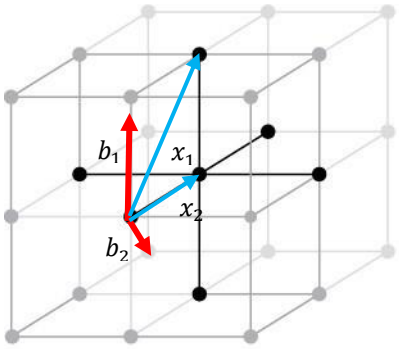


Fig. 6. Lattice representation using basis (b_1, b_2) and (x_1, x_2)

In fig. 6 above, (b_1, b_2) for instance is a vector or a basis of the lattice. By continuous addition and subtraction of number in the vector in any multiples, the lattice describes a pattern that continues to the infinity. The hardness of the lattice problem lies in finding points close 0 or any specified point in the lattice. For lattices with fewer dimension (e.g. 2 dimensions), the problems are quite simple, but as the lattice scales to higher dimensions problem becomes nontrivial and intractable such that both classical approach and quantum approach cannot solve the problem in polynomial time. Within the diagram above, an example would be that one set of points could be the private key and another set of points that are further away could be the public key. Cryptanalytically, if brute force is used to search of all possibilities to derive the private key from the public key, and even if quantum computers can speed up this search, it would still take a sizable and unrealistic amount of time. Even a quantum computer unable to solve hard problems based on lattices in polynomial of time. In this paper, we present two lattice-based hard problems, which singles out lattice based cryptography as a favorite and high profile option for post-quantum cryptography.

1. Unique Shortest Vector Problem (uSVP) - This problem is defined as follows: Given two positive integers, n and m, a parameter $\rho \geq 1$, let the basis of a lattice, $L \subset R^m$ that verifies the condition $\beta_2(L) > \rho.\beta_1(L)$ be $(b_0, \dots, b_{n-1})$. The challenge is find a vector $v \in L$ such that $\|v\|_2 = \beta_1(L)$. The problem reduces to Shortest Vector Problem (SVP) if $\rho = 1$. uSVP searches for shortest vector in $\mathcal{L}$.
2. Bounded Distance Decoding (BDD) - This problem is defined as follows: Let n and m be positive integers and $\alpha > 0$ a parameter, and given a basis $(b_0, \dots, b_{n-1})$ of a lattice, $\subset R^m$, and an element $t \in R^m$ that verifies $dist(t, L) \leq \beta_1(L)$. The challenge is find a vector x $\in$ L closest to t.

The uSVP and BDD problems are intractable even with the use of quantum computers for lattices of considerably high dimension because these problems have strong foundational mathematical hardness properties. Given lattice instances that truly unpredictable with some range of parameters, the uSVP and BDD problems are NP-hard [20].

B. Multivariate Cryptography

A multivariate polynomial is a general case of polynomial where there are multiple variables defined as a standard polynomial whose coefficients are multiple-variate polynomials. A Polynomial is expressed in only one variable e.g. $x^3 + 3x - 1$. On the other hand, multi-variate polynomials are expressed in more than one variables like $x^4 - y^2z^2 + z^4$. The significance of multivariate polynomial in cryptography stems from the fact that a multivariate quadratic polynomial map over a finite field is capable to create a trapdoor one-way function. Given this characteristics, a public key for the cryptosystem can take the general form: $P = (p_1(w_1, \dots, w_m), \dots, p_m(w_1, \dots, w_m))$, where p_1 is a quadratic nonlinear polynomial in $\mathbb{w} = (w_1, \dots, w_m)$. Hence, the multivariate quadratic, MQ system with m equations in n variables all coefficients in $\mathbb{F}_q$ is given by the polynomial notation:

$$p_k(w_1, \dots, w_m) := \sum_{i,j} P_{ij}{}^k w_i w_j + \sum_i L_i^k w_i + c^k$$

In vector notation, we represent the above equation as follows:

$$p_k(w_1, \dots, w_m) = wP^k w^T + L^k w + c^k$$

with K as the "base field" equivalent to the finite field having q elements, $\mathbb{F}_q$. Thus, a public key cryptosystem in which a multivariate polynomial is the basis of its trapdoor is a multivariate-based PKC system. The polynomial $p_k(w_1, \dots, w_m)$, a system of quadratic equations defined over a finite field has a nontrivial solution. This operation is same as inverting a multivariate quadratic map, believed to be NP-hard. However, the set of quadratic equations required for MPKC must be ideal generated by those polynomials because with a random set of quadratic equations, a trapdoor is inexistent. The security of a multivariate-based cryptosystem is based on the difficulty of efficiently obtaining a solution to the multivariate quadratic problem defined as:

Retrieval Number:100.1/ijeat.E41530612523
DOI: 10.35940/ijeat.E4153.0612523
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Given a set of m quadratic polynomials in n variables $\mathbb{w} = (w_1, \dots, w_m)$, solve the system: $p_1(w) = \dots = p_m(w) = 0$. We can formulate the ideal polynomial, IP problem as: Given two polynomial maps $F_1, F_2: K^n \to K^m$. The problem is to look for two linear transformation L_1 and L_2 (if they exist) such that: $P(w_1, \dots, w_m) = L_1 \circ F \circ L_2(w_1, \cdots, w_m)$ where F is a quadratic map with certain structure, which allows the easy inverse computation F^{-1}. L_1 and L_2 are full-rank linear maps used to mask the special structure, F. For example, given a plaintext $M = (w_1, \dots, w_m)$, the ciphertext is given by the polynomial evaluation $P(M) = (p_1(w_1, \dots, w_m), \dots, p_m(w_1, \dots, w_m)) = (c_1, \dots, c_m)$ To decrypt the ciphertext $(c_1, \dots, c_m)$, we need knowledge of the trapdoor to make it feasible to invert the quadratic map to compute the plaintext as: $(w_1, \dots, w_m) = P^{-1}(c_1, \dots, c_m)$.

C. Hash-based Cryptography

Hash functions play important role in design and development of various security applications that provide security services including authentication, message integrity and nonrepudiation due to security properties [21]. Encryption is a (revertible) two-way function where a message, m encrypted into unreadable format using an encryption algorithm is decrypted back to a readable format using a decryption algorithm. Unlike encryption, hash functions are one-way functions and there is no "dehashing" function to invert a hashed message due to pre-image resistance. Another very important property of hash functions is that given an adversary with total freedom to select messages, the adversary must not find two messages with the same hash value given same hash function and hash key. That is to say given two messages, m_1 and m_2, $HASH(k, m_1) \neq HASH(k, m_2)$, where k is the hash key. This property is called collision resistance. Thirdly, given a message, m_1, it should be impossible for the adversary to obtain another message m_2 such that the hash of message m_1 is not same as message m_2 using the same hash key k. This is a stronger property than collision resistance known as second pre-image resistance. We see application of hash functions in data integrity checks, authentication algorithms, digital signatures algorithm, rootkit detection, generation of public/private key pair, secure socket layer connections, and virus checking. Having considered the security properties of hash functions, there is no gain saying that hash functions are useful candidates in many cryptographic primitives. This gives rise to hash based cryptography, a post quantum cryptographic method leveraging the irrevertibility of a hash value for design or specification of cryptographic cryptosystems. This post quantum cryptographic scheme basically, derives from a one-time signature (OTS) scheme that must use each key pair to sign and verify a message only once. Most of the signature schemes rely on one-way functions (hash functions) for their security. The suitability of a cryptographic hash function in post quantum computing security is because it has provable security against collision attacks based on the assumption that the problem of finding collisions is polynomial-time reducible, hence solving P is easier than finding collisions, which is unsolvable in polynomial time. This means that the problem of finding collisions is in the family of NP-hard computationally.

D. Code-based Cryptography

Coding theory is an aspect of mathematics, which studies different coding properties and analyze their individual fitness to use such as transmission of messages over noisy channels. Decoding, on the other hand, deals with converting these codes to normal messages without errors. Amongst the many decoding techniques is used to recover messages transmitted over noisy channels syndrome decoding. This method of decoding provides a highly efficient way to decode a coded information sent over a noisy channel. Syndrome decoding is a hard mathematical problem even for quantum computers, therefore generally proposed as a computational hardness assumption for code based cryptosystem safe against quantum cryptanalysis. Code based cryptography originated from the works of McEliece [22] in 1978 and Niederreiter [23] in 1986. As the current cryptographic systems like RSA, DHKE, ECDH, ECDSA, Elgamal, are soon becoming quantum vulnerable primitives, code based approach to post quantum cryptosystem security is one of the mathematical techniques penciled down to provide security against quantum computers and algorithms. Code based cryptography is premised on the syndrome decoding problem described in this paper. We define the problem as follows:
Given (m, k, w) where $k \leq m$ and $w \leq m$, a parity check matrix $H \in \mathbb{F}_q^{(m-k)\text{x}\, m}$ and a vector $s \in \mathbb{F}_q^{m-k}$ (the syndrome), the problem is to recover a vector $x \in \mathbb{F}_q^m$ of Hamming weight $\leq w$ such that $Hx = s$.
For random linear codes, for example, where H is a random matrix, the problem discussed here is NP-hard hence its suitability as the basis of code based cryptography. The syndrome, s serves as errors identifier in the received codeword and its value indicates the position of the code where the error occurs.

E. Supersingular Elliptic Curve Isogeny Cryptography (SECIC)

Mathematically, elliptic curves isogeny is explained thus:
Isogeny is a rational map $\phi: E \to E'$ between elliptic curves which maps the identity to identity. The degree of the isogeny ϕ is its degree as a rational map: $\deg \phi = [K(E): \phi^*(K(E'))]$ The hard problem on which the supersingular elliptic curve isogeny based cryptography depends for the desired security against quantum-induced vulnerability is finding isogenies between elliptic curves as in fig. 7. In terms of computational theory of isogenies, there are two kinds of isogenies namely separable and inseparable isogenies. Every purely inseparable isogeny has the form $\pi(x, y) = (x^q, y^q)$ for some prime power of $q = p^n$ and degree of $\phi = q$. Every (non-constant) separable isogeny has a finite kernel and is determined by its kernel (up to isomorphism): $G = \ker \phi \leftrightarrow \phi: E \to E'/G$ and $\deg \phi = |G|$.

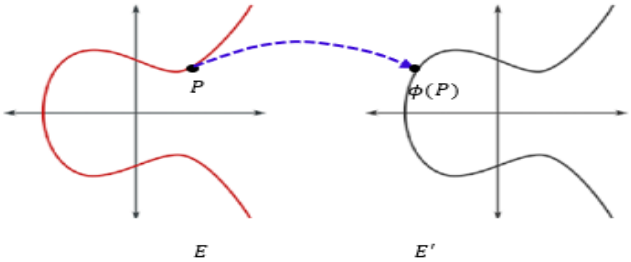


Fig. 7: Isogeny between two Elliptic Curves

This two isogenies are very different computationally but are easy to compute and classify. Separable isogenies are essentially all defined by their kernel. As expressed in the equation describing G, there exist a correspondence between the finite subgroup $G = \ker \phi$ with isogeny with finite kernel $\phi: E \to E'/G$ where $\deg \phi$ is expressed in terms of the size of the group, $|G|$.

We can use Vélu's formula to compute isogenies as follows: Given an elliptic curve E and a finite subgroup $G \subset E$, the map

$$\phi(P) = (\phi_1(P), \phi_2(P))$$

where

$$\phi_1(P) = x(P) + \sum_{Q \in G\{O_E)} [x(P+Q) - x/Q)]$$

$$\phi_2(P) = y(P) + \sum_{Q \in G\{O_E)} [y(P+Q) - y/Q)]$$

is a separable isogeny $\phi: E \to E'$ with $\ker \phi = G$. The above theory is employed in designing cryptosystems such as SIDH, CSIDH and SIKE. These cryptosystems, especially SIKE, have shown resistance against quantum attacks because breaking the systems is exponential in the size of the input. Even in cases where the existence of an isogeny in one direction implies the existence of a dual isogeny in the opposite direction (the source is same as the target), these functions are not inverses of each other: their composition is not the identity. This is because an isogeny is not an isomorphism. At this point, we make a differentiation between ECDH, an ECC based instance and SIDH, a supersingular isogeny based instance. Let the sender be *A* and the receiver be *B*. We first describe ECDH where two parties (*A* and *B*) wish to communicate with each other. Given a point on the elliptic curve, *P*, *A* computes $[n_A].P$, sends the result to *B* and *B* computes $[n_B].P$ sends the result to *A*. The key exchange is as shown in fig. 8. Thereafter, both *A* and *B* compute $[n_A n_B].P$ using their individual n values. Revealing *P* and *A*'s public key, $[n_A].P$ does not reveal its private key, $[n_A]$. Likewise revealing *P* and the *B*'s public,

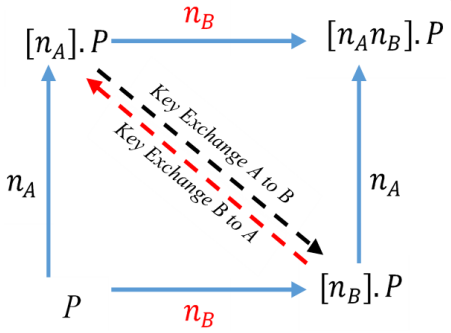


Fig. 8. ECDH

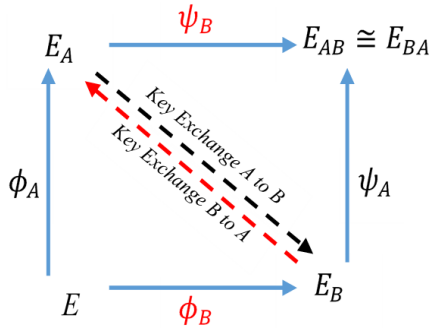


Fig. 9. SIDH

key, $[n_B].P$ does not reveal its private key, $[n_B]$. Hence even if an adversary have access to the points on the elliptic curve *P*, the sender's public key, $[n_A].P$, and the receiver's public key, $[n_B].P$, it is not computationally feasible for him to compute $[n_A n_B].P$. The receiver's private key is the isogeny ϕ_B. The security of ECDH depends on the hardness of ECDLP which, however, is broken if the adversary had access to a large-scale quantum-computer, a reason good enough to introduce isogeny based approach to cryptography. Fig. 9 shows the construction of a quantum-safe algorithm using isogeny. This uses a more general functions to map elliptic curves to elliptic curves. Here, we replace the scalar multiplication map, $[n_A]$ by the isogenies ϕ_A and ψ_A. Also, we replace the scalar, $[n_B]$ by the isogenies ϕ_B and ψ_B. Notice that the isogenies ϕ_A and $\phi_B$is equivalent to ψ_A and ψ_B respectively but they originate from different curves. The sender's private key is the isogeny ϕ_A and The sender's private is used to compute ψ_A while the receiver's private key is used to compute ψ_B.

F. Symmetric Key Quantum Resistance

Symmetric cryptosystems e.g. DES, Triple DES, and AES etc. use a single to both encrypt and decrypt messages. DES is a 64-bit block cipher technically using 56-bit key size. Due to its relatively short key length, DES is vulnerable to brute force attack and hence the cryptosystem has been broken and rendered insecure. Because of this shortcoming, AES was developed to address DES's insecurity. The new cryptosystem uses 128, 192, or 256 bits key sizes and is secure against classical based attacks. With a quantum computer having enormous computational ability, a 128-bit AES will become insecure. Generally, it is possible to safely increase the key length to 256 bits in a symmetric-key-based system as required by the information security standard. With this, the adversary requires to try not less than half of the key instances to compromise the system. Grover's algorithm running on a quantum system would break symmetric key systems in polynomial time. This entails searching the key space using only $\sqrt{2^n}$ CPU cycles to find the secret key as against the Brute force attack, which requires 2^n cycles. This vulnerability to Grover's algorithm is mitigated greatly by increasing the key length to 256 bits, which of course is a legal AES key size making AES-based schemes to be secure post-quantum cryptosystems [24]. However, just doubling the key length of AES is not enough to withstand the speed of the quantum computer. A key is strong and secure if the private key entropy high, i.e. the generator needs to generate truly random numbers for the private keys. Entropy measures private key's randomness and is a fundamental component of effective encryption. The implication of a low enthropy key is that an attacker can compromise the private key even the key length increased to 256 bits. Quantum random number generators (QRNGs), which uses the unpredictability of quantum measurements in generating random numbers have also been used recently. To make AES quantum resistant, we must use private keys with high entropy in addition to increased AES key length. By considering the level of security of AES that would be achieved assuming the private keys are generated with low-entropy-based random number generator,

Retrieval Number:100.1/ijeat.E41530612523
DOI: 10.35940/ijeat.E4153.0612523
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

we would appreciate why simply increasing the physical key size to 256 bits would not guarantee quantum attack resistance for AES, but also using keys from a high entropy source. Fig. 10 shows the visualization of the running times Grover's algorithm given n = 256 bits.

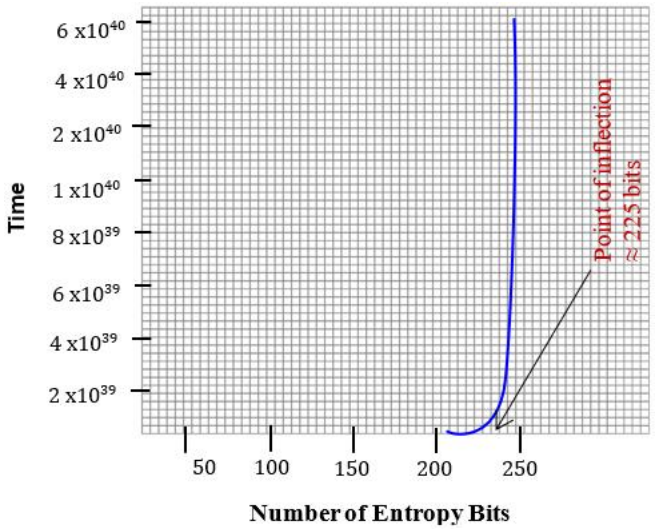


Fig. 10. 256-bit AES private key with high entropy Resistance against Grover's Algorithm

We assume the entropies of the private key ranges between 32 and 256 bits. This demonstrates the fact that even with the key length increased to 256 bits, low entropy values between 32 and 224 have no impact on time needed by Grover's algorithm to break the system. It is evident that the running time of the algorithm went exponential only at the point of inflection at about 225 bits entropy.

VII. QUANTUM COMPUTING BASED ALGORITHMS

In this section, we are only interested in the dynamics of how a quantum algorithm is designed having treated the basics of quantum computing including superposition of qubits and the property of entanglement. A quantum algorithm (e.g. Shor's and Grover's algorithms) basically consist of two parts; the quantum part composed of quantum circuits for performing difficult exponentiation operations or complex searching operation and a classical component to handle any classical operation. Quantum algorithms proceeds in two stages. The first stage is the input stage where the algorithm prepare an initial quantum state based on the input data, and implement a quantum circuit on it. The quantum circuit specifies the stepwise manipulation of the data held as qubits. In the second stage, the output from the circuit is measured and some post-processing operations done using a classical computer to get the final output. However, we may require more than one iterations of the algorithm to obtain an optimal result. Here, we present Shor's algorithm and Grover's algorithm, the two quantum algorithms that threaten to dislodge today's cryptographic systems.

A. Shor's Algorithm

Shor's algorithm is one of the quantum algorithms used with quantum computers to cryptanalyze the conventional cryptographic systems we have mentioned earlier, particularly the asymmetric (public key) cryptographic systems. In 1994, Peter Shor [25] designed an algorithm that showed that factorizing large integers was fundamentally possible using a quantum computer. Modern asymmetric cryptography such as RSA, ECDH, Elgamal, etc. are vulnerable to quantum based attack with Shor's algorithm. For example, let us find the prime factors of 255 using Shor's algorithm. To do this, a 8-qubit register is needed (255 = 11111111). Shor's algorithm defines parallel computations for all values in the range 0 to 255 in a single computational cycle. In the algorithm, n is set to 255, a random number is selected such that $1< x < n-1$. The selected number, e.g. x = 2 is raised to the powers of the 8-qubit register, in this case 2^0 , 2^1 ,…, 2^{254} , 2^{255} . In each case, the algorithm computes $f(q) = x^q \bmod n$, where q is a state of the qubit register. We present a partial evaluation of this in Table I.

Table I: Two 8-Qubit Registers with values q and Remainders from (q)

Register q	0	1	2	3	4	5	6	7	8	…	248	249	250	251	252	253	254	255
Register R	1	2	4	8	16	32	64	128	1	…	1	2	4	8	16	32	64	128

Notice the interesting recurring pattern in the R register. This leads to finding the smallest positive integer, r that satisfies $x^r \equiv 1 \bmod N$, called the period of the function, $f(q) = x^q \bmod n$. Hence, the period of the function is r = 8, which is then used to calculate a possible factor, P using $P = x^{r/2} - 1$. Therefore, a possible factor of 255 is $P = 2^{8/2} - 1 = 15$. The computation is repeated with different r values if the result, P is not a prime number. The generation of the period is very important to Shor's quantum algorithm and relies on the superposition property of the quantum computer. Shor's algorithm is made up of two parts, a classical part to carry out some post-processing that need a classical computer and a quantum subroutine part. Fig. 11 shows the quantum subroutine for a variant of Shor's algorithm for factoring integers. The most tasking job in the circuit is the quantum transform executed by the sequence of gates, U that performs the modular exponentiation modulo N i.e. ($f(q) = x^q \bmod n$), where N is the number whose prime factor is computed. The leftmost boxes, H are the Hadamard gates used to create a superposition of the qubits states. The box labelled QFT^{-1} is the controlled rotation gate used in conjunction with the Hadamard gate to perform Fast Fourier Transform. After transformations are performed, the output is measured yielding an approximation to the period, r.

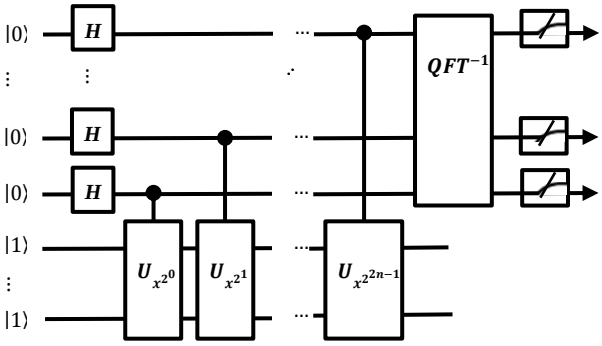


Fig. 11. Quantum Circuit for Shor's Algorithm

The heart of Shor's quantum algorithm is the Period-Finding subroutine, which the output is use subsequently in computing the possible factors.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

B. Grover's Algorithm

Glover's algorithm is another quantum computing algorithm devised by Lov Glover in 1996 [26]. Grover's algorithm is a search algorithm that can search an unsorted database with quadratic efficiency. In the problem of searching through a database for a particular entry, let us as assume that we have N items in an unstructured database X, and it is desirous to determine if item x is present in X. This will take a classical search algorithm about N operations to complete the task. Nevertheless, Grover's algorithm running on a quantum computer can complete the same task in approximately the square root of N operations, (√N). This means that given an unsorted database consisting of 10,000 records, to find a single record based on a query will take a classical algorithm 10,000 operations to determine whether the required record exists in the database or not whereas Grover's algorithm uses only 100 operations to get the task done. A mathematical formulation of the above problem will offer more understanding of Grover's algorithm. Given search function f, an n-bit input, x and 0 or 1 as output, we represent the search function as f(x) =1 for the case where x is found otherwise f(x) = 0. We can relate this to the cryptanalysis of AES in the CPA model.

Given a known plaintext-ciphertext pair such that c=AES(k, m), where m is a known message and k is unknown encryption-decryption key, we can define a function f that decrypts c with all possible combination of the 128-bits of the AES key and determine the key, k which gives the massage-ciphertext mapping. The function outputs 1 if the decryption of the ciphertext produces a message that matches the known plaintext, and indication that the key has been successfully cryptanalyzed. As earlier pointed out, Grover's algorithm requires $\sqrt{2^{128}}$ i.e. 2^{64} operations to break a 128-bits AES, a very trivial task for the algorithm running on a quantum computer hence the proposal to increase the key length to 256 with high entropy for the source of the randomness of the key. Technically, Grover's algorithm use two quantum circuits, one circuit implements the function *f*, called the Oracle while the other is the diffusion operator. Grover's algorithm leverages the superposition property of a quantum computer to apply a single circuit to all possible states of the 128-bits of AES simultaneously. The Grover's diffusion operator flips the number around the mean such that the probability amplitudes is shifted causing a slight increase in probability to measure the correct element and decrease in the probability of measuring the wrong element. Fig. 12 shows a 64-qubit circuit for breaking a 128-bits AES crypto with Grover's algorithm using a phase oracle.

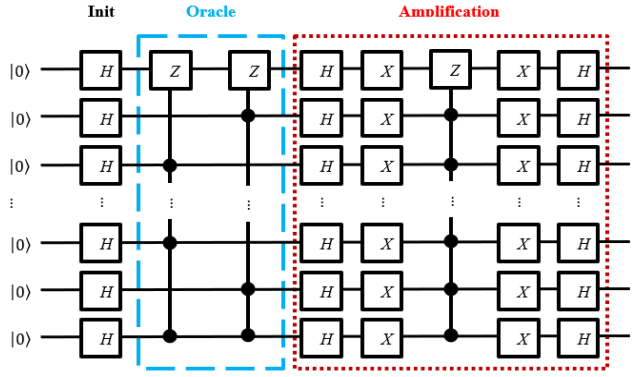


Fig. 12: A Quantum Circuit Implementing Grover's Algorithm for 128-bits AES Cryptanalysis following the implementation of Figgatt et al [27].

The boxes marked H represent a single qubit Hadamard gates, the Z marked boxes are the Pauli Z gates while the X boxes are the CNOT gates. Other applications of Grover's algorithm include estimation of the mean and median of a set of numbers and solving the collision problems.

VIII. DISCUSSION

Even though there is still a number of years before the actualization of quantum computing, it a worth idea to consider what will happen in the post-quantum era. There is no gainsaying that the enormous computing capability of a quantum computer will render the current cryptographic systems insecure. We have objectively looked at the claim that RSA, Elgamal, elliptic curve Diffie Hellman (ECDH), ECDSA, etc. will crumble under cryptanalytic attack involving quantum algorithms running on quantum computers. We found that all present cryptographic systems derived their security from the assumption of hardness of mathematical problems including integer factorization, discrete logarithm problem, and elliptic curve discrete logarithm problem. Based on this, we also found why and how these assumptions (where valid for classical computers and algorithms) are invalid when quantum computers and algorithms are involved. A quantum computer uses "qubit" as its basic unit of information processing as opposed to a classical computer. Hence, a quantum computer by its superposition and entanglement properties can perform several parallel computations in the same time a classical computer is processing a single bit of data. The importance of current cryptosystems as building blocks of very many higher-level security solutions, such as SSL, TLS, HTTPS, PGP, etc., led us to evidently, investigate the quantum computer's ability to render these solutions insecure because the security of the cryptographic primitives directly determine the security of the security solution which forms the bedrock of most organizations today. This finding enabled us to appreciate the enormity of threats posed by quantum computers and quantum-based algorithms such as Shor's algorithm that is capable of cryptanalyzing all present public key cryptographic systems and Grover's algorithm capable of breaking 128-bit-AES. Instead of just accepting solutions to this impending cryptographic doom, we set out to study advance mathematical problems that could form good security materials against quantum computing cryptanalytic power. This work confirmed that under coding theory, syndrome decoding is a hard enough mathematical problem that can defeat quantum-based cryptanalysis. Others are lattice-based mathematical problems, supersingular elliptic curve isogeny-based problem, multivariate problem, hash-based problem (from the assumption that given f(x) = y, it is both theoretically and computationally infeasible to compute x given y) and symmetric key quantum resistance (which enables AES to be quantum resistant by increasing key length from 128 to 258 with high entropy). We found that these mathematical problems would provide the level of security desired to provide continued security of applications and protocols, hence migration to post-quantum cryptographic era will be seamless.

Retrieval Number:100.1/ijeat.E41530612523
DOI: 10.35940/ijeat.E4153.0612523
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

IX. CONCLUSION

Grover's algorithm, running on a quantum computer, presents a threat to existing symmetric cryptographic systems especially AES using 128-bit key length. In addition, Shor's algorithm is a threat to public key cryptographic systems, e.g. RSA, ECDH, ECDSA, Elgamal, etc. For the AES cryptographic system to be post quantum compliant, a high entropy source for the randomness of the private key in addition to doubling the key length from 128 bits to 256 bits is required. However, based on this study, there is no remedy for the public key cryptosystems' weakness against quantum-enabled computers and algorithms. Completely different and more advanced mathematical problems are recommended by researchers to form the basis of the security of cryptosystems resistant to the impending cryptanalytic catastrophe. Even though attempts would be made in the future to compromise post quantum algorithms designed from these advanced mathematical problems, it will be difficult for non-experts in the field to understand the very complex manoeuvres of such hard problems.

DECLARATION

The involvement efforts contribution were in the following areas of the research paper.

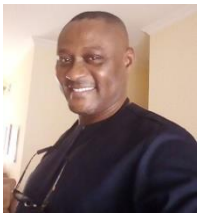
Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	The article does not require ethical approval and consent to participate with evidence.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	**Joshua J. Tom** (Conceptualization of the project idea, analysis of quantum cryptographic algorithms, analysis of post quantum cryptographic techniques, project administration, Supervision, Visualization, Writing original abstract), **Promise Nlerum** (Writing conclusion, review of quantum computing concepts), **Bukola A. Onyekwelu** (Review of classical cryptographic algorithms, reformat of references), **Wilfred Adigwe** (Project administration, analysis of classical and post quantum algorithms), **Richard E. Eyo** (Search for related research papers, determination of relevance and review of classical algorithm.

REFERENCES

1. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in Workshop on the Theory and Application of Cryptographic Techniques, Springer Berlin Heidelberg, 1984.
2. N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, 48 (1987), 203-209. [CrossRef]
3. V. Miller, "Uses of elliptic curves in cryptography", Advances in Cryptology– Crypto '85, Lecture Notes in Computer Science, 218, Springer-Verlag, 1986, pg. 417-426 [CrossRef]
4. D. Johnson, A. Menezes, S. A. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)", International Journal of Information Security 1(1):36-63, 2001, Springer. [CrossRef]
5. S. Vanstone, "Responses to NIST's Proposal", Communications of the ACM, 35, July 1992, 50-52 (communicated by John Anderson).
6. A. P. Menezes, Van, Oorschot, and S. Vanstone, CRC Press, 1996. Handbook of Applied Cryptography.
7. A. Sun, et al., "The QR-code reorganization in illegible snapshots taken by mobile phones," in Proc. Int. Conf. on Computational Sci. and its Applicant, 2007, pp.532-538. [CrossRef]
8. A. K. Lenstra, Integer Factoring. In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA. 2011, pp 611–618. [CrossRef]
9. V. I. Nechaev, Complexity of a determinate algorithm for the discrete logarithm. Math Notes 55(2):155–172. Translated from Matematicheskie Zametki 55(2): 91–101, (1994). This result dates from 1968. [CrossRef]
10. V. Shoup, Lower bounds for discrete logarithms and related problems. In: Proceedings of EUROCRYPT '97. Lecture notes in computer science, 1997. vol 1233, pp 256–266. [CrossRef]
11. Kleinjung et al. "Factorization of a 768-bit RSA modulus" (PDF). International Association for Cryptologic Research (2010-02-18). Retrieved 2010-08-09. [CrossRef]
12. O. Goldreich, Avi Wigderson, "IV.20 Computational Complexity", in Gowers, Timothy; Barrow-Green, June; Leader, Imre (eds.), The Princeton Companion to Mathematics, Princeton, New Jersey: Princeton University Press, 2008, pp. 575–604, ISBN 978-0-691-11880-2, MR 2467561. See in particular p. 583. [CrossRef]
13. McCurley, K. S. The Discrete Logarithm Problem. Proceedings of Symposia in Applied Mathematics, vol. 42, 1990. [CrossRef]
14. M. A Nielsen, Chuang, Isaac L. X. Quantum Computation and Quantum Information. Cambridge University Press, 2010. p. 13. ISBN 978-1-107-00217-3.
15. P. Shor, (1997). "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer∗". SIAM Journal on Computing. 26 (5), 1995 pg. 1484–1509,. [CrossRef]
16. C. P Williams, Explorations in Quantum Computing. Springer (2011). pp. 9–13. ISBN 978-1-84628-887-6.
17. C. Hughes, J. Isaacson, A. Perry, R. F. Sun, J. Turner, Entanglement. In: Quantum Computing for the Quantum Curious, 2021. Springer, Cham. [CrossRef]
18. P. K. Pradhan, Sayan Rakshit and Sujoy Datta. "Lattice Based Cryptography: Its Applications, Areas of Interest & Future Scope." 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (2019): 988-993. [CrossRef]
19. Nguyen, P. Q., & Stern, J. The two faces of lattices in cryptology. In International Cryptography and Lattices Conference (pp. 146-180). Springer, Berlin, Heidelberg, (2001). [CrossRef]
20. R. Mélissa, Extended Security of Lattice-Based Cryptography. Cryptography and Security [cs.CR]. Équipe CASCADE, Département d'Informatique de l'ENS de Paris; Université PSL, 2020.
21. S. Edem, G. Vivek, G. R. Sandhya, Role of Hash Function in Cryptography. Conference: National Conference on Computer Security, Image Processing, Graphics, Mobility and Analytics. 2016. DOI: 10.22161/ijaers/si.3. [CrossRef]
22. R. A. McEliece, Public-key cryptosystem based on algebraic coding theory, DSN Progress Report, 1978, 114–116.
23. H. Niederreiter, Knapsack-type cryptosystems and algebraic coding theory, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform., 15, 1986, 159–166.
24. F. Song, A Note on Quantum Security for Post-Quantum Cryptography. In: Mosca, M.(eds) Post-Quantum Cryptography. PQCrypto.Lecture Notes in Computer Science, 2014, vol. 8772. Springer, Cham. [CrossRef]

25. P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring.” Proceedings 35th Annual Symposium on Foundations of Computer Science 1994: 124-134.
26. L. K. Grover, "A fast quantum mechanical algorithm for database search". Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, 212–219. [CrossRef]
27. C. Figgatt, D. Maslov, K. A. Landsman, , N. M. Linke S. Debnath, & C. Monroe, "Complete 3-Qubit Grover search on a programmable quantum computer", Nature Communications, 2017. Vol 8. [CrossRef], [PMid], [PMCid]

AUTHORS PROFILE



Dr. Joshua J. Tom received his Ph.D. degree in Computer Science with specialization in Cyber Security from Federal University of Technology, Akure, Nigeria in 2018. He is currently a research fellow and lecturer in the Department of Mathematics and Computer Science, Faculty of Basic and Applied Sciences of Elizade University, Ilara-Mokin, Nigeria. His current research interests are information and cyber security, cryptography, artificial intelligence and machine learning, security of cyber-physical systems, internet of things and related technologies. He is a member of several professional bodies including Cyber Security Experts Association of Nigeria (CSEAN), Nigeria Computer Society (NCS), Computer Professionals of Nigeria (CPN), and African Academics Network (AAN).

Dr. Nlerum Promise Anebo currently works at the Computer Science and Informatics Department, Federal University Otuoke, Bayelsa State. He holds BSc, MSc and PhD degrees, all in Computer Science. Dr. Promise is a Senior Lecturer and the former Acting Head of Department, from October 2016 to October 2021. His current Project is 'An Efficient Coordination Language for Pervasive Computing Systems. He has special interest in Machine learning, Software Engineering, Pervasive Systems and Mobile Computing.

Dr. Bukola A. Onyekwelu holds a Ph.D. in Computer Science from the Federal University of Technology, Akure, Nigeria in 2015. She is a lecturer in the Department of Mathematics and Computer Science, Faculty of Basic and Applied Sciences of Elizade University, Ilara-Mokin, Nigeria. She is a member of several professional bodies including Society of Digital Information and Wireless Communications (SDIWC), Computer Professionals of Nigeria (CPN), Nigeria Computer Society (NCS), Nigeria Women in Information Technology (NIWIIT), Organization for Women in Science for the Developing World (OWSD).

Adigwe Wilfred is a PhD holder in Computer Science he specialized in Data Communication. He has been a lecturer for eighteen years and has held the following positions to credit - two times HOD of Computer Science, acting Dean Faculty of Computing, presently SIWES Director of the above mentioned University

Richard E. Eyo Richard holds MSc. in Information Security with specialization in Smart Cards, Tokens, Security and Applications from Royal Holloway, University of London, United Kingdom in 2017. Post Graduate Diploma in Computer Engineering from University of Uyo, Nigeria in 2015, Higher National Diploma (HND) in Computer Science from OSISATECH Polytechnic, Enugu, Nigeria. Richard attended Cybersecurity Assurance Specialist Program, Baxter Clewis Training Academy (U.S.A) 2021, specializing in Payment Card Industry Data Security Standard (PCI DSS). Richard is a member of Chartered Institute of Information Security (CIISec). He is an experienced IT/Support Specialist presently working with River strong, USA.