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

Some of the papers on lattice basis owe respect to randomization reduction or deterministic reduction. 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 the 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 abounds 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 basis distribution. In this paper, it is shown that each vector representation has the potential to generate cryptographically secure number of keys. This follows 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=>16 and r=>16). A fitting rule is already available with the necessary mechanism to produce 1: n relationship for a plaintext against many ciphertext. It is that found in Open Knight Tour (OKT or KT) movements. This can easily be modified to absorb larger lattice basis. They 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. The sole intent is to solve the common problem of our time from practitioner' perspective. It is agreed on all ground 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 x n) formation. In absence of any back track; It does enumerate all points in Euclidean space if and only if n=>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 x 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 lattices. 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, mitigates 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), φ(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 & φ (N)}.*** *Choose d such that,*

{ *e*d mod φ (N) = 1}:

SOLUTION TO THE PROBLEM:

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 call for paper in cryptography. 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. Whether, it could be proved or not is only a matter of time and tools. This means that our crown jewel cannot depend on any mathematical function based on Fermat's theorem to be safe guarded eternally. 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.

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 $\mathbf{B=\{b_1,...,b_d\}}$ we denote the lattice it generates by $\boldsymbol{L(B) = \{\sum_{i=0}^{d}; xibi | xi \in Z\}}$. Its rank is d, and the lattice is said to be of full rank if d=n. We identify the basis $\mathbf{\{b_1,...,b_d\}}$ with the $\mathbf{\underline{n \times d}}$ matrix containing $\mathbf{\underline{b}_1,...,\underline{b}_d}$ as columns, which enables us to write the shorter $\mathbf{L(B) = \{B_x | x \in Z^d\}}$. We use both the terms lattice point and lattice vector for elements of a lattice.

A bit more time will be spent to introduce a new thought or insight for understanding 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 of the base is not enough to establish this as a case of NP-Hard problem [2]. That alone, could have been insufficient, or not good enough for quantum resistant encryption. If a lattice basis must be retained; 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 decision making. 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.

Asymmetric:

Encrypted data (c) = msg^{e} mod N

Decrypted data (msg) = $\mathbf{c^{d}}$ mod N

AES will suffer a similarly if not the same fate as RSA. The future of quantum computer will certainly vilify it. We don't really have to wait into the future anyway. People are already saving petabytes of data in cloud. These will be disclosed as soon as quantum computer becomes available.

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 (2^{8}) 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 a key derivation mechanism reduced to Pseudo Random Number Generators (PRNG). 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 a particular output* of 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 2^{64} iteration. At least, one can unravel useful information that will lead to breaking of such scheme using quantum computer as level playing field [3]. 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.

Encrypted data (c) = (msg, D): (msg xor D) mod Ct

Decrypted data (msg) = (c, D): (c xor D) mod Ct

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 [4]. The time to perform the feat is usually said to be in polynomial time. In that case the RSA math 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 classical cryptography useless. Another problem arises with the periodicity of lattice constructs. The begs the question. Is there a way to infuse the lattice with enough noise 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?

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 translation, transposition, transformation and substitution serves us well by forming an algorithm that covers lattice face key infrastructure and architecture. The face of lattice is 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,

Total flux = $\iint$ **f.n. dS** [where n=1].

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=>C=>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 $\boldsymbol{H_x}$ of a value $\boldsymbol{x}$ that occurs with probability $\mathbf{Pr}[\boldsymbol{x}]$ is

$$\boldsymbol{Hx} = -\mathbf{log2}(\,\mathbf{Pr}\,[\boldsymbol{x}]).$$

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

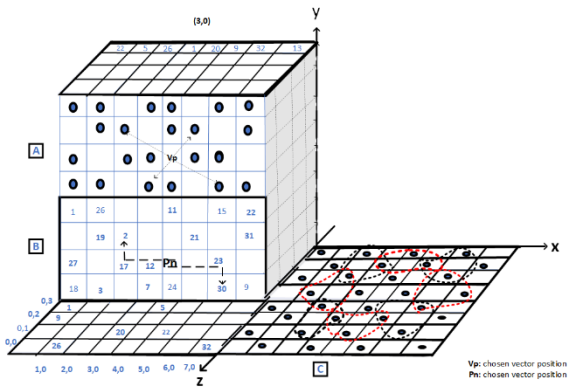
$$\begin{aligned}\boldsymbol{H}(\boldsymbol{X}) &= \boldsymbol{E}[\boldsymbol{Hx}] \\ &= \sum_{\boldsymbol{x}} \mathbf{Pr}[\boldsymbol{x}]\boldsymbol{Hx} \\ &= \sum\nolimits_{\boldsymbol{x}} -\mathbf{Pr}[\boldsymbol{x}]\mathbf{log}\,\mathbf{log}_{\mathbf{2}}(\mathbf{Pr}[\boldsymbol{x}]).\end{aligned}$$

It is submitted in quality from observation: That it is not a common knowledge to think of **n**=**∞** with respect to the equation of Galois field **GF(2^{p})** which essentially claims its validity from Euclidean space. In programmatical (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 **GF (P^{n})** where **n**=**∞**. 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 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 x 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 deducted 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. 1.0

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=>16 and r=>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=>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.

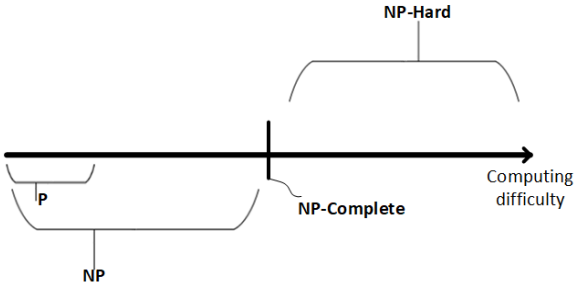


Fig. 2.0 Computational difficulty

P ≠ 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 ЄNP**
   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 ЄP the X ЄP
   ii. If Y ЄNP then X ЄNP

X not in P unless P = NP

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

X is NP-Hard if every problem Y ЄNP 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 (**Ș)** 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 ⊆ x *y*z.

Is there a subset S ⊆ T such that every element, **Єx ∪ y ∪ z** is in exactly one, **s Є 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 3.0**. 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:

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

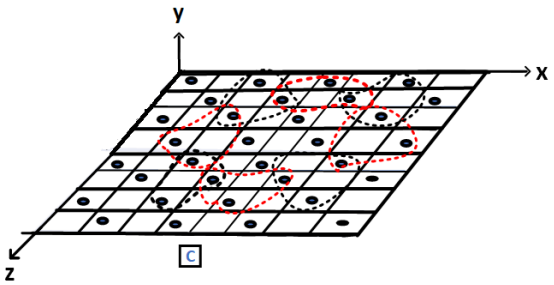


Fig. 3.0 3DM on OKT

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 chase 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 x 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=<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 the 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].

Relying on the above claims and premises 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_{nth}. This security could be applied in telecommunications, CPS, IoT, 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!

distinguished arrangement = 10! / 3! * 2! = **1,209,600**

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

7888906754

…

8889067547

8890675478

8906754788

,...

9067547888

,...n^{th}

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**

Iff 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

…n^{th}

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 (Pn). 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 $P_{n=1}$ … + $P_{n=2}$ … +… $P_{n=5}$ 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 => M_1
encrypt=>[ciphertext$_1$]^[P spkt$_n$][P kt$_n$] = M_1

CT_1 + silent password = CT_2 => M_2
encrypt=>[ciphertext$_2$]^[P spktn][P kt$_n$] = M_2

CT_2 + silent password = CT_3 => M_3
encrypt=>[ciphertext$_3$]^[P spkt$_n$][P kt$_n$] = M_3

CT_3 + silent password = CT_4 => M_4
encrypt=>[ciphertext$_4$]^[P spkt$_n$][P kt$_n$] = M_4

CT_4 + silent password = CT_5 => M_5
encrypt=>[ciphertext$_5$]^[P spkt$_n$][P 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 (bijecture) 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 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.

Data + MPIN encrypt = CT_1 --> M_1 encrypt =>[ciphertext$_1$]^[P kt$_{n=1}$] ^[P spkt$_{n=1}$] ^ [P (MPIN) M$_{n=5}$] = M_1

M_1 + MPIN encrypt = CT_2 => M_2 encrypt =>[ciphertext$_2$]^[Pkt$_{n=2}$]^[P spkt$_{n=2}$]^ [P (MPIN) M$_{n=5r}$] = M_2

M_2 + MPIN encrypt = CT_3 => M_3 encrypt =>[ciphertext$_3$]^[Pkt $_{n=3}$]^[P spkt$_{n=3}$] ^ [P (MPIN) M$_{n=5}$] = M_3

M_3 + MPIN encrypt = CT_4 => M_4 encrypt =>[ciphertext$_4$]^[Pkt$_{n=4}$]^[P spkt$_{n=4}$] ^ [P (MPIN) M$_{n=5r}$] = M_4

M_4 + MPIN encrypt = CT_5 => M_5 encrypt =>[ciphertext$_5$]^[P kt$_{n=5}$] ^[P spkt$_{n=5}$]^ [P (MPIN) M$_{n=5r}$]^[RM$_3$ MPIN es] = M_5

Following the above process, the mpin encrypt shown is that of the recipients. If one is sending a message requiring ZKP. For example, M_3mpin of position (MPINkt$_{n=3}$) is tripped and sent with the message:

Data + mpin encrypt = CT_5 --> M_5 encrypt -->[ciphertext$_1$]^[M_3pin encrypt]^[Pkt$_n$]^[P mpinkt$_n$] -- > M_5

Note the removal of M_3mpin key positions. Data + mpin encrypt = CT_5 --> M_5 encrypt -->[ciphertext$_1$]^[M_3mpin encrypt]^[Pkt$_n$] -- > M_5

On your device you have M_2mpin encrypt: [M_2mpin encrypt]^[P mpinkt$_{n=2}$] Note the replacement of the unstripped M_2mpin with M_3mpin keys' position [M_2mpin encrypt]^[P mpinkt $_{n=3}$]

in this order the attacker may never be able to go back to M_1 if at all they gain access to the network. M_3mpin could be used as a digital signature of each user in the network.

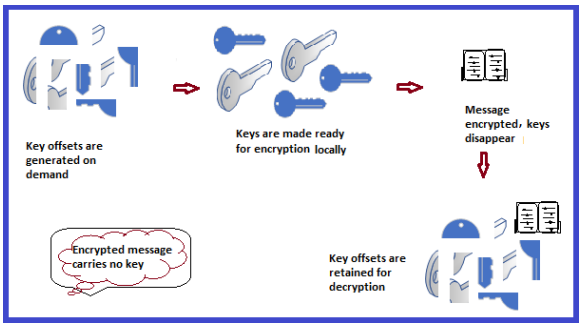
Simply put

1. The M_3PIN 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 (Pn) 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*****



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)

ADVANTAGES (NEW APPROACH or AXIOMS):

1. GF 2^p where p =< 8; -> GF 2^p, where p !<8 & p > 8 | ∞ (or goes to infinity).
2. Non-Deterministic reduction insinuating 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 originates 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 vertices. 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 generation. 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. Addroundkeys

   c. shiftrow

   d. mixcolumns

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

   Irreducible polynomial is no longer a question of symmetric cryptography. No key schedule or register. You can equally 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: They have the largest number of vectors vis-a-vis largest flux.

5. Cipher keys (P) are no longer saved as they are generated from any position on the matrix (lattice face) upon request by NP. 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=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 you can get started from any indexed point or vector. The origin 0 to any other part produces a different entropy flux. While the order of the positions are

regular (deterministic), they generate chaotic set of numbers. This generates a new set of 680digit long numbers. This knowledge reveals the changing nature of the message' ciphertext as well. When similar contents are encrypted the ciphertext are usually different. Thus, hashing could only be needed for CRC or message integrity check. P! = NP || 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 is Homomorphic encryption mechanism [11]. The homomorphic encryption (HE) properties makes possible the flexibility of the algorithm (M_1-M_5) as public key encryption. This encrypts from this wrapping technique could be used for ZKP.
8. The complexity is O (n = ***message.length***)
9. KEM, Digital signature and seeming public key encryption is built within the algorithm from the scratch. The mixes of attributes e.g MPIN, eFRI, Address and Password can bring about a God mode permission for IAM operations in all kinds of environments with respect to business logic reflected.
10. Plain text to cipher text 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.

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:

SÈTÍåmNĵÁÐNm»ÐE¹ċÝ»#EĆÐNdÁÁijdNij#Ýåm#ÍN¹ċNÁ¿Á»Ðċ:ÁĮNÁ¿Á:N¹ĆċdÁNâĆċNÈ:ÄÁ»d¹#:ÄNå¹<Nć:N¹ĆådNEċd¹ĮNćNijNÝċå:ÝN¹ċNÁħEÍ#å:NEÈTÍåmNĵÁÐNm»ÐE¹ċÝ»#EĆÐ<NSÈTÍåmNĽÁÐNQ»ÐE¹ċÝ»#EĆÐNådNT#dÁÄNċ:N#dÐijijÁ¹»åmNm»ÐE¹ċÝ»#EĆÐĮNdċNÆå»d¹NÍÁ¹NÈdN¹#ÍĵN#TċÈ¹NdÐijijÁ¹»åmNm»ÐE¹ċÝ»#EĆÐ<ğıåĴþĮ

Sample #2 CIPHERTEXT:

)ĄĂnÏąØ{åèØąßè¯ġOëßá¯ÈèØĨååyĨØyáëÏąánØġOØåíåßèO>åDØåíå>ØġÈOĨåØðÈOØĄ>ÞåßĨġá>ÞØĨġ@ØÉ>ØġÈÏĨØ¯OĨġDØÉØyØëOÏ>ëØġOØåÍ¯náÏ>Ø¯ĄĂnÏąØ{åèØąßè¯ġOëßá¯Èè@Ø)ĄĂnÏąØCåèØ«ßè¯ġOëßá¯ÈèØÏĨØĂáĨåÞØO>ØáĨèyyåġßÏąØąßè¯ġOëßá¯ÈèDØĨOØĀÏßĨġØnåġØĄĨØġán{ØáĂOĄġØĨèyyåġßÏąØąßè¯ġOëßá¯Èè@ēĮĬĝĜĮ

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 no longer necessary owing to the face of a lattice structure. Where the basis collection follows a certain strict rule. Practice had shown the decadence of the paradigm of one plain text and one cipher text: Where plain text leads through a key to cipher text. The information provided shows clearly a fitting premise indicating: Intermediate representation. More so, that falsifying responses whether it is verification or intermediate response fostering secrecy of the hidden message could be impossible in a bounded abstraction.

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 generation of seeds or 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] 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.

[3] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, 1997.

[4] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," 2002.

[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] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2009.