[A. Carlson, G. Gang, T. Gang, B. Ghosh and I. K. Dutta. (DATE). Evaluating True Cryptographic Key Space Size, 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2021, pp. 0243-0249, doi: https://doi.org/10.1109/UEMCON53757.2021.9666530]

Evaluating True Cryptographic Key Space Size

Albert Carlson
CIT
Austin Community College
Texas, USA
albert.carlson@austincc.edu

Garret Gang
Schweitzer Engineering Laboratories
Pullman, USA
garretrgang@gmail.com

Torsten Gang
Infosys Corp.
Bengaluru, India
torsten.gang@infosys.com

Bhaskar Ghosh
CACS
University of Louisiana at Lafayette
Louisiana, USA
bhaskar.ghosh1@louisiana.edu

Indira Kalyan Dutta
Comp and Info Sci
Arkansas Tech University
Arkansas, USA
idutta@atu.edu

***Abstract*—Cybersecurity professionals have relied on the key space of a cipher to compare encryption algorithms and select the best encryptions for transmitted data. Peer reviewed strong ciphers have been assumed to maintain strength for all messages. It is thought that only brute force attacks can break these ciphers, so the key space calculation for these algorithms uses the maximum key space to determine the unicity distance. Unfortunately, the key space is heavily dependent on the user and habits of the user, as well as the content of the message. In this paper, we present factors that affect the key space size and show that the effects of these factors can seriously decrease the security of a cipher for a particular message. By considering these factors, a cybersecurity practitioner can properly assess vulnerability and choose the best security for that message.**

***Index Terms*—Cryptography, Key Spaces, Applied Mathematics, Cybersecurity**

I. Introduction and Background

One of the most common cybersecurity measures for messages is encryption. This technique is used to prevent attackers from reading the sensitive content of messages and keep information shared between users secure. Normally the strength of a cipher is measured in terms of the key space of a cipher. The cipher algorithms is assumed to be known to the attacker [1], [2] but so strong that the only attack possible is the Brute Force attack [3]. If this condition is correct then comparisons between algorithms can be made based on key space size ($|k_c|$). Assuming it takes a time of t_p to present a solution to a computer and evaluate whether or not the message has been decrypted, then it takes time t_b to recover a message, where

$$t_b = \frac{|k_c| t_p}{2} \tag{1}$$

on the average [4]. Therefore, comparisons between the security of a cipher is apropos, and the larger the key space the stronger the security. Cryptographers are aware of this math and produce ciphers that have scalable key spaces that can be increased as the value t_p drops with increasingly faster hardware. As the speed of hardware increases, ciphers with smaller key sizes have been abandoned for those with larger key sizes due to the increased key space and security related to the larger key size.

Key spaces are calculated using counting theory [5]. Most ciphers have key spaces that can be relatively large ($|k_e| \geq 10^{20}$) for even "easy" ciphers. Computers can quickly run brute force attacks for numbers of this size, so correspondingly larger key spaces are used. The simplest, least secure cipher algorithms have key spaces of size of $|A|!$ for Substitution (S) and 2^b for Permutation (P) ciphers.

Product ciphers [6] are made up of different combinations of S and P ciphers [7]. For example, the product cipher known as the Advanced Encryption Standard (AES) is made of combinations of P ciphers, XOR ciphers (a form of S cipher), S boxes, transposition of rows and columns (a form of P cipher), and a rotation of data (another form of P cipher) [3]. Multiple rounds of these ciphers are used to complete the encryption. In this case the key space is the Cartesian cross product of the possible keys. The resulting key space is found by multiplying the key space for each of the ciphers in use to create the product cipher. The total key space (k_t) is calculated by:

$$|k_t| = \prod_{i=1}^{n} |k_{c_i}| \tag{2}$$

for the n ciphers in the product structure.

Key space size is a key piece of information in calculating the amount of information that is necessary to recover an encrypted message. Shannon derived a formula for the amount of information that accumulates in an encrypted file and how it relates to the number of characters required, on the average, to be able to uniquely identify the correct key [2]. This measure, called the "unicity distance," arises from the concepts of "entropy" [8] and "redundancy" [2].

Entropy was first suggested by Hartley [8] and was recognized as an important measure for encryption by Shannon [2].

978-1-6654-0690-1/21/$31.00 ©2021 IEEE

Mathematically, entropy ($H(x)$) is defined as:

$$H(x) = -\sum_{i=1}^{n} pr(x_i) lg(pr(x_i)) \tag{3}$$

$H(x)$ describes the amount of information that is gained from knowing what the next letter, or group of letters, when they are revealed. Shannon later made use of entropy in order to define the effect of redundancy in language.

Carlson noted that, "Shannon [2] then demonstrated that enough information is contained in n-grams (groups of n consecutive letters) to effect the solution of a Caesar cipher (a specialized type of the S cipher). S cipher decryption methods often use letter frequency tables and n-grams to recover keys. Morton [9] expanded language statistics to incorporate words and then sentence structure." [10]

Further, "Shannon [2], [11] noted that every language contains redundancy. He further established that redundancy can be quantified as

$$R_\lambda = 1 - \frac{H(x)}{H_{max}(x)} \tag{4}$$

and interpreted as the tendency of symbols in a language (λ) to be repeated. Patterns in the language that are not removed prior to sending a coded message provide an opportunity for attack [12]. S ciphers have traditionally been attacked through the employment of redundancy, expressed statistically by letter, n-gram, and word frequencies [12]. An n-gram is a group of n consecutive letters found in a text.

"Shannon also stated that the average amount of information required to distinguish between spurious keys and the actual key is determined by

$$n = \frac{log|K|}{R_\lambda log|A|} \tag{5}$$

where $|K|$ is the key space size, $|A|$ is the size of the alphabet in language λ, and R_λ is the redundancy of language λ. The quantity n is known as the "unicity distance" for the cipher in language λ" [10].

Unfortunately for a legitimate user of a cipher, all keys are not always unique [13]. Two cipher keys, k_1 and k_2, are said to be "equivalent," if for a message (M), $D_{k_1}(M) = D_{k_2}(M)$. Keys need not be equivalent for all possible messages. For example, assume the message (M) does not contain the symbols q, x, or z in its plaintext and is encrypted using a S cipher. Any key that contains the same mappings for all symbols except q, x, or z is equivalent to any message not including q, x, or z. Without those symbols in the message, it is impossible to differentiate other possible keys from the correct key (k_e).

Keys are applied to messages and files that are composed of groups of symbols of n-grams. There are two types of n-grams: "allowed" and "forbidden" [2], [10]. An n-gram that does not occur in the use of a particular language by a particular user, or group of users, is said to be forbidden. An n-gram that is not forbidden for a user, or group of users, and is found at least once in a corpus of communications is said to be allowed. Allowed n-grams are the basis of what are generally known as "language statistics" that comprise a large corpus of data used in the attempt to break encrypted messages when the attacker does not possess the key to the file/message. Lewand and others [2], [14], [15] showed how language statistics are employed in breaking intercepted messages. The exact methods for using these statistics varies from cryptographer to cryptographer.

Some approaches use statistics in a limited role. For instance, Shannon used the statistics for each n-gram only once [2]. However, Carlson notes that there are many n-grams available for analysis [10]. For example, for an n-gram of size i, applied to a message of size $|M|$, there are $|M| - i + 1$. n-grams. The total number of n-grams in the message, for a range of n-gram sizes from 2 to j, are n_{mg} n-grams. The total number of n-grams in a message is given by

$$n_{mg} = \sum_{i=2}^{|M|} |M| - i + 1. \tag{6}$$

Single letters are not normally considered in n-gram analysis because there are no forbidden letters in the alphabet. Alphabetic characters are more often seen and used in the role of letter frequency analysis. Larger lengths of n-grams become difficult to work with because the possible number of n-grams for a particular n grows exponentially. For a given n in an alphabet (A), consisting of $|A|$ letters, there are $|A|^n$ possible n-grams. As the size of n increases, the number of forbidden n-grams also increases (Table I). However, the increase in the number of forbidden n-grams quickly overcomes the number of allowed n-grams. The difference is so dramatic that it is easier and more efficient to store the number of allowed n-grams and manipulate that data than to store the forbidden n-grams for use.

TABLE I
n-GRAM NUMBERS FOR ENGLISH

n	No. Forbidden	No. Allowed	Total No. n-grams	% Forbidden
1	0	26	26	0.0000%
2	15	661	676	2.2189%
3	6261	11315	17576	35.6224%
4	347292	109684	456976	75.9979%
5	11251945	629431	11881376	94.7024%
6	306789115	2126661	308915776	99.3116%

Each value for n is a different set of allowed/forbidden n-grams. The possible number of sets that could be applied is given by $|M| - 1$. A difficulty in using sets of allowed/forbidden n-grams is the possibility of biasing the data towards a particular author. As stated earlier, all authors have a "style" [9] that distinguishes one writer from another. An author's style, if used as a set to describe general language use, can inaccurately describe the general use of that language and cause erroneous results. Empirical results indicate that style bias becomes a problem for $m \geq 8$ [15].

The number of n-grams that can be formed, and need to be checked, for a string of n letters is given by l_n, where l is $|A|$ and n is the number of symbols in the string. English uses $|A| = 26$. As n increases, representing each combination as a bit results in increasingly larger number of bits. By the time that $n = 6$, the total number of bits is 308,915,776 bits, or slightly over 38 MB. Because of limitations on the amount of memory available and the effort required to store and retrieve data from such a large file, n-grams larger than $m = 6$ were not considered for use.

Multiple n-grams may reside in a string. For a string of size x, where $x = 6$ letters, the number of n-grams available is given by:

$$num_{(n-grams)} = \sum_{i=2}^{6} x - (i - 1) \tag{7}$$

Combining the concepts of equivalent keys, $H(x)$, R_λ, and n is the idea that inside a file/message, there will be portions of the message where data accumulates very quickly. While most researchers characterize the *average* statistics of a set of messages/files, very little is mentioned about specific cases. Much like gambling, the average will prevail over time due to the Law of Large Numbers [5]. However, even though casinos make a great deal of money on a probability difference of approximately 4%, there are still winners in the casinos. These winners exploit runs of probability that are temporarily in the favor of the gambler. The same happens in messages and files. Understanding that all ciphers are S ciphers [16] and that S ciphers allow for patterns to bleed through [17]. This leads to the concept of an entropy, and hence a unicity distance that is applicable in that portion of the message to break the patterns found there. Known as “local entropy” and “local unicity distance” [10], this explains why cryptographers can exploit patterns found in certain parts of a file/message and recover information that is still applicable to the entire file/message.

All of these concepts are central to understanding why depending on the maximum key space for a cipher does not adequately nor accurately describe the security of encryption for all messages. It is necessary to know how instances of particular messages vary from the average behavior of data encryption.

II. Isomorphic Key Reduction, Isomorphic Sets, and Corpus Constraints

Not all encrypted messages using the same cipher algorithm are equally secure. One of the more interesting questions in cryptanalysis is why some messages are cracked while others remain safe for a much longer period of time. If same cipher is used for any two messages it would seem that both messages should be equally secure and be safe for the maximum number of keys in the key space for that cipher. But this is not true, different messages may have vastly different security and susceptibility to decryption. Shannon’s equation for unicity distance gives clues as to why such a difference in breaking encrypted files and messages exists.

Consider the equation for n (Eq. 5). That equation has three variables of interest: the key space of the cipher ($|k|$), the size of the alphabet of the language ($|A|$), and the redundancy of the language (R_λ). Each of these will be considered in turn.

A. Key Space

The key space of the message typically is given as the maximum key space for the cipher algorithm. No allowance is made for possible heuristic reductions in the number of keys that are examined. For most peer reviewed ciphers it is assumed that the cipher algorithm belongs to those ciphers for which the only real attack is a brute force attack. However, for a large number of ciphers there are attacks that have been identified that reduce the number of operations that need to be done in order to break an encryption. These heuristic attacks range from the use of language statistics [2], [3] on the S cipher to the Slide and Differential attacks [18] on round based product ciphers. Security professionals often rely on the key space measure to select ciphers used to protect data. However, when they do select encryption algorithms these analysts will often select the algorithms based on the maximum key space rather than consider the effect of the decreased effective key space due to message content for known attacks. Compounding this error, cybersecurity experts rarely take the interaction of the message and encryption algorithms into account when deciding on which cipher or security measures to use. The content of the message does affect the choice.

One problem that occurs frequently, especially with smaller messages and files, is the problem of isomorphic (equivalent) keys and their effect on the key space [**?**]. Many times this situation occurs when not all of the letters in an alphabet appear in the message/file. Low frequency letters, such as ‘v,’ ‘k,’ ‘x,’ ‘q,’ ‘j,’ or ‘z’ do not appear in many messages. When they do not appear, then the mapping for those symbols do not matter in an encryption. If two or more symbols are not used, then equivalent keys are possible. These keys can them be grouped into sets of isomorphic keys. Each of these isomorphic keys can be represented by a single key selected from the set, known as the “systematic isomorph.” Since each of the keys will result in the same encryption/decryption, only one key needs to be checked from each isomorphic key set. If the key is rejected, then each of the keys in the set can also be rejected. If the key successfully decrypts the message/file, then any of the keys in the set will have the same effect.

Identifying isomorphic keys in a message or file depends on the patterns in the message. This methodology is limited to encryption algorithms that do not employ randomization routines, such as those found in encryption modes. Modes, such as Cipher Block Chaining (CBC), Counter (CTR), Cipher Feedback (CFB), Output Feedback (OFB), and Propagating CBC (PCBC) are not directly susceptible to this methodology. Use of these modes defeats direct analysis of isomorphic key spaces, but the principle is still applicable to its constituent encryption algorithms.

Since all ciphers are S ciphers [16], at their base and S ciphers do not disguise patterns, it is possible to analyze cipher text and count the number of symbols found in the message. Using the 1:1 principle for encryption [10], it is simple to count the number of unique characters to arrive at the number of alphabetic characters that are used in the message. Using that data, it is then possible to calculate both the number of systematic isomorphs, or unique keys, as well as the size of the isomorphic key sets. In an S cipher the number of systematic isomorphs in the message is simply the number (T) of unique cipher text characters.

In an analogous manner, P cipher key spaces can also be reduced. This technique is built upon the practice to represent an alphabet with encodings that are contiguous within a run of values. For example, in the ASCII encoding, lower case English letters span the decimal values 97 - 122 (61h - 7Ah). Upper case letters span a similar run covering 65 - 90 (41h - 5Ah). If a message is composed of lower case letters, then all characters in a message will begin with the binary values "011." Since the P cipher retains all of the original '0' and '1' bits in the message, the bits that do not change that are known as "static bits" can easily be identified.

Assume that the P cipher used allows mapping bits to any of the bits in the block, even those outside the byte in which the bits originate. Further, for ease of illustration, assume the block is 3 bytes. Now, define a function $\hat{\bigcap}$ that operates on two bits in the same location in different blocks (a and b) such that

$$r = \hat{\bigcap}(a,b) = \begin{cases} b_{a,i} & \text{if } b_{a,i} = b_{b,i} \\ x & \text{otherwise} \end{cases} \tag{8}$$

where x is the symbol x and indicates a difference between the bits. Once a bit location is identified as changing (not static, ie. dynamic) then the bit permanently retains that designation. On the average, if data is random, there is a probability of each bit changing between blocks of data of $pr(b_{a,i} \neq b_{b,i}) \approx \frac{1}{2}$. Therefore, if blocks are sequentially compared using $\hat{\bigcap}$, in approximately $lg(|B|)$ block comparisons between unique blocks, the bits that do not change will be identified because they have not changed. This application of the $\hat{\bigcap}$ function can stop when the number of bits that are not designated by the entry 'x' is equal to the number of known static bits ($|S_t|$) [10]. That is $|S_t| = |B| - |x|$; where $|B|$ is the number of bits in the block and $|x|$ is the number of bits in the block known to be dynamic using this algorithm. Each of these static bits reduce the number of unique mappings for bits and comprise isomorphic keys. Therefore, for a P cipher applied to a message (M) there are

$$(|B| - |S_t|)!\binom{|S_t|}{C} \tag{9}$$

unique keys, where $C = min(|0's|, |1's|)$ [10].

Calculating the number of isomorphs in an isomorphic key sets uses the information collected while determining the number of systematic isomorphs. Again, the values depend on the type of cipher. For an S cipher applied to a message (M), there are $k_e = (|A| - |T|)!$ systematic isomorphic keys [10]. Let T be the set composed of each unique $x_i \in M$. The partial key $T \mapsto A'$ contains all of the information required to decrypt M. Any key containing the partial key $T \mapsto A'$ will correctly decrypt M. The number of symbols that do not appear in the message is given by $|A| - |T|$. Selecting each of the unused symbols and counting the number of mappings for each symbol gives $(|A| - |T|)!$ possibilities.

For a P cipher applied to a message (M), there are

$$k_e = \frac{|B|! - \binom{|S_t|}{C}(|B| - |S_t|)!}{\binom{|S_t|}{C}(|B| - S_t)!} \tag{10}$$

keys in the set of keys for each systematic isomorph.

So long as the cardinality of the set of systematic isomorphs is greater than 1, ie.$|k_e| > 1$. then the resulting key space is smaller than the maximum key space by the factor (f_e) = $\frac{1}{|k_e|}$.

and the security is similarly reduced. A graph of the drop off of the key space versus the number of characters not seen in the message for an S cipher for single letters is shown in Fig. 1. This figure is a log based graph on the Table II. The effect is a log drop as fewer characters are seen. Key space reduction of this type scale with with the size of the block.

B. The Alphabet Size

Alphabets are the collections of the basic symbols that are combined to make words in a language. Each character is required and constitutes a minimal set of characters for a particular language. However, when working with blocks of characters in a language the statistics for the new block language are not a simple scaling up of the original alphabet.

Consider the differences between English when the symbols of the language are taken one at a time versus when the language is broken into blocks of two characters at a time. Shannon called these combinations of blocks of the original language "*n*-grams" where the number substituted for n indicated the size of the block in terms of the number of symbols from the original alphabet. An *n*-gram is a consecutive run on n alphabetic symbols from the text that have spaces and punctuation removed. Shannon used the statistics associated with each of these *n*-grams as important language statistics in breaking ciphers [2]. The probability of a particular *n*-gram appearing in text varies from 0 $\leq$ pr(*n*-gram) $\leq$ 1. If the decryption of a particular set of symbols or block is an *n*-gram has a $pr(n - gram) = 0$ then the proposed decryption key is wrong and is abandoned [2]. Further, Shannon separated *n*-grams into two groups: those which appear in language use ($pr(n - gram) \;\; > \;\; 0$) and those that never appear in the syntax of a language ($pr(n - gram) \; = \; 0$). Those that appear are said to be "allowed" *n*-grams and those that never appear are said to be "forbidden." Shannon also indicated that once a forbidden *n*-gram is encountered, any other larger *n*-gram built upon, or containing the forbidden *n*-gram will also be a forbidden n-gram. Therefore, the percentage of allowed n-grams falls as the size of the block (n) increases. This effect occurs very quickly. Carlson's empirical study of n-gram size showed that by the time $n = 6$ the number of allowed n-grams fell to approximately 0.6114% of the possible combinations of letters in English [10]. In fact, the number of forbidden n-grams exceeded the number of allowed n-grams when $n = 4$.

Carlson, in his dissertation, suggested that when a block of characters is used in encryption that the combination of characters in the block be considered as a "metacharacter" in a "metalanguage" based on the same underlying natural language. The alphabet for the metalanguage is made up only of the allowed n-grams of size $n = |block|$ in the base language. While the size of the alphabet falls greatly, the key space size also falls precipitously. Since all ciphers considered in the paper are S type cipher at heart, their key space is governed by the size of the alphabet. For an S cipher the key space is calculated by $|K| = |A|!$. A reduction in the alphabet size by even a single metacharacter only reduces the size of the metaalphabet by 1, but reduces the key space by A. As an example, consider English as the base language for a metalanguage of size $n = 3$. The total number of possible combinations of three symbols is $|K_{max}| = 26^3 = 17576$. The empirical studies conducted by Carlson showed that only 11,315 3-grams (64.377%) were found in a collection of over 200,000,000 n-grams in the corpus of English texts spanning multiple genre from the 1500s to the present time [10].

Although Carlson focused on an alphabet consisting only of lower case letters in a language, other symbols can be considered to be part of the alphabet. Adding spaces, punctuation, numbers, and upper case letters is a valid way to increase the alphabet size. Allowed metacharacters (n-grams) are still a much smaller set than the total "possible" set if the forbidden n-grams are added into that set. All languages have a syntax that restrict the number of allowed n-grams, since there are patterns in language that bleed through encryption.

The main result is that the key space is based on the actual alphabet size, rather than the maximum size of character combinations for discrete letters. Key space is greatly reduced, as shown in Table III. Therefore, a brute force attack based on the allowed letters/metacharacters in the language can be prosecuted much more quickly than previously thought.

C. Language Redundancy

The third variable in the unicity distance equation is the redundancy of the language (R_λ). Most cryptographers, when evaluating the unicity distance, use the standard values calculated for a natural language. For example, the accepted value for the redundancy of English was calculated by Shannon [11] to be $R_{English} \approx .75$. This measure is related to the *average* data for the language. However, each individual actually uses a language differently from other users. Langendoen and Postal, in their work on the Theory of the Vastness of Natural Languages [19] indicate that each person has their own *unique* language (λ_i) that *intersects* with other personal languages and results in a mutually understood natural language, such

TABLE II
KEY SPACE VS UNSEEN CHARACTERS

|A|	Unseen	|K|
26	0	4.03291E+26
26	1	4.03291E+26
26	2	2.01646E+26
26	3	6.72152E+25
26	4	1.68038E+25
26	5	3.36076E+24
26	6	5.60127E+23
26	7	8.00181E+22
26	8	1.00023E+22
26	9	1.11136E+21
26	10	1.11136E+20
26	11	1.01033E+19
26	12	8.41942E+17
26	13	6.47648E+16
26	14	4.62605E+15
26	15	3.08404E+14
26	16	1.92752E+13
26	17	1.13384E+12
26	18	62990928000
26	19	3315312000
26	20	165765600
26	21	7893600
26	22	358800
26	23	15600
26	24	650
26	25	26
26	26	1

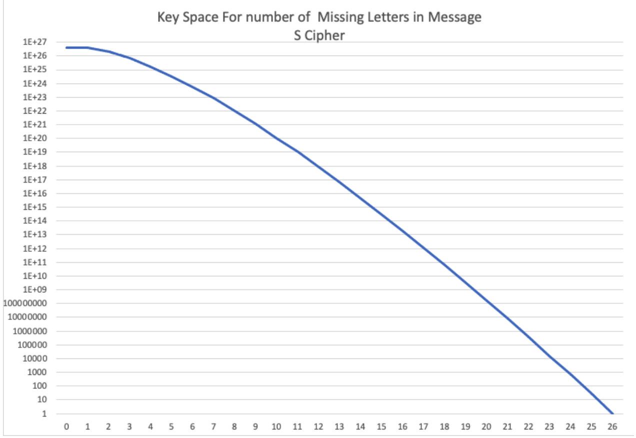

Fig. 1. Key Space vs. Unseen Characters

TABLE III
EFFECTIVE KEY SPACE FOR S CIPHERS COMPARING MAXIMUM ALPHABET TO ALLOWED ALPHABET

N	A^N	Allowed A^N	Max Key Space	Actual Key Space
1	26	26	26!	26!
2	676	661	676!	661!
3	17576	11315	17576!	11315!
4	456976	109684	456976!	109684!
5	11881376	629431	11881376!	629431!
6	308915776	2126661	308915776!	2126661!

as English. They indicate that there are a transfinite number of human natural languages $|\lambda| = \aleph_0$. This indicates that each person has a unique $H(x)$, R_λ, and n distance. Some users will naturally have more security related to their messages due to the habits that they routinely use in communication. Some will also have increased security related to the lexicon and how that lexicon is used in speaking or writing. Unicity distance is directly related to redundancy and redundancy as a measure of stylometry [9]. In general, different communicators (speakers or authors) will have a different probability density function (pdf) for a language. Each language (λ_i) must be considered separately when selecting an encryption for a message.

The individual habits of a user will also be reflected in shorter portions of a message or file. The "local" environment of a portion of the message or file will also have susceptibility to attacks. That susceptibility depends directly on the habits and stylometry of the individual user. Any data that is gained from work in a local portion of the message and file becomes side information that can be applied throughout the entire message or file, changing the local entropy and unicity distance for the remainder of the message.

III. EFFECT OF REDUCTION AND ISOMORPHIC KEYS ON DATA SECURITY

In the previous section of this paper, it has been shown that key space, the use of the alphabet in a language, and the style of a user all mathematically bound the effective unicity distance for an encrypted message or file. This brings up two questions: how do these named factors affect cybersecurity and the security of any particular message and is it possible to draw conclusions about the nature and practice of cryptography? The answer to the first question about the use of language by an individual user in a message clearly impacts the key space, and hence the security, of the message. Both the habits of the user and the content of the message can radically vary the key space from the mathematical bounding imposed by the maximum key space of the encryption algorithm employed by the system. Therefore, the message should be evaluated prior to the selection of the encryption for any message. The answer to the second message is an an unequivocal yes, it is possible to generalize and draw conclusions about how the message affects security.

The first lesson that should be drawn from this data is that the key space quoted for a cipher should be seen as a maximum bound. So, for a message (M), the key space is dependent on the characters seen in the message. Let $K_{M,c}$ represent the key space for the message using cipher (c) and K_c represent the maximum key space for a cipher (c). Then $|K_{M,c}| \leq |K_c|$. The unicity distance for a message is

$$n_{M,c} = \frac{log(|K_{M,c}|)}{R_\lambda log(|A|)}. \tag{11}$$

And by extension

$$n_{M,c} \leq n_c = \frac{log(|K|)}{R_\lambda log(|A|)}. \tag{12}$$

Assessing how much of a difference that there is between the actual key space and unicity distance can vary greatly. At times the difference is so great that messages and files are susceptible to being recovered by attackers with much less effort that the sender realizes. Instead of relying solely on the key space as a measure of security the determination of how secure a message is when encrypted depends on *both* the cipher algorithm used for encryption *and* the exact content of the individual message or file. In turn, the content of the message depends on the habits of the user and indirectly on the subject matter and the way it is presented. Security and the measures taken should depend on the individual message.

The approach taken in evaluating the key space and related variables in the calculation of the unicity distance explains a number of observed phenomena in encryption. It explains why, even with the same cipher, some messages will be easier to crack than other messages. It has been assumed that the same cipher means the same level of protection. It does not. A smaller key space and less characters in an alphabet result in a shorter, sometimes critically so, unicity distance. Brute force attacks will also be easier for some messages than they will be for other messages using the same cipher for encryption. Rainbow tables may even be possible for attacking even what are thought of as the most secure and strong ciphers. It all depends on the content of the message or file that is being encrypted.

Personal habits and stylometry also affect the unicity distance of an encrypted message. Examples of personal language habits are the use of articles and repeated phrases in texts. Morton identified more than 30 such habits in his text on the subject [9]. Stylometry issues can also reduce the key space and the number of alphabetic characters that appear in a message. Security always depends on the user, the message, and the cipher.

Ciphers encrypting large blocks of data will increase the security of a message, but they are also susceptible to the same problems as a cipher encrypting a single character at a time. By viewing the block as a metacharacter in a metalanguage, analysis can be scaled for use to these block ciphers. Therefore, block ciphers must also be applied to messages in light of the content of the messages.

IV. Conclusion

In this paper it has been shown that the use of the maximum key space calculation, when used as a measure of security, has been misunderstood and misapplied. Shannon's equations relating to the security of encrypted files and messages clearly shows that a "one size fits all" analysis of key space is incorrect. Key spaces are often much smaller than the maximum and smaller than assumed. The sources of variation from the maximum can come from, as a minimum, the following:

1) Isomorphic (equivalent) keys,
2) Smaller size of the alphabet due to forbidden n-grams,
3) The use of metacharacter analysis of a block encrypted message,
4) Local entropy, redundancy, and unicity distance in a message,
5) The effects of syntax and the semantic content of a message, and
6) The effects of user(s) stylometry in the actual content of the message or file.

Each of these sources of reduced entropy add up to reduced security. A user cannot take the maximum size of key space, and therefore the maximum security, for granted. When choosing the method of transferring files, encryption algorithm, and security measures for communications the cybersecurity professional must take into account the content of the message. Using the maximum key space in selecting a cipher and applying security scheme should never be used as a shortcut for setting the level of security and taking the proper security measures.

As the key space is a message which are often smaller than assumed, sometimes very small, messages should be evaluated before being sent. It may even be possible to use this assessment to select the best cipher and security measures for a transmitted message or file. When cybersecurity professionals understand the variation in file security they can then take a more informed set of measures to properly protect the data they release onto networks for transmission.

This paper has given guidelines for properly evaluating the security of an encrypted message. In particular, the role played by message content has been highlighted, along with language usage in communications. Examples were also provided to demonstrate the concepts presented, including how seemingly large key spaces are really much smaller than typically quoted. As a result, the cybersecurity expert must consider, and adjust for, message content in order to properly assess message security and select the proper measures to maximize message safety.

References

[1] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5 – 83, 161 – 191, 1883.

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

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

[4] Matthew Bishop. *Computer Security: Art and Science*. Addison-Wesley Professional, Boston, 2003.

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

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

[7] B. Ghosh, I. Dutta, S. Khare, A. Carlson, and M. Totaro. Isomorphic cipher reduction. In *The 12th IEEE Annual Information Technology, Electronics, and Mobile Communication Conference*, 2021. Manuscript Submitted for Publication.

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

[9] Andrew Morton. *Literary Detection*. Scribners, New York, 1978.

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

[11] Claude Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 30:50 – 64, 1951.

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

[13] A. Carlson, B. Ghosh, I. Dutta, S. Khare, , and M. Totaro. Key space reduction using isomorphs. In *The 12th IEEE Annual Information Technology, Electronics, and Mobile Communication Conference*, 2021. Manuscript Submitted for Publication.

[14] Robert Lewand. *Cryptological Mathematics*. Mathematical Association of America, Washington D.C., 2000.

[15] Albert Carlson and Robert Hiromoto. Using set theoretic estimation to implement shannon secrecy theory. In *The Proceedings of the Third IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 435 – 438, 2005.

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

[17] Thomas Barr. *Invitation to Cryptography*. Prentice Hall, Upper Saddle River, 2002.

[18] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *Advances in Cryptology – CRYPTO '97, Lecture Notes in Computer Science*, pages 513–525, 1997.

[19] D. Terence Langendoen and Paul Postal. *The Vastness of Natural Languages*. The Camelot Press, Ltd., Southampton, 1984.