

Lattice-Face Key Infrastructure (LFKI) for Quantum Resistant Computing

**USC Incubator 1225 Laurel St. Columbia, SC 29201

LokDon security research project:

**Josiah Johnson Umezurike | jumezurike@lokdon.com | September 12th 2018

Keywords: OTP, Qubit, SVP, CVP, Lattice-basis, 2048 Bits, AES, Cryptography, QR, QI, Blockchain

Abstract

In this paper we present a new idea by exploring a hybrid system designed to share symmetric and asymmetric properties. LFKI is code named, end-to-end cryptographic system for cloud, mobile, internet of things (IOT) and devices (ECSMID). Until now, there had not been much done on lattice faces as a hybrid cryptographic solution. Here in, we do not owe respect to only randomization reduction or deterministic reduction. We embrace a collective approach to defining the old age question of what problem is hard enough in NP to resist a quantum assailant. These biases, especially non-deterministic reduction is used to show that lattices are interesting hard problems within the set of NP Complete problems. Though the shortest vector problem (SVP) seems promising. It is nearly enough to facilitate and establish lattice basis an exception from the priori art [1]. The many configurations of their vertices seem to dismiss the wonderful properties of the dynamic faces that abound in various lattice constructs. The elements of these faces found in between regions bounded by the vertices and edges are of great interest to cryptography. When represented as numerical values serve as mathematical images of the lattice basis distribution. It is demonstrated that each vector representation has the potential to generate cryptographically secure number of keys. They follow a somewhat rigid rule; deterministic and yet a chaotic arrangement of the lattice vectors represented within a matrix of column (c) and rows (r), where ($c \geq 16$ and $r \geq 16$). A fitting rule is already available with the necessary mechanism to produce 1: n relationship for a plaintext against many ciphertext. This is found in Open/Knight Tour (O/KT) movements and can easily be modified to absorb larger lattice basis. Lattice face are ready made with properties that are closely related to the regular vectors of Euclidean space.

Introduction

This article is an observation from over 20 years research work. The work is not done by a mathematician but by a security professional. The sole intent is to solve the common problem of our time from practitioner' perspective. It is agreed on all grounds, the havoc quantum computing will bring to the modern cryptography. Consequently, it is sufficiently relevant to be prepared pre and post quantum. The understanding of Euler, Hamiltonian cycle and lattice basis paved the way in drawing the relationship needed to harmonize the open knight tours (OKT) in the genre of Hamilton' path. The similarities under study shows the pervasiveness of Hamilton's path in grid ($n \times n$) formation. In absence of any back track; It does enumerate all points in Euclidean space if and only if $n \geq 5$.

Hamilton' cycle, when applied to grid or chessboard, it clearly proves that it is indeed a hard NP as the grid become richly connected. When this exercise extended to the operation of AES (Rijndael) which commonly lies on 4×8 grid. It is possible to expand the scope of AES to develop a 2048-Bit AES-hybrid using the bounded region between the edges and vertices of a lattice face. The result is a low-cost, high entropy, endpoint to endpoint cryptographic system for cloud, mobile and IoT devices (ECSMID). The reference specification is a category of hard NP problems closely related to numbered faces of a lattice basis or matrix. This cryptography shows the properties of both symmetric, asymmetric cryptography or public key infrastructure (PKE, KEM and DS).

OBJECTIVE:

To show that there is a cryptographic formation following a lattice basis that fits into an ideal set of hard NP complete problems known to be resistant to quantum computing. A matrix could be observed as a numerical image of a lattice basis to bring about a low cost, pervasive and high entropy cipher which hybridizes and increases AES capacity to roughly 10 times. Thereby, resists post-quantum attacks and cancel the effects of pre and post-quantum breaches.

An Overview of current cryptography

The Frailty of PKI and AES: There are numerous talks about PKI. Ponemon institute, Gartner, IBM and many other reliable and prolific sources had mentioned their worries about the future of PKI as we know it. More so, PKI and AES are the dominant part of the mechanism securing the internet transactions of today. The banks, health, retail, government and all entities use these two pieces of technology.

They are supposed to secure and make private each communication whenever you access any secure website. It is a scientific knowledge that PKI is based on mathematics:

Where in, $(N = p * q)$, $\phi(N) = (p-1) (q-1)$ where (e, N) is the public keys and (d, N) is the private key.

There is a condition $e \{integer; 1 < e < \phi(N); co-primes (sharing no factors) with N \& \phi(N)\}$. Choose d such that,

$\{ed \bmod \phi(N) = 1\}$:

In the wake of these problems are many proposals for the direction of modern cryptography. There are:

1. Lattice basis cryptography
2. Code base cryptography
3. Multivariate cryptography.

Some of these are the second runners up of NIST's call for paper in cryptography 2019. This means that they are still being considered in the second round of NIST standardization for modern cryptography. It is a scientific fact that any mathematical problem is there to be solved. One can clearly the reason for NIST call. However, considerations were only given to the use of these schemes to deliver AES keys following the 50year old trend or tradition.

This means that our crown jewel cannot depend on any mathematical function based on Fermat's theorem or any other. To achieve the desired goal. A favorable design will be that which references a quantum cryptographic model (QCM) as a relevant strategy for securing the internet in times to come. Otherwise, quantum computing will wreak havoc on modern day cryptography whenever it finally gets into the hands of consumers. Let us take a serious look at what a lattice really means in a mathematical sense of it.

The Problem:

If anyone can obtain the factors of the large number N with d (public key) any message will be decrypted. At the time of writing it is known that RSA is cracked. You should also note that Quantum computing has the potentials to solve the

math and/or crack these large primes (N) in a short period of time according to Shor' Algorithm [2]. The time to perform the feat is usually said to be in polynomial time. In that case the RSA math show earlier, will no longer be a hard problem of a non-deterministic polynomial (NP). Qubit is the stable standard signal state of a quantum computer: Again, the development of any quantum resistant algorithm could not afford to dismiss that notation typical to a qubit. In fact, one cannot neglect this idea and it cannot be over emphasized. This new qubit factor will also render any form of primitive cryptography useless. Another problem arises with the periodicity of lattice constructs. That begs the question. Is there a way to infuse the lattice with enough diffusion that will trigger more than translational changes in bases to bring about entropy and complexity so dynamic that it will be impossible to decipher the permutation of the basis transformation?

Another serious problem that we must consider lies in the diffusion of AES.

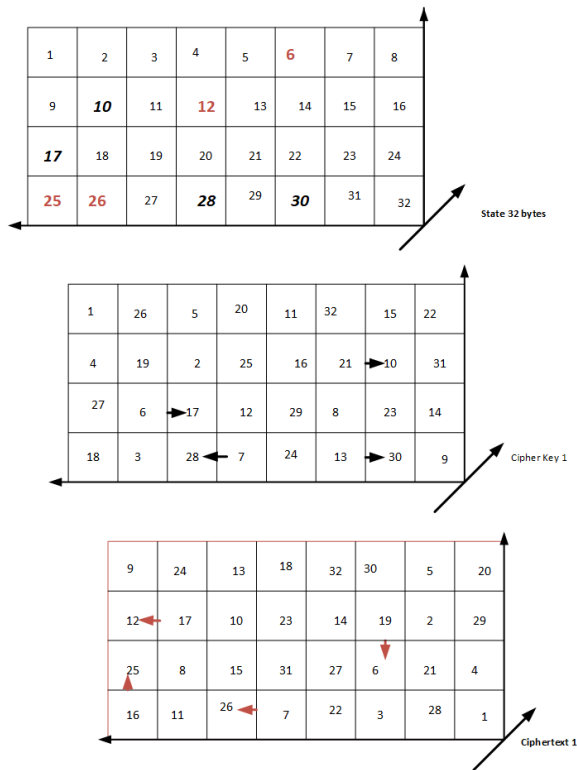
1. Maximum distance separable (MDS) matrix introduced by Shannon Claude. – This is an identity matrix multiplication of A to produce a transformed matrix A;

Let the matrix $\mathbf{A} = \frac{I}{A}$, by joining identity matrix I to A. This introduces an invertible linear transformation.

2. Ciphertext produced after the transformation of AES will have just a key to lead to a plaintext.
3. The constant nature of the produced ciphertext will make it

susceptible to byte-wise brute-force attacks.

According to the substitution box (S-box) AES. This matrix is based on Rijndael finite field. An attack byte-wise will be successful today. One will have about 256 options or combinations for each byte in the 32byte arrangement of 256 bits block of AES. This means that in a 256 bits AES block size. Each sub block contains at the most 4 bytes or 32 bits. There are about 256 elements that could fit in one sub block at some point. This sub block will have about (256 s-box elements * 32 sub blocks) = 8192 combination of s-box elements or 32bits in it. Here 32 bits is equivalent to double words. Attacking bitwise could seem hopeless at first but going about the attack from byte perspective is another smart way achieving the same goal. A study of the diagram in **fig. 1.0** will quickly show why attacking in byte-wise will be successful. It is much better to integrate the bits in a way to have more structure or visual patterns.



Brute force analysis of AES 256
Fig 1.0

SOLUTION TO THE PROBLEM:

Observation of the weaknesses of AES is the motivation behind this paper. A generous diffusion of semantically unsound messages was presented on behalf of AES. Going by the facts known about AES, such entropy cannot be accomplished in less than 14 rounds. In fig. 1.0 notice the patterns in cipher key 1 and ciphertext 1. This is clear by a visual comparison to state, 32byte of structured messages. After the completion of the 14th round of AES, the resulting output is ciphertext₁. Imagine using the S-Box directly with 256 elements where each matrix cell has the capability of 32bits or more. Lattices in their natural forms explain this abstraction in a new light. What then is a lattice?

A lattice is a set of all integral linear combinations of a given set of linearly independent points in \mathbf{Z}^n . For a basis $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_d\}$ we denote the lattice it generates by $L(\mathbf{B}) = \{\sum_{i=1}^d x_i \mathbf{b}_i | x_i \in \mathbf{Z}\}$. Its rank is d , and the lattice is said to be of full rank if $d=n$. We identify the basis $\{\mathbf{b}_1, \dots, \mathbf{b}_d\}$ with the $n \times d$ matrix containing $\mathbf{b}_1, \dots, \mathbf{b}_d$ as columns, which enables us to write the shorter $L(\mathbf{B}) = \{\mathbf{B}\mathbf{x} | \mathbf{x} \in \mathbf{Z}^d\}$. We use both the terms lattice point and lattice vector to describe the elements of a lattice basis.

A bit more time will be spent to introduce a new insight, towards quaint understanding of the shortest vector problem (SVP) of graph and path. While it is generally a consideration for being an NP-Hard problem. Randomization reduction without considering the face holding the basis is not enough to establish this as a case of NP-Hard problem [3]. That alone, could have been insufficient, or not good enough for quantum resistant encryption. Quantum computers are created to solve mathematical problems impossible to human mental speed. If a lattice basis must be retained as the frontier of modern cryptography; it must be an interesting one with some elegant properties that could be reduced to randomized, non-deterministic and deterministic biases. The intention is not to be overly critical. There is a need to be proactive. One's intent will be to find the right solution out of many; not to accept a solution that is not ripe. This is neither to wait for a solution to present itself. A potent lattice or ideal lattice and its image must be dynamic, with certain rigid rules, yet precise in decision making affirmative to the output. It must possess a distribution of probabilistic basis transformation with respect to the input

and output (references are made to homomorphic encryption). -- The image of the lattice basis is a bounded matrix of interest. Let us look, once more into the relationship between encrypted and decrypted messages.

Asymmetric:

$$\text{Encrypted data (c)} = \text{msg}^e \bmod N$$

$$\text{Decrypted data (msg)} = \text{c}^d \bmod N$$

It is clear from the above that an assailant only requires (d, N) to decrypt the message. Although, this does not apply to AES in mathematical sense. It quietly applies in byte-wise brute force of AES cipher key.

Symmetric:

Let Ct = cipher template length; where the length is the same as the keys used to perform wholistic encryption of the message. The message is added to extended key K of period D which could be a 64 bits passphrase or more. Note that, a modulo arithmetic (XOR) is used herein. It is a common knowledge that AES is one form of the family of symmetric key cryptography. The strength of AES is synonymous to the irreducibility of polynomials of GF (2⁸) or 8th degree. Symmetric key cryptography (SKC) uses a secret key: They are commonly known as passwords or passphrases and mostly manual driven. It is interesting to note that the key used to perform the actual encryption in AES sometimes are derived from these passwords via key derivation mechanism capable of a pseudo random number generator (PRNG). Password based key derivation function (PBKDF-2) is a good example.

The block sizes of AES are defined (128bits, 192bits and 256bits). These and many other reasons add to their weaknesses before quantum computing brute force attacks. PRNG will generate AES keys of 16, 24 and 32 bytes to match the block sizes respectively. If the message doesn't fit the block. It is then padded with IV so that it will fit the chosen block. Grover's algorithm is a quantum algorithm that finds with high probability the unique input to a black box function that produces an output of a defined value, using just ($O\sqrt{N}$) evaluations of the function, where N is the size of the function's domain. Despite the effort vested in making AES secure, Grover is saying that it is probable half the time, to brute-force AES – 128 in 2⁶⁴ iterations. At least, one can unravel useful information that will lead to breaking of such scheme using quantum computer as a level playing field [4]. Here in, it is implied that the time is in quantum domain not polynomial time.

ECSMID proposes the use of seeds in social security numbers, driver license number and phone numbers. It is recommended to use 10-20 digits number arranged in one order. These numbers could be picked off vectors capable of becoming seeds for generating 680 digits number from each position on the matrix of n * n. We will talk more about this on another paper. These data are fed into the algorithm just like traditional classics and/or primitives of today.

$$\text{Encrypted data (c)} = (\text{msg}, D): (\text{msg xor } D) \bmod \text{Ct}$$

$$\text{Decrypted data (msg)} = (\text{c}, D): (\text{c xor } D) \bmod \text{Ct}$$

AES cipher substitution is directly derived from the substitution box (S-Box).

M₁

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

SL-M₁

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

LFKI default matrix is 16*AES Fig. 2.0

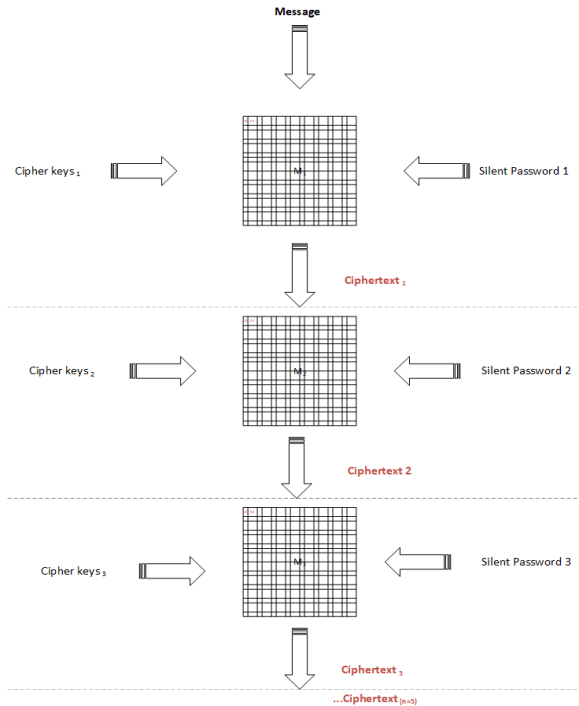
In Fig. 2.0 transformation of LFKI is completed via the natural occurrence of the lattice basis which under observation was noted analogous to OKT. One can say there is a similarity between the new protocol and AES. It is necessary to perform an exercise; a proof by visualization. This will help us to clearly establish without a doubt these claims. We will attempt to prove this abstract connection, not only to dispel doubts, but to deepen understanding of the trail modern cryptography could blaze.

AES will suffer a similarly if not the same fate as RSA. If we do not apply this new mechanism. The future of quantum computer will certainly vilify it as well as any other contraption that does not comply to the dynamic reality of quantum computing model (QCM). We don't really

have to wait into the future anyway. People are already saving petabytes of data in the cloud. In due time these could be disclosed as soon as quantum computer becomes available.

AES will suffer a similarly if not the same fate as RSA. If we do not apply this new mechanism. The future of quantum computer will certainly vilify it as well as any other contraption that does not comply to the dynamic reality of quantum computing model (QCM). We don't really have to wait into the future anyway. People are already saving petabytes of data in the cloud. In due time these could be disclosed as soon as quantum computer becomes available.

The minimum OKT necessary is 5 modes. Which simply translates to 5 different ciphers because the state is mapped or substituted to the knight template (KT_n) to produce cipher templates (CT_n). Structured messages are fed into the CT_(n≤5). This method will accommodate future changes in code points and or code units. This will cover any future changes in the value sizes for types e.g word (WORD), double word (DWORD) and quantum word (QWORD).



Dynamic LFKI rounds Fig. 3.0

One of the major advantages observed in this paper is the fact that all other inputs in the circuit change except for the original message. This is a superior mechanism to surpass the challenges of quantum superposition.

Technical Specification

To solve this problem from a technical perspective. It is imperative to draw an analogy from 3D shapes and their properties: Especially surface area (face) with breadth. A cuboid and other favorable dimensions of lattice basis will suffice for this development. Their properties like face, edges and vertices come in handy in unbounded and bounded space. You can get a flux from these properties as a result of vectors forming regular point in Euclidean space to enhance orientation as seen in lattice basis. In programmatical (code) terms, the idea of a matrix transformation:

Translation, transposition, and substitution serves us well by forming an algorithm that covers lattice face key infrastructure and architecture. The face on lattices are commonly known to have points or vectors. Same goes to a matrix which is a quantitative representation of the lattice following certain strict rules. Therefore,

$$\text{Total flux} = \iint \mathbf{f} \cdot \mathbf{n} \cdot d\mathbf{S} \text{ [where } n=1\text{]}.$$

For our purpose the vector accent will not be needed. As scalar and vector delineation blurs in the region of SVP.

It means that any normal face in a shape will have a regular arrangement of point in Euclidean space (lattice). In this sense, following the elements of Galois' field; a matrix, mathematically can hold a lattice's contents: It is then noted that a lattice is only a form which can be reflected or translated. It will have points upon which forces can interact with it. This means that changes in choosing any of these points could change the matrix or the indices they bear. Below is the explanation of informational entropy.

Mathematically, this is expressed as $H(C) = H(M|C)$, where $H(M)$ is the informational entropy of the plaintext and $H(M|C)$ is the conditional entropy of the plaintext given the ciphertext C . This implies that for every message M and corresponding ciphertext C , there must be at least one key K that binds them as a one-time pad. Mathematically speaking, this means $K \Rightarrow C \Rightarrow M$, where K, C, M denotes the distinct quantity of keys, ciphers and messages. In other words, if you need to be able to go from any plaintext in message space M to any cipher in cipher-space C (encryption) and from any cipher in cipher-space C to a

plain text in message space M (decryption), you need at least $|M|=|C|$ keys (all keys used with equal probability of $1/|K|$) to ensure perfect secrecy [5].

It is also a standard practice to increase entropy by introducing seed candidates capable of deriving PRNs, silent noise (passphrases or codes) penetration and other manipulations that permeates cipher text to remove structure in plaintext (original message) cause entropy in the ciphertext. This could be achieved through modulo arithmetic by adding (XOR-ing) numerical values of passphrases (characters, special characters) of the UTF-8 to the original messages. This will be fully explained later in this paper. According Shannon, the common knowledge of entropy is in the information content H_x of a value x that occurs with probability $\Pr[x]$ is

$$H_x = -\log_2(\Pr[x]).$$

The entropy of a random source is the expected information content of the symbol it outputs, that is

$$\begin{aligned} H(X) &= E[H_x] \\ &= \sum_x \Pr[x] H_x \\ &= \sum_x -\Pr[x] \log_2(\Pr[x]). \end{aligned}$$

It is submitted in quality from observation: That it is not a common knowledge to think of $n=\square$ with respect to the equation of Galois field $\mathbf{GF}(2^n)$ which essentially claims its validity from Euclidean space. In grammatical (code) terms, the idea of a matrix translation, transposition, transformation and substitution serves us well by forming an algorithm that covers

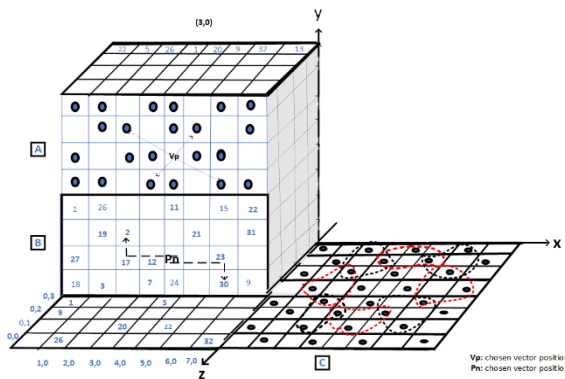
lattice face key infrastructure and architecture. Imagine that, this is in opposition to present day symmetric cryptography limiting scopes.

In cryptography this means that those points can represent encryption and decryption components of data by satisfying $\mathbf{GF}(2^n)$ where $n=\square$. The flux analogy herein depends on the surface area or orientation of the shape and forces (analysis) on them. The changing flux will be likened to the changing entropy at every turn of the algorithm (operation) owing to noise. The total flux is the product of the basis surface area, force and normal vectors. It is therefore possible to create a system of quantum immunity or resistance for the computation by replacing the vectors or points with characters of written words. Carefully chosen, are certain Unicode characters (i.e numbers). These formulate the standard state (ST): Subsequent generation of numbers from these face/s or seeds, following position $P_{(n=0)} - P_{(n=255)}$ give rise to other sets (680 digits long) which could be used as cipher templates (CT). These points become numbers generated from the chaotic regularity found in faces of sky, snowflakes and silicon shapes (of course in 2D and 3D).

The proposed algorithm comes with a powerful wrapping (Mode₁ – Mode₅ or M₁-M₅) mechanism. That's what makes it possible to be used as an exchange channel in the order of PKI public and private key. However, the school of thought defers from the popular opinion of Shortest Vector Problem (SVP) associated with the current lattice basis solution for cryptography. It is deduced from the research that open knight tour on a lattice face is a harder NP problem

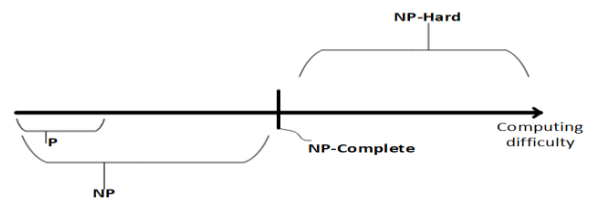
than the notion of SVP [6]. It cannot be solved by a quantum computer as long as the matrix is equal and greater than 16 for the columns as well as the rows: Given a matrix in column major (c,r), where a full rank is $n \times d$. let $n = d$. It follows that $\{c\}=16=\{r\}$.

Note: The point closest to the chosen vector in SVP is orthogonal to all other points of interest. Finding the shortest path is the reason why this problem is of interest to cryptographers. This can never be deduced with certainty needed for integer mathematics. In lattice diagram 'A' fig. 1.0 the periodicity is very clear more so, all points sought to determine the shortest vector path are orthogonal. Now, look very closely at lattice diagram 'B' fig. 2.0 The periodicity is also clear as the denoted impression in 'A'. Although the bases are replaced by numbers just like a matrix would have within. When the numbers or the lattice bases are rearranged. A measure of difficulty arises in a way the problem becomes harder. The path to finding the shortest vector is no longer a linear one. Or is it? By Pythagoras it still is orthogonal respecting the base orientation.



Lattice base and matrix mix Fig. 5.0

A quantum Turing-machine with qubits orientation cannot sniff with certainty the positions of any legal open knight tour (OKT) on lattice face if the column (c) and the row (r) of the matrix are respectively of $c \geq 16$ and $r \geq 16$. If the position (P_n) that generates any set of cryptographically secure keys is unknown. If any set of keys generated from the matrix positions (P_n) follow $n!$ where $n \geq 256$ is unknown. If comparing any two positions (P_1) to (P_2) on the lattice does not sniff out similar 680-digit long keys. Giving any input, it is said that the decision is impossible. Else, this is probably the hardest NP problem and will not resolve in polynomial-time.



Computational difficulty Fig. 4.0

$P \neq NP$ and no one is sure of $P = NP$ as it is not polynomial resolvable as earlier explained. In corollary, one can find a common NP-Hard problem which allows similar inputs as the OKT. In that case lattice basis are best suited for this reduction. Let X represent a lattice with regular point(s) in Euclidean space.

It is agreed on equal footing that Hamiltonian path and open knight tour (OKT) are NP Complete [7].

It is also a common knowledge that the Shortest Vector Problem (SVP) of a lattice-based cryptography is an NP-Hard problem. See Ajtai works for details. We will only try to reduce the hard

problem to NP to prove that OKT is equally a hard problem.

To prove that OKT is a hard NP problem: We only need to re-state the theorems. We will follow these steps:

- 1) We deduce that **X ∈ NP**
This could be done in (i) or (ii)
 - i) Polynomial time algorithm
 - ii) Certificate and verifiers
- 2) Reduce from known NP to the problem Y to X.
 - i. If Y ∈ P the X ∈ P
 - ii. If Y ∈ NP, then X ∈ NP

X not in P unless P = NP

X is NP Complete if X ∈ NP & X is NP-Hard.

X is NP-Hard if every problem Y ∈ NP reduced to X

In this case inputs for X and Y are the same e.g coordinates, **Vp** or **Pn**. There will be no polynomial time algorithm for this proof. There is still a known problem 3DM (**§**) that is NP-Hard. If we can fit this problem into **Y**, then **Y** too is NP-Hard.

Proof: Y is NP-Hard

Given: 3D matching (variable gadget). Disjoint set x, y, z each size n given triples $T \subseteq x * y * z$.

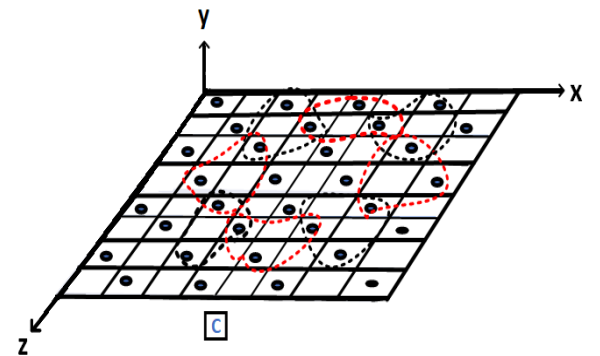
Is there a subset $S \subseteq T$ such that every element, $\epsilon x \cup y \cup z$ is in exactly one, $s \in S$. Following a legal knight move OKT could only be on black dot (Y) or white square (N) at once?

Method: Reduction of X to Y.

Three-dimensional matching (3DM) is NP Complete (Theorem). It is going to be a graphical prove. To make this easy, we set up an 8 by 8 matrix of black dots and white squares. See diagram C **Fig 4.0**. Let X represent the lattice basis (SVP) and Y represent OKT. Lattice basis (SVP) had been reduced to NP-Hard problem earlier[8]. Other precedence, 3-SAT was reduced to 3DM [7].

To Prove:

§ ≤p Y (If we could solve **§** we could solve Y).



3DM reduction to OKT Fig. 4.0

It is noted that in a deterministic Turing machine the answer is in affirmative for all inputs following the algorithm. You have the graph and the path to trace. This is quite analogous to the knight on a standard chess board. This same analogy is akin to non-deterministic mechanism given any input for decision of Y (black) or N (white). In this is more like a black dot or white square.

Following a certain strict rule which compels the knight or the input to touch on one of two (2) nodes if at the vertex (corner); four (4) nodes if on the edges and eight (8) nodes if at the middle of the board. It will trace the path to the nearest

node no backtrack is allowed. This solution could go in a loop within a changing or expanding bases.

Open knight path traced from any corner of $n \times n$ graph will have 2^n nodes of connection for 3 moves at the most. This is counted from $n=0$ position (Initial point) where $n=0$ is not really a move.

1. The assumed position (P_n) on the corner is not counted as the first move such that no move is considered for initial position $n=0$. This means that the number of nodal connections at any chosen path will have 2^n nodes; where $0 \leq n < 3$. Only one node will be activated to move on to the next point of decision in the path. This is how the numbers are generated.

End of proof:

The open knight tours satisfy the condition of 3DM where in, a response of true (Y) or false (N) is entered to satisfy that only one element of the triplets could be held in T. If the path found for the legal knight is correct. The clause must be black dot else white square. The path of a legal move, is a certificate which the machine must verify (one can also say that it is polynomial algorithm satisfied by the input and output of instruction sets) by counting black as a YES or white square as a NO. This method does not need to worry about garbage collection in the circuit for fear of tautology [9].

Notice a clear demonstration that there is no need to perform the garbage collection technique in this implementation reference. The said technique is performed as an extra layer

in the reduction to show that the set \mathcal{S} will connect to every other dot in Euclidean space. Relying on the above claims, premises and theorems we submit this reference specification of an algorithm that combines symmetric and asymmetric cryptography using zero knowledge triangle flow and homomorphic encryption, standing strong enough to resist attacks from quantum computing. - Lattice-Face Key Infrastructure (LFKI)-- It recognizes and applies:

a) public key encryption - 2048 bits AES-hybrid is used for encryption in wraps or modes

b) key encapsulation - positions of key sets are encrypted with msg and separated

c) digital signature - attributes are formed and stored as encrypts (HE properties are used)

d) Hashes are not used in the classical sense for authentication: They only suffice for initial plain text integrity (digest) check

e) CRC or checksum is not pushed here because of HE: If the hashes match, the original plaintext is the same as the current one.

The minimum modes for any encryption done is usually 5 or M_5 for this system. However, you can encrypt anything (a message etc.,) from M_1 to M_n^{th} . This security could be applied in telecommunications, cyber physical systems (CPS), IoT, information technology (IT), aeronautics, lithography, medicine and health, retail, finance and

education. This will be the hybrid of all times.

Infrastructure of LFK

It has an elegant, simple and easy to implement approach. Our social mode of interaction on the media had made possible for us to easily figure out what works. Many profiles today are comprised of attributes. Therefore, we reduce data into certain groups for seemingly public key implementation.

Digital Nucleus Aggregator (DnA): These are attributes that can be converted to encrypted strings for various intermediate representation in the digital space. e.g Name, SS#, eFRI, DOB, PIN, Address, password Gender, Driver license# etc. It could be anything of your choosing. Profiles rely on DnA as their building blocks for intermediate representation in this reference. DnA are derived from profiles attributes as we will demonstrate later.

Digital Data Nucleus Authority (DDnA): These are integration of multi DnAs. This could be held locally or externally in a data base or function-running code platform such as lambda in aws cloud. The architecture creates a data bank as good as a phone book of today. This is where all the intermediate representation could be found in encrypt forms following a homomorphic encoding or encryption algorithm.

Architecture of data

Let's revisit the phone number as a seed input: There are many orderly ways to pick out 2 distinct numbers from an arrangement of 10 digits--> 788 890 6754.

However, we will first calculate the arrangements with repeats in 788 890 6754. We start with:

8's

Let $n = 10$ and $k = 3$

$$nPk = 10! / 3! = 604,800$$

7's

let $k=2$

$$1/2!$$

$$\text{distinguished arrangement} = 10! / 3! * 2! = \mathbf{1,209,600}$$

The above means that there are **1,209,600** ordered ways of arranging

7888906754

...

8889067547

8890675478

8906754788

,...

9067547888

,...nth

Furthermore, one can arrange these numbers in twos. What is the arrangement of choosing from 10 two digits (0-99) in five different sets? If we must arrange these numbers in five sets of twos. It will be another $(10*9*8*7*6*5*4*3/2!)/5$ Ways or distinguished arrangement = **181,440**

If and only if all two digits are distinct.

78 88 90 67 54

...

88 89 06 75 47

88 90 67 54 78

89 06 75 47 88

,...

90 67 54 78 88

...nth

Each of these numbers could be used as seed for 680 digits long encryption keys: They become offsets and are only made ready when needed.

There is a whole algorithm to address non-repeat of the said digits of numbers and that is not within the paper's purview. Rest assured no number is repeated in the algorithm. Each of these 2 distinct numbers (seeds) from the 10 digits number arrangements are found on the matrix as positions (P_n). They will further generate another 680 digits long numbers following the certain algorithm. The 680 digits long numbers will be used as the encryption keys. Normally 5 sets of 680 digit long from $P_{n=1} \dots + P_{n=2} \dots + \dots + P_{n=5}$ are needed. At least, for the proposed reference implementation. Each position generates a one-time set of 680 digits numbers. In fact, the idea is richly emphasized in this paper.

Full M_5 mechanism This method could operate on any DnA propped by any

attribute. Note we will demonstrate DnA using password as input. We will also demonstrate volumetric data scheme using the message and any DnA as input for this algorithm.

You can also use the message C in place of the password.

Password + silent password = $CT_1 \Rightarrow$
 $M_1 \text{ encrypt} \Rightarrow [ciphertext_1]^{[P \text{ spktn}][P \text{ kt}_n]} = M_1$

$CT_1 + \text{silent password} = CT_2 \Rightarrow M_2$
 $\text{encrypt} \Rightarrow [ciphertext_2]^{[P \text{ spktn}][P \text{ kt}_n]} = M_2$

$CT_2 + \text{silent password} = CT_3 \Rightarrow M_3$
 $\text{encrypt} \Rightarrow [ciphertext_3]^{[P \text{ spktn}][P \text{ kt}_n]} = M_3$

$CT_3 + \text{silent password} = CT_4 \Rightarrow M_4$
 $\text{encrypt} \Rightarrow [ciphertext_4]^{[P \text{ spktn}][P \text{ kt}_n]} = M_4$

$CT_4 + \text{silent password} = CT_5 \Rightarrow M_5$
 $\text{encrypt} \Rightarrow [ciphertext_5]^{[P \text{ spktn}][P \text{ kt}_n]} = M_5$

When an offset is added to the length of the encrypted message C or CT (ciphertext). That no longer represents the length of the message. Rather a periodic random key D is used to match the length of the message. This does not void the condition of the classical stream cipher requirements: Superficially, each byte of the plaintext and ciphertext are one to one function (bijection) since both share similar length as the key size. However, a detailed observation proves a distribution that shows n numbers of ciphertext for any plaintext. There is an introduction of randomization by using some random string (silent password (SL) as used randomly in this reference).

This increases the entropy of key length bearing a perfect secrecy [10]. Especially the one-time pad scenario cannot outlive the philosophy:

"Perfect secrecy is a strong notion of cryptanalytic difficulty".

Also note that in as much as the keys are seeded and generated. The dynamic distribution scheme of these keys makes certain; no expended key will be generated from the faces of the lattice position (P_n) or the matrix. And neither will the generated keys be used again. Every 680 long key is used just once. Let's explore volumetric data scheme in this algorithm. We are XOR-ing the message with the modular PIN (MPIN). A PIN is naturally 4-6 digits numbers. In this reference two characters represent each of the PIN numbers making the overall characters $2 * PIN$.

$$\begin{aligned} \text{Data} + \text{MPIN encrypt} &= \text{CT}_1 \rightarrow M_1 \\ \text{encrypt} &=> [\text{ciphertext}_1]^{[P \text{ kt}_{n=1}]} \wedge [P \text{ spkt}_{n=1}] \wedge [P (\text{MPIN}) M_{n=5}] \\ &= M_1 \end{aligned}$$

$$\begin{aligned} M_1 + \text{MPIN encrypt} &= \text{CT}_2 \Rightarrow M_2 \text{ encrypt} \\ &=> [\text{ciphertext}_2]^{[P \text{ kt}_{n=2}]} \wedge [P \text{ spkt}_{n=2}] \wedge [P (\text{MPIN}) M_{n=5r}] \\ &= M_2 \end{aligned}$$

$$\begin{aligned} M_2 + \text{MPIN encrypt} &= \text{CT}_3 \Rightarrow M_3 \text{ encrypt} \\ &=> [\text{ciphertext}_3]^{[P \text{ kt}_{n=3}]} \wedge [P \text{ spkt}_{n=3}] \wedge [P (\text{MPIN}) M_{n=5}] \\ &= M_3 \end{aligned}$$

$$\begin{aligned} M_3 + \text{MPIN encrypt} &= \text{CT}_4 \Rightarrow M_4 \text{ encrypt} \\ &=> [\text{ciphertext}_4]^{[P \text{ kt}_{n=4}]} \wedge [P \text{ spkt}_{n=4}] \wedge [P (\text{MPIN}) M_{n=5r}] \\ &= M_4 \end{aligned}$$

$$\begin{aligned} M_4 + \text{MPIN encrypt} &= \text{CT}_5 \Rightarrow M_5 \text{ encrypt} \\ &=> [\text{ciphertext}_5]^{[P \text{ kt}_{n=5}]} \wedge [P \text{ spkt}_{n=5}] \wedge [P (\text{MPIN}) M_{n=5r}] \wedge [RM_3 \text{ MPIN es}] = M_5 \end{aligned}$$

Following the above process, the mpin (IR) encrypt shown is that of the recipients. If one is sending a message requiring ZKP. For example, M_3 mpin of position ($MPIN_{kt_{n=3}}$) is stripped and sent with the message:

$$M_4 + \text{MPIN encrypt} = \text{CT}_5 \Rightarrow M_5 \text{ encrypt} \\ \Rightarrow [\text{ciphertext}_5]^{[P \text{ kt}_{n=5}]} \wedge [P \text{ spkt}_{n=5}] \wedge [P (\text{MPIN}) M_{n=5r}] \wedge [RM_3 \text{ MPIN es}] = M_5$$

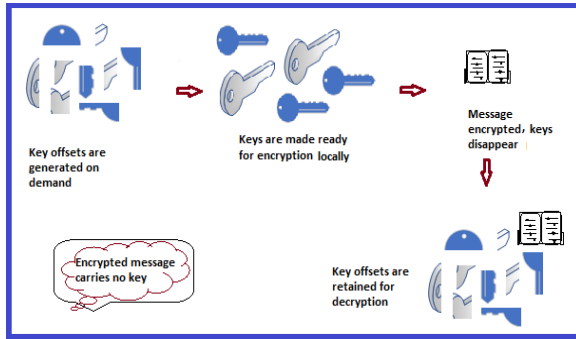
Note the removal of M_3 mpin key positions. $M_4 + \text{MPIN encrypt} = \text{CT}_5 \Rightarrow M_5 \text{ encrypt} \Rightarrow [\text{ciphertext}_5]^{[P \text{ kt}_{n=5}]} \wedge [P \text{ spkt}_{n=5}] \wedge [P (\text{MPIN}) M_{n=5r}] \wedge [RM_3 \text{ MPIN es}] = M_5$

On the receiver' device there is M_2 MPIN encrypt: $[M_2 \text{mpin encrypt}]^{[P \text{ Mpinkt}_{n=2}]}$ Note the replacement of the unstripped M_2 mpin with M_3 mpin keys' position $[M_2 \text{mpin encrypt}]^{[P \text{ Mpinkt}_{n=3}]}$.

In this order a polynomial attacker may never be able to go back to M_1 if at all they gain access to the network. M_3 mpin could be used as a digital signature of each user in the network. This can easily incorporate in any API.

Simply put

1. The M_3 PIN or any other mode chosen except for M_1 and M_5 will serve as the Public key and intermediate representation (IR) for (ZKP)*****
2. The seeding positions (P_n) serve the purpose of key encapsulation (KEM)*****
3. Signatures (reflecting biometrics this time) are infused in the IR of ZKP*****
4. Public key encryption or any encoding is borne within the scheme as a whole*****



LFKI kanban Fig.

5.0

C++ Package demonstration

1. KnightSolver.cpp (This solves the open knights tour with numbers >> OKT)
2. st.cpp (This is the unicode component order of written or spoken words >> ST)

<<96 chars for Latin-1 Supplement

<<4 chars for ASCII punctuation and symbols

<<26 chars for Lowercase Latin alphabet

<<6 chars for ASCII punctuation and symbols

<<26 chars for Uppercase Latin alphabet

<<7 chars for ASCII punctuation and symbols

<<10 chars for ASCII Digits

<<16 chars for ASCII punctuation and symbols

<<63 chars from Latin Extended A

All are totaled at 256 bytes (2048 bits)

3. revnum.cpp (Reverse the cipher template derived after mapping)
4. filecrypt.cpp (This does the mapping of ST to KT is done with this)
5. KnightCell.cpp (The instruction codes for the knight move is here)
6. main.cpp (this takes care of the implementation we desire)

Quantum Attack Diffusion

Solution for unmasking some secret string: Password, passphrase and keys favors a relative oracle fashioned in quantum computing model with particle superposition setup, following Hardaman's transform to simulate a equal superposition of two qubits (x). This will guarantee that the same output ($f(x)$) is obtained from the quantum circuit. Simon' algorithm [11] shows that it is possible to solve for the secret string (s) in polynomial time following his set of instructions for some Quantum Turing machine (QTM). The probability of not finding a linearly independent vector could increase from negligible

$$\prod_{k=1}^{\infty} \left(1 - \frac{1}{2^k} \right) = 0.288788 \dots > \frac{1}{4}$$

to substantial. In contrast, this notion further begs for a countermeasure that supports quantum resistance against any attack. This will be possible if certain conditions are built into cryptographic algorithms. The mechanism will do well if opposing dynamic strings are built into the algorithm regardless of the model in question. Find these conditions below.

Condition 1:

Dynamic or changing secret keys should be used for diffusion.

Condition 2:

Dynamic passphrases should be used as this follows 1.

Condition 3:

Dynamic ciphertext if and only if 1 and 2 are true.

Condition 4:

The unsoundness of the input (message) is satisfied and untampered for the given round.

These conditions trigger a corollary, that,

$$f: \{0, 1\}^n \rightarrow \{0, 1\}^n$$

such that there is a secret string

$$s \in \{0, 1\}^n, \text{ and } f(x) = f(x \otimes s).$$

Superposition is not always the case in quantum computer's supremacy just like a chain reaction is not always the case for a favorable outcome with nuclear radiation in non-sinister production. The critical points in both scenarios cannot be decoded with our crude instruments in our time-space reference. We know that at some point a quantum bits (qubits) could entangle. More so, atoms could engage in chain reactions of a highly radioactive system. – In this case the critical point is as destructive as one's inability to quantify the time of occurrence.

The point stressed here, is to reduce the generation of similar outputs by two bit-wise disparate inputs. This is the insight gathered from the research. It is important for us to note that a continuous output stream of linearly dependent

strings of a quantum circuit or any other circuit is needed for the unmasking of the semantically sound strings. Thus, consideration is given to a diffusion mechanism as the foremost approach running parallel to Simon's algorithm.

Where $f: \{0, 1\}^n \rightarrow \{0,1\}^n$; is bit-string output or input,

for $s \in \{0, 1\}^n$ there exist another secret string $t \in \{0,1\}^n$ such that $f(x) \neq f(x \otimes s)$

$$\leftrightarrow f((x \otimes s) \otimes t) \neq f(x \otimes s) \otimes t \neq g(f(x \otimes s) \otimes t) \otimes t \neq \dots f_{n-1}((x_{n-1} \otimes s_{n-1}) \otimes t_{n-1}) \dots$$

$$j(i(h(g(f((x \otimes s) \otimes t) \otimes t) \otimes t) \otimes t) \otimes t) \otimes t$$

(t) parallel in Simon' algorithm Fig. 6.0

In the simulation 2 qubits (2*4bit-strings) in a kind of superposition are used to demonstrate the basic modes. Any inconsistency or weaknesses there in is quickly spotted in this configuration. This insight designates with certainty a larger understanding of more complex configuration demonstrated in (16 * 16) bytes matrix where a basis colonizes (8-32 bits). One can improve the validity of this corollary with certainty by following the exact steps used in solving Simons' problem. The difference is the continuous introduction of diffusion with the secret (t) fed to the oracle at every turn in the circuit

or quantum circuit. It is clear that (t) will mask the output fast enough with enough certainty half of the times. Microsoft excel sheet was used to bring the idea to a larger crowd of non-mathematics population. Of course, there are many quantum simulation software like 'mathlab', QuEst, Qrack, Scaffold and many more.

ADVANTAGES (NEW APPROACH or AXIOMS):

1. $GF 2^p$ where $p \leq 8$; solutions are no longer bounded by irreducible polynomial of 8th degree. $GF 2^p$ is submitted under new conditions, where $p \leq 8$ & $p > 8 \mid \infty$ (or goes to infinity).
2. Non-Deterministic reduction insinuates that hard problem arises from $16 * 16$ matrix e.g We embodied OKT as a hard (NP-Complete) problem with other complexities and biases to derive ciphertext from cryptographic engine. It is also noted that this very system does not originate lattice base cryptography but shades light on the form.
3. Knight's tour (KT) could NOT be solved in polynomial time within unbounded field. A matrix of scope is of bounded field that could hold solutions of KT just like the elements of lattice basis. The changing nature of the nodes owing to the decision needed to advance to another element happened as a deterministic reduction. There is also a randomized reduction of seeding the key generators. The bigger the scope the more time it will take to negotiate and decide a fitting node just like in neural networks. With

this in view balancing symmetric stream of block (key) significant size, encryption time and implementation could yield cryptography of the future.

4. Similarly, AES exhibits the characteristics observed by the movement of the values held in the indices of GF of scope $16 * 16$ matrix or lattice basis. Each knight' tour opens at 0 position by tracing a clean sweep the elements of the matrix and closes at another position 255. Therefore, the new approach:

- a. Sub bytes
- b. Add round keys
- c. shift row
- d. mix columns

Using a mapping scheme of ST to KT and multi-mode-wrapping to achieve the afore mentioned states.

Irreducible polynomial is no longer a question of symmetric key cryptography for the fact that quantum computers will probably solve them. The new protocol is a non-suspect because it has no key schedule or invertible linear. To understand this context is possible to draw an analogy of 3-D space e.g a cube. A cube has faces (6), edges (12) and vertices (8). We are using the faces here: These are external to the popular context in cryptography. They have the largest set of vectors (numbers) vis-a-vis largest flux.

5. Cipher keys are no longer saved as they are generated from any position on the matrix (lattice face) upon request. Each position has a different set of numbers to be generated. 5 sets of (680 long digits) from 5 different positions are chosen from the matrix of 16×16 (256 bytes or 2048 bits). Attributes are chosen prior to be arranged into $n \leq 5$ different modes of encrypt for each attribute or payload fed into mode one all the way to mode five (M_1 - M_5).
6. The keys always change for any single message because the position on the lattice face changes as well. You can get started from any indexed point or vector. The origin 0 to any other part produces a different entropy flux. The order of these positions is seemingly regular (deterministic), they generate chaotic set of numbers. A new set of 680 digit long numbers. This knowledge reveals the changing nature of the message' ciphertext as well. When similar contents are encrypted the ciphertext are decisively different in the new order. Thus, hashing could only be necessary for cyclic redundant check (CRC) or message integrity check. $P \neq NP \parallel P$ not a subset NP.
7. The output or ciphertext from the message input in M_1 is used as input in M_2 . The ciphertext from mode two is used as the input in mode three (M_3). The ciphertext from mode three is used as input for mode four (M_4). The ciphertext from mode four is used as input for mode five M_5 . This protocol shows the characteristics of

homomorphic encryption mechanism (HE) [12]. The homomorphic encryption (HE) properties makes possible the flexibility of the algorithm (M_1 - M_5) as public key encryption management. These encrypts from this wrapping technique are used for ZKP.

8. The complexity is $O(N = \mathbf{message.length})$
9. Key encapsulation mechanism (KEM), digital signature and seeming public key encryption is built within the algorithm from the scratch. The change mode mix of attributes e.g MPIN, eFRI, Address and Password can give IAM operations facilitating god mode permissions in all kinds of environments with respect to business logic.
10. Plaintext to ciphertext relationship is (1: $n > 1$) number of ciphertexts: This is necessary to establish HE.

ASSUMPTIONS:

1. Modern primitives of cryptography only recognize 2S or 2 stable standard signal state. e.g 0/1
2. Post-quantum cryptography must recognize 4S or 4 stable standard signal state e.g various atomic state or photon' superposition.
3. We assume an ideal environment without anomalies the logic circuit.
4. We assume a high level of diffusion for AES.

Basic Analysis of QC & LFKI

We summed up axioms based on the current information and the implementation of modern cryptography.

Pre-quantum computing (Currently):

Encryption (bits)	Size of Dword (bits)	Stable standard signal state (unitless)	Block size (bytes)	State of the Art
256	8	2	32 bytes	256 bits AES
2048	8	2	256+ bytes	2048 bits ECSMID

Post-Quantum Computing:

Encryption (bits)	Size of Dword (bits)	Stable standard signal state (unitless)	Block size (bytes)	QC Resistance (bits)
256	8	4	32 bytes	128 bits AES
2048	8	4	256+ bytes	1024+ bits ECSMID

The table is a potent and simple approach to presenting a quantum-immune or resistance cryptography. This simplifies the complexity to the understanding the work of cryptography done with primitives of lattice basis. It is clear by now that quantum computing will be the death of AES and many other crypto systems. The nature of quaternary number manipulation makes this possible. Do check out the C++ operation of this algorithm as well as the android application:

<https://youtu.be/sx0YBK4RYcw>

<https://www.youtube.com/watch?v=feWVdhwkYJk>

Sample #1 CIPHERTEXT:

SÈTíâmNjÁDNm»DE¹cÝ»#ECÐNdÁÁj d
Nij#Ýâm#IN¹cNÁ¿Á»Ðc:ÁjNÁ¿Á:N¹Ccd
ÁNâCçNÈ:ÁÁ»d¹#:ÁNâ¹<Nc:N¹CádNEc
d¹jNcNijNÝcâ:ÝN¹cNÁhEÍ#â:NEÈTíâmNj
ÁDNm»DE¹cÝ»#ECÐ<NSÈTíâmNLÁDN
Q»DE¹cÝ»#ECÐNádNT#dÁÁNc:N#dDij
ijÁ¹»âmNm»DE¹cÝ»#ECÐjNdçNÆâ»d¹
NíÁ¹NÈdN¹#jN#TcÈ¹NdDijjÁ¹»âmNm»
DE¹cÝ»#ECÐ<gíáJp!

Sample #2 CIPHERTEXT:

)AĀñiāø{âèøāβè_ǵOëßá_ÈèØíāāyīØyá
ēlāánøǵOøáíáβèO>âDøáíá>øǵÈOíāø
ðÈOøA>Pāβīǵá>Pøīǵ@øÈ>øǵÈīīø_Ø
īǵDøÉøyøèOí>èøǵOøáí_ nái>ø_ AĀñi
āø{âèøāβè_ǵOëßá_Èè@Ø)AĀñiāøCā
èØ«βè_ǵOëßá_ÈèØīīøĀáíāPøO>øáíè
yyāǵβīāøāβè_ǵOëßá_ÈèDøīOøĀīβīǵø
nāǵøAīøǵán{øáĀOAgøīèyyāǵβīāøāβè
_ǵOëßá_Èè@ēīīǵG!

MESSAGE TEXT:

" Advanced Encryption Standard (AES) is a symmetric encryption algorithm... Following is an online tool to generate AES encrypted password and decrypt AES encrypted password. It provides two mode of encryption and decryption ECB and CBC mode."

We mentioned ASCII wide character for C++. However, Unicode representation were explored with java for those unfamiliar with C++. You can run the ciphertext output on 'cryptool' to see how it defies today's analysis of cryptography.

At this point, I am able, to show that each instance of message encryption produces distinct ciphertexts. There could be a contextual similarity yet the ciphertext of the smallest character in the message will be different at every iteration. This is against the prediction of

cryptographic primitives. However, it is a strength we need to tap into.

CONCLUSION:

One might not fully understand all the possibilities in the proposition of the algorithm. It is imperative that interests remain piqued to the possibilities pristine in an area requiring courage and anomalous thought process. It is clearer that a removal of the garbage collection phase in reduction of SAT to 3DM is relevant as well as the removal of certain traditions of computer science tantamount to the growth of cryptography. These loop back to the face of a lattice structure. Where the basis collection follows a certain set of strict rules. Practice had shown the decadence of the paradigm of one plain text and one cipher text: Where a key leads a plain text to a cipher text. The information provided shows clearly a fitting premise indicating: Intermediate representation. More so, that falsification of any responses whether it be verification or intermediate response fostering secrecy of the hidden message could be impossible in a bounded abstraction presented.

Mathematical functions that satisfies one reduction bias for NP complete problems, can no longer lead cryptography in the age of quantum computing. These problems are no longer considered hard problems. Moving forward, there is a need for harder problems within the set of NP problems. We surmise that giving the infinite samples of lattice or matrix vectors: They are indeed more than capable when dealing with the challenges posed by quantum

computing. The regularity of the points in Euclidean space are endowed with chaotic arrangements within the lattice basis. Especially, when the individual basis is reduced to cryptographical secure numbers this is owed to their expansive nature.

The LFKI generation of seeds and keys for encryption are much more efficient in entropy, fast, backward compatible on hardware/software. They are transparent, visible and fittingly complex. We have built several applications with this to note the interesting flow of this security architecture. Many other implementations of this skeleton abound. This has a great potential for possible commercial uses. Let us know what you think and what you will do with this as well as what you will like us to modify together. We will continue the research work.

REFERENCE

- [1] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 1996.
- [2] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," 2002.
- [3] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 2001.
- [4] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys.*

- Rev. Lett.*, 1997.
- [5] T. Laarhoven, "Solving Hard Lattice Problems and the Security of Lattice-Based Cryptosystems.," *IACR Cryptol. ePrint ...*, 2012.
 - [6] N. Johansson and J.-Å. Larsson, "Quantum Simulation Logic, Oracles, and the Quantum Advantage," *Entropy*, 2019.
 - [7] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)," *Comput. Intractability*, 1979.
 - [8] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, 1982.
 - [9] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 1971.
 - [10] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell Syst. Tech. J.*, 1949.
 - [11] D. R. (1994) Simon, "On the Power of Quantum Computation 1 Introduction 2 Quantum Probability and Computation," pp. 1–12.
 - [12] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2009.