[A. Carlson, S. Khare, I. K. Dutta, B. Ghosh and M. Totaro. (Jan. 26-29, 2022). Space Selection and Abstraction in Set Theoretic Estimation, *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2022, pp. 0012-0020, doi: https://doi.org/10.1109/CCWC54503.2022.9720845]

Space Selection and Abstraction in Set Theoretic Estimation

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

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

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

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

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

***Abstract*—Set Theoretic Estimation has been used in diverse applications for quite some time. Most applications use a Hilbert space for problem solving; however, if a distance metric is not needed, the complexities and features of Hilbert space may not be required. A recent attempt to extend STE methodology to cryptography has led to a refinement of the set space used for this application. In some cases, such as cryptography, a topological space provides the necessary functions and structure. Solving this problem in a less restrictive space allows for ease of implementation and increased computational speed. A less ordered topological set space in which data is set and manipulated is described, along with the required functions to operate on the data. Possible extensions of this space abstraction are also presented for problems exhibiting similar characteristics. Cryptography is considered a difficult problem in any space, so the problem is both a relevant and illustrative demonstration of the results of solution space selection. We employ set methods and select an appropriate space in which to solve cryptography problems.**

I. Introduction

Set Theoretic Estimation (STE) is an emerging technique that has been applied successfully in diverse fields. This includes class identification, which applies not only to system analysis, but also to the analysis of regular languages, connectedness problems, discernibility, and communications [1], [2]. In this paper, we explore the potential applicability of set theoretic space selection and abstraction to the problem of decryption. We present an elementary framework and examine its properties when applied to several cipher systems; namely the shift, substitution (S), permutation (P), and block ciphers mixing S and P methods. Although early in development, the method appears to offer potential merit for justifying further research and development.

II. Background

A. Set Theoretic Estimation

The first step in understanding STE is to define the symbols used to communicate allowable operations and concepts precisely. The notation of both First Order Predicate Logic (FOPL) [3] and standard set operations apply. Common symbols, such as $\forall$, $\exists$, and $\exists!$ are defined as meaning "for all," "there exists," and "there exists exactly one," respectively. Standard definitions are also applied to the $\cap$, $\cup$, $\subset$, and $\in$, as "intersection," "union," "subset," and "element of," respectively [4]. Rules are expressed as assertions in STE. Assertions consisting of the symbols become the language and grammar of STE. Each rule or constraint is represented by its own unique set. Information known about both the inputs and rules is treated similarly. An assertion, A, takes the set of possible inputs and gives a set of resulting outputs, or solutions, for the operation, O, as specified in the rule. For a particular input, i, this output is expressed as

$$O_i = A(i) \tag{1}$$

The set of all possible solutions for all possible inputs, called a "property set," is described as

$$O = \bigcup_{j=1}^{m} O_j \tag{2}$$

and is found in an m-dimensional solution "space" known as Ξ^m. The space is composed of elements called "points," each

representing a member of a solution set, $\Phi_n = O$. Points in Ξ^m are formally known as the set theoretic estimates. Multiple rules can be asserted, resulting in multiple "property" sets in the solution space. For each rule asserted, there will be a Φ representing the solutions and $\bar{\Phi}$ consisting of all other points in the solution space. Values of Φ are determined by the nature of the application. For instance, in a communications application expressed in terms of voltages Φ would also be a range of voltages. There may be many distinct subsets of Φ that are defined by the input to a rule.

Traditional STE is used in Ξ^m space, which can be a finite or infinite Hilbert space. Hilbert space is both a complete metric and vector space. Many Hilbert spaces exist, but the Hilbert space selected for use will implement a specific metric used to calculated distance between points in the space. Selection of such a space depends on the application and the nature of the points being ordered. The result of the metric is a mathematically ordered set, a bounded volume, where points in Ξ^m have a definite spacial relationship with each other. Mathematical and set operations are performed on the shapes and points in the space. The geometric shapes resulting from the ordering of the set greatly simplifies visualizing the operations on the sets.

The function

$$d(\cdot,\cdot) : \Xi \times \Xi \rightarrow [0, +\infty)$$

is said to be a distance, or metric [5], function if it has the following characteristics:

$$(\forall (a,b) \in \Xi^2) d(a,b) \;=\; 0 \;\Leftrightarrow\; a \;=\; b$$

$$(\forall (a,b) \in \Xi^2) d(a,b) \;=\; d(b,a)$$

and

$$(\forall (a,b,c) \in \Xi^2) d(a,c) \;\leq\; d(a,b) \;+\; d(b,c)$$

The diameter of a set, S, is represented by $\delta(S)$ and is said to be bounded if, for the set S,

$$\delta(S) \;<\; +\infty$$

The goal of STE is to find an answer to a problem described by set membership rules. Starting in Ξ^m, each set Φ_n is considered in turn. Each Φ_n contains only those possible solutions that follow the rule that the assertion describes. Since Ξ^m contains all possible solutions, then the solution P must be in the solution space and must be a member of the property set for each assertion if a solution exists. That is, if $\exists P$ then

$$P \;\in\; \Xi^m$$

and

$$\forall \Phi_n \;\rightarrow\; P \;\in\; \Phi_n$$

Further, if P is in each of the solution sets, then it must also be true that

$$P \;\in \bigcap_{i=1}^{n} \Phi_i$$

If a solution is found in the intersection of the sets, the intersection is said to be "consistent." If it is not, then the intersection is said to be "inconsistent." If the resulting solution has exactly one answer, $\exists!$, the solution is said to be "ideal."

When a particular input is applied to the assertion represented by the set Φ_n, then part of the set represents the valid outputs for that input. Φ_n is then constricted to a subset Ψ_i. If one set subsumes another set, that is if

$$\Psi_i \;\subset\; \Phi_n$$

then one can work with Ψ_i rather than Φ_n during calculation. By being a member of Ψ_i, membership in Φ_n is also established. Further, if a single set, Ψ_j, is subsumed partially by a group of other sets, i.e., if

$$\forall m \in \Psi_j \;\rightarrow\; \exists \Psi_i \mid i \neq j \;\wedge\; m \in \Psi_i$$

where $\wedge$ represents the logical 'AND' operator, then Ψ_i may be removed from calculations without loss in generality.

Subsuming sets reduces the overall number of sets. Combettes recommends the elimination of those sets that add little or no value in determining the final solution when applied with other property sets to an input [5]. Eliminating non-contributing property sets results in increased computational efficiency, as well as time and memory savings.

Most operations defined for STE directly manipulate sets. However, there are no general rules for construction of these sets. Each set must embody an assertion that describes the output in some fashion and excludes other portions of Ξ^m when applied to an input. Constraints about a system may also be asserted using distance functions or by setting hard boundaries.

Though often used, distance metrics are not required for STE. Points can be manipulated by purely Boolean operators in a topological space. The topological space can be used to reduce computational effort if solutions can be readily calculated for the property sets. No distance metrics or geometrical treatment is required and, therefore, the additional constraints of metric and vector spaces are no longer needed.

Combettes also notes that the approach to many estimation problems identifies solutions in Ξ^m that violate the constraints of the original problem. In response to this potential violation, the goal of STE is to "follow the notion of feasibility," [5] and produce solutions whose main function is consistent with all known *a priori* data about the problem in the post application analysis.

B. Communication Theory of Secrecy Systems

Shannon defined several measures of interest in the secrecy paper. One of the more important statistical and informational measures is "unicity distance," (n). Unicity distance is the average number of encrypted symbols needed to break a cipher. Mathematically, n is given as

$$n \approx \frac{log_2|K|}{R_\lambda log_2|A|} \tag{3}$$

where R_λ is the redundancy of the language, $|K|$ is the number of keys in the cipher, and $|A|$ is the number of symbols in the encrypted alphabet. For instance, R_λ has been calculated to be approximately 0.75 for the English language. In general, languages follow statistical patterns that vary slightly from user to user and message to message. Implied in the unicity distance is the need to find ways to increase unicity in order to keep encrypted data secret.

The existence of an $R_\lambda > 0$ indicates that a language has redundancy and is, therefore, susceptible to statistical attack. Redundancy provides a clue to the possible role of the duplicated symbol or collection of symbols present. With respect to letter probability, non-uniform distribution of letters in the language can be exploited to correlate their occurrences to symbols in the encrypted alphabet. Some of these measures include letter and word frequency, word size, and combinations of letters. Cryptographers have successfully decrypted messages using these statistics for many years.

Among the techniques that Shannon explored was the use of language regularity that appear as word repetition and patterns. Such regularities were believed to be statistically characterized. Since the elemental analysis is done at the character level, identifying word patterns ultimately result in letter patterns, which can then be measured and described statistically. The chance of encountering a particular combination of letters is based on their frequency in the language. Shannon called the combination of m letters "$m-$grams." As m becomes larger, many combinations of arbitrary letters cannot be decrypted into understandable text. $m-$grams can be characterized by how often they appear in a language.

Using the shift cipher, Shannon demonstrated how $m-$grams are used in decryption. Shannon selected a plaintext of the letters and encrypted them using one of the possible keys. He then applied all possible decryptions for the encrypted data. Starting at the first letter of the resulting cryptotext, each key was applied, resulting in 25 possible decryptions. For $1 \leq m \leq 6$, Shannon formed the appropriate $m-$gram for each possible decryption. He then checked the probability of each $m-$gram for each possible decryption. If the probability fell to 0, the key was abandoned as impossible.

Trivially, all decryptions are possible for a single letter, but beginning with the 2-gram combinations, some keys can be eliminated. Shannon noted that there are different statistics for the case where the first letter in the encryption begins a word and another set of statistics for letters occurring elsewhere in words [6]. Shannon also calculated the entropy to show it was decreasing towards a solution. In time only a single key with a probability greater than 0 remained. The selected key was the correct key, demonstrating the correct application of a brute force attack aided by the $a\ priori$ knowledge of $m-$gram frequency and probability in English of ordered letters.

While Shannon did not specifically indicate the mathematical nature of his attack, it included commonly used techniques. These sets are the $m-$grams for $1 \leq m \leq 6$. Each $m-$gram result is a set with a possible decryption using the encrypted message, $E_k(M)$, as the input. The key is used as a transformation function from the encrypted to decrypted space. The goal is $\exists! k \rightarrow D_k(E_k(M)) = M$. However, it is also acceptable to find a set of keys, $k \subset K$ that reduces the probable keys to a smaller number than all possible keys.

Describing the desired result in terms of $m-$grams with their probabilities fulfills the conditions of STE. Intersections are made for $m-$grams that are members of all of the defined sets. Shannon ignored $m-$grams that contain smaller $m-$grams whose probability is 0. Bayes Law [7] shows that if an $m-$gram has no chance of occurring, the probability of any other $m-$gram built on it is also non-occuring. Once an $m-$gram with 0 probability is encountered, the key associated with that $m-$gram can be eliminated from further consideration.

C. Topological Space

Evaluating the space abstraction in STE as a consequence of Shannon's $m-$gram method of message decryption yields the following list of requirements for the space:

1) The space must be finite;
2) The space is discrete;
3) No distance metric is required; and
4) Set operations are defined in the space, especially intersection.

Hilbert space is infinite and continuous, typically using a distance metric based on how the points in STE are ordered as part of the solution methodology. A finite subset of the Hilbert space could be employed, but Hilbert space has more features than the problem requires. Therefore, it may be easier to manipulate a different space and more closely follow the characteristics of the solution without including unnecessary overhead in processing data. One type of space that embodies these characteristics is topological space.

Topological spaces are spaces where sets define the space [8]. They are typically used to formalize concepts about a set, such as convergence, connectedness, and continuity. Each space is defined over a set X by a topology T. T is a set of subsets that include X, the empty set Ø, and subsets of the set X of interest. Any sets formed by applying the $\bigcup$ and $\bigcap$ operators to any collections or sets found in T are also found in T. Elements found in T are called "points." Sets in T can also be said to be either "open" or "closed." Open sets do not include their boundaries while closed sets do.

The composition of a topological space makes it well suited for representing and operating on sets. Topological spaces cannot be infinite because T is a set and Russell's paradox forbids an infinite number of subsets [9]. Neither does a topological space have a distance function. However, logic and set functions can be applied to sets in the space, so long as the proper topological space is selected. Thus topological spaces are a more natural choice than Hilbert spaces for this type of STE operation, since it reduces the computational overhead needed to deal quickly with encryption.

III. Theoretical Framework

To illustrate the proper use of topological space, we will first show how the space is populated for a decryption problem. Assume that a decryption problem involving a message, M, is to be solved using STE. We must first select a topological space in which to apply the technique. Without any *a priori* knowledge all keys that can decrypt M are equally probable. That is, $\forall k_i \in K$, where k_i is a possible key in the key space, K, and $M = D_k(M)$,

$$p(k_i = k) = \frac{1}{|K|}$$

Thus, the space must contain the set of all possible keys as an upper bound.

Let the solution set be populated with the power set of the key space, denoted as $Pow(K)$. All possible subsets of K reside in the space, allowing set operations between subsets in the space. For set operations $\cup$ and $\cap$ applied to two subsets in the space, the set that results from the set operation will also be in the set. All Φ_n applied to the message M will also be in the space. No extraneous information resides in the population of the space and it is not necessary to perform an Optimal Bounding Ellipsoid (OBE) function [10]–[12] at the conclusion of a set operation to make future mathematical calculations easier. Further, no error can be inserted into the set because the set is never expanded for computational convenience (since we are not operating on volume, but rather operating on the set). The space is closed over set operations so manipulations of the set of remaining possible keys is efficient.

The key set used in topological space depends on the cipher used in encryption and the message, M, under consideration. Ciphers are designed to exhibit a one-to-one mapping. That is, each key maps the input message, M, to a unique cipher text encryption $E_k(M)$. Therefore, for each cipher text block $\exists! k$ results in the correct decryption. Shannon predicated his calculation for unicity distance on the assumption that $H(k|E_k(M)) = 0$ [13]. However, there are cases when several keys can yield the same decryption given the same input message. In these cases $E_{k_i}(M) = E_{k_j}(M)$, where $i \neq j$. Such keys are said to be *equivalent* for message M.

IV. The Application

A. An Example Using Various Ciphers

Verifying that Shannon's m-gram example is compatible with STE provides an instance of a non-Hilbert space application in topological space. The goal of the experiment is to use a source corpus to provide information about m-grams in a language and then attempt to decrypt a cipher encrypted data stream using data from the corpus. Success is indicated by recovering the correct key value used in the encryption. To demonstrate our approach, we begin with texts known to be written in English. Each text is separately encrypted using a cipher that ensures that the encrypted data stream is not in plaintext form. Samples of literature taken from the Project Gutenburg [14] library in .txt format serve as the corpus for m-grams, as well as the source for encryption. The entire process is automated.

Unlike Shannon's m-gram approach, which involves finding the most likely m-gram for the input string, our approach eliminates m-grams that are not valid. Forbidden m-grams encountered during decryption indicate that the key used in the decryption is incorrect. Since forbidden combinations of letters cannot occur in a valid string of a message in the target language, impossible m-grams can only come from an incorrect decryption caused by using the wrong key for the message being decrypted[1]. When forbidden m-grams are encountered the key used to create the decryption can then be eliminated from further consideration. In effect, we look for strings that cannot occur in normal speech (m-grams with a 0 probability). Shannon stopped his consideration of a key when a forbidden m-gram was encountered, but he continued seeking the key whose probability was 1. Eliminating all impossible keys does the same thing, leaving only that

[1]We will deal with the case of deliberately-inserted "pads" of nonsense plaintext in a separate paper. This paper deals only with the general development of the STE method for shift, S, P, PS, PSP, and SPSP block ciphers.

key which is possible. We call this method the "Last Man Standing" scenario.

The steps required to complete STE decryption are as follows:

1) Train the m-grams from the corpus - The input data first has all non-alphabetic characters removed (including spaces). Encrypted data is read in as strings of the size being trained. Training starts with the first character in the file and increments one character at a time until each m-gram is processed. Processing consists of recording that an acceptable m-gram was found and then incrementing the position in the file by one m position. Any m-grams found are removed from a list of "forbidden" m-grams. What remains are m-gram combinations not found in the language. Training continues until the entire file has been processed and each m-gram of interest has been trained. In this specific implementation of m-gram training, the size of m-grams has been limited to $2 < m < 6$, because the number of possible m-grams to track quickly increases.
2) File Selection and Encryption - An electronic file is selected and a key is pseudo-randomly selected by means of a cryptographic pseudo-random number generator. The file is encrypted using the selected key and all non-alphabetic characters are removed from the input file. Several types of encryption, including the shift, S, P, PS, PSP, SPSP, and block ciphers of the same type were used. The texts selected for this experiment, and their respective authors, is found in Table I.
3) Decryption - All possible keys for the cipher are constructed and marked as possibly correct. The encrypted data file, constructed in step 2, is then decrypted and analyzed. Starting with two letters from the encrypted file, each key is applied to determine if it can possibly be the correct key. If the data resulting from the decryption violates the m-gram probabilities, the key is marked as "not possible" and is removed from further consideration. After the final key on the list is attempted, the number of keys still possible is counted. Cycling continues until less than two keys remain or the input file runs out of data.

Finding a single key indicates that the decryption process has selected a key as the correct decryption key. A result of 0 keys means that the program could not find the key and all possibilities were eliminated. Because the program selected the encryption key in step 2, the original key is available for comparison. The success of the decryption is verified by comparing the encryption and decryption keys.

Implied in the forbidden m-gram technique is the knowledge of which m-grams are permissible and which are not. No ordering is required. The only information needed is whether or not a particular m-gram is allowed. For m-grams of size n, there is no relationship between m-grams to determine which are and are not allowed. The number of m-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. Because of limitations on the amount of memory available and the effort required to store and retrieve data from such a large file, m-grams larger than $m = 6$ are not presently considered for use.

TABLE I
TWO BYTE BLOCK FILES USED FOR TESTING

File	Title	Author
1linc11cp.txt	The Writings of Abraham Lincoln	Abraham Lincoln
1onwr10cp.txt	On War	Carl von Clauswitz
alice30cp.txt	Alice in Wonderland	Lewis Carroll
1anne11cp.txt	Anne of Green Gables	Lucy Maud Montgomery
hoend10cp.txt	Howard's End	E. M. Forser
jandc10cp.txt	Jefferson and His Colleagues	Allen Johnson
jmlta10cp.txt	The Jew of Malta	Christopher Marlowe
lglass18cp.txt	Through the Looking Glass	Lewis Carroll
wwill10cp.txt	The Wind in the Willows	Kenneth Grahame
wwrld10cp.txt	The Way of the World	William Congreve

Multiple m-grams may reside in a string. For a string of size x, where $x \geq 6$ letters, the number of m-grams available, $|mg|$ is given by

$$|mg| \;=\; \sum_{i=2}^{6}(x-(i-1)) \tag{4}$$

With each symbol input from the ciphertext, up to five data points are added to the sum of knowledge about the decryption. All languages have inherent symbol repetition. Inputs may be repeated if the letters received are repeated. Gathering more than one m-gram per input symbol helps offset data repetition. In Shannon's example of m-gram use, only the m-gram formed at the beginning of the ciphertext was considered. Other m-grams formed from the middle of the stream were not. However, the use of mid-stream (intermediate) m-grams is valid because it is equivalent to beginning the decryption at an arbitrary point within the encrypted message.

All keys are assumed to be equally possible at the beginning of the decryption. There is no $a\ priori$ knowledge about the keys that would reduce that number until encrypted letters are

analyzed. Letters are received one at a time and analyzed in the same order. Decryption does not wait for the full string to be completed prior to analyzing the message. The goal is to achieve a decryption using as few letters as possible, developing the solution as the string develops.

Developing a solution depends on applying property sets that can gradually eliminate possible solutions. In this experiment, the language of the message was English. As the English language set is entirely contained in the natural language set, the set of natural languages can be ignored.

The remaining property sets used in the example consist of those applied to letters, words, and sentences. Letter sets are made up of forbidden m-grams. Forbidden m-grams are compiled during the training process and kept in sets by the size of the m-gram. Word sets are made up of two dictionaries: a lexicon of words in the language of interest (not including proper nouns) and a list of proper nouns drawn from multiple languages. Sets for words are applied to the decrypted data to ensure that the entire data stream can be split into a continuous group of words. All possible word solutions are produced. The word set does not ensure that the word combinations in the potential message are grammatically correct. This task is left to the last set which is a set composed of grammatically correct sentences. A routine is then called that attempts to parse the word grouping. If a parsing on one or more of the possible word groupings is returned as grammatically correct, then the data is readable in some form.

The steps of this method have the advantage of being very modular. Modularity facilitates testing and allows comparisons of the results to focus on the differences between corpora. Sets are formed and called as needed. Sets can also be from any corpus or language that uses an alphabet.

Testing membership in each of the sets used in the example can yield one of two results: true or false. True indicates that the selected decryption keys possess the property of the set. There is no degree of membership. Membership is either complete or not at all.

Several sets of tests were conducted to demonstrate the effectiveness of STE in topological space. For encryptions that involve encrypting the message one alphabetic character at a time, the shift, substitution (S), permutation (P), and block ciphers composed of PS, PSP, and SPSP ciphers were tested. By testing the S and P ciphers, both confusion and diffusion encryption techniques are shown to be susceptible to STE.

B. Single Byte Cipher Results

Single byte ciphers are those which operate on a single character in the English language, represented in ASCII by a single byte of data. The shift, S, and P ciphers are all single byte ciphers. In this case, the P cipher permutes within the same character or byte. Results for these ciphers are as shown in Table II.

TABLE II
TEST RESULTS FOR THE SINGLE BYTE CIPHER

Cipher	No. Tests	% Solved	Mean Characters	Time
Shift	4916	95.487%	5.55	$<$ 0.5ms
Substitution	1437	85.53%	256	50.7 s
Permutation	1047	99.85%	256	0.563 s

For a shift cipher in English $n \approx 1.3$ characters [13]. For the S cipher, the $n \approx 28$ characters and 4.85 characters for the P cipher. The mean, μ, is the average number of characters that are required by the program to find the correct decryption key for the encrypted file. Although the results are above the P cipher's unicity distance, STE methods decrypted more quickly than other common decryption methods. Typical results are approximately $50 \times n$ for the S and P ciphers, while STE methods are about $15 \times n$. The time required to solve the P cipher is shorter than that needed for the S cipher. Reduced time is a result of the reduced key space size. An S cipher in English has a key space of 26! keys while the P cipher has only 8! keys.

Errors encountered in testing primarily came from incomplete characterization of the English language. Because English borrows words from many languages such as Danish, German, French, and American Indian, it is not uncommon to find "foreign" phrases embedded in texts. Most languages are not "pure," rather they borrow from other languages. Borrowing phrases makes identification of the base language more difficult. Some borrowed words violate the normal letter combination found in a language. Analysis techniques based on a languages' rules can be fooled by the inclusion of foreign words, thus yielding an incorrect answer. Methods using a strict membership criteria, such as the one implemented in the application used in this paper, are vulnerable to mixed language problems. This may have contributed to increased n values seen in the tests.

Names found in the text are also likely to cause errors in decryption. The use of names has the same effect as foreign words because many names are borrowed from other languages or are constructed from sounds. Some literary works feature constructed names, such as Burroughs' "Tarzan of the Apes." Many of the names found in that work have no basis in any natural language and prior to their publication were never heard. The advent of travel between cultures has also dispersed names beyond the language and area where they originated. Humans readily recognize names as words not normally found in the language and treat them as identifiers. Compensating for the inclusion of new words is accomplished by allowing the forbidden m-grams to occur several times

before eliminating the $m-$gram mappings in question. In other words, a number of violations (or relaxations [15]) should be permitted before eliminating any given key.

C. Multi-byte (Block) Encryption Results

Most modern ciphers deal with the ease of breaking single byte ciphers by encrypting groups of characters at a time. Multi-byte, or block ciphers, have a larger alphabet, resulting in a larger unicity distance. The STE methods used to decrypt block ciphers are the same as for single byte ciphers if the blocks are treated as characters in a language. For ease in working with these blocks, several definitions are required.

Definition 1. meta-s-character

A meta-s-gram (meta(s, m)) is an m-gram composed of m meta-s-characters. For example, the text composed of 'theonl' is a meta(3,2) made up of two different meta-3-characters '*the*' and '*onl*'. Note that a meta(s,m) is equivalent to an m-gram of the size $s * m$. □

Definition 2. meta-s-gram (meta(s, m))

A meta-s-character is an *m*-gram of size $s = |m - gram|$ alphabetic symbols from the original language. For example, the meta-character '*the*' is referred to as a meta-3-character. Meta-characters are treated as a single symbol in the language. Block ciphers that encrypt s characters at a time are encrypting a meta-s-character. □

Meta-s-characters and meta(s, m)'s take the place of the letter and m-grams in the encryption and decryption algorithms for block ciphers. Block cipher testing for this paper was conducted for S, P, PS, PSP, and SPSP ciphers.

Every block encrypted text in this experiment was correctly decrypted, regardless of the cipher type employed. The time required for decryption of each cipher type was nearly identical (see Table III). It should be noted that standard deviations for the decryption of each text using all encryption methods was small, amounting to less than 0.03% of the mean in all cases. Variance in decryption times can most likely be attributed to the overhead of background tasks in the computer used to host the tests.

Variation in the time and number of characters required for decryption appears to be dependent on several properties of the files. The properties identified were:

1) Author style;
2) File size;
3) Non-standard English (such as names, place names, and imaginary words);
4) The era in which the work was written; and,
5) The original language in which the work was written.

TABLE III
TWO BYTE BLOCK DECRYPTION RESULTS

File	S (sec)	P (sec)	PS (sec)	PSP (sec)	SPSP (sec)	Mean (sec)
1linc11cp.txt	675	669	671	676	661	670.4
1onwr10cp.txt	14507	14442	14682	14273	14185	14417.8
alice30cp.txt	44617	44690	44386	44470	44473	44527
1anne11cp.txt	778	770	773	774	775	774
hoend10cp.txt	861	847	854	848	795	841
jandc10cp.txt	1387	1381	1391	1388	1398	1389
jmlta10cp.txt	12680	12723	12488	12616	12624	12626.2
1glass18cp.txt	7851	7664	7828	7603	7716	7732.4
wwill10cp.txt	765	743	750	744	750	750.4
wwrld10cp.txt	546	550	550	546	552	548.8
Average	8466.6	8447.9	8437.3	8393.8	8392.9	8427.7

Addressing each point in order, we will start with time variation due to the different authors. Authors have distinct styles of writing [16], including the use of similar sentence structure and lexicon in all of their works. Reusing the same patterns in structure and words, result in a set of m-grams trained with those patterns. Consequently, authors that share similar stylistic patterns should decrypt in similar times and number of ciphertext characters. For example, *Alice in Wonderland* and *Through the Looking Glass*, both written by Lewis Carroll, showed similar decryption times.

Message/file size also factored into the efficiency of breaking the file in a particular cipher (see Table IV). The smallest text file sizes in the test set were *Alice in Wonderland*, *Through the Looking Glass*, *On War*, *The Jew of Malta*, and *The Way of the World*. All of these files were less than 117 kB in size, while all other test files were larger than 245 kB. Shorter messages contain less data and, therefore, a reduced probability of low entropy events such as m-gram redundancy. In light of corpus size, these decryption results support Shannon's contention that having more data in a message increases the probability of correct message decryption [6].

Alice in Wonderland had the greatest diversity of names of all the files tested. It also took the longest time of all text files to decrypt. Correspondingly, *Through the Looking Glass* also took longer to decrypt than other test files, due to the presence of names and imaginary words contained in the text. Low frequency m-grams, including forbidden m-grams, required more search time and effort to decrypt correctly. *The Jew of Malta*, a work that included a large number of foreign names and locations also had problems with low frequency m-grams resulting from those words. Patterns in those words, and consequently these m-grams, are not as likely to be represented in the m-gram sets.

TABLE IV
TWO BYTE INPUT META-2-CHARACTERS

File	Title	File Size	Symbols Seen
1linc11cp.txt	The Writings of Abraham Lincoln	350	105
1onwr10cp.txt	On War	483	3464
alice30cp.txt	Alice in Wonderland	106	7610
1anne11cp.txt	Anne of Green Gables	483	102
hoend10cp.txt	Howard's End	459	102
jandc10cp.txt	Jefferson and His Colleagues	315	139
jmlta10cp.txt	The Jew of Malta	105	3524
lglass18cp.txt	Through the Looking Glass	117	2061
wwill10cp.txt	The Wind in the Willows	245	102
wwrld10cp.txt	The Way of the World	115	102

Corpus data was derived from the same works of English used for decrypting meta-1-character files. During the time periods covered by the corpus, English language use evolved, changed, and was re-characterized. Word and usage patterns regularly change with popularity over time. Changes in lexicon and language habits can result in literary era dependent m-gram sets and, therefore, give rise to different decryption performance. Customizing m-gram sets for a particular era, over which the language has remained relatively static, may increase future decryption efficiency and accuracy. Sets of data derived from the same time period as the message are more likely to consist of the same word usage and frequency patterns as the message. Customized time period language property sets require further research and are beyond the scope of this paper.

The original language of a text is also important. As expected *On War*, originally written in German and later translated into English, decrypted in a similar amount of time as *The Jew of Malta* (written in English, but uses foreign settings), due to location names and foreign words. Even though foreign texts are translated into English, names, locations, and other proper nouns retain their original language patterns [16]. Foreign locations and names are usually transliterated, producing low frequency m-grams that are mistaken as being imaginary words. More low frequency m-grams increase the number of characters and time required for decryption. Foreign texts in languages closely related to English take about the same effort to decrypt as English texts with foreign names and locations. Reducing the effect of foreign words would require the use of additional property sets drawn from a corpus of the original language. Future work involves adding such a corpus and then comparing the decryption times in cases using different property set combinations.

Because there are many more meta(s, m)s as the block size increases, using the same corpus for allowable meta(2,m)grams presents much smaller coverage for larger block sizes. A corpus sufficient for meta-1-characters has less coverage for meta-2-characters, and even less coverage for meta-3-characters. Thus, as the size of the meta-s-character increases, the size of the corpus needed to represent the language also must increase in size. Since memory limitations restricted experimentation to the use of meta(2,3) and meta(2,4) sets for meta-2-characters, only those cases were evaluated for multi-byte tests. For meta-1-characters, the percentage of coverage from the corpus for meta(1,3) is 64.3% and for meta(1,4) the coverage is 24%. If the same corpus is used for the meta-2-characters, coverage is reduced to 0.597% for meta(2,3)s and 0.003% for meta(2,4)s. An increase in the corpus size and composition is required to ensure a sufficient number of meta-s-characters in the meta-language are represented; however, the specific amount of increase needed by the corpus requires further study.

The upper limit of the size of m for the meta(s, m)s included in the corpus is dictated by stylometric and training constraints. If too many examples of an author's work are included in the corpus, the corpus may become over-trained. Over-training represents one author's style while ignoring the patterns found within the body of language. Similarly, as the size of the meta(s, m)s used as property sets (m) increases, the meta(s, m)s reflect the style of the author used as a source in the corpus, thus over-training results and correct keys are rejected. Over-training can be avoided by using a diverse set of authors in the training set and limiting the size of the m-grams used to characterize the language.

V. CONCLUSION

Problems traditionally solved using STE have typically been placed in a Hilbert space [5]. However, the application of STE to decryption using language based analysis is shown to require far less structure than is embedded in a Hilbert space. m-gram based STE decryption does not require a continuous, or infinite, space as the application does not require a distance metric. STE instead requires the application of set functions on points in the space. As such, topological space is better suited to set operations because it consists of groups of subsets taken from the full set of possible keys.

Unicity distance is dependent not only on the cipher and language, but the content of the message that is encrypted. It is quite possible to have two messages of the same length where one message contains enough information to be decrypted while the other message does not.

STE in topological space proved to be effective in solving decryption problems. Using a space appropriate for the application and problem results in a more efficient solution and reduces the work required to implement the algorithm.

As such, applications having the same characteristics as the STE based decryption example are better placed

REFERENCES

[1] Sanjeev Kulkarni and David Tse. A paradigm for class identification problems. *IEEE Transactions on Information Theory*, 40(3):696 – 705, 1994.

[2] Richard Wells. Application of set-membership techniques to symbol-by-symbol decoding for binary data transmission. *IEEE Transactions on Information Theory*, 42(4):1285 – 1289, 1996.

[3] Geoffrey Hunter. *Metalogic, An introduction to the Meta-theory of Standard First-Order Logic*. University of California Press, Berkeley, 1971.

[4] Felix Hausdorff. *Set Theory*. Chelsea Publishing Co., New York, 1962.

[5] Patrick Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182 – 208, 1993.

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

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

[8] John Kelley. *General Topology*. D. Van Nostrand Company, Princeton, 1955.

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

[10] Eric Walter and Helene Piet-Lahanier. Estimation of parameter bounds from bounded-error data: A survey. *Mathematics and Computers in Simulation*, 32:449 – 468, 1991.

[11] Eli Fogal and Yih fang Huang. On the value of information in system identification - bounded noise case. *Automatica*, 18(2):229 – 238, 1982.

[12] Jr. John Deller and Yih fang Huang. Set-membership identification and filtering for signal processing applications. *Circuits Systems Signal Processing*, 21(1):69 – 82, 2002.

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

[14] The Gutenburg Project. Main page. Internet, 2008.

[15] Shmuel Peleg and Azriel Rosenfeld. Breaking a substitution cipher using a relaxation algorithm. *Communications of the ACM*, 22:598 – 605, 1979.

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