AI Brute Force Cryptanalysis-An Underestimated Threat

Gideon Samid *

Posted Date: 19 May 2026

doi: 10.20944/preprints202605.1168.v1

Keywords: supervised learning; brute force cryptanalysis; threat vector

 doi:10.20944/preprints202605.1168.v1

Article

AI Brute Force Cryptanalysis—An Underestimated Threat

Gideon Samid

Electrical, Computer and System Engineering, Computer and Data Sciences, Case Western Reserve University, Cleveland, OH; gideon.samid@case.edu

Abstract

AI beats humans in chess, exceeds humans in language translation, drives more safely than most of us -- we better believe that it can brute-force analyze ciphertexts and extract from them the secret plaintext -- no matter how much math complexity is packed into the exposed data. The noted cryptographer Adi Shamir, predicted years ago: "Encryption," he said, "would not be cracked, it would be circumvented." Which is exactly what AI does to cryptography. No matter how hidden the pattern, AI will discern it. Post quantum cryptography may or may not be effective against quantum computers, but it would surely be ineffective against the brute force analysis of neural networks and other AI means. The only way to survive a charging Grizzly bear is to confuse it by arriving at a junction that spreads to many roads, letting the bear roam ahead in the wrong direction. The only way to escape the AI onslaught is to meet AI with waves of entropy -- content-devoid bits, to sidetrack the AI bear off and away. Fortunately, academia is ready with a new class of tools: Pattern Devoid Cryptography. Explained here. No time to wait, the AI bear is shaking the ground under our feet.

Keywords: supervised learning; brute force cryptanalysis; threat vector

1. Introduction

Throughout the history of cryptography there was one method to crack secrets which was guaranteed to work. It was called "Brute Force Cryptanalysis". You simply try all the possible keys until you find the one that translates the ciphertext to the plaintext. You can't go wrong! The only problem was time. Cryptographers built a large "key space" -- listing so many possible keys that even if thousands of computers try in tandem it would take them thousands of years on average to spot the right key. So the method worked, but it was too slow.

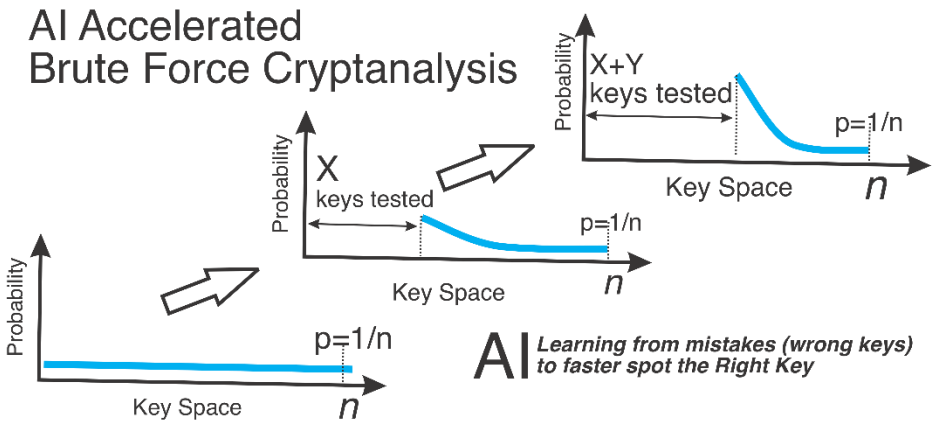
Along came artificial intelligence, AI. This is a capability to crunch data, hammer it, massage it, and extract from it its full conclusion potential. When one uses a wrong key over a captured ciphertext, they get a wrong plaintext. Before the advent of AI this 'wrong plaintext' was tossed away, no good. But AI says: wait a minute. The wrong plaintext that emerged out of applying the wrong key, does hold information about the right key. Indeed, when AI tests m keys out of the key space containing n keys, it is collecting m 'bad plaintext' results: a wealth of information. That is exactly what AI needs in order to do its magic, see pattern, extract conclusions.

So when m keys have been tested, the remaining n-m keys can be rank ordered by their probability to be the right key. What does AI do next? It tests the remaining keys starting with the high probability ones. If lucky the right key is found right away. Otherwise, after checking additional t keys, then AI has m+t bad plaintext data to evaluate, which in turn creates a more spiked probability curve over the remaining n-m-t keys. On it goes, leading to replicating the pre-AI brute force analysis in a fraction of the time: seconds instead of years!

And it does not matter how complex, how mathematically loaded is cipher, AI ignores the internal complications of the cipher, it simply analyzes its output in response to its input.

So it looks like all the traffic on the Internet will become public, no encryption will survive the onslaught of AI. But that is not the case. We can protect ourselves by using non-trivial ciphertexts.

 doi:10.20944/preprints202605.1168.v1



2. Non Trivial Ciphertexts

Trivial ciphertexts are written as a string of content-bearing bits. Every bit is important in. processing the ciphertext into the corresponding plaintext. This is known as "the avalanche effect". The key holder extracts the plaintext from ALL the bits of the ciphertext. It takes the AI user longer, but through "learning from mistakes" the right plaintext is flashed out.

Non-trivial ciphertexts are different. Content-devoid bits are being mixed with the content-bearing bits. The power of these non-trivial ciphers is that the intended reader, using a secret content/void discrimination key can identify which bits are content-bearing and which ones are content-devoid. Then the intended recipient brushes away the content-devoid bits, and remains with the content bearing bits -- now it is a trivial ciphertext for which the recipient has a key to process it into the plaintext.

What happens with the AI attacker? This attacker does not possess the discrimination key that tells him if a bit is content-bearing or content-devoid. The attacker suspects that each of the bits in the non-trivial ciphertext is content-bearing -- and AI analyzes it. Because of the Avalanche Effect, mis-designating even a single bit (content bearing or void) will prevent the attacker from extracting the right plaintext.

What happens when AI crunches content-devoid bits? Hallucination -- the grand failure of AI. Hallucinations come to play by seeing in the ciphertext a wrong plaintext and believing it is the right one.

What is more: the transmitter is free to pump in as many content-devoid bits as they like. It does not disturb the intended recipient, they brush them off bit by bit. The picture is completely different from the point of view of the AI attacker: the more bits before them, the greater the confusion, the wilder the hallucinations.

Non-trivial ciphertext are generated by Pattern Devoid Ciphers. Since these ciphers have no pattern, they have no pattern for AI to crack.

As AI emerges and becomes more potent, cryptography is likely destined to gradually abandon the mathematical complexity ciphers they use today, and shift to non-trivial ciphers.

Short Review of Common PDC Ciphers

Here is how BitFlip [4] works: given an alphabet A comprising n letters a_1, a_2, ... a_n, let every letter be associated with an arbitrary/random number of key bit strings, where each key string is of arbitrary/random size. These key strings are considered the cryptographic key. When Alice wishes to send Bob letter a_i she randomly selects one of the key strings, associated with a_i, k_{ij} and sends Bob a string s_i which marks a Hamming distance h from k_{ij}:

$$h = Hamming(kij, si)$$

Note: The Hamming Distance can be replaced by any of the other distance metrics mentioned herein as well as others [9].

 doi:10.20944/preprints202605.1168.v1

Then Alice checks the Hamming distance between s_i and all the strings associated with the rest of the alphabet (this is the 'Confusion Test'). If any of these Hamming distances equals to h, then Alice builds another string s'_i for which:

$$h = Hamming(kij, s'i)$$

and sends it out. Then she runs again the Confusion Test. If it fails, she repeats until the Confusion Test holds.

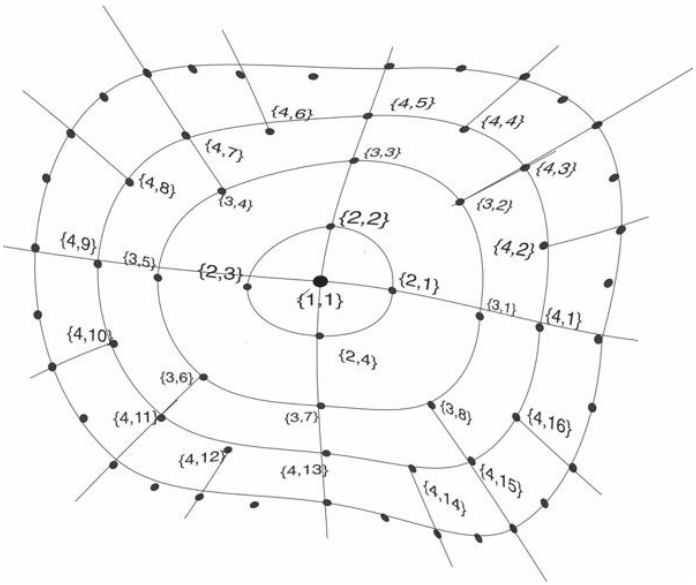
Bob will ignore all strings that fail the confusion test and read the letter a_i once it is confusion free.

This simple routine is exercised letter after letter. Alice can also send Bob a string that shows no Hamming distance h from any key string -- Bob will ignore it too.

The BitFlip cipher is immunized against AIA Brute Force attack. The key is of unknown size, there is no complex math to short-cut, and the ciphertext is not trivial, it contains an unknown amount of noise.

Polar Lattice Cryptography is another example for Pattern Devoid Ciphers that are immunized against AI Accelerated Brute Force attack [5]. Consider an arbitrary number of concentric circles with an arbitrary number of rays emanating from the shared center. Different rays extend to different circles. Every intersection between a circle and a ray is an addressable point. Each letter of the alphabet is associated with a starting point and a terminal point. When Alice wishes to communicate to Bob a given letter she marks an arbitrary path from the respective starting point to the respective terminal point. The path is defined by a series of points. Steps are taken: up/down (along the ray), right/left (along the circle) from one addressable point to the next. If the path leads from the starting point of letter a_i to the terminal point of letter a_i, and there is no confusion with any other letter then Bob knows that Alice sent the letter a_i. Here too, the ciphertext includes open ended unilateral randomness that flows from Alice to Bob -- AIA-BF attack defeated.

Topological Resilience of the Polar Lattice



3. Literature Survey

Ronald Rivest, [17], declared "Machine learning and cryptanalysis can be viewed as Sister fields, since they share many of the same notions and concerns." Ever since this observation has been challenged because cryptography deals with exact values, while AI is most powerful by dealing with progressive approximations, and measured similarities.

It is only lately that the literature has addressed the question of AI as a new worrisome cryptanalytic tool. Some trace the interest to Gohr's article [20] which addressed a very specific case, (Speck32/64). Gohr concluded that "In the setting considered, this works reasonably well". The

 doi:10.20944/preprints202605.1168.v1

neural networks needed only a few minutes to be trained. This led Gohr to insist that AI may develop to become a very powerful cryptanalysis agent. Gohr asserts: "This paper is the first to show that neural networks can be used to produce attacks quite competitive to the published state of the art against a round-reduced version of a modern block cipher."

Several researchers have applied AI techniques for other limited situations (DES, RC4) [21–24]. None reported an alarming breakthrough. More recent publications, [19], report limited attempts to unleash neural networks on a cryptanalytic challenge, with no impressive results. These attempts may be judged as too narrow.

All in all the literature reflects a timid approach to the proposition that advanced AI is rising to become a front-line serious threat vector to mainstay cryptography. By comparison, Peter Shor published his paper on the threat of quantum computers [25] in the early 90's. It took almost a quarter of a century for the cryptologic community to arm itself against this threat. Nonetheless, AI today is much more advanced than quantum computing was in the 90s. The AI threat vector may present itself much faster.

A certain elaboration of the material herein can be found in [29].

4. Randomness Defeats AI

AI extracts pattern from situations that any non-AI tool, including human intuition, has concluded to be 'purely random'. AI neural networks discern order that remained veiled before the best non-AI math tools. AI is therefore truly revolutionary. AI spots diseases, discovers new medicine, discerns archeological data, and as practiced by the author establishes AI Assisted Innovation [6] all beyond non-AI capability. Cryptography is no exception.

Yet, AI can be "poisoned", it can be defended from. In particular, randomness defeats AI onslaught. Randomness is a subtle notion. Any finite random series examined by AI will yield some expressed pattern that would be totally false, namely not a pattern that was hidden in the apparent randomness, rather AI hallucination.

Applied to cryptography randomness can be enhanced over the key, and over the ciphertext, creating a randomness-packed cipher that can withstand quantum attack, AI attack, and any yet unimagined pattern-seeking attacks on the captured ciphertext.

The shared key used by the communicators is random, however in all common ciphers the key is of known size and durable throughout the communication session and beyond. Extra randomness can be added by building a key of secret size, and using a randomized part thereto for every new session, even for every new statement, or with maximum variability, changing the key for every new letter -- totally voiding any AIA-BF attack.

A potent and robust defense is one that resorts to ciphers that allow for unilateral randomness to be injected by the transmitter such that the intended reader will identify it and ignore it but the AIA BF cryptanalyst will vainly attempt to spot pattern thereto.

It is noteworthy that NIST recently recognized the cryptographic power of unilateral randomness which is well used in its "Learning With Errors" Cryptography. [8].

A whole new class of ciphers exemplifies the above and serves well as AIA-BF defense. The class is known a Pattern Devoid Cryptography [2]

5. AIR-AI: AI Resistant AI: AI Perfect Cipher

AI can be used to construct the perfect security cipher that resists all cryptanalytic attempts including AI cryptanalysis. Claude Shannon has proven [3] that the Vernam Cipher rightly implemented, offers perfect secrecy. He proved it by showing that knowledge of the captured ciphertext does not change the cryptanalyst outlook on the set of probable plaintexts. Namely before capturing the ciphertext, the cryptanalyst listed n plaintext candidates P_1, P_2, ...P_n to be the plaintext encrypted and sent out as the public clear ciphertext; and after omnipotent hammering and squeezing

 doi:10.20944/preprints202605.1168.v1

of the ciphertext, the cryptanalyst remained with the same list of candidates. In other words, security is perfect if knowledge of the ciphertext does no impact the series of probable plaintext candidates.

Applying the same principle here let P_1, P_2, ...P_n be the list of probable plaintext candidates. While P_1 is the plaintext that is actually sent to the intended recipient. The transmitter will choose *n* random keys K_1, K_2, ... K_n then encrypt P_i with K_i to generate C_i. Next the transmitter will cast ciphertexts C_2, ..C_3, ... C_n as noise relative to key K_1 and combine the (n-1) decoy strings with C_1 to build the combined ciphertext CC. The intended reader will ignore the decoys and decrypt CC to P_1 (using the shared key K_1). The attacker using AI, even privy to quantum computing , and being omni math-talented, at most will extract the *n* plaintext candidates they were listing before cryptanalyzing CC, and hence CC offers perfect security.

AI is used by the transmitter to generate plaintexts P_2, P_3,... P_n from knowledge of P_1. [12]

6. Outlook

In principle today's cryptography relies on complex mathematics. This is the stuff AI is good at cracking, even though nothing spectacular has been shown yet with respect to cryptography. There are plenty of examples [26] demonstrating how neural networks, Gaussian processes and symbolic regression retrieve mathematical complexities. AI is free to 'play' with any cipher, compare input and output to extract its hidden patterns. AI is extremely powerful with approximations and similarities and therefore, as presented here, the accelerated brute force strategy appears the most promising. And while not much academic interest has been demonstrated so far, it may well be that a great deal of action is hidden in the clouds of global geopolitics.

The impact of quantum computing on cryptography took a quarter of a century to be properly regarded. We should be faster with respect to the AI threat. We do have robust solutions. The general outlook for cryptography is to be moving away from trivial ciphertexts and known, fixed size keys. Future cryptography will contaminate' the ciphertext with unilateral randomness and deploy secret size keys which are randomly cut. Cryptography of the future appears likely to migrate from "Hidden Pattern Cryptography", HPC, today to "Pattern Devoid Cryptography", PDC [2] tomorrow.

References

1. "Weak Keys" YouTube https://www.youtube.com/watch?v=NazOCJio_6w&t=101s
2. Samid, "Pattern Devoid Cryptography" https://www.intechopen.com/online-first/pattern-devoid-cryptography
3. Claude Shannon. Communication theory of secrecy systems. Bell System Technical Journal, 28:656-715, October 1949.
4. Popov, Samid .BitFlip: A Randomness-Rich Cipher IACR 2016/627
5. Samid, "Polar Lattice Cryptography" https://www.opastpublishers.com/open-access-articles/polar-lattice-cryptography.pdf
6. Samid, "Artificial Intelligence Assisted Innovation, AIAI". https://www.intechopen.com/chapters/75159
7. List of Relevant Patents https://patents.justia.com/inventor/gideon-samid
8. Regev, O. (2005). *On lattices, learning with errors, random linear codes, and cryptography*. In Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC) (pp. 84–93).
9. Joshua Noble, 2022, "What are distance metrics?" IBM https://www.ibm.com/think/topics/distance-metrics
10. Samid, ".Lifeboats on the Titanic Cryptography " IACR 2025/587
11. Samid, ""Tesla Cryptography:" Powering Up Security with Other Than Mathematical Complexity " IACR 2023/803
12. Samid, "AI Resistant (AIR) Cryptography " IACR 2023/524
13. Samid, "The Prospect of a New Cryptography: Extensive use of non-algorithmic randomness competes with mathematical complexity" IACR 2023/383
14. Chhetri et al "Post-Quantum Cryptography and Quantum-Safe Security: A Comprehensive Survey" arXiv:2510.10436

 doi:10.20944/preprints202605.1168.v1

15. Samid, G. (2025). Non-Trivial Ciphertexts: Decryption Variety, Contents Discrimination. *J Electr Comput Innov*, *2*(2), 01-09.
16. Emanuele Bellini & Anna Hambitzer "Limitations of the Use of Neural Networks in Black Box Cryptanalysis" Conference paper First Online: 13 October 2022 p 100–12 https://link.springer.com/chapter/10.1007/978-3-031-17510-7_8#citeas
17. R. Rivest "Cryptography and Machine Learning Laboratory for Computer Science Massachusetts Institute of Technology Cambridge, MA 02139
18. SLOANE, N. J. A. (1982). ERROR-CORRECTING CODES AND CRYPTOGRAPHY PART II. *Cryptologia*, *6*(3), 258–278. https://doi.org/10.1080/0161-118291857064
19. Lucas J. C. Andrade et al "A Methodology to Evaluate the Security of Block Ciphers Against Neurocryptanalytic Attacks" Conference paper 13 July 2024 pp 117–127 https://link.springer.com/chapter/10.1007/978-3-031-64650-8_11
20. Aron Gohr "Improving Attacks on Round-Reduced Speck32/64 using Deep Learning" IACR 2019/037
21. Jung-Wei Chou, Shou-De Lin, and Chen-Mou Cheng. On the effectiveness of using state-of-the-art machine learning techniques to launch cryptographic distinguish- ing attacks. In Proceedings of the 5th ACM workshop on Security and artificial intelligence, pages 105–110. ACM, 2012.
22. AidanN.Gomez,SicongHuang,IvanZhang,BryanM.Li,MuhammadOsama,and Lukasz Kaiser. Unsupervised cipher cracking using discrete GANs. In International Conference on Learning Representations, 2018.
23. Alexander Klimov, Anton Mityagin, and Adi Shamir. Analysis of neural cryptog- raphy. In International Conference on the Theory and Application of Cryptology and Information Security, pages 288–298. Springer, 2002.
24. Elena Laskari, Gerasimos Meletiou, Yannis Stamatiou, and Michael Vrahatis. Cryptography and cryptanalysis through computational intelligence. In Computational Intelligence in Information Assurance and Security, pages 1–49. Springer, 2007.
25. Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. *In* Proceedings of the 35th Annual Symposium on Foundations of Computer Science *(FOCS*), pp. 124–134. IEEE.
26. Udrescu SM, Tegmark M. AI Feynman: "A physics-inspired method for symbolic regression." Sci Adv. 2020 Apr 15;6(16):
27. Harding, Dresdale, Hearn, 2026 "Loki's Shield", The Exponential Academy, Delaware, USA`
28. Samid, 2025 "Negotiating Darwin's Barrier: Evolution Limits Our View of Reality, AI Breaks Through". Applied Physics Research, Vol 17 No 2.
29. Samid, "AI Accelerated Brute Force Cryptanalysis" https://arxiv.org/abs/2605.08690