[Albert Carlson, Benjamin Williams et al. (Sep. 27, 2025). Proof of Increased Security of Polymorphism. Pre-publication Release. IEEE.]

Proof of Increased Security of Polymorphism

Albert Carlson[*], **Benjamin Williams**[†], **Sai Ranganath Mikkilineni**[‡], **Christopher Briscoe**[§], and **Mandeep Singh**[¶]

[*]Computer Science Department, National University, San Diego, CA, USA.
[†]Computer Science Department, University of Idaho, Moscow, ID, USA.
[‡]College of Business, Delaware State University, Dover, DE, USA.
[§]Department Chair of Mathematics and Physics, Emet Classical Academy, New York, NY, USA.
[¶]Independent Researcher.

Email: [*]acarlson2@nu.edu, [†]will9847@vandals.uidaho.edu, [‡]SMikkilineni@desu.edu, [§]Cbriscoe@emetclassicalacademy.org, [¶]msingh9001@gmail.com

***Abstract*—Polymorphic ciphers are ciphers that continually change the encryption algorithm and keys used for encryption to reduce the susceptibility of encrypted messages to unintended decryption. This property drew the attention of the research community to these ciphers. Despite the growing interest in polymorphic ciphers, there is no literature effectively demonstrating the strength and speed of these ciphers. In this paper, we utilize cipher properties such as idempotence, mathematical phenomena like the birthday paradox, and cipher characteristics such as the unicity distance and isomorphism to demonstrate that polymorphic ciphers are stronger than single-key ciphers. We present our analysis through four comparative cases, examining the strength of a single-key cipher and a polymorphic cipher based on the encryption shard/block size. The conclusion from our analysis is that polymorphic ciphers are always at least as secure as a single-key cipher. We also present the conditions under which polymorphic ciphers are always more secure than non-polymorphic ciphers.**

***Keywords*—Cryptography, polymorphic encryption, decryption, brute force attack, cipher strength**

I. Introduction

Polymorphic cipher algorithms, also known as "morphing ciphers," have recently garnered the attention of cryptographers [1], [2]. Their researchers claim that polymorphic ciphers offer superior security compared to single-key ciphers. However, critics have pointed to a lack of literature that either proves or disproves this assertion. Although Carlson et al. developed a single key equation for the polymorphic ciphers [3], the examples provided do not demonstrate the equivalence of security between polymorphic ciphers and single key ciphers. This paper presents the conditions under which a polymorphic cipher is guaranteed to be "stronger" (mathematically more secure) than a single-key cipher.

Following this section, we present the background information that is necessary to frame the context of the proofs and provide the required concepts essential to follow the paper's development. In Section III, we present the proofs and mathematical relationships that verify the assertion that polymorphic ciphers are superior in strength to single-key ciphers, even if the ciphers used in the polymorphic sequence are less secure than the single-key cipher. In Section IV, we present our Conclusions along with possible Future Work, where we summarize our work in this paper and set the stage for further investigations.

979-8-3315-6501-5/25/$31.00 ©2025 IEEE

II. Background

A. Strength of a Cipher

Evaluating the strength of a cipher involves determining how long it can keep the message or file it protects secure. The standard measure of security is based on how many keys must be tested to find the correct key. These keys are browsed from the key space of the cipher. In some cases, the ciphertext (CT) gives clues as to the correct key, facilitating a "heuristic" attack [4] on the cipher. The ciphers that aren't prone to heuristic attacks are difficult to decrypt and are peer reviewed as relatively "hard" or "strong" ciphers [5].

Unlike a heuristic attack, a brute force attack is guaranteed to decrypt a message eventually [6]. This attack attempts to try each key in the key space to break the cipher. While it is always effective, running the attack can take a considerable amount of time, sometimes so much time that the attack becomes mathematically intractable. The average number of keys ($|K_{avg}|$) to be tried before

we solve the cipher by randomly picking a key is given by,

$$|K_{avg}| = \frac{|K|}{2} \tag{1}$$

The brute force attack sets the standard for all attacks.

The key space for different types of ciphers has been extensively researched, yielding the following results. Each major type of cipher is listed, along with its key space.

1) Substitution cipher - The most basic type of cipher. A substitution (S) cipher maps the plain text (PT) onto its ciphertext (CT) using the relationship $\forall PT$:

$$PT_i \mapsto CT_i \tag{2}$$

with each mapping being unique for $A \mapsto A'$. This is the same problem as picking different color balls out of a bag without replacement [7]. A and A' may or may not be identical alphabets. In this case, the key space ($|K|$), the possible mappings between the two alphabets, is known to be

$$|K| = |A|! \tag{3}$$

2) Permutation cipher - The permutation (P) cipher reorders the bits in a block of PT to make the CT output [8]. The key shows the mappings from the PT bit location to the CT bit location. Any size block is possible. The key space for the P cipher is (see [9])

$$|K| = 2^{|B|} \tag{4}$$

3) Cascading ciphers - Following the thinking that if one encryption is good, then multiple encryptions of the same message must be much better, the cascade cipher [10] was developed. In a cascade cipher, any number and order of encryptions can be applied. The formula for the encryption is

$$E_k(M) = E_{c_0,k}(E_{c_1,k}(...(E_{c_n,k}(M)...) \tag{5}$$

Since the same key is used for each cipher, it is only necessary to find the key for the least secure cipher. Once found, the message immediately becomes clear. Therefore, the target key space is that of the simplest cipher in the cascade to break.

4) Product ciphers - These ciphers [11] are similar to the Cascade cipher; the product cipher uses the same general form, but with one major difference. Product ciphers use different keys for each encryption cipher. Therefore, the formula for encryption becomes

$$E(M) = E_{c_0,k_0}(E_{c_1,k_1}(...(Ec_n, k_n(M)...) \tag{6}$$

A common mix of ciphers, forming a product cipher, involves various combinations of S and P ciphers, with PSP and SPS being the basis for round ciphers. The key space for product ciphers is the product of the key spaces for each cipher in the product chain.

$$|K| = \prod_{i=1}^{n} |K_i| \tag{7}$$

Maurer et al. [10] indicated that the product cipher is at least as strong as its strongest cipher. There may be any number of ciphers applied in the product cipher.

5) Serial ciphers - A serial cipher uses a serial stream of PRNG-generated numbers as keys in encrypting a message. As each new block of information is presented, the PRNG (Pseudorandom Number Generator) number is then used as a key, typically for an XOR cipher, to encrypt the data. Because the PRNG is deterministic, the cycle for the PRNG is also the number of keys in the cycle. If the block size is $|B|$, then the key space is

$$|K| = 2^{|B|} \tag{8}$$

B. Isomorphism

Isomorphism is the substitution of one cipher with another to ease the analysis and decryption of a set of ciphers that comprise a product cipher. The ability to replace one cipher with an equivalent cipher comes from work done by Feistel during the early 1970s. He stated that all ciphers are S ciphers [12]. Each cipher defines a mapping in its own way, but a mapping nonetheless. Therefore, every basic cipher can be replaced by an equivalent S cipher with the proper key.

The replacement of ciphers by an equivalent S cipher is important because it allows for the use of the property of idempotence [13] (or "composition under closure" [14]). Idempotence refers to the property that for repeated applications of S ciphers, there exists an equivalent S cipher with a new key. That is, for two substitution cipher keys S_0 and S_1, $\exists S'$ such that

$$E_{S_0}(E_{S_1}(M)) = E_{S'}(M) \tag{9}$$

This property is known as "idempotence" [15] and has the effect of allowing many ciphers to be reduced to an equivalent S cipher and then be combined with other S ciphers. For example, AES [16] is a PSP cipher. However, P ciphers can be reduced to S ciphers, making it effectively an SSS cipher, which can be combined into a single S cipher via idempotence. Idempotence applies to ciphers of the same block size and without morphing [3].

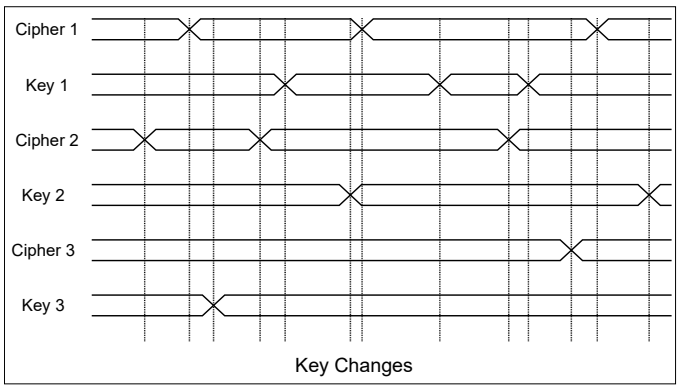

Fig. 1. Time Domain Multiplexing

C. Polymorphism

Idempotence does not apply to polymorphic, or many-form/morphing messages. This type of encryption changes cipher types and keys (also known as "key pairs") frequently and irregularly. Further, the keys and cipher types can change independently of each other on a randomly selected schedule. The time that a key or a cipher selection is active is known as the "time to live," or TTL. The shorter the TTL, the more secure the message, as the information in the encrypted shard using the key pairs is minimized. Although the speed at which key pairs can be changed is limited, the independence of the key/cipher selections allows for smaller shards with the same key pair. This effect is illustrated in Figure 1 [17]. Each shard consists of a series of keys that are different from those of every other shard. Each shard requires a different key pair combination, effectively reducing the shard size and increasing the key space.

III. Analysis

A non-sharded encryption has a single-key/cipher pair, with $|K| = C$, where C is a constant for the entire message. So long as the shard results in

$$C \leq \prod_{i=1}^{n} |K_i| \tag{10}$$

where $|K_i|$ can be the key space for either a single cipher or a product cipher

$$|K_i| = \begin{cases} |C| & \text{if a single cipher} \\ \\ \prod_{j=1}^{m} |K_j| & \text{if a product cipher} \end{cases} \tag{11}$$

Comparing product ciphers and polymorphic ciphers requires applying the concept of idempotence to the ciphers. Each cipher in the product grouping can be reduced to the S cipher, including the P cipher [18]. Product ciphers can be easily reduced to a single S cipher using the idempotence property. Even P ciphers and serial ciphers can be reduced, but it is easy to compare key spaces. However, even after reducing all of the ciphers used in a polymorphic cipher engine, each shard constitutes a different cipher.

Assume that the shards have the same size as those of another cipher. That is, if the single cipher/key encryption has a block size of $|B|$ characters, then after isomorphic reduction, the key space is $|K_S| = |B|!$. Again, assume that the polymorphic encryption uses uniform shard sizes of $|B|$. Then, assume the entire message is encrypted using a polymorphic cipher, and each cipher/key pair is also identical. In this case, the key space ($K_P|$) is identical for each shard, and the key spaces for the P and S ciphers are identical. That is,

$$|K_S| = |K_P| \tag{12}$$

However, the probability of two keys i and j where $i \neq j$ being identical in a random key selection is

$$pr(k_i = k_j) = \frac{1}{|K|^2} \tag{13}$$

The probability that all keys in a message of n shards are equivalent is

$$pr(\forall i \rightarrow k_i = k) = \frac{1}{|K|^n} \tag{14}$$

As n approaches ∞, $pr(\forall i \rightarrow k_i = k)$ goes to 0. Therefore, this is almost impossible. If even one shard has a different key, then the key space for the messages is

$$|K_P| > |K_i||K_j| \geq |K_S|^2 \tag{15}$$

For a key space with more than a single key, it is always true that

$$|K_S|^2 > |K_S| \tag{16}$$

A better approximation would involve the Birthday paradox [19]. The birthday paradox bounds the probability of any two random choices from a set being identical. Deterministic selection of keys, such as those generated by a PRNG, is more likely to produce collisions in the key values. Still, unless the PRNG cycle is trivial (that is, a cycle in PRNG, $\lambda = 1$) or there is only a single block in the message, then there will be at least two different keys.

The same analysis can be applied to the cipher selection process.

Next, consider the case in which the shard size of a polymorphic cipher can change, as well as the cipher/key pair. In the case of a differently sized shard, if the block size also changes, then the keys of two differently sized blocks cannot be identical. Therefore, for those two blocks

$$|K_P| \geq |K_{S_1}||K_{S_2}| \tag{17}$$

So long as one of the shards has a block size greater than or equal to the block size of the single-key cipher, then the key space will also be greater than that of the single-key cipher. Even repeated small block size ciphers add up quickly. The larger the message, the more likely it is that the key space for the polymorphic cipher will exceed that of the single-key cipher's key space. For example, a 64-character block has the same key space as approximately 17 blocks of 8 characters. Further, if any shard has a block size larger than the block size of the single key cipher, then it must be that $|K_P| > |K_S|$. This is true for all but trivial messages.

Next, consider the case where the polymorphic cipher uses identical, but differently sized blocks than the original cipher. It has been previously demonstrated that if the block size is larger than that used by the original cipher, the polymorphic cipher is more secure. But what if the block size of the polymorphic encryption is smaller than that of the original encryption? For the smaller block size, the key space is given by

$$|K_p| = \prod_{i=1}^{n} |A|^{|B_i'|}! \tag{18}$$

In this case $|K_P| > |K_S|$ when

$$\prod_{i=1}^{n} |A|^{|B_i'|}! > |A|^{|B|}! \tag{19}$$

The largest block size possible in this case is $|B_i'| = |B - 1|$. In this situation, it is necessary to determine how many shards must be processed by the polymorphic cipher to achieve and exceed the level of security that is achieved in the single cipher case. A polymorphic cipher must have at least two cipher/key pairs. Therefore, the relationship of a two-shard solution for cipher/key pair security is

$$|A|^{|B|} = |A|^{B-1}|A| \tag{20}$$

This assumes that the second shard has a character size of 1. So the minimum number of shards for equal security is two shards, while three such shards are guaranteed to exceed the key space of the single cipher.

Next, assume that the block size is one. Therefore

$$(|A|^{|B|})! \leq \prod_{i=1}^{|B|} (|A|)! = (|A|!)^n \tag{21}$$

where n is the number of single-character shards needed for the equation to be true.

Using Sterling's approximation [20] for $(|A|^{|B|})!$, the size of the key space for the single-key cipher is

$$|K_S| \approx \sqrt{2\pi |A|^{|B|}} \left(\frac{|A|^{|B|}}{e} \right)^{|A|^{|B|}} \tag{22}$$

Therefore, we seek the lowest number of characters n, such that,

$$(|A|!)^n \geq \sqrt{2\pi |A|^{|B|}} \left(\frac{|A|^{|B|}}{e} \right)^{|A|^{|B|}} \tag{23}$$

Let $m = |A|^{|B|}$, so that the equation becomes,

$$(|A|!)^n \geq \sqrt{2\pi m} \left(\frac{m}{e} \right)^{m} \tag{24}$$

Then the task is to solve for the value of n, the number of shards of size one character that have the same, or larger key space than the key space for the single-key cipher. At that point, the user can be assured that the security of the polymorphic encryption is as strong, or stronger, than that of its single-key counterpart.

$$n = log_{|A|!}(|A|!)^n \geq log_{|A|!} \left(\sqrt{2\pi m} \left(\frac{m}{e} \right)^{m} \right) \tag{25}$$

Taking the $log_{|A|!}$ of both sides of the equation leaves a long and complicated value on the right side of the equation. Next, that value is simplified to make it easier to calculate and draw conclusions about the equation. To save space, the notation is also simplified slightly. A is used in place of $|A|$ and B is substituted in place of $|B|$.

The key concept of $log(AB) = log(A) + log(B)$ is then repeatedly used to simplify the equation [21]. As a first step, note that

$$\sqrt{2\pi m} = \sqrt{2\pi}\sqrt{m} \tag{26}$$

and then the equation is subjected to the separation of the multiplicands.

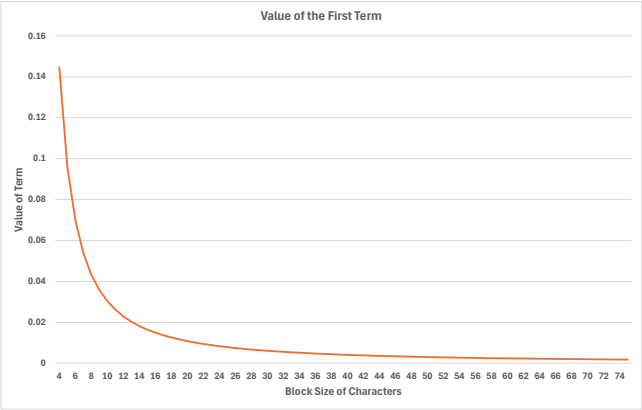

Fig. 2. Limit of First Term Value

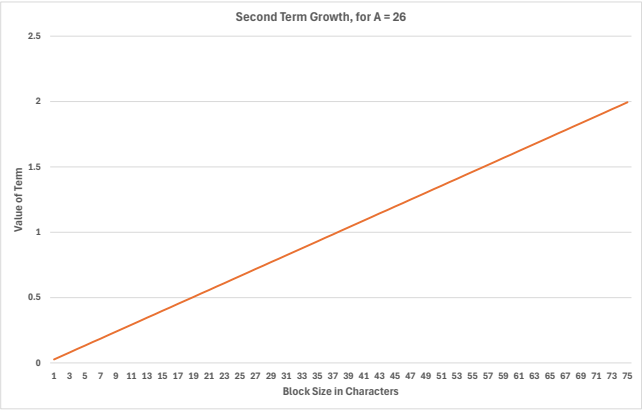

Fig. 3. Rise of the Second Term Value

$$n \geq log_{A!}\sqrt{2\pi} + log_{A!}\sqrt{A^B} + log_{A!}\left(\frac{A^B}{e}\right)^{A^B} \quad (27)$$

$$\approx log_{A!}(A^B)^{\frac{1}{2}} + A^B log_{A!}\left(\frac{A^B}{e}\right) \quad (28)$$

$$= A^B log_{A!}\left(\frac{A^B}{e}\right) + C \quad (29)$$

The first term, $log_{A!}\sqrt{2\pi}$, is so small compared to the remaining values that it may be ignored. The smallest alphabet used in human language is that of the Rotokas language [22], which has only twelve characters. Most languages have much larger alphabets of more than 20 characters, making the log of the term approach zero as the size of A increases (see Figure 2).

The second term is $log_{A!}\sqrt{A^B}$. While A^B grows exponentially, it is offset by taking its square root and even further reduced by taking the log of the result. This term rises linearly and can be easily estimated using the slope of the curve shown in its growth. The example shown in Figure 3 is charted for English ($|A| = 26$). Even at $B = 75$, the number of characters added is less than 2. For the sake of calculation, this number will be ignored when evaluating the final term.

The third, and last, term is A^B tempered by $log_{A!}\left(\frac{A^B}{e}\right)$. Using English as the example used to illustrate that ratio (see Figure 4), the value of A^B rises linearly but slowly. Also, until $B = 20$, the multiplier is less than 1, meaning that fewer than A^B blocks are required.

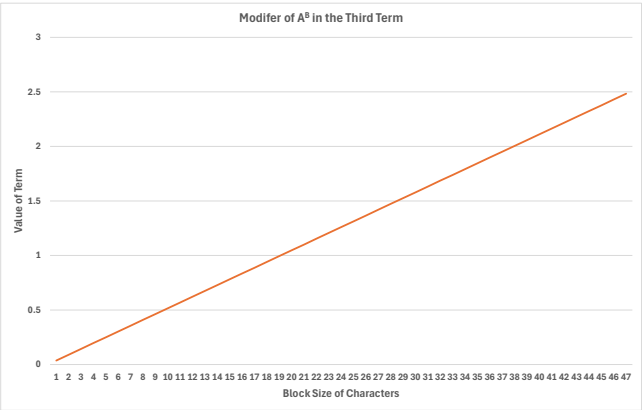

Fig. 4. Rise of the Third Term Value

An estimate of the range of characters required to guarantee that the polymorphic cipher will have more security is given by the number of characters in the range $1 \leq n \leq (A^B)C' + 2$ where C' is the multiplier of the third term. The high end of the range occurs only when the polymorphic cipher uniformly uses a block size of one character. In practice, this does not happen. Larger block ciphers are typically used in the encryption library, making the security break-even point much closer to a single block.

If the message is shorter than n, then there is not enough information available to determine the correct encryption. This n is the theoretical minimum; however,

in practice, an average of 10 - 50 times the value for the theoretical n is required [23]. Any message whose size is below that of the theoretical n is deemed to be trivial, since it cannot be reliably decrypted correctly.

IV. Conclusions and Future Work

A polymorphic cipher will be at least as secure when the cipher/key pairs of the polymorphic cipher do not change and the block sizes are identical to the original cipher. In this case, the number of blocks required to ensure superior security and key space is 2 (see Table I). However, if at least one shard has a larger block size, then only a single block is needed to guarantee greater security.

Perhaps a more telling comparison is the worst-case imbalance between the single-key cipher and the sharded polymorphic cipher with a block size of one character. While this is a somewhat artificial comparison, since most polymorphic encryptions utilize strong ciphers with $|B| >> 1$, the comparison helps illustrate the range of required characters. Although the number of characters (note that this refers to individual characters, not blocks) is larger, it is not excessively large. This number should be interpreted in the context of the unicity distance for the cipher. If the message is shorter than the practical unicity distance of the single-key cipher, then the two ciphers are equally safe, and the polymorphic cipher is just as secure as the single-key cipher. However, typically, the number of characters/blocks shown in Table I is reached before the unicity distance is met. Messages of less than the unicity distance for the single key cipher are termed "trivial" for the comparison.

Polymorphic ciphers must be more secure than single-key/key-pair ciphers when

1) Block ciphers change keys or ciphers pseudorandomly, and block sizes are the same
2) The polymorphic cipher varies block sizes, and the message is more than trivially long
3) The polymorphic cipher has block sizes different than the original cipher

These figures demonstrate the superior security of polymorphic ciphers. Future work requires that polymorphic ciphers be designed to avoid the use of single-character ciphers and ensure that larger ciphers are selected early in the encryption process. By early, it is meant that these larger key ciphers appear at least once before the unicity distance is reached for the message.

Ciphers used in encryption libraries must also be evaluated for their key space, so that the encryption program can use them for cipher selection. Such a rating must be made in light of the isomorphic reduction of ciphers to be accurate.

Largest Polymorphic Block Size	Characters to Guarantee
1	$A^B log_{A!}\left(\frac{A^B}{e}\right) + C$
$\lvert\overline{K_P}\rvert$	$(A^B - \lvert\overline{K_P}\rvert) + 1$
$\lvert K_S\rvert$	2
$> \lvert K_S\rvert$	1

TABLE I
Summary of Character to Guarantee

References

[1] Anshu Sharma. Protect and index sensitive data with polymorphic encryption. *The New Stack*, May 2022. [Online; accessed 9-July-2025].

[2] Skyflow. https://www.skyflow.com/post/what-is-polymorphic-encryption, July 2024. [Online; accessed 9-July-2025].

[3] Albert Carlson, Indira K. Dutta, Bhaskar Ghosh, and Michael Totaro. Modeling polymorphic ciphers. *Sixth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2021.

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

[5] Jeffrey L. Vagle. Furtive encryption: Power, trusts, and the constitutional cost of collective surveillance. *Indiana Law Journal*, 90(1).

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

[7] Sheldon Ross. *A First Course in Probability*. MacMillan Publishing, Inc, New York, 1976.

[8] R.B. Lee, Z.J. Shi, Y.L. Yin, R.L. Rivest, and M.J.B. Robshaw. On permutation operations in cipher design. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, volume 2, pages 569–577 Vol.2, 2004.

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

[10] Uli Maurer and James Massey. Cascade ciphers: The importance of being first. *Journal of Cryptology*, 6(1):55 – 61, 1993.

[11] Alex Biryukov. *Product Cipher, Superencryption*, pages 1969–1970. Springer Nature Switzerland, Cham, 2025.

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

[13] Richard Wells. *Applied Coding and Information Theory*. Prentice Hall, Upper Saddle River, 1999.

[14] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM (JACM)*, 44(4):527–548, 1997.

[15] 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.

[16] National Institute of Standards, Technology (NIST), Morris J. Dworkin, Meltem Sonmez Turan, and Nicky Mouha. Advanced encryption standard (aes), 2023-05-09 04:05:00 2023.

[17] Jim Kurose and Keith Ross. *Computer Networking A Top-Down Approach*. Pearson, 6th edition, 2013.

[18] 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.

[19] E. H. McKinney. Generalized birthday problem. *The American Mathematical Monthly*, 73(4):385–387, 1966.

[20] E. W. Weisstein. Stirlings approximation. https://mathworld.wolfram.com/StirlingsApproximation.html.

[21] John W. Coburn and Jeremy P. Coffelt. *College Algebra*. McGraw - Hill, New York, NY, 3rd edition.

[22] Stuart Robinson. The phoneme inventory of the aita dialect of rotokas. *Oceanic Linguistics*, 45(1):206–209, 2006.

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