A Survey of Entropy Failure as an Attack Surface: Artificial Intelligence and Modern Cryptanalysis

Albert Carlson[*], **Robert Carlson**[†], **Hans C. Mumm**[‡]

Peer Reviewer: **Keeper L. Sharkey**[§]

[*]COBET, National University, San Diego, CA, USA. [†]National University, San Diego, CA, USA.
[‡]Quantum Security Alliance
[§]ODE, L3C

Email: [*]acarlson2@nu.edu, [†]r.carlson1950@student.nu.edu, [‡]hans.mumm@gmail.com [§]sharkey@odestar.com

I. Introduction

Artificial Intelligence (AI) is currently a popular field of study in academia. Industry is also taking notice of technological advances with AI and applying these to a range of use cases. One of the important use case now being addressed by AI is cybersecurity, specifically encryption. Encryption is universally used to protect the secret information of governments, companies of all sectors, and individuals. It is so essential that governments are now investing tens of billions of dollars to read the messages of rival countries using quantum computers [1] and are expending massive amounts of money to record as many messages of their targets in an attack called "harvest now, decrypt later" (HNDL) [2]. While quantum computers are projected to become available in the future [3], AI can enable advanced decryption techniques today.

AI has been used in many forms since Turing employed early, simpler versions than are available today to break the Enigma cipher daily in World War II. Various other techniques were used decades before Artificial Neural Networks (ANNs) and their variants, as well as Large Language Models (LLMs), to break ciphers. Modern AI has come of age and will be further advanced by quantum computers (QCs).

The key question at this time is whether encryption can remain secure at the commercial, diplomatic, and governmental levels, given AI's capabilities. In answer to this question, AI is capable of breaking historical ciphers, seriously degrading ciphers considered "hard," and of breaking "novel" (yet to be designed) ciphers. Multiple examples of this capability exist.

This demonstrable finding raises the question of how to protect sensitive messages from AI attacks. The answer to this question begins with recognizing that AI-based decryption is a systemic threat now and will only intensify in the future. Users must increase the entropy of ciphers and apply that entropy [4] in a manner that maximizes the advantage of randomness for the legitimate user(s) and minimizes attackers' ability to leverage patterns in cipher text (CT) to ensure the secrecy of their messages.

II. Background on Cryptography

A. Randomness and Entropy

Randomness is the property of being unpredictable. This property applies to bits, bytes, and characters. The manifestation of randomness is the mathematics that says that if an observer with no a priori knowledge is asked to guess the next bit in a sequence, the probability of correctly guessing the bit (g), as the number of guesses (n) approaches infinity, will be

$$g = .5 \tag{1}$$

This definition also implies that, across the full sample of bits, the average number of 0 or 1 bits will be half the sample size. Full randomness allows for numbers and "short" sequences of characters to occur during a run of data points, unlike pseudo-randomness, where sequences will repeat and cycle with a cycle length.

Pseudo-randomness is often used in place of true randomness. True randomness cannot be calculated [5] because knowing the formula for the next number in a sequence renders it perfectly predictable. Even if the sequence source is unknown, it takes knowledge of relatively few numbers in the sequence to reconstruct the formula for a pseudo-random number generator (PRNG). The number of instances required to formulate a pseudo-random number is denoted by ρ and can be as small as 3. Even the the widely used Mersenne Twister developed in 1997 by Makoto Matsumoto and Takuji Nishimura can be broken in as few as 624 instances [6]. The strength of the PRNGs varies widely across the spectrum. The National Institute of Standards and Technology (NIST) provides testing at the bit level using SP 800-22, a statistical test suite [7], and recommendations for good bit-level routines in SP 800-90B, which is a recommendation that specifies the design principles, requirements for, and the tests for the validation of the entropy sources used by random bit generators [8].

Entropy is described next, however to point out, there is no way to definitively show that any sequence is truly random. Knuth noted that an observer could only verify randomness for a monitored sequence of size $2N - 1$ symbols [9], N is

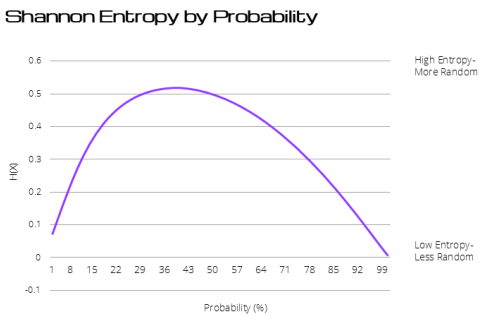

Fig. 1. Entropy vs Probability

the number of characters in the random sequence. However, mathematics shows that some functions are truly random. These functions may be used to generate true randomness, but they must be securely delivered to be usable.

In encryption operations, randomness for a discrete random variable (x) is manifested by Shannon entropy ($H(x)$) [10]:

$$H(x) = -\sum_{i=1}^{n} P(x_i) log\big(P(x_i)\big) \tag{2}$$

where $P(x_i)$ is the probability of a discrete random variable (x_i). Figure 1 plots Shannon entropy verses probability. Entropy, rather than complexity, is key to keeping information safe during encryption. Shannon used Hartley's equation (C) [11] for calculating the surprise of information, or entropy which is akin to counting the highest possible number of distinguishable values for a given amplitude (A) and precision $\pm\Delta$:

$$C = \log(1 + A/\Delta) \tag{3}$$

Entropy is a key descriptor of the secrecy in ciphers.

B. Ciphers

Ciphers [or cipher texts (CT)] are hashes and encoding functions that obscure the input [or plain text (PT)] message so that it is difficult to read and understand if the reader is not a legitimate party to the message. Modern encryption functions satisfy the 1:1 assumption, mapping each PT symbol to exactly one CT symbol. The exact mapping depends on the secret key/seed selected (Shannon, 1949; Kerckhoffs, 1883). Normally, this seed is "randomly" selected and depends on the selection algorithm's entropy, and its legitimacy is verified by knowing the secret key. The chance of "guessing" a string of length L of the right characters in a row (P_L) is

$$P_L(CT) = \frac{1}{\alpha^n} \tag{4}$$

where α is the size of the alphabet (which is the set of all possible symbols of letters, numbers, etc.) which is strictly positive. Therefore, as the size of $L \to \infty$, the $P_L(CT) \to 0$.

Recently, it has been shown that ciphers can be modeled as a single type, the substitution (S) cipher. Through a process known as "isomorphic cipher reduction" [12], [13], all non-randomized ciphers can be replaced by an equivalent S cipher with the appropriate key. In response to this reduction, cipher designers created functions that can surround the cipher to add entropy, known as "cryptographic modes" [14]. However, these modes are susceptible to side-channel attacks [15] that directly reveal the PT and are therefore ineffective for keeping a message safe [16]–[18].

C. Breaking Ciphers

To select the most effective methods for protecting encrypted information, it is essential first to understand what breaking a cipher entails and the most successful attacks on such information. Breaking a cipher means the attack reduces the effort required, usually by eliminating the number of keys below what is needed for a brute-force attack [19]. This does not necessarily mean that the full plain text is returned, although that is the goal of decryption. Further, this reduction in time and effort should be below the point where the key space can be computationally brute forced in a "reasonable" time. The break is often made using heuristic algorithms [20], a method that enables fast, informed decision-making. The conventional view of a cipher break is to return the original encryption key. However, the goal of decryption is to return the readable original message. Cipher breaks can be made using a variety of methods, including hardware, software, and hybrid heuristic algorithms. It does not have to be a single approach; it can be a combination of steps, which include AI or crypto-breaking algorithms.

There is only one method that is guaranteed to break a cipher – the Brute Force Attack [14], [20]. In this attack, the full key space of the cipher is enumerated, and each key is tested sequentially. Ideally, a key is selected at random from the key space and, if that key does not prove to be the correct key, it is discarded. This procedure is repeated until the correct key is found. For this reason, cryptographers aim to create keys with the largest possible key space and to make the time required to check the encryption as large as possible.

Information has a practical value for a given time [21]. Information that is protected beyond that time has no practical value to the attacker and is likely to have been revealed by other means. Supporting this observation is the general practice of setting declassification dates for sensitive, secret information [22]. The idea of this approach is to make it so time-consuming and resource-intensive that breaking the cipher becomes mathematically infeasible in terms of cost, and the time expended would render the information no longer valuable to the attacker.

Characterizing the Brute Force Attack in terms of the average time required to solve the decryption ($t_{t,avg}$) depends on the key space of cipher ($|K|$) and the time required to apply a selected test key and determine if it results in a decrypted readable message (t_{try}). Therefore, the average time to brute force a cipher is,

$$t_{t,avg} = \frac{|K| t_{try}}{2} \tag{5}$$

Ideally, the average time to break a cipher using the Brute Force attack ($t_{t,avg}$) is maximized and exceeds the lifetime of

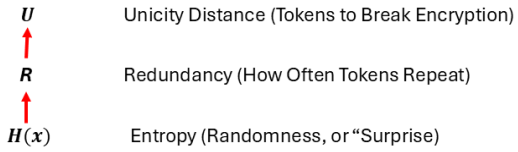

Fig. 2. Relationship of Shannon Measures

a user. The best ciphers have an average break time longer than the age of the universe. In this security approach, it is important to accurately characterize the size of the key space to ensure the safety of the information [23].

The comparison of security (S) offered by different ciphers [24] arises from a comparison of the unicity distance (U) of the two ciphers [10], where U is defined by:

$$U = \frac{H(x)}{R} \tag{6}$$

such that R is the PT redundancy in bits per character. Note that U can be defined in terms of redundancy

$$R = 1 - \frac{H(x)}{H_{max}(x)} \tag{7}$$

which in turn is directly dependent on the entropy of the message and language. With these definition of U and R, S is written as:

$$S = \frac{U_1}{U_2} = \frac{\frac{log|K_1|}{R_\lambda log|\alpha|}}{\frac{log|K_2|}{R_\lambda log|\alpha|}} = \frac{log|K_1|}{log|K_2|} \propto \frac{|K_1|}{|K_2|} \tag{8}$$

K is the key space (which is the set of all possible keys that can be used with a specific encryption algorithm). Large-key-space ciphers only work if the secret key is changed for each message, or, better yet, is used only for as long as the information accumulated in a portion of the message remains below the message's unicity distance. Historically, and scientifically, the best way to protect any information obscured in a cipher is to maximize the entropy (see Figure 2, the relationship of Shannon Measures, and Figure 3, the relationship of Entropy to Security).

Fig. 3. Relationship of Entropy to Security

Building on the ideas of Shannon information accumulation and entropy for message protection, the Vernam cipher [25], also known as the One Time Pad (OTP) [26], and polymorphic ciphers were developed. The OTP lets the least amount of information accumulate in a cipher, but is heavily dependent on entropy [27] to prevent the Venona attack. The Venona attached showed that Soviet reuse of one-time pad keys let U.S. analysts led by Meredith Gardner use early computing and cryptanalysis to decrypt messages, exposing hundreds of spies—including atomic spies at Los Alamos National Lab like Klaus Fuchs and networks tied to Julius Rosenberg—and proving that human key-management errors can defeat even mathematically secure encryption [26], [28]. Polymorphic ciphers [24] are an enhancement of the OTP with similar performance with fewer resources, but are equally dependent on entropy for selecting cipher and key pairs.

D. Tokenization and the Link to Entropy

Entropy depends on the configuration of data being evaluated. For example, bit-sized data has a different entropy than word-sized chunks of a message. Recognizing this difference, it is important to specify the data size. Tokenization is the process of breaking information into smaller units, called "tokens," that a system can understand and process. When a person reads a sentence, their brain naturally breaks it into words. Computers do something similar, but they must be told precisely how to divide information. A token might be a word, part of a word, a single letter, a number, or even a byte of data. The way the user chooses to break information into tokens has a significant impact on how machines interpret and process it. Understanding the relationship between tokens and information is essential to modern AI, cybersecurity, and information science, as small differences in tokenization can have significant consequences for intelligence, efficiency, and security.

Recalling that entropy is a measure of uncertainty or randomness; in the simplest terms, it answers the question: "How predictable is this information?" If something is very predictable, it has low entropy. If it is highly unpredictable, it has high entropy. This is true for information theory and entropy applied to physical systems as well (like chemical systems).

A key concept is that tokenization and entropy are inextricably connected: tokenization alters what the system perceives as "symbols," and entropy quantifies how unpredictable those symbols are. Changing how computers break data into tokens alters the statistical patterns in the data and, therefore, its entropy. The same sentence can have very different entropic values depending on whether it is broken into words, letters, or smaller pieces.

For instance, when tokenizing text by words, some words like "the" or "and" appear very often. That makes the data

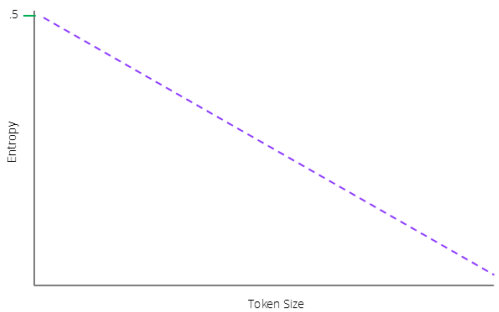

Fig. 4. Entropy vs Token Size

more predictable and lowers entropy. However, when tokenizing the exact text into individual letters, the symbol distribution becomes more balanced and less predictable, increasing entropy. Using subword tokens (pieces of words), the results are something in between using full words and using letters. So even though the text's underlying meaning has not changed, the measured randomness does change because the token structure has changed.

Tokenization has material consequence in many fields. In artificial intelligence, language models learn by predicting the next token. Good tokenization makes patterns easier to understand and reduces unnecessary complexity. Poor tokenization can make data appear more random than it really is, which makes learning harder and less efficient. In data compression, tokenization affects the amount of redundancy that can be removed. If tokens capture meaningful patterns, entropy goes down and compression improves. In cybersecurity and cryptography, entropy is critical because secure systems depend on unpredictable values. If data is poorly tokenized or encoded, hidden patterns can reduce entropy and weaken security.

At a deeper level, tokenization can change structure, whereas entropy concerns a measurement (see Figure 4, entropy verses token size). Tokenization decides how information is organized into units. Entropy expresses how much uncertainty exists within those units. Together, these determine how information behaves in AI systems, how secure cryptographic systems are, and how efficiently data can be processed. Tokenization and entropy are inseparable concepts in modern computing systems. Together, these shape the behavior of AI systems, the security of cryptographic mechanisms, and the efficiency of data processing.

III. Artificial Intelligence

Artificial Intelligence has its foundational mathematics roots in stochastic, Bayesian-based problems developed and simulated in the late 18^{th} through 19^{th} century. After proving its effectiveness on those problems, AI was applied to linear data using machine learning (ML). Large amounts of data can train neural networks (NNs) and help them identify patterns. Towards the beginning of the 21st century, AI gained the ability to handle nonlinear data and classify it correctly. Lately, AI has been used to gather and classify data, as well as to manipulate it to solve more general problems.

Bayes' theorem calculates the probability (P)of an event (X) based on prior knowledge of related conditions or new evidence (Y):

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{9}$$

where $P(X|Y)$ is the posterior probability of event X happening given Y, $P(Y|X)$ is the likelihood probability of evidence Y occurring given that X event is true, $P(X)$ is the initial probability of event X before any knowledge of condition Y, and $P(Y)$ is the probability of the evidence Y without any knowledge of the event X taking place. Bayes' theorem is also known as conditional probability which updates, or revises, beliefs by incorporating new data, making it foundational for reasoning under uncertainty. Key applications include machine learning, medical diagnostics (false positives), and spam filtering. One concept to note, is that if the data that is supplied to the algorithm, then there is back-in bias to the result.

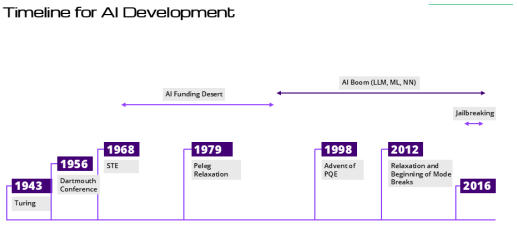

Fig. 5. Timeline for AI Development

Throughout the development of AI (see Figure 5, timeline for AI development), the concept as taken on dual definitions. The more restrictive goal is to create machines that mimic human abilities, often exceeding human performance in certain tasks. Building on the first goal, the second concerns creating a self-aware consciousness that is "alive," self-directed, and capable of generating novel thoughts. The former definition is already achieved and is being both improved and exploited. Through Bayes' theorem alone, self-awareness, self-direction, and independent thought may never be possible or even desirable. However, the use of AI to extend machine capabilities and augment human resources and population capabilities is underway.

AI is not new or experimental; it has been ongoing for decades. Done with a fraction of the computational power now being brought to bear on the problem. This makes it even more effective in the present day with the resources available. The thing that makes the AI revelation of the 21st century significant is the amount of data we are able to process with data centers, high-performance computing (HPC), and enhanced massively parallel compute using Message Passing Interface (MPI). To point out, AI is energy intensive and in 2015 the Department of Energy (DOE) announced the Genesis Mission [29]. The Genesis Mission is a U.S. federal initiative to build the world's most powerful integrated scientific platform, using artificial intelligence and advanced computing to transform how research and discovery are conducted. Its goal is to connect the nation's leading supercomputers, experimental facilities, AI systems, and vast scientific datasets to double the productivity and impact of American research within the next decade. By enabling AI-driven simulation, automated experimentation, hypothesis testing, and predictive modeling, the mission aims to accelerate breakthroughs in energy technologies, discovery science, national security, and advanced materials. It brings together national laboratories, universities, and industry partners to harness federal scientific data and computing resources in pursuit of challenges once considered too complex to solve.

Can Ciphers be broken using AI?: In the last decade, there has been increased activity, research, and confidence in AI for many applications. However, AI has been used for breaking ciphers since at least World War II. Categories of cryptologic breaks are defined int he following table (Table I, categories of cryptologic breaks): AI was formally defined in the 1950s

Type	Characterized as:
Full	Returns complete key or decrypted message
Partial	Returns an incomplete key or message
Reduction	Reduces the number of keys that are valid but no tokens can be read
No	No key mappings are eliminated or readable

TABLE I
CATEGORIES OF CRYPTOLOGIC BREAKS

by John McCarthy, but in retrospect, it can be demonstrated that the techniques used during World War II were, indeed, AI. As of the first quarter of the 21st century, the definition of AI has evolved from the original definition articulated at the 1956 Dartmouth Workshop on AI [30]. At that meeting, AI was defined to be "the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it." The Dartmouth Workshop was the official beginning of the era of AI, but its roots extend further back.

A. Survey the basics of AI, its structure, and use in cryptography

For the applications of AI to electronic computers and cryptography, the first pivotal event can arguably be when Alan Turing originally asked the question, "Can computers think?" His answer was embodied in the Turing test [31], which said that when an observer, blindly querying a human and a computer, could not tell the difference between the two sources of answers, then AI would be achieved. This view has (incorrectly) led to the modern, biased view of AI. While most people view AI as Large Language Models (LLMs), Machine Learning (ML), Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and related implementations of AI algorithms, the scope of AI is much larger [32]. Section 238(g) of EO 13960 defines AI as,

- "Any artificial system that performs tasks under varying and unpredictable circumstances without significant human oversight, or that can learn from experience and improve performance when exposed to data sets.
- "An artificial system developed in computer software, physical hardware, or other context that solves tasks requiring human-like perception, cognition, planning, learning, communication, or physical action.
- "An artificial system designed to think or act like a human, including cognitive architectures and neural networks.
- "A set of techniques, including machine learning, that is designed to approximate a cognitive task.
- "An artificial system designed to act rationally, including an intelligent software agent or embodied robot that achieves goals using perception, planning, reasoning, learning, communicating, decision-making, and acting." [33]

Although AI can generate novel combinations of information, its outputs are grounded in patterns derived from existing data rather than independently conceived ideas. AI does not originate ideas through lived experience or conscious insight, but instead recombines and extrapolates from prior data. Under the definition of Presidential Executive Order (EO) 13960, which directs U.S. federal agencies to promote and coordinate the use of artificial intelligence in government operations in a manner that is lawful, trustworthy, transparent, and consistent with national values, by establishing common principles and policy guidance for AI design, development, acquisition, and use, with respect to cryptography, AI has been relevant and used in the analysis of the Enigma cipher. The Enigma cipher was a method of encrypting messages using the Enigma machine (see Figure 6, military model Enigma I circa 1930), an electro-mechanical device most famously used by Nazi Germany during World War II.

Turing and the staff at Bletchley Park created specialized hardware, known as the "Bombe" [34]. This electromechanical device helped discover some of the settings of the Enigma machine. Later, the Colossus computer [35] was adapted to quickly eliminate different keys. In effect, the Colossus accelerated a brute-force attack on the cipher.

After World War II, Shannon showed how language statistics could be used in heuristic attacks [10]. Peleg and Rosenfeld used this information to create a technique known as "relaxation" [36] in 1979. This algorithm is applied to blocks of encrypted symbols in an S cipher and attempts to match CT blocks to PT blocks based on block frequency. Often, the mappings are not all correct. By analyzing the errors in the decrypted text, new mappings are selected, and the process proceeds iteratively. The guiding principle is to minimize the total error in the readable decryption for the mappings. Each iteration provides more information for mappings. In effect, this is equivalent to unsupervised training of a neural network

Fig. 6. Military Model Enigma I circa 1930

using ML.

Carlson used a variation of relaxation in attacking ciphers [24] that allowed for patterns not found in the normal lexicon of a language to be correctly decoded even when they violate normal language statistics. This is very useful when encountering foreign names, places, and words. Again, this is similar to the ML/AI training.

Another advance in non-neural network AI in cryptography is the use of Set Theoretic Estimation (STE). First suggested in the late 1960's by Witsenhausen and Schweppe [37], [38], this technique starts by considering all solutions, known as "estimates," and eliminates all solutions (s) shown to be incorrect, or "impossible." Estimates are categorized by the output properties they exhibit when applied as inputs. These groupings are known as "property sets" (ϕ_i). Since an estimate must show all required properties to be a correct solution, intersecting the property sets yields the possible solutions for the problem's particular inputs. That is, the solution must be a member of all of ϕ_i:

$$s \in \bigcap_{i=1}^{n} \phi_i \tag{10}$$

One way to visualize this algorithm is to project the estimates into a 3-dimensional (3D) space. Estimates are arranged by using a "distance function" [39], where the distance (d)between to points (a, b) in 3D space (x, y, x) is:

$$d < a, b >= \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} \tag{11}$$

also known as the Euclidean distance.

When all estimates are placed in the space, different property sets are formed, and the boundaries of the estimates within these sets are used for intersection calculations. As each is applied, the new solution set is formed by intersecting the old solution set with the property set. Since the solution sets are monotonically decreasing in size, they eventually converge to the desired solution or equivalent solutions for the problem.

Dr. Albert Carlson showed that STE could be used to fully break ciphers and recover their original PT. In his dissertation [24], he accurately recovered over 90% of 80,000+ encrypted files. A range of common cipher structures, including substitution (T), permutation (M), substitution–permutation (TM), permutation–substitution (MT), and their multi-round combinations (i.e., TMT, MTM, etc.), across several block sizes, was investigated. NNs can be similarly modeled to STE constructs where the "estimates" are projected into a hyperspace and separated by a sphere sized to a diameter (S):

$$D = \frac{\text{confidence interval}}{2} \tag{12}$$

of NNs are trained to learn a boundary between points on the edge of belonging to a set and those that do not. STE is then seen as a precursor to the NN and works in the same manner. It is a form of AI NN functions. Carlson further suggested moving the process [24] into topological space [40] in 2006. This change enabled more efficient application of set operations and was straightforward to implement. STE also shows the probability of success for a solution by using a measure applicable to AI – the probability of an estimate existing in a property set.

This measure is well known as the Asymptotic Equipartition Property (AEP) [41]. The AEP gives the probability that a particular estimate is a valid solution to a given problem. As such the probability of success for a solution is:

$$\begin{aligned} P(x_i) \in \{s\} &= \frac{1}{P(x_i \in \phi_1)} \frac{1}{P(x_i \in \phi_2)} \cdots \frac{1}{P(x_i \in \phi_n)} \\ &= \prod_{i=1}^{n} \frac{1}{P(x_i \in \phi_i)} \end{aligned} \tag{13}$$

Problems that are characterized by the same mathematics are versions of the same base problem and can be solved using the same approaches (Fraleigh,2003). This suggests that STE and AI are part of the same family of solutions. Another example of this is probability of chemical hypotheses using new experimental data, linking theoretical models with observations and enabling tasks such as parameter estimation, molecular modeling, spectroscopy analysis, and reaction optimization under uncertainty. However, it should be explicitly mentioned, that using these probabilities for chemistry is not an application of quantum probabilities and does not represent exact physics of the system, rather its use is an extrapolation of experimental data to improve our understanding of the chemical process.

Now, back to cyber; an attempt to sidestep heuristic attacks on ciphers has led to attacks on randomization routines intended to defeat language-statistical approaches, known as "modes" [14]. When combined with the Advanced Encryption Standard (AES), this encryption method is the current security standard (NIST, 2001). A recent collection of side-channel trace exploitation (STE) attacks targets ciphers operating in multiple block cipher modes—Electronic Codebook (ECB), Cipher Block Chaining (CBC), Propagating Cipher Block Chaining (PCBC), Counter Mode (CTR), Output Feedback (OFB), and Cipher Feedback (CFB)—all of which have now been broken using STE [16], [42]. Each of these papers contains an example of the full STE-based break of at least one of the modes.

The preceding technology review shows that AI has been successfully used in attacking cryptography for decades (see Figur 7, mode break timeline). More familiar forms of AI, such as NNs and LLMs, merely continue the attacks. These algorithms are currently accepted in academic and popular literature as breaking historical ciphers, including the S and P ciphers and certain block ciphers. However, the fact that all

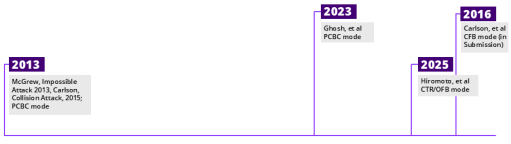

Fig. 7. Mode Break Timeline

ciphers reduce to the S cipher indicates that this approach is successful, using the following techniques, all of which are related to, or are AI-based, for breaking all ciphers. Simple one-byte S ciphers are already automatically solved online [43], [44].

In addition to the history of attacks, at present NNs, LLMs, and ML [45], [46] are regularly used to break ciphers. The common view is that AI cannot break new, novel encryptions. This is not the case. Hackers are not using cryptography to bypass the controls used on AI chatbots and assistants. Input filters on the agents are programmed to reject instructions designed to get the agent to reveal data, such as how to create bombs. In response, the hackers developed an attack vector known as "jail-breaking" [47]–[49]. The goal of the attack is to conceal the actual request by encrypting it, then instructing the agent to decrypt the message and follow its instructions. Using known ciphers will not work, since the filter can be easily programmed to try known ciphers and reject the decrypted message. Therefore, the attacker uses novel ciphers, which the agent solves. The success of the attacks is evidence that AI can, and does, break even novel ciphers.

AI algorithms excel at taking a corpus of data and making decisions when no algorithm is defined to separate choices. STE and cipher reduction [43] share this same characteristic with AI. The characteristic indicates that the functions can be identical via abstract algebra [50] and topology [40]. While AI algorithms may not return the full decryption, AI can perform many functions that relieve cryptographers of tedious work, enabling them to interpret data and complete the decryption. The AI process can reduce the number of possible solutions, enabling a brute-force search to be applied efficiently and quickly to the AI's output.

AI can also easily and efficiently follow human-developed attacks. This does not denigrate the role of AI but rather points to the need for humans to design the attacks that AI runs. While AI can develop new algorithms for a problem, it cannot devise a completely new approach to attacks. Once an approach or a new attack vector/attack point is identified, the AI can take over.

The next step in encryption issues, especially with AI, is quantum encryption (QE). Quantum computers (QCs) have been around since 1990s and where theoretically defined in the 1960s. Programs written on those quantum machines ushered in the post-quantum environment (PQE). Most people do not understand the differences between the capabilities of machines in quantum and classical environments. A good starting place to learn about quantum concepts is the following books by Dr. Keeper L. Sharkey, "Quantum Chemistry and Computing for the Curious: Illustrated with Python and Qiskit Code" [51] and "Advanced Technologies for Humanity (Chapter 10: Look Before you Leap: Demystifying Quantum Computing Enigmatic Frontier)" [52].

There are some important similarities between the two. First, the principles of math apply equally to both. mathematical theory is the same in both environments. The math itself never changes; only the way it's implemented changes. In fact, quantum mechanics is a set of mathematical postulates that describe the nature of particles including chemistry and its interaction with light. "Quantum" is not nature; it's math. However, quantum computers are using very specific forms of nature such that the measurements made for computing might be described by quantum mechanics (if the math is not too difficult to write).

Math is seen in encryption related to PQE, which is built on functions, such as prime and semiprime factorization [53] algorithms, the discrete log problem [54], and Euler's Totient function [55] (another equation using classical probabilities), all of which are not breakable on classical computers, but are known to be breakable on QCs. Symmetric key algorithms now in use in the conventional computing environment are more resistant to PQE breaks. To overcome the speed increase from conventional to Quantum computers (known to be an increase of $\sqrt{2}$) [56], a key length increase of a factor of 2 is typically applied for safety. As a result, not all ciphers can make a safe transition to the PQE.

NIST recognized this problem and initiated a search for new quantum-safe ciphers, known as Quantum Proof Algorithms (QPAs). Four such algorithms were selected [57]. These algorithms may be insufficient for security [58]. Without a library of safe, secure encryption algorithms, only symmetric encryption is available in the classical environment. If the same algorithms are used in both the classical and quantum environments, they are equally susceptible to AI. Any AI programs designed for and used on QCs will be faster and at least as effective. It should also be noted that QCs are controlled by classical machines, making all QCs susceptible to classical attacks; therefore, AI will be effective in both environments.

IV. Conclusions

AI is already here. Modern AI was defined in the mid-1950s and is already being used to break encryption. Historically, electronic computers and circuitry have been used since the 1940s with increasingly widespread and effective results. Improvements in computing hardware, software capabilities, and AI algorithms have led to effective encryption breaks. This trend toward greater effectiveness and increased speed will continue to accelerate.

QCs are in the same class as AI interms of its ability to affect encryption. Quantum hardware promises more secure networks and machines, but also introduces additional vulnerabilities and attack vectors. However, one of the most important facts about quantum machines is seldom mentioned, and is almost invariably ignored: a quantum computer (QC) is a hybrid computing environment in which the QC is controlled and programmed, and communication takes place via classical computers. Succinctly stated, QCs encompass all new vulnerabilities, as well as those of classical computers and of the communication and networking control planes. Adding quantum AI (QAI) programs only increases the AI attack capabilities, enabling faster attacks on any encryption and

obscured messages used within the QC. Although quantum-safe ciphers will eventually be developed and certified as safe, the effort will entail high costs in terms of key space, memory, and resources. From that point on, the arms race for breaking those Quantum Proof Algorithms will take place much as they do now in the conventional domain.

The question of how to keep encrypted messages and information safe from AI-based advances that break encryption, now occurring in both the conventional and quantum domains, is more important than ever. What does it take to frustrate AI? Assuming that a brute-force attack is not a viable option, breaking a cipher requires gathering hints of patterns and statistical information revealed by the encryption algorithm in the CT data stream. Since all ciphers reduce to the S cipher, and S ciphers leak patterns, it all comes down to leaving no clues. That means that no patterns are available to apply to heuristic attacks on that data. Without patterns and clues, all data revealed by the CT stream should come as a surprise to the attacker. Only random, uninformed guesses are possible.

To achieve the fewest patterns in the encryption, a morphing cipher, such as a polymorphic cipher, must be used. Selections of the shards used in the polymorphic cipher must also be random to be safe [25], [27]. For the selection to be random, it must use a randomness source as close to true randomness as possible, while still being either deterministic or allowing for shared entropy. Entropy is needed now due to the deployment of encryption, and probability-based functions are being affected by attacks, especially by AI-based attacks. Until this happens, AI will undermine encryption and, consequently, security.

The safest encryption has the largest unicity distance. Since unicity distance is directly dependent on entropy, for any given cipher and language, the best way to increase U is to keep the entropy as high as possible. When $U = 0.5$, no patterns will exist, and the data stream will be uniformly distributed. The closer the entropy comes to 0.5, the better. Until the best-in-field entropy is employed, vulnerabilities will remain that can be exploited in the decryption process.

REFERENCES

[1] S. Duranton. Quantum computing takes off with $55 billion in global investments, https://www.forbes.com/sites/sylvainduranton/2024/06/26/quantum-now/. *Forbes*.

[2] Keyfactor. Harvest now, decrypt later: A new form of attack, https://www.keyfactor.com/blog/harvest-now-decrypt-later- a-new-form-of-attack/.

[3] S. Krastanov, M. Heuck, J. Shapiro, and P. Englund. Room-temperature photonic logical qubits via second-order nonlinearities,. *Nature Communications*, 12(191).

[4] Frontier Technologies Laboratory . Cryptographic resilience in the ai quantum age: A predictive indexing approach to entropy assessment,.

[5] Rod Downey and Denis R. Hirschfeldt. Algorithmic randomness. *Communications of the ACM*, 62(5):70 – 80, 2019.

[6] Melissa E. O'Neill. Pcg: A family of simple fast space-efficient statistically good algorithms for random number generation. Technical Report HMC-CS-2014-0905, Harvey Mudd College, Claremont, CA, 9 2014.

[7] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. A statistical test suite forrandom and pseudorandomnumber generators forcryptographic applications. Technical Report SP 800-22, National Institute of Standards and Technology (NIST).

[8] NIST. National institute of standards and technology, tehcnical report sp 800-90b, https://csrc.nist.gov/pubs/sp/800/90/b/final.

[9] Donald Knuth. *The Art of Computer Programming Volume*, volume 2 Seminumerical Algorithms. Addison-Wesley, 3^{rd} edition, 1998.

[10] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656 – 715, 1949.

[11] Paul Garrett. *The Mathematics of Coding Theory*. Pearson/Prentice Hall, Upper Saddle River, 2004.

[12] Albert Carlson, Bhaskar Ghosh, Indira Dutta, Shivanjali Khare, and Michael Totaro. Keyspace reduction using isomorphs. *12th Annual International Conference and Workshop on Computing and Communication (IEMCON) 2021, Vancouver, Canada*.

[13] Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15 – 20, 1973.

[14] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons Inc., New York, 2nd edition, 1996.

[15] Harald Søndergaard and Peter Sestoft. Referential transparency, definiteness and unfoldability. *Acta Informatica*, 27(6):505 – 517.

[16] Albert Carlson, Bhaskar Ghosh, and India K. Dutta. Using the collision attack for breaking cryptographic modes. *13th International Congress on Computing, Communication, and Networking Technologies 2022, Kharagpur, India*.

[17] Bhaskar Ghosh, Albert Carlson, and Indira Dutta. A demonstrable break of pcbc mode. *International Symposium on Networks, Computers and Communications (ISNCC): Trust, Security and Privacy (ISNCC 2023), Dohar, Qatar*.

[18] Robert E. Hiromoto, Albert Carlson, and Mandeep Singh. Breaking the counter (ctr) mode. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*.

[19] B. Schneier. A self-study course in block-cipher cryptanalysis. *Cryptologia*, 24(1):18 – 34.

[20] N. Nalini and G. Raghavendra Rao. Attacks of simple block ciphers via efficient heuristics. *Information Sciences*, 177(12):2553–2569, 2007.

[21] Albert Carlson, Benjamin Williams, Sai Ranganath Mikkilineni, and Mandeep Singh. The value of information. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC), LasVegas, NV, 6 − 8 January*, 2025.

[22] D. Elliot Bell and Leonard J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report 2547, MITRE, 1 March 1973.

[23] Albert Carlson, Torsten Gang, Garrett Gang, Bhaskar Ghosh, and Indira Dutta. Evaluating true cryptographic key spacesize. *Ubiquitous Computing, Electronics, & Mobile Communication Conference (UEMCON 2021)*, 1 - 4 December 2021.

[24] Albert Carlson. *Set Theoretic Estimation Applied to the Information Content of Ciphers and Decryption*. PhD thesis, University of Idaho, 2012.

[25] John Earl Haynes and Harvey Klehr. *Venona: Decoding Soviet Espionage in the United States (Yale Nota Bene)*. Yale University Press, 1999.

[26] Uli Maurer. A universal test for random bit generators. *Journal of Cryptography*, 5(2):89–105, 1992.

[27] J. Kelsey, B. Schneier, D. Wagner, and C. Hall. Cryptanalytic attacks on pseudorandom number generators. *Fast Software Encryption, Fifth International Workshop Proceedings*, pages 168 – 188, 1998.

[28] Albert H. Carlson, Sai Ranganath Mikkilineni, Michael Totaro, and Christopher Briscoe. A venona style attack to determine block size, language, and attacking ciphers. *International Symposium on Networks, Computers, and Communications, (ISNCC 2022)*, 2022.

[29] National Design Studio. Genesis mission: A national mission to accelerate science through artificial intelligence, https://genesis.energy.gov/.

[30] Dartmouth University. Keywords for ai literacy | writing program, https://writing.dartmouth.edu/teaching/ai-literacy/keywords-ai-literacy.

[31] A. Turing. Computing machinery and intelligence. *Mind*, 49:433 – 460.

[32] B. Mehling. *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers*. Cambridge University Press, 2021.

[33] National Archives. Promoting the use of trustworthy artificial intelligence in the federal government, https://www.federalregister.gov/documents/2020/12/08/2020-

27065/promoting-the-use-of-trustworthy-artificial-intelligence-in-the-federal-government.

[34] G. Welchman. *The Hut Six story: Breaking the Enigma codes*. MM Baldwin.

[35] B. J. Copeland. *Colossus: The secrets of Bletchley Park's code-breaking computers*. Oxford University Press.

[36] Shmuel Peleg and Azriel Rosenfeld. Breaking a substitution cipher using a relaxation algorithm. *Communications of the ACM*, 22:598 – 605, 1979.

[37] Fred Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, AC-13(1):22 – 28, 1968.

[38] Hans Witsenhausen. Sets of possible states of linear systems given perturbed observation. *IEEE Transactions on Automatic Control*, AC-13(1):556 – 558, 1968.

[39] Patrick Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182 – 208, 1993.

[40] John Kelley. *General Topology*. D. Van Nostrand Company, Princeton, 1955.

[41] Thomas Cover and Joy Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc, New York, 2nd edition, 2005.

[42] Robert E. Hiromoto, Albert Carlson, and Mandeep Singh. Breaking the counter (ctr) mode. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*.

[43] Bhaskar Ghosh, Indira Dutta, Shivanjali Khare, Albert Carlson, and Michael Totaro. Isomorphic cipher reduction. *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2021*, pages 947–953.

[44] R. Kübler. Where machine learning meets cryptography, solving the cryptographically-relevant learning parity with noise problem via machine learning, https://towardsdatascience.com/where-machine-learning-meets-cryptography-b4a23ef54c9e/.

[45] A. Gohr. Improving attacks on round-reduced speck32/64 using deep learning. *Advances in Cryptology – CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA,August 18–22, 2019, Proceedings, Part II*, page 150 – 179.

[46] I. Dinur, O. Dunkelman, N. Keller, E. Ronen, and A. Shamir. Efficient detection of high probability statistical properties of cryptosystems via surrogate differentiation. *Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, page 98 – 127.

[47] P. Hall. Cryptographers show that ai protections will always have holes, https://www.quantamagazine.org/cryptographers-show-that-ai-protections-will-always-have-holes-20251210/. *Quantamagazine*, 2025.

[48] D. Handa, Z. Zhang, A. Saeidi, S. Kumbhar, M. N. Uddin, RRV Aswin, and C. Baral. When 'competency' in reasoning opens the door to vulnerability: Jailbreaking llms via novel complex ciphers, https://arxivorg/abs/2402.10601. *arXiv*.

[49] S. Harris, J. Hadi, and U. Zukaib. Cryptography: Against ai and qai odds, https://arxiv.org/abs/2309.07022. *arXiv*, 2023.

[50] John B. Fraleigh. *A First Course in Abstract Algebra*. Addison-Wesley, 7^{th} edition, 2003.

[51] Keeper Sharkey and Alain Chance. *Quantum Chemistry and Computing for the Curious: Illustrated with Python and Qiskit® code*. Packt Publishing.

[52] Randall Nichols, Patricia Ackerman, Doug DeMaio, Brent Knaple, Haley Larson, Robert McCreight, Hans Mumm, Rajesh Murthy, Keeper Sharkey, and Julie Ryan. *Advanced Technologies for Humanity*. Pressbooks, kindle edition.

[53] A. Overmars and S. Venkatraman. A fast factorisation of semi-primes using sum of squares. *Mathematical and Computational Applications*, 24, 2019.

[54] Alfred Menezes, Paul von Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography*. CRC Press, New York, 1996.

[55] L. Euler, A. Diener, and A. Aycock. Theoremata arithmetica nova methodo demonstrate, https://arxiv.org/abs/1203.1993. *arXiv*, 2012.

[56] Lov K. Grover. Quantum computing: How the weird logic of the subatomic world could make it possible for machines to calculate millions of times faster than they do today. *The Sciences*, pages 24 – 30, 1999.

[57] NIST. Nist announces first four quantum-resistant cryptographic algorithms, https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms.

[58] Albert H. Carlson, Hans C. Mumm, Keeper L. Sharkey, and Merrick Watchorn. Quantum chemistry for detecting cybersecurity threats to information systems. Technical report, Quantum Security Alliance, 2022.