[Albert Carlson et al. (Oct. 27-30, 2021). Key Space Reduction Using Isomorphs, 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). Source: https://ieeexplore.ieee.org/document/9623158]

Key Space Reduction Using Isomorphs

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

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

Shivanjali Khare
ECECS
University of New Haven
Connecticut, USA
skhare@newhaven.edu

Michael Totaro
CMIX
University of Louisiana at Lafayette
Louisiana, USA
michael.totaro@louisiana.edu

***Abstract*—Block ciphers are said to have key spaces that are large enough to prevent a brute force attack from breaking them within the lifetime of the attacker; however, messages obscured using those ciphers are regularly broken. Some are broken because of increased computer capabilities, while others are broken because of industry reliance on invalid assumptions. These assumptions include the idea that all encrypted blocks of text are equally likely to appear in a message and that only the original key can decrypt a message. In this paper, we show that information theory techniques using isomorphs can reduce the key space to a size that makes the subsequent brute force attack possible in a shorter and easily achievable time frame.**

***Index Terms*—Block Ciphers, Encryption, Key Spaces**

I. Introduction

A classical algorithmic problem has been a topic of inquiry since the beginning of computing: namely, determining whether two graphs are structurally identical or isomorphic. An extensive range of applications can be found on this problem ranging from chemistry to computer vision. An issue closely related to this is that of detecting symmetries in a graph or general combinatorial structures. Logic, algorithmic group theory, and quantum computing are areas that are of interest on the more theoretical side.

In cryptography, the decryption technique that is guaranteed to succeed every time is to systematically to apply every key in the key space [1], which is widely accepted. A key space is the set of all possible key values that a particular enciphering algorithm admits. Once the correct key is tried, the original message is recovered. This attack is known as a "brute force" attack because no heuristics or information learned during each decryption attempt are used to guide key selection. Brute force attacks are slow and inefficient [2]. Using equation 1, the average number of keys from the key space (K) that must be attempted before recovering the key (k_a) is given by [3].

$$k_a = \frac{|K|}{2} \tag{1}$$

978-1-6654-0066-4/21/$31.00 ©2021 IEEE

This paper challenges the argument of "statistical improbability" and mathematically demonstrates that the key solution space is not as large as currently held. This reduction in the key space is due to the existence of equivalent keys, which give rise to "isomorphs" [4].

II. Background

Modern ciphers use techniques that are assumed to safeguard what can be used to eliminate potential keys from the overall key space. These ciphers also assume that the message itself offers no clues as to the key. If these conditions are met, the only effective attack left to the hacker is a brute force attack. When using these techniques, the key space becomes a measure of cipher strength. Using equation 2, the larger the key space, the stronger the cipher.

$$Security \propto |K| \tag{2}$$

Modern ciphers have large combinatoric key spaces [5], so that extensive effort is expended in brute force attacks and it is statistically improbable to break the cipher in a reasonable time. As a consequence, all messages using modern ciphers should be theoretically secure; however, messages encrypted using these ciphers have been, and are, readily broken. For example, Bernstein [2] describes a verified implementation of an attack against the reduced Fast Syndrome-based hash functions [6]. Several cryptanalyic algorithms have also been executed on FPGAs such as using the COPACOBANA machine [7], which includes exhaustive key-search for DES or solving systems or solving systems of multivariate quadratic equations.

There are two possible reasons for these breaks:

- The solution space is not as large as previously thought and/or
- The ciphers are susceptible to heuristic attacks.

Section III, presents a detailed and extended discussion on equivalent keys, isomorphic reduction, and complexity of isomorphic reduction vs. brute force. This section also

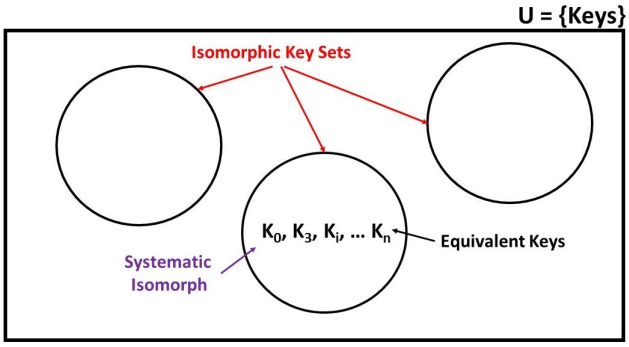

Fig. 1. Illustration of Isomorphic Keys and Their Relationships to Each Other

provides the proposed theorems and corollary. Section IV presents an example to illustrate the proposed isomorphic key space reduction. Finally, in Section V, we discuss our conclusions and future research questions.

III. Discussion

A. Equivalent Keys

All ciphers are designed to exhibit a one-to-one mapping between plain text and cipher text. Each key maps the input message, M, to a unique cipher text encryption $E_k(M)$. Ideally each symbol in alphabet A is used at least once in the message in order to force every mapping to be solved. Let a "block" be a group of $|B|$ continuous characters treated as a single unit (equivalent to Shannon's definition of an *n*-gram [8]) for encryption and decryption. Most modern ciphers deal with blocks in the message at the same time, encrypting them together instead of encrypting each symbol in the language on its own. This type of cipher is known as a block cipher [9], [10] and is used to complicate breaking the cipher. However, there are cases of finite length messages when several keys can yield the same decryption given the same input message. In these cases $E_{c,k_i}(M) = E_{c,k_j}(M)$, where $i \neq j$. Two keys (k_i and k_j) are considered equivalent for a particular message and cipher if

$$D_{c,k_i}(M) = D_{c,k_j}(M) \tag{3}$$

(see Fig. 2). The existence of equivalent keys implies that a decryption solution does not necessarily return the original mapping for all letters $\in A$. For any cipher, the maximum key space consists of all of the symbol permutations that make up key mappings.

Two variables (iso_i and iso_j where $i \neq j$) are said to be isomorphs of a function ($f(x)$) if the variables are related in some manner to each other and

$$f(iso_i) = f(iso_j) \tag{4}$$

There may be many isomorphs for a function. If those isomorphs are gathered together in a set, the set may be represented by a single member of the set, called the "systematic isomorph." When applied to the key (k) of an encryption function ($E_{c,k}$) isomorphs result in the same cipher text for a message [11] for each isomorph. Isomorphs are keys that, when applied to encryption or decryption of the same cipher, result in the same encryption or decryption of the same message. Each equivalent key is an isomorph in the key space of the cipher.

As an example of the impact of isomorphs, consider an alphabet $A = \{a, b, c, d\}$. Further, let a substitution cipher that maps $A \mapsto A$ be applied to a message. In this case, the key space is $4! = 24$ keys [12]. For the message $M = abbbaba$, applying the key $k = \{a, b, c, d\} \mapsto \{b, a, d, c\}$ results in the encrypted message $E_k(M) = baaababa$. Multiple isomorphs exist in the key space for this message. For example, the keys {b,a,c,d} and {b,a,d,c} will result in exactly the same encryption and decryption, and are therefore isomorphs. Twelve such isomorphic sets exist for this message.

A simple example does not address ciphers that are more complex than a substitution cipher or block ciphers. The alphabets of block ciphers are similar. It can be shown that all known ciphers, including block cipher, are ultimately substitution ciphers [13].

Substitution cipher keys are actually mappings from plain text (PT) to cipher text (CT), which can be denoted for each character (i) by $PT_i \mapsto CT_i$. Ciphers may encrypt by operating on a single character in a message or operate on a block of characters. If each block of characters is considered to be a character composed of a group of language characters, or a "metacharacter," of n characters (denoted as "meta*n*character" where $n \in \{2 \leq n < \aleph_0\}$), then a key for any size block becomes possible [4]. A metacharacter comprises of a symbol of the matalanguage alphabet. A metalanguage is a language that is formed from a base language but uses blocks of letters that has the same mix of single characters. Therefore, all ciphers have the same susceptibility to isomorphic reduction.

B. Isomorphic Reduction

In substitution ciphers, if not all of the characters (or meta*n*characters) in the alphabet are used, then the existence of equivalent keys is possible [4]. Each group of isomorphs can be replaced by a systematic isomorph chosen from the sets of equivalent keys. The key space then reduces to the number of systematic isomorphs for a brute force attack. We call this elimination of isomorphic keys "isomorphic reduction." The reduction factor (R) for s sets is

$$R = \frac{1}{|s|} \tag{5}$$

The use of isomorph reduction can also be used on parts of the message as well as on the entire message. Assume that a CT message is partially decrypted with the unencrypted cipher text characters denoted by "*"'s. A segment of the message reads "$iam*egend$." The remaining isomorphs in the key space with mappings that decrypt the specific section of code with any value for the unknown character are assembled for a brute force check. Only the letter "l" makes sense in this section of code, requiring a mapping $* \rightarrow l$. All other isomorphs can then be eliminated.

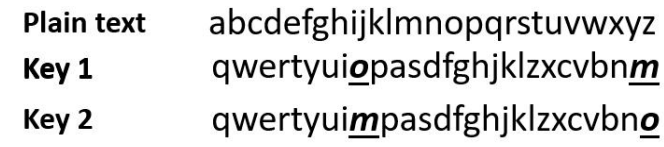

Fig. 2. Illustration of Two Isomorphic (Equivalent) Keys in English

Let M be a message composed of symbols $x_0, x_1,...x_n$ in a language whose alphabet is A. There are $|A|$ symbols in the alphabet and $\forall x_i$, $x_i \in A$. Further, let T be composed of all of the unique $x_i \in M$ and the cipher text alphabet be represented by A'.

We recall the following theorem from Carlson [4].

Theorem : *For a S cipher applied to a message, M, there are $(|A| - |T|)!$ equivalent keys.*

Proof: For two keys, k_i and k_j to be equivalent for a message, M,

$$E_{k_i}(M) = E_{k_j}(M) \rightarrow D_{k_i}(E_{k_j}(M)) = D_{k_j}(E_{k_i}(M))$$

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.

To illustrate further, consider an alphabet $A \mapsto A$ (a monoalphabetic mapping) using an S cipher. Further, let A = $\{0,1,2,3,4\}$, and a message M = $\{11212112\}$. In this example, $|A| = 5$ and the number of mapped keys $|T| = 2$. Of the five symbols in the alphabet, two are mapped. The mappings of the remainder of the symbols are irrelevant to the decryption of the message. Assuming that the mappings for the characters in the message are the characters $c_0 \mapsto$ '1' and $c_1 \mapsto$ '2', then the equivalent keys that correctly decrypt the message (M) are:

$$\{c_2, c_0, c_1, c_3, c_4\}$$
$$\{c_2, c_0, c_1, c_4, c_3\}$$
$$\{c_3, c_0, c_1, c_2, c_4\}$$
$$\{c_3, c_0, c_1, c_4, c_2\}$$
$$\{c_4, c_0, c_1, c_2, c_3\}$$
$$\{c_4, c_0, c_1, c_3, c_2\}$$

or 6 keys rather than 120 keys in the keys space.

This theorem can be deduced from the isomorph [4] (equivalent key) argument, which is as follows:

Let $x, y \in A$. Let x be a plain text character and y be a cipher text character. Without loss of generality, let $x, y \in \{0, ..., |A| - 1\}$. A substitution cipher with key k is an encryption such that $\forall x_i \in A$, $\exists! y_i \in A$ such that $y_i = x_i + k_i \ mod \ |A|$. $i \neq j$ implies that $y_j = x_j + k_j \ mod \ |A|$ is such that $y_j \neq y_i, x_j \neq x_i$, and

$$k_j \neq k_i, k_i, k_j \in \{0, ..., |A| - 1\} \ \forall \ y, x, k \in \{0, ..., |A| - 1\}$$

Let M be a message composed of letters $x \in A$ such that $\{x \in M\} \subseteq A$. Let this set $\{x \in M\} = T$. Without loss of generality, enumerate the $x_i \in T$ such that $i < j$ implies x_i first appears in M prior to the first appearance of x_j, $j \neq i$. Let $m = |T|$ and $n \geq |A|$. Then we can write the enumerated set T as

$$T = \{x_1, ..., x_m\}$$

For a substitution cipher with key k, we then have the enumerated cipher text messages $T' = \{y_1, ..., y_m\}$ with $y_i = x_i + k_i \ mod \ |A| \ \forall \ x_i \in T$ and with $k_i \neq k_j$ if $i \neq j$. Clearly $(|T| = |T'| = m) \leq n$. The substitution cipher over M is then defined by $k = \{k_0, ..., k_n\}$ where $i < j \Rightarrow$ substitution k_i; it first occurs prior to the first occurrence of substitution k_j in the encryption of M. k can now be described as a tree. Given (x_i, y_i), k_i is specified. There are now $|A| - 1$ unspecified k_i remaining in k and the total number of possible specifications remaining is $(|A| - 1)!$.

Now given (x_2, y_2), k_2 is also specified. There are now $|A| - 2$ unspecified k_i remaining in k and the total number of possible remaining specifications remaining is $(|A| - 2)!$. By induction, after the n^{th} pair (x_n, y_n) and their specified k_n are given, there remain $k - n$ unspecified substitutions and the possible specifications is $(|A| - n)!$ But, $n = T$, therefore, the number of isomorphic keys that encrypt M into the same cipher text y is

$$|k| = (|A| - |T|)!$$

Equivalent keys, or isomorphs, set an upper bound on the key space for a substitution cipher, eliminating all but the systematic isomorph for each set of isomorphs. Any substitution cipher has a key space which can be limited using isomorphs. Permutation ciphers (P) are a special case of a substitution cipher in which the bits in a letter, or symbol, are reordered. Any mapping which does not retain the same number of '0' and '1' bits in the symbol are impossible. If a block (multi-letter) cipher is used those bits may be spread across the entire block. However, for a block permutation if message is treated as if the block is a single metacharacter the data is retained in the metacharacter and the data is still limited to the same symbol. This allows using the same procedure to place an upper bound on a permutation cipher key space.

The number of equivalent keys in a P cipher also depends on the characters found in the message. Let B be the block of letters on which a P cipher is applied. $|B|$ is the number of bits being permuted with S_t being the static bits in the block. Static bits are bits whose plain text value never changes in the encoding of the plain text letter in an electronic representation, such as ASCII. ASCII encoded lower case letters all begin with the most significant bits '110,' followed by the specific bits for each letter. The permutation mapping is the same for each block. Static bits will be mapped to the same location in the encrypted byte, and because they are static, the encrypted bits are also static - unchanged in all blocks of the CT. Unchanging bits can be exploited and are easily identified.

Let the number of static ones in a block of M be represented by $|1's|$ and the number of static zeros be represented by $|0's|$. Then let $C = min(|1's|, |0's|)$, giving the size of the least represented value of the static bits. For example, assume that a P cipher is applied to a message comprised exclusive of blocks consisting of ASCII encoded letters (no numbers, spaces or punctuation) encrypted in three letter blocks. In this case there will be 9 static bits (3 blocks with three '110' patterns), six '1' bits and three '0' bits in each block. In this case, $C = 3$, the number of the '0' static bits in the block.

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

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

unique keys.

Proof: Let the permutation matrix k be formed with size $|B| \times |B|$. There are then $|B|$ choices for placement of the '1' term in the first row of the matrix. In the second row of the matrix, the '1' cannot be placed in the same column as that in the first row of the matrix. Therefore, the number of choices remaining is $|B| - 1$. For the third row of the matrix, the number of choices is similarly $|B|-2$. Therefore, by induction, the number of permutation matrices is

$$|k| = |B| \times (|B| - 1) \times (|B| - 2) \times ... \times (2) \times (1) = |B|!$$

Now assume the plain text block being encoded contains $|S_t|$ static bits. As these bits make no contribution to the entropy in the cipher text, the remaining encrypted block is equivalent to a permutation cipher applied to a block of $|B| - |S_t|$ bits. Thus the isomorphic key subspace contains $|k'| = (|B|-|S_t|)!$ keys.

Within the original plain text vector, all distributions of static bits are isomorphic to a systematic vector $\bar{x_s}$ containing $|1's|$ '1' bits as its first entries and $|0's|$ '0' bits as its next entries. Denote this subvector of static bits ($\bar{s}$) as $\bar{s} = \{1...10...0\}$. For example, consider a non-ASCII encoding where $|S_t| = 5$ and $|1's| = 3$ then $\bar{s} = \{11100\}$ and $C = \min(|1's|, |0's|) = 2$. The number of isomorphic permutations of $\bar{s}$ is found by rearranging the locations of the '0' bits by exchanging their positions with the '1' bits, e.g.

11100	11001	10011	01011
11010	10101	00111	
10110	01101		
01110			

Note that $\binom{5}{2} = \frac{5!}{3!2!} = 10$, the number of isomorphic permutations just illustrated. In general, the number of isomorphic permutations of $\bar{s}$ plain text vectors is $\binom{|S_t|}{C}$.

Let $\bar{y_s}$ be the isomorph cipher text obtained from the isomorph plain text $\bar{x} = (\bar{s} : \bar{x'})$. Then

$$\bar{y_s} = \bar{x} \begin{vmatrix} I & 0 \\ 0 & k'_{22} \end{vmatrix}$$

where I is $|S_t| \times |S_t|$, and k'_{22} is $(|B| - |S_t|) \times (|B| - |S_t|)$. Then $\bar{y_s} = (\bar{s} : \bar{x'_s}\bar{k'_{22}})$ where $\bar{x'_s}$ is the non-static subvector of $\bar{x_s}$. All possible cipher texts are isomorphic to $\bar{y_s}$ and the cardinality of this set is equal to the product of the informative submappings $\bar{x'}\bar{k'_{22}}$ and the number of isomorphic transformations on $\bar{x_s}$ [14]. Therefore, the number of unique keys is

$$k = (|B| - |S_t|)!\binom{|S_t|}{C}$$

Corollary i: *For a P cipher applied to a message M, there are*

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

spurious, or "image," keys.

Proof: The size of the total key space universe is $|B|!$ By Theorem 1 within this universe the number of unique isomorph keys is $(|B| - |S_t|)!\binom{|S_t|}{C}$. Therefore, the number of image keys is

$$\text{|key space universe|} - \text{|isomorphic key subspaces|} = |B|! - (|B| - |S_t|)!\binom{|S_t|}{C}. \quad (6)$$

Therefore the number of image keys is

$$\begin{aligned} k_e &= \frac{\text{|keyspace universe|}}{\text{|isomorphic key subspaces|}} - 1 \\ &= \frac{\text{|keyspace universe|} - \text{|isomorphic key subspaces|}}{\text{|isomorphic key subspaces|}} \\ &= \frac{|B|! - \binom{|S_t|}{C}(|B| - |S_t|)!}{\binom{|S_t|}{C}(|B| - |S_t|)} \end{aligned}$$

For the message being decrypted, the key space can be replaced by the set of systematic isomorphs

$$|K_M| = \text{|systematic isomorphs|} \quad (7)$$

and

$$\forall M \rightarrow |K_M| \leq |K_c| \quad (8)$$

Further, if the message is not as large as the alphabet ($|M| < |A|$), or if the number of blocks (B) is less than the number of metacharacters that can be constructed for the blocks ($B < |A|^n$), then equivalent keys must exist. Even if $B > |A|^n$, redundancy reduces the number of unique blocks or alphabetic characters seen, making it more likely that a message will have equivalent keys. A message of the size shown in Table I is the minimum size of a message that can avoid isomorphs since the file size must exceed the alphabet size in order for each character to appear in the message. A much larger message will typically be required due to language redundancy and the effect is exacerbated with increasing key size.

TABLE I
KEYS FOR A GIVEN BLOCK SIZE

Key Size (bits)	Block Size (bytes)	Alphabet Size
8	1	26
16	2	676
24	3	17576
32	4	456976
40	5	11881376
48	6	308915776
56	7	8031810176
64	8	209×10^9
72	9	5.43×10^{12}
80	10	141×10^{12}

The key space for a message does not have to be identical to that of the key space for the language and cipher in general. Each message must be evaluated on an individual basis taking into account the cipher text seen in the encrypted message. Messages of identical lengths may have vastly different information content. As a result, one message may be subject to decryption while another with similar size but different content may not reveal enough information to be decrypted.

C. Complexity of Isomorph Reduction vs. Brute Force

The brute force attack [9], [15] is known to be of complexity

$$O(n) = \frac{C}{2}|K| \tag{9}$$

Isomorph reduction reduces the key space to $|s|$ before any heuristic algorithms are applied. The complexity of isomorph reduction then becomes

$$O(n) = \frac{C}{2}|s| \tag{10}$$

Isomorphs can vary between $1 \leq |s| \leq |K|$, depending on the content of the message and the cipher used for encryption. Comparing the complexity of a brute force attack using isomorph reduction to a brute force attack yields a reduction of complexity (Υ_r) of

$$\Upsilon_r = \frac{|s|}{|K|} \tag{11}$$

In all cases, $0 < \Upsilon \leq 1$ because $|s| \leq |K|$. In most languages, the number of allowable combinations of letters is far below the number of possible letter combinations, ensuring that for block ciphers there is a significant difference between $|s|$ and $|K|$. Smaller messages and messages with more repetition will have fewer systematic isomorphs and a smaller key space after isomorph reduction.

IV. EXAMPLE

To illustrate the isomorphic key space reduction, assume that a message submitted for decryption is an English language plain text message, where $|M|$ = 1000 characters, $|B|$ = 6 bytes (48 bits), and $|T| = 120$ unique characters are found in the message.

A block of 6 bytes results in

$$|A|^{|B|} = 26^6 = 308,915,776 \tag{12}$$

possible combinations of plain text to cipher text mappings. For a 6 byte block, the number of allowed 6-grams, the six block alphabet ($|A_{6,\text{English}}|$), is

$$|A_{6,\text{English}}| = 92,674 \tag{13}$$

meta6characters. With 120 unique meta6characters, a total of

$$(|A_{6,\text{English}}| - |T|)! = (92,674 - 120)! = 92,524! \tag{14}$$

possible isomorphic keys exist. For a 48 bit key, there are

$$2^{|B|*8} = 2^{48} = 2.82 \times 10^{14} \tag{15}$$

possible mappings for each meta6character. Since the number of possible keys is lower than the full number of combinations, the number of possible keys can be examined more quickly than the number of blocks. Therefore, a comparison of efficiency will involve the keys for the block rather than total blocks. The possible number of mappings is reduced significantly and the time required to test is similarly reduced ($92524! << 2.81 \times 10^{14}!$). This number represents the upper bound of mappings for a brute force attack on the encryption. The effect is a much smaller key space to check, enabling decryption even of highly complicated ciphers. This analysis demonstrates that complicated obscuring does not necessarily mean effective obscuring.

V. CONCLUSION

The goal of decryption is to recover the original message; it is not to recover the exact key that created the encryption. Modern encryption systems are "broken" (so to speak) because the key space typically is much smaller than expected. A simple analysis of a cipher's key space is to calculate all possible mappings. If every possible block (meta*n*character) is found in a message, then key space would be an effective measure of cipher strength; however, most messages are not long enough to justify this assumption. Messages that do not contain every character in an alphabet have equivalent keys, or isomorphs, in their key space.

In this paper, we have shown that, when there are isomorphs in the key space, the key space collapses as a result of the isomorphic reduction. Sets of isomorphic keys can be replaced by the systematic isomorph to represent multiple keys with a single key. This technique can be used on all or part of the entire message in order to speed decryption. The number of isomorphs, as well as the number of unique keys, can be calculated just by examining the cipher text of a message. Isomorphs also work on portions of a message. If a portion of the message has a combination of plain text and cipher text, then isomorphic reduction can be applied to the key space.

Key spaces depend upon the alphabet of the message language. For block ciphers, the alphabet is the collection of meta*n*characters of the language. Not all combination of characters in the message language are found in the collection of meta*n*characters. Restricting the key space to

those meta$n$characters and then applying isomorphic reduction constricts that key space. Calculations of key space size often incorrectly assumes that all block combinations are possible, making the perceived key space larger than it actually is in reality. As for future work, we will look at different languages to find out allowed vs. forbidden rules. This leads us to ask if it is the same? Or if it is due to some tactic rules or semantic content?

By increasing the number of metacharacters, we theoretically increase the security because there are more members of the alphabet. With larger alphabet sizes, it will change the log of the keyspace vs. the log of the size of the alphabet in the unicity key distance. The log of K goes up linearly as:

$$n = C\frac{log|K|}{log|A|} \tag{16}$$

But, the size of K goes up by a factorial, so $A \uparrow, K \uparrow$ as k' changes the equation to:

$$n = C\frac{log|A'|}{log|A|} \tag{17}$$

but the limit is as shown in Fig.1

Further studies of metalanguages are needed in order to ascertain issues relating to "allowed vs. forbidden." The key space is restricted to the allowed meta$n$characters of the block size of the encryption, and is thereby subjected to isomorphic reduction of the keys space. The effect of isomorphic reduction is to make the key space small enough for a brute force attack to return the original message in time to use that message. Cryptographers should re-evaluate security with information content and isomorphic reduction as factors in message security. The point of isomorphic reduction is this: multiplying significantly the key space does not necessarily increase security. It is possible to recover messages encrypted, even in the most complicated ciphers thought to have overwhelmingly large key spaces if the message allows for isomorphic reduction.

REFERENCES

[1] Messaoud Benantar. *Introduction to the public key infrastructure for the internet*. Prentice Hall, 2002.

[2] Daniel J Bernstein. Understanding brute force. In *Workshop Record of ECRYPT STVL Workshop on Symmetric Key Encryption, eSTREAM report*, volume 36, page 2005. Citeseer, 2005.

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

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

[5] KG Srinivasa, V Poornima, V Archana, C Reshma, KR Venugopal, and LM Patnaik. Combinatorial approach to key generation using multiple key spaces for wireless sensor networks. In *2008 16th International Conference on Advanced Computing and Communications*, pages 279–284. IEEE, 2008.

[6] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In *International Conference on Cryptology in Malaysia*, pages 64–83. Springer, 2005.

[7] Sandeep Kumar, Christof Paar, Jan Pelzl, Gerd Pfeiffer, and Manfred Schimmler. Breaking ciphers with copacobana–a cost-optimized parallel code breaker. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 101–118. Springer, 2006.

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

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

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

[11] Petteri Kaski and Pateric R. J. Ostergård. *Classification Algorithms for Codes and Designs*. Springer, 2006.

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

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

[14] Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra, Vectors, Matrices, and Least Squares*. Cambridge University Press, 2018.

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