

Revising the 1:1 Principle in Cryptography

Benjamin Williams*, Albert Carlson[†], Robert E. Hiromoto[‡],
Sai Ranganath Mikkilineni[§], Mandeep Singh[¶], and Christopher Briscoe^{||}

*[¶]Independent Researcher

[†]Computer Science Department, National University, San Diego, CA, USA.

[‡]Computer Science Department, University of Idaho, Idaho Falls, ID, USA.

[§]College of Business, Delaware State University, Dover, DE, USA.

^{||}Department Chair of Mathematics and Physics, Emet Classical Academy, New York, NY, USA.

Email: *rybecarethdar@gmail.com, [†]acarlson2@nu.edu, [‡]hiromoto@uidaho.edu,

[§]SMikkilineni@desu.edu, [¶]mandeep.singh@gmail.com, ^{||}Cbriscoe@emetclassicalacademy.org

Abstract—For hundreds of years, cipher designers have operated on the principle that plain-text tokens must map to a single ciphertext character, and vice versa. It is widely understood that this bijective mapping ensures that an encrypted token can be decrypted to recover the original plaintext token. However, other mappings that violate this 1:1 principle of mapping have been shown to be possible and practical. Among those mappings are polyphonic and polyalphabetic ciphers (mappings). The 1:1 principle remains relevant but must be expanded to encompass all possible mappings from plaintext to ciphertext. With the constraint of a single (plaintext) token mapping to a single (ciphertext) token, cipher designers can explore other types of ciphers that have desirable security characteristics. This paper redefines the 1:1 principle in terms of sets and explores its implications. This paper does not intend to present implementations and, by extension, does not present experimental evidence, both of which certainly deserve attention in forthcoming future work.

Keywords—Cryptography, entropy, probability mass function, polyphonic cipher, polyalphabetic cipher, encryption, 1:1 principle, bijective function, set theory, keys, S cipher, mappings.

I. INTRODUCTION

Obfuscating messages is one of the key functions used to protect data in cybersecurity. The input message must be deterministically converted into a form that can be recovered by legitimate parties to the conversation, yet be difficult for an attacker to read. This constraint of determinism on mappings has been accepted as necessary for encryption, although this assumption, in this form, has recently been challenged. Slightly altering the assumption to remove the arbitrary constraint of one symbol in the input alphabet mapping uniquely to a single cipher output allows the cryptographer to modify the language statistics of a cipher. Since language statistics are among the best sources of information about the contents of a ciphertext, changing those statistics can greatly increase the entropy and, therefore, the security of the message.

In this paper, the 1:1 assumption is redefined to widen the allowable mappings for sets rather than single characters. By so doing, it will be shown that encryption can be used to shape the language statistics in a way that complicates attacks on the data contained in the message.

979-8-3315-4568-0/26/\$31.00 ©2026 IEEE

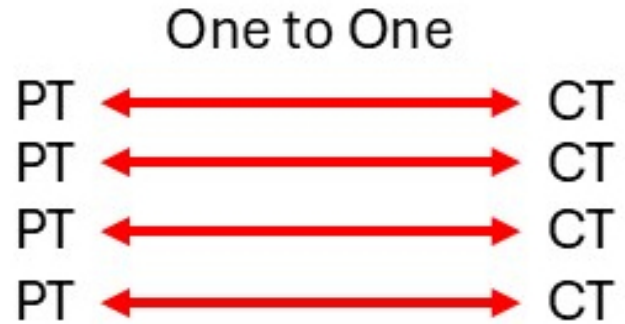


Fig. 1. One-to-One Mapping

A. Paper Composition

The paper consists of five major sections. Section I introduces the problem and explains its importance. The second section contains background information for readers who are not very familiar with the underpinnings, to ease their understanding of the subject matter. Section III presents the proposed redefinition of the principle that generalizes the use of sets in mappings. The implications of this redefinition are presented in Section IV. The bulk of the paper is in this section, which offers suggestions for using these mappings to design new ciphers. The final section, Section V, presents the paper's conclusions and suggests future work on the subject and related offshoots.

II. BACKGROUND

A. Language Statistics

Human languages are composed of a finite number of symbols, also known as "tokens." These tokens comprise an alphabet of characters, or a combination of characters called "n-grams" [1], that are combined to form words, sentences, and larger portions of a message. There is a limited set of combinations in the language that comprise the lexicon and syntax of the language. If those tokens are counted, they result in a characteristic probability mass function (PMF) for the language. For example, the PMF for English is shown

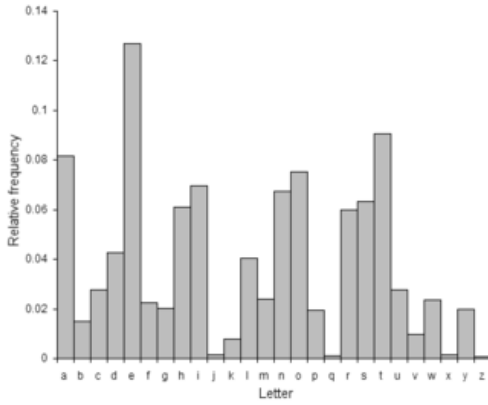


Fig. 2. Probability Mass Function (PMF) for English [6]

in Figure 2. These numbers are compiled in several standard sources [2], [3]. It is well established that ciphers, such as the substitution or S cipher, can show the patterns of the original language [4]. Carlson et al. have also shown that all ciphers without built-in randomization can be reduced to the S cipher via isomorphic cipher reduction [5] and broken using the same methods. Cryptographers have long used this information, combined with the knowledge that mappings follow the 1:1 principle.

B. The 1:1 Principle and the S Cipher

Cryptography is an obscuring function that makes a message unreadable to parties unauthorized to see the data in the encrypted message. This function relies on a “secret” known only to the parties to the communications [1], [7]. Cryptographers assume that the cryptographic function is known and will typically not use a function that has not been analyzed and peer-reviewed. The secret key is changed with each transmission to prevent attackers from analyzing messages and reconstructing the key.

In cryptography, ciphers are bijective functions [8]. That means the input alphabet maps only to the output alphabet, with each input uniquely mapping to one output, and vice versa (see Figure 1). For encryption, the input set is the original message, or plaintext (PT), and the output is the obscured message, or ciphertext (CT). This property is known as the 1:1 principle [9] and is a constraint that ensures a message can be obscured so that the inverse mapping decrypts to the same message.

Keys can be considered mappings that replace the original message (plaintext) with an unreadable message (i.e., ciphertext) unless the function is reversed. All ciphers that do not have internal randomization can be replaced by an equivalent Substitution (S) [4] cipher via a property known as “isomorphic cipher reduction” [5]. Even complex functions can be reduced using the property of “idempotence” [10]. Therefore, the secret key for a cipher can be represented as a mapping ($\mapsto$), making the study of keys and the rules

Case	PT Set Type	CT Set Type
Singleton	Single Element	Single Element
One to Many	Single Element	Set
Many to One	Set	Single Element
Many to Many	Set	Set
Mixed	Set	Set

TABLE I
POSSIBLE SET MAPPINGS

governing their mappings important. It also suggests that the S cipher should be a subject of study for design cryptographers.

The 1:1 principle produces a mapping that resembles a perfect hashing function. This is natural, since hashes are a type of encryption that is often used as a technique for mapping a “key” to a location where data is stored [11]. Hashes are also used to verify that the contents of two files or messages are identical [12]. Hashes are both injective and bijective [8]. The main difference between the hash and the cipher key used in comparisons is that only a very few mappings are ever used, and there is no way for an attacker to accumulate enough information to break the hash. For mappings, there is typically insufficient structure in the accesses to recover them. In this case, using a hash is not typically critical for security, and the data in the mappings changes frequently.

III. RESTATING THE 1:1 PRINCIPLE USING SETS

Let the PT alphabet ($|A|$) consist of sets of PT letters so that $\{A\} = \{\{PT_0\}, \dots, \{PT_n\}\}$. Similarly, let the CT alphabet be represented as sets of CT symbols in the CT alphabet (A') so that $\{A'\} = \{\{CT_0\}, \dots, \{CT_n\}\}$. The alphabet A' does not have to be the same as the alphabet A , nor is it required that $|A| = |A'|$. Nor is it necessary that the individual $|PT_i| = |CT_i|$, however, but most often true that $|A| \leq |A'|$. When $|A'| < |A|$, it is due to the case where not all letters are used in a file or message, or it may be that the number of sets of tokens in the mapping for $A \mapsto A'$ is less than $|A|$.

To accommodate the different types of valid key mappings in cryptography, the 1:1 principle should be restated as

$$\{PT\} \mapsto \{CT\} \quad (1)$$

This definition of the 1:1 principle is very general and permits all valid mappings of cryptographic keys.

There are five possible types of mappings under this new definition. These options are summarized in Table I. Each possible mapping type will permit all valid mappings to be considered separately.

- 1) Singleton - Also known as the traditional 1:1 mapping (single character to single character) [9]. Each PT in the input alphabet is uniquely mapped to a single CT alphabet character. There are no duplications of mappings to a CT character. That is, for two PT characters i and j

$$E_c(PT_i) = E_c(PT_j) \text{ iff } i = j \quad (2)$$

Duplications in a mapping are known as “collisions” [13]. In this cipher style, the PT and CT sets each consist of

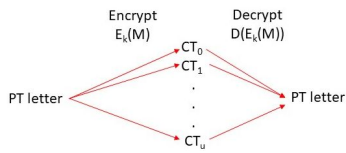


Fig. 3. One to Many back to One

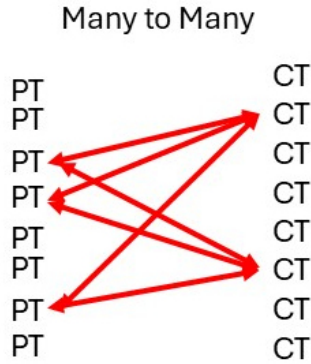


Fig. 4. Many-to-Many Mappings

a single member. Each PT and CT set has a cardinality of 1 ($\forall i \in \{A\} \rightarrow |PT_i| = |CT_i| = 1$). The traditional 1:1 mapping is used in most modern ciphers. This model is widely used, well understood, and proven effective in many situations.

- 2) One-to-Many - In this configuration, a single PT character can be mapped to one or more CT characters. All CT characters in the CT set map back to only the single PT that points to them. A representation of this case is found in Figure 3. The use of this model allows high-frequency PT letters to map to multiple CT letters, thereby altering a language's letter statistics [9].
- 3) Many-to-Many - Sets consisting of more than a single PT alphabet character to a set of multiple CT characters. This model is shown in Figure 4. Such a configuration can be very effective if the members of the PT set are easily distinguishable from one another in the language's normal token configurations, such as syllables, words, or m-grams [1].
- 4) Many-to-One - The model in which a set of PT tokens maps to a single CT token, and that CT token maps to a set of PT tokens. In this case, shown in Figure 5, several PT characters can be combined to make use of low-frequency alphabet tokens that are easily sorted from each other, making selection at the receiving node easy. The effect of this type of mapping is that fewer tokens are needed in the decryption alphabet, while also altering the message's PMF so that it no longer precisely matches the PT's PMF.
- 5) Mixed (hybrid) - In some cases, there will be a mix of mappings. Some PT tokens will map to a single CT token; others, depending on their frequency in the PT language,

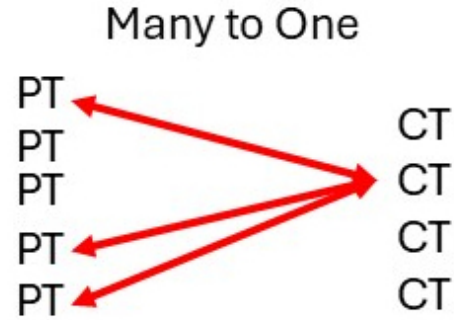


Fig. 5. Many to One Mappings

will map to multiple CT tokens. It is conceivable that the mappings will be a mixture of the basic mapping types discussed. The effect of using this mode of mapping is that it is possible to engineer the mappings for creating the desired cryptographic system response.

The purpose of this redefinition is to account for the possible encryption types available to the cipher designer. Rather than restricting ciphers to a singleton mapping, many other options are available. Users and designers should be aware of the various configurations and what each allows in terms of protection, as well as the inherent liabilities associated with each. This classification also frees up designers to explore different techniques and directions to develop more appropriate ciphers for different situations.

IV. SPECIFYING THE ALPHABET TO CREATE A HUFFMAN STYLE MAPPING FOR ENCRYPTION

In the context of cryptography, "One-to-One" refers exclusively to a function in which each input maps to exactly one output, and each output maps to exactly one input. However, there is another way to interpret the principle, and while it differs in this application, it is by no means useless.

Alphabets are most often thought of as lists of single characters, but in cryptography, this need not be the case. AES, for example, treats the set of all possible 16-byte strings as its input and output alphabets. In general, stream ciphers often map the alphabet of all possible 8-bit values of a language to the same alphabet ($A \mapsto A$). Block ciphers use the set of all possible strings of the block length as the input and output alphabet. Mappings from an alphabet of one character length to an alphabet of another character length are almost never used in cryptography, the one exception being hash functions, which can typically take an input of arbitrary size and map it to a fixed-size output.

Outside of the field of cryptography, this type of imbalanced mapping is much more common. For example, mismatched alphabet-sized mapping is the core mechanic used in data compression. A comparatively simple form of compression is Huffman Encoding [14], where a one-byte input alphabet is used, and characters that have high frequency in the input

message are mapped to sub-byte-sized output characters, and uncommon input characters are mapped to output characters greater than a byte in size. This ultimately reduces the data size without losing information, resulting in lossless compression.

In cryptography, techniques like this encoding mapping can also be valuable in other contexts. The most exploited linguistic property of encrypted data when attempting to crack an encryption is patterns from the original data that leak through the ciphertext. Compression algorithms work by removing inherent redundancy and encoding the input in a form with fewer patterns. As a result, there are fewer patterns that can leak through to be exploited. It also means that an attacker must be able to split bit runs at the appropriate points to process the input data correctly. This strategy of mapping an input alphabet to an output alphabet with variable character sizes shows potential value in cryptography.

It is not strictly necessary for the input alphabet to have a fixed size, either. This variable size of an output alphabet is something good compression algorithms use. For example, the word “the” is used extremely frequently in English (high probability and frequency). If instead of, or in addition to, mapping the individual characters of this word to output characters a few bits (b) wide, then the word is represented by a string of size $3b$. However, if the entire word is mapped to a character that is b bits wide, the compression ratio improves considerably. Using the same technique for many extremely common words, at the expense of longer outputs for very uncommon words or individual characters, can dramatically improve the overall compression ratio. This common strategy is used in the best compression algorithms. Not all common words will produce the same-sized output when mapped. The symbol size varies across words, making it difficult to guess the token boundaries. Patterns then become much more difficult to isolate and use in decryption.

In the context of polymorphic encryption, these techniques play an even more interesting role. The mappings have the same effect as implementing randomly sized sharding. Without knowing the exact, variable-sized mappings, an attacker must guess where to divide the language tokens (character/metacharacter boundaries).

Variable compression, such as Huffman-type mappings, also provides another unique benefit for polymorphic encryption: encoding of characters and words that cross byte boundaries. Splitting shards at byte boundaries with varying bit-sized token representations (encryptions) can sometimes produce a shard that will contain only part of the encoding of a character or word, while the next shard contains the rest of it. Some bytes may even contain fractionally encrypted shards. This may sometimes result in sub-bit level splitting. For example, if an 8-bit character is encoded in 3 bits (see Figures 6 and 7), there is no way to split those three bits that does not result in sub-bit splitting of the original 8-bit character. A 1 by 2 split of the post-compressed data is equivalent to a $2\frac{2}{3}$ by $5\frac{1}{3}$ split of the original 8-bit character. In practice, this means that if a character or word is split across shards, both shards must be decrypted to fully decompress it. Therefore, it may

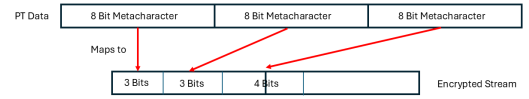


Fig. 6. Huffman Mapping with Compression

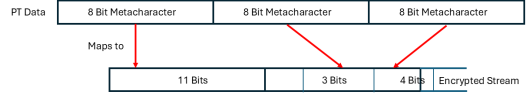


Fig. 7. Huffman Mapping with Expansion

be impossible to decompress decrypted shards with fractional characters at their boundaries without decrypting both adjacent shards. Further, it may be difficult or impossible to decompress any series of shards where at least one shard boundary does not end on a character boundary. This will depend on the specific compression algorithm used.

The Huffman compression/encryption technique increases both entropy and entropy density. Entropy is increased by splitting shards in the middle of metacharacter tokens. Entropy density is increased by using compression techniques to reduce redundancy. The overall result is significantly increased security by reducing the possibility of detectable patterns from the plaintext leaking into the ciphertext and by creating dependencies between shards that must be decrypted (or cracked) independently.

V. DISCUSSION

Models serve as a vehicle for understanding how and why various protocols and systems operate. By examining the various parts of a model, approaches to customizing and using the system to the user’s advantage are suggested. In the case of redefining the 1:1 principle, the newer model allows categorizing which techniques are useful and how to implement them. This mode identifies which mappings are useful for designing ciphers that are more resilient to attack and clarifies how the tools available in the mode can be used to quickly and effectively sculpt the operation of those ciphers. The goal is to create ciphers that appear to originate in other languages or disrupt the natural patterns of the language [15], [16]. Ideally, the goal would be to change the patterns and probabilities of the language statistics so as to give a uniform distribution of tokens, thus breaking the tendency of S ciphers to leak the original patterns in the source language. Using the five types of mappings shown in Table I, the model will be discussed in more detail.

The previous framing of the 1:1 principle is a subset of the newer definition. In the new list, it is the first application of set-to-set mapping. In this pairing, each set used is a singleton set. That is $\forall sets \in mapping \rightarrow |S| = 1$. These sets are called “singleton,” or “unit” sets [17].

Singleton-set mappings to singleton sets are characteristic of both hashes and S ciphers. The math of both functions shows that they are closely related, and that techniques used for either

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X/Z	Y

TABLE II
A SAMPLE BOX CIPHER

function provide insight into how to approach the other. If the input set is denoted as PT and the output set is CT , and the tokens are in their respective alphabets, then the result is a key mapping. This is a proven cipher type that encompasses most practical ciphers in use today and those used historically.

The second case is when the PT set is a singleton, but the output token set consists of more than one member. Such schemes are known as “homophonic” encryption [18]. Has not been used to design a cipher, only to mix up language statistics. Has never gained much traction. Requires a larger CT alphabet and may take longer to transmit if there is an increased size of token encoding due to the larger CT alphabet. Also, most cryptographic instructors emphasize the need to adhere to the singleton 1:1 principle definition.

The third case is one in which a set of multiple PT tokens is mapped to a single character. Mappings of this type have been used in prior ciphers. This cipher type is known as a “polyphonic” cipher [19]. Known since at least the late 16th century in Italy [20], this type of cipher has been characterized by ciphers such as the Vigenere and Enigma ciphers [4].

In another instantiation, the letters of a PT alphabet are arranged in a square matrix. There is a difficulty with this type of arrangement, as most languages do not have an alphabet size such that $\sqrt{|A|} \in \mathbb{Z}^+$, a positive integer. Therefore, in order to be able to use such a structure, one or more PT symbols are combined as a set in one or more boxes.

Letters can be distributed in any manner desired, with any two characters combined for one of the boxes. Once the contents of the box are defined, Various rules may be used to encrypt letters.

One such use of the many-to-one model is to take all of the PT s in the message that never appear and combine them into a set that maps to a single letter in the CT alphabet. Another is to collect very low-frequency letters, such as q, k, x, v, and z, into a single set. In that set, the user can easily distinguish which letter belongs in the decrypted text and use that mapping. If it is found that some tokens are largely orthogonal to each other, those letters can also be placed in sets with each other, and a single CT token can represent both. Using this method, letter mappings can be saved for other high-frequency tokens.

The advantages of this approach are that low-frequency and orthogonal tokens can be combined, reducing the number of CT tokens required for secure encryption. This can alter the system’s PMF, making it much more difficult for an attacker to break the cipher. However, the constraint of multi-element sets of PT tokens makes effective design more difficult and may even be impossible to impose. The practice may even

weaken the cryptographic algorithm’s security.

The fourth possible mapping scheme is one in which more than a single element is collected into each PT mapping, and the mapping points to a similar multi-element CT token set. As with the many-to-one model, CT tokens are stored in the practice so they can be consumed to encrypt other sets. This practice will spread the letter-statistical probability across the PMF for the encryption algorithm, but the disadvantage is having to decide which PT tokens are orthogonal enough to be disambiguated from one another in the normal course of using the language. If the letters cannot be reliably differentiated from each other in practice, this model should not be employed.

Perhaps the most useful of the models is that of the hybrid mapping construct. This is the fifth model and allows mixing any or all of the other mappings as it makes sense to use. That means low-frequency or orthogonal PT tokens can be combined into a set (or sets) to reduce the CT token count. Single PT tokens can be mapped to other singleton CT tokens to preserve the PMF frequencies in an effort to make a uniformly (or more uniformly) distributed PMF. In addition to this practice, the PMF may be designed to simulate that of another language. Cryptonulls [21] can also be added in this manner using the many-to-many or one-to-many models, also to alter the PMF for the language.

One of the advantages of the hybrid model is that an attacker must recover the mapping types before they can concentrate on the mappings of individual sets. Unused PT s can be consolidated, and the substitutions used can be crafted in a variety of ways in the same cipher algorithm. This model allows for the special-purpose design of different ciphers. One of the unique design capabilities is to create rotating polymorphic ciphers of various types and configurations to confuse attackers further and increase security. The main disadvantage is that using this model can be more involved and difficult to design.

Any one of these models may result in a CT alphabet that is larger than the original PT alphabet. If that CT alphabet is larger than 256 characters, the number of bytes required to represent tokens in the CT alphabet could result in a message that is longer than the original message. Transmission time may be affected, and it may take longer to send the encrypted message than the PT message. However, depending on the message’s importance, this constraint could be acceptable to the user.

A summary of the uses for each type of mapping is found in Table III. Each type of set mapping is covered, and it provides guidance on what each mapping accomplishes—combining the first four types results in the application of the fifth type - a complete design kit.

The use of this model has several advantages. First, the PMF of a transmission can be changed so that it does not correspond to any known natural language [15]. There are sufficient possible mappings available that a message PMF can be made to resemble any language, allowing for the variance in the use of a language by any particular practitioner of a language [16]. If it is correctly mapped, it may be possible to

Mapping	Primary Use
1:1	Traditional encryption mapping
1:many	Break up frequency to disguise it
many:1	Combine low frequency, null, and unused characters, combine orthogonal characters
many:many	Combine orthogonal characters
hybrid	Complex cryptographic designs

TABLE III
PRIMARY USE OF VARIOUS SET MAPPINGS

make the CT PMF of an encrypted message resemble that of a completely different message. In effect, this model allows for “designer” PMFs.

Hiromoto et al. [9] presented a method that, given enough CT tokens, could map a PT message to a CT output with a uniform distribution. This is much more effective if a hybrid mapping scheme is used to sculpt the output. Such a transmission requires knowledge of the PT content and the PMF to more easily design the cipher mappings. The conclusion is that, to be perfectly secure, the user must understand the relationships among the message patterns and how they relate to the PMF. The encryption must be matched to the message type and content.

Most heuristic [22], [23] attacks are based on language patterns. Those language patterns are encoded in the PMF of a language and its related messages. Entropy can be introduced into encryption by altering the data in the PMF mappings. The information is still present, but unless the attacker knows how the mappings in the key are combined, the original PMF is only one of many. Using these mappings disrupts the PMF for a language, making it much more complicated and potentially irrelevant.

VI. CONCLUSIONS AND FUTURE WORK

The 1:1 assumption in its present form has been used to guide cipher design for as long as the S cipher has been used. However, that principle proved inadequate with the introduction of polyalphabetic and polyphonic encryption. Both of these encryption methods use mappings, either from a set of PT tokens to a single CT token or from multiple CT tokens back to a single PT token. Including these mapping methodologies in a more general definition is required.

This paper presents a definition that allows all types of mappings, using sets of tokens as the basic unit of keys rather than single tokens in a language/alphabet. This leads to five possible types of mappings: singleton set to singleton set; singleton set to multiple-token sets (multiple-token sets); multiple-token sets to multiple-token sets; and a hybrid mixture of any type of set to any set.

Examples of each mapping allowable under the new definition were presented, along with their advantages and disadvantages. This data, along with the model, enables a system that provides a systematic way to develop ciphers using a compact representation and the ability to mix different effects. Engineers can then use both the model and the information to design ciphers that can change the message’s PMF to resemble

another language, or no language at all, depending on the designer’s requirements.

The proliferation of new ciphers can create a much wider library of cryptographic functions available to security professionals. Libraries composed of these and previously designed ciphers can be used to implement polymorphic engines to protect messages.

REFERENCES

- [1] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656 – 715, 1949.
- [2] David Powers. Applications and explanations of zipf’s law. *NeM-LaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning*, pages 151 – 160, 1998.
- [3] Robert Lewand. *Cryptological Mathematics*. Mathematical Association of America, Washington D.C., 2000.
- [4] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons Inc., New York, 2nd edition, 1996.
- [5] Bhaskar Ghosh, Indira Dutta, Shivanjali Khare, Albert Carlson, and Michael Totaro. Isomorphic cipher reduction. *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2021*, pages 947–953.
- [6] University of Notre Dame. Letter frequencies in the english language, <https://www3.nd.edu/~busi-forc/handouts/cryptography/letterfrequencies.html>.
- [7] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5 – 83, 161 – 191, 1883.
- [8] Ted Sundstrom. *Mathematical Reasoning: Writing and Proof 2nd Edition*. Prentice Hall, 2003.
- [9] Benjamin Williams, Albert Carlson, Robert E. Hiromoto, and Sai Ranganath Mikkilineni. Polymorphic hashing to obscure the 1:1 principle. In *IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC 2026)*.
- [10] Richard Wells. *Applied Coding and Information Theory*. Prentice Hall, Upper Saddle River, 1999.
- [11] Kurt Mehlhorn and Peter Sanders. *Algorithms and Data Structures*. Springer.
- [12] National Institute of Standards and Technology (NIST). Guide to integrating forensic techniques into incident response , sp 800-86, <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-86.pdf>.
- [13] Albert Carlson, Bhaskar Ghosh, and India K. Dutta. Using the collision attack for breaking cryptographic modes. *13th International Congress on Computing, Communication, and Networking Technologies*, Oct. 3 - 5, 2022.
- [14] David Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098 – 1101.
- [15] James Allen. *Natural Language Understanding*. Benjamin/Cummings, Redwood City, 1987.
- [16] D. Terence Langendoen and Paul Postal. *The Vastness of Natural Languages*. The Camelot Press, Ltd., Southampton, 1984.
- [17] Robert Stoll. *Sets, Logic and Axiomatic Theories*. W. H. Freeman.
- [18] 1974 Fred A. Stahl, University of Illinois. A homophonic cipher for computational cryptography.
- [19] Brian Lennon. *Passwords: Philology, Security, Authentication*. Harvard University Press.
- [20] Aloys Meister. *Die Geheimschrift im Dienste der Papstlichen Kurie von ihren Anfängen bis zum Ende des XVI. Jahrhunderts*. Creative Partners, LLC.
- [21] Adam Gordon. *Official (ISC)2 Guide to the CISSP CBK - Fourth Edition*. (ISC)2 Press.
- [22] N. Nalini and G. Raghavendra Rao. Attacks of simple block ciphers via efficient heuristics. *Information Sciences*, 177(12):2553–2569, 2007.
- [23] Oxford Dictionary. "heuristic", [https://www.oed.com/search/advanced/meanings?](https://www.oed.com/search/advanced/meanings?texttermtext0=heuristic&termopt0=wordphrase), <https://www.oed.com/search/advanced/meanings?texttermtext0=heuristic&termopt0=wordphrase>.