

The Problem with Regular Multiple Byte Block Boundaries in Encryption

Albert Carlson^{*}, Indira Dutta[†], Bhaskar Ghosh[‡], and Robert Hiromoto[§]

^{*}Computer Science Department, Austin Community College, Austin, TX, USA.

[†]Dept. of Computer and Information Sci., Arkansas Tech University, Russellville, AR, USA.

[‡]Dept. of Computer and Information Sci., Arkansas Tech University, Russellville, AR, USA.

[§]Computer Science Department, University of Idaho, Moscow, ID, USA.

Email: ^{*}albert.carlson@austincc.edu, [†]idutta@atu.edu, [‡]bghosh@atu.edu, [§]hiromoto@uidaho.edu.

Abstract—Product ciphers composed of multi-byte permutations and substitutions are generally considered to be more secure than the substitution ciphers. However, this is not the case. In fact, product ciphers using the same boundaries can be isomorphically reduced to substitution ciphers. These product block ciphers are characterized by block definitions having the same defined block sizes. This also means that such product ciphers can be attacked using the same methodology as can be applied to all substitution (S) ciphers. We show that Permutation-Substitution-Permutation (PSP) ciphers are insignificantly more secure than a substitution cipher and propose a chosen plaintext attack for solving these product ciphers.

Index Terms—Cryptography, Set Theory, Polymorphic Encryption, Ciphers, Shannon Theory

I. BACKGROUND

Since Shannon introduced the concept of increasing security by compounding ciphers, it has generally been accepted that product block ciphers [1] of the form PSP [2], [3] are more secure than a cipher employing only a P or an S cipher. This is not true for block ciphers whose encryption algorithms end at byte (character) boundaries. However, blocks of integral character size suffer from a significant weakness. We show that as the block size of the PSP cipher increases above two symbols the additional security gained is insignificant compared to a simple substitution cipher of the same block size.

A. Isomorphic Reduction and Key Space Reduction

Feistel, a cryptographic researcher at IBM in the mid-to-latter half of the 20th century, wrote in *Scientific American* that all ciphers are, at their heart, S ciphers [4]. Carlson took this fact and extended it [5] for all ciphers. His work demonstrates that characterizing the blocks as members of a new alphabet whose character size is identical to block size in a “metaalphabet” allows the cipher to be “reduced” to the S cipher [6]. This allows larger blocks to be abstracted into more easily attacked ciphers.

Using the concept of isomorphic cipher reduction as a base, it is also possible to simplify the evaluation of the keys by making use of Set Theoretic Estimation (STE) [7], [8], [9]. While the reduction of ciphers simplifies the attack on the cipher text (CT), it often comes at the expense of increasing

the key space for the equivalent cipher. The strategy then turns to quickly reducing the keys that have to be examined for recovering the original plain text (PT). This is often accomplished by using the technique of examining “equivalent keys” [5]. An equivalent key is one which produces the same encryption or decryption for a message as another key. This typically occurs when the message does not use all of the characters in the alphabet, allowing for multiple permutations of the keys that keep the mappings for characters that appear but have different mappings for missing alphabetic characters [10]. All of the keys that show this property are said to be “isomorphic” to each other. That is, they have the same form of outcome and can be grouped together. Since they will encrypt/decrypt with the same results, they can be grouped into sets with each other. Taking any of the keys as representative of the entire set, known as the “systematic isomorph” for the set, that key is tested for its ability to break the encryption. If it does not correctly decrypt the message then none of the other keys in the set will do the job, either. If the systematic isomorph does successfully decrypt the message then any of the keys will also work. While the correct key may not be the systematic isomorph any of the keys can be considered correct because they are indistinguishable from the encryption key. The goal is to find any key that affects decryption, and in this case there exist n keys where

$$n = |\{S_I\}| \quad (1)$$

where $\{S_I\}$ is the set of isomorphs to the equivalent key. This means that for any such situation the number of keys that needs to be tried in a brute force attack [11] against a message is the number of systematic isomorphs.

B. PSP Equivalence

Consider a message encrypted by a P cipher, followed by an S cipher. Without loss of generality, let the plain text consist only of lowercase alphabetic English characters with all spaces and punctuation removed. The message uses standard ASCII encoding and the permutation employs a three character (byte) block. Byte and block boundaries are known for the encrypted message. The equivalence of the PS and S ciphers is as follows:

1) *Theorem 1*: Product ciphers of the form PS or SP aligned at character byte boundaries provide negligibly more security, in terms of greater unicity distance, over a substitution cipher.

Proof: Shannon defined the unicity distance for a cipher as [2]

$$n = \frac{\log|K|}{R_L \log|A|}. \quad (2)$$

Let the number of bits in a block be represented by B . If the block is on byte boundaries, then $B = 8m$, where m is the number of bytes in a block. The number of symbols in the alphabet is given by $|A|$. For a block of m characters, with one character/byte, the number of characters in the new, or metaalphabet of the language is $|A|^m$. The key space of a product cipher is the product of the key spaces of the individual ciphers [12]. For simplicity, we analyze a substitution product cipher (PS) and then generalize it for the PSP product cipher.

The product cipher PS is composed of a total of B bits. By selecting one bit at a time without replacement [12] there are $B!$ possible keys [3]. For an S cipher of m characters in an alphabet A , there are at most $|A|^m$ possible symbols [3]. Some combinations may be forbidden [13] (never encountered in the language) - such as 'q wz'. For the P cipher

$$n_p \leq \frac{\log(B!|A|^m)}{R_\lambda \log(|A|^m)}, \quad (3)$$

whereas for a substitution cipher of the same block size

$$n_s \leq \frac{\log|A|^m}{R_\lambda \log(|A|^m)}. \quad (4)$$

Let S be defined as the "security measure" of a cipher compared to the S cipher. S shows the relative strength of any type of cipher compared to a block S cipher of identical key size. Therefore,

$$S = \frac{n_p}{n_s} \quad (5)$$

Then

$$S = 1 + \frac{\log(B!)}{\log(|A|^m)} \quad (6)$$

Let

$$\epsilon = \frac{\log(B!)}{\log(|A|^m)}. \quad (7)$$

This equation becomes

$$S = 1 + \epsilon. \quad (8)$$

The value of ϵ for small block sizes is shown in Table I. Using ϵ for this measure is not accidental. Normally ϵ is the error found in a function and is bounded in Set Theoretic Estimation (STE) problems [5], [14]. In this case, it is also represents the additional security provided by a product cipher with respect to the S cipher. That additional security becomes negligible after as little as three bytes.

Choosing the values of n as a comparison of cipher strength is intuitive and mathematically valid. Since n is proportional

TABLE I
SECURITY FOR A BLOCK OF m BYTES

m	ϵ
1	0.173103
2	0.008216
3	0.000355
4	1.49×10^{-5}

to the number of keys (key space) and the effort required to break a cipher using the brute force attack [11].

2) *Theorem 2*: Within the encoded representation of a symbol, a P encryption is equivalent to an S encryption.

Proof: Let the symbol A of an alphabet A' for a language be represented by a collection of n bits where $n = |\log(A')|$. A' consists of all combinations of bits for a symbol of size $|n|$ and $A \subseteq A'$. A permutation preserves the number of 1 and 0 bits but rearranges them into another symbol, $B \in A'$. A permutation results in a mapping of $A \mapsto A'$ - the definition for a substitution cipher.

3) *Theorem 3*: Within the boundaries of an encoded representation of a symbol, the PSP reduces to an S cipher.

Proof: By Theorem 2, P and S ciphers are equivalent within a symbol boundary. Therefore, PSP reduces to SSS. S ciphers are idempotent [3]. Therefore SSS cipher reduces to a single S cipher.

By using a meta language consisting of symbols of size $|m|$, the size of an encrypted block, all permutation takes place inside symbol boundaries. PSP ciphers can be attacked as if they were S ciphers. One effective attack on PSP ciphers is the chosen plaintext attack. The selection of chosen plain text is simplified by searching the language statistics for a block of size $|m|$. Without loss of generality, take a block PSP cipher for $|m| = 3$ encrypting English.

C. Composite Keys

Evaluating cipher strength for product block ciphers is similar to the method used for simple block ciphers, such as the P cipher. Ciphers with a more complicated structure, such as block product ciphers [1], require a bit more analysis. Product ciphers are composed of multiple ciphers, often with different key spaces. For product ciphers each cipher key must be identified. This results in a key space size composed of the cross product of the key space of each of the constituent ciphers. That means that for a product cipher with n ciphers the key space is

$$|K| = \prod_{i=1}^n |K_i| \quad (9)$$

This equation reflects the math saying that each cipher must be solved independently of the other ciphers. In effect, there is still a single key that is composed of the correct key for each cipher that is applied linearly for decryption. A key, so composed, is called a "composite" key. The combination of

the various keys for the ciphers come from the cross product of each key space.

Cascade ciphers [1] are a different type of block cipher than the traditional product cipher. While they are composed of n different ciphers, all of the ciphers use the same key. Therefore, only one cipher needs to be solved. That cipher is the “easiest” cipher in the group. By easiest it is meant the cipher that can be most easily broken. Typically, this is the cipher with the smallest key space for the selected key. Cascade ciphers fall under the same analysis as previously presented for the P cipher, adjusted for the proper key space analysis for the weakest cipher used in the cipher structure. Because of this reduction in key space, only the product cipher will be considered in the algorithm.

II. ALGORITHM

The algorithm for use when product ciphers are applied to blocks that are a width of n bytes involves treating the entire block as a single character in a meta language. Each symbol in the meta language is an m -gram of English. This reduces the number of symbols in the alphabet in the meta language to the number of m -grams allowed in English [5], [15]. Each symbol is analyzed for common language statistics based on their appearance in English [16]. The permutation is then confined at a single symbol level and is equivalent to an S cipher mapped according to the permuted bits. The same approach can be used for any block cipher having the same size block. Even if the block size varies for each block cipher and key, so long as the block size is not varying over the message, the different ciphers will have block boundaries that coincide with a cycle length of at most

$$\lambda = \prod_{i=1}^n |k_i| \quad (10)$$

The cycle may be smaller, depending on the lowest common multiple (LCM) of the key sizes for the different ciphers in the product cipher. But, in no case will λ be less than the largest $|k_i| = |k_i|_{max}$. Therefore the range of the cycle is

$$|k_i|_{max} \leq \lambda \leq \prod_{i=1}^n |k_i| \quad (11)$$

Setting the metacharacter size to λ allows for the next step in the algorithm, which is applying isomorphic cipher reduction [6], [17]. Decryption then proceeds using the procedure for the S cipher [5]. This method was demonstrated by Carlson, et. al in breaking a variety of ciphers, including AES with CBC, with success [18], [19]. However, if the length of the encrypted message is greater than λ then the shard (submessage) being evaluated cannot be decrypted using isomorphic cipher reduction. The high level description of the algorithm is shown in Figure 1.

This method is made viable by the alignment of the blocks of the various ciphers aligning periodically. It is important to note that this algorithm can be used even if each of the ciphers have different block sizes. However, as long as each

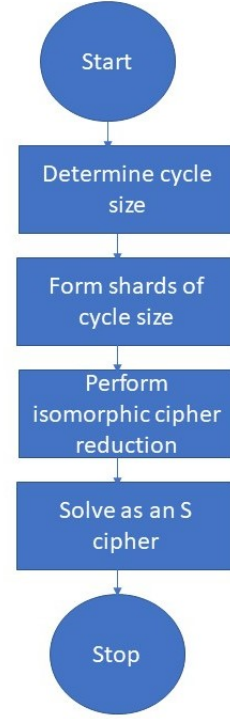


Fig. 1. The High Level Algorithm

key terminates on a block boundary, the size of the keys does not matter. Eventually, the blocks of each cipher will align. If such confluence was not made then it would be impossible to employ metacharacters in the attack against the encryption scheme.

There are additional problems, though, with using block ciphers as they are traditionally implemented. That is, with bytes that are integer multiples of full bytes. These are now considered.

A. Implementation of the Algorithm

For a cipher of the type defined by the PSP model, the key space is $B!|A|^m B!$. This includes product ciphers that have been subjected to isomorphic cipher reduction. Even the most complex block ciphers now in use employ this PSP architecture. Most often this comes in the form of an initial P cipher used to mix the input plain text, followed by multiple rounds of combinations of S and P ciphers. They may be “unbalanced,” having the plain text split into two, or more segments, and different encryptions applied to each segment. Finally, the last step is another mixing P cipher. It has been previously shown that P ciphers can be replaced with S ciphers with the object of applying idempotence to the encryption algorithm. The final result is an S cipher as shown in the proof in Section Completing the substitution into the formula for S results in Section I-B. The security ratio is given by

$$S = 1 + 2\epsilon. \quad (12)$$

Increasing the block size results in a larger number of possible symbols in the metalanguage. This means that having a product cipher with ciphers of different block size can potentially be more secure. An actual increase in security comes if the LCM of the various block sizes is not identical to the maximum block size. That is, a security increase results if

$$LCM > |K_{max}| \quad (13)$$

where $|K_{max}|$ is the maximum number of keys.

The maximum number of characters in the meta language for a block size of m bits is $(A_{max})^m$, where $A_{max} = 2^{8m}$ for ASCII characters. In practice typically it is the case that $|A| < A_{max}$. There are several reasons for this reduction in the alphabet size. While, in general, there is additional security from having a larger alphabet, it is to the advantage of cryptographers to remove symbols that help code breakers, such as capital letters, spaces, and punctuation. A larger alphabet gives a larger key space in an S cipher ($|K| = |A|!$), making larger alphabets desirable. However, this effect can be overwhelmed by the loss of entropy for certain members of the alphabet. Restricting encrypted symbols to lowercase letters is one such compromise that changes the entropy. For example, the use of upper case letters can give the attacker information about where sentences begin and indicate proper nouns. Proper nouns tell the attacker that a group of letters indicate a place or name. That information restricts the mapping of following symbols to the set of names and places, also indicating that the information is of special significance and interest. The reduction in alphabet size due to the conversion to lower case only letters restricts the alphabet size to $|A| = (\frac{26}{256})A_{max}$. The number of symbols in the alphabet is further reduced because some combinations of letters (m -grams) are never encountered in a language. These combinations of characters are called “forbidden” m -grams and can comprise a major part of the permutations of a potential alphabet (see Table II), based on the survey of English prose style from 1600 - 2000AD done by Carlson, et al [13]. Similar lists can be found online compiled by various academic [20] and commercial sources [21]. The conclusion is that increasing the block size does increase the key space, but not as much as would appear when the actual number of symbols are compared to the maximum possible increase. The reduction in useful expansion to the alphabet occurs much less rapidly than would be indicated by just analyzing the size of the block permutation.

Every block that is broken contributes to the overall information that can be applied to the final decryption. The question becomes, how many blocks must be found before the message can be reasonably read by filling in the remaining blocks easily? There is no single answer to this question. Rather, there is a range of the number of blocks, specified as a percent number rather than an absolute number. Shannon indicated that such a range can be accepted as an answer. As motivation to support such a procedure and range it should be noted that the entropy for information accumulation in speech and transmitted messages comes from the variation of the

user habits of the speech or message. Langendoen and Postal [22] established the variation of individual speech patterns based on the theory that each person has their own personal language with the associated habits and probabilities. They posit that the natural languages we are familiar with are the intersection ($\cap$) of individual languages that are shared by the users, such as English, German, Russian, and so on. This is highly suggestive of a possible Set Theoretic Estimation (STE) approach to a solution [5], [9]. Further, Lucks [23] notes that the probability density function (pdf) of words in English (and the same concept, with different pdf numbers, applies to other natural languages) is that roughly half of a message consists of less than 150 words. Depending on the habits of the individual user, these common words can be arranged in various configurations, but at some point, a “normal” user of the language can correct lowercase the missing information.

As an experiment, we assumed that the same concept applied to m -grams in a language. However, there was no indication of how much information would be required to recover the original text using language statistics. To test this assumption, arbitrary levels of broken (or identifiable mappings) were assumed. High probability 3-grams were appended to the beginning of a text and encrypted, yielding known mappings in the encryption. The rest of the mappings were solved for using a language statistic attack in STE. The results are shown in Demonstration 1. Further indications concerning how much information is required comes from the history of World War II. The vital information about the attack at Midway Island came from the US breaking the Japanese JN-25 cipher. The amount of readable data amounted to a range of 10% - 15% of the transmitted data [24]. The implication from these two data points suggests that the range required for reading most of the message is from 10% - 25% to have at least a 50% chance of being able to assemble the rest of the message from what has been revealed. Many more studies will have to be done on this subject before the range can be narrowed. Further, due to the fact that different authors and users have different habits, it is likely that the number will never be more than a range.

Other studies will help to constrict this range. This range will probably never converge to a single number since there is a large variance in the language habits and pdf for individual language practitioners. However, at this point in time, there is an insufficient number of studies in this area to narrow that range.

1) Proposition: By selecting the 3-grams seen most frequently in our corpus, those used to derive the data used in Table II, and applying them to the encryption process, the mapping for the substitution is recovered. This supposition is based on the Lucks observation [23] on the frequency of words in a language. A list of m -grams was constructed after examining the contents of 41 texts and 294,838,109 characters covering the period of the 1500's to the late 20th century over multiple genre [5]. Each m -gram was ordered by the frequency of appearance in the corpus. The list included only

TABLE II
NUMBER OF FORBIDDEN/ALLOWED SYMBOLS

Letters	%Forbidden	%Allowed	Number Allowed
1	0.0000%	100.0000%	26
2	2.2189%	97.7811%	661
3	35.6224%	64.3776%	11315
4	75.9979%	24.0021%	109684
5	94.7024%	5.2976%	629431
6	99.3116%	0.6884%	2126661

TABLE III
3-GRAM COVERAGE FOR ATTACK

No. 3-grams	% Plain text Covered
449	50.025%
1242	75.008%
11315	100.000%

those highest frequency m -grams such that

$$\sum_{i=1}^n pr(x_i) = \lceil V \rceil \quad (14)$$

where V is the target frequency coverage. The test was run for coverages of 75%, 50%, and 25%. Fifty tests were run with different texts for each coverage level in order to meet the minimum level of statistical significance [25].

2) *Demonstration 1:* Table III shows how many 3-grams are required to recover a portion of the plain text. On average the analysis required less than 450 blocks to return the mappings for half of the plain text message. At coverage levels of 75% and 50% all tests returned a complete solution for the files. At 25% a varying amount of text was returned. There was still sufficient information to get the gist of the entire message with at least 25% of the file decrypted.

To limit the ability of an attacker to do isomorphic cipher reduction it is important to ensure that at least one of the ciphers used in the product cipher does not have a key size that ends on a regular byte boundary. Once the cipher is off of the byte boundary then an adjustment to the key space is made to ensure that the encryption remains at that bit in the byte. The way to visualize this process can be seen in Figure 2. The strategy is to choose the sharding of cipher and keys on different schedules. Such an alignment multiplies the effect of offsetting the shard boundaries and both causes the effective key space to be exponentially larger and makes the effective length of time that the composite key space is valid to be much shorter than could otherwise be achieved. However, unless at least one of the ciphers or keys is aligned off the byte boundary, the keys will cycle at the rate of the LCM of the block size. Using the off byte boundary eliminates the LCM and ensures that there can be no cycle. An example of this type of key space application was demonstrated by Dutta, et al [26].

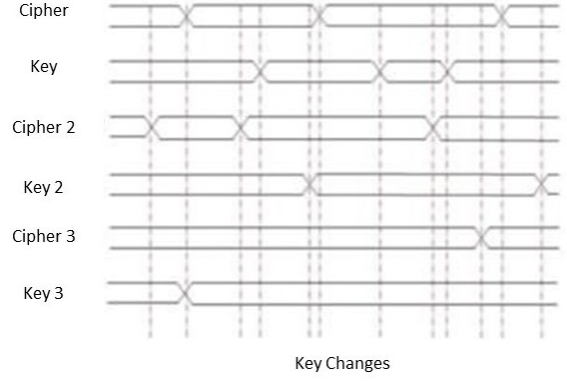


Fig. 2. Ciphers and Key Changes on Byte Boundary Schedules

III. CONCLUSIONS

Maximizing the security for a message requires eliminating regularity in encryption practices and habits. In calculating the security measure, S , between complex ciphers and the S cipher, the value of ϵ drops quickly as key size (m) increases. At the same time the value of the byte size (B) grows linearly while the value of $|A|^m$ grows exponentially. Therefore, S tends towards 1 as the block size increases. Increasing the number of symbols in a block does increase the unicity distance. However, the number of symbols is typically substantially lower than the maximum, resulting in a smaller key space than indicated by the maximum capacity of the meta alphabet. Little additional security, if any, is gained by using product ciphers of the PS, SP, PSP, or SPS form. All of these types of ciphers have the same security as a block S cipher due to isomorphic cipher reduction and idempotence.

The solution to this problem is to use a product block cipher that employs a different key and cipher selection for each cipher in the product cipher. If the schedule employs key sizes that are integer multiples of blocks then there is a cycle where all of the block boundaries align and isomorphic cipher reduction is possible. At least one of the ciphers should have a key size that ends inside a byte and remains at that same bit in a byte. This practice prevents the cycle at the LCM of the keys.

Another question arose during analysis of the practice of using block sizes that aligned to a multiple of the key size. That question was how much of the plain text must be revealed in order to be able to make reasonable guesses as to the mapping of the rest of the message. Historical data and our own experiments resulted in a range of from 10% to 50%. Other studies are required and will help to constrict this range. This number will probably not be a firm number due to the variance in language habits for people and pdf of words in a language.

Finally, ciphers employing combinations of permutation and substitution must not employ blocks that are regular and have byte boundaries. Such codes make the user susceptible to more effective attacks than previously identified. That susceptibility

is measurable and predictable. It also indicates that it may be more efficient to use S ciphers with staggered block sizes and offsets rather than specifically using a structure of PS, PSP, SP, and SPS ciphers.

[26] Indira K. Dutta. *A High-Performance Lightweight Polymorphic Encryption System for Constrained Internet of Things (IoT) Device*. PhD dissertation, University of Louisiana, Lafayette, 2021.

REFERENCES

- [1] Uli Maurer and James Massey. Cascade ciphers: The importance of being first. *Journal of Cryptology*, 6(1):55 – 61, 1993.
- [2] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656 – 715, 1949.
- [3] Richard Wells. *Applied Coding and Information Theory*. Prentice Hall, Upper Saddle River, 1999.
- [4] Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15 – 20, 1973.
- [5] Albert Carlson. *Set Theoretic Estimation Applied to the Information Content of Ciphers and Decryption*. PhD thesis, University of Idaho, 2012.
- [6] 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)*.
- [7] Hans Witsenhausen. Sets of possible states of linear systems given perturbed observation. *IEEE Transactions on Automatic Control*, AC-13(1):556 – 558, 1968.
- [8] Fred Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, AC-13(1):22 – 28, 1968.
- [9] Patrick Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182 – 208, 1993.
- [10] Albert Carlson, Bhaskar Ghosh, Indira Dutta, Shivanjali Khare, and Michael Totaro. Keyspace reduction using isomorphs. *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*.
- [11] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons Inc., New York, 2nd edition, 1996.
- [12] Sheldon Ross. *A First Course in Probability*. MacMillan Publishing, Inc, New York, 1976.
- [13] Albert H. Carlson, Richard B. Wells, and Robert E. Hiromoto. Using set theoretic estimation to implement shannon secrecy theory. pages 435 – 438., 2005.
- [14] Thomas Cover and Joy Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc, New York, 2nd edition, 2005.
- [15] Albert Carlson, Garret Gang, Torston Gang, Bhaskar Ghosh, and Indira K Dutta. Evaluating true cryptographic keyspace size. In *UEMCON 2021 IEEE 12th Annual Ubiquitous Electronics and Mobile Communication Conference*, New York, NY, 2021. IEEE.
- [16] Robert Lewand. *Cryptological Mathematics*. Mathematical Association of America, Washington D.C., 2000.
- [17] Cesar Polcino Milies and Sudarshan K. Sehgal. *An Introduction to Group Rings, Algebras and Applications*. Kluwer Academic Publishers, 2002.
- [18] Albert Carlson, Patrick Doherty, Isaiah Eichen, and James Gall. Breaking cbc, or randomness never was happiness. Internet Video, August 2015.
- [19] Albert Carlson, Patrick Doherty, Isaiah Eichen, and James Gall. Using collisions to break cbc, 2016.
- [20] Notre dame university, letter frequencies, <https://www3.nd.edu/busiforc/handouts/cryptography/letter-frequencies.html>.
- [21] Practical cryptography, "english letter frequencies," view-source:<http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/>.
- [22] D. Terence Langendoen and Paul Postal. *The Vastness of Natural Languages*. The Camelot Press, Ltd., Southampton, 1984.
- [23] M. Lucks. A constraint satisfaction algorithm for the automated decryption of simple substitution ciphers. In *CRYPTO 1988*. CRYPTO, 1988.
- [24] Station hypo, march 5, 1942 station hypo started reading japanese jn-25; station hypo, <https://stationhypo.com/2016/03/05/march-5-1942-station-hypo-started-reading-japanese-jn-25/>: :text=breaking the japanese code known, enciphered using an additive 20table.
- [25] R. Lyman Ott and Michael T. Longnecker. *An Introduction to Statistical Methods and Data Analysis 7th edition*. Cengage Learning, 2016.