AI Resistant (AIR) Cryptography

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

Abstract: highlighting a looming cyber threat emanating from fast developing artificial intelligence. This strategic threat is further magnified with the advent of quantum computers. AI and quantum-AI (QAI) represent a totally new and effective vector of cryptanalytic attack. Much as modern AI successfully completes browser search phrases, so it is increasingly capable of guessing a rather narrow a-priori list of plausible plaintexts. This guessing is most effective over device cryptography where the message space is limited. Matching these guesses with the captured ciphertext will greatly accelerate the code breaking process. We never faced such a plaintext-originated attack on a strategic level, and never had to prepare for it. Now we do. Proposing to apply a well-known martial art tactics: using the opponent's strength against them: constructing ciphertexts that would provide false answers to the AI attacker and lead them astray. We are achieving this defensive measure by pivoting away from the norm of small, known-size key and pattern-loaded ciphers. Using instead large keys of secret size, augmented with ad-hoc unilateral randomness of unbound limits, and deploying a pattern-devoid algorithm with a remarkably low computational burden, so it can easily handle very large keys. Thereby we achieve large as desired unicity distances. This strategy has become feasible just when the AI threat looms. It exploits three new technologies coming together: (i) non-algorithmic randomness, (ii) very large and inexpensive memory chips, and (iii) high throughout communication networks. These pattern-devoid, randomness rich ciphers also turn up to be an important option in the toolbox NIST prepares to meet the quantum challenge. Avoiding the computational load of mainstay ciphers, AIR-cryptography presents itself as the ciphers of choice for medical, military and other battery-limited devices for which data security is paramount. In summary: we are pointing out a fast emerging cyber challenges, and laying out a matching cryptographic answer.

1.0 Introduction

Laying down the theoretical framework for cryptography Claude Shannon in his seminal paper [6] has defined cryptanalysis as an operation that changes the a-priori probability distribution over the message space to posteriori distribution of smaller entropy. Shannon's starting point was the notion that the cryptanalyst claims no circumstantial knowledge and hence regards all possible messages as equally likely. Shannon further introduces the notion of 'unicity distance' -- the message size, above which the ciphertext commits to a single key, as long as the plaintext is assumed to be a standard human language, like English. Shannon computes this unicity distance for a simple substitution cipher to be as few as 50 letters of the English alphabet. Alas, he points out that it is possible to use 'ideal systems' which extend the unicity distance further and further.

The traditional strategy for cryptanalysis was to "hammer the ciphertext" and extract from it the committed plaintext. Cryptographers built greater and greater complexity to frustrate this effort and cryptanalysts, time and again, outsmarted the cryptographers, first by cracking their code, second by hiding the fact that the code was cracked, and third, by actively promoting the idea that the cracked code is unbreakable. This is the prevailing game today.

This configuration is about to change. A new technology has emerged, a powerful new cryptanalytic tool. The ignorance-based a-priori distribution of message candidates to be the one encrypted into the given ciphertext is now being replaced by an AI-deduced distribution of much lower entropy. Many authors remarked on the similarity between cryptography and machine learning. The mathematical formality is very similar [5,8,12] . A major difference is in the fact that the cryptanalyst faces an adversary who does its best to make it difficult for the cryptanalyst to learn the plaintext from the ciphertext. Such actions are dynamic and unpredictable. By contrast, in machine learning the learner (the equivalent to the cryptanalyst) is contending with a disinterested 'enemy'. The body of data being hammered by the machine earning network is indifferent to this effort. Another difference is that the cryptanalyst has very limited data to work with -- only the ciphertext, while the learning network sends its tentacles to far and away data of limited but not negligible relevance. Network are increasingly adept to properly account for partially relevant information sources.

These are two very powerful advantages for machine learning over cryptanalysis. Together they suggest to the cryptanalyst a strategy of shifting as much effort from straight cryptanalysis to the increasingly powerful AI tools.

In practice it means that instead of using the flat a-priori distribution list, the cryptanalyst will deploy AI tools to collapse the disruption list to a low entropy version, or say, to identify a limited number of highly plausible plaintext messages, so that one of them is the sought after message -- the one used to generate the ciphertext. Because of the reasons mentioned above the

work of the cryptanalyst will be easier if he or she devotes part of their efforts to machine learning for the purpose of generating a low entropy a-priory list.

AI network designers for other purposes have been consistently surprised at the level of deduction exercised by the AI. One would expect a similar surprise with respect to extracting hidden pattern from the monitored circumstances of the cryptography users; creating a strategic advantage for the code breaker.

At first glance it appears that the cryptographer is helpless. This emerging vulnerability is not in their domain. It is in the pre-cryptographic stages. The AI extracted a-priori distribution is deduced from circumstantial data involving mainly the transmitter. The cryptographer is not in the picture at that stage, yet the cryptographer fails because of the enormous amount of circumstantial information exposed by the transmitter and hunted and processed by the cryptanalytic AI.

This is the challenge that cryptographers must get ready for. Nothing that appears to be developing in the offing serves as a good answer. The much expected, and much debated post-quantum ciphers are not going to be much of a help because again, the cryptanalyst exploits pre-cryptography patterns associated with the users and especially with the transmitter. Every cipher is easier to compromise given both the ciphertext and the matching plaintext rather than only the ciphertext. A recent published work showcases just this point [12]. And since the nominal ciphertext commits to a single plaintext all that is needed in order to discard an a-priori likely plaintext is to determine whether it is likely that a pair of ciphertext and plaintext candidate can be matched with a standard key out of the well-known key space. This is an easier mathematical challenge than to extract the plaintext and the key out of the ciphertext -- with respect to all mainstay ciphers as well with respect to the post quantum cipher candidates.

Cryptographers need to think out of the box.

1.1 AI v. Cryptography Literature.

Ronald Rivest in 1999 [8] observed that: “Machine learning and cryptanalysis can be viewed as Sister fields, since they share many of the same notions and concerns.” He compares the AI challenge with the cryptographic challenge: “This problem can also be described as the problem of learning an unknown function (that is, the decryption function) from examples of its input/output behavior and prior knowledge about the class of possible functions.”

Rivest further observes that since in classical cryptographic schemes the key size is known, the unicity distance is also well defined, which is in opposition to machine learning where the space

of the target functions is of unknown size. The described AIR cryptography moves closer to AI in as much as the key size is unbound from the point of view of the cryptanalyst.

Rivest clearly admits that machine learning techniques would be helpful for the cryptanalyst holding matching pairs of ciphertext/plaintexts. We argue that it is much easier to appraise the likelihood for a random pair to have a matching key, as is exemplified by So. [12]

Anees [5] claims that “it is now more relevant to apply ML techniques in cryptography than ever before. Their paper offers an extensive review of nineteen publications threading AI with cryptography.

2.0 AIR Cryptography Strategy

No sooner does one recognize the emerging cryptographic vulnerability to modern AI technology, the solution strategy emerges with a touch of obviousness. The AI attack is wielded with a limited a-priori list of plaintext candidates. Let therefore the ciphertext be constructed with a terminal list of plaintexts that each qualify as the generating plaintext. To the extent that this terminal list is closer to the a-priori list, that is the extent to which the ciphertext is protected from the onslaught of AI.

Any 'distance' and difference between the terminal set and the a-priori set will create a cryptanalytic opening which the code writer can control the size thereof, by simply adjusting the terminal set.

In other words one develops cryptographic resistance (AIR) to AI aided cryptanalysis by pivoting away from the standard practice of using ciphertexts that commit to their generating plaintext. The price of creating such non-committed ciphertexts is larger keys, and greater communication burden.

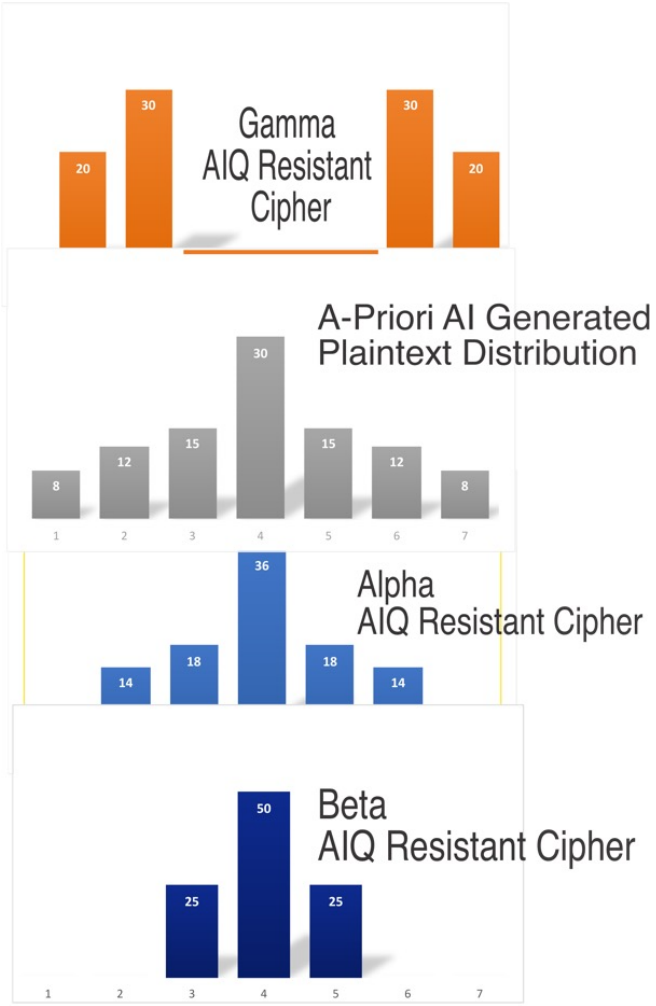
2.1 Larger keys

Claude Shannon has shown [6] that for a key the size of the processed message, the ciphertext may be perfectly secret on account of de-facto creating a terminal list that is identical to the a-priori list. For a key a bit smaller than the processed message the terminal list will be smaller and more unflat than the a-priori list but the user controls by how much.

We consider ciphers for which the key size is part of the key secret, and where larger keys do not impose an exponential increase in nominal processing. Hence tracking computational load (side channel cryptanalysis) will not be cryptanalytically helpful. In particular, we consider ciphers which when they are used with keys smaller that the processed message, they point to a terminal list that while smaller than the a-priori list, is nonetheless inclusive of the plaintext candidates of the highest probability. In other words the effective AI resistant cipher will drop low probability options from the terminal list, and keep all the high probability options in tact. Thereby limiting the cryptanalytic vulnerability created by the smaller key. Note: such strategy cannot be monitored via entropy calculations because for it to work there must be no change with respect to the high probability candidate plaintext in the a-priori list.

Illustration: Let P_1, P_2, ... P_7 be seven plaintext candidates pointed to by an AI engine processing all the relevant information. The AI inference assigned these seven plaintext options the following probabilities respectively: 8%, 12% 15%, 30%, 15%, 12%, 8%. A perfect AI-resistant ciphertext will be associated with seven distinct keys K_1, K_2, K_3, K_4, K_5, K_6 , K_7 each decrypting the ciphertext to one of the seven plaintexts listed by the AI engine. In that case the cryptanalyst learns nothing from the ciphertext -- perfect secrecy.

Consider now an AI-resistant cipher , α, with a somewhat smaller key space such that it is associated only with five out of the seven keys: K_2, K_3, K_4 , K_5, K_6 . This will evaluate to a respective terminal list of five plaintext candidates: P_2, P_3, P_4 , P_5, P_6 . The gap between the α terminal list and the a-priori list is minimal. Only two low-probability plaintext candidates were eliminated. The cryptanalyst will now recalculate the probabilities for the remaining candidates according to the former information resulting in the seven candidates having the following probabilities by order: 0%, 14%, 18%, 36%, 18%, 14%, 0%.



AIQ Resistant Cipher Options

We may now consider another AI-resistant ciphertext of smaller size so that only three keys remain viable: K_3, K_4, K_5, reducing the a-priori list to the β-terminal list: P_3, P_4, P_5. In this case the cryptanalyst has better results, four candidates are eliminated and the new probability distribution is: 0%, 0%, 25%, 50%, 25%, 0%. 0%. Alas even this smaller weaker cipher leaves in tact the high probability plaintext candidates, so the cryptanalytic gain is limited.

Lastly we consider an even weaker AI resistant cipher for which the ciphertext is associated with four keys: K_1, K_2, K_6 , K_7, and hence the list of viable plaintext candidates is reduced to P_1, P_2, P_6 , P_7. In this case the high probability candidates are eliminated, which implies a big cryptanalytic impact relative to a state of ignorance of the ciphertext. The respective probabilities for the cryptanalyst are: 20%, 30%, 0%, 0%, 0%, 30%, 20%.

This illustrates the battleground between the AI resistant cipher user and the respective cryptanalyst. The former wishes to deploy a ciphertext associated with a terminal list as close as possible to the a-priori list and the cryptanalyst wishes to exploit whatever gap there is between the two lists.

2.2 Communication Burden

We consider a class of ciphers called decoy tolerant. These are ciphers where the content-bearing bits (CBB) are mixed with content-free bits (CFB), so that the recipient of the combined flow can readily distinguish between these two classes, discard the CFB and decrypt the CBB. To the extent that both the CBB and the CFB appear random, and indistinguishable, the attacker must regard the entire ciphertext bit flow as content bearing. The cipher may be set up so that the CFB will be configured such that they will point to cryptographic keys which will decrypt the total bit flow to any number of plaintext candidates. Since the ciphertext does not point to the key used to decrypt it, it is impossible for the cryptanalyst to identify the key actually used to generate the bit flow from the various other keys that fit with the same ciphertext. For a sufficient long list of CFB the cipher will generate a terminal list that would be as close as desired to the a-priori list.

2.3 Tailored Terminal Lists

The most aggressive defense strategy against AI and AIQ threats is to apply AIQ in order to establish a defensive a-priori list reflecting the circumstances as seen by the adversary. The idea is to mimic the attack a-priori list as much as possible. Let P^*_1, P^*_2, ... P^*_n reflect the n

plaintext candidates pointed to by the AIQ capability employed by the defense -- the cipher users. Ideally all the high-probability plaintext candidates in the defensive list will be built in to the ciphertext. That way, to the extent that the attack a-priori list is similar to the defensive a-priori list the cryptanalyst will find the attack on the ciphertext to be unproductive, it would have no or minimal impact on their original a-priory list.

All decoy tolerant ciphers may be operated via some q alphabets A_i, A_2, ... A_q where each alphabet uses different bit strings, and where sending a string b_{ij} that points to letter j in alphabet i (a_{ij}) is regarded as decoy by all interpreters in possession of none, one or any of the alphabets other than A_i. That way q messages may be sent such that the resultant ciphertext decrypts to each of these q messages, (q-1) of them are decoys.

3.0 QAI Resistant Ciphers

We offer here a cursory description of AI and QAI resistant ciphers, which can be reviewed in details following the respective reference:

3.1 BitMap:

This cipher [10] is based on a simple principle: a travel path may be defined either through the series of visited destinations, or through an ordered list of the travel roads. Anyone holding a map will readily shift from one expression, say list of visited places, to the other expression: list of traversed roads. Or vice versa. Absent the map one would be able to construct at will maps that would match a given plaintext to a given ciphertext. A list of destinations: a, b, c, d. Can be matched to a list of roads x, y, z by constructing a map where road x leads from a to b, road y leads from b to c, and road z leads from c to d. Given another candidate list of roads, say p, q, r, one will construct a map where road p leads from a to b, road q leads from b to c and road r leads from c to d. For a key large enough such that the travel paths don't intersect such cipher will present to the QAI cryptanalyst a terminal list every bit as large as the a priori list with which the attack is made. Hence -- zero vulnerability.

When the amount of message traffic used with BitMap increases to the point that the travel paths begin to intersect, then the perfect secrecy is lost. However, if it is well done then the loss is minimal, it affects the low probability candidates in the a-priori list.

The bitmap key is of secret size. In other words, the attacker does not know how large is the map. It is so designed to make it a secret where the travel beings and where it ends. Furthermore BitMap is a powerful decoy-tolerant cipher.

Implementing BitMap decoy tolerance: The idea is surprisingly simple. One selects an alphabet comprising (n-1) letters which are mapped to any common digital language say ASCII or Base64. We call it the payload alphabet. Humanly readable letters are then defined as a string of the payload letters. These letters are all of fixed size so they can be concatenated to a sequential string of the payload letters. The plaintext written with the payload alphabet will have occasions where a letter will repeat itself may be even more than once. One then interjects between any such repeating letters another letter the n-th letter in count. By so doing all the repetitions are removed and the plaintext is written with the adjusted payload alphabet comprising n letters. That plaintext has one distinct property, it has no repetition of any letter. Each letter in the sequence (except the first and the last) is preceded and followed by a letter different than itself.

The no-repetition n-letter payload alphabet string is regarded as processed plaintext. Should one take the processed plaintext and replace any letter there in with a sequence of same letters, then the result will be readily reversed by the intended reader of this string: simply collapsing all letter repetitions to a single letter.

In the BitMap setting one writes many spots in the map as named with the same letter so that a travel path over such a zone generates a repetition of the same letter, which in turn is not confusing the intended reader who will replace all repetitions with a single occurrence of the letter. The view from the eyes of the cryptanalyst that is not aware of the map and the same letter zones therein is that each letter may be a different one. Letter repetition creates decoy tolerance that burdens the cryptanalytic chore for the attacker at the same time it is readily negotiated by the intended reader.

3.2 BitFlip.

This cipher [11] is based on 'one-to-many-many-to-one' (O2M-M2O) relations. Let a be a bit string denoting a certain letter a*, in a given alphabet. In a simple cipher the relationship a-a* is kept secret such that a transmitter sends letter a* to a recipient by sending him the string a. These early ciphers are readily compromised via letter frequency analysis and similar methods. We consider now a different string b, and a decision relation R. R will take both a and b as input and generate a decision δ_R {0,1} = {no, yes} .

The transmitter will send letter a* to the recipient by sending them a string b for which $\delta_R(a,b) = 1$, over a well defined relation R.

The relation R can be chosen so that there are many strings $b_1, b_2, , ...b_i$ for which

$\delta_R(a, b_i) = 1$.

And also for any string b there are many strings a_1, a_2, ... such that $\delta_R(a_j, b) = 1$. This is the one-to-many-many-to-one relation.

Given A, an n letters alphabet a_1, a_2, ... a_n, and given a message m comprising a series of letters from A. A transmitter will send m to a recipient by sending it letter by letter in order. Each letter a_i will be sent by selecting a b string from a large list b_{i1},b_{i2}....., such that $\delta_R(a_i, b_{ij}) = 1$ for j=1,2,.... while for any other letter a_k for $k \neq i$ there exists: $\delta_R(a_k, b_{ij}) = 0$, for j=1,2,…

The recipient will evaluate a string b_{ij} to point to letter a_i only if the above terms are met. Namely the transmitted string b_{ij} evaluates to "1" based on the relation R while it evaluates to "0" with respect to all other letters in alphabet A. Any transmitted string that does not hold up to these terms is considered 'decoy' and is discarded.

As described such a cipher is clearly decoy tolerant. The relaion R may be chosen such that the one-to-many and many-to-one attributes will be so rich that given a series of b strings (a ciphertext) there are numerous possible a strings to represent a given alphabet A, thereby creating a large as desired terminal list.

For BitFlip the relation R was selected as the Hamming distance between strings a and b. For each of the n letters of A there exist a secret string a and a secret Hamming distance t. $\delta(a, b) = 1$ if and only if the Hamming distance between strings a and b is t: $t = H(a, b)$. So to send a letter a_i the transmitter will choose a string b_{ij} such that:

$$t_i = H(a_i, b_{ij})$$

and

$$t_i \neq H(a_k, b_{ij}) \text{ for } k= 1,2,...(i-1),(i+1),....n$$

The number of b strings that share the desired Hamming distance t from a is rising exponentially with the size of a, and the number of a strings with which b has a Hamming distance t is also rising exponentially with the size of b (where $|a| = |b|$). By selecting the size of a and b the users control the size of the terminal list.

4.0 Pattern Devoid Cryptography

The threat projected from AI and more so from QAI is grounded in the surprising capability of AI to detect patterns beyond human visibility, even challenging human understanding after a pattern has been spotted. The pre-encryption side (the plaintext side) of the cryptographic game

is extremely patten-heavy. People resort to cryptography in order to handle particularly sensitive situations -- loaded with pattern, and hence a fertile ground for AI to analyze and emerge with an a-priori list of plausible plaintexts under the given circumstances. This can't be helped.

Defense against that threat can come wholly from the ciphertext side, and that by suppressing pattern to the extent possible in the way the plaintext is processed into the ciphertext. Pattern loaded ciphers are a juicy target for persistent AI and QAI. The seriousness of this threat warrants an in-depth evaluation of the direction chosen for cryptography in the foreseeable future.

The series of ciphers presented as AIQ resistant do belong to a class of pattern-devoid cryptography where randomness, shared and unilateral, is used very richly to mount a cryptanalytic barrier to equal and even supersede the security projection of the mainstay ciphers today. These ciphers are regarded as Trans-Vernam because they all regard the one-time-pad Vernam cipher as their legacy predecessor. Yet, Trans Vernam ciphers are more advanced in many ways. They are more convenient to use; they are more secure because they don't disclose the size of the plaintext, they are decoy-tolerant and they offer tailored capability towards setting up effective terminal lists of plaintext candidates.

5.0 Device Cryptography

Life is operated and carried by increasingly more and more devices of all sorts: utility, medical, law enforcement, public service, military, etc. Many of them are battery operated and rely on encryption to report their readings and to receive their commands. The language used by these devices both for sensory reporting and for control is quite limited. The a-priori message space is small compared to what people say in their encrypted messages. And as such device cryptography is a more effective target for AI-cryptanalysis. Which in turn means that device cryptography should be first in line to deploy AIR cryptography as described herein.

6.0 Outlook

The power of AI keeps surprising its designers, there is no visible limit to the measure of conclusions that AI and in particular QAI will extract from reams of data of some relevance to a given issue. One must account for the possibility that AI cryptanalysis will become highly effective to amount to a catastrophic threat on life in cyberspace. This in turn calls for alarm and awareness of the need to develop AI resistant cryptography to stay one step ahead of this threat.

Reference

This article is a direct extension of the following two publications:

1. Samid, G. “The Prospect of a New Cryptography: Extensive use of non-algorithmic randomness competes with mathematical complexity” https://eprint.iacr.org/2023/383
2. Samid, G. “Pattern Devoid Cryptography” https://eprint.iacr.org/2021/1510

The AIR Cryptography mentioned in this article is demonstrated in the following interactive demonstrations:

3. BitFlip http://wesecure.net/learn/BitFlipEncrypt.php
4. Unary Cryptography https://unarycryptography.com

Other References:

5. Anees. A et al “Machine Learning and Applied Cryptography” https://doi.org/10.1155/2022/9797604
6. Claude Shannon. Communication theory of secrecy systems. Bell System Technical Journal, 28:656-715, October 1949.
7. Hellman, M 1. “An extension of the Shannon theory approach to cryptography”. IEEE Transactions on Information Theory, V. 23, 3 1977, pp. 289 – 294
8. Rivest R “Cryptography and Machine Learning.” https://people.csail.mit.edu/rivest/pubs/Riv91.
9. Samid, G. "Shannon Revisited: Considering a More Tractable Expression to Measure and Manage Intractability, Uncertainty, Risk, Ignorance, and Entropy" https://arxiv.org/abs/1006.1055
10. Samid, G. "BitMap Lattice: A Cyber Tool Comprised of Geometric Construction", US Patent 10,911,215, Feb 2, 2021
11. Samid, G. "BitFlip: A Randomness Rich Cipher" 2017, Gideon Samid, Sergei Popov, https://eprint.iacr.org/2017/366.pdf
12. So. J. “Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers,” https://towardsdatascience.com/where-machine-learning-meets-cryptography-b4a23ef54c9e Claude Shannon. Communication theory of secrecy systems. Bell System Technical Journal, 28:656-715, October 1949.M

For an entertaining read check out "The Cipher Who Came in from the Cold" a recently published thriller envisioning the way the CIA, the NSA and the FBI meet the challenge of the new cryptography.