[Dr. Albert H. Carlson, Dr. Robert E. Hiromoto. (Jun. 21, 2013). Reduction of Key Space Using Isomorphs and Language Characteristics. Cloud+MD. Reproduced for educational purposes only. Fair Use relied upon.]

Reduction of Key Space Using Isomorphs and Language Characteristics

Dr. Albert H. Carlson
Cloud-MDs
Henderson, NV 89011

Dr. Robert E. Hiromoto
Computer Science Dept.
University of Idaho
Moscow, ID 84844

June 21, 2013

Abstract

Block ciphers are said to have key spaces large enough to prevent a brute force attack from breaking them inside the lifetime of the attacker. However, messages obscured using those ciphers are regularly broken. Some are broken because of increased computer capabilities, but 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 it possible the subsequent brute force attack possible in a shorter and easily achievable time frame.

1 Info/Background

The most inefficient professionally accepted decryption technique, but the only one guaranteed to succeed, is to systematically try every key in the key space. Once the correct key is tried, the original message is recovered. This attack is known as the "brute force" attack because no heuristics or information learned during each decryption attempt are used to guide key selection. Brute force attacks are slow. The average number of keys from the key space (K) that must be attempted before recovering the key (k_a) is given by

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

Modern ciphers use techniques that are assumed to safeguard which can be used to eliminate potential keys from the overall key space. These ciphers also assume that the message itself gives 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. The larger the key space, the stronger the cipher.

Modern ciphers have large combinatoric key spaces, 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. But messages encrypted using these ciphers have been, and are, readily broken. 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. This paper will contest the argument of "statistical impossibility" and mathematically demonstrates that the key solution space is not as large as currently believed. This reduction in the key space is due to the existence of equivalent keys, that give rise to "isomorphs."

2 Discussion

2.1 Equivalent Keys

All ciphers are designed to exhibit a one-to-one mapping between PT and CT. Each key maps the input message, M, to a unique cipher text encryption $E_k(M)$. Ideally each symbol in the language 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 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 [4] and is used to complicate breaking the cipher. However, there are cases 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

$$D_{c,k_i}(M) = D_{c,k_j}(M) \tag{2}$$

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 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

$$f(iso_i) = f(iso_j) \tag{3}$$

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 [1] 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. 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.

This simple example does not address ciphers that are more complex than a substitution cipher or block ciphers. It can be shown that all known ciphers, including block cipher, are ultimately substitution ciphers [2]. 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 $n \in \{2 \leq n \leq \aleph_0\}$), then a key for any size block becomes possible [3]. Therefore, all ciphers have the same susceptibility to isomorphic reduction.

2.2 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 [3]. 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

$$R = \frac{1}{|s|} \tag{4}$$

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 that specific section of the 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.

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'.

Theorem 1: *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. □

As an example, consider an alphabet $A \mapsto A$ (a monoalphabetic mapping) using a 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.

Lemma 1: *For a S cipher applied to a message (M), there are* $(|A| - |T|)!$ *isomorphic keys.*

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

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

This lemma can be deduced from Wells' isomorph [3] (equivalent key) argument which is given below:

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 2: *For a P cipher applied to a message, M, there are* $\frac{(|B| - |S_t|)!}{\binom{|S_t|}{C}}$ *equivalent keys.*

Proof: Static bits in a P cipher can be found by using a modified inter-

section. For two blocks, B_i and B_j, and bit n in those blocks (denoted by $B_{i,n}$ and $B_{j,n}$), let $B_{st} = B_i \hat{\bigcap} B_j$, where m is the number of bits in the block,

$$B_{st} = B_i \hat{\bigcap} B_j = \begin{cases} 0, & \text{if } B_{i,n} = B_{j,n} = 0; \\ 1, & \text{if } B_{i,n} = B_{j,n} = 1; \\ x, & \text{if } B_{i,n} \neq B_{j,n}. \end{cases}$$

For M, a bit, b_i, is static *iff* $\forall B \in M, b_i \neq x$. The static bit set, S_t, is composed of the unique $B_{i,j}$ that are static. The number of remaining partial keys is the permutation of the dynamic bits, or $(|B| - |S_t|)!$ because the static key mappings are isomophic. The static bits can reduce the number of equivalent keys, depending on the combination of static bits. Bits may have one of two values, '1' or '0'. There are $\binom{|S_t|}{C}$ distinct possibilities for the combinations of static '1' and '0' bits. The maximum number of equivalent keys occurs when all of the static bits are of the same bit value. Dividing the maximum number of equivalent keys by the number of unique static bit combinations results in the total number of equivalent keys given by

$$\frac{(|B| - |S_t|)!}{\binom{|S_t|}{C}} \tag{5}$$

keys. □

Corrollary : *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 $\bar{y_s} = (\bar{s} : \bar{x'_s}k'_{22})$ 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 $\bar{x_s}$. Therefore, the number of unique keys is

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

□

Corollary 2: *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

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

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

$$|K_M| = |\text{systematic isomorphs}| \tag{6}$$

And

$$\forall M \rightarrow |K_M| \leq |K_c| \tag{7}$$

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 1 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.

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}

Table 1: Keys for a Given Block Size

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 length 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.

2.3 Language Combinations

Language is made up of patterns consisting of various repeated symbols that Carlson et. al [5] call "meta$n$characters." The meta$n$characters found in messages are the alphabet of the meta$n$language. Cryptographers often argue that any, and every, combination of meta$n$characters is possible in a message. However, this proves to be untrue. While there are more potential character combinations than the number meta$n$characters that can permuted, the percentage of allowed meta$n$characters falls well below 0.003% of the total possible permutations from the original message language by the time a 48 bit key is used. Limiting the possible mappings to allowed meta$n$characters is followed by finding the isomorph reduction due to unused meta$n$characters in the alphabet and message size.

Ciphers have the property of being one-to-one and onto. If they did not, decryption would not be possible. Disguising this property has led to the development of algorithms that attempt to add changes to the encryption by changing the key through calculable functions, such as XOR'ing the data with an initialization vector (CBC [4]). Such a decryption effort only requires knowing the function used and does not effect the key. Therefore, the addition of such a randomization function can be ignored in the analysis of identifying the correct systematic isomorph.

2.4 Redundancy and Language Patterns

Varying message content caused Shannon to bound the effect of patterns as if they were independent random variables (IRVs) [6]. However, language is not random. Language is full of patterns that depend a message's content. Such patterns manifest themselves in words [7] and in sentence structure. Languages have formal rules [8], lexicons [9], and associations between word context [10]. Rules, agreement for symbol meaning, and contextually related words all indicate that patterns, and therefore redundancy, will be present in a message.

Repetition, the key to decryption, commonly occurs in language. The occurrence of repetition results in lower entropy, making it easier to find the key. Each word makes a pattern in the language that can be exploited wherever and whenever it appears. Shannon estimated that letter redundancy in English is about 0.75 [6]. The presence of grammar and a lexicon further limit language patterns. For example, English has approximately 54,000 distinct word families [11, 12]. However, 50 words comprise 43 - 50% of the words normally used in written and spoken English [13], while only 1000 words cover 74 - 84% of the words used. Approximately 95% of commonly used English is covered by a mere 3000 words [13] or about 3% of the total words found in English. Similar lists and measures are available for French [14], Spanish [15], Dutch [16], and Chinese [17], demonstrating the constricting effect for a variety of natural language groups (see Table 3). While the exact number of words itself varies, the numbers for each language remains similar.

Languages have characteristic frequencies of letters and words [4, 18]. Evaluating the cipher text of messages for redundant meta*n*characters has long been used in decryption. Since most languages only require 3,000 - 4,000 words to make up 95% of the language [13, 14, 15, 16, 17] with its constituent

patterns and word combinations most meta*n*characters in the alphabet will be variations of those patterns. Zipf postulated that the more frequently a word appears the shorter the length of the word [19]. This further restricts the constituent symbols in blocks. Those chunks will make up the majority of blocks appearing in the encrypted message. For each redundant block in a message, the number of blocks that must be seen in the message to be seen for total alphabet coverage is increased by one. The number of unique words (w_u) in a sufficiently long message (m_l) will be bounded by

$$w_u \leq 3000 + .05(m_l) \tag{8}$$

For shorter messages, it is likely that $w_u << 3000$ words, greatly restricting the isomorphs making up the the key space. Word size in English is approximately 5.1 characters per word, limiting the average number of meta*n*characters in the language [18]. The exact number of words needed for coverage will also vary based on the subject matter used in corpus collection.

2.5 Complexity of Isomorph Reduction vs. Brute Force

The brute force attack [4, 20] 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.

2.6 Rainbow Tables

A method commonly used to speed brute force attacks is the rainbow table [21]. Rainbow tables can be used to trade memory for speed in discovering the key to a hash table or cipher. Strings of encryptions are assembled starting with high probability combinations of letters (m-grams or blocks) and encrypting them with possible keys, to attempt to find which keys are possible. Searching the strings results is a fast way to check for encryption mappings without having to encrypt each block with every key. Pre-computing the mappings means that the table can be reused without having to continually expend the effort of calculation.

Rainbow tables can greatly speed an attack, cutting the time needed to determine whether or not a key is correct. When used in conjunction with the key space reduction methods of meta$n$character and isomorph reduction, rainbow keys further can speed the time to decryption.

Other purely heuristic algorithms may be available to reduce decryption complexity. When employed, with key space reduction, they disprove the assumption that modern ciphers cannot be solved in useful time. Such assumptions are meant to demonstrate futility in even attempting decryption of messages obscured with modern ciphers. This assumption is shown to be false in the following example.

3 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 figure 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. A second example, one that can be easily verified, involves single letter characters and a language with an alphabet that consists of the characters a, e, i, s, and t. A monoalphabetic substitution ciper is used to encrypt plain text to cipher text. That is, $A \mapsto A$. The cipher text message received is "$ststii$," consisting of only 3 of the characters in the alphabet - s, t, and i. Each mapping is represented by a vector represented by:

$$a \mapsto k_a \; e \mapsto k_e \; ...t \mapsto k_t \;\text{ or }\; k_a k_e k_i k_s k_t \tag{16}$$

Let x represent a "don't care" in the mapping, ie. it does not matter what other unique mapping is use for that portion of the key. Then, in this case, only 60 keys of the possible 120 keys (5!) are unique. Those keys (isomorphs) are

xxaei	xxeis	xxist	xxtae
xxaes	xxeit	xxita	xxtai
xxaet	xxesa	xxite	xxtas
xxaie	xxesi	xxits	xxtea
xxais	xxest	xxsae	xxtei
xxait	xxeta	xxsai	xxtes
xxase	xxeti	xxsat	xxtia
xxasi	xxets	xxsea	xxtie
xxast	xxiae	xxsei	xxtis
xxate	xxias	xxset	xxtsa
xxati	xxiat	xxsia	xxtse

xxats	xxiea	xxsie	xxtsi
xxeai	xxies	xxsit	
xxeas	xxiet	xxsta	
xxeat	xxisa	xxste	
xxeia	xxise	xxsti	

On the average, only 30 of these isomorphs need to be attempted using a brute force attack in order to find the correct key.

Heuristics typically used in the decryption process include block frequency, words, sentences, grammar rules, and the context of the message. Each of these properties reduces the entropy of the message and can be used to eliminate possible meta*n*character mappings. Once all of the heuristic information is applied and mappings are reduced, brute force tests are begun. The effect is a much smaller key space to check, enabling decryption even of highly complicated ciphers. The results of the analysis show that complicated obscuring does not necessarily mean effective obscuring.

4 Polymorphism as a Countermeasure

Shannon based his analysis of cipher strength on the plain text language of a message, the cipher, and key selected for use with that cipher [22]. However, there is evidence that supports the assertion that the analysis should actually be done on the message that is being transmitted and encrypted [3] and the cipher. Messages encode the language characteristics but allow for a finer determination of uncertainty than is available for an average message.

The amount of information required to break an average message is known as the "unicity distance" (n) [22]. Once that number of symbols has been exceeded there is enough data to decrypt the message. In practice, the unicity distance does not reveal enough information to break most codes. Practically, most cipher text breaks require from 50 to 300 times the unicity distance [3]. Therefore, a message that is in the range of

$$1 \leq x < 50n \tag{17}$$

characters is safe because not enough information is available, on the average, to effect decryption. The closer that x is to 1, the more safe the message. Assume that a message (M) is broken into parts, called blocks (b_i) such that

the message is composed of each B_i that are concatenated ($\|$) together, then the message is

$$M = \|_{i=1}^{n} B_i \tag{18}$$

If each block is considered as a separate message and encrypted with a different cipher and/or key, then each block must be separately decrypted as if they are are different messages. The encryption is similar to the One Time Pad (OTP) [4] applied to the blocks. OTPs are known to be the only mathematically perfectly secure (unbreakable) cipher, so long as the method to select keys is random [23]. Each block, if kept well below the unicity distance, is impossible to solve because of the lack of information and acts like a symbol in a new language. The changes, known as a "polymorphic key progression algorithm," or PKPA, emulate an OTP without having the associated overhead. Because the blocks are larger than letters in an alphabet the key changes are infrequent enough that it is very difficult to gather enough data to break a function that selects the next key and/or cipher, making key progression functions feasible without compromising security.

Blocks used in a PKPA do not have to be of uniform size. It is better that the length of the blocks are not uniform but rather that they are sized using some heuristic measure. One of the best measures is local entropy [22, 24]. Local entropy measures the entropy, or uncertainty, in part of a message. If the entropy falls too low, then there is enough information in the block for decryption. The block for which the entropy is calculated is then capped below that length and the next block is constructed and examined. Isomorphs are not a problem because the formulas presented in Theorem 1 and its corollaries are applied to calculate the entropy. Language patterns and redundancy are also factored in via the entropy calculations. PKPAs that employ entropy based message decomposition do not suffer from the limitations of reduced key space due to isomorphs and language patterns.

5 Conclusion

Modern encryption systems are broken because the key space typically ends up being much smaller than predicted. The goal of decryption is to recover the original message, not to recover the exact key that created the encryption. If every possible block (meta*n*character) is found in a message key space would be an effective measure of cipher strength. However, most messages are not long enough and their content contains previously unrecognized linguistic

repetition gives clues about the key. Equivalent keys (or isomorphs) exist for many messages and using any of these keys will decrypt the message. Thus only one of the keys in the set need be considered.

The key space can be further reduced using other techniques. The first of note is the existence of forbidden letter combinations are found in the language. Eliminating blocks which never occur in practice (forbidden meta$n$characters) quickly eliminates possible mappings in the key space. Redundancy in a language also plays a role in reducing the number of possible combinations. Commonly used words also tend to be small, resulting in variations of blocks based on those words. For meta6characters the number of allowed combinations in English is only 0.003% of all possible symbol combinations [3].

If multiple decryptions are done on different messages with the same key, then yet more of the key can be revealed by intersecting the isomorph sets to distill both sets into a single smaller set. Once known, these mappings can also be eliminated from the solution space mappings for the other substitutions, again reducing the possible remaining key space. A situation in which the key does not change exposes the message to an even smaller key space with known mappings from other decryption efforts.

Each of these techniques arises from easily measurable features of the message. The cipher determines the maximum key space size, but the message determines how much reduction will result from applying each technique. By using the characteristics of the language, a new reduced key space can be constructed. The size of the key space can be easily calculated to compare the effort needed to run a brute force decryption.

Future work into key space reduction should include the collection of word and m-gram coverage for all natural languages. Variations and combinations based on the word structure of languages should also be collected, correlated, and disseminated to help establish patterns in those languages. Further work needs to be done to answer the question of how significant meta$n$characters are as n is increased above $n = 2$.

Preventing the cryptographer from collecting enough information to use language statistics to help decrypt the message will ensure that only a brute force attack will be effective. An effective way to combat key space reduction is to change key space often. A polymorphic key progression based on a strong pseudo-random number generator will also help to keep messages safe. Analysis of any encryption system must also take the message into account. Security is a function of the message, language, cipher system, and key. Only by using information theory, Shannon theory, and heuristics can the full effect

of repetition, rules, grammar, and lexicon on encryption be evaluated and steps taken to further protect the information. A PKPA with the message divided using local entropy calculations is one example of how a cipher system can defeat isomorph reduction. At the present time, a PKPA is the only system known to be effective in dealing with isomorphic reduction and the related attacks.

While modern ciphers make use of a much larger key space than earlier ciphers, key space is not the only factor in securely transmitting and storing data. New techniques must be developed and used to minimize the impact of heuristic and statistical algorithms in decryption. Based on information theory, these techniques will probably center around polymorphic key progressions and entropy calculations to ensure that accumulated repetition and a sufficiently large corpus is unavailable to the attacking cryptographer. Until those methods are identified and regularly used, key spaces will routinely be narrowed based on message content. Messages will continue to be broken and their content revealed to attackers.

References

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

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

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

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

[5] Albert H. Carlson, Robert E. Hiromoto, and Richard B. Wells. Breaking block and product ciphers applied across byte boundaries. In *The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 733–736, 2011.

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

[7] Victoria Woollaston. Anatomy of a hack: How crackers ransack passwords like qeadzcwrsfxv1331". Internet, 28 May 2013.

[8] William Delaney O'Grady, Michael Dobrovolsky, and Francis Katamba. *Contemporary Linguistics: An Introduction*. Copp Clark Pitman, Ltd, 3rd edition, 1996.

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

[10] S. Small, G. Cotreell, and M. Tememhaus, editors. *Lexical Ambiguity Resolution*, chapter Resolving Lexical Ambiguity Computationally with Spreading Activation and Polaroid Words, pages 73 – 107. Morgan Kaugmann Publishers, San Mateo, 1988.

[11] H. J. DuPuy. The rationale, development and standardization of a basic word vocabulary test. Technical report, US Government Printing Office, Washington, DC, 1974.

[12] R. P. Goulden, P. Nation, and J. Read. How large can a receptive vocabulary be? *Applied Linguistics*, 11:341–363, 1990.

[13] Paul Nation and Robert Waring. *Vocabulary Description, Acquisition and Pedagogy*. Cambridge University Press, 1997.

[14] B. New, M. Brysbaert, J. Veronis, and C. Pallier. The use of film subtitles to estimate word frequencies. *Applied Pscholinguistics*, 28:661, 2007.

[15] Fernanco Cuetos, Maria Glez-nosti, Analia Barbn, and Marc Brysbaert. Subtlex-esp: Spanish word frequencies based on film subtitles. *Psicolgica*, 32:133143, 2011.

[16] E. Keuleers, M. Brysbaert, and B. New. Subtlex-nl: A new frequency measure for dutch words based on film subtitles. *Behavior Research Methods*, 42(3):643–650, 2010.

[17] Q. Cai and M. Brysbaert. Subtlex-ch: Chinese word and character frequencies based on film subtitles. *PLoS One*, 5(6):8, 2010.

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

[19] George Zipf. *The Psychology of Language*. Routledge, London, 1936.

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

[21] Philippe Oechslin. Making a faster cryptanalytical time-memory trade-off. In *Lecture Notes in Computer ScienceAdvances in Cryptology: Proceedings of CRYPTO 2003, 23rd Annual International Cryptology Conference*, 2003.

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

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

[24] 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.

Albert H. Carlson earned his Ph.D. degree at the University of Idaho in Computer Science. Albert earned a BS Computer Engineering from the University of Illinois, Urbana and an MS Computer Science at the University of Idaho. With over 20 years experience as an Engineer and Engineering Manager, he has worked in safety critical circuits, consumer and industrial electronics, communications, control circuits, and embedded systems. He has been involved in the design of sate of the art cell phones in the early 1980's, industrial gas dehydrators, electronic warfare equipment, the first microprocessor based train doors, custom integrated circuits of many types, Zenith HDTV, the first rack mounted mini-DSLAMs, and polymorphic key progression algorithms. He developed course work and taught engineering and security at many levels, including the US Army Intelligence Center and School at Ft. Devens, the University of Illinois, and the University of Idaho. Presently, he is researching set theory and decryption technology.

Robert E. Hiromoto received his Ph.D. degree in Physics from the University of Texas at Dallas. He is professor of computer science at the University of Idaho and former Department Chair. His areas of research include ad hoc wireless mobile networks for unmanned autonomous vehicles, and information-based design of sequential and parallel algorithms, decryption techniques using set theoretic estimation, and parallel graphics rendering systems. Recently Dr. Hiromoto was awarded a Fulbright Fellowship to study Wireless Self Authentication in the Ukraine.

m	No. Forbidden	No. Allowed	Total No. m-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%

Table 2: m-gram Numbers for English

Language	Words for Coverage
English	3000
French	3680
Spanish	3000
Dutch	4000
Chinese	6000
Russian	7000

Table 3: 95% Coverage for Various Languages