SET THEORETIC ESTIMATION APPLIED TO THE INFORMATION CONTENT OF CIPHERS AND DECRYPTION

Albert H. Carlson
Austin Community College

SET THEORETIC ESTIMATION APPLIED TO THE INFORMATION CONTENT OF CIPHERS AND DECRYPTION

A Dissertation

Presented in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

with a

Major in Computer Science

in the

College of Graduate Studies

University of Idaho

by

Albert H. Carlson

May 2012

Major Professors: Robert E. Hiromoto, Ph.D. and

Richard B. Wells, Ph.D.

AUTHORIZATION TO SUBMIT DISSERTATION

This dissertation of Albert H. Carlson, submitted for the degree of Doctor of Philosophy (Ph.D.) with a major in Computer Science and titled "SET THEORETIC ESTIMATION APPLIED TO THE INFORMATION CONTENT OF CIPHERS AND DECRYPTION," has been reviewed in final form. Permission, as indicated by the signatures and dates given below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor ______________________________ Date______________
Robert E. Hiromoto

Co-Major Professor ______________________________ Date______________
Richard B. Wells

Committee
Members ______________________________ Date______________
Paul W. Oman

______________________________ Date______________
Robert E. Rinker

Department
Administrator ______________________________ Date______________
Gregory Donohoe

Discipline's
College Dean ______________________________ Date______________
Larry Stauffer

Final Approval and Acceptance by the College of Graduate Studies

______________________________ Date______________
Jie Chen

Abstract

Set Theoretic Estimation (STE) has been known and applied to various problems since 1969. Traditionally, STE has been used to solve vector problems in a Hilbert space using a distance metric to create a volume in that space. Given this type of space structure, Optimal Bounding Ellipsoid (OBE) algorithms are typically used to simplify STE estimate processing; however, OBEs are not bounded to the original estimate volume defined by the STE problem. At times the OBE algorithm includes spurious estimates in the new volume that may incorrectly expand the solution set. In contrast to prior implementations of STE, the algorithms used in this dissertation are based in topological space. Errors may still be present in the selected property sets, but these sets are used only as a priori information that are static and does not expand (change). Therefore, by choosing a topological space the problems associated with OBEs are avoided.

For the first time, STE has been applied to decryption and STEs effectiveness is demonstrated. The problem of diffusion across byte boundaries is address by assuming that a block of symbols (meta-s-characters) are encrypted in that block. Language patterns are not obscured in meta-s-characters because of the constraint on the fixed block size. Furthermore, it is shown that all block ciphers are block substitution ciphers. Since all block ciphers are substitution ciphers, a single attack is effective against substitution, permutation, and block ciphers composed of combinations of substitution and permutation ciphers. The BCBB algorithm that decrypts block substitution ciphers is introduced and the results of its application are presented.

Property sets designed to complement (how do they complement each other) each other are presented for the decryption problem. Further, it is then shown that that the property sets contain the properties of the Asymptotic Equipartion Property (AEP). Via the AEP, it is shown that Information Theory comes under the umbrella of STE.

Acknowledgments

I would like to thank my major professors and committee for the guidance they have provided during the course of this research. The committee is as follows:

Co-Major Professors: Dr. Robert Hiromoto
Computer Science Department
University of Idaho
Dr. Richard B. Wells
Microelectronic Research and Communications Institute
University of Idaho

Committee Members: Dr. Paul Oman
Computer Science Department
University of Idaho
Dr. Robert Rinker
Computer Science Department
University of Idaho

I have had help from some exceptional undergraduate students. They include: Frank Mitchell, Erik Schweller, Paul Nathan, Brandon Arp, and Sarah Mitchell. Their interest in my work has helped me think more deeply about it than I might have otherwise. They have also helped develop tools to help test my ideas and explaining things to them has helped me to clarify many concepts and procedures in my own mind.

At home I have been supported by my wife, Tina, and my children, Ariana, Robert, and Alan. They have sacrificed a lot of time with me so I could work. They have understood the need for my work and listened to math and ideas that they do not fully understand without complaint. I love them and hope that they will still be just as

enthused about my work in the future.

Throughout my youth, I had several people who kept me walking along the path of computer science. My parents, Martin and Nola Carlson, made sure I had the opportunity to learn. Martin worked many long hours so that his family could live. Nola read with all of the children and provided direction for academic growth. At school, Mr. Charles Oklpek and Mr. David Drymiller directed me towards science, engineering, and mathematics. My gratitude to David Drymiller, who always told me to learn enough to teach my teacher. I hope I have succeeded.

Dedication

I would like to dedicate this work to those members of the Armed Forces who daily work in rooms without windows, behind guarded doors, in the service of their country. They use the techniques of cryptography and cryptanalysis to protect others. They will never be personally recognized for their work. They are not, however, forgotten or unappreciated.

Contents

Authorization to Submit Dissertation ii

Abstract iii

Acknowledgements iv

Dedication vi

Contents **vi**

List of Tables x

List of Figures xii

1 Problem Statement and Literature Survey **1**

1.0.1 Current State of the Art 1

1.0.1.1 Cryptography 1

1.0.2 Shannon Theory 3

2 Set Theoretic Estimates and Property Sets **17**

2.1 Overview 17

2.2 Review of Present Theory 19

2.2.1 Property Sets 24

2.2.2 Topology of STE 26

2.2.3 Optimal Bounding Ellipsoids 28

2.2.4 Information and Shannon Theory 40

2.2.5 Unifying STE and Information Theory via the Asymptotic Equipartition Property 44

2.2.6 Assembly and Use of Property Sets for Decryption 49

2.3 Chapter Summary and Contributions of This Dissertation 55

3 Developing the Corpus for Decryption 57

3.1 Overview 57

3.2 The Corpus 58

3.3 Property Sets 59

3.4 Conclusion 64

4 Decryption of Single and Multi Byte Ciphers Using STE 67

4.1 Overview 67

4.2 Specific Cipher Type, Algorithm Application, and Results of Testing 68

4.2.1 Shift Ciphers 68

4.2.1.1 Empirical Results For the Single Byte Shift Cipher 70

4.2.2 Substitution Cipher 71

4.2.2.1 Theoretical Basis for Single Byte Substitution Cipher 71

4.2.2.2 Testing Procedure for the Single Byte Substitution Cipher . 74

4.2.2.3 Empirical Results for Single Byte Substitution Ciphers . . . 75

4.2.3 Permutation Cipher 76

4.2.3.1 Theoretical Basis for Single Byte Permutation Ciphers . . . 76

4.2.3.2 Testing Procedure for Single Byte Permutation Ciphers . . . 86

4.2.3.3 Empirical Results for Single Byte Permutation Ciphers . . . 92

4.3 Chapter Summary 94

5 Applying STE to Product and Block Ciphers with Diffusion Across Byte Boundaries 97

5.1 Overview . . . 97

5.2 Review of Theory . . . 98

5.2.1 Chapter Definitions . . . 98

5.2.2 Product Ciphers . . . 99

5.2.3 Block Ciphers . . . 100

5.3 PS Security Equivalence to the S Cipher . . . 104

5.4 PSP Security Equivalence to the S Cipher . . . 108

5.5 Empirical Work . . . 112

5.5.1 Overview . . . 112

5.5.2 Property Sets . . . 114

5.5.3 Algorithm Description for Decrypting Product and Block Ciphers (BCBB) . . . 117

5.5.4 Improvements to the BCBB . . . 121

5.5.5 BCBB Decryption Results . . . 122

5.6 Advancements to the State-of-the-Art . . . 127

5.7 Chapter Summary . . . 128

6 Summary, Conclusions, and Future Work 130

6.1 Summary . . . 130

6.2 Conclusions . . . 132

6.3 Future Work . . . 133

Appendix 152

A Complexity Analysis 152

List of Tables

2.1 UOBE Algorithm Factors 52

2.2 m-gram Numbers for English 52

3.1 Training Corpora for English 66

4.1 Shift Cipher Results 70

4.2 Substitution Cipher Results 77

4.3 Permutation Cipher Results 93

4.4 Permutation Decryption Files Failing Successful Decryption 95

4.5 Permutation Decryption Results with Relaxation 95

5.1 Security for a Block of m Bytes 111

5.2 Two Byte Block Files Used for Testing 118

5.3 Two Byte Block Decryption Results 118

5.4 Two Byte Input meta-2-characters 129

List of Figures

1.1 Feistel Round 16

4.1 First Algorithm Step 77

4.2 Shift Cipher Algorithm 78

4.3 Substitution Cipher Overhead 79

4.4 Substitution Cipher Algorithm 80

4.5 Keys per Unique Symbol Count in a Message 85

4.6 Permutation Cipher Algorithm, Part 1 89

4.7 Permutation Cipher Algorithm, Part 2 90

4.8 Relaxation 91

5.1 Permutation Cipher 102

5.2 PS Type Cipher 102

5.3 Byte Permutation 103

5.4 Permutation Inside Byte 103

5.5 Bitwise Permutation 103

5.6 Unicity Distance for Substitution Cipher of n-Byte Blocks 112

5.7 ϵ vs. Block Size (in bytes) 113

5.8 S_R vs. Block Size (in bytes) 113

5.9 Metacharacter Percentage for Block Size 2 116

5.10 BCBB Algorithm Setup 119

5.11 BCBB Algorithm Body 120

5.12 BCBB2 Algorithm 123

Chapter 1

Problem Statement and Literature Survey

1.0.1 Current State of the Art

1.0.1.1 Cryptography

Cryptography and cryptanalysis are about obscuring and revealing the information contained in a message. Recovering the message without knowing the original key is the objective of the cryptanalyst while keeping data obscured is the objective of the cryptographer. Both facets of the science are of interest in this study, with stronger emphasis placed on revealing obscured information. Although an encrypted message may seem to be a collection of chaos, decryption of the message is possible. By employing statistical and heuristic knowledge about a language efficiently, even small messages can be decrypted. Identifying and organizing information for efficient use is central to the decryption effort. A method that has not been applied before this effort is Set Membership Theory (SMT). By organizing information into sets, it should be possible to operate on the encrypted message using the assembled sets to aid in decryption even if the key is unknown from the onset.

Modern ideas about cryptanalysis can be traced to ideas codified in the late 19th century. Auguste Kerckhoffs was a 19^{th} century French military officer who was interested in the military uses of cryptography. In 1883, Kerckhoffs published an article, meant to be used as a guide for military ciphers used in the field, that gave six characteristics of strong ciphers. Until Kerckhoffs [1] stated these principles of security for ciphers, 'security by obscurity' was considered a vital part of cryptosystem design. With his published principles, Kerckhoffs changed the focus of cryptanalysis from security by obscurity to peer-reviewed trap door algorithms with large key spaces which are computationally infeasible to break. Because of Kerckhoffs' work, mathematical and numerical methods became much more important in cryptography. The six principles presented by Kerckhoffs in his seminal paper are as follows:

1. The cipher must be practically, if not mathematically, indecipherable;

2. The cipher method is assumed to be known by the enemy (no security by obscurity), so that the message may fall into the enemy's hands and is not immediately decipherable. Unless the key is known, the message will remain secret;

3. The key must be accessible without written notes and changeable at the will of the users;

4. The cipher must be applicable to telegraphic (electronic) media;

5. The cipher method must be portable; and

6. The cipher must be easy to use, not requiring excessive training or the application of many rules.

The advent of using electronic computers to implement and break ciphers followed Kerckhoffs' fourth rule, but altered the definition of practically indecipherable, memorable, portable, and easy to use.

1.0.2 Shannon Theory

Just over twenty years after Kerckhoffs article was published, Claude Elwood Shannon was born in Petoskey, Michigan. Shannon was destined to become a major contributor in Logic Design, Cryptography, and Communications Theory. In 1938, Shannon wrote his influential Master's Degree Thesis, entitled, "A Symbolic Analysis of Relay and Switching Circuits" [2], that applied Boolean Logic [3] to relays and switches, paving the way for a mathematical basis of modern Logic Design Theory. During the late 1930's, Shannon worked for Vannever Bush on the forerunner of the analog computer. Shannon earned his PhD from MIT in 1940, with an early mathematical treatment of the principles of Mendelian genetics [4]. By the end of World War II, Shannon created a classified report on cryptosystems, which was used as the basis for his later work on secrecy systems. After cryptography, Shannon concentrated on communications systems, where he established the basics of Information Theory.

The mathematical foundations that allowed Shannon to introduce Information Theory [5] included work in set methodology, set theory, entropy and the definition of topology. Important contributions to Shannon's work came from various sources, working independently on different problems, from the late 19^{th} century through the first half of the 20^{th} century. Near the beginning of the 20^{th} century, Bertram Russell introduced a paradox that later became known as "Russell's Paradox," [6] involving the possible composition of power sets and collections. Russell's Paradox sets limits on the composition of a Topological Space that can be used in Set Theoretic Estimation. Based on Hartley's equation [7], the amount of uncertainty, or “surprise,” in the data encountered in a set of symbols is given by entropy. Entropy relates the probability distribution function (pdf) for a set with probability of encountering an element x_i, given as $p(x_i)$, to uncertainty. Hartley’s equation is

$$H(x) = -p(x_i) \sum_{i=0}^{n} log_2[p(x_i)]. \tag{1.1}$$

Entropy is maximized when the probability of seeing any member of a set is equal, that is $H(X)$ is largest when $\forall x_i \in X$

$$p(x_i) = \frac{1}{|X|}.$$

Set Theory [8] was introduced in the 1880’s. Cantor put forth the framework that later allowed mathematicians to explore and systematize problem requirements. In the early-to-mid 20^{th} century, there was an effort to define the definition of problem spaces mathematically. This study, known as Topology [9], is a branch of geometry that deals with the connectivity of a problem and attempts to operate on the common elements of a problem in terms of the objects represented in the space. The shape of the object is not as important as the characteristics of the object, which cannot be altered by stretching or distortion. Requirements for a space that has a set X in it include:

1. The empty set, $\emptyset$, and X are contained in the space;

2. Given any two sets, x_i and x_j in the topological space, $x_i \cup x_j$ is also contained in the topological space; and

3. Given any two sets, x_i and x_j in the topological space, $x_i \cap x_j$ is also contained in the topological space.

Using the proper space in which the characteristics of interest remain constant allows operation on the essential and common basis of interest. For Set Theoretic Estimation (STE) problems, the spaces where common attributes are exploited are Hilbert Space [10] and Topological Space [9, 11].

Hilbert Spaces [10, 12] are closed vector spaces and metric spaces in n-dimensions, denoted by Ξ^n, which is characterized by the following properties:

1. Distance between two points - The distance between any two points a and b in Ξ^n, given by $\langle a, b \rangle$, is constant for any two points. That is

$$\langle a, b \rangle = \langle b, a \rangle \tag{1.2}$$

2. Identity - for the points a and b in Ξ^n

$$\langle a, b \rangle = 0 \quad iff \quad b = a \tag{1.3}$$

3. Vector distance - for any three distinct points a, b, and c in Ξ^n

$$\langle a, c \rangle \leq \langle a, b \rangle + \langle b, c \rangle \tag{1.4}$$

4. Closure - if a sequence of vectors approaches a limit, then the limit is also contained in the space. An example of a Hilbert Space is the three-dimensional space in which we live.

A general Topological Space [9, 11] is a space that satisfies the first three conditions listed above with respect to a set (X) in the topological space. There is no requirement for vector distance or metric distance to be valid in the space. Distance metrics are functions that have the first three properties noted for Hilbert Spaces and specify how $\langle a, b \rangle$ is calculated.

A topological space is dimensionless and may be populated by sets. Let the topological space be populated by the power set of a set. Each possible subset of the original set is contained in the space, and the union of each subset is also contained in the same space. The intersection of any set is also contained in the power set and, therefore, in the same topological space. A convenient way to visualize a topological space is to use a Venn Diagram to represent subsets of interest in the space. The universe represented by the Venn Diagram is the power set. Topological spaces have practical application in both Shannon Theory and Information Theory for the decryption of ciphers.

Shannon introduced his analysis of communications in "A Mathematical Theory of Communication" [13] in 1948. The paper established the foundations of Information Theory. Shortly after "A Mathematical Theory of Communication" appeared, Shannon addressed the subject of ciphers in "A Communication Theory of Secrecy" [5] and showed how Information Theory could also be applied to cryptography. Shannon had previously worked in the field of cryptography during World War II and applied his mathematical rigor to the subject, analyzing how statistics could be applied to describe and operate on ciphers. In "A Communication Theory of Secrecy," Shannon introduced important information theoretic measures of ciphers that included redundancy and unicity distance.

The redundancy of a language is the tendency for characters in a language to be repeated. It can be calculated using the equation

$$R_\lambda = 1 - \frac{H(x)}{H_{max}}. \tag{1.5}$$

Redundancy relates repetition to the entropy of the message. Approximating the redundancy in the English language was the subject of Shannon's 1951 paper [14], "Prediction and Entropy of Printed English." Shannon's approach to measuring redundancy was to treat redundancy as an independent random variable (IRV) and measure the number of guesses a practitioner of the language took to guess the next letter in a message. Shannon found that practitioners were more likely to correctly guess the next letter when substantial portions of a word or sentence were revealed. More uncertainty was encountered at the beginning of words and sentences. Shannon set the redundancy for English at $R_{English} = .75$.

Redundancy of symbols in language infers that the symbols in the alphabet of a language are not uniformly-distributed independent variables. Collecting examples of language from a corpus of literature in that language can provide statistics for that language. If the corpus is large enough, the Law of Large Numbers [15] indicates the statistics will be applicable for instances of data similar to the corpora. Information used to decrypt ciphers has traditionally come from a variety of statistics. A useful language statistic that Shannon demonstrated for decryption in his work, "Communication Theory of Secrecy," is that of m-grams. An m-gram is a collection of letters where the m-gram is a compound symbol $< x_0, x_1, .., x_{m-1} >$ contained in the plaintext message and $x_i \in A$ of the language. Shannon made use of the property that not all m-grams are found in a language and that, given combinations of ciphertext symbols, some keys may be eliminated from consideration in decryption.

Morton [16] builds on the idea of statistics in literature in his book, "Literary Detection." He shows that there are more than 35 sources of statistics that can be employed to identify an author. The same statistics can be used to assist decryption. Statistics form the basis for the property sets [12] utilized in Set Theoretic Estimation.

The unicity distance (n) provides a measure of how many symbols, on the average, are required to eliminate all spurious keys for decryption of a message. Unicity distance is based on the size of the key space $(|K|)$ in a language (λ), an alphabet (A) of size $|A|$, and the redundancy (R) of the language. Unicity distance, measured in symbols, is given by the equation:

$$n = \frac{log(|K|)}{R_\lambda log(|A|)}. \tag{1.6}$$

For a particular cipher in a language, the unicity distance is constant. For different ciphers encrypting messages in the same language, the difference in unicity distance depends solely on the key space. Different unicity distances imply that key space can be used to compare ciphers and says that, in general, larger key spaces will result in stronger ciphers. The unicity distance provides a way to compare the security of a cipher as well as predict the minimum message size needed for complete cipher decryption.

Shannon [5] also introduced the concept of mixing cipher types. Ciphers can employ both confusion and diffusion. Confusion comes from the substitution of one symbol for another; diffusion results from the spreading of information across more than one symbol. Shannon reported that certain ciphers increase the overall security when applied as a series of cryptographic functions. The preferred order of applying the ciphers is:

$$F = LSLSLT, \tag{1.7}$$

where L represents a linear cipher, S represents a substitution cipher, and T represents a transposition cipher. Modern block ciphers employ mixes to increase their security and hide language statistics.

Throughout history, many ciphers have been used to obscure information transmitted between parties. An excellent summary of the basic cipher types is found, along with their analysis, in Schneier's "Applied Cryptography" [17]. Ciphers may appear to be unreadable but can be mathematically easy to solve. Single encryption function ciphers have now been replaced by product and cascading ciphers [18]. Maurer, et al. analyzed the security of product and cascade ciphers, concluding that a cascade cipher was only as strong as the weakest cipher in the chain and that a product cipher was at least as strong as the first cipher applied in the encryption process. Maurer, et al, concluded that for a product cipher the encryption is at least as strong as that of the strongest cipher used in the encryption process.

An extension of the product cipher is the block cipher [17]. Block ciphers operate on multiple symbols simultaneously and are designed to implement both diffusion and confusion. Diffusion spreads information across the block with the goal of disguising the language characteristics of the plaintext. Substitution is the arbitrary mapping from the alphabet of the plaintext language to the ciphertext alphabet. Mappings may be arbitrary and do not have to be calculable from some regular function ($F(x_i)$). At least one substitution function is included in the block cipher, denoted by 'S' in the formula for F. A popular form of the block cipher employs a "Feistel Round" [19] implementation (See Figure 1.1). A 'round' is a group of functions that is performed on data for encryption. Each round is a step in the total encryption process. Feistel created a modular system that uses a single key and various ciphers in a single round. The input block is divided into two sub-blocks, the Left (L) and Right (R) sub-blocks. The R sub-block is encrypted using an exclusive OR (XOR,$\oplus$) cipher

$$E_{XOR,k}(B) = B \oplus k \tag{1.8}$$

with the key and then undergoes the application of additional substitution, permutation, XOR, and other functions before undergoing a rotation equal to half of the block size. The number and types of functions vary from cipher to cipher but the basic architecture of the round does not vary. It is generally accepted that sixteen rounds are sufficient for good data mixing [17]. In addition to the sixteen rounds, a permutation normally precedes the first Feistel Round and follows the final Feistel Round. Some Feistel Round ciphers implement more than 80 ciphers during the encryption process of a single block of the message. Feistel Round ciphers are widely used. Reasons for their popularity are that they use the same key for encryption and decryption [17], and can be efficiently implemented in both hardware and software, and are thought to be among the most secure available for use. Feistel Round ciphers are extensively studied for weakness by cryptographers. Weak keys for Feistel Round ciphers have been identified for attacks, as have various chosen plaintext attacks. The most effective attacks on a Feistel Round block cipher are the Linear [20] and Differential [21] Attacks, both examples of heuristic chosen plaintext attacks.

Linear cryptanalysis attempts to find an approximation to the action of a cipher. That is, the Linear Attack tries to approximate the substitution for key blocks of input and then generalize that substitution for the entire cipher. While the attack is mathematically possible, it requires vast amounts of chosen plaintext to be successful. The Linear Attack was inspired by Biham, et al's Differential Attack.

Differential cryptanalysis uses chosen plaintext to try and uncover statistical patterns revealed by applying the selected plaintext. The differences between the ciphertext produced for each of the differential plaintexts are analyzed with the internal structure of the cipher in mind. High probability differences are identified and traced through the the predefined substitution box (S-box) structure to reconstruct the key. Though an effective attack against many ciphers, most ciphers are now designed with resistance to the Differential Attack in mind.

On the average, both the Differential and Linear Attacks have similar effectiveness, measured in the number of chosen plaintext blocks that must be processed and compared, on the average. Presenting and evaluating a single block takes much more than a single instruction. The complexity of the solution is typically of an order larger than the number of blocks processed. Goldreich [22] suggested that the complexity of a decryption solution was a better computational measure of security than the more commonly used key space.

Key space comparisons give a mathematical comparison that does not translate well into computational feasibility. Computational security depends on the computational feasibility of a function and Goldreich argues that the measure is more representative of real security.

Computational efficiency requires organization and application of data to a problem. A methodology that takes advantage of the structure and nature of a problem, as well as the known data relating to the problem, is required. During the late 1960's, two researchers were attempting to characterize the behavior of "fast moving" and "evasive" targets for the US Navy. Within months of each other, Witsenhausen [23] and Schweppe [24] independently published papers that proposed a new approach to solving vector problems. Witsenhausen and Schweppe proposed taking the total possible solution space and reducing it to eliminate those solutions that were not possible.

Early research building on Witsenhausen and Schweppe focused on solving vector-based problems [25] and estimating the present state of linear dynamic systems. Other researchers discovered that this technique, now known as Set Theoretic Estimation (STE), could be used to characterize the parameters of problems [26, 27, 28, 29, 30] or model problems [29, 31]. In order to use STE, the problems must have "bounded error" and be resolvable to definite sets. Problems are expressed in a Hilbert space using a distance metric to order the estimates inside the space.

Bounded error refers to the characterization of accumulated error (ρ) in the solution of a problem. If the error in estimating the membership in a set can be guaranteed to be below some limit ($\lim(\rho)$) throughout the solution of the problem then:

$$\epsilon \leq \lim(\rho) \tag{1.9}$$

Bounding the error ensures that the amount of error introduced into a problem is both known and can be accounted for in the solution. In a volume based STE solution the bounded error allows for the introduction of spurious solutions in order to simplify calculations during the solution process. Spurious solutions are eliminated during further processing, but the addition of errors slows the algorithm execution and can adversely effect execution time. Ideally, the bound on error would be no error added during the simplification process.

Property sets, or sets made of estimates that display a certain property, are applied to input values and matched to the set of estimates. Only those estimates that exhibit the property are kept in a solution set. The solution set is the volume remaining from the intersection of the previous solutions set and the property set. Mathematically, expressing the remaining volume becomes more difficult as succeeding property sets are applied. The volume of the remaining set may be bounded by a geometric figure, known as a "bounding ellipsoid," [32, 33, 34]. The solution set is then defined as the estimates contained by the volume of the bounding ellipsoid. To make it easier to process the remaining set of estimates for the possible solution, an "optimal bounding ellipsoid" (OBE) [35, 36, 37] is applied, which guarantees that the solution's estimate is contained in the ellipsoid. Some error is accepted in order to ease calculation. The question of the fitness of the OBE is open because while "optimal" is desired, optimal is sometimes unnecessary [38] and difficult to calculate. Deller, et al. [39] presented a unifying theory for OBE problems, called the UOBE, in 1994. The UOBE shows how to trade off interpretability and convergence for use in STE applications. After finding the OBE, the next iteration in applying property sets begins. Some methods of OBE use the least squares method [36, 40], while other methods make assumptions about bounding [37, 41] to set the optimization parameters. All of the methods assume that the problem is being solved in a Hilbert space.

Although it is used in a number of applications, STE remains relatively obscure as a methodology, and work in the field is not widely shared. Combettes [12] summarized the state of STE in the early 1990's, showing that STE is applicable to a wide variety of problems [42, 43, 44], and not just those in vector and parameter estimation problems. McCarthy and Wells [45] demonstrated that SMT and STE techniques can be applied to communications and digital design problems.

STE, as commonly implemented, suffers from the constraints of the space, requirements for a distance metric, and the overhead incurred by OBE. A substantial effort is also expended in geometrical calculations mapping the problem into volumes in the space. Ideally, STE could be used without having to map the sets into a Hilbert space and bounding would come without a penalty.

Ciphers, which act like noise on communications channels, make it natural to ask the question: Is STE more effective for organizing and employing *a priori* information about a language than present decryption algorithms and techniques?

In an attempt to answer that question, this dissertation makes the following contributions:

- Applies STE to Cryptography - STE has been used in a variety of applications [12]. Until this dissertation work, STE has never before been applied to Cryptography.

- Applies STE in a Topological Space - STE has traditionally been used in a Hilbert or Metric [46] space. The use of a Topological space eliminates the need for a distance metric and simplifies the implementation of STE decryption.

- Uses Language Statistics as Property Sets - Language statistics function well as property sets, as demonstrated by experimental results for this dissertation.

- Employs Local Entropy and Unicity Distance to Direct Attacks - Entropy and unicity distance are typically used to characterize a cipher, given a specific language. Calculating entropy and unicity distance on a portion of a message can yield information on what part of a message to attack.

- Reduces Ciphers for Analysis - Analysis of a cipher can be accomplished by converting a cipher into another type of cipher. For instance, under some circumstances a Permutation (P) cipher can be replaced by a Substitution (S) cipher. Properties of ciphers, such as idempotence [47], can be applied to simplify decryption.

- Reduces Feistel Rounds - Using cipher reduction techniques, it can be shown that some Feistel Round ciphers can be simplified for decryption.

- Evaluates Entropy and Unicity Distance to Messages - Messages are instantiations of a language. Applying local entropy and unicity distance to the message allows the cryptanalyst to decide what messages may be decrypted and how much information is needed for the decryption.

- Identifies and Isolates Static and Dynamic Bits in P Ciphers - A technique to isolate the role of bits in a P cipher leads to techniques to reduce the amount of effort required to decrypt the message in the P cipher.

- Identifies Block Boundaries in Ciphers - Using the identification of static and dynamic bits in a P cipher allows the cryptanalyst to quickly identify block boundaries and block size in a P block cipher.

- Applies a Chosen Plaintext Attack to Block Ciphers - This dissertation presents a simple chosen plaintext attack for block ciphers, one that employs language statistics. The attack demonstrates that PSP type block ciphers are S ciphers.

- Uses Meta Symbols to Counter Diffusion in Block Ciphers - Diffusion is often cited as a way to spread information to disguise language statistics. By selecting characters in a new language that are the size of the block employed by the cipher, it is possible to defeat the diffusion of information by keeping it in the same symbol.

- Establishes STE as a Branch of Information Theory - To date, Information Theory and STE have been separate. This dissertation links STE to Information Theory, implying that Information Theory techniques are valid in STE.
- Establishes Applications Between the AEP and STE - The AEP is shown to hold for STE.

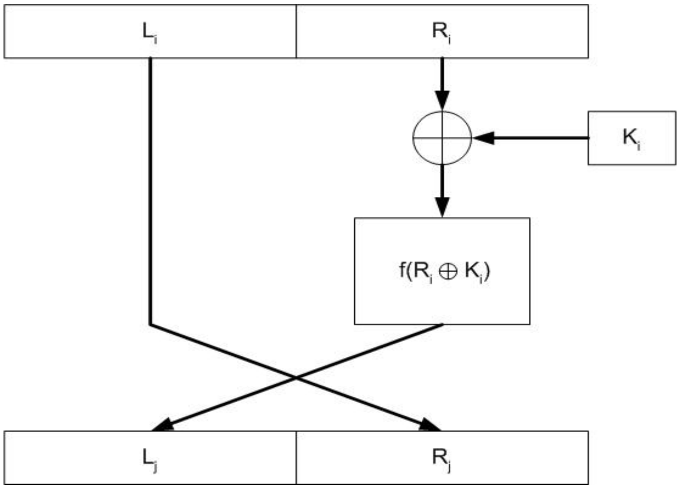

Figure 1.1: Feistel Round

Chapter 2

Set Theoretic Estimates and Property Sets

2.1 Overview

Set Theoretic Estimation (STE) is not commonly used as a problem solving methodology since not all problems are well suited for it. However, it is ideal for the use of problems with bounded errors and that are analyzed with respect to solutions that need to demonstrate a certain set of properties. In STE, an algorithm is designed to help differentiate between inputs that show the desired property and inputs that do not. The property set may be deterministic in membership or the set may be determined probabilistically.

While STE is very effective, the method relies on the geometric manipulation of a solution set projected as a volume in vector space. Distance metrics strongly influence how easily a problem is solved. The distance metric gives a weighting function to the property that it expresses and makes manipulation of the solution set based on that property more mathematically convenient. Given the metric selected, the application of repeated property sets to the ordered solution estimates can result in partial solution sets whose volume is difficult to describe and manipulate mathematically. To respond to the problem of working with irregular solution set shape, it is common to bound the volume with an ellipsoid [36, 42]. Error may be introduced in the bounding process by including previously eliminated estimates inside the ellipsoid, but the ease of manipulating the set is thought to offset the "small" amount of error that results from such bounding. Minimizing that error is a goal of research on Optimal Bounding Ellipsoids (OBEs) [37] and is further discussed in section 2.2.3. *A priori* information, information known previous to the implementation of STE through input data or world knowledge, also influences the quickness/accuracy of the problem solutions. Without *a priori* knowledge of the input data and the behavior of all property sets, the choice of which metric will yield optimal estimate ordering and results is often impossible to predict. The more *a priori* knowledge given, the easier it is to choose an appropriate distance metric.

In some cases, the added error and the calculation of OBEs can be avoided by changing the space in which STE is applied. By employing a topological space populated by the power set of the solution set ($Pow(S)$), the distance metric and OBE calculations can be avoided altogether. No error is added to the solution set because no bounding occurs.

One significant contribution of the study presented in this dissertation is to show that STE is a constituent of the Asymptotic Equipartition Property (AEP) [48]. This makes STE a branch of information theory. Unification of STE and information theory is significant because this unification further indicates that STE can be applied to problems that are currently solved by using clasical information theory methodology. Combettes [12] cited an extensive list of other STE applications that included digital filtering [49, 50, 51, 52, 53, 54], mathematics [55, 56], statistics [57], antenna theory [58, 59, 60, 61], acoustics [62, 63, 64, 65], signal processing [66, 67, 68, 69, 70, 71, 72, 73, 74], information theory [75], medical imaging [76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87], signal recovery

[88, 89, 90], spectral estimation [91], linear equation solving [92], speech processing[93], optics [94, 95, 96, 97, 98, 99, 100], biology [101, 102], image recovery [103, 104, 105, 106], control systems [107, 108], cloud trajectories [109], electrical engineering [110], communications [111], neural networks [112, 113, 114], remote sensing [115], magnetics [45, 116], and geography [117]. Recently, much of the work in STE and OBE has been in signal processing [39] and voice processing [93]. Even though STE has been used in many fields [25, 26, 27, 28, 35, 36, 37, 40, 42, 44, 93, 118, 119, 120, 121, 122, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 123, 61, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 92, 88, 89, 90, 62, 63, 64, 65, 91, 94, 95, 96, 97, 98, 99, 100, 101, 102, 75, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117], it has never before been applied to cryptography. Since information theory is the basis for modern decryption, STE would be a natural choice of methodology to apply to current decryption efforts. This dissertation explores the use of STE as a method for decryption – a new and important contribution.

2.2 Review of Present Theory

Set Theoretic Estimation (STE) traces its roots to two seminal papers, written in 1968 by Witsenhausen [23] and Schweppe [24], and published almost simultaneously. Both papers were concerned with military research into what was termed “evasive” targets. Witsenhausen and Schweppe understood that the actual path taken by the target must come from the set constructed of all of the possible paths from the initial location to the end location. The goal was to reduce the number of possible paths to a single path taken by the target. To do this, they viewed each vector as a state in a progression of states. Any information known about the target or path at the time the solution was attempted, called *a priori* [48] information, was used to eliminate possible paths in the solution set.

The theory as originally outlined by Witsenhausen and Schweppe can be more generally explained as follows. Typical STE problems are vector based and either arrive at an answer derived from a function of vectors or estimate the coefficients of such a function. However, STE problems must meet several other basic conditions. The requirements for an STE application are:

1. The problem in question must have a deterministic $f(x)$ and inverse $f^{-1}(x)$;
2. The problem must have bounded error for any input;
3. Each possible answer to the function must be computable; and
4. The problem must be expressible using different properties that differentiate between groups of inputs, inferring set membership.

In his 1990 paper,"The Foundations of Set Theoretic Estimation," Patrick Combettes [12] gave an excellent overview of the basics of STE operation. Since STE deals primarily with logic expression and set theory many of the symbols normally associated with both fields are used. The notation of both first order predicate logic (FOPL) [124] and standard set operations apply. Of particular applicability is the intersection ($\cap$) operation.

STE is governed by set membership rules in which solutions are described by intersections of non-empty sets. Each possible solution, called an estimate [12], is a member of the overall solution set. The estimates in the solution set may potentially be separated into many different sets.

The rules describing what characteristics the correct solution must possess are encoded into property sets. Since the outcome of a problem is a reflection of the input to the process, property sets contain the estimates that can be shown to cause the desired output, given the characteristic the property set encodes. Estimates can be parsed into sets by rules that define the desired characteristics of the property sets. These rules are expressed as assertions in STE. 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 (x), this output is expressed as

$$O_x = A(x). \tag{2.1}$$

Taking the inverse of the assertion gives the members of the property set that are used in STE operations. Intersecting the solution set with the property sets produces a new solution set. Repeated application of the STE algorithm yields a final set of acceptable answers to the problem.

Obviously, different inputs can result in different sets of estimates. Solutions, however, must display the same properties. A correctly described solution to a problem will be composed of one or more property sets that have already been parsed to distinguish between correct and incorrect solutions. A problem may have a set of equally acceptable solutions or it may have a single ($\exists !$) solution estimate.

A simplified example of STE application to the field of cryptography is when a key applied to an input of ciphertext either results in a decryption in English or it does not. Based on the result of the decryption, the key is either placed into the set of keys that decrypt successfully (where the decryption is understandable in English) or fail to decrypt (the decryption is not understandable in English). The composition of a property set (Φ_i) is constructed by testing the input or by pre-computing the set membership. All of the keys that are successful are said to show the "property" of Φ_i. Any set that divides estimates by properties is known as a "property set" where

$$\forall x \in M \rightarrow \begin{cases} x_i \in \Phi_j & \text{if } \Phi_j(x_i) = \text{true} \\ x_i \notin \Phi_j & \text{if } \Phi_j(x_i) = \text{false} \end{cases} \tag{2.2}$$

STE operations most often take place in a Hilbert space of m dimensions. The space is denoted as Ξ^m, where m depends on the requirements of the problem. Each estimate in Ξ^m is arranged according to a distance metric $(d < a, b >)$. Bounding the estimates in the space results in a volume that can be manipulated and altered. As estimates are eliminated from consideration, the volume is adjusted to exclude the rejected estimates. The resulting volumes can become cumbersome to manipulate, so the volume is again bounded by using another ellipsoid that is easier to manipulate. However, Optimal Bounding Ellipsoids (OBEs) are useful only if the effort required to calculate and apply the OBE is less than the effort to manipulate the space by simplifying the problem and avoiding the OBE entirely.

Witsenhausen and Schweppe felt that employing a Hilbert space in STE was more natural because the problems they were interested in were vector based. Using Hilbert spaces is also advantageous in that the solution process can be easily visualized. Thus Hilbert spaces have become the standard setting for STE problems.

Both Witsenhausen and Schweppe noted that STE could deal with errors, or "noise," in a problem. Schweppe [24] showed that, while noise could obscure the solution, if the noise was bounded STE was still able to return the correct answer. Another suggestion Schweppe made was to use a bounding ellipsoid in which the correct state is guaranteed to be found. Schweppe explained that the introduction of the bounding ellipsoid may have introduced errors if the solution space was smaller than the bounding ellipsoid. Witsenhausen suggested iterative bounding ellipsoids be chosen to minimize the total error during processing [23]. Early attempts at selecting bounding ellipsoids introduced enough error to indicate that further research was needed to optimize the method. Additional study was required to find techniques to ensure that bounding ellipsoids came as close as possible to the actual volume of the solution set while still minimizing introduced error.

Following the lead of Witsenhausen and Schweppe, others soon found applications for the STE technique. In fact, STE has seen successful application in diverse fields ranging from speech, signal processing, medical imaging, and image projection problems [119, 120, 121, 122]. One field that has greatly benefited from STE is that of class identification. STE has been applied not only to system analysis, but also to regular language analysis, communications [125], discernibility, and connectedness problems.

While STE can be very effective, the methodology as usually implemented suffers from several shortcomings. These shortcomings include:

1. STE efficiency is highly dependent on the distance metric used. Choosing a suboptimal distance metric can slow operation and increase the error introduced in each application of property sets;

2. While conceptualizing estimates in a volume aids in solving the problem, the mathematics resulting from operating on the volume can be complex and overly difficult;

3. Optimal Bounding Ellipsoid (OBE) algorithms and calculating new volume limits decreases computational efficiency and adds processing time;

4. It is undesirable to add any errors (if avoidable). Repeatedly eliminating the same wrong answer adds additional effort and slows the solution process;

5. Vector spaces are not appropriate for all problems. STE, as usually practiced, is not easily applicable outside of non-vector spaces; and

6. No effective general method of selecting property sets currently exists.

This work will attempt to improve on reducing error introduced during the application of property sets, introduces the use of topological spaces in STE, and shows that OBE algorithms are unnecessary in decryption applications.

2.2.1 Property Sets

Since Hilbert space property sets operate on volumes clustered by means of a distance metric in a space, and since the nature of the space can radically affect the ease of problem solving, the choice of the space itself becomes important. Topology [9] is the study of underlying commonalities and similarities in shapes. Solutions to a problem can be said to exhibit certain characteristics or "properties." Since solutions to problems exhibit certain characteristics, problems represented as shapes can be solved in topological spaces that emphasize the desired characteristics of these problems. Properties represent a partial solution to the problem solving for one characteristic of the solution. Not all spurious solutions can be eliminated by a particular property; however, by considering all unique properties one at a time in the solution space, it is possible to identify a set of properties that, when applied, will eliminate all spurious solutions from the solution set.

The set of all possible estimates for all possible applied inputs demonstrating a desired property is called a "property set," and is denoted by Φ_n. For an assertion (A), the property set will contain all estimates (x) such that

$$f(x) \rightarrow A. \tag{2.3}$$

In the case of encryption, the set constrains the keys (k_i) for the input values (x) in a set (M), as expressed by the following equation

$$\forall x \in M : O_{k_i}(x) \in A. \tag{2.4}$$

Multiple rules for distinguishing between different estimates can be asserted, resulting in multiple property sets in a particular solution space. For each rule asserted, there is a Φ, which represents the solutions, and a Φ^{-1}, which consists of all other points in the solution space. The units and values of members of Φ are determined by the nature of the application. For instance, a communications application may be expressed in terms of voltages; so Φ would be a range of voltages. In addition, there may be many distinct subsets of Φ that are defined by the input to a given rule. The total volume (V) of the solution in the space is defined by

$$V = \bigcup_{i=1}^{n} \Phi_i. \tag{2.5}$$

In this equation, i represents each of the property sets for the problem.

Working in Ξ^m, each set (Φ_i) is considered in turn. Each Φ_i contains only those elements that follow the rule that the assertion describes. Since Ξ^m contains all possible solutions, if a solution exists then the solution (P) must be inside the solution space and must be a member of the property set for each assertion. That is, if $\exists P$ then

$$P \in \Xi^m \tag{2.6}$$

and

$$\forall \Phi_i \rightarrow P \in \Phi_i. \tag{2.7}$$

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

$$P \in \bigcap_{i=1}^{n} \Phi_i. \tag{2.8}$$

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

As stated previously, no systematic general method is known to identify and select property sets. Describing the characteristics of a solution and experimenting with the effect of each set (known as mensuration), in combination with other identified property sets is the standard technique heuristic used at this time.

2.2.2 Topology of STE

Topology is important in set methodology. STE can be performed in a variety of spaces, depending on the needs of the problem. Many of the problems solved with STE are represented by vectors that require a vector space, such as a Hilbert space, for representation. A shape evaluated as an intersecting volume of possible solution estimates is formed. As previously mentioned, intersected volumes often form irregular shapes that are difficult to describe and operate on geometrically. Calculations can be simplified by approximating the irregular volume with a geometrically regular shape that entirely bounds the volume of interest. This is done in a topological space.

All topological spaces are applied to a space (X) and a set (T). For a space to be considered a topological space, it is necessary that:

1. $T \in X$;
2. $\emptyset \in X$;
3. $\forall X, Y \subseteq T \rightarrow X \cup Y \subseteq T$; and
4. $\forall X, Y \subseteq T \rightarrow X \cap Y \subseteq T$.

The simplest of all topological spaces follow these rules. It is important to note that for any space made up of the power set of X, called $pow(X)$, there is closure over the $\cup$ and $\cap$ operators. Thus, $pow(X)$ is a topological space.

Other specialized topological spaces exist. They include:

1. T_0 space, also known as a "Kolmogorov space." The T_0 space is a topological space in which every pair of points is topologically distinguishable. That is, $\forall x, y \in X \rightarrow \exists A | x \in A, y \notin A$ or $y \in A, x \notin A$;

2. T_1 space, also known as a "Frechet space." A T_1 space is a topological space in which any two points $(x, y \in T)$ can be separated;

3. T_2 space, also known as a "Hausdorff space." The Hausdorff space is a topological space that is separated into "neighborhoods." Neighborhoods are sets such that $\forall x, y | x \neq y, \exists U, V : x \in U, y \in V \rightarrow U \cap V = \emptyset$;

4. T_3 space, is both a Hausdorff space and a regular space (where $\forall C \subset X$ and a point $p \notin C$ have non-overlapping neighborhoods). That is, given any closed set F, and $\forall x \notin F, x \in U$, and $F \subset V \rightarrow U \cap V = \emptyset$;

5. $T_{3\frac{1}{2}}$ space, also known as a "Tychonoff space." Tychonoff spaces are function separable. That is, for $y \in F, x \notin F$, $\exists f$, a continuous function mapping to a number line such that $f(x) = 0$ and $\forall y \in F, f(y) = 1$;

6. R_0 space, also known as a "Symmetric space." Symmetric space is similar to T_1 space, but all points are topologically distinct;

7. Metric space. A metric space is a space with a global distance metric, $g(x, y)$ such that $\forall x, y \in X$, the following rules apply:

   1. $g(x, y) = 0$ *iff* $x = y$;

   2. $g(x, y) = g(y, x)$; and

   3. $g(x, y) + g(y, z) \geq g(x, z)$;

8. Vector space. A vector space is a space closed under vector addition and scaler multiplication; and

9. Hilbert space. A Hilbert space is a vector space with an inner product such that $|f| = \sqrt{< f, f >}$ turns the space into a complete metric space. A Euclidian space is an example of a specialized Hilbert space.

Bounding volumes in vector spaces (such as Hilbert space) requires the use of special geometric shapes. Bounding a volume in n-dimensions is normally done using either a hypertope or a bounding ellipsoid. Hypertopes have the advantage of more closely approximating the actual shape of the estimate volume but they suffer from complex representation due to the many small faces forming the hypertope. Bounding ellipsoids are easier to modify and represent, and are therefore more often used to track and describe the estimate volume. However, approximating an actual volume may result in the inclusion of estimates that should not be in the solution set. Therefore, keeping the volume as small as possible while still bounding the estimate volume is desirable.

2.2.3 Optimal Bounding Ellipsoids

All measurements suffer from some amount of uncertainty and introduced error [30]. Errors can be introduced from several possible sources, but the most common cause of error introduction is from sensor limitation when recoding data and the truncation of finite precision. A more difficult problem in reducing error is characterizing the probability density function (pdf) associated with a noise source. In some cases, the assumption of bounded error is warranted; but in many cases, it is impossible to know if the assumption is correct or even to verify if it is *a posteriori* [30]. However, in some cases it is possible to give absolute bounds on induced error. Knowing the bounds of the error gives rise to the use of multiple sets of solutions rather than a single ($\exists!$) solution. It is interesting to note that for some problems the introduction of the error term can actually result in a more robust solution[28].

Since Witsenhausen [23] and Schweppe's [24] introduction of STE to vector-based problems, substantial interest in the methodology has centered on error estimation. In a Hilbert space, estimates may be eliminated in non-continuous locations in the volume due to the distance metric used. The function that describes the boundaries of the volume can become very difficult to work with. In response to that problem, early research focused on bounding the estimates in the space by either a hypercube [126], hypertope [126], or an ellipsoid [25]. Bounding using either a hypercube, a hypertope, or an ellipsoid redefines the estimate space as a continuous volume for each iteration of the solution process. Bounding the estimate set and operating on the volume can introduce error by including estimates that were previously eliminated or by adding estimates outside the original volume. However, bounded error is necessary for convergence to the solution set.

Walter and Piet-Lahanier [30] authored an excellent survey of set method implementations of bounding ellipsoids and hypercube-based models for both linear parameter (LP) and non-linear parameter problems. Basic to all LP problems was the intersection of volumes bounded by hyperplanes in time data sequences.

Ellipsoids allow for a compact way to describe, or bound, a volume. An ellipsoid with its center at θ_c, and having its orientation and size given by the matrix M, is described by

$$E(\theta_c, M) = \{\theta | [\theta - \theta_c]^T M [\theta_t - \theta_c] \leq 1\}. \tag{2.9}$$

If θ_c is not unique, the ellipsoid is said to be "degenerate." A degenerate ellipsoid, bounded by two parallel hyper-planes, results in a family of bounding ellipsoids that can be constructed around the region of interest. If R_k is the region of interest, resulting from the latest measurement, and E_{k-1} is the bounding ellipsoid, resulting from the first $k-1$ measurements, then $E(\alpha)$ is the family of ellipsoids whose results contain R_k. The goal is to find an $\alpha*$ that minimizes the volume of $E(\alpha*)$. Unfortunately, algorithms for calculating the minimal bounding ellipsoid, also called the Optimal Bounding Ellipsoid (OBE), have greater than linear computational cost [27, 30, 127].

Another approach to simplify the description of an irregular volume of estimates is Orthotopic [30] Outer Bounding, which bounds a polygon with a rectangular framework. Bounds are identified by finding the values in n-dimensions for $\theta_{i,max}$ and $\theta_{i,min}$ $\forall i$. While the bounds are easier to calculate, the orientation of the polygon can lead to the inclusion of large areas that do not originally belong in the solution intersection. Variations of this method include calculating individual faces of a multi-dimensional object and the exact description of each of the vertices of that object [30]. However, neither method is as popular with set methodology practitioners as the bounding ellipsoid method.

Another set of researchers, Bertsekas and Rhodes [25], formalized the application of STE to linear time invariant problems or problems of the form

$$\dot{x} = Ax(t) + Bu(t) \tag{2.10}$$

measured by

$$y(t) = Cx(t) + v(t). \tag{2.11}$$

In these equations, $x(t) \in \Re^m$ is the state of the system at time (t); $u(t) \in \Re^m$ is the input at time t; and $v(t) \in \Re^m$ (often referred to as $\omega(t)$) is the noise encountered at time (t). A, B, and C are matrices of appropriate size and dimension for the problem. Bertsekas and Rhodes assumed that a problem in the form they presented could be expressed as a function of the input and noise, with the noise corrupting the final measurement of the input function. Further, the authors assumed that there was an ordering in a metric space for each solution to the vector that could be operated on within a bounding ellipsoid.

In general, hyperplanes can be constrained around the estimate volume, but hyperplanes may not always be parallel to each other. To respond to the problem of dealing with non-parallel hyperplanes, Clement and Gentil [127] introduced an algorithm called FHC1. This algorithm is recursive and takes multiple passes to converge to the solution set. While the bounding ellipsoid is not guaranteed to be optimal, it can be constructed more easily. The non-parallel hyperplane generally intersects the structure in more than one place. A new hyperplane, one tangential to the intersected volume, replaces the old hyperplane. The result of the action is two bounding ellipsoids that are intersected and bounded, resulting in a single, smaller volume of estimates.

Fogel [27] researched bounding ellipsoids in an environment where noise was constrained by energy. Addressing the same type of problem as Betsekas and Rhodes, Fogel attempted to identify the parameters of a linear time invariant problem. This was required since in the use of an iterative algorithm, an unknown error may cause convergence problems.

There are two approaches to parameter determination: deterministic and probabilistic. Normally, the output of the function is considered deterministically and the error component is treated probabilistically. The object of bounding is to create a set (θ) that bounds the parameter vector (S_p) such that $S_p \in \Theta$. The center of the bounding ellipsoid is considered to be the "true" S_p. The state estimator closely resembles the Kalman-Bucy filter which is known to be an optimal state estimator for Gaussian white noise [28]. Gaussian white noise is a random process ($\dot{z}(t)$) that has a zero mean and a variance of $\sigma^2 = \infty$ [128], such that for all $t \neq s$, $\dot{z}(t)$ is independent of $\dot{z}(s)$ and

$$E(\int f(t)\dot{z}(t)dt)^2 = \int f(t)^2 dt. \tag{2.12}$$

Thus, $\dot{z}(t)$ is the analog version of an independent sequence of random variables. Measured data has the form

$$y_k = \theta^T x_k + \omega_k. \tag{2.13}$$

In this equation, θ^T is a vector of the unspecified problem parameters; x_k is a vector composed of the previous inputs and outputs measured, and ω_k is the noise. This equation is an example of a parameter estimation problem common to ARMA, or "autoregressive moving average," models. ARMA [30] is a function used to predict future values of a time series based on a past history of observed values. Also referred to as the "Box-Jenkins" model, the ARMA model consists of two parts: an autoregression and a moving average. The general form of this function is

$$x_t = C + \sum_{i=1}^{p} \phi_i x_{t-1} + \epsilon_t. \tag{2.14}$$

In application, the constant term (C) is often omitted because it does not affect the moving average or regression. The term ϵ_t, the error term, represents a bounded unknown error.

Fogel showed that the noise is constrained, as demonstrated by the equation

$$\sum_{i=1}^{k} \omega_k^2 \leq F(k). \tag{2.15}$$

In this equation, $F(k)$ is monotonically increasing and places a bound on ω_k^2. Fogel also demonstrated that for ω_i, composed of Gaussian white noise, y_k converged [27]. It is, therefore, possible to make a bounding ellipsoid, with bounding noise, using this algorithm. The iterative process employed by this algorithm results in parameter estimates that are more likely to fit into an estimable boundary as a result of using the bounded error.

Following his original paper, Fogel continued his bounding ellipsoid work in a paper co-authored with Yih-fang Huang. Fogel and Huang [28] explored Multiple Input Multiple Output (MIMO) systems. Previously, STE problems had been restricted to Single Input

Single Output (SISO) problems. Thus STE was proven to be a useful technique for any problem that had bounded error.

Sinha and Kwong [129] built on Fogel and Huang's work and demonstrated that by using a technique of mathematical decomposition, the extension of STE to MIMO systems was possible. The paper focused on finding the optimal vector (θ_k^o) where

$$\theta_k^o = \bigcap_{i=1}^{k} S_i \tag{2.16}$$

and

$$S_i = \{\theta : r_i(y_i - \theta^T x_i)^2 \leq 1; \theta \in \Re^n\}. \tag{2.17}$$

Fogel and Huang also implemented an easily calculable iterative algorithm to calculate θ_{k+1} where

$$\theta_{k+1} = f[\theta_k, y_{k+1}, x_{k+1}, r_{k+1}] \tag{2.18}$$

on the condition

$$\Theta_k = [\theta : \sum_{i=1}^{k} q_i r_i (y_i - \theta^T x_i)^2 \leq \sum_{i=1}^{k} q_i; q_i \geq 0]. \tag{2.19}$$

This process is known as the weighted least squares (WLS) method. The WLS method was developed for use with a zero mean Guassian white noise ω_i component. The recursive algorithm converges under these assumptions and is a better bounded ellipsoid algorithm than was originally presented by Schweppe [25] because of its dependence on *a priori* observations.

Clement and Gentil [127] proposed a different model for limiting the volume of the bounding ellipsoid based on Fogel and Huang's work. Clement and Gentil's model uses the conditions listed in the WLS method, but additionally states that

$$A(z^{-1})x_k \quad = \quad B(z^{-1})u_k; \tag{2.20}$$

$$y_k = x_k + \eta_k; \tag{2.21}$$

and

$$|\eta_k| \leq \eta_k^m. \tag{2.22}$$

In these equations, y_k is the measured output at time (k); u_k is the unknown true input at time (k); and η_k is the bounded noise term at time (k). The terms $A(z^{-1})$ and $B(z^{-1})$ are determined using the following equations

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}; \tag{2.23}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-n}. \tag{2.24}$$

In the equations above, a_i and b_i are autoregressive parameters.

The various methods used to deal with bounding ellipsoids showed similar enough elements that Deller and Huang [35] were able to introduce a way to unify the various OBE methods. Deller co-authored a number of papers with his students concerning OBE [35, 36, 37, 39, 93], pursuing the mathematical basis by which such a theorem could be established for use with signal and voice processing. The concept of "optimal" [38] should be examined in the context of what is being optimized. Most OBE algorithms minimize the volume of the bounding ellipsoid in an attempt to reduce induced error. Deller and Huang noted that most bounded error problems are too complex for signal processing. They also noted that most general problems rely on the Combettes Abstract Model [12] and are not amenable to real-time applications.

As mentioned before, an alternative to OBEs uses hyperplane bounding to simplify the "optimal" constraint and eliminates the difficult task of minimizing a volume. At time

(t), the bounded error conditions imply two hyperplanes at

$$H_1^+ = [\theta | y_t = \theta_t^T + \gamma_t] \tag{2.25}$$

and

$$H_1^- = [\theta | y_t = \theta_t^T - \gamma_t], \tag{2.26}$$

where the error, $(\omega_{(t*)})$, is expressed/present under the condition

$$\omega_{t*} \leq \gamma_t \forall t \leq N. \tag{2.27}$$

The aim of hyperplanes is to construct a sequence of bounding ellipsoids that, at each t, enclose the polytope in some way. However, not all applications of property sets will cause a reduction to the set of all possible solutions.

In a later paper, Joachim, Deller, and Nayari estimated that less than 10% of the applied intersections will result in a reduction of the estimate volume [37]. OBE algorithms can be sped up substantially by eliminating unnecessary and computationally expensive operations. It is, therefore, desirable for a "selective updating" to occur. In selective updating, the updating procedure is followed only if a change in the bounded area is detected. A change in the bounded area occurs when

$$H_{t+1} = P_t \cap H_t \neq H_t. \tag{2.28}$$

While the process to detect a change can be costly, it is often advantageous because the detection algorithm may be much less computationally intensive than calculating and fitting the new bounding ellipsoid.

One such selective updating method was discussed by Rao [41], employing the

Dasgupta-Huang (DH) OBE algorithm and using a "forgetting factor." The forgetting factor $(1-\lambda)$ is a function of the positive gain (λ_t) that can be used to constrict or enlarge a bounding ellipsoid. If the bounding ellipsoid contains the intersection of $E(t-1)$ and a property set (S_t) then

$$\begin{aligned} E_t &= \{\theta \in \Re^n | (1-\lambda_t)[\theta - \theta(t-1)]^T P^{-1}(t-1)[\theta - \theta(t-1)]\lambda_t[y(t) - \theta^T \phi(t)]^2 \\ &\leq (1 - \lambda_t \sigma^2(t-1) + \lambda_t \gamma^2\}, \end{aligned}$$

which can be simplified to

$$E_t = \{\theta \in \Re^n | [\theta - \theta(t)]^T P^{-1}(t)[\theta - \theta(t)] \leq \sigma^2(t)\} \tag{2.29}$$

where

$$P^{-1}(t) = (1 - \lambda_t P^{-1}(t-1) + \lambda \phi(t)\phi^T(t), \tag{2.30}$$

$$\begin{aligned} \sigma^2(t) &= (1-\lambda_t)\sigma^2(t-1) + \lambda_t \gamma^2 \\ &- \frac{\lambda_t(1-\lambda_t)[y(t) - \phi^T(t)\theta(t-1)]^2}{1 - \lambda_t + \lambda_t \phi^T(t) P(t-1)\phi(t)}, \end{aligned}$$

and

$$\theta(t) = \theta(t-1) + \lambda(t)P(t)\phi(t)[y(t) - \phi^T(t)\theta(t-1)]. \tag{2.31}$$

The value and range of λ_t has a profound effect on the size of the bounding ellipsoids. In the DH OBE, the value of λ_t is chosen to minimize σ^2, $\forall t$. While the gain decreases the size of the OBE, it does not guarantee that the resulting OBE is optimized.

Since the goal is to determine when a change to the ellipsoid occurs, rather than when optimal bounding takes place, σ^2 is still useful. Using the calculation for σ^2, it is possible to determine if the bounding ellipsoid should be changed. Taking the intersection of the estimate volume and a property set may not yield a reduced estimate volume. If the set is unchanged, no change in the OBE is required. That is, if

$$\sigma^2(t+1) + \delta^2(t) \leq \gamma^2 \tag{2.32}$$

where

$$\delta(t) = y(t) - \phi^T(t)\theta(t-1), \tag{2.33}$$

then $\lambda_t = 0$ and no update occurs. Otherwise, the update to the ellipsoid has a gain of

$$\lambda_t = min(\alpha, \omega_t) \tag{2.34}$$

where

$$\omega_t = \begin{cases} \alpha & \text{if } \delta^2(t) = 0 \\ \frac{1-\beta(t)}{2} & \text{if } G(t) = 1 \\ \frac{1}{1-G(t)}[1 - \sqrt{\frac{G(t)}{1+\beta(t)[G(t)-1]}}] & \text{if } 1 + \beta(t)[G(t)-1] > 0 \\ \alpha & \text{if } 1 + \beta(t)[G(t)-1] \leq 0. \end{cases} \tag{2.35}$$

The value α is the user-chosen upper bound on λ_t, such that $0 \leq \alpha \leq 1$. Additionally

$$G(t) = \phi^T(t)P(t-1)\phi(t) \tag{2.36}$$

and

$$\beta(t) = \frac{\gamma^2 - \sigma^2(t-1)}{\delta^2(t)}. \tag{2.37}$$

It has always been assumed that OBEs have a monotonically decreasing volume.

However, some linear functions may have parameters that do not remain inside the calculated bounding ellipsoid. Values that vary outside of the ellipsoid are said to "jump." Jumping may be the result of incorrectly selected parameters for the bounding process or an input sequence that biases the ellipsoid. If the observed value jumps, there needs to be a way to increase the size of the bounding ellipsoid to include the observed data.

Enlarging the ellipsoid is accomplished by using a "rescue" algorithm [41]. Rescue algorithms do not increase the bounding ellipsoid to immediately re-include the point observed to jump, but accomplish the rescue over a short discrete period of time. Large discontinuities are handled very well by the combination of rescue algorithms and forgetting factors. This gives rise to "relaxation" techniques that allow recovery from unanticipated errors that cannot be characterized prior to observing data.

Deller and Huang proposed a unified OBE algorithm. For this algorithm, the initial conditions are

$$\theta_0 = 0, \tag{2.38}$$

$$k_0 = 0, \tag{2.39}$$

and

$$P_0 = 10^{-4} I. \tag{2.40}$$

For optimal behavior, two positive weighting sequences $\{\alpha_i\}$ and $\{\beta_i\}$ are chosen, according to the particular OBE used. The ω_t values are chosen either to maximize or minimize the size of the ellipsoid at time (t). Selective updating is also employed. So, for each time interval $t = 0, 1, 2, ..., N$, the following occurs:

1. Using the incoming data, (y_t, x_t), optimal values for α_t and/or β_t are found;

2. If there is no optimal α_t or β_t, the data set is discarded; and

3. $C_t = P_t^{-1}$, θ_t, and k_t are updated using

$$P_t = \frac{1}{\alpha_t}[P_{t-1} - \frac{\beta_t P_{t-1} x_t x_t^T P_{t-1}}{\alpha_t + \beta_t G_t}] \tag{2.41}$$

where

$$\begin{aligned} \theta_t &= \theta_{t-1} + \beta_t P_T x_t \epsilon_t; && (2.42) \\ k_t &= \alpha_t k_{t-1} + \beta_t \gamma_t - \frac{\alpha_t \beta_t \epsilon_t^2}{\alpha_t + \beta_t G_t}; && (2.43) \end{aligned}$$

G_t is the scaler

$$G_t = x_t^T P_{t-1} x_t; \tag{2.44}$$

and

$$\epsilon_t = y_t - \theta_{t-1}^T x_t. \tag{2.45}$$

Steps one through three are repeated continually until the algorithm arrives at an answer.

The factors of α_t and β_t vary between the particular OBE algorithms and input sequences that bias the ellipsoid. The α_t and β_t values for various algorithms are summarized in Table 2.1.

Despite the value of OBE, hyperplane, and hypertope algorithms in STE, the cost of analysis and design of these algorithms is high. Avoiding the computational cost of regularly applying these algorithms is highly desirable. If the error resulting from smoothing the volume using OBE, hypercubes, and hypertopes can be eliminated, execution time could be greatly reduced. The technique used in this dissertation does not rely on volumes, making the use of smoothing techniques unnecessary, and thus presents significant savings in terms of both computational time and costs. The technique constitutes an original contribution of this research.

2.2.4 Information and Shannon Theory

Implementing an STE algorithm requires both the identification and selection of property sets. Unfortunately, the selection of property sets remains more of an art than a science, relying on the user's intuition. Many of the property sets identified in this dissertation are the same data sets used by Shannon in his 1949 paper on cryptography [5].

Shannon [5] demonstrated that enough information is contained in m-grams (groups of m consecutive letters) to effect the solution of a Caesar cipher (a specialized type of ssubstitution cipher). Substitution cipher decryption methods often use letter frequency tables and m-grams to recover keys. Both approaches suggest that language statistics make good property sets. Morton [16] expanded language statistics to incorporate words and then sentence structure, providing another possible source for language property sets. Language statistics are so important to the field of decryption that collections of such statistics were published before World War II [130] and still exist in various university collections [131]. Shannon and Morton later established a standard model where the author gathers a custom set of statistics and then applies them to a decryption problem.

Shannon [5, 14] noted that every language contains redundancy. He further established that redundancy can be quantified as

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

and interpreted as the tendency of symbols in a language (λ) to be repeated. Patterns in the language that are not removed prior to sending a coded message provide an opportunity for attack [47]. Substitution ciphers have traditionally been attacked through the employment of redundancy, expressed statistically by letter, m-gram, and word frequencies [47].

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

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

where $|K|$ is the keyspace size, $|A|$ is the size of the alphabet in language λ, and R_λ is the redundancy of language λ. The quantity n is known as the “unicity distance” for the cipher in language λ.

In information theory the notation of $|x|$ means ”the size of ‘x’”. Throughout the remainder of this dissertation, this notation shall be used exclusively. In no instance is the notation used to refer to the more commonly known absolute value function.

Shannon [5] used this information to decrypt a Caesar cipher [17, 47]. In Shannon’s algorithm, the ciphertext is considered one m-gram at a time starting with a 2-gram. Then the probability of the m-gram being correct is examined. Shannon was searching for the m-gram whose probability, given the decryption key associated with the resulting plaintext, was 1 ($p(m\text{-gram}) = 1$). If the probability of a particular decryption is 1, then all other decryptions must have an equal probability of being 0. As such, any m-gram being investigated as a possible plaintext that has a $p(m\text{-gram}) = 0$, is removed from further consideration. Shannon made use of the fact that once a group of letters that cannot appear sequentially together in the language for n letters, then $\forall m > n$, no m-gram containing that n-gram will exist in the language.

Though Shannon was able to prove that his technique was viable, his technique left room for improvement. Shannon only used m-grams one letter at a time. Consequently, information that could have been gained by repeatedly applying m-grams during analysis of the ciphertext was lost. A more precise method of applying m-grams is to check for m-grams that have a probability of 0 and then remove those keys from consideration. Although Shannon only applied his technique specifically to a Caesar cipher, he did imply that the same technique could be useful in the decryption of other ciphers.

Shannon's technique was later modified to use larger sets of m-grams and increase testing precision. In the 1970s, Peleg and Rosenfeld [132] made use of larger m-grams and Bayesian [15] methods to demonstrate a break of substitution ciphers. In the approach taken by Peleg and Rosenfeld, called "relaxation," a message is broken down into 3-grams. A possible key is selected and the message is decrypted using that key. The probability that each 3-gram is correct, given the selected key, is then measured. Two letters in the key are randomly selected and their roles in the key are exchanged. The probabilistic effect to the key is again measured. If the probability that the message is correctly decrypted is higher, the change to the key is accepted. Otherwise, two new letters are randomly selected. In both cases, the process is repeated until the algorithm determines that the decryption is correct.

Peleg and Rosenfeld indicated that relaxation as a technique had a difficult time determining when to stop the decryption process. They further noted that the blank spaces between words were left in place and given as blank spaces in the ciphertext. Under these conditions, the relaxation technique results in a correct decryption about 80% of the time and requires about 5,000 characters of ciphertext. The significance of their work is that m-grams, in particular 3-grams, are useful as a language statistic for decryption by unifying iterative algorithms with statistical analysis to create high probability answers.

About the same time that Peleg and Rosenfeld presented relaxation, another researcher explained the role of redundancy in language. Although Morton [16], was not directly researching cryptography, his efforts had a great impact on it. Morton, a theologian with formal training in statistics, published his findings in a book entitled, *Literary Detection*, and called his work “stylometry.” Stylometry was developed as a method to determine the actual authorship of disputed Old Testament texts. Morton began studying the literary habits of authors and tried to apply statistical methods to “fingerprint” them.

During the investigation of such texts, Morton demonstrated that author styles do exist. He was able to identify almost forty characteristics/properties relevant to personal literary style. Properties of style ranged from the average size of a sentence (in words) to the use of articles, such as “a,” “of,” and “the.” Of special interest were his conclusions about word usage. Morton concluded that words which appear infrequently did not occur enough to use in statistical analysis and were, therefore, not significant. He suggested instead that words of smaller size gave better clues to authorship. Morton further demonstrated that style still remains valid after translation. Thus, words and sentence structures can be considered indicative of the correct content in the decryption of a literary work.

Lucks [133] studied characteristic patterns in the English language with respect to cryptograms. The combination of stylometry and the use of m-grams in words allowed Lucks to demonstrate the strength of targeting the blank spaces between words. Once the spaces between words were revealed, Lucks used a dictionary to attempt decryption. His algorithm made use of word size and the pattern of letters found in a word. As letters were uncovered, the dictionary searched for matching patterns, with known letter(s) in the right position(s). Word and letter statistics were combined to decrypt simple messages. Of particular interest was the discovery that more than half of the words commonly used in written English can be compiled into a list of less than 150 words. This fact suggests that redundancy and letter frequency are highly important to decryption.

2.2.5 Unifying STE and Information Theory via the Asymptotic Equipartition Property

It has long been thought that STE and information theory shared many similarities, but the proof needed to unify the two had been noticeably lacking. If STE is a branch of information theory, then it can be used in the field of cryptography. Using the Asymptotic Equiparition Property (AEP), it is possible to prove that STE is indeed a branch of information theory.

Prior to the presentation of this proof, several definitions are required. The definition of terms used in this section are as follows:

Definition 2.1: Side Information

External knowledge of a problem is termed "*side*" information. Side information includes language statistics, *a priori* information about the message source, and all data known about the dependencies in a key space that have been derived during the decryption process.

Side information is made up of two distinct types of information: the *a priori* information which does not change and derived information, or information derived from the encrypted message, which may change. Frequency of letters in a language, the number of characters in the language, or the redundancy of the language are examples of *a priori* side information. Derived information is obtained by applying the *a priori* side information to the input message. For instance, data about the key of a substitution cipher may disallow specific mappings of cipher text symbols to a particular plaintext letter, given the intercepted message. More than one mapping may remain as possible key mappings. While the correct mapping is still unknown, impossible mappings, identified by *a priori* side information, need not be considered. The resulting data becomes derived side information.

Mathematically, let S_a denote *a priori* side information and S_d represent derived side information. The complete collection of side information is represented by the symbol S in the following

$$S = S_a \cup S_d.$$

In general,

$$S = \bigcup_{j=0}^{m} S_{d_j} \tag{2.46}$$

where S_{d_j} is the side information derived in round j of the decryption process. Each iteration derives new data about the key. Data concerning the key becomes part of the derived side information that can be applied for the next iteration of decryption. Iterations continue until either 1) no more information can be derived from the given input message or 2) a readable plaintext message appears. In decryption, the derived side information is a combination of the remaining keys in the key space that can be the correct decryption key for the message. In this manner, an algorithm that eliminates all impossible keys until only one remains, known as the "Last Man Standing Algorithm" [134], can be implemented in side information.

Definition 2.2: Run

Mathematically, let $c_i, c_{i+1}, \ldots c_n$ be a string (compound symbol), consisting of characters in a message of size $|M|$. The string begins at position i in the message and continues for a total of $n - i + 1$ characters where

$$0 \leq i \leq n \leq |M|.$$

A run is the substring of the message ($|M|$), starting at c_i and ending at c_n.

Definition 2.3: Equivalent keys

Two cipher keys, k_1 and k_2, are said to be equivalent, if for a message (M),

$$D_{k_1}(M) = D_{k_2}(M).$$

Keys need not be equivalent for all possible messages. For example, assume the message (M) does not contain the symbols q, x, or z in its plaintext and is encrypted using a substitution cipher. Any key that contains the same mappings for all symbols except q, x, or z is equivalent to any message not including q, x, or z. Without those symbols in the message, it is impossible to differentiate other possible keys from the correct key (k_e).

To this point, it has been assumed that keys are totally accurate. But inaccurate decryption can occur and the deviation from the set of correct equivalent keys can be characterized.

Lemma 2.1: The error (e) is a measure of the number of keys that do not demonstrate the encryption property Φ_i *and is bounded.*

Proof: For any key space (K), k_e must reside in the key space for the encryption. There are $K_s \leq |K| - 1$ spurious keys, with $K_s = |K| - 1$ *iff* every key in K possesses the designated property. If a property uniquely identifies k_e, there are no spurious keys. The error must be

$$0 \leq \epsilon \leq |K| - 1$$

keys for each Φ_i.

Lemma 2.2: Property sets are independent random variables with an intersection that is no larger than the smallest property set.

Proof: Since k_e must be a member of every Φ_i, any key that does not show every property cannot be a solution. Thus it is necessary that

$$k_e \in \bigcap_{i=1}^{n} \Phi_i.$$

Unless a property set $\Phi_j \subseteq \Phi_k$, where $j \neq k$, the property sets are independent of each other. If $\Phi_j \subseteq \Phi_k$, then the property set Φ_k is subsumed and replaced by Φ_j because $\Phi_j \cap \Phi_k = \Phi_j$. Therefore, the property sets may be regarded as independent random variables. Applying each property set to the input message results in a set of keys that are valid for decryption given the message (M). As each additional property set is intersected with the results from the other property sets, the resulting set of possible keys tends to constrict. For example, $\forall j, k$,

$$|\Phi_j \cap \Phi_k| \leq |\Phi_j|$$

then

if: $\Phi_j \subseteq \Phi_k$, then $\Phi_j \cap \Phi_k = \Phi_j$;
if: $\Phi_j \supseteq \Phi_k$, then $\Phi_j \cap \Phi_k = \Phi_k$ and $|\Phi_k| \leq |\Phi_j|$;
or if: $\Phi_j \not\subset \Phi_k$ and $\Phi_k \not\subset \Phi_j$, then $|\Phi_j \cap \Phi_k| < ||\Phi_j|, |\Phi_k||$

as neither property set is a subset of the other. □

Theorem 2.3: Property sets are typical sets defined by the Asymptotic Equipartition Property (AEP)

Proof: More than one key may show the property (P). Each key that shows the property (P) is equally likely to be a solution for the decryption problem if no *a priori* side information is available. Since each key (k_j) is equally likely, the probability of any key in

Φ_i being the actual encryption key (k_e) is

$$p(k_j = k_e) = \frac{1}{|\Phi_i|}.$$

The probability of a key that does not show the property (P) of the encryption being in the property set Φ_i is 0. By selecting a part of the message and checking for the property (P), the keys for an encryption can be separated into two sets: one that has the property (P) and one that does not. Using the terminology of the Asymptotic Equipartition Property (AEP) [48], Φ_i is the typical set for the property (P).

The probability of the intersection of independent random variables is given by the product of the probabilities of each random variable [15]. If a key does not appear in one of the property sets, then the probability of it appearing as a solution for the decryption also goes to 0.

The probability of a key from the solution set appearing in the intersection of all of the property sets is given by

$$p(\Phi_0, \Phi_1, \Phi_2, ..., \Phi_{n-1}, \Phi n) = \prod_{i=1}^{m} \frac{1}{|\Phi_i|}.$$

For each property set (Φ_i), the key tested will tend to one of two probabilities

$$p(\Phi_0, \Phi_1, \Phi_2, ..., \Phi_{n-1}, \Phi n) = \begin{cases} \prod_{i=1}^{n} \frac{1}{|\Phi_i|} & \text{if } \forall i \rightarrow k_e \in \Phi_i \\ \\ 0 & \text{otherwise} \end{cases}$$

The set with non-zero probability is the typical set for the combined property sets. Each member of the typical set has the same probability of being the key used for encryption. For the message received, several keys may be equivalent and yield the same result in decryption. The error (ϵ) in a typical set consists of those equivalent keys (excluding the original encryption key). This is the definition of the typical set ($A_\epsilon^{(n)}$). Thus, STE [12] follows the AEP [48]. □

The unification of STE to information theory is a major contribution of this dissertation.

2.2.6 Assembly and Use of Property Sets for Decryption

The use of STE for decryption demands the identification and implementation of property sets. To date, there has been no systematic method used to identify what property sets should be applied to a specific problem. The reason for this is that identification of the property sets remains highly dependent on the experience and expertise of the STE method designer and/or implementor. Property sets are selected based on their ability to remove possible estimates from the set of possible solutions to the identified problem. The ease of deciding the set membership of the estimates is another factor in deciding which sets should be employed. Statistics are a good source for property sets and are the deterministic version of a probability density function (pdf).

Shannon [5] discussed his approach to decrypting the substitution cipher with a probabilistic method based on language statistics. His method involved looking at a portion of the ciphertext and attempting to match the pattern in the ciphertext to various plaintext decryptions. Using the probability of a particular decrypted plaintext, occurring in English, the mapping was accepted if, for the ciphertext, $< c_0, c_1, ..., c_n >$; for the plaintext, $< p_0, p_1, ..., p_n >$; and for key (k_e)

$$p(E_k(< p_0, p_1, ..., p_n >) =< c_0, c_1, ..., c_n >) = 1.$$

Keys are rejected *iff*

$$p(E_k(< p_0, p_1, ..., p_n >) =< c_0, c_1, ..., c_n >) = 0.$$

Shannon began with a group of m letters in a row, taken from a ciphertext, where $1 < m \leq |M|$. The grouping of letters, known as m-grams, has an associated probability, drawn from a body (or "corpus") of examples of proper language use. Each grouping of m letters, starting with the first letter in the corpus (ignoring any non-alphabetic character) is counted. The next m-gram is then formed by shifting the first letter in the m-gram one alphabetic character to the right in the text. His method notes that if any m-gram cannot occur ($p(< p_0, p_1, ..., p_n >) = 0$) for some value of m, then no n-gram with $n \geq m$ containing the m-gram is possible in the language. Thus, when $p(m\text{-gram}_{ct} \mapsto m\text{-gram}_{pt}) = 0$, the key that produced that m-gram in decryption can be removed from future consideration.

Definition 2.1: Forbidden m-gram

An m-gram that does not occur in the use of a particular language by a particular user, or group of users, is said to be forbidden.

Definition 2.2: Allowed m-gram

An m-gram that is not forbidden for a user, or group of users, and is found at least once in a corpus of communications is said to be allowed.

Shannon's elimination of keys that show forbidden m-grams is done while searching for the single m-gram, with $p(m\text{-gram}) = 1$, and is not the main thrust of the algorithm. It is, however, close to the goal of STE, where all keys eliminate any following keys that decrypt the ciphertext into a forbidden m-gram, in an attempt to key the only m-gram with non-zero probability. This similarity indicates that m-grams are candidates for property sets.

Although Shannon used the statistics for each m-gram only once, there are many

m-grams available for analysis. For example, for an m-gram of size i, applied to a message of size $|M|$, there are

$$|M| - i + 1$$

m-grams. The total number of m-grams in the message, for a range of m-gram sizes from 2 to j, are n_{mg} m-grams. The total number of m-grams in a message is given by

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

Single letters are not normally considered because there are no forbidden letters in the alphabet. Larger lengths of m-grams become difficult to work with because the possible number of m-grams for a particular m grows exponentially. For a given m in an alphabet (A), consisting of $|A|$ letters, there are $|A|^m$ possible m-grams. As the size of m increases, the number of forbidden m-grams also increases (Table 2.2). However, the increase in the number of forbidden m-grams quickly overcomes the number of allowed m-grams. The difference is so dramatic that it is easier to store the number of allowed m-grams and manipulate that data than to store the forbidden m-grams for use.

Each value for m is a different set of allowed/forbidden m-grams. The possible number of sets that could be applied is given by $|M| - 1$. A difficulty in using sets of allowed/forbidden m-grams is the possibility of biasing the data towards a particular author. As stated earlier, all authors have a “style” [16] that distinguishes one writer from another. An author’s style, if used as a set to describe general language use, can inaccurately describe the general use of that language and cause erroneous results. Empirical results indicate that style bias becomes a problem for $m \geq 8$ [134].

Letter frequencies have long been used in the decryption of various ciphers [5, 12].

Algorithm Name	Year	$\alpha_t(\lambda_t)$	$\beta_t(\lambda_t)$	Optimizes	Reason for Change
Fogel-Huang OBE	1982	$\frac{1}{k_{t-1}}$	$\frac{\lambda_t}{\alpha_t}$	μ_t^{vol} or μ_t^{tr}	Seminal
SM-WRLS	1985	1	λ_t	μ_t^{vol} or μ_t^{tr}	Relates OBE to RLS
Dual SM-WRLS	1992	λ_t	1	μ_t^{vol} or μ_t^{tr}	Addresses Numerical Stability
Suboptimal SM-WRLS	1993	1	λ_t	μ_t^{vol} or μ_t^{tr}	Computational Efficiency
SM-SA	1993	$\frac{\Lambda_{t-1}}{(\Lambda_{t-1}+\lambda_t)}$ $\Lambda_t \equiv \sum_{\tau=1}^{t} \lambda_\tau$	$\frac{\lambda_{t-1}}{(\Lambda_{t-1}+\lambda_t)}$	μ_t^{vol} or μ_t^{tr}	Convergence of μ_t^{vol}
Dasgupta-Huang OBE	1987	$1 - \lambda_t$	λ_t	k_t	Convergence of k_t
Quasi OBE	1997	1	λ_t	k_t	SM Filtering Relates OBE and NLMS
MW QOBE	1997	$\{1\}_{\tau=t-k_1}^{t+k_2}$	$\{\lambda_t\}_{\tau=t-k_1}^{t+k_2}$	k_t	Optimizing Multiple Weights

Table 2.1: UOBE Algorithm Factors

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 2.2: m-gram Numbers for English

However, letter frequencies are a probability-based measure of language and cannot easily be applied to property sets. The statistical form of letter sequences is equal to the number of times a particular letter appears in a substring (known as a “run”) of the message. The number of letters appearing in the run are counted and analyzed. The maximum number of times the letter appears over a large sample of runs will reflect the probability of the letters appearing in future runs. If a letter appears in the corpus no more than n times, then a key that maps ciphertext to plaintext, so that the letter appears $> n$ times, may be eliminated. No rule yet exists for selecting run length. However, for this dissertation, a choice of the run length equal to the unicity distance was used as a starting point for empirical testing. At least n characters will have to be seen, on the average, to successfully complete a decryption with no prior knowledge of the key.

Identical letters appearing sequentially are another potential source of information in a substitution cipher environment. Few letters appear two times in a row, such as “pp”; and even fewer appear three times in a row. The patterns found in the ciphertext can then be exploited, with forbidden multiple letters being removed from consideration as part of the key.

The contents of property sets will not be identical from language to language. Different languages often have different alphabets and even those languages using the same alphabet have different letter frequencies and allowed/forbidden m-grams [134, 135]. However, while the exact content of the sets may differ, the algorithm to use them does not. Note that the set architecture of the methodology allows for replacing sets for one language with sets having a similar role in another language without having to alter the algorithm. All that is required when processing the data is that data sets for the correct language be used in the message. The algorithm used is an iterative algorithm that begins with a message (M) as the input to the decryption process. At the end of the first iteration, only the ciphertext message is available. In the second iterative round, any a

priori side information available for M is applied. Successive iterations continue until there is either no new derived information or there is a solution to the decryption. Each iteration in the decryption process can be represented by its iteration number (r). The first decryption iteration increases the knowledge about the decryption by supplying a message (M). The derived side information begins as the empty set

$$S_0 = \emptyset.$$

The result of adding the ciphertext message in the original round of information provides information on the alphabet (A) of the ciphertext and results in

$$S_1 = \{A\}.$$

The result of the first iteration of decryption is the application of *a priori* information about the type of encryption used, the language of the message, language statistics, and so forth. Side knowledge is increased so that

$$S_2 = \{\{A\}, \{Ap_1\}, \{Ap_2\}, ..., \{Ap_n\}\}$$

where Ap_k represents the *a priori* information that can be applied to M. Possible keys for each encryption comprise their own sets in S_r. No new information is derived when

$$S_{r+1} = S_r.$$

New data derived from the iteration is added to side information by either inserting a new set of data or expanding a set $\{Ap_i\}$ already in S_r.

When no new data may be directly derived from M and S, an attempt to derive new information may be made via a “guess.” Guesses assume that a piece of information is correct and decryption proceeds under that assumption. The goal is to either arrive at an “acceptable” solution or to create a contradiction between the assumption and possible solutions. Assuming a piece of information with low probability is especially effective, as it may lead to quick contradiction. Contradictions are added as new side information by expanding the *a priori* set, which potentially leads to better derived information, and ultimately a solution.

2.3 Chapter Summary and Contributions of This Dissertation

In this chapter, STE has been introduced and its development as a methodology was explored. The major interests currently surrounding OBEs and bounding hypertopes was explained. Of particular note was the computational time and costs for optimal bounding of STE problems. A basic overview of information theory was also discussed. STE was proved to be a branch of information theory using the AEP. As mentioned previously, a main contribution of this thesis is the assignment of STE to information theory which explains the natural connection between STE and its use in the field of cryptography.

This dissertation also makes the following contributions with respect to STE:

1. Demonstrates that STE is a branch of information theory. This dissertation also proves that STE follows the Asymptotic Equipartition Property. Therefore, that STE is a branch of information theory;

2. Applies STE to decryption;

3. Applies STE in a simple topological space rather than a Hilbert, or metric, space.

Populating a simple metric space with the power set of the solution allows operation on pure sets;

4. Eliminates the OBE problem in STE. A set-based space does not require mapping estimates to volumes nor require bounding, thus eliminating the calculations required for bounding. In addition to simplifying the complexity of processing, no new errors are introduced from the eliminated bounding process;

5. The elimination of the need for a distance metric. A set-based implementation does not require a mapping function nor suffer from the problems of a metric that is not well suited for selected property sets;

6. The identification of allowed and forbidden m-grams, along with the statistics of such data sets in English. The identification and application of the specific property sets in English are also new to this dissertation; and

7. The identification of a set of property sets that have been successfully used in the decryption of substitution ciphers.

Chapter 3

Developing the Corpus for Decryption

3.1 Overview

Shannons theory was developed during a time when computers were not available for general research. Data was gathered by groups of people manually reviewing, collecting, documenting, and analyzing data. Costs were extremely high and the time to complete a single study limited the quality of information that laid the foundations of their analysis. Given the advanced technology in todays computational systems, a reinvestigation of data related to the redundancy of languages may be useful. M-grams have proven to be of use in exploring decryption. To date only arbitrary m-grams have been employed. The use of words and sentences have not been considered. Words are variable length m-grams and sentences are composed of a varying number of words. Words limit the further combination of subsequent m-grams or words. Sentences made up of words also limit the possible combination of m-grams. Additionally, using semantical information embedded within words and sentences, allows keys to be isolated more rapidly.

Any information known about the problem is encoded as sets in the solution space. Each rule or constraint is represented by its own unique set. Data may be a member of more than one set, but must be in at least one set to be considered. Multiple rules can be asserted, resulting in multiple "property" sets in the solution space. For each rule asserted there is an Φ_n representing the solutions, and Φ^{-1} consisting of all other points in the solution space.

Before use each identified property set must be gathered from a source that represents the language of the message. The property sets then characterize the language from which they are taken. This section describes the data sets used in the research for this dissertation and how those sets were collected. These sets were later utilized for decryption of various cipher types. The results yielded by application of the property sets will be discussed in a later chapter.

3.2 The Corpus

Developing a solution depends on applying property sets that can gradually eliminate possible solutions. Property sets based on the statistics of a language require characterizing the language. However, language changes over time [136]. A representative sample of the language was used to collect the necessary statistics to create property sets.

Comprehensive corpora representing English are not available. Those statistics that are available, such as letter frequency, do not detail their sources [130]. Therefore, it was necessary to assemble a corpora to collect statistics for this dissertation. Set collection begins with texts known to be written in English. Samples of literature taken from the Project Gutenburg [137] library in .txt format serve as the corpus for m-grams, as well as the source for encryption. Texts were gathered to represent written English from the late 15th to early 21st centuries. Works chosen for the corpus come from both halves of each century in order to reflect language drift over time. Works from different genre are included to avoid lexicon bias. At least two works were selected from each author. Texts selected for the corpus are shown in Table 3.1. The corpus is composed of 41 texts and 294,838,109 characters. The entire collection process is automated. Dictionaries for the English language are readily available in electronic format [138]. Sentence structure is taken from Chomskys work and implemented as a basic syntactic parser [139].

3.3 Property Sets

Property sets describe the characteristics and patterns of a language. Beginning with the most general characteristics of languages, each language must be a member of the natural languages, be composed of an alphabet of symbols, and must have syntactic rules. But these sets do not have to be expressly collected, since these general sets identified can be subsumed. For instance, the natural language set used in characterizing the STE solution to decryption begins with the *a priori* knowledge that the message is written in a natural language. In this set of experiments, the language of the message was English. The English language set is entirely contained in the natural language set. Therefore, the English language set subsumes the natural language set and the set of natural languages can be ignored. Similarly, the set of alphabetic characters and syntactic rules are subsumed by m-gram sets. Therefore, these sets need not be collected.

Among the techniques that Shannon explored is the use of language regularity that appear as word repetition and patterns. The premise is that these regularities can be statistically characterized. Since the elemental analysis is done at the character level, identifying word patterns results 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.

The redundancy in a particular language, R_L, depends on the probability density function for the language. Redundancy provides a clue to the possible role of the duplicated symbol or collection of symbols. 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. Cryptographers have used this method for many years in guessing keys to encrypted messages. Overall, the probabilities have proven very useful for decryption. Some of these measures include letter and word frequency, word size, and combinations of letters. Shannon empirically determined the redundancy of the English language [14] and specified it as a lower limit. Further refinement of the redundancy has not included restrictions placed on letter order by considering word and sentence constructs.

Letter redundancy can be measured either for part of the message or for the entire message. The global redundancy has the advantage of more closely representing average language use but may be inaccurate for small messages. Redundancy over a portion of the message reflects word patterns and is better for smaller messages and message fragments. This dissertation makes use of both sets. Global frequency figures were gathered by reading each character from the corpus and keeping count of the individual characters read, as well as the total number of characters read. Letter frequency is then calculated.

Information on letter frequency for portions of the texts, called "runs, were collected for a run of r characters. For each of the texts in the corpus the first r characters were read and the number of appearances for each letter was counted. At the conclusion of processing the run, the pointer in the file was incremented and the next run was processed. Processing continued as long as a run remained for processing. At the end of run processing, the maximum count for each letter is saved. The maximum number of appearances in a run is related to letter frequency and provides information on letter appearances. Letters that appear quite frequently can be separated from those that appear only infrequently.

In his work on secrecy systems, Shannon uses the adjacency of letters called m-grams for decryption [5]. M-grams are a run of m letters in a row that appear in a text of length L. Because there are $(L - m - 1)$ m-grams in a section of code being processed, and m-grams with $2 \leq m \leq L$ are possible, a great deal of information is available about a ciphertext. Shannon uses m-grams as the basis for his decryption methodology for the shift cipher, a subset of the substitution cipher. Starting at an arbitrary position in a ciphertext, Shannon begins by analyzing the resulting text from decryptions using each of the keys for the cipher. Shift ciphers are an easy case, because there are only 26 possible keys, and are easy to illustrate for the same reason. The resulting plaintext from each key is checked for the probability that it is a (the) key that produces a readable message. Because some combinations of m-grams do not result in an understandable message, the probability of one of the keys producing the correct message converges to 1. Other keys will produce plaintext whose probability of being correct converges to 0. Additional symbols are added and analyzed until the probability of one of the resulting plaintext streams becomes 1. The probability of an m-gram occurring is measured empirically by actually counting the number of occurrences in a representative corpus. Prior to Shannons work, cryptographers used compilations of empirical statistics on the subject [130]. It would be easier to calculate a simple probability density function using Baysian statistics [15]; however, the variability of style and language usage [14] make the calculation very difficult. Knowledge about the probability of letters and m-gram frequency is important.

To train the m-grams from the corpus, the input data has all non-alphabetic characters removed. Corpus 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. For instance, if 2-grams are being trained, each set of two characters is processed, beginning with the first character. Processing consists of recording that an acceptable m-gram was found and then incrementing the position in the file by one position. Any m-grams found are removed from a list of “forbidden m-grams. What remain 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. Each file in the corpus is similarly processed until the entire corpus is read. 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 increases quickly. For any m, the number of m-grams is given by 26^m. When $m = 6$, the number of 6-grams is 308,915,776. 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 which are not, allowed. The set of 2-grams is easily partitioned by noting that for an input of a 2-gram a value of 1 is returned if the 2-gram is forbidden and 0 is returned if the 2-gram is not forbidden.

$$forbidden = \begin{cases} 0 & \text{if } \ m\text{-gram} = \text{permitted} \\ 1 & \text{if } \ m\text{-gram} = \text{forbidden} \end{cases}$$

The same is true for other m-gram sets. 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, or slightly over 38 MB. Because of limitations on the amount of memory available and the effort required to store and retrieve data from such a large file, m-grams larger than $m = 6$ were not considered for use.

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

$$num_(m - grams) = \sum_{i=2}^{6} x - (i - 1) \tag{3.1}$$

With each symbol input from the ciphertext, up to five data points are added to the sum of knowledge about the decryption. Inputs may be repeated if the letters received are repeated. All languages have inherent symbol repetition. Gathering more than m-grams per input symbol helps offset data repetition. In Shannons example of the use of m-grams, only the m-gram formed from the beginning of the ciphertext was considered. Other m-grams formed from the middle of the stream are not considered. The use of midstream (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 with as few letters as possible, developing the solution as the string develops. Hirst described self-developing solutions as Polaroid, [140] referring to the self-developing camera film.

The remaining property sets used in the dissertation consist of those applied to words and sentences. Word sets are made up of two dictionaries:

1. A lexicon of words in the language of interest, (excluding proper nouns) and

2. 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 sentences composed are grammatically correct. This task is left to the last set, which is the set of grammatically correct sentences. A routine is 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.

When using a key to break a decrypted plaintext, a string of words in English that are understandable as a sentence or group of sentences must be formed. Checking the decrypted text for words requires that words are formed by the text. Partial words may be recognized, but full words will definitely be found in the dictionary for the language. It stands to reason that the number of symbols required for recognition should be close to the length of words (on the average). The average length of an English word is 4.83 characters. Finding the completed word and the beginning of the next word defines the first word and is necessary for unambiguous word identification. At least one additional letter beyond a word is required to terminate the word. The average of 4.83 symbols plus 1 symbol for bounding the first word gives an expected average of 5.83 symbols.

Sentence structure restricts which words may be placed together. By restricting word combinations, the m-grams spanning the words are limited. A full set of allowable m-grams is defined by the allowable words and sentence structure. Words and sentences give more total information than m-grams alone, but they require more symbols to initially apply. While words and sentence structure are more accurate they also require more input data. For more complex encryption methods with unicity distances longer than the average word length, words and sentences add to the effectiveness of a pure letter based, or m-gram based, approach.

3.4 Conclusion

Property sets are used in STE to represent the patterns and statistics of language. However, many of the sets are not readily available. Sets that are currently available do not include data on what examples of the language were used to create the sets. This chapter describes the sets that were used in this dissertation, the corpora from which the sets are drawn, and the procedure for deriving the property sets.

The steps of the STE method have the advantage of being very modular. Sets are formed and called as needed. Sets can be formed from any corpus or language that uses an alphabet. Modularity greatly facilitates testing and allows comparisons of the results to focus on the differences between corpora. Testing membership in each of the sets used 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, it is either complete or not at all.

Author	Title	Genre	Century of Work
Asimov	Foundation	Science Fiction	20th
Asimov	Fantastic Voyage II	Science Fiction	20th
Bacon	Advancement of Learning	Philosophy	16th
Bacon	The New Atlantis	Philosophy	16th
Boswell	Journal, Tour to the Hebredes	Travel	18th
Boswell	Life of Samuel Johnson	Biography	18th
Bronte	Jane Eyre	Gothic, Romance	18th
Bronte	The Professor	Gothic, Romance	18th
Bulfinch	Bulfinchs Mythology	Classics	19th
Bulfinch	Legends of Charlemegne	Classics	19th
Bunyan	Exhortation Peace and Unity	Philosophy	17th
Bunyan	Works of Bunyan v. 1 - 3	Philosophy	17th
Burroughs	Tarzan of the Apes	Action	20th
Burroughs	Lost Continent	Action, Science Fict.	20th
Carroll	Alice in Wonderland	Fantasy	19th
Carroll	Through the Looking Glass	Fantasy	19th
Christie	Mysterious Affair at Styles	Mystery	20th
Christie	Secret Adversary	Mystery	20th
Defoe	Robinson Carusoe	Action	18th
Defoe	Moll Flanders	Political, Adventure	18th
Dickens	Great Expectations	Political	19th
Dickens	A Christmas Carol	Political	19th
Fitzgerald	Flappers and Philosophers	Fiction	20th
Fitzgerald	Beautiful and the Damned	Fiction	20th
Grey	The Plainsman	Western	19th
Grey	Riders of the Purple Sage	Western	19th
Hume	Dialogues Cncrng Nat. Relig	Religion	18th
Hume	Principles of Morals	Philosophy	18th
Milton	Paradise Lost	Religious, Poetry	17th
Milton	Aereopagitica	Poetry	17th
O. Henry	Cabbages and Kings	Short Story	19th
O. Henry	Options	Short Story	19th
Poe	Collected Works, V. 1	Mystery, Horror	19th
Poe	Collected Works, V. 2	Mystery, Horror	19th
Scott	Ivanhoe	Adventure, Romance	18th
Scott	Kenilworth	Adventure, Romance	18th
Shakespeare	Complete Works	Historical, Poetry	16th
Stevenson	Dr. Jekyll and Mr. Hyde	Horror	19th
Stevenson	Kidnapped	Adventure	19th
Swift	Gulliver's Travels	Political Satire	17th
Swift	A Modest Proposal	Political Satire	17th

Table 3.1: Training Corpora for English

Chapter 4

Decryption of Single and Multi Byte Ciphers Using STE

1

4.1 Overview

Property sets are of little use in STE if they are not properly utilized to determine a set of solutions for the problem. If ideally applied, a property set will eliminate possible estimates each time it is applied. Some property sets are more useful when applied to sets of estimates that have not yet been subjected to analysis; others excel when relatively few estimates are left in the solution set. Knowing which property sets to employ and when further application will not yield additional results are key to the efficient application of STE.

STE is applicable for the decryption of every type of block cipher. Three commonly used ciphers were examined and used to compare the functionality and efficacy of STE when applying concepts from information theory to decryption. Ciphers specifically investigated were:

1. The Shift cipher;
2. The Substitution (S) cipher; and
3. The Permutation (P) cipher.

In this chapter, the application of the STE algorithm for the decryption of the shift, S, and P ciphers are discussed. Different STE algorithms specifically attuned to each cipher's weaknesses are presented. The performance of each of the algorithms when applied to various English texts from Project Gutenberg are reported [137]. An evaluation of each cipher decryption result is presented.

4.2 Specific Cipher Type, Algorithm Application, and Results of Testing

4.2.1 Shift Ciphers

The shift cipher [17] is a simple cipher with very few keys. Its key is a number ranging from $0 < k < |A|$. If each letter in A is numbered from 0 to $|A| - 1$, the key is added to the number of the plaintext letter in order to obtain the ciphertext. The ciphertext letter (c) is related to the plaintext letter (p) by the formula

$$c = (p + k)\ mod\ |A|$$

When $k = 3$ the shift cipher is called the Caesar cipher [17, 47]. The small number of keys in other ciphers allows the tracking of each individual key inside the key space during cryptoanalysis.

All single byte STE algorithms for cipher testing used a corpora drawn from English texts found on the Project Gutenberg website. However, the cipher tested corpora did not use the same texts used to generate the English language property sets found in Chapter 3. The general STE algorithm discussed in Chapter 3 was used and then fine-tuned using heuristic data for each cipher to form new property sets. Heuristic property sets were created through research of cryptanalysts, linguistic specialists, language statistics, and stylometry. The information in these sources allowed the characterization of language and the formation of cipher specific property sets. Individual cipher algorithms used for all the single byte ciphers discussed were specifically designed to increase decryption speed and accuracy.

Each shift cipher decryption began the same way by choosing a random key and then encrypting a message using that key (see Figure 4.1). Next, all possible keys were listed, with each key marked as "active" (not rejected). Each key is an estimate in the solution space. Then the property sets were applied, beginning with a group of characters equal to the smallest m-gram property set to be used. At the conclusion of this process, the next character from the message was added to the string being considered and the same property sets were reapplied. For each new letter in the message, the newly formed ciphertext m-grams were decrypted using each of the active keys. If the decryption of the m-gram was found in the property set (i.e. it was allowed), the estimate remained active. Otherwise, the estimate was rejected and was no longer considered as a possible decryption key. The algorithm continued until no letters remained in the message, all possible keys were rejected, or a unique solution was found (see Figure 4.2).

4.2.1.1 Empirical Results For the Single Byte Shift Cipher

The unicity distance (n) for the shift cipher is 1.3 characters [47]. Test results for the algorithm were based on the use of only the 3-*gram* and 5-*gram* sets (see Table 4.1). Test results for both 3-*gram* and 5-*gram* sets produced the same decryption results as using all m-gram sets, but reduced computational time and memory use.

A total of 4,916 tests were conducted with a decryption success rate of 95.487%. Decryption took an average of 5.55 characters (with a standard deviation of 1.42 characters) or approximately $4.27n$. The amount of the time needed to come to a solution was to short to measure using the program's time measurement function (t < 1 ms). Files that did not correctly decrypt were due to the inclusion of foreign names and locations, as well as imaginary words found in the text. Such words are interpreted as invalid in the data sets. Adding a corpus of names, words and locations will solve this decryption problem.

Though the shift cipher is not considered to be a secure cipher, the results prove that the STE approach is viable for shift cipher decryption. It is not, however, cryptographically interesting since the small key space of the shift cipher can be easily defeated by a brute force attack. Although other ciphers with a substantially larger key space can be attacked using brute force, the amount of time it would take to apply each key makes the attack infeasible.

Measure	Results
Mean	5.55 characters
Std Dev	1.42 characters
No. Tests	4916
Correctly Solved	95.487%

Table 4.1: Shift Cipher Results

4.2.2 Substitution Cipher

Shift ciphers are a simple form of a general substitution cipher. Substitution (S) ciphers map a character $c \mid c \in A$ to a symbol set A'. The sets A and A' may be identical or they may be different. The only requirements for the sets are that a unique mapping exists, where $A \mapsto A'$ and $|A| \leq |A'|$. In English, S ciphers have a key space of 26! keys, 25! more keys than the shift cipher. However, substitution ciphers do not disguise language statistics.

4.2.2.1 Theoretical Basis for Single Byte Substitution Cipher

A decryption solution does not have to be found that includes the correct mapping for all letters $\in A$. Ciphers are designed to exhibit a one-to-one mapping. That is, each key maps the input message (M) to a unique ciphertext encryption, $E_k(M)$. Therefore, for each ciphertext block, $\exists! k$, resulting in the correct key. However, there are cases when several keys can yield the same decryption for a message. This is true when the message does not contain all of the letters in A. In these cases,

$$E_{k_i}(M) = E_{k_j}(M), \tag{4.1}$$

where $i \neq j$. Such keys are said to be "equivalent" for message M.

Lemma 4.1: *For an 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),

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

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. $\square$

Equivalently, this Lemma can be deduced from Wells' isomorph (equivalent key) argument: Let $x, y \in A$. Let x be a plaintext character and y be a ciphertext 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 $y_i = x_i + k_i \; mod \; |A|$ and $i \neq j$ 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 ciphertext 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; 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 ciphertext

y is:

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

As an example, let an alphabet $A \mapsto A$ using a S cipher. Further, let A={0,1,2,3,4}, and a message $M = \{11212112\}$. Therefore, $|A| = 5$ and $|T| = 2$. 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:

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

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

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

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

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

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

The number of equivalent isomorphic keys listed is 6, which, by Lemma 4.1:

$$(|A| - |T|)! = (5 - 2)! = 3! = 6$$

4.2.2.2 Testing Procedure for the Single Byte Substitution Cipher

The decryption algorithm for the single byte substitution cipher began by forming a solution matrix that mapped a ciphertext character to a plaintext character. Each applied property set sought to eliminate one or more possible mappings in the matrix. The property sets implemented for decryption of S ciphers included: letter frequency (using runs of letters) and allowed m-grams (from $2 \leq m \leq 6$). The run length of the original message determined the portion of the message on which analysis began. First, the program completed the background tasks needed to track partial keys (i.e. assembly of a solution matrix, reading in of property sets and ciphertext, and calculation of character frequency). The computational overhead portion of the program used can be seen in Figure 4.3. Then decryption took place according to the algorithm shown in Figure 4.4.

When decrypting S ciphers, each symbol was mapped independently, with the constraint that all symbol mappings were unique. A key for the S cipher was represented by a matrix of each ciphertext character to every plaintext character. The intersection in the matrix of plaintext to ciphertext characters contained the information about the mapping. A ‘1’ indicated the mapping was known, a ‘0’ indicated the mapping was impossible, and a ‘.’ meant the mapping was possible, but not yet confirmed as the belonging to the key. Each matrix row and column were only allowed to contain a single ‘1’.

Beginning with the run size of the original message, letter frequency and allowed m-gram property sets were applied. Property sets helped infer which estimates could be eliminated. For example, if the mapping for the letter ‘q’ was known, the mapping for the following character could be considerably narrowed. In formal English, the letter exclusively appearing after ‘q’ is ‘u.’ Therefore, a majority of other potential mappings could be invalidated. The algorithm continued iterating over the message until all spurious estimates were eliminated, all ciphertext had been processed, a solution had been found for the message processed to that point, or no solution was found.

4.2.2.3 Empirical Results for Single Byte Substitution Ciphers

The unicity distance for the S cipher is between 25 and 26 characters [17, 47]. Property sets used during testing included: letter run, allowed m-grams ($2 \leq m \leq 5$), and multiple letter sets. A total of 1,437 tests were run with 85.53% decrypting correctly. An average of 256 characters, or $9.85n$, were required for successful decryption. Messages decrypted in an average of 50.7 seconds. Tests run on S cipher encrypted texts are summarized in Table 4.2. S cipher texts took more time to run and required more of the message to decrypt than the shift cipher. This time difference was to be expected as the key space for a S cipher is $|A|!$. There is a large difference between the number of keys in shift and substitution ciphers (26 vs. 4.03 x 10^{26} keys). Thus the time needed to sort through possible keys increases.

The reduction of decryption accuracy between shift and S cipher testing is partially due to incomplete characterization of the language in the property sets, especially the m-gram sets. Although the language property sets used were trained with a more comprehensive corpora than seen before, they are still susceptible to deviations from language norms and introduction of new author styleship. Therefore, unless all users of a language are surveyed during the training process, some examples of language use may be missed. Thus potential errors in decryption are created. These potential errors will remain, regardless of the training used, but the property sets used in this dissertation are thought to have reduced such errors. For example, US Navy code breakers during World War II were typically able to decrypt only about 10 - 15% of any given message encrypted using Japan's JN-25 naval code according to Cmdr. Joe Rochefort, Commander of Hawaii's "Station Hypo" unit. Other factors which affected decryption accuracy included texts containing proper names, foreign names, foreign words, and imaginary words.

One very practical topic for future research is to develop an algorithm to exploit the phenomenon of equivalent keys discussed in section 4.2.2.1 based on fuzzy rather than crisp parsing of the key solution set. Mathematically this can be approached by replacing the crisp '1' and '0' entries in the mapping matrix with fractional entries $0 \leq f \leq 1$ denoting the level of confidence in a plaintext to ciphertext mapping. This idea is analogous to the method of soft output decision decoding used in block turbo error correction codes [141]. It is already known that STE easily accommodates so-called "fuzzy" decision-making techniques [12]. Such an approach can improve the decryption percentage by reducing information loss in the mapping matrix.

4.2.3 Permutation Cipher

Permutation (P) ciphers are primarily used to disguise language statistics [5, 17]. Taken as a block of n bits from a stream of bits (representing symbols from bits b_m to b_{m+n}), permutation ciphers reorder the bits in the block according to a mapping key [17]. Bits may be placed anywhere in the permuted block and may even be placed in more than one location inside of the block.

Movement of these bits results in diffusion across byte boundaries. Diffusion of information makes it difficult to collect and organize dispersed information, strengthening the security of a cipher. The P cipher is important theoretically as it is a major component of cipher mixing for block ciphers, such as the permutation substitution permutation (PSP) cipher. The PSP cipher is considered to be a strong cipher as it follows Shannon's cipher mixing formula [5]. The importance of PSP ciphers will be discussed further in Chapter 5.

4.2.3.1 Theoretical Basis for Single Byte Permutation Ciphers

P ciphers can be solved in two steps. Mathematically speaking, let a P cipher be applied to a block of bytes. $B \in A$ is an encoded representation of the character. $|B|$ is the number of

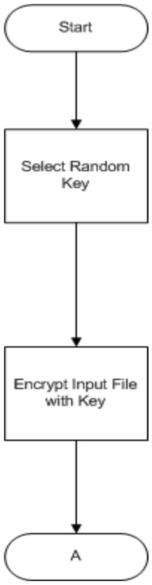

Figure 4.1: First Algorithm Step

Measure	Results
Mean	256 characters
Mean Time to Solve	50.7 seconds
No. Tests	1437
Correctly Solved	85.53%

Table 4.2: Substitution Cipher Results

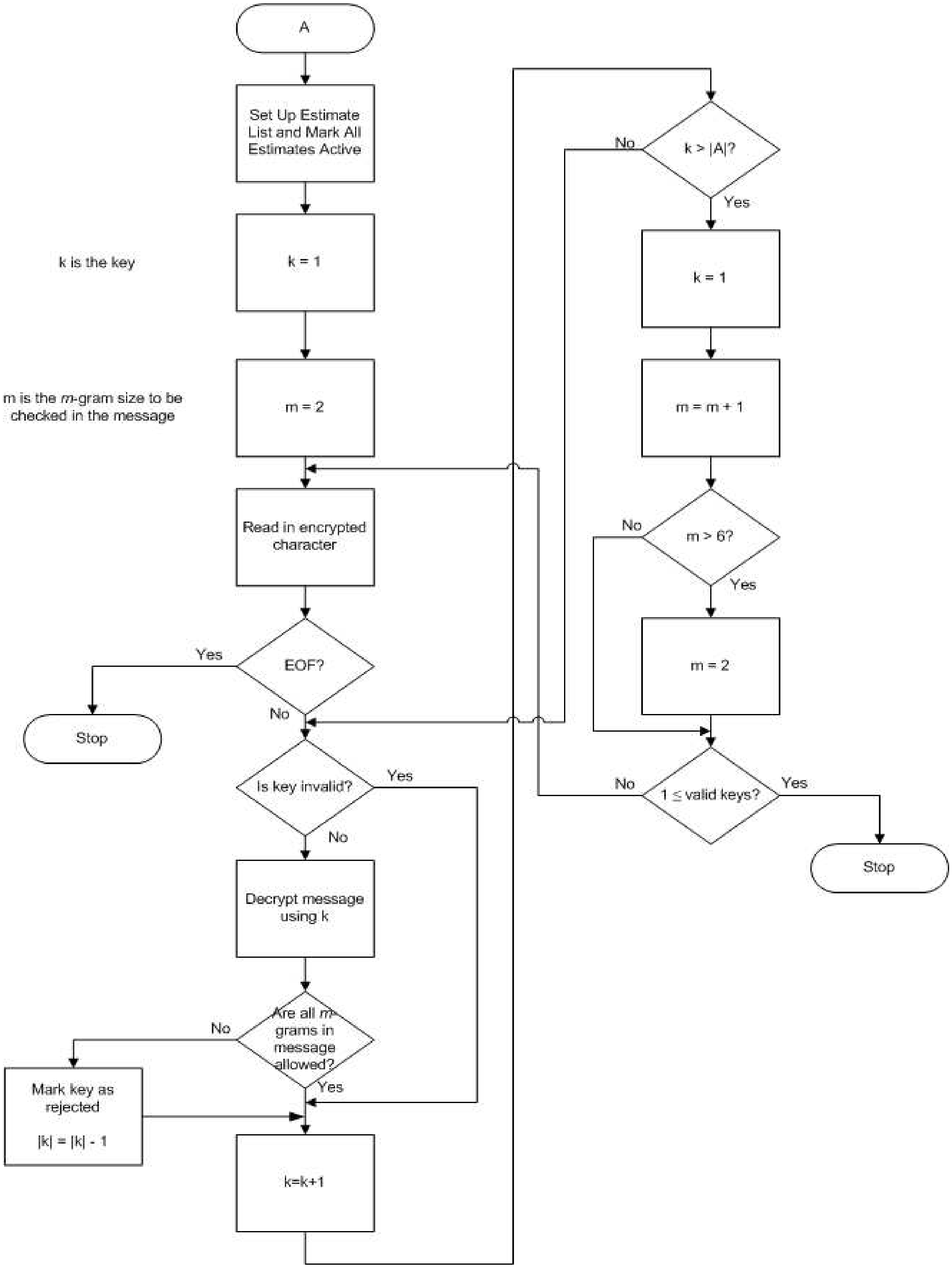

Figure 4.2: Shift Cipher Algorithm

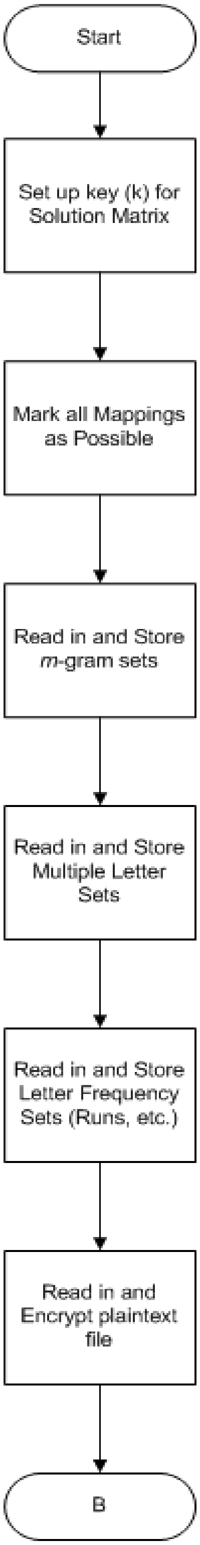
Start
Set up key (k) for Solution Matrix
Mark all Mappings as Possible
Read in and Store *m*-gram sets
Read in and Store Multiple Letter Sets
Read in and Store Letter Frequency Sets (Runs, etc.)
Read in and Encrypt plaintext file
B

Figure 4.3: Substitution Cipher Overhead

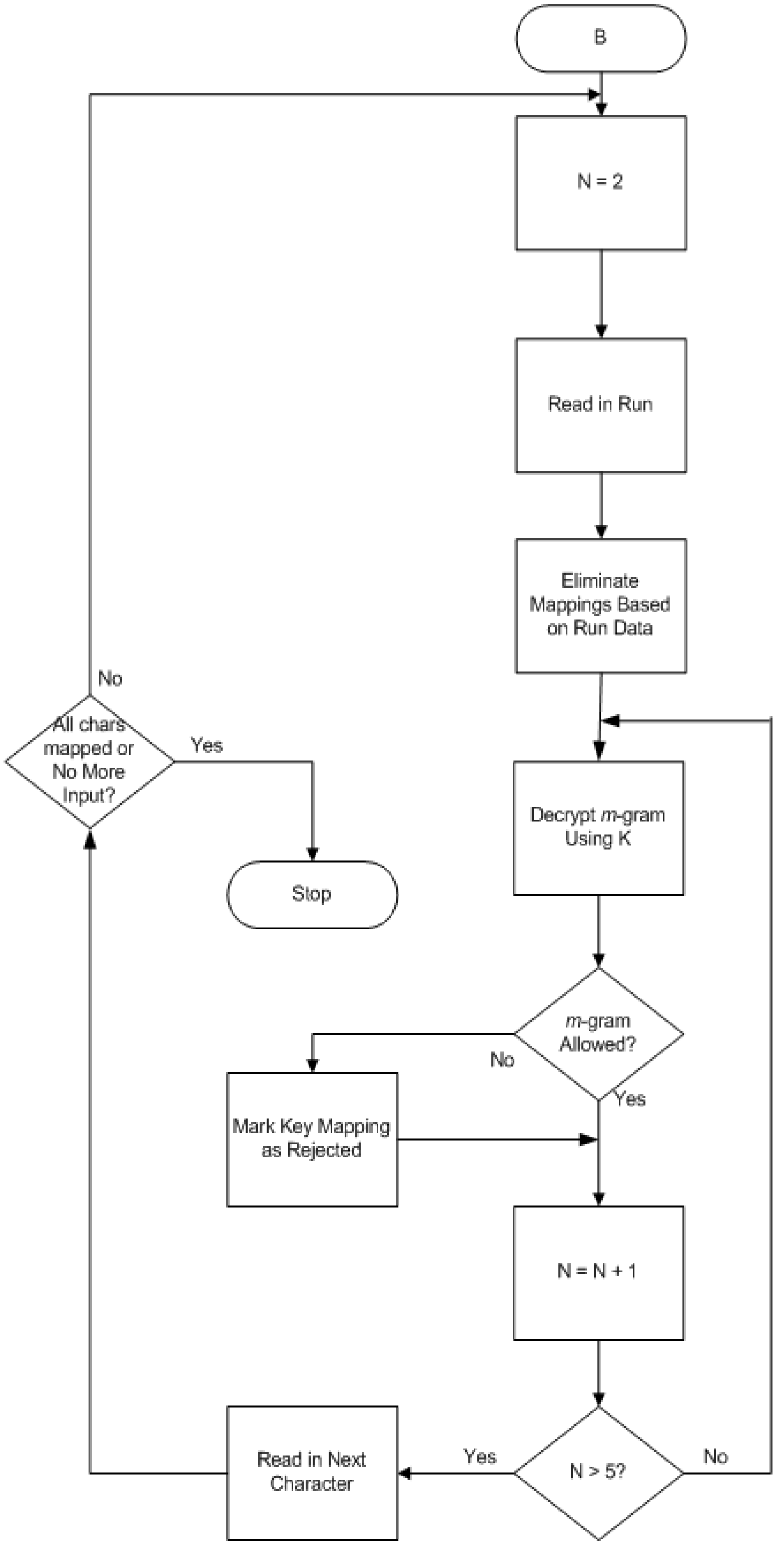

Figure 4.4: Substitution Cipher Algorithm

bits being permuted; S_t is the static bits in the block. Static bits are bits whose plaintext value never changes. Static bits may be of either bit value, determined by the particular encoding used. For example, each letter in ASCII encoding begins with the bits 110. Because permutation mappings do not change bit values, static bits remain "static" in all blocks of the ciphertext. The entropy associated with those bits is 0, since their position can be determined and absolutely known.

Mathematically speaking, let the number of static '1's in a block of message M be represented by $|1's|$ and the number of static '0's be represented by $|0's|$. Let

$$C = min(|1's|, |0's|) \tag{4.2}$$

For example, assume that a P cipher is applied to a three byte ASCII letter-only message. There would be nine static bits in each block (six '1' bits and three '0' bits). In this case, $C = 3$, the number of the '0' static bits in the block.

Theorem 4.1: *For a P cipher applied to a message (M), there are*

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

unique keys.

Proof: For a message (M), a bit at location (i) of a byte (B) is static *iff* $\forall B_x, B_y \in M, B_{x,i} = B_{y,i}$. The static bit set (S_t) is composed of the unique static bit mappings B_i. The number of remaining partial keys is $(|B| - |S_t|)!$. Depending on the combination of static bits, the static bits can reduce the number of equivalent keys. 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. The remaining $B - |S_t|$ bits can be selected $(B - |S_t|)!$ unique ways. Combining the possible choices for both the static and dynamic keys is the product of both sets of choices. Therefore, for the P cipher, the total number of unique keys is given by:

$$k = (|B| - |S_t|)!\binom{|S_t|}{C}. \tag{4.3}$$

Equivalently, while this proof follows Poincare and Lakotos' model [142] for proofs, a more formal proof follows. Theorem 4.1 can be proved using Wells' isomorph argument:

Let permutation matrix k be $|B| \times |B|$. There are then $|B|$ choices for placement of the '1' term in the first row. In the second row the '1' cannot be place in the same column as that in the first row. Therefore the number of choices remaining is $|B| - 1$. For the 3^{rd} row the number of choices is similarly $|B| - 2$. By induction, therefore, the number of permutation matrices is:

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

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

Within the original plaintext 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 as $\bar{s} = \{1...10...0\}$. For example, if $|S_t| = 5$ 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 re-arranging the locations of the '0' bits by exchanging their positions with the '1' bits. e.g.

11100

11010

10110

01110

11001

10101

01101

10011

00111

01011

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 $\bar{s}$ plaintext vectors is $\binom{|S_t|}{C}$.

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

$$\bar{y_s} = \bar{x} \left| \begin{matrix} I & 0 \\ 0 & {k'}_{22} \end{matrix} \right|$$

where I is $|S_t| \times |S_t|$, ${k'}_{22}$ 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 ciphertexts 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 4.1: *For a P cipher applied to a message (M), there are*

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

image keys.

Proof: For the P cipher, there are a total possible $B!$ keys, neglecting equivalent keys. Any isomorphic keys must be equivalent. The total number of equivalent keys is $|B|! - \binom{|S_t|}{C}(|B| - |S_t|)!$. Because the remaining keys come from the same key space, the equivalent keys are equally distributed within the isomorphic subspaces, resulting in

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

equivalent keys for each isomorph key.

Alternately, the Corrollary can be deduced more formally as follows:

The size of the total keyspace universe is $|B|!$ Within this universe the number of unique isomorph keys is $(|B| - |S_t|)!\binom{|S_t|}{C}$ by Therorem 4.1. Therefore, the number of image keys is

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

The number of image, or “spurious” keys is, therefore:

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

□

For a message (M), the key space is dependent on the characters seen in the message. Let $K_{M,c}$ represent the key space for the message using cipher (c) and K_c

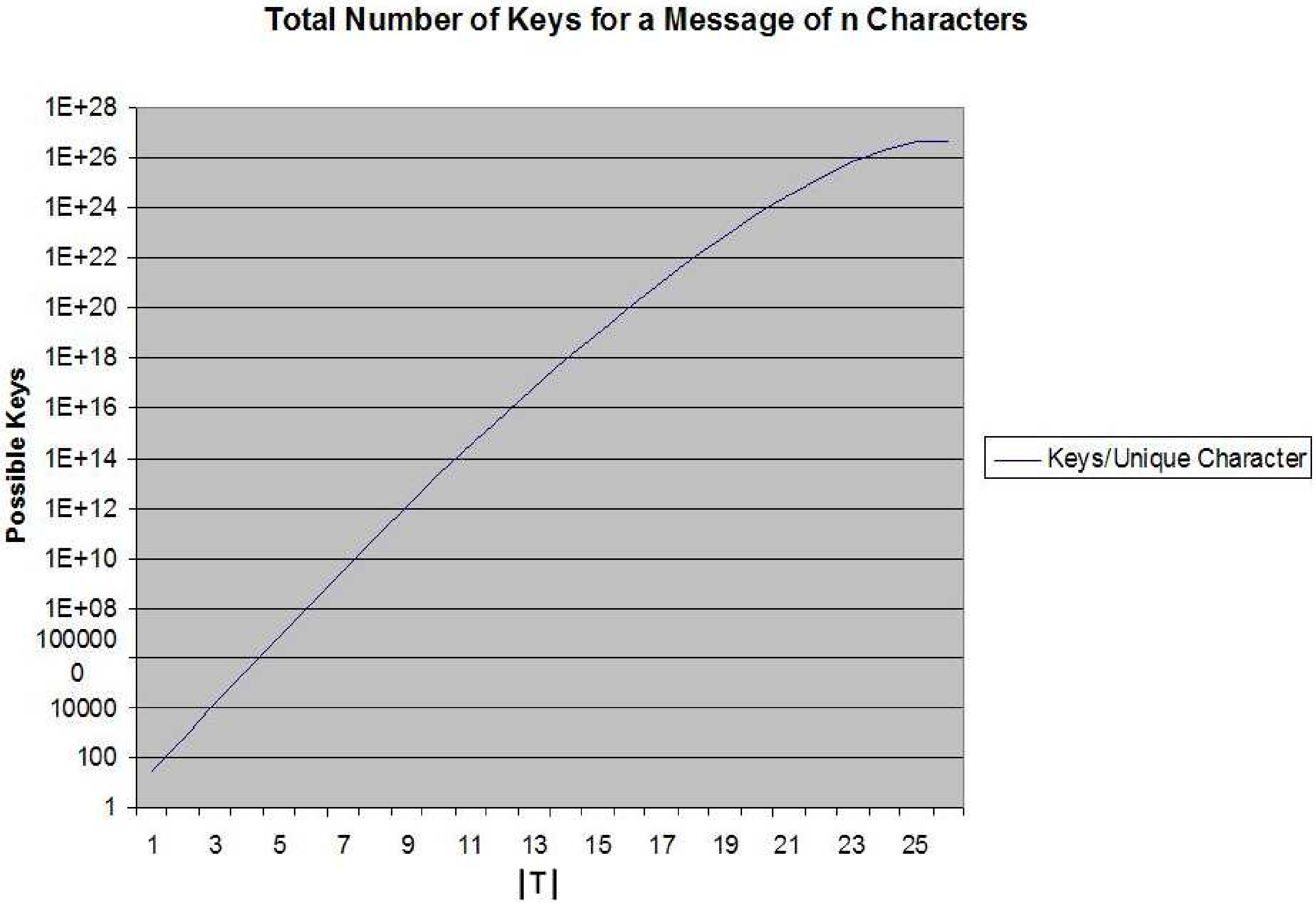

Figure 4.5: Keys per Unique Symbol Count in a Message

represent the maximum key space for a cipher (c). Then

$$|K_{M,c}| \leq |K_c|. \tag{4.4}$$

The unicity distance for a message is

$$n_{M,c} = \frac{log(|K_{M,c}|)}{R_\lambda log(|A|)}. \tag{4.5}$$

By extension

$$n_{M,c} \leq n_c = \frac{log(|K|)}{R_\lambda log(|A|)}. \tag{4.6}$$

4.2.3.2 Testing Procedure for Single Byte Permutation Ciphers

A corpus of 600 texts downloaded from the Gutenberg Project in the 2003 download CD was used for testing the permutation decryption algorithm.

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 ciphertext 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 different content, may not reveal enough information to be decrypted.

As described in the last section, the first step in decrypting the P cipher was to find and map the static bits. The time required to find the static bits depended on the redundancy of the message. Static bits in a P cipher can be found by using a modified intersection. For two blocks, B_i and B_j, and any bit n in those blocks (denoted by $B_{i,n}$ and

$B_{j,n}$), define a template vector $B_s = B_i \hat{\bigcap} B_j$, where:

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

If plaintext letter distribution is random, then maxentropic non-static bits have a 0.5 probability of each character changing value. Hence, on the average, every dynamic bit changes value at least once in each $log_2(|B| - |S_t|)$ bytes. However, language redundancy reduces the probability of bits varying from byte to byte. The empirical testing on static bit identification was never more than two bytes over the lower limit. Therefore, the cost of finding the static bits (c_{Sb}), in bytes, was always in the range of

$$log_2(|B| - |S_t|) \leq c_{Sb} \leq log_2(|B| - |S_t|) + 2 \tag{4.7}$$

The second task was to find the key for the non-static bits (see Figure 4.6). Reordering the bits in the encrypted block did not change the bit values in any way. Therefore, the number of bits with a value of '1' (and consequently, the number of bits with a value of '0') remained constant in each byte during the encryption process. Each decrypted byte needed a minimum of four, but not more than seven, '1' bits to decrypt into readable text. Using the knowledge of the encrypted block size, it was then possible to count the total number of '1' bits and partition letters into sets based on the number of '1's in their encoded byte. Sets were identified by the number of '1's (n) and was denoted as L_n. This property set was called the "Number of Ones." For any two bytes in the message (B_a and B_b), each key (k_i) was then checked and retained *iff*

$$D_{k_i}(B_a) \in L_a \ \ \&\& \ \ D_{k_i}(B_b) \in L_b \tag{4.8}$$

Following the number of ones set, the allowed n-gram set was applied to further reduce keys resulting in impossible mappings.

At the beginning of the intersection process, the number of keys eliminated was very high. As the process continued, reduction of the number decayed exponentially and the progress in key reduction slowed. Let m be the estimated slop of the reduction and note $m < 0$. The algorithm searched for a point where the slope began to flatten and the magnitude of the slope fell below $|m| = 1$. Termination for the stated slope condition works well for a majority of keys. However, some keys display a much shallower slope and the termination condition stopped the process too soon. An arbitrary decision was made to also require that less than 50,000 possible keys also remained in the solution set S. Consequently, the new termination condition became

$$(|m| < 1) \wedge (S \leq 50,000).$$

The next group of sets applied were the forbidden n-grams (see Figure 4.7). By taking the observed ciphertext and possible remaining keys, the ciphertext was decrypted byte by byte according to the key (see Figure 4.8). Each decrypted block was then compared against the list of forbidden 3-grams. If the decrypted block was forbidden, the key was eliminated from consideration. If the decrypted block was allowed, the decrypted text seen to that point was analyzed for forbidden m-grams. If one was found in the run, the key was eliminated from consideration. Keys were applied and eliminated until one or zero keys remained (see Figure 4.8).

In summary, the accuracy of the results depended heavily on the composition of the forbidden n-gram sets. Proper nouns representing the names of persons and places caused keys to be incorrectly eliminated from consideration. Relaxation increased the accuracy of the final result while only adding a small number of additional blocks during the

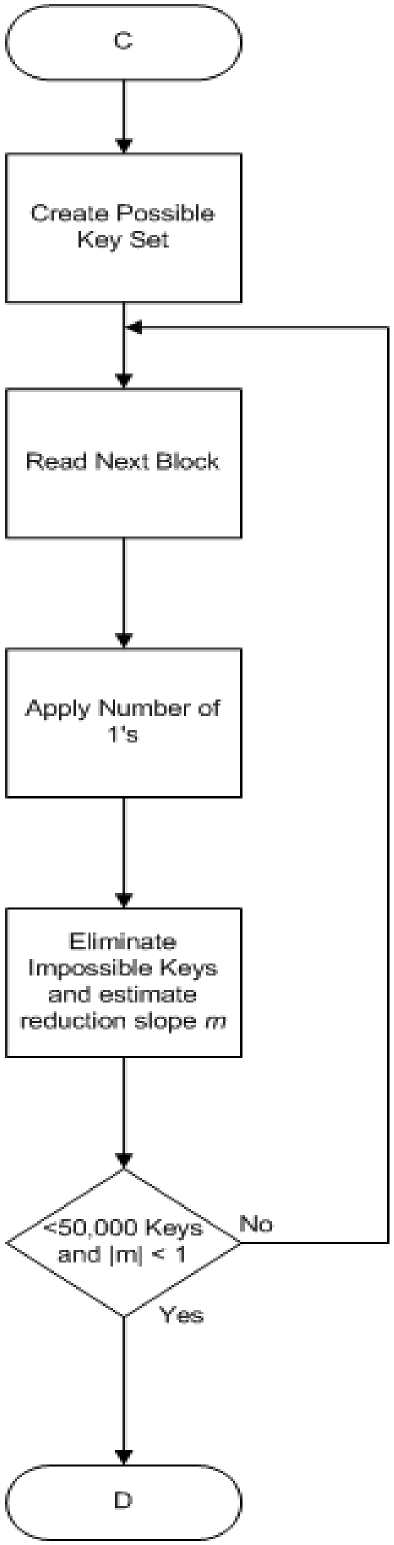

Figure 4.6: Permutation Cipher Algorithm, Part 1

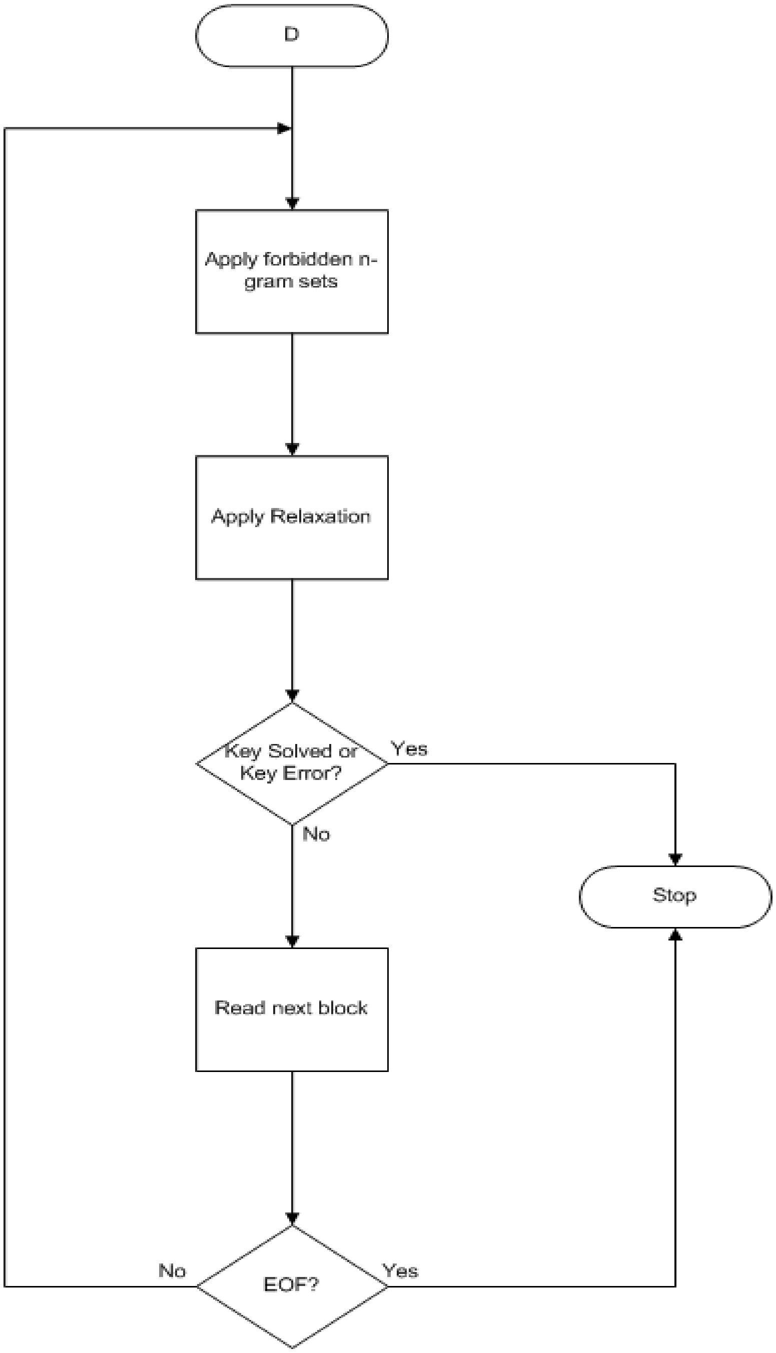

Figure 4.7: Permutation Cipher Algorithm, Part 2

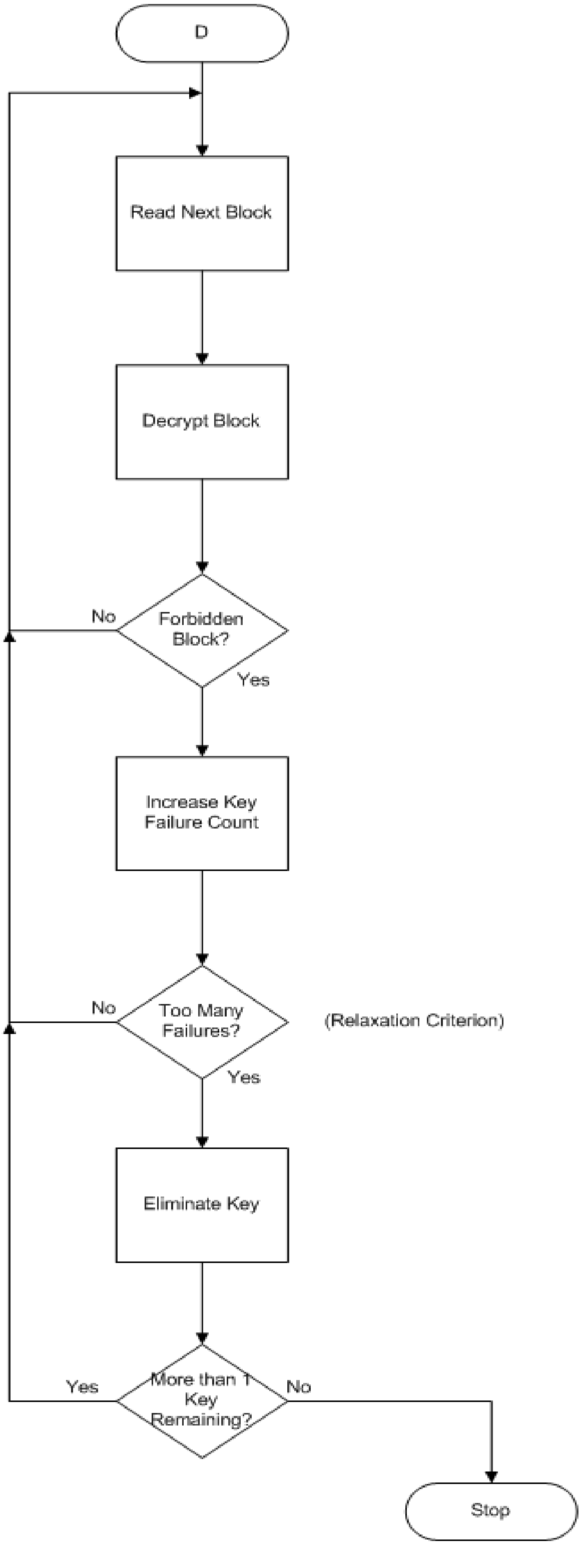

Figure 4.8: Relaxation

decryption process.

Throughout testing, the sources of error in the decryption process were noted. The correct key was not eliminated until the forbidden n-grams were applied. All of the steps leading up to the application of the forbidden n-gram sets did not address issues of style and language use. Rather, the sets dealt with the combinatorics of the alphabet in a language. Stylometry was reflected in the experiments that allowed variation in n-gram content.

The attack on the permutation cipher involved a two-step process, taking advantage of the ASCII representation of characters. Static bits are identified and solved in the first step. Solving for the static bits in the encryption is $O(lg(n))$, with possible deviation from the theoretical complexity due to redundancy in the language.

The second step found the bit mappings for dynamically changing bits. This began with finding the mappings that, given the ciphertext block of input, form ASCII letters for all of the bytes in the block. The set of mappings were then intersected with the possible mappings for the message prior to that block. When the set was small enough to be searched exhaustively, the sets of allowable n-grams were applied. Any mapping that produced forbidden n-grams was discarded. Because the inclusion of foreign languages and different author styles, the program included relaxation techniques. Relaxation allowed several failures, in terms of forbidden n-grams encountered in the text, before eliminating the mapping.

4.2.3.3 Empirical Results for Single Byte Permutation Ciphers

A single byte P cipher has a unicity distance (n) of 2.66 for English ASCII plaintext. Results for testing done on the P cipher are summarized in Table 4.3. Consisting of 1,047 decryption attempts, 99.85% of the files were decrypted successfully. An average of 19.34 characters ($7.27n$) and 0.563 seconds were needed for each decryption. Because the key space for a P cipher only has 360 keys, the unicity distance is much smaller than for a S cipher. The average number of characters required for successful decryption of a P cipher is lower than that needed for a S cipher. However, compared to the shift cipher, the number of characters and time needed for decryption was higher. This difference is again attributable to the larger key space of P ciphers compared to the shift cipher. Files that did not decrypt correctly failed due to the presence of foreign and imaginary words.

Measure	Results
Mean Time to Solve	.563 seconds
No. Tests	1047
Correctly Solved	99.85%

Table 4.3: Permutation Cipher Results

Applying the algorithm to all texts, regardless of the language of the text, a total of 1,485 decryptions were attempted (see Table 4.5). Each test randomly selected a file from the corpus, chose a random permutation key, encrypted the text, and then attempted to find the key and decrypt the text. Approximately 71.38% (or 1,060 files) of the tested files were decrypted correctly. A number of the files that could not be correctly decrypted contained non-English words, often in the form of foreign names and places. Additional files were undecryptable due to the use of non-standard English inside the text files. These files, whose texts were completely comprised of a foreign language, were considered to be controls. Thus, their "failure" was expected and confirmed the findings for single-byte ciphers. By removing files that contained entirely foreign (invalid) text, the percentage of correctly decrypted files rose to 77.82%. The files encountered, and the reasons for excluding certain files tested from consideration, are given in Table 4.4.

The unrelaxed success percentage was nonetheless greater than the percentage of the testing set that was comprised of English texts. This raises the interesting conjecture that there are some significant structural similarities between English, the Romance languages and those deriving from old Germanic, since the success percentage was almost exactly the average of the English fraction and the English plus those other languages fraction. However, this conjecture goes beyond the scope of this dissertation.

In response to the errors introduced by names, places, and non-standard language use, a "relaxation" technique was implemented. Relaxation is meant to allow a limited number of names, places, and foreign words to be processed without eliminating the ciphertext to plaintext mapping associated with the words. Assuming that the words triggering decryption failures are infrequently found in the files, each mapping is assigned a number of failures before that mapping is eliminated from consideration. The results for 0 - 3 relaxations is shown in Table 4.5. Above 3 relaxations the increased decryption accuracy falls off. Final results for the P cipher included 6 relaxations. A slight dip in accuracy occurred at 2 relaxations due to the increased number of non-standard English words found with the additional input characters read. The key was found later in the message and as a result, more of the message was input for analysis.

4.3 Chapter Summary

In this chapter, the performance of STE was presented when applied against the shift cipher, S cipher, and P cipher. STE methodology was found to be sound for decryption of the shift, S, and P ciphers. Decryption required less than $10n$ characters in all cipher cases, a lower character count than presently available using current decryption techniques [132]. Peleg reports that an average of approximately 5,000 characters (approximately $192n$) is required using his, and other, decryption algorithm.

STE-based decryption algorithms were found to be effective even when only using 3-grams as the forbidden m-gram property set. However, for each cipher tested a different algorithm was employed that used heuristic data to guide the development of the attack.

File Name	Language	Failure Reason	Number of Failures
pimil10cp.txt	English	Technical Language	2
hpaot10cp.txt	English		1
remus10cp.txt	English	Non-standard English	2
0ddc809acp.txt	Italian	Foreign Language	2
8gs3410cp.txt	English	Foreign Names and Places	3
galli10cp.txt	Latin	Foreign Language	1
rnpz810cp.txt	Polish	Foreign Language	7
mthts11cp.txt	English		1
8lndp10cp.txt	German	Foreign Language	3
anidl10cp.txt	Italian	Foreign Language	2
g1001108cp.txt	English	Foreign Names and Places	1
esper10cp.txt	Esperanto	Foreign Language	1
1mart10cp.txt	English	Insufficient Corpus Size	1
scarp10cp.txt	English	Foreign Names and Places	1
kalev10cp.txt	Finnish	Foreign Language	7
kalec10cp.txt	Finnish	Foreign Language	1
7mynr10cp.txt	Dutch	Foreign Language	6
41001108cp.txt	English	Foreign Names and Places	1
21001108cp.txt	English	Foreign Names and Places	1
8rdsl10cp.txt	English	Foreign Names and Places	1
clprm10ucp.txt	Icelandic	Foreign Language	3
8clel10cp.txt	English		3

Table 4.4: Permutation Decryption Files Failing Successful Decryption

Relaxation Errors	Tests to Decrypt	Correctly Decrypted Number	Correctly Decrypted (Percent)
0	1485	1060	71.38%
1	2239	1924	85.93%
2	749	622	80.34%
3	1600	1586	99.125%

Table 4.5: Permutation Decryption Results with Relaxation

The question then arises whether or not a single STE-based algorithm can decrypt S, P, and product (combination) ciphers. This question will be addressed in Chapter 5.

The contributions of this dissertation, with respect to the shift, S, and P ciphers are as follows:

1. Design of algorithms that applies STE to the shift, S, and P ciphers;

2. An algorithm that identifies static bits arising from the encoding of characters; and

3. An algorithm that targets and retrieves static bits.

In addition, the theorem proofs in this chapter reinforce the usefulness of the topological space paradigm upon which the STE method is based. This is particularly illustrated by the notion of isomorph texts and isomorph keys when these are likened to a "center neighborhood" in the neighborhood systems approach to topology. The notion of a "center neighborhood" is analogous to the notion of the center of an ellipsoid in OBE approaches to STE.

Chapter 5

Applying STE to Product and Block Ciphers with Diffusion Across Byte Boundaries

5.1 Overview

Most modern encryptions are product and block ciphers consisting of combinations of S and P cipher. S ciphers provide confusion and P ciphers supply diffusion. Confusion substitutes one character for another while diffusion distributes information across the encrypted message. The intent of diffusion is to disguise language patterns in the original message by spreading the pattern throughout the message. Confusion alone fails to disguise language patterns, making it susceptible to frequency and redundancy based attacks. Diffusion can easily defeat such tactics by spreading information throughout the block. Diffusion that rearranges information inside the same byte in which it is found ("bit-wise diffusion") is easily defeated. Information diffusion outside the original byte ("diffusion across byte boundaries") is much more difficult to defeat. Diffusion across byte boundaries can be attacked using the assumption that a block comprises a single character in a language, a “meta-character.” Language patterns are not disguised using the meta-character assumption.

This chapter will discuss the extension of the STE techniques used for single byte ciphers to block and product ciphers. The use of meta-characters to address diffusion across byte boundaries is explained. An evaluation of the equivalent security added by combining P and S ciphers will also be given. The subsequent application of the algorithm, testing, and results are also presented.

5.2 Review of Theory

5.2.1 Chapter Definitions

The decryption algorithm for block (product) ciphers used in this chapter is based upon an unusual parsing of symbols within a language. Defeating permutation across block boundaries requires treating the language as if the block of characters was actually a single character of a different language. The block comprises a character made up of characters, or a “meta-character,” which is part of a “meta-language.” Prior to the presentation of the material in this chapter, several definitions are required. The definition of terms used in this section are as follows:

Definition 5.1: meta-s-character

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. □

Definition 5.2: meta-s-gram (meta(s,m))

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 5.3: Meta-language

A meta-language is a language composed of meta-s-characters embedded in the same natural language. A different meta-language exists for each meta-s-character size. For example, a meta-s-character ‘*flyinsk*’ is drawn from a meta-language with an alphabet using ‘f,’ ‘l,’ ‘y,’ ‘i,’ ‘n,’ ‘s,’ and ‘k.’ □

5.2.2 Product Ciphers

Product, or combination, ciphers are ciphers resulting from the serial application of encryptions to a single plaintext message. Encryption can utilize a single symbol in the language or can be extended to a meta-s-character. In this cipher, a message is initially encrypted using one key and cipher. The resulting encryption is then re-encrypted using another cipher and/or different key [18]. Many subsequent ciphers and keys can be applied, depending on the time and resources available. Information is not restricted to the same byte of data in which it originated.For example, permutations on multi byte blocks allow for any bit in the block to be permuted to any other location inside the block, even across byte boundaries. Ciphers that diffuse data are specifically chosen so that they allow diffusion across byte boundaries. This is a much more difficult problem than bit-wise decryption. The problem is so difficult that cryptanalysts have chosen to create different attacks rather than deal with the diffusion [21, 17]. The algorithm described in this chapter deals directly with the diffusion across byte boundaries, a contribution of this research.

Product ciphers are thought to be more secure than a single byte S or P type cipher. A measure of relative security is derived from the key space (K_{pc}) [18], which is

$$|K_{pc}| = \prod_{i=1}^{n} |K_{c_i}|, \tag{5.1}$$

where K_{pc} is the keyspace of the product cipher and K_{c_i} is the keyspace of cipher c_i.

Since Shannon introduced the concept of compounding ciphers to increase security, it has been generally accepted that PSP product ciphers [5, 47, 143] are more secure than a cipher consisting of only a permutation (P) (see Figure 5.1) or a substitution (S) cipher. All product ciphers, such as the PS and PSP, are block ciphers. A property of block ciphers is that all information in the block is kept inside the block during encryption. This fact can be used to solve the problem of decryption of block ciphers. The assumption of additional security is not true for block ciphers whose encryption algorithms end at byte (character) boundaries and are encoded using ASCII (see Figure 5.2). Block ciphers with a size of $n \times c$ characters per block suffer from a significant weakness; that is, information is confined within the block.

5.2.3 Block Ciphers

Block ciphers [18] encrypt all characters in a block at the same time. Encryption can take the form of any combination of ciphers. Maurer, et al. [18] state that product ciphers have a keyspace comprised of the product of the key spaces of the constituent ciphers making up the product cipher. Larger key spaces mean a higher unicity distance unless an attack or principle (such as idempotence) exists that reduces the key space. Since all ciphers ultimately reduce to either S or P ciphers, block ciphers are either S, P, or some combination of the two.

The diffusion arising from permutation can take several forms and greatly affects the key space. Permutation in the block can take one of three forms:

1. Permutation of entire bytes (bytewise permutation).

Entire bytes within the block are reordered. Reordering is similar to a transposition cipher (which moves characters inside of bits, see Figure 5.3) [17]. Reordering bytes within the block does not sufficiently diffuse data and so is easily defeated during decryption attacks. There are only $B!$ combinations of symbols, where B is the number of bytes in the block.

2. Permutation of bits inside the same byte where they originally appear.

   All bytes consist of the reordered bits of the same byte (see Figure 5.4). All information remains inside the same byte and so is easily attacked. The number of possible mappings is limited to $(b!)^B$, where b is the number of bits in the byte and B is the number of bytes in the block.

3. Permutation within the block, but not restricted to the original byte in which they appear (bitwise permutation). Decryption of bitwise diffusion is a difficult problem [17, 47]. Any bit in the block can be placed at any bit location inside the same block (see Figure 5.5). This reorganization makes it more difficult to reassemble the bytes in the block in correct order. The number of possible mappings become $(bB)!$ where b is the number of bits in a byte and B is the number of bytes in the block. In this dissertation, all P ciphers will use bitwise permutation as it is considered the most difficult type of diffusion to break.

Treating a block of characters as if it was one character solves the problem of lost information from the original message. The meta-characters have the same characteristics as symbols in any other language, including redundancy and frequency. While there are a correspondingly higher number of meta-characters with a larger meta-character block size, the redundancy of the meta-characters decreases and becomes more uniform. But, because the meta-language behaves in the same way as the original language, it is possible to use the same techniques for block ciphers as was used for single byte (character) ciphers.

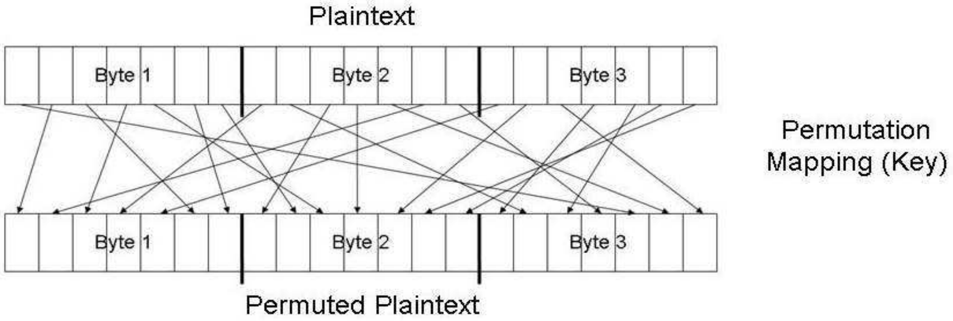

Figure 5.1: Permutation Cipher

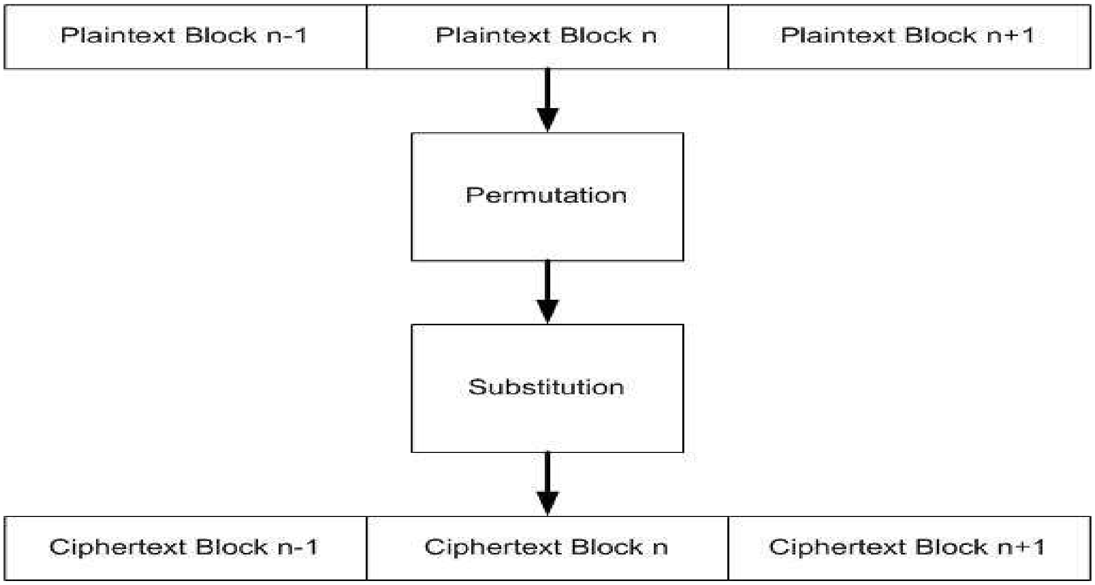

Figure 5.2: PS Type Cipher

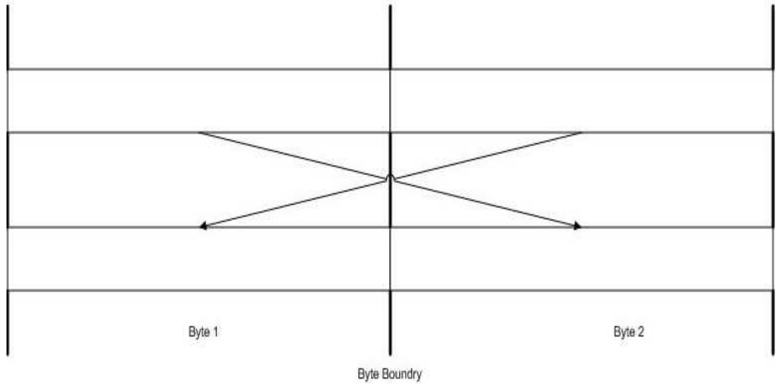

Figure 5.3: Byte Permutation

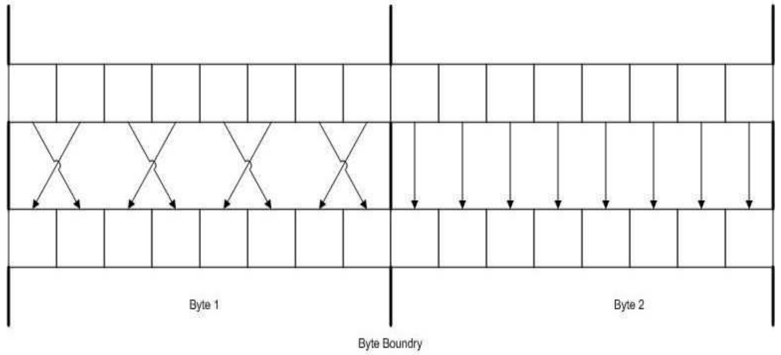

Figure 5.4: Permutation Inside Byte

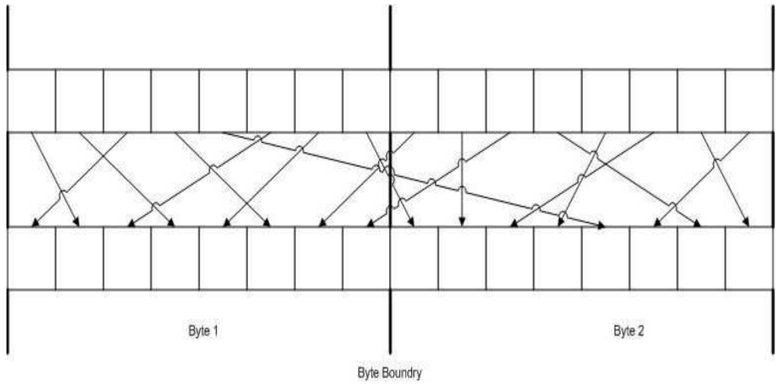

Figure 5.5: Bitwise Permutation

However, the security of block and product ciphers may not be as strong as previously thought. When Feistel introduced the Feistel round cipher (a form of PSP block cipher using permutation across byte boundaries) in 1973, he stated that all ciphers decompose into S type ciphers [19]. Using a meta-s-character whose number of constituent symbols is equivalent to the product cipher block size also results in the reduction of the P, XOR, and rotation ciphers to the S cipher. The AES cipher [17], while not composed of Feistel rounds, makes heavy use of rotations, substitutions, and XOR ciphers. The same cipher reduction applies to AES, as well. In both cases, security is limited by block size, rather than enhanced by cipher mix complexity.

5.3 PS Security Equivalence to the S Cipher

Block ciphers of PS and PSP type are composed of a P cipher and an S cipher. The size of the key space for the P cipher is $b!$ where b is the number of bits in the block [17, 47, 15]. Similarly, the S cipher maps an alphabet A to another group of symbols, A', where $A \mapsto A'$. For S ciphers, it is also possible that $A = A'$. The key space is $|A|!$, where $|A|$ is the number of symbols in the alphabet [17, 47, 15]. Substitution can also encrypt blocks of letters at a time; e.g., mapping a block of two characters to another two character group.

Combining both P and S ciphers into a block cipher using identical block boundaries results in a key space of size $b!|A|!$. When extending the block cipher to a PSP cipher, the size of the key space becomes $b!|A|!b!$.

Although the key space increases rapidly as each new cipher is added to the product cipher, it is unclear if the additional overhead involved in the encryption of a PS or a PSP cipher results in increased security. To address this question, a comparison of block substitution cipher security to the security of PSP type ciphers must be made.

Consider a case where a message is encrypted first by a P cipher and then by an S cipher. Without loss of generality, let the plaintext consist only of lower-case alphabetic characters with all spaces and punctuation removed. Assume that standard decryption assumptions apply (i.e. the message uses standard ASCII encoding, the permutation employs a three character (byte) block, and that the byte and block boundaries are known for the encrypted message). The equivalent security of the PS and S ciphers is shown in the following theorem.

Theorem 5.1: *Product ciphers of PS or SP type, aligned at character byte boundaries, provide equivalent security, in terms of greater unicity distance, to a block substitution cipher with the same block size.*

Proof: Let the number of bits in a block be represented by b_m. If the block begins at byte boundaries, then $b_m = m * e$, where m is the number of bytes in a block and e is the number of bits in a byte. The number of symbols in the alphabet is given by $|A|$. For a block of m characters, with one character per byte, the number of characters is $|A|^m$. The key space of a product cipher is the product of the key spaces of the individual ciphers [15]. The repetition in an arbitrary language, λ, is given by R_λ [5].

The product cipher PS is composed of a permutation of b_m bits. For an S cipher of m characters in an alphabet A, there are at most $|A|^m$ possible symbols [47]. Therefore, a PS cipher will have a key space of

$$|K_S| \times |K_P| = b_m!|A|^m!.$$

This provides an upper bound for the key space, since some combinations may be forbidden (i.e. are never encountered in the language) such as ‘qwz’ [134]. For the PS cipher the unicity distance, n_{ps}, is

$$n_{ps} \leq \frac{log(b_m!|A|^m!)}{R_\lambda log(|A|^m)},$$

whereas for a substitution cipher of the same block size the unicity distance, n_s, is

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

Let

$$S_R = \frac{n_{ps}}{n_s}$$

be a measure of the relative security of the two ciphers. Then

$$S_R \quad = \quad \frac{log(|A|^m!) + log(b_m!)}{log(|A|^m!)}.$$

Therefore,

$$S_R = 1 + \frac{log(b_m!)}{log(|A|^m!)}.$$

Let

$$\epsilon = \frac{log(b_m!)}{log(|A|^m!)}. \tag{5.2}$$

Then

$$S_R = 1 + \epsilon.$$

The minimum number of bits in the representation of a symbol is determined by an application of Hartley's Equation [7]. The lower bound for b_m is

$$b_m = \lceil log_2 |A|^m \rceil. \tag{5.3}$$

Substituting the minimal representation into Equation 5.4 results in

$$\epsilon = \frac{log(\lceil(log_2|A|^m)\rceil!)}{log(|A|^m!)} < 1$$

since $\forall x > 1$, $log_2(x) < x$. Therefore, $\epsilon < 1$ and decreases as m increases.

Expansion algorithms commonly used in encryption, such as DES [17], fill a block with extra permuted bits derived from the input data. The mapping is $b_m \mapsto b_{m'}$ bits, where $b_{m'} = b_m + n$. The expansion to $b_{m'}$ must be unique because $b_m \mapsto b_{m'}$ is one-to-one and onto. The number of characters that can be represented by $b_{m'}$ is $2^{b_{m'}}$; therefore,

$$\forall b_{m'} > 0 \rightarrow b_{m'} < 2^{b_{m'}}$$

and

$$\forall b_{m'} > 0 \rightarrow log(b_{m'}) < log(2^{b_{m'}}).$$

The expansion of the symbol merely maps the alphabetic character to a different symbol representation. Therefore, there is no security advantage gained by expanding the size of the character representation [48]. □

Corollary 5.1: *The relative security for a PSP cipher is given by* $S = 1 + 2\epsilon$

Proof: It can be deduced from Theorem 5.1 with the key space for a PSP cipher given by $b_m!|A|^m!b_m!$ and Equation 5.4.□

Values of ϵ (see Equation 5.4) using ASCII encoding for small block sizes are shown in Table 5.1 and Figure 5.7. Figure 5.8 illustrates the behavior of S_R as a function of block size. In particular, for blocks of size $m > 3$, the relative security S_R between PS and S ciphers is negligible.

5.4 PSP Security Equivalence to the S Cipher

Although the PS cipher has been shown to reduce to an S cipher, it is also necessary to show that the additional 2ϵ of extra security calculated in Section 5.3 is an upper bound. Through cipher reduction, it can be shown that there is no additional security added.

Definition 5.3: Cipher Reduction

A cipher C_1 using key k_i, whose encryption is denoted by $E_{k_i,C_1}(M)$, is said to reduce to cipher C_2 for a message, M, *iff* $\forall k_i, M$

$$\exists k_j | E_{k_i,C_1}(M) = E_{k_j,C_2}(M).$$

Note that each k_j remains constant for a given k_i.

Axiom 5.1: In order for cipher C_1 to reduce to cipher C_2, the range of the cipher for C_1 must be a subset of the range of the encryption function for C_2. That is $\forall M$

$$E_{k_i,C_1}(M) \in E_{k_j,C_2}(M).$$

Axiom 5.2: Any cipher C_1 reduces to itself when $k_i = k_j$.

Axiom 5.3: If cipher C_1 reduces to cipher C_2, then cipher C_1 can be replaced by cipher C_2 by using an appropriate key for C_2.

Theorem 5.2: *A permutation encryption P reduces to a substitution encryption S within the encoded representation of a symbol.*

Proof: Let the symbol s_i of an alphabet A be represented by a collection of n bits where $n = \lceil log(A) \rceil$ [7]. Let B be the set of all values represented by n bits. The values in the set B range from 0 to $2^n - 1$. Therefore, $A \subseteq B$. A permutation [17] preserves the number of '1' and '0' bits but rearranges them into another symbol. By preserving the same number of bits, $E_k(A) \in B$. A permutation results in a mapping of $A \mapsto B$, which is known to be a special case of the S cipher [17, 47]. Since a unique substitution key exists for every permutation mapping, where $A \mapsto B$ [17], P reduces to S within the boundaries of an encoded symbol. □

Lemma 5.1: *A substitution cipher does not necessarily reduce to a permutation cipher.*

Proof: P preserves the number of '1' and '0' bits whereas S may or may not preserve the number of '1' and '0' bits for all character mappings. □

There are cases when a S cipher reduces to a P cipher. An example of this happens when the symbols used in the message, M, are mapped to encrypted symbols whose number of '1' and '0' bits do not change and the mapping is consistent for the bits in all characters. This is not required by substitution since any symbol in the plaintext can map to any symbol in the ciphertext. Therefore, not all S ciphers reduce to P ciphers.

Corollary 5.2: *If cipher C_1 reduces to cipher C_2 it does not necessarily follow that C_2 reduces to C_1.*

Proof: It can be seen using the preceding lemma, that P reduces to S, but S does not necessarily reduce to P. □

Definition 5.4: A compound symbol (meta-symbol), X, is an ordered n-tuple of characters $< x_0, x_1, ..., x_i >$ regarded as comprising a single symbol (or block), where $x_i \in A_\lambda$ and A_λ is the alphabet of language λ.

Corollary 5.3: *Under the condition of symbol-byte (meta-character) boundary alignment, a PSP cipher is idempotent to an S cipher with identical block boundaries.*

Proof: Let the symbol in a block cipher be a meta-character [47] defined as being the same size and having the same boundaries as the cipher block. Further, let the S block cipher be applied to the same meta-character. As seen in Theorem 2, P and S ciphers are equivalent within a symbol boundary. Therefore, a PSP cipher reduces to an SSS cipher. S ciphers are idempotent and associative with each other [47]. Therefore, a SSS cipher reduces to a single S cipher. □

Let S_S be the relative security measure between block substitution ciphers of block size m and n where $m > n$, defined by

$$S_S = \frac{n_{S_m}}{n_{S_n}}.$$

Using substitution, it can be seen that

$$\begin{aligned} S_S &= \frac{\frac{log(A^m!)}{R_\lambda log(A^m)}}{\frac{log(A^n!)}{R_\lambda log(A^n)}} \\ &= \frac{log(A^m!)log(A^n)}{log(A^n!)log(A^m)} \\ &= \frac{nlog(A)log(A^m!)}{mlog(A)log(A^n!)} \end{aligned}$$

which ultimately reduces to

$$S_S = \frac{n}{m}\frac{log(A^m!)}{log(A^n!)} > 1.$$

For $m > n$ and $n \geq 1$, S_S will increase as m increases. Therefore, the security of a substitution cipher increases as the block size increases. The additional security as a function of block size is shown in Figure 5.6. Note that the unicity distance increases exponentially as the block size increases.

As the block size of the PSP cipher increases above two symbols, the additional security gained is insignificant when compared to a S cipher of the same block size (see Figure 5.8). Therefore, no additional security is gained by the use of a PSP cipher or product ciphers derived from it.

One tactic taken by cipher designers is to increase security by multiplying the number of ciphers in the encryption. Designers assume that each cipher in the encryption must be broken separately. Therefore each additional cipher in the encryption should result in increased security through multiplying the size of the message's key space. But the complexity introduced in the process is ineffective because each cipher in the encryption can ultimately be reduced to an S cipher, limiting the overall key space. Therefore, breaking a more complicated cipher takes the same amount of effort as breaking an S cipher of the same block size. As such, the incremental overhead in preparing and processing an encrypted message can be avoided by only using S ciphers. Cipher reduction and elimination of the effects of diffusion results in faster decryption. This finding is an original contribution made by this dissertation.

m	$log(b_m!)$	$log(A^m!)$	ϵ
1	4.605	26.6056	0.173103
2	13.3206	1621.275	0.008216
3	23.7927	66978.08	0.000355
4	35.4202	2.40×10^6	1.49×10^{-5}
5	47.91165	7.89×10^7	6.07×10^{-7}
6	61.09391	2.49×10^9	2.46×10^{-8}
7	74.85147	7.61×10^{10}	9.84×10^{-10}
8	89.10342	2.27×10^{12}	3.92×10^{-11}
9	103.787	6.68×10^{13}	1.55×10^{-12}
10	118.8547	1.93×10^{15}	6.14×10^{-14}

Table 5.1: Security for a Block of m Bytes

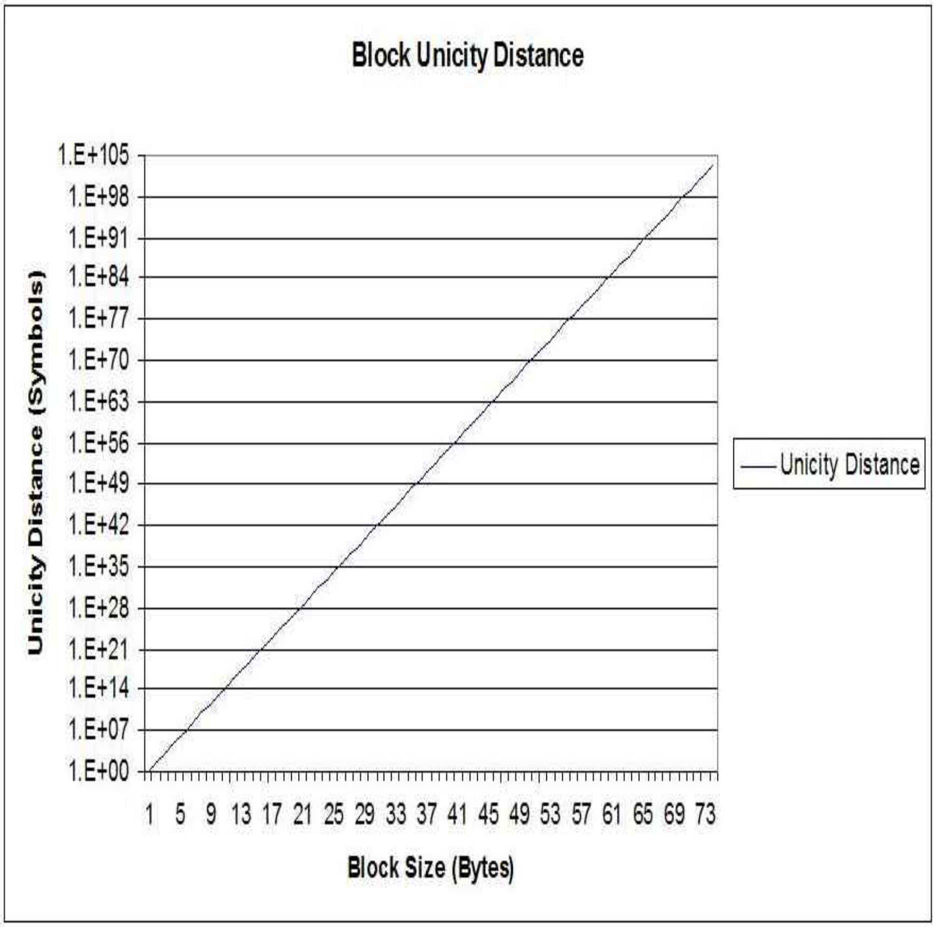

Figure 5.6: Unicity Distance for Substitution Cipher of n-Byte Blocks

5.5 Empirical Work

5.5.1 Overview

It has now been proven that PS and PSP product ciphers provide no additional security to a cipher encryption. Reduction of product and block ciphers to an S cipher indicates that it is possible to use the BCBB algorithm to break S, P, PS, PSP, and SPSP ciphers. The choice of property sets for the BCBB algorithm, the composition of the algorithm, and the results of block encrypted cipher testing are next presented.

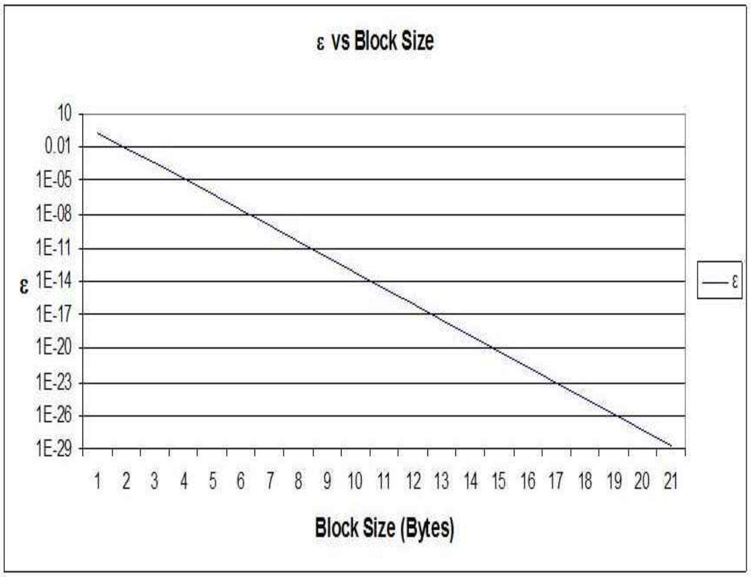

Figure 5.7: ϵ vs. Block Size (in bytes)

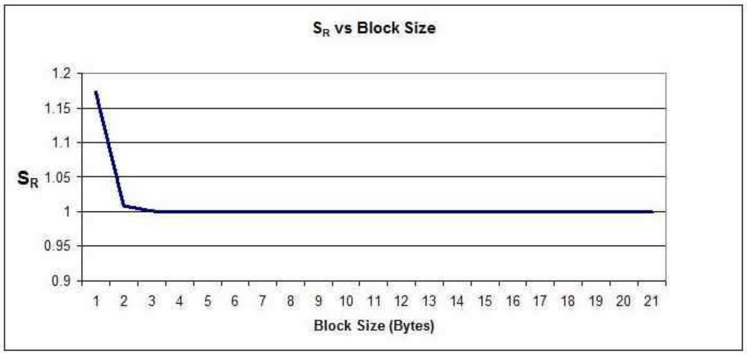

Figure 5.8: S_R vs. Block Size (in bytes)

5.5.2 Property Sets

When choosing property sets for use in decrypting multi byte product ciphers one important fact was used; all ciphers ultimately reduce to an S cipher [19]. As such, the same form of attack effective on single byte S ciphers should be effective on all types of block ciphers. And, since S ciphers do not obscure the patterns found in language, those patterns can still be utilized in the decryption process. Therefore, the same sets used for decrypting the S cipher should be valid for a general attack on product ciphers. However, data sets need to be based on meta-s-characters of the block size being decrypted. In this case, only the meta-s-character frequency and allowed meta(s,m) sets for $m = 3$ and $m = 4$ were used for decryption tests. These sets are a subset of the property sets used in the single byte cases described in Chapter 4.

The first property set applied to the BCBB algorithm for a product cipher is the global frequency property set. Global redundancy is applied only once. Global frequency is the frequency of characters in the meta-alphabet found in the message. Following the Law of Large Numbers [15], the larger the message, the more likely the meta-s-character frequency from the message will accurately reflect the meta-language alphabet frequency. meta-2-character frequencies are shown in Figure 5.9. Both high and low frequency characters are of interest in this set. A large division in the data collected can be seen between the first 10 and subsequent members of the meta-2-character set. This division is referred to as the “high frequency threshold.” The top ten meta-2-characters on the frequency list were dubbed “high frequency” meta-2-characters. When interpreting data returned from the BCBB algorithm, higher frequency meta-2-character combinations is assumed to belong to the top ten frequency set.

Similarly, the corpus identifies a set of low frequency characters. “Low frequency” meta-2-characters fell within the bottom 5% of the meta-2-character frequency list. Again, the threshold was set by finding a frequency where it is possible to distinguish between sets of meta-2-characters. Any meta-2-character occurring more frequently than the threshold cannot be mapped to any member inside the low frequency set. Thus the mapping(s) can be eliminated. Together the initial application of the global frequency set resulted in a reduction of mappings in the solution matrix.

Other property sets selected for use were the frequency of meta-s-characters and the forbidden meta(s,m) set, with $m = \{3, 4\}$. The use of meta(s,m) sets subsumes the redundancy and multiple letter sets used for the algorithm in Chapter 4, meaning no additional property sets needed to be applied.

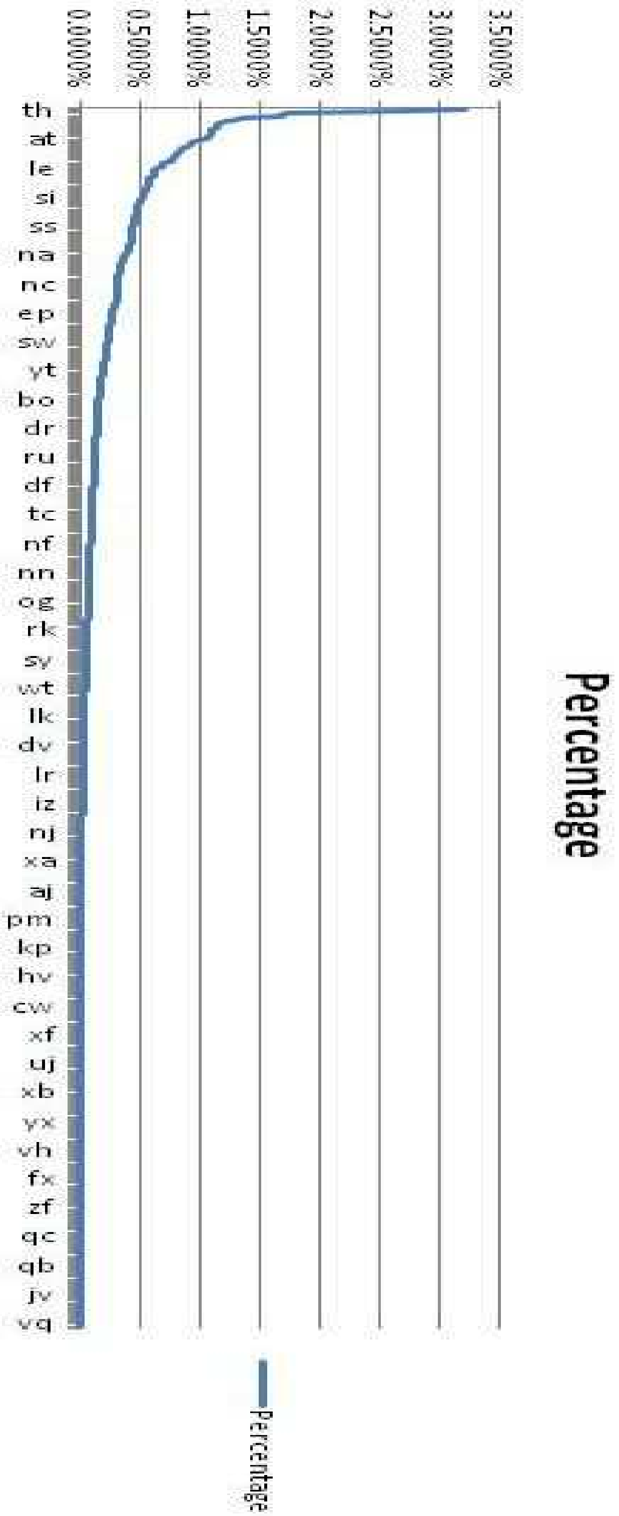

Figure 5.9: Metacharacter Percentage for Block Size 2

5.5.3 Algorithm Description for Decrypting Product and Block Ciphers (BCBB)

Tests were run on texts selected for their representation of English language from the early 19th to late 20th century. Each text was encrypted using a randomly chosen key and five different encryptions: P, S, PS, PSP, and SPSP. All tests were run on the same laptop computer running Windows 7. Code was compiled in Visual Studio 2007. Each of the texts tested is shown in Table 5.2 and results for each decryption is shown in Table 5.3. The same files were decrypted employing both meta-1-character and meta-2-character encryptions. Testing shows the difference in time required to decrypt both encryption block sizes. Table 5.3 shows the average time required, by meta-s-character size, for decrypting an average file. All decryption times were measured in seconds.

The algorithm, which is called the Block Cross Byte Boundary (BCBB) algorithm, started by reading in both the encrypted text and the data sets needed for decryption. The algorithm then set up a solution matrix of possible mappings from ciphertext to plaintext meta-s-characters. The mapping is stored by means of a hashtable that associates invalid key mappings for a particular ciphertext meta-s-character to a plaintext meta-s-character. Mappings that did not appear in the hashtable were still considered possible. Lists of meta-s-characters seen in the encrypted text and mappings found to be part of the key were also maintained. A flowchart for the setup portion of the program that handled preparation for decryption is found in Figure 5.10.

Once the solution structure was set up, the algorithm began to eliminate mappings. The procedure for mapping elimination is shown in Figure 5.11. The first property set applied was the global meta-s-character frequency data. The entire message was processed and then compared against a normalized global frequency list. High frequency meta-s-characters passing the threshold were mapped to a select set of plaintext characters

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

Table 5.2: Two Byte Block Files Used for Testing

File	S (sec)	P (sec)	PS (sec)	PSP (sec)	SPSP (sec)	Mean (sec)	STD (sec)
1linc11cp.txt	675	669	671	676	661	670.4	5.98331
1onwr10cp.txt	14507	14442	14682	14273	14185	14417.8	195.8997
alice30cp.txt	44617	44690	44386	44470	44473	44527	123.0203
1anne11cp.txt	778	770	773	774	775	774	2.9155
hoend10cp.txt	861	847	854	848	795	841	26.3154
jandc10cp.txt	1387	1381	1391	1388	1398	1389	6.2048
jmlta10cp.txt	12680	12723	12488	12616	12624	12626.2	88.7028
lglass18cp.txt	7851	7664	7828	7603	7716	7732.4	105.9448
wwill10cp.txt	765	743	750	744	750	750.4	8.792
wwrld10cp.txt	546	550	550	546	552	548.8	2.6832
Average	8466.6	8447.9	8437.3	8393.8	8392.9	8427.7	56.6462

Table 5.3: Two Byte Block Decryption Results

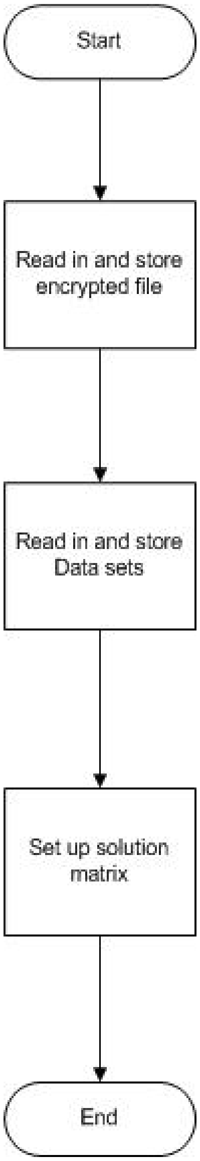

Figure 5.10: BCBB Algorithm Setup

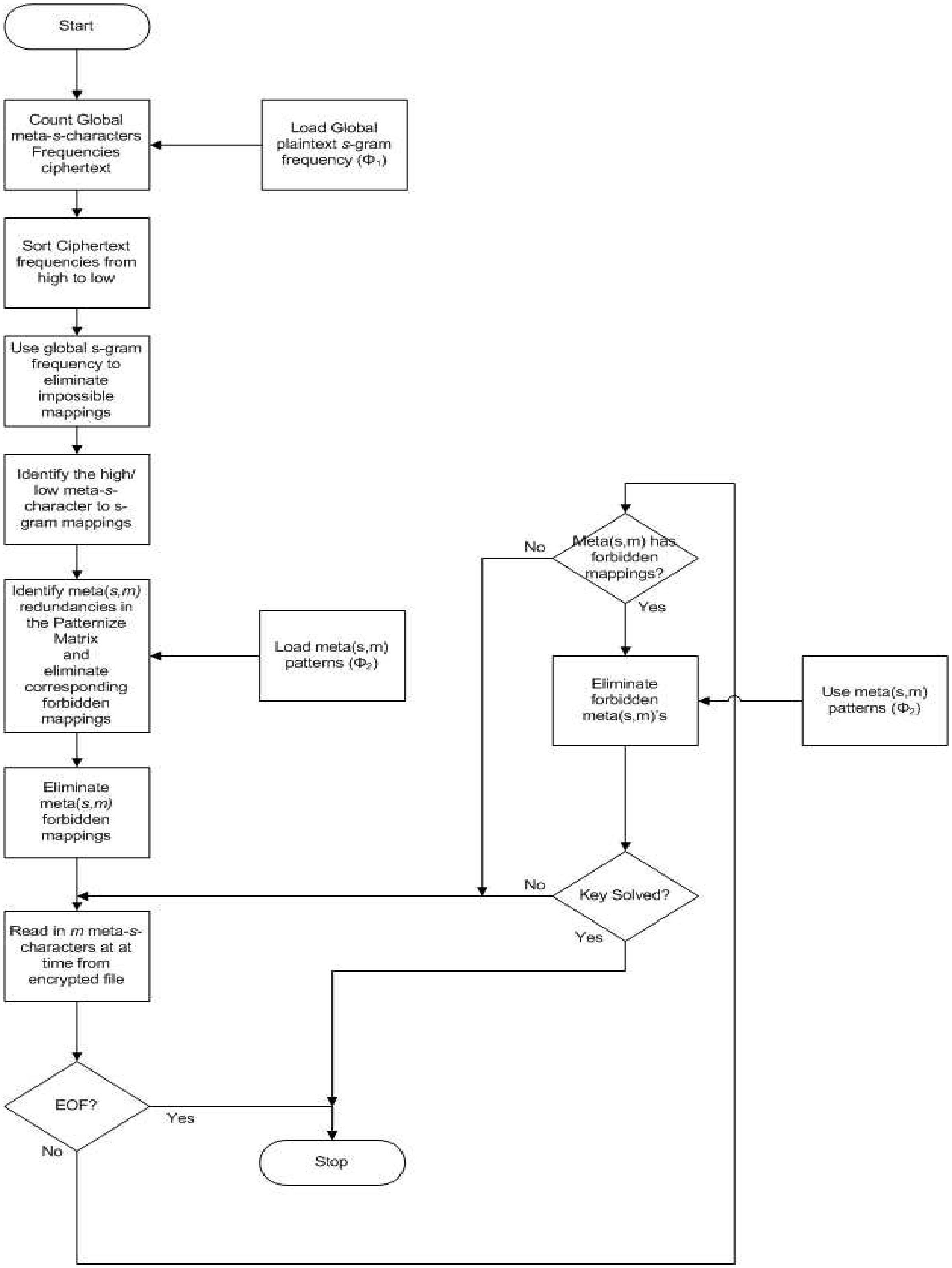
Start
Count Global meta-s-characters Frequencies ciphertext
Load Global plaintext s-gram frequency (Φ_1)
Sort Ciphertext frequencies from high to low
Use global s-gram frequency to eliminate impossible mappings
Identify the high/ low meta-s-character to s-gram mappings
Identify meta(s,m) redundancies in the Patternize Matrix and eliminate corresponding forbidden mappings
Load meta(s,m) patterns (Φ_2)
Eliminate meta(s,m) forbidden mappings
Read in m meta-s-characters at at time from encrypted file
EOF?
Yes
No
Stop
Meta(s,m) has forbidden mappings?
No
Yes
Eliminate forbidden meta(s,m)'s
Use meta(s,m) patterns (Φ_2)
Key Solved?
No
Yes

Figure 5.11: BCBB Algorithm Body

that contained the only characters seen above the threshold.

Next, the message was checked for redundant meta-s-characters in each meta(s,m) gram for all m selected for evaluation. Redundancy in a meta(s,m) gram yields low entropy information. Therefore, processing the redundancies further eliminated mappings.

After frequency and redundancy checks, the main body of message analysis begins. A meta-s-character was read from the message and the meta(s,m) gram set is applied to the portion of the message that was being analyzed. This process is iterated on by introducing a new meta-s-character from the message and the same analysis applies until the message is decrypted or there were no more meta-s-characters left for evaluation.

5.5.4 Improvements to the BCBB

The BCBB algorithm processes the message one meta-s-character at a time. This algorithmic approach is not efficient because it requires maintaining a solution matrix that grows exponentially in memory. In addition the size of the solution matrix increases in size as the meta-s-character increases in size. Therefore, the solution is memory bound.

Another version of the BCBB algorithm, called the BCBB2 algorithm was designed and is currently being implemented (see Figure 5.12). BCBB2 uses the same property sets as BCBB, but it operates on keys rather than creating a solution matrix. Key mappings are restricted to those composed of plaintext meta-s-characters found in the message. The results of the global frequency check are used to generate keys based on the probability of mappings. The message is decrypted using the selected key. Each resulting decrypted meta(s,m) is checked for membership in the forbidden meta(s,m) set. If a meta(s,m) is forbidden, the key must be incorrect. This forbidden meta(s,m) is saved as an invalid key mapping, called a “rule.” Rules encode partial key mappings that are invalid. Any key containing rules cannot be valid and can be removed from the solution set. All rules for a given key are found and recorded. Then the next key is checked. A key containing any rule in the rule set is discarded. Keys are checked until a key generating no new rules, the solution, is found.

BCBB2 does not create a solution matrix. Reducing the key space to the characters seen should reduce memory requirements and speed processing. Rules will eliminate keys without decryption by checking mappings, decreasing the amount of time spent on each key. Processing the meta(s,m) sets should also require less computational effort.

5.5.5 BCBB Decryption Results

Every text was correctly decrypted regardless of the cipher type employed. The time required for decryption was nearly identical for each cipher type (see Table 5.3). It should be noted that standard deviations was small, amounting to less than 0.03% of the mean in all cases. Variance in decryption times was most likely due 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.

Authors have distinct styles of writing, including the use of same sentence structure and lexicon in all of their works. Reusing the same patterns in structure and words results

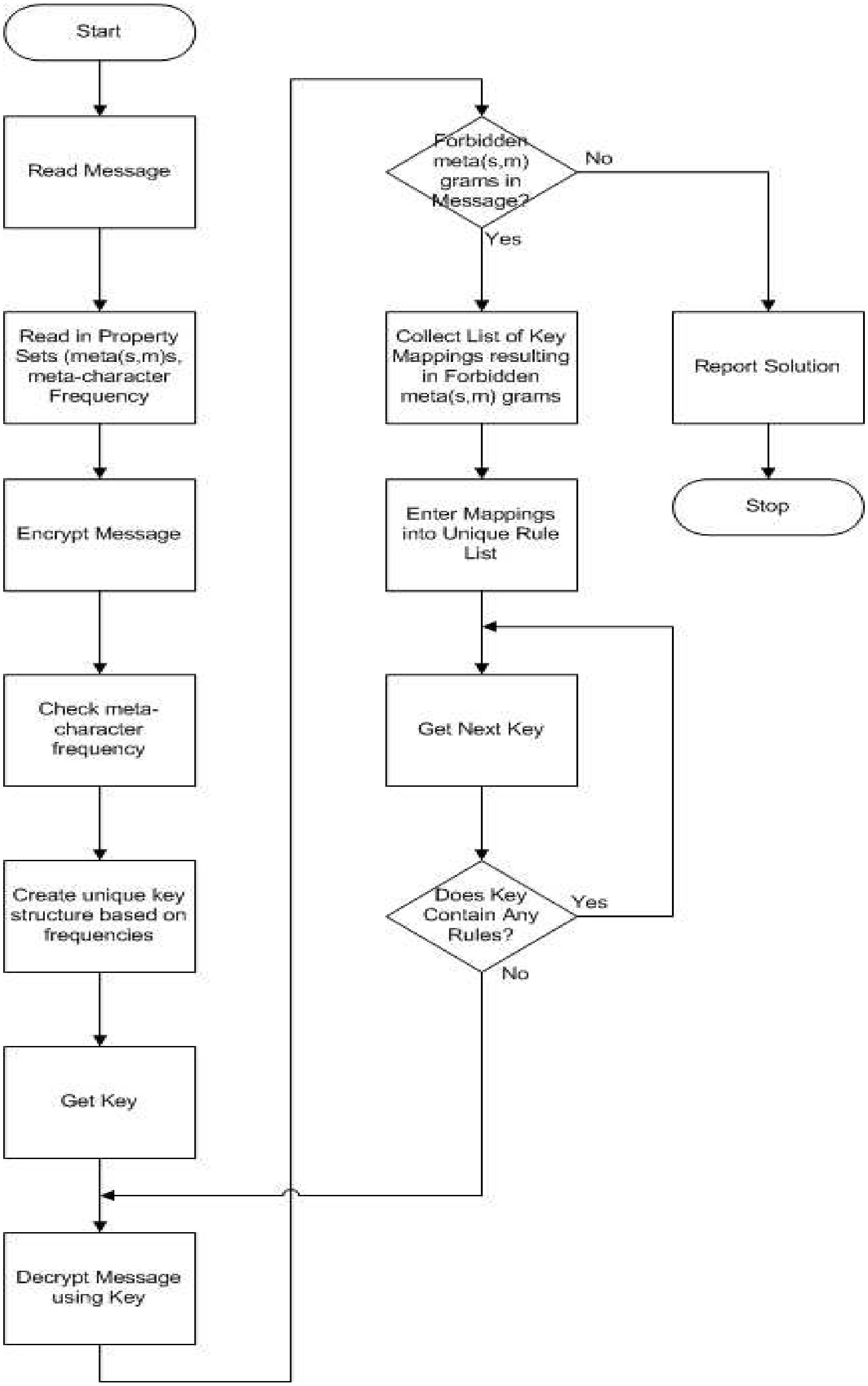
Start
Read Message
Read in Property Sets (meta(s,m)s, meta-character Frequency
Encrypt Message
Check meta-character frequency
Create unique key structure based on frequencies
Get Key
Decrypt Message using Key
Forbidden meta(s,m) grams in Message?
No
Yes
Collect List of Key Mappings resulting in Forbidden meta(s,m) grams
Enter Mappings into Unique Rule List
Get Next Key
Does Key Contain Any Rules?
Yes
No
Report Solution
Stop

Figure 5.12: BCBB2 Algorithm

in a set of m-grams trained with those patterns. As a consequence, authors that share similar patterns of styles should decrypt in similar times and number of ciphertext characters. For example, *Alice in Wonderland* and *Through the Looking Glass*, both by Lewis Carroll, showed similar decryption results.

Message (file) size also factored into the efficiency of breaking the file in a particular cipher (see Table 5.4). The smallest text file sizes in the test set was *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, while all other test files were larger than 245 kB. Shorter messages contain less data and a lower probability of low entropy events, such as redundancy in m-grams. Decryption results, in light of the corpus size, supports Shannon's contention that having more data in a message increases the probability of correct message decryption.

Of all the files tested, *Alice in Wonderland* had the greatest diversity of names. It 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 imaginary words and names contained in the text. Low frequency 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 the low frequency m-grams that result from those words. Patterns in those words, and consequently the m-grams, are not as likely to be represented in the m-gram sets.

Corpus data was derived from the same works of English used for decrypting meta-1-character files (see Chapter 3). During the time periods covered by the corpus, English usage evolved, changed, and has been re-characterized. Word and usage patterns regularly change with popularity. Changes in the 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 accuracy and efficiency. Sets of data derived from the same time period as the message are more likely to consist of the same patterns of word usage and frequency as the message. Customized time period language property sets require further research and are beyond the scope of this dissertation.

The original language of a text is also important. Even though foreign texts are translated into English, there are names, locations, and some words that retain the patterns of the original language [16]. Foreign locations and names are transliterated, producing low frequency m-grams that are mistaken as imaginary words. More low frequency m-grams increases 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. *On War*, originally written in German, decrypted in a similar amount of time as *The Jew of Malta*, as expected. 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. Longer processing time for each of these works resulted from the need to evaluate a larger number of characters due to the presence of foreign words in the texts.

As the size of the meta-s-character increases, the number of meta(s,m) grams in a language also increases. Successful decryption using the forbidden meta(s,m) sets necessitates having enough of the language represented in the sets to find valid language patterns for most messages. Variations in language style and lexicon affect the set size and membership. On the average, smaller allowed meta(s,m) gram sets are less likely to contain all of the meta(s,m) grams found in a message. The necessary size of the sets, compared to the meta-language, has not previously been studied and is unknown.

Definition 5.5: Meta(s,m) gram coverage

The coverage of a meta(s,m) gram set is the percentage of meta(s,m) grams contained in a set used to represent the patterns in a meta-language. For a particular meta(s,m) gram size (m) and alphabet (A), the coverage of the set is given by:

$$\text{Coverage} = \frac{|\text{meta(s,m) grams set}|}{|A|^m} \times 100 \tag{5.4}$$

□

Higher frequency patterns are more likely to be represented in the data sets and the total percentage of patterns represented indicates how likely the pattern is to be found. Therefore, the chance of eliminating a correct mapping is related to the frequency of the pattern appearing in the language.

Because there are many more meta(s,m)s as the meta-s-character size increases, using the same corpus for allowable meta(2,m)grams presents a 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. As the size of the meta-s-character increases the size of the corpus needed to represent the language also needs to 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 the 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, but 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. 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.

5.6 Advancements to the State-of-the-Art

Work in decrypting multi byte ciphers resulted in the following advances to the state-of-the-art:

1. The concept of meta-s-characters in ciphers - Defining meta-s-characters allows a cryptanalyst to keep diffused information in the same character. If all the information is kept together, then diffusion is defeated and decryption becomes easier. Diffusion across byte boundaries can be defeated using the concept of meta-s-characters.

2. Reduction of Fiestal round ciphers to an S cipher - Using the concept of meta-s-characters, Feistel rounds can be reduced to an S cipher. Solving each encryption in a Feistel round cipher, such as DES, might benefit from such an attack based on block substitution.

3. A single approach can be used to solve most ciphers (based on the S cipher) - The algorithm used in this dissertation successfully decrypted S, P, PS, PSP, and SPSP ciphers using the same data sets. The same approach was tested on single byte and multi byte ciphers, proving that the approach is valid for all block cipher sizes and types.

5.7 Chapter Summary

In this chapter the decryption of block ciphers using STE has been discussed. Diffusion across byte boundaries (permutation) is one mixed cipher technique Shannon suggested to increase cipher security. Making the assumption that the message is composed of meta-s-characters the same size as the block defeats the effects of diffusion. Under the meta-s-character assumption, all block ciphers reduce to an S block cipher and may be decrypted using the same decryption method. This method was successfully tested on multi byte ciphers which include all of the commonly used block cipher types used at this time. Results showed that the performance for each type of cipher was the same, demonstrating that the patterns in the ciphers were not disguised.

Tests conducted on a test corpus of files drawn from English literature and translations of foreign literature into English from the end of the 19th century to the 20th century showed the following:

1. A general algorithm exists for decrypting block and product ciphers consisting of blocks of 1 and 2 alphabetic characters in the language. Ciphers tested included S, P, PS, PSP, and SPSP ciphers.

2. Decryption of S, P, PS, PSP, and SPSP block ciphers with blocks consisting of 1 and 2 characters did not show significant decryption time differences for decryption. Decryption required the same number of ciphertext characters for each type of cipher. This indicates that language patterns were not disguised by cipher mixing.

3. File size had some effect on the amount of data required for encryption since a larger message typically has more low entropy events that can be evaluated.

4. Patterns were not sufficiently disguised to prevent cryptanalysis by mixing P and S constructs.

File	Title	File Size (kB)	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

Table 5.4: Two Byte Input meta-2-characters

Chapter 6

Summary, Conclusions, and Future Work

6.1 Summary

Since the first published description of Set Theoretic Estimation in 1969, it has been applied to various problems. Up until this dissertation all STE problem applications required their formulation in Hilbert space, a vector space with a metric function to define distance measures. This dissertation brings information theory under the framework of STE via the Asymptotic Equipartition Property. It is also demonstrated that the STE is a richer method with applications in cryptography using a simpler topological space. The reformulation of cryptanalysis within the framework of STE has allowed the decryption of ciphers with diffusion across byte boundaries. This is the first published example of this type of decryption. Furthermore, it is proved that all block ciphers are a form of the substitution cipher. Consequently, the general attack algorithm presented in this dissertation can be used against all block ciphers.

The tests run using the Block Cross Byte Boundary algorithm used m-gram, multiple letter runs, word, and sentence structure property sets. Tests conducted on the corpus of files described in Chapter 3 showed the following results:

1. Shift cipher

   All test files were successfully decrypted. An average of 5.53 ciphertext characters were evaluated before decryption was achieved. Decryption of each test occurred in less than 1 ms.

2. Substitution cipher

   The substitution decryption algorithm decrypted 85.53% of the files correctly and required an average of 256 characters to decrypt. Most failed decryption efforts were attributable to forbidden m-grams contained in names, foreign words, and imaginary words contained in the text. Decryption took approximately 50.7 seconds. Testing results were faster than average automated decryption efforts which require approximately 5000 characters [132].

3. Permutation cipher

   The permutation decryption algorithm decrypted 99.85% of the files correctly. Decryption of P ciphers took an average of 19.34 characters and 0.563 seconds.

4. Block ciphers

   The block ciphers tested in this dissertation included: S, P, PS, PSP, and SPSP ciphers. All of these ciphers involve permutation across byte boundaries. Under the assumption that the use of a meta$s$character with an s the same size as the block, all the block ciphers tested were solved in the same number of characters and in a nearly identical period of time. On the average, decryption of block ciphers took 3,462.2 characters and 8,427.7 seconds to solve. The longest decryption took 44,690 characters, with the shortest 546 characters.

6.2 Conclusions

The research reported in this dissertation investigated the use of STE in the field of cryptography. The study and subsequent implementation of STE in cryptography can be summarized as follows:

1. STE can be applied in branches of information theory.

   Because STE follows the AEP, as shown by the relationship of the typical set to the solution set, information theory is a branch of STE.

2. STE can take place in non-Hilbert spaces

   Most STE applications take place in Hilbert spaces. This dissertation has demonstrated that STE can be applied in a topological space, avoiding the need for a distance metric and simplifying complications arising from OBEs.

3. STE is useful in cryptography and cryptanalysis

   STE has been used in a variety of applications prior to this research. However, until this dissertation, STE had not been used in cryptography. Successful decryption of S, P, PS, PSP, and SPSP ciphers of different block sizes occurred using fewer characters than typically needed for automated decryptions. Results from this research indicate that STE concentrates information and uses it more efficiently than other decryption methods.

4. All block ciphers can be reduced to the S cipher

   As long as information remains in the same character (or meta*s*character), all ciphers reduce to the S cipher.

5. A single decryption algorithm can be used to decrypt all block ciphers

Results from research tests demonstrate that S, P, PS, PSP, and SPSP ciphers can be decrypted using the same algorithm. Breaking encrypted block cipher messages into chunks the same size as the block and treating those blocks as a single meta*s*character in a metalanguage keeps information in the same character. The number of characters and time needed for decryption in each cipher case were the same, indicating that patterns are not obscured by product cipher diffusion.

6. Product ciphers do not increase the security of encryption because of cipher mixing

   Product ciphers increase security as the size of the block increases. Larger blocks result in a larger key space which increases security. Some encryption algorithms attempt to increase security by using diffusion. These efforts fail because all block ciphers ultimately reduce to the S cipher.

7. Systematic property set development for cryptanalysis.

   This dissertation presented a systematic methodology for the development of property sets for cryptanalysis. What is particularly contributive in this methodology is that the property sets are defined and constructed to complement one another. The result is an overall good span of coverage of the dimensions of the information space of cryptanalysis.

Conclusions 1 - 7 constitute original contributions to knowledge.

6.3 Future Work

During the course of this research, a number of topics have presented themselves that are interesting, but beyond the scope of this dissertation. The most pertinent topics are listed below:

1. Treating author identification as a decryption problem

   Stylometry indicates that an author's writing habits lead to identifiable language patterns. Such patterns form the basis for author specific property sets. As such, an author may be identified using customized property sets drawn from a corpus of their work to decrypt a target file whose author is in question;

2. Developing methods to classify words that may belong to sets of names and foreign words.

   The most common reason for failure in the decryption method was attributed to the inclusion of names, foreign words, and imaginary words appearing in decryption files. The addition of m-gram and word sets from dictionaries of names and/or foreign languages would be highly desirable. The inclusion of these sets would constitute a form of relaxation and potentially increase the number of correct decryptions using STE algorithms and language property sets. Furthermore, such a method would also improve the ability to attack ciphers of plaintexts that include special code words;

3. Key elimination using failed mappings.

   Implement the BCBB2 algorithm and compare its performance to the BCBB algorithm. An effective key generation algorithm is needed. Reduction of memory use and increasing the speed of the algorithm are the primary goals for this effort; and

4. Develop and algorithm to exploit equivalent keys.

   Equivalent keys reduce the keyspace of a cipher. Creating the set of equivalent keys and using that set of keys as the keyspace may speed decryption efforts. The concept of isomorphic neighborhoods can be used to design algorithms that can parallelize decryption for large keyspace ciphers.

This dissertation has demonstrated that appropriate selection of property sets used in STE can reduce decryption time and minimize the number of characters needed for successful decryption. While this research has answered basic questions about the use of STE in cryptography, additional work could prove fruitful. In light of these developments, STE should be added as a tool in cryptanalysis.

Bibliography

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

[2] Claude Shannon. A symbolic analysis of relay and switching circuits. *Transactions of the American Institute of Electrical Engineers*, 57:713 – 723, 1938.

[3] George Boole. *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities (reprinted with corrections).* MacMillan, London, 1854.

[4] N. J. A. Sloane and Aaron D. Wyner, editors. *Collected Papers of Claude Shannon.* IEEE Press, New York, 1993.

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

[6] Bertrand Russell. *Principles of Mathematics.* At the University Press, Cambridge, 1903.

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

[8] Georg Cantor. *Gesammelte Abhandlungen.* Springer-Verlag, Berlin, 1932.

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

[10] Jon von Neumann. *Mathematical Foundations of Quantum Mechanics.* Princeton University Press, Princeton, 1955.

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

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

[13] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379 – 423, 623 – 656, 1948.

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

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

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

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

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

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

[20] M. Matsui. Linear cryptanalysis method for des cipher. In *Advances in Cryptology – EUROCRYPT '93*, pages 386–397, 1993.

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

[22] Oded Goldreich. *Foundations of Cryptography I.* Cambridge University Press, Cambridge, 2001.

[23] Hans Witsenhausen. Sets of possible states of linear systems given perturbed observation. *IEEE Transactions on Automatic Control*, AC-13(1):556 – 558, 1968.

[24] Fred Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, AC-13(1):22 – 28, 1968.

[25] Dimetri Bertsekas and Ian Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Transactions on Automatic Control*, AC-16(2):117 – 128, 1971.

[26] Belforte, Gustavo, Bona, Balisio, Cerone, and Vito. Parameter estimation algorithms for a set-membership description of uncertainty. *Automatica*, 26(5):887 – 898, 1990.

[27] Eli Fogal. System identification via membership set constraints with energy constrained noise. *IEEE Transactions on Automated Control*, AC-24(5):752 – 797, 1979.

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

[29] John Norton. Identification and application of bounded-parameter models. *Automatica*, 23(4):497 – 507, 1987.

[30] 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.

[31] John Norton and Sandor Veres. Outlines in bound-based state estimation and identification. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 790 – 793, 1993.

[32] M. Milanese and Antonio Vicino. Estimation theory for dynamic systems with unknown but bounded uncertainty: An overview. In *IFAC*, pages 231 – 238, 1991.

[33] M. Milanese and Antonio Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: An overview. *Automatica*, 27(6):997 – 1009, 1991.

[34] Konstantinos Tsakalis, Michael Deisher, and Andreas Spanias. System identification based on bounded error constraints. *IEEE Transactions on Signal Processing*, 43(12):3071 – 3075, 1995.

[35] 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.

[36] *Least-Square Identification with Error Bounds for Real-Time Signal Processing and Control*, volume 81, 1993.

[37] *Multiweight Optimization in OBE Algorithms for Improved Tracking and Adaptive Identification*, 1998.

[38] Lotfi Zadeh. What is optimal? *IRE Transactions on Information Theory*, 4(1):3, 1958.

[39] Jr. John Deller. Unifying the landmark developments in optimal bounding ellipsoid identification. *International Journal of Adaptive Control and Signal Processing*, 8:43 – 60, 1994.

[40] D. J. Leal, G. Georgantzis, and P. D. Roberts. Parameter estimation in uncertain models of nonlinear dynamic systems. *Electronic Letters*, 14(22):718 – 720, 1978.

[41] Ashok Rao and Yih fang Huang. Analysis of finite precision effects on a recursive set membership parameter estimation algorithm. *IEEE Transactions on Signal Processing*, 40(12):3081 – 3085, 1992.

[42] John Deller Jr. Set membership identification in digital signal processing. *IEEE ASSP Magazine*, 6(4):4 – 22, 1989.

[43] P. L. Combettes M. Beidir and B. Picinbono. A general framework for the incorporation of uncertainty in set theoretic estimation. *IEEE International*, 3:349 – 353, 1992.

[44] T.T. Tay and M.H. Tan. A robust adaptive performance enhancement controller using set membership identification. *IEEE Transactions on Automatic Control*, 37(10):1542 – 1548, 1992.

[45] Stephen McCarthy and Richard Wells. Model order reduction for optimal bounding ellipsoid channel models. *IEEE Transactions on Magnetics*, 33(4):2552–2568, 1997.

[46] Iriving Kaplansky. *Set Theory and Metric Spaces*. Allyn and Bacon, Boston, 1972.

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

[48] Thomas Cover and Joy Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc, New York, 2nd edition, 2005.

[49] A. Abo-Taleh and Moustafa Fahmey. Design of fir to-dimensional digital filters by succesive projections. *IEEE Transactions on Circuits and Systems*, 31(9):801 – 805, 1984.

[50] James Cadzow and Tso-cho Chen. Algebraic approach to two-dimensional recursive digital filter synthesis. *IEEE Transactions on Acoustics, Speech*, 37(5):655 – 664, 1989.

[51] Ramin Nobakht, M. Reha Civanlar, and Susan Ardalan. Comments on 'design of a class of time-constrained fir digital filters by successive projections. *IEEE Transactions on Circuits and Systems*, 37(12):1581, 1990.

[52] S. C. Pei and I. I. Yang. Design of a class of time constrained fir filters by successive projections. *IEEE Transactions on Circuits and Systems*, 36(1):164 – 167, 1989.

[53] Roberto Tempo. Robust estimation and filtering in the presence of bounded noise. *IEEE Transactions on Automated Controls*, 33(9):864 – 867, 1988.

[54] Ahmet Cetin and Rashid Ansari. Iterative procedure for designing two-dimensional fir filters. *Electronics Letters*, 23(3):131 – 133, 1987.

[55] Ron Aharoni, Abraham Berman, and Yair Censor. An interior points algorithm for the convex feasibility problem. *Advances in Applied Math*, 4(4):479 – 489, 1983.

[56] Ron Aharoni and Yair Censor. Block-iterative methods for parallel computation of solutions to convex feasibility problems. *Linear Algebra and Its' Applications*, 120:165 – 175, 1989.

[57] James Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 2nd edition, 1985.

[58] Ovildio Bucci, G. Franceschetti, G. Mazzerella, and G. Panariello. Intersection approach to array pattern synthesis. *IEE Proceedings - H*, 137(6):349 – 357, 1990.

[59] Ovildio Bucci, G. Franceschetti, G. Mazzerella, and G. Panariello. Reconfigurable arraysby phase-only control. *IEEE Transactions on Antennas*, 39(7):919 – 925, 1991.

[60] Hamdi Elmikati and A. A. Elsohli. Extension of projection method to nonuniformly linear antenna arrays. *IEEE Transactions on Antennas and Propagation*, 32(5):507 – 512, 1984.

[61] Geoff Poulton and Stuart Hay. Efficient design of shaped reflectors using successive projections. *Electronic Letters*, 27(23):2156 – 2158, 1991.

[62] *Optimal Pulse Shape Design Using Projections Onto Convex Sets*, 1988.

[63] M. Reha Civanlar and H. Joel Trussell. Constructing membership functions using statistical data. *Fuzzy Sets and Systems*, 18(1):1 – 13, 1986.

[64] M. Reha Civanlar and H. Joel Trussell. Digital signal restoration using fuzzy sets. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):919 – 936, 1986.

[65] Patrick Combettes and H. Puh. Parallel projection methods for set theoretic signal reconstruction and restoration. *Proceedings of IEEE the International Conference on Acoustics, Speech, and Signal Processing*, 5:297 – 300, 1993.

[66] Susan Curtis, Alan Oppenheim, and Jae Lim. Signal reconstruction from fourier transform sign information. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3):643 – 657, 1985.

[67] M. H. Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(2):140 – 154, 1982.

[68] S. Hein and Avideh Zakhor. Reconstruction of oversampled band-limited signals from $\sigma\delta$ encoded binary sequences. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:161 – 164, 1996.

[69] Yih-fang Huang. A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(5):1331 – 1334, 1986.

[70] A. Papoulis. A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Transactions on Circuits and Systems*, 22(4):735 – 742, 1975.

[71] N. T. Thao and M. Vetterli. Optimal mse signal reconstruction in oversampled a/d conversion using convexity. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:161 – 164, 1996.

[72] Trussell H. Joel and M. Reha Civanlar. The feasible solution in signal restoration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(2):201 – 212, 1984.

[73] Avideh Zakhor and Alan Oppenheim. Reconstruction of two-dimensional signals from level crossings. *Proceedings of the IEEE*, 78(1):31 – 55, 1990.

[74] Aggelos Katsaggelos, Jan Biemond, Ronald Schafer, and Russell Mersercau. A regularized iterative image restoration algorithm. *IEEE Transactions on Signal Processing*, 39(4):914 – 929, 1991.

[75] Stephane Mallat. Zero-crossings of a wavelet transform. *IEEE Transactions on Information Theory*, 37(4):1019 – 1033, 1991.

[76] Yair Censor. Parallel application of block-iterative methods in medical imaging and radiation therapy. *Mathematical Programming*, 42(2):307 – 325, 1988.

[77] Yair Censor, D. E. Gustafson, A. Lent, and H. Tuy. A new approach to the emission computerized tomography problem: Simultaneous calculation of attenuation and activity coefficients. *IEEE Transactions on Nuclear Science*, NS-26(2):2775 – 2779, 1979.

[78] R. T. Chin, C. L. Yeh, and W. S. Olson. Restoration of multichannel microwave radiometric images. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 7(4):475 – 484, 1985.

[79] G. Crombez. mage recovery by convex combinations of projections. *Journal of Mathematical Analysis and Applications*, 155(2):413 – 419, 1991.

[80] M. Hedley, H. Yah, and D. Rosenfeld. Motion artifact correction in mri using generalized projections. *IEEE Transactions on Medical Imaging*, 10(1):40 – 46, 1991.

[81] G. T. Herman. A relaxation method for reconstructing objects from noisy x-rays. *Mathematical Programming*, 8(1):1 – 19, 1975.

[82] G. T. Herman. Mathematical optimization versus practical performance: A case study based on the maximum entropy criterion in image reconstruction. *Mathematical Programming Study*, 20:96 – 122, 1982.

[83] H. Kudo and T. Saito. Sinogram recovery with the method of convex projections for limited-data reconstruction in computed tomography. *Journal of the Optical Society of America*, 8(7):1148 – 1160, 1991.

[84] Jorge Llacer and Eugene Veklerov. Feasible images and practical stopping rules for iterative algorithms in emission tomography. *IEEE Transactions on Medical Imaging*, 8(2):186 – 193, 1989.

[85] Peyma Oskoui-Fard and Henry Stark. Tomographic image reconstruction using the theory of convex projections. *IEEE Transactions on Medical Imaging*, 7(1):45 – 58, 1988.

[86] Peyma Oskoui-Fard and Henry Stark. A comprehensive study of three reconstruction methods for a limited-view computer tomography problem. *IEEE Transactions on Medical Imaging*, 8(1):43 – 49, 1989.

[87] Rangaraj Rangayyan, Atam Dhawan, and Richard Gordon. Algorithms for limited-view computed tomography: An annoted bibliography and a challenge. *Applied Optics*, 24(23):4000 – 4012, 1985.

[88] Ahmet Cetin. An iterative algorithm for signal reconstruction from bispectrum. *IEEE Transactions on Signal Processing*, 29(2):2621 – 2628, 1991.

[89] Ahmet Cetin. Convolution-based framework for signal recovery and applications. *Journal of the Optical Society of America*, 5(8):1193 – 1200, 1988.

[90] Patrick Combettes. Signal recovery by best feasible approximation. *IEEE Transactions on Image Processing*, 2(2):269 – 271, 1993.

[91] Patrick Combettes and H. Joel Trussell. Set theoretic autoregressive spectral estimation. *Proceedings of the Fifth ASSP Workshop on Spectrum Estimation*, 37(3):261 – 264, 1990.

[92] Yair Censor and T. Elfving. New methods for linear inequalities. *Linear Algebra and Its' Applications*, 42:149 – 211, 1982.

[93] Jr. John Deller and T. C. Luk. Linear prediction analysis of speech based on set-membership theory. *Computer Speech and Language*, 3:301 – 327, 1989.

[94] Steven Ebstein. Stellar speckle interferometry energy spectrum recovery by convex projections. *Applied Optics*, 26(8):1530 – 1536, 1987.

[95] R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35(2):237 – 246, 1972.

[96] Neal Gallagher and B. Liu. Method for computing kinoforms that reduces image reconstruction error. *Applied Optics*, 12(10):2328 – 2335, 1973.

[97] R. Burge. Two optimization approaches to cohoe design. *Optics Communications*, 73(3):188 – 194, 1989.

[98] W. Duane Montgomery. Optical applications of von neumann's alternating-projection theorem. *Optics Letters*, 7(1):1 – 2, 1982.

[99] Joseph Rosen and Joseph Shamir. Application of the projection onto constraint sets algorithm for optical pattern recognition. *Optics Letters*, 16(10):752 – 754, 1991.

[100] Bahaa Saleh and Karen Nashold. Image construction: Optimum amplitued and phase masks in photolithography. *Applied Optics*, 24(10):1432 – 1437, 1985.

[101] P. Gilbert. Iterative methods for the three-dimensional reconstruction of an objects from projections. *Journal of Theoretical Biology*, 36(1):105 – 117, 1972.

[102] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29(3):471 – 481, 1970.

[103] G. T. Herman, A. Lent, and P. H. Lutz. Relaxation methods for image reconstruction. *Communications of the ACM*, 21(2):152 – 158, 1978.

[104] S. S. Kuo and R. J. Mammone. Image restoration by convex projections using adaptive constraints and the l_1 norm. *IEEE Transactions on Signal Processing*, 2(1):97 – 104, 1961.

[105] Anne Landraud. Image restoration and enhancement of the characters, using convex projection methods. *CVGIP: Graphical Models and Image Processing*, 53(1):85 – 92, 1991.

[106] C. P. Mariadassour and B. Yegnanarayana. Image reconstruction from noisy digital holograms. *IEE Proceedings - F*, 137(5):351 – 356, 1990.

[107] Robert Kosut. Adaptive control via parameter set estimation. *International Journal of Adaptive Control Signal Processing*, 2(4):371 – 399, 1988.

[108] F. M. Schlaepfer and Fred Schweppe. Continuous-time state estimation under disturbance bounded by convex sets. *IEEE Transactions on Automated Control*, 17(2):276 – 280, 1972.

[109] Fred Schweppe and H. K. Knudsen. The theory of amorphous cloud trajectory prediction. *IEEE Transactions on Information Theory*, 14(3):95 – 101, 1968.

[110] M. Goldburg and II Robert Marks. Signal synthesis in the presence of an inconsistent set of constraints. *IEEE Transactions on Circuits and Systems*, 32(7):647 – 663, 1985.

[111] Kostic, Zoran, Sezan, Muhammed, and E. Titlebaum. Estimation of the parameters of a multipath channel using set-theoretic deconvolution. *IEEE Transactions on Communications*, 40(6):1006 – 1011, 1992.

[112] II Robert Marks. Class of continuous level associative memory neural nets. *Applied Optics*, 26(10):2005 – 2010, 1987.

[113] II Robert Marks, S. Oh, and Les Atlas. Alternating projection neural networks. *IEEE Transactions on Circuits and Systems*, 36(6):846 – 857, 1989.

[114] Muhammed Sezan, Henry Stark, and S. J. Yeh. Projection method formulations of hopfield-type associate memory neural networks. *Applied Optics*, 29(17):172 – 186, 1990.

[115] R. D. Palmer, J. R. Cruz, and D. S. Zernic. Enhanced autoregressive moving average spectral estimation applied to the measurement of doppler spectral width. *IEEE Transactions on Geoscience and Remote Sensing*, 22(9):358 – 368, 1991.

[116] Harold Sabbagh, Karen Nashold, and Thomas Roberts. An eddy-current model and algorithm for three-dimensional non-destructive evaluation of advanced composites. *IEEE Transactions on Magnetics*, 24(6):3201 – 3212, 1988.

[117] William Menke. Applications of the pocs inversion method to interpolating topography and other geophysical fields. *Geophysical Resource Letters*, 18(3):435 – 438, 1991.

[118] Richard Wells. Algorithms for threshold level selection and decoder logic design in decision aided symbol-by-symbol data receivers for magnetic recording applications. *IEEE Transactions on Magnetics*, 31(5):2527 – 2535, 1995.

[119] Richard Barakat and Garry Newsam. Algorithms for reconstruction of partially known band-limited fourier transform pairs from noisy data ii: The non-linear problem of phase retrieval. *Journal of Integral Equations*, 9(1 (Supplement)):77 125, 1985.

[120] Messaoud Benidir and Bernard Picinbono. Nonconvexity of the stability domain of digital filters. *IEEE Trans. Acoustics, Speech, Signal Process*, 38(8):1459 1460, 1990.

[121] Yair Censor. Parallel application of block-iterative methods in medical imaging and radiation therapy. *Math. Programming*, 42(2):307 325, 1988.

[122] G. Crombez. Image recovery by convex combinations of projections. *Journal of Math. Analysis Applications*, 155(2):413 419, 1991.

[123] Geoff Poulton. Antenna power pattern synthesis using method of successive projections. *Electronic Letters*, 22(20):1042 – 1043, 1986.

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

[125] 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.

[126] M. Milanese and G. Belaforte. Estimation theory and uncertainty interval evaluation in presence of unknown but bounded errors: Linear families of models and estimators. *IEEE Transactions on Automated Control*, AC-27:408 – 414, 1982.

[127] Thierry Clement and Silviane Gentil. Recursive membership set estimation for output-error models. *Mathematics and Computers in Simulation*, 32:505 – 513, 1990.

[128] Hui-Hsiung Kuo. *White Noise Distribution Theory.* CRC Press, Boca Raton, 1996.

[129] Naresh Sinha and Y. H. Kwong. Recursive estimation of the parameters of linear multivariable systems. *Automatica*, 15:471 –475, 1979.

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

[131] Carnegie Mellon University Computer Science Department. Cmu artificial intelligence repository. Internet, February 1995.

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

[133] M. Lucks. A constraint satisfaction algorithm for the automated decryption of simple substitution ciphers. In *CRYPTO 1988*. CRYPTO, 1988.

[134] 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.

[135] *An Information-Based Approach to Cryptography*, 2007.

[136] Kenny Smith. *The Transmission of Language: models of biological and cultural evolution*. PhD thesis, Theoretical and Applied Linguistics, School of Philosophy, Psychology and Language Sciences, The University of Edinburgh, 2003.

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

[138] Alan Beale. The 12 word dicts list, release 5. Technical report, Internet, April 2012.

[139] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[140] 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.

[141] J. Hagenauer, E. Offer, and L. Pupke. Iterative decoding of binary and block and convolutional codes. *IEEE Transactions on Information Theory*, 47(2):543 – 553, 1996.

[142] Imre Lakotos. *Proofs and Refutations.* Cambridge University Press, Campbridge, 1976.

[143] Douglas R. Stinson. *Cryptography, Theory and Practice.* Chapman & Hall/CRC, Boca Raton, 3rd edition, 2006.

Appendix A

Complexity Analysis

Consider the BCBB algorithm presented in Chapter 5 of this dissertation and the typical method of breaking a Fiestel round block cipher by solving each cipher separately in the encryption (hereafter called the Fiestel break method). In this appendix the complexity of both algorithms is compared and it is shown that the BCBB algorithm requires less operations than the Fiestel break algorithm for a product block cipher.

The BCBB algorithm evaluates the frequency of meta-s-characters in the message and then evaluates the meta(s,m) grams in the file. BCBB solves a S cipher and is $O(n^2)$.

Fiestel round ciphers are an example of product ciphers with a regular internal structure that can be generalized for all block ciphers. A Fiestel round cipher can be broken into two parts: the rounds (d_r) and ciphers applied before and after the rounds in the encryption process (d_i). Let D be the total number of ciphers in a product cipher such as a Fiestel round cipher, R be the number of rounds in the block cipher, $|C|$ be the number of ciphers in a round, and $|P|$ be the number of ciphers applied before and after the rounds. Ciphers used in the encryption can be S, P, or XOR in any combination. Then the total number of ciphers in the product cipher encryption is

$$D = d_r + d_i = R|C| + |P| \tag{A.1}$$

It is shown in Theorem 5.2 and Corollary 5.3 that all block ciphers are block S ciphers. Solving for each cipher in the round requires solving only a block S cipher.

Without loss of generality, let the solution of a S block cipher be the same for all S block ciphers and denoted by I, which represents the number of instructions required to decrypt a message for a block cipher of size s characters. The number of instructions required for a BCBB decryption is I, since only a single S cipher is solved. In contrast, a Fiestel break requires $D \times I$ instructions for decryption. A Fiestel break requires D times more instructions than the BCBB algorithm.

To illustrate the calculation of complexity difference, consider the DES cipher. The DES cipher is a Fiestel round cipher in which there are 4 ciphers per round ($|C| = 4$). Sixteen rounds (R=16) are required for the encryption, with permutations preceding and following the rounds ($|P| = 2$), for a total of $D = 66$ ciphers. The number of ciphers required to solve the DES cipher using the Fiestel break is $D = 66$ times the number of ciphers the BCBB would break. In addition, the Fiestal break is typically solved by breaking the sequence of applied cipher rounds in reverse order, which infers an additional decryption complexity of $O(|C|!)$ if Kerckhoffs assumption of full knowledge of the cipher is not followed. Notice that Kerckhoffs assumption is never used in the BCBB algorithm. Product block ciphers that have more ciphers in the rounds will require correspondingly more cipher decryptions and instructions to decrypt. For any block product cipher using the Fiestel break the BCBB algorithm requires less instructions to achieve decryption.