Lifeboats on the Titanic Cryptography

NIST-PQC is Titanic Style Hi-Tech, Pattern Devoid Ciphers are the Titanic Lifeboats

Gideon Samid
Electrical, Computer and System Engineering
Computer and Data Sciences
Case Western Reserve University, Cleveland, OH
Gideon.Samid@CASE.edu

Abstract: The Titanic was the ship that "could not sink," fortunately its designers installed lifeboats (not enough) despite having no logical grounding for this waste of space and material. It was out of respect for unforeseen surprises. NIST-Post Quantum Ciphers represent the best and the brightest in world crypto intelligence. They are certified as good for their purpose. And likely so, alas, not surely so. If we could find a crypto equivalent for the Titanic Lifeboats, should not we load them up for our journey? Indeed, pattern-devoid cryptography is the crypto equivalent of the lifeboats that mitigated the Titanic disaster. Pattern-Devoid cryptography (PDC) may be deemed inelegant, inconvenient, and bloated, but it will hold up against quantum computers more powerful than expected, and more importantly, it will hold up against adversarial mathematical talent greater than expected. Which is why we should put up with its negatives, and install it just in case the Titanic story repeats itself in cyberspace. This article elaborates on this proposition.

1.0 Introduction

Cryptography has evolved as a battle of wits between the code builders and the code breakers. The trophy shifted from time to time, occasionally with dramatic consequences. Most notably is Alan Turing's cracking of the Nazi Enigma. It was profoundly consequential because the Enigma designers were so confident in its cryptanalytical "unsinkability" that they have not considered changing, or upgrading. And the Allies on their side, sacrificed blood and treasure to alley any Nazi suspicion that the cipher was compromised.

This battle of wits continues today. The prevailing ciphers that carry and support life on cyberspace appear vulnerable to quantum computers, and hence the US National Institute for Science and Technology, NIST, is leading new wave of smarter ciphers, (Post Quantum Ciphers, PQC) to beat the prospective quantum attack. The battle rages between the NIST designers and the NIST adversaries bent on cracking the new wave. As of this writing NIST forwarded five PQ ciphers, which by their number, serve as an admission that NIST is worried that neither one of them is serving their declared purpose. Therefore NIST is busy searching for more algorithms that will be better than the current ones. Do the NIST adversaries have an Alan Turing class cryptanalyst that will lead them to victory? This is clearly an open question. What is not an open question is the level of damage to life on cyberspace that will be happening if NIST-PQC is compromised. Utilities, power, payment, healthcare, transportation, government, military -- all systems grinding to a halt.

"*Well, argued*" claimed one colleague: "*We just ensure that we are smarter than our adversaries. What else is there?*" he asked.

I then familiarized him with the principles of innovation science [85], and in particular with the idea of systematically challenging established precepts. For most of its history cryptography has been a battle of wits. Can this premise be challenged? He doubted, so I described a game where in turn one player throws two dice and the other tries to guess the outcome (2-12). Clearly the smarter mathematician will always guess 7 and win against a random chooser. Now let's change the game by playing with one die only. What happens -- the mathematician has lost its edge because randomness rules supreme. Can we then develop an approach to cryptography where randomness will be so pervasive that like with the dice case, math advantage and computational advantage will he voided? Cryptography then will become a battle of randomness projected and randomness scanned -- wits and mathematical talent will become irrelevant.

Examining the role of randomness in the current crop of ciphers we find it is limited to the choice of a key within a well-defined key space. How can we give randomness a bigger role?

We take aim at the key. Today the key space is not a secret, the key size is known, and the *entire key* is used to generate the ciphertext from the plaintext. These fixed parameters may all be randomized. the key space may be randomized, the key size may be independently randomized, and each time a plaintext is converted to ciphertext, a randomized portion of the key may be used for the process.

Ciphers that pack that much more randomness may dispose of the mathematical crypto barrier as much as the one die game voided the mathematical wit contest. In other words, we are striving for pattern devoid ciphers that project so much randomness around the protected secrets that mathematical complexity is no longer required. Randomness admits no shortcuts, thereby forcing the cryptanalyst into brute force attack.

Randomness may also be introduced at the ciphertext level. Today the ciphertext is all 'hot' -- every bit in it is critical and must be analyzed and evaluated by the cryptanalyst in order to extract the plaintext. What if we could mix such proper ciphertext bits with noise -- random bits that bear no content. We may call it decoy ciphertext. We will need, of course, to devise a way for the recipient to discriminate between noise and content-bearing bits, while keeping the cryptanalyst in the dark as to which bits are content-bearing and which are noise. This will at least slow down the attacker.

Such a Decoy Tolerant Cipher, DTC, gives the message transmitter a new power that was not available before: the power to project security responsive to the perceived threat. Less critical messages will be protected with small quantities of noise, and critical secrets will be protected with large quantities of noise. The transmitter presumably knows which message is critical and which is not -- so the transmitter is the proper determinator of the level of decoy protection.

Such *randomness leveraging ciphers* , RLC, are messy by comparison. Factors that are fixed in the prevailing ciphers become randomized in the RLC. The actual implementation procedures will be more complex; implementation in small storage situation, or in small communication capacity environment will be a much bigger challenge for RLC deployment, so wherever safe, let's use the prevailing ciphers, but since the prevailing ciphers may face the cyber equivalent of the Titanic disaster, we better develop some effective randomness leveraging ciphers, to serve, at the very least as the cyber equivalent of the Titanic lifeboats.

2.0 Randomness v. Pattern, a Security Question

Pattern is order. Order can be expressed in different ways. Each way may lead to a different insight regarding that order. What is invisible one way, is visible in the other way. Writing numbers in the decimal systems makes it very easy to determine if a number is divisible by 5 or 10. It is less visible using octal, hexadecimal, binary, or the Roman numbering system. The Laplace transform offers insight for integration, not visible otherwise. Using Numerization [89] prime numbers appear to show a pattern not visible otherwise. Operation regarding an order in one form that may look hard, may look easy when the same order is written in a different language. No one claims that any given order cannot be written in forms not yet identified, and hence no one would claim that an operation that is difficult to carry out when the order is expressed in one way, is necessarily difficult when expressed in a different way. Hence all the ciphers we use today which are based on a pattern, an order, are subject to being compromised through writing them differently. In fact many early ciphers are being decrypted simply by changing the order of the ciphertext. And for that reason, all pattern-loaded ciphers generate a battle of wits between cipher builders and cipher crackers. The question before us is: *can we project security without pattern. Is pattern devoid cryptography feasible?*

3.0 PDC Solution Concept

There are many prospective ways to achieve Pattern Devoid Cryptography (PDC). Here we present one particular concept: PDC - Solution Concept 1 (PDC-SC1), that has been manifesting itself in several actual ciphers in service today.

Consider a plaintext alphabet α comprising of *a* letters: A_1, A_2, ...A_a.

A message M written in α is comprised of m letters M_1, M_2, ... M_m where each $M_i = A_j$, for all i=1,2,..m and for some j for j=1,2,...a

We consider a pattern-devoid cipher, PDC, as one which will allow a transmitter to communicate M to a recipient with whom a secret key K was shared. The communication of M will be carried out letter by letter.

Each letter in α would be associated with its "pointer space". A pointer is a data element which may be represented as a bit string. π_i represents the pointer space for letter A_i, it comprises p_i pointers. The p_i pointers are marked with natural numbers 1,2,...p_i. Pointer $P_{ij} \in \pi_i$, and is pointer marked as count j.

Each letter in α would also be associated with an "evaluator space". An evaluator is a data element which may be represented as a bit string. E_i is the evaluator space for letter A_i, it comprises of e_i evaluators. The e_i evaluators are marked with natural numbers 1,2,...e_i. Evaluator $E_{ij} \in E_i$, and is evaluator marked as count j.

Consider a binary evaluation function, β, which takes in a pointer and an evaluator, and computes a binary result 0 -- unfit, 1 - fit.

$$\{o,1\} = \beta(P_{ij}, E_{ik})$$

To transmit letter A_i to the recipient, the transmitter would use a pointer P_{ij}, for which there will be at least one evaluator E_{ik} such that:

$$\beta(P_{ij}, E_{ik}) = 1$$

while:

$$\beta(P_{lu}, E_{lv}) = 0$$

For all $l \neq i$, for all $u = 1,2,...p_l$, and for all $v = 1,2,...e_l$

This is regarded as the Letter Transmission Terms (LTT)

The recipient evaluating P_{ij} with β over α will find out that only letter A_i is marked with β=1, all other letters in α show β=0. Thereby the recipient will realize that the pointer sent to them P_{ij} is pointing to A_i.

Message M will be transmitted letter after letter as described above.

In a full scale PDC the pointer space for each letter will be an infinite set of pointers. Also the evaluators spaces are with no set limit. However, their identity is part of the shared secret key, K, so it must practically be limited, nonetheless its size is a secret.

Decoy:

When the recipient receives a pointer P that does not meet the letter transmission terms, LTT, they regard it as a decoy pointer, and it is disregarded.

A pointer will fail to meet the LTT either by failing the fit (β=1) for all the letters in α (the zero case), or by failing the unfit test (β=0) for less than (n-1) letters (the confusion case).

A decoy is a security enhancer since the attacker cannot distinguish between it and a bona find pointer, so it disrupts the cryptanalysis effort. The more decoys are sent before, after, and during the sending of the m letters of M, the greater the security over the secrecy of message M.

The *zero-case decoy* is easy to achieve. One simply selects a pointer P that does not belong to any of the pointer spaces. The *confusion case* decoy may be generated by chance when selecting pointer P_{ij} to point to letter A_i, it so happens that β evaluates to 1 for some other letter A_k, $k \neq i$:

$$1 = \beta(P_{ij}, E_{kl})$$

The recipient of pointer P_{ij} will realize that the pointer points to letter A_i and to letter A_k, and therefore it is a decoy.

Leveraged Randomness

The PDC is designed to apply randomness at any decision point, in order to wash off any pattern which may be a hook for a cryptanalyst. It is like building a smooth vertical wall to prevent a climber from climbing.

Letter Transmission Procedure (LTP): When transmitting letter A_i, the transmitter does:

1. randomly selects an evaluator E_{ij} from the available e_{ij} evaluators.
2. randomly selects a pointer P_{ik} from the P_i pointer space.
3. evaluates compliance with the letter transmission terms.

If the evaluation in (3) is positive (compliance) then the transmitter transmits P_{ij}.

If the evaluation in (3) is negative (non compliance) then the transmitter randomly chooses between two options:

(i) restart the letter transmission procedure,

(ii) initiate a confusion resolution procedure CRP.

Confusion Resolution Procedure: In the case where a pointer P_{ij} which points to letter A_i, also points to letter A_k ($k \neq i$), a state of confusion is identified. The transmitter then sends this pointer to the recipient and then restarts the LTP, sending the same letter A_i again. In the second round for sending off a pointer for A_i, compliance may be achieved, and in that case this second pointer will be transmitted. Confusion resolved. The recipient will disregard the first pointer for A_i as decoy (because it failed the LTT test), and interpret the second pointer as pointing to A_i.

If the second pointer that was selected to transmit A_i, also results in confusion, namely it points to letter A_l, ($k \neq l$), and perhaps to more letters, then the transmitter transmits this second pointer. The recipient finds out that the first pointer pointed to letters i,k, and the second pointer pointed to letters i,l. Letter i is the common letter between the two pointers. This will tell the recipient that the combined two pointers point to letter i. In case k = l, a third pointer is processed, pointing to Ai, sooner or later letter Ai will remain as the only constant among all the pointers, and it will be identified as the target of this set of pointers.

Example: Transmitter wishes to send letter 3. The chosen pointer-1 points to letters (3,7,1). Confusion. The transmitter sends pointer-1 to the receiver, and then the transmitter randomly picks pointer-2 to point to letter 3, but, alas, this pointer points to other letters too (2,7,3,9). Confusion still reigns because both letters 3 and 7 appear in the list of both pointers. The transmitter sends pointer-2 to the recipient, and pick a third pointer to point to letter 3. This pointer points to the following letters (1,2,3,8,9). Examining the three "hit lists" from the three pointers one realizes that letter 3 is the only letter that appears in all three pointed-to "hit lists", and hence by communicating to the recipient all three pointers the recipient will evaluate as above and conclude that these three pointers communicate letter 3. The cryptanalyst will have no knowledge of activating a confusion resolution procedure and to what degree.

In general let the set S_l includes all the letters of α for which the β function showed a fit (β=1) while the transmitter activated the RLP to select a pointer to point to A_i. When the transmitter reactivates RLP and selects another randomized pointer to A_i, this pointer is associated with set S_2 which includes all the letters of α which are pointed to by the second pointer. And similarly set S_k will include all the letters pointed to by the pointer selected in the k round for pointing to A_i.

As the value of k increases (more and more rounds of RLP), it eventually reaches a point k = *l* where the only letter pointed to by all the *l* pointers is A_i. At that point the CRP terminates. The recipient will evaluate the *l* sets and also determine that A_i is the only persistent letter in all the *l* sets, and hence this is the letter which the transmitter intended to transmit.

The PD cipher needs to be designed so that the CRP is handled in a reasonable time, without allowing the confusion to linger and prohibitively slow down the transmission.

Any arbitrary message M written in alphabet α will be so transmitted letter by letter.

3.1 Off Randomness Procedures (PDC-SC1)

As described the PDC-SC1 calls for random choices. However the transmitter may opt for off-random choices to project more security. We consider the following:

1. Non repeat

2, Seclusion

3. Enhanced decoy

"Non Repeat" is a modification applicable to a large enough pointer space. Every pointer used once is marked for no further use.

"Seclusion" is a modification where subsets of the evaluators and subsets of the pointers are secluded from the RLP and kept as a fresh reserve for extra sensitive transmissions, or in case of an increased threat.

"Enhanced decoy" is modification where the transmitter actively searchers for pointers that point to no letter in α. The more decoys mixed with the contents bearing transmission the greater the cryptanalytic barrier.

3.2 Security

The key, K of the PDC cipher is the set of the various evaluators for all the letters in α. β is assumed public. K is selected randomly.

In the common ciphers the entire key is used for every instant of encryption. Once an attacker discovers that a given key, K, fits a known set of transmissions, they know the same key is used going forward. With PDC a randomized part of the key K is used for every instant of encryption. The next instant may use a previously unused part of the randomized key such that any knowledge gained from previous transmissions will have no bearing on the identity of the newly used evaluator. What is more, every evaluator may be used with one item in an infinite or very large space of pointers. Some of these pointers may point to different letters. This mounted randomness allows no room of analytics. So while the traditional method where the entire key is used every time, is vulnerable to analytics, the PDC leaves no opening for pattern revealing cryptanalysis, simply because there is no pattern to be revealed. Every step is randomness controlled.

Having discounted mathematical breach we are left with (i) progressive cryptanalysis, (ii) exhaustive cryptanalysis, and (iii) circumstantial cryptanalysis. We will see how the transmitter may defend against all attacks.

The randomness used to project security with PDC is of two types (i) bilateral -- shared, and (ii) unilateral -- unshared. The first one is the key, and the second one is the decoy and the confusion resolution procedure. The transmitter will use unilateral randomness to project security commensurate with the threat.

3.2.1 Progressive Cryptanalysis

A cryptanalyst is catching the series of pointers P_1, P_2, ... P_i sent by the transmitter. Let's give the cryptanalyst the advantage of chosen plaintext. It is not possible to have a chosen ciphertext because the PDC operates on the basis of built-in randomization.

The cryptanalyst then will assume that pointer P_1 is designed to send letter A_i. This assumption may be wrong because the transmitter could have sent a zero case decoy. And even if right then it may be that the same pointer points also to letter A_j, and if used a second time will point to A_j not to A_i.

Assuming no confusion and no zero decoy, the cryptanalyst will be able to define an evaluator space for A_i, these are all the evaluators which yield $\beta=1$ with pointer P_1. In a proper PDC β is designed such that every pointer will be associated with an infinity of evaluators and every evaluator will be associated with an infinity of pointers. In finite PDC these infinities will be replaced by large finite sets. If in the chosen plaintext setting the letter A_i is sent out again and again, each time a different pointer is used. Each of these pointers could be a zero decoy or CRP. step. But assume neither of the above and assume that repeat pointers all point to A_i, because the evaluator's space is infinite or at least very large, depends on the cipher, then nothing more than defining this space can be done by the cryptanalyst.

The cryptanalyst drowns in the equivocation defense of the PDC; no progressive analysis can yield a breach which is defined as discovering the key -- the identities of the evaluators.

3.2.2 Exhaustive Cryptanalysis

This brute force approach will call for the cryptanalyst to randomly assign evaluators to the letters in α, and interpret the stream of pointers (the ciphertext) accordingly. If the interpretation is short of a plausible message then, a new set of evaluators is randomly selected and the stream of pointers is evaluated with it. This test goes on until a plausible message is extracted.

The way the PD cipher is built controls the amount of work needed in order to spot a plausible message. The combination of bilateral randomness and unilateral randomness will give the users the power to make this cryptanalysis strategy infeasible.

What is more, even if a plausible message is spotted, there is always a suspicion that there are more plausible messages than can be tailored to this ciphertext, and if so, then there is no way to discriminate between them, leading to terminal equivocation.

3.2.3 Circumstantial Cryptanalysis

Let M_1, M_2, ...M_n be the n most plausible messages that could have been encrypted to a captured PDC ciphertext (stream of pointers). These n messages are deduced from the particular circumstances of the case in point. Each message i of those messages comes with an a-priori likelihood. Pr_i, and these likelihoods define an a-priori entropy H_0. Complete cryptanalysis of the ciphertext will reduce the entropy to zero. $H_c = 0$.

The effort to stop cryptanalysis may be defined as the effort to prevent the entropy from slipping. If these n messages may all be interpreted from the given stream of pointers, then the post entropy equals the a priori entropy $H_c = H_0$, and no cryptanalysis was accomplished.

There are two ways to defend against this cryptanalysis: (i) blind randomness, and (ii) composite ciphertext.

In blind randomness the parties select more evaluators to build the key from, and select evaluators that point to larger and larger pointer space. The transmitter will incorporate in the ciphertext more and more decoys to increase the probability for a false message to be consistent with the transmitted ciphertext. This will pile up the amount of equivocation that needs to be resolved.

Composite Cipher: In composite cipher methodology the transmitter identifies the same n most plausible messages M_1, M_2, ... M_n; one of them, say M_i, presumably is the right one. The transmitter then is building n keys K_1, K_2, ... K_n for all these n messages. For i= 1, 2, ...n. The transmitter is then encrypting the n plausible messages each with its respective keys, creating n ciphertexts (n streams of pointers) C_1, C_2, ... C_n.

C_1 is the ciphertext that is interpreted back to the right message M_1.

The transmitter is then putting together a composite cipher, CC, in which all the above mentioned n ciphertexts are mixed together. The mixing should be letter-wise namely the order of the letters. for a given ciphertext will not be disturbed, but between any two successive letters (pointers to these letters, actually), any number of pointers from other ciphertexts may be injected.

The transmitter then will enact a broader version of the randomness leveraging procedure. The letter transmission terms, LTT, will be adjusted to include evaluators from keys K_2, K_3, ...K_n. all of which will need to show $\beta = 0$. Let P be any randomly selected pointer designed to point to letter A_i interpreted through K_1. If P points to another letter in α per K_1, then it violates the letter transmission terms and the confusion resolution procedure is invoked. However in the broader version of the LTT, if P when applied to any evaluator of any of the other (n-1) keys is evaluating as a fit (β=1), then P violates the broader version of the LTT. In that case the next step will be to reinvoke the source of randomness and to select another pointer that will point to A_i per K_1, and test again the broader LTT. And so repeatedly until a pointer is found that satisfies the broad letter transmission terms. By having the pointer space large enough one reduces the expected number of rounds needed until compliance with the broad LTT is achieved.

Once achieved we have built a composite cipher which when interpreted by K_1, evaluates to M_1. The recipient using K_1 will read M_1 which is the transmitted plaintext. Alas, all other n-1 plausible messages will be associated with a keys K_i for i=2,3,,,,n that will interpret the composite cipher message as M_i. Say then that an omnipotent cryptanalyst, will reach the conclusion that the captured composite ciphertext may represent any of the n plausible messages that fit the prevailing circumstances.

This conclusion was held by the cryptanalyst before holding the content of the ciphertext (the composite ciphertext), so having knowledge of the content of the composite ciphertext CC has not added any knowledge to the cryptanalyst and has not reduced the entropy of the situation.

We have described here a method to ensure that a ciphertext generated from the PD cipher may be made mathematically secure,

3.3 Specified Ciphers

Pattern Devoid Cryptography has been published as a chapter in a book on cyber security. https://www.intechopen.com/online-first/pattern-devoid-cryptography. A more recent PDC "Polar Lattice Cryptography" is published here: https://eprint.iacr.org/2025/452 [88]

3.3.1 Polar Lattice Cryptography [88]

The Polar Lattice cryptography (PLC) is based on a secret polar structure where each letter of α is marked with a starting point on the lattice and an end point on the lattice. A point being an intersection of a ray with a circle of the polar lattice. These two points define the evaluator. A pointer is a trail that start with the starting point and ends at the end point. This trail is the pointer. Since one can mark a trail starting at the starting point and that keeps going and going until it hits the end point, it is therefore possible to chart infinite number of such trails -- marking an infinitely large pointer space linked to a single evaluator.

3.3.2 BitFlip [7,65,77]

A simple but powerful PDC is based on Hamming distance between bit strings. Here is a brief introduction. Consider base64 as the plain language. It is comprised of 64 letters. In its simplest form we use a randomness source to choose 64 evaluator string, each comprising 32 bits. We use a β evaluation function as follows: β = 1 if the Hamming distance between a pointer string and a letter evaluator is zero. β=0 otherwise. It boils down to identity. the pointer string is identical to the evaluator string. This comes down to a simple fixed code cipher, which of course, it easy to crack based on a little bit of text. Alas, the evaluator space is of size 2^{32} = 4294967296 so there are plenty of decoy pointers to be mixed with the content-bearing pointers. That means the transmitter, if he gets nervous, can wrap the 'real; pointrs with a deluge of decoy pointers without pre coordinating with the recipient. The recipient will readily discriminate between the real and the decoy. The composite ciphertext may be so long that several plausible plaintext messages could be extracted from it, without any indication which is the right one.

To pack more randomness, we can change the definition of β to be a fit (β=1) if and only if the hamming distance between the evaluator string and the pointer string is h, where h may be randomly picked per alphabet or per letter. The larger the value of h (up to h=16) the greater the randomness input because there are many pointers that would point to a given letter.

Next we can pick a random number of evaluators per each of the 64 letters of the Base64 plaintext alphabet. And choose randomly an evaluator for transmitting the same letter.

We can further charge the cipher with randomness by defining β over two strings of unequal size. Two bit strings, A and B of size a bits and b bits respectively where $a \geq b$ may be processed as follows. Align A and B such that the first bit in A, a_1 is matched with the first bit of B, b_1, read the Hamming distance at that state, h_1. If a=b the procedure ends. Otherwise shift string B so that b_1 is matched with a_2, and read the Hamming distance, h_2. Keep shifting B under A until the last bit in B is matched with the last bit in A. This will result in b Hamming readings: h_1, h_2, ... h_b. Pick a condition over these b values. For example β(A,B) = 1 if $h_{max} = 0.5b + 1$, for b even and $h_{max} = 0.5(b+1)$ for b odd. Another option is for β = 1 if the gap between h_{max} and h_{min} is a preset number.

This extension of Hamming distance allows two strings of different size to be β-determined. Thereby the cipher will pack more randomness per the size of each evaluator string that may be different, and the size of the matched pointer which will also be different.

The cipher can be defined so that the stream of pointers flows in a continuum that does not allow the attacker to readily dissect it into individual pointers, alas the recipient will be able to properly interpret the ciphertext stream. In the Polar Lattice cipher a pointer string is identified when it is fully submitted, so the recipient knows that the next bit represents the next pointer. With BitFlip this can be done be expressing each bit with two bits thus 1 → 10, and 0 → 01, leaving 11 and 00 to serve as delimiters. A large pack of successive pointers can then be transposed in order to hide the pattern.

Along the right key (the series of 64 evaluator strings), the transmitter can build decoy key, K_d comprised of different set of 64 evaluators. The transmitter could then pack a decoy message M_d into the composite ciphertext such that M_d will be written through K_d. The recipient does not have to be aware of K_d and would not read M_d, only the right message M, but an omnipotent cryptanalyst will extract both M and M_d and will not be able to determine which is the right one. The same can happen with more than one decoy message, to cover all the plausible messages based on the circumstances.

The formal presentation of BitFlip is found in the reference, [7, 65, 77] the above is for purpose of illustration as to the ability of the PDC user to pour as much randomness, equivocation as desired, to achieve a level of security as high as desired, up to mathematical secrecy.

4.0 Lifeboat Implementation

NIST PQC being more elegant and better fitting into the larger cyber security operation, and also enjoying the trust of the cryptographic community, it makes sense to keep NIST-PQC installed and operational.

However, given that the Titanic Syndrome cannot reasonably be brushed away, one should install a *ready-to-activate* pattern devoid cipher. In the event that it becomes clear that NIST algorithms have been compromised, then the PDC should kick in and serve its purpose. In such a case the added discomfort associated with PDC will be of little concern.

The crew on the Titanic has been trained to operate the lifeboat, similar training is due for the cyber security team in critical cyber security centers.

Reference

1. “An extension of the Shannon theory approach to cryptography”. Martin Hellman, IEEE Transactions on Information Theory, V. 23, 3 1977, pp. 289 - 294
2. "A New Perspective of Geometry and Space as an Evolutionary Organizer of Data." Gideon Samid, http://www.dgsciences.com/Geometry_H7n18.pdf
3. "A Unary Cipher with Advantages over the Vernam Cipher" Gideon Samid, https://eprint.iacr.org/2020/389
4. "Anonymity Management: A Blue Print For Newfound Privacy" Gideon Samid, The Second International Workshop on Information Security Applications (WISA 2001), Seoul, Korea, September 13-14, 2001 (Best Paper Award).
5. "Artificial Intelligence Assisted Innovation" Gideon Samid, https://www.intechopen.com/online-first/artificial-intelligence-assisted-innovation
6. "At-Will Intractability Up to Plaintext Equivocation Achieved via a Cryptographic Key Made As Small, or As Large As Desired - Without Computational Penalty." Gideon Samid, 2002 International Workshop on CRYPTOLOGY AND NETWORK SECURITY, San Francisco, California, USA September 26 – 28, 2002.
7. "BitFlip: A Randomness Rich Cipher" 2017, Gideon Samid, Sergei Popov, https://eprint.iacr.org/2017/366.pdf
8. "BitMap Lattice: A Cyber Tool Comprised of Geometric Construction", US Patent 10,911,215, Feb 2, 2021
9. "Chaos-based Cryptography: A Brief Look Into An Alternate Approach to Data Security" A Sharif, NI Raihana, A Samsudin - Journal of Physics 2020 https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012110/meta
10. "Communication Theory of Secrecy Systems". Claude Shannon (1949) http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf
11. "Cryptography of Things: Cryptography Designed for Low Power, Low Maintenance Nodes in the Internet of Things" Gideon Samid https://search.proquest.com/openview/8897dc1c4858b327796917b8fcdff7ae/1?pq-origsite=gscholar&cbl=1976348
12. "Cyber Passport: Preventing Massive Identity Theft. " Gideon Samid, https://eprint.iacr.org/2016/474
13. "Denial Cryptography Based on Graph Theory." Gideon Samid (2004) US Patent 6,823,068.
14. "Drone Target Cryptography" Gideon Samid, https://eprint.iacr.org/2016/499
15. "Effective Concealment of Communication Pattern (BitGrey, Bitloop)" US Patent 10,673,822 June 2, 2020

16. "Encryption Sticks (Randomats)" Gideon Samid ICICS 2001 Third International Conference on Information and Communications Security Xian, China 13-16 November, 2001
17. "Encryption-On-Demand: Practical and Theoretical Considerations" Gideon Samid https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.215.2463&rep=rep1&type=pdf
18. "Equivoe-T: Transposition Equivocation Cryptography." Gideon Samid, International Association of Cryptology Research, ePrint Archive https://eprint.iacr.org/2015/510
19. "Essential Shannon Security with Keys Smaller than the Encrypted Message the Encrypted Message" Gideon Samid, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.1585&rep=rep1&type=pdf
20. "FAMILY KEY CRYPTOGRAPHY: Interchangeable Symmetric Keys; a Different Cryptographic Paradigm" Gideon Samid https://eprint.iacr.org/2021/458
21. "Feeding Cryptographic Protocols with Rich and Reliable Supply of Quantum-Grade Randomness" Gideon Samid, https://eprint.iacr.org/2020/968
22. "Feeding Cryptographic Protocols with Rich and Reliable Supply of Quantum-Grade Randomness" Gideon Samid, https://eprint.iacr.org/2020/968.pdf
23. "Fingerprinting Data" Gideon Samid, https://eprint.iacr.org/2018/503
24. "Hush Functions Extended to Any Size Input versus Any Size Output." Gideon Samid, https://eprint.iacr.org/2012/457.pdf
25. "Intractability Erosion: The Everpresent Threat for Secure Communication" Gideon Samid The 7th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2003), July 2003.
26. "Larger Keys, Less Complexity" Gideon Samid, A Strategic Proposition." https://eprint.iacr.org/2018/406.pdf
27. "Proposing a Master One-Way Function." Gideon Samid, https://eprint.iacr.org/2007/412
28. "Randomized Bilateral Trust (RABIT): Building Connectivity for Cyber Space" US Patent 10,798,065, Oct 6, 2020
29. "Randomness as Absence of Symmetry" Gideon Samid, THE 17TH INTERNATIONAL CONFERENCE ON INFORMATION & KNOWLEDGE ENGINEERING (IKE'18: JULY 30 - AUGUST 2, 2018, LAS VEGAS, USA) http://bitmint.com/SymRand_Vegas_H8518R.pdf
30. "Randomness in digital cryptography: A survey" K Marton, A Suciu, I Ignat - Romanian journal of information science 2010 https://www.academia.edu/download/46676431/Randomness_in_Digital_Cryptography_A_Sur20160621-25262-h5ar54.pdf
31. "Randomness Rising - The Decisive Resource in the Emerging Cyber Reality" Gideon Samid, Int'l Conf. Foundations of Computer Science | FCS'18 | https://www.bitmint.com/RandomnessRising_GSamid_H1o16.pdf
32. "Re-dividing Complexity between Algorithms and Keys" Gideon Samid, International Conference on Cryptology in India, 2001 - Springer https://link.springer.com/chapter/10.1007/3-540-45311-3_31
33. "Rivest Chaffing and Winnowing Cryptography Elevated into a Full-Fledged Cryptographic Strategy" Gideon Samid, 2018, Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE); Athens, (2018). https://search.proquest.com/openview/8ea94f941732d85fb24512d5e7582820/1?pq-origsite=gscholar&cbl=1976356
34. "Secret Signaling System". US Patent 1310719A. Gilbert S. Vernam (1918)
35. "Shannon Revisited: Considering a More Tractable Expression to Measure and Manage Intractability, Uncertainty, Risk, Ignorance, and Entropy" Gideon Samid, https://arxiv.org/abs/1006.1055
36. "SpaceFlip: Unbound Geometry Cryptography." Gideon Samid, https://eprint.iacr.org/2019/285.pdf
37. "Spaceflip: Unbound Geometry Security" US Patent 10,790,977, Sept. 29, 2020
38. "T-Proof" Gideon Samid https://img.chainnews.com/paper/71f69315d015d9fc5dd4ffbc97f87aab.pdf
39. "T-Proof: Secure Communication via Non-Algorithmic Randomization." Gideon Samid, https://eprint.iacr.org/2016/474
40. "Tailored Key Encryption (TaKE)" Gideon Samid, https://eprint.iacr.org/2000/011.pdf
41. "The Myth of Invincible Encryption" Gideon Samid, Digital Transactions May-June 2005

42. "The Rock of Randomness: A physical oracle for securing data off the digital grid": Gideon Samid, Gary Wnek, Material Research Society Bulletin 09 April 2019
43. "The Ultimate Transposition Cipher (UTC)." Gideon Samid, https://eprint.iacr.org/2015/1033.pdf
44. "Threat Adjusting Security" Gideon Samid, https://eprint.iacr.org/2018/084.pdf
45. "Transmitter for Encoding Information with Randomly Flipped Bits and Transmitting That Information Through a Communication Channel", US Patent 10,728,028 Jul 28, 2020
46. "User Centric Cryptography" Gideon Samid, Proceedings of the International Conference on Security and Management (SAM); Athens, (2018) https://www.proquest.com/openview/a60ecf397b6c46373356a1d4369dce5d/1?pq-origsite=gscholar&cbl=1976342
47. "What a 100-year-old Idea can teach us about Cybersecurity" World Economic Forum, Nov 2017 https://www.weforum.org/agenda/2017/11/what-a-100-year-old-idea-can-teach-us-about-cybersecurity
48. "When Encryption is Not Enough--Effective Concealment of Communication Pattern, even Existence (BitGrey, BitLoop)" Gideon Samid, https://eprint.iacr.org/2019/556
49. "Algorithmic Randomness and Complexity" School of Mathematics and Computing Sciences, Downey, R, Hirschfeld, D. Victoria Univ. Wellington, New Zealand. http://www-2.dc.uba.ar/materias/azar/bibliografia/Downey2010AlgorithmicRandomnes s.pdf
50. "Communication Theory of Secrecy Systems" Claude Shannon http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf
51. "Computability and randomness" Niels A. The University of Auckland, Clarendon, Oxford, UK, 2008
52. "Deniable Encryption" Rein Canetti, Cynthia Dwork, Moni Naor, Rafail Ostrovsky CRYPTO '97Volume 1294 of the series Lecture Notes in Computer Science pp 90-104Date: 17 May 2006
53. "Probabilistic Encryption" Goldwasser, Micali, Jr. of Computer and System Science, Vol 28, No 2, pages 270-299
54. "Shannon's Proof of Vernam Unbreakability" https://www.youtube.com/watch?v=cVsLW1WddVI
55. "STRENGTHENING THE SECURITY FOUNDATION OF CRYPTOGRAPHY WITH WHITEWOOD'S QUANTUM-POWERED ENTROPY ENGINE" Richard Hughes, Jane Nordhold http://www.whitewoodencryption.com/wp-content/uploads/2016/02/Strengthening_the_Security_Foundation.pdf
56. "Survey on Cryptographic Obfuscation" Ma t e Horva th 9 Oct 2015 International Association of Cryptology Research, ePrint Archive https://eprint.iacr.org/2015/412
57. "The Unending Cyber War" Gideon Samid, DGS Vitco ISBN 0-9635220-4-3 https://www.amazon.com/Unending-Cyberwar-Gideon- Samid/dp/0963522043
58. "The Code Breakers" David Kahn, The MacMillan Co. 1967.
59. "Edward Snowden: The Untold Story" Wired Mag. Aug 14, 2014
60. "The Innovation Solution Protocol" (Innovation^{SP}), https://InnovationSP.net
61. "Kerckhoffs' Principle" http://www.crypto-it.net/eng/theory/kerckhoffs.html
62. "Equivoe-T: Transposition Equivocation Cryptography" US Patent 10,608,814 March 31, 2020.
63. "SpaceFlip: Unbound Geometry Cryptography" Gideon Samid https://dblp.org/rec/journals/iacr/Samid19.html
64. "Unary Cryptography Demonstration Site" https://UnaryCryptography.com
65. "BitFlip Cyber Demonstration" http://wesecure.net/learn/BitFlipEncrypt.php
66. "SpaceFlip Plus: Ordinal Cryptography" US Patent 11,159,317 * Oct 26, 2021
67. "Efficient Proof of Knowledge of Arbitrarily Large Data Which Remains Undisclosed" US Patent 10,594,480, March 17, 2020.
68. "Cyber Companion: Attaching a Secondary Message to a Primary One" US Patent 10,541,954 Jan21, 2020
69. "Live Documentation (LiDO)" US Patent 10,733,374, Aug 4, 2020
70. "Method for Inhibiting Mass Credentials Theft" US Patent 10,395,053 Aug 27, 2019
71. "Effective Concealment of Communication Pattern (BitGrey, BitLoop)" US Patent 10,673,822, June 2, 2020

72. “Quantum Random Number Generation” https://www.idquantique.com/random-number-generation/overview/
73. "Rock of Randomness" US Patent 10,467,522 Nov 5, 2019
74. "Proving Material Identity with Quantum Randomness -- Financial and General Applications" US Patent 10,754,326. Aug 25, 2020
75. “BitMint Hard Wallet: Digital Payment without Network Communication: No Internet, yet Sustained Payment Regimen between Randomness-Verifiable Hard Wallets” Gideon Samid, 2020 IOT, Electronics and Mechatronics Conference (IEMTRONICS), IEEE International.
76. “Transmitter for Encoding Information with Randomly Flipped Bits and Transmitting That Information Through a Communication Channel” US Patent 10,728,028, July 28, 2020.
77. “Advanced BitFlip: Threat Adjusted, Quantum Ready, Battery Friendly, Application Rich Cipher” US Patent 10,541,808, January 21, 2020.
78. "Split Security Solutions", US Patent Application 17/510,324, Oct 25, 2021
79. "Randomized Bilateral Trust (RABIT): Trust Building Connectivity for Cyber Space (FigLeaf)" U. S. Patent 10,798,065, October 6, 2020.
80. "AI Resistant (AIR) Cryptography" Gideon Samid, IACR Archive https://eprint.iacr.org/2023/524
81. "Tesla Cryptography:" Powering Up Security with Other Than Mathematical Complexity" Gideon Samid, IACR Archive https://eprint.iacr.org/2023/803
82. The Prospect of a New Cryptography: Extensive use of non-algorithmic randomness competes with mathematical complexity", Gideon Samid, IACR Archive https://eprint.iacr.org/2023/383
83. "Applications of Artificial Intelligence to Cryptography" Blackledge, J. & Mosola, N. (2020) , Transactions on Machine Learning & Artifical Intellengence6th June 2020. doi:10.14738/tmlai.83.8219
84. "Understanding Complexity of Cryptographic Algorithms" Baeldung, Bucharest, Romania, Francisco Yepes Barrera, May 2023 https://www.baeldung.com/cs/cryptographic-algorithm-complexity
85. Samid, G. "AI Assisted Innovation". Chapter. https://www.intechopen.com/chapters/75159
86. "Pattern Devoid Cryptography" InfoTech Press, London https://www.intechopen.com/online-first/pattern-devoid-cryptography
87. "Sending Secrets by Sending Only Plaintext" https://eprint.iacr.org/2025/438
88. "Polar Lattice Cryptography" 2025. https://eprint.iacr.org/2025/452
89. "A Different Way to Count, Add, and Multiply" 2025. G. Samid https://www.preprints.org/manuscript/202503.0082/v1