Regis University
ePublications at Regis University

All Regis University Theses

Spring 2010

The Rubik's Crypto-Cube: a Trans-Composite Cipher

Daniel R. Van der Vieren
Regis University

The Rubik's Crypto-Cube: A Trans-Composite Cipher

A thesis submitted to
Regis College
The Honors Program
in partial fulfillment of the requirements
for Graduation with Honors

by

Daniel Robert Van der Vieren

Advisors:
Dr. James Seibert and Dr. Tim Trenary
Department of Mathematics

May 9, 2010

Contents

Acknowledgements

To Dr. Seibert and Dr. Trenary. Without your constant encouragement and support, this project would not have been a reality. Both of you made math fun.

And to all those who continued to inquire about my thesis progress. You all kept me honest.

Chapter 1

Codes and Cubes

1.1 Origins of Cryptography

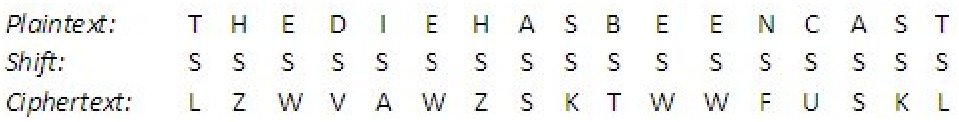
Cryptography, the art or science of writing messages in code to disguise the content, has been a source of interest for millenia. Those who exchange secret messages do so through the medium of a *cryptosystem*, a single set of devices used in order to encrypt plaintext and decrypt ciphertext. Encrypting involves changing *plaintext*, a message in an intelligible state, into *ciphertext*, the message in an unreadable form. The ciphertext confuses adversaries, but by using the properties of the cryptosystem, the receiver can decrypt the ciphertext back into the original message.

For thousands of years, humans have tried to devise methods of hiding messages from enemies. Secure communication prevents other nations from intercepting sensitive material, and the use of codes and cryptography continue to assist in maintaining security of personal information. Originally, the exchange of messages occurred via horseback or foot. Now in the 21st century, however, the technological advances allow us to correspond with computers, phones, and other devices.

1.2 The Caesar Cipher

One of the earliest forms of cryptosystems to be historically-documented is the Caesar cipher. This cipher found application in times of war, where military officers could transfer messages between eachother. The advantages of sending such correspondences are obvious. The scheme involved changing a plaintext message by shifting the letters a specific distance, by consequence creating a message that was unreadable. Below, the message "THEDIEHASBEENCAST" can be shifted by a specific letter (e.g. "S"). The letters of the alphabet can be represented as numbers, with A $= 0$, B $= 1$, C $= 2$, etc. Adding the letter pairs together, we see a new cipher-letter. Notice that there is a wrap-around that occurs when two letters near the end of the alphabet add up to a number greater than

25, or 'Z'. When a 'S' is added to another 'S,' the result is 'K.' This wrap-around results from modular arithmetic. In addition modulo 26, 16 added to 20 is equivalent to 10 since 36 divided by 26 has a remainder of 10. Every multiple of 26 is equivalent to zero. Other cryptosystems will use modular arithemetic in the encryption process.



Plaintext:	T	H	E	D	I	E	H	A	S	B	E	E	N	C	A	S	T
Shift:	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Ciphertext:	L	Z	W	V	A	W	Z	S	K	T	W	W	F	U	S	K	L

Figure 1.1: A Caesar Shift Cipher. Shift = 'S'.

This shift cipher has a well-known weakness. Since there are only 26 letters in the English alphabet, three of these letters not even existing at the time of the Romans, an ancient cryptanalyst would have little difficulty decrypting the ciphertext. *Brute force*, an attack on a cryptosystem that employs trying every possible "option" for the plaintext, can take a matter of minutes with this particular system. An attacker can reverse the shift of the ciphertext by "subtracting" to find the plaintext. If the adversary is lucky, he or she will find the plaintext before exhausting all shift possibilities.

1.3 Necessary Components to a Cryptosystem

Those who exchange secret messages use a key. A *key* in cryptography is any exchanged word or clue known by both the sender and receiver that assists in the decrypting a ciphertext. In the example above, the key would simply be the letter 'S'. In other cryptosystems, the key might be a word or phrase that signals a change to the plaintext or a clue about how to read a ciphertext in order to decrypt the message.

Keys are exchanged between two individuals. For our example we will call these two *Antony* and *Caesar*. An adversary, *Brutus*, will often attempt to get in the way of the message exchange, and will do anything in his power to intercept the message and exploit its contents. Due to the potentiality of the message falling into the hands of Brutus, Antony and Caesar must share a secret key to prevent the reading of the message by Brutus or another enemy.

The security of the key is vital. If an adversary were to discover the key, as well as the decryption method of the cryptosystem, then all communication between two parties using this system can be deciphered. *Kerckhoffs' principle*, one of the most important published principles for military ciphers created in the late 19^{th} century, explains the significance of the key in cryptography:

> *"The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience."* [4]

This says that there is no requirement for the encryption scheme to be secret, but rather just the key, *k*. Kerckhoffs intended the actual scheme itself to be secure enough so that the secrecy of the key would suffice in maintaining security despite an adversary having knowledge of the system's algorithms. We must always assume the enemy knows the method of encryption and decryption. With this in mind, the key k must be protected.

Three arguments support Kerckhoffs' principle. First, it is easier for communicating parties to exchange and remember a short key than maintain the secrecy of a large and complicated algorithm. It is quite difficult and cumbersome to secretly share and store a program that is thousands of times larger than a key stream. The second argument explains that in the hypothetical situation that a key is lost or leaked to dishonest parties, the individuals trying to maintain secrecy can refresh a key. Replacing an entire algorithm and its software is an unwieldy burden. The third argument supports the use of multiple keys amongst numerous individuals in an organization; without multiple programs or algorithms, a variety of keys instead facilitates communication exchange.

Kerckhoffs wasn't alive to share this insight with Julius Caesar. Since the substitution cipher has only 26 different shifts, brute force attack could crack any code in a matter of minutes by hand. Cryptanalysts with a purpose have the motivation to break these codes, and for this reason, substitution is hardly ever used in the real world.

In creating a new cryptosystem, security is a primary concern. The intercepting party may want to either read a particular message, find the key to read all transferred messages, corrupt the message before it gets to its receiver, or pretend to be the sender and communicate with the receiver without his or her knowledge. Depending on the situation, the security may become even more important. Efficiency of the cryptosystem, on the other hand, is not necessarily essential to security of the system, but there a cryptosystem that does not require much effort to encipher and decipher is preferred to one that is inconvenient for both parties. Knowing what security means in the world of cryptography can provide assistance in creating the cryptosystem. "A cryptographic scheme for a given task is *secure* if [and only if] no adversary of a specified power can achieve a specified break" [4]. This definition, however, does not make any assumptions on the strategy of the individual or "power." Additionally, there is no assumption being made about how the abilities are implemented.

1.4 The Substitution Cipher

Substitution systems are not strong enough to confuse a creative adversary. Even the substitution of letters in the message for other letters in the alphabet (e.g. 'A' representing 'F,' 'B' representing 'Z', etc) is not difficult to crack. There are $26!$ ways (4.0329×10^{26}) of selecting the arrangements of the substitutions, but for those readers familiar with *Cryptoquotes* in the newspaper, we can see that knowledge of common words provides assistance in breaking the cipher relatively quickly.

The Cryptogram in Figure 1.2 is not too difficult to solve, especially since the words are broken up with spaces. Words like "LITTLE," "THE," and "ONE" can be deduced from the grouping of letters. In the real world, however, spaces might not be present, and thus knowledge of vocabulary cannot be used to decipher the message. Therefore, we

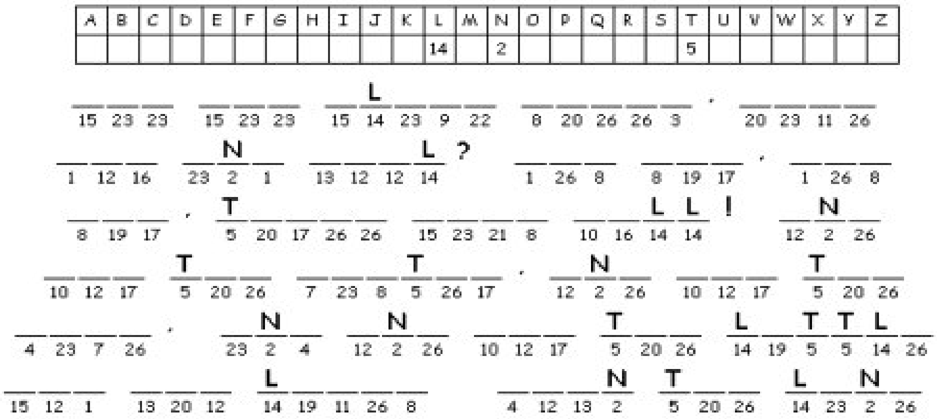


Figure 1.2: Cryptoquote. Can you solve it?

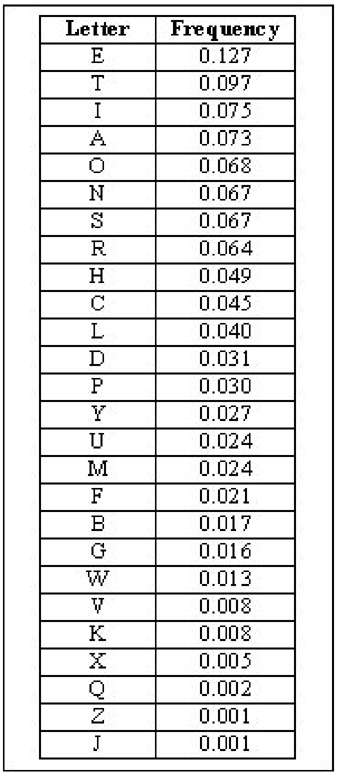
must utilize another method: frequency analysis. *Frequency analysis* is an attack on the system which exploits the fact that certain letters of the alphabet appear more often than others in literature. A frequency table breaks down the letters by probability of occurrence in any message.

Using this information, we can count the number of occurrences of a particular character in the ciphertext, and substitute the letter occurring most often with an 'E' or a 'T.' Continuing down the table (Figure 1.3), we can find other common letters in the ciphertext until all are replaced. Sometimes, it is necessary to substitute letters that don't always have as common of occurrence as those in the ciphertext, but certainly this process is not as difficult as brute force with the checking of 26! permutations.

1.5 The Transposition Cipher

A *transposition cipher* is a system that relies on the use of the same letters between the plaintext and ciphertext, but the ciphertext scrambles the letters in order to hide a message. Instead of a permutation of the letters as in the substitution cipher, there is rather a permutation of placement in this system. Often, transposition ciphers will take a portion of the plaintext and fit it to a rectangular block (i.e. 5×5, 6×9, etc), and then order the letters in a the block for instance with the message written horizontally. An example would look like Figure 1.4.

Notice that the transposition block uses a forward and reverse key, which is a set of numbers describing the order of the columns. The forward key from the figure, "2 5 1 3 4," will create the ciphertext. The '2' over the first column moves this column to the second ciphertext column, the '5' over the second plaintext column situates this column at the furthest right side of the transposition block of the ciphertext. The chosen method of organizing the ciphertext for this example is by rearranging the columns first, and then reading left to right, first row, then second row, etc. The message, "IFMATH-

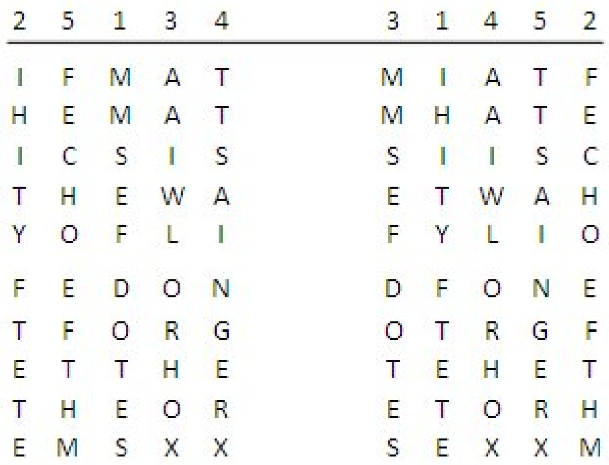


Letter	Frequency
E	0.127
T	0.097
I	0.075
A	0.073
O	0.068
N	0.067
S	0.067
R	0.064
H	0.049
C	0.045
L	0.040
D	0.031
P	0.030
Y	0.027
U	0.024
M	0.024
F	0.021
B	0.017
G	0.016
W	0.013
V	0.008
K	0.008
X	0.005
Q	0.002
Z	0.001
J	0.001

Figure 1.3: Frequency Table

EMATICSISTHEWAYOFLIFEDONTFORGETTHETHEOREMS" will be jumbled to the ciphertext, "MIATF MHATE SIISC ETWAH FYLIO DFONE OTRGF TEHET ETORH SEXXM". An adversary who knows the sending and receiving parties well might guess that the message has something to do with "mathematics" due to the prevalence of 'M's', 'A's' and 'T's' in the ciphertext.

In deciphering the code back the the original plaintext message, the decoder will place the columns in the order that the reverse key denotes. For the example above, the '1' column will be moved to the far left, and then the '2' column moved next to the first until the five columns are situated in the original block ordering. The problem with this strategy, though, is that cryptanalysis can break this code in a matter of seconds. The columns being split and rearranged back to the original order would create intelligible plaintext in one out of 5! ways. It most likely wouldn't take all 120 tries, even working with a pad and paper, as the pairings of letters noticed by the cryptanalyst would significantly narrow the permutations.



2	5	1	3	4
I	F	M	A	T
H	E	M	A	T
I	C	S	I	S
T	H	E	W	A
Y	O	F	L	I
F	E	D	O	N
T	F	O	R	G
E	T	T	H	E
T	H	E	O	R
E	M	S	X	X

3	1	4	5	2
M	I	A	T	F
M	H	A	T	E
S	I	I	S	C
E	T	W	A	H
F	Y	L	I	O
D	F	O	N	E
O	T	R	G	F
T	E	H	E	T
E	T	O	R	H
S	E	X	X	M

Figure 1.4: Transposition Cipher; Plaintext and Ciphertext

1.6 The Rubik's Cube and Its Magical Potential

Erno Rubik, a Hungarian teacher of architecture and design from Budapest, devised a toy that could rotate around three axes and create over 43 quintillion permutations for its $3 \times 3 \times 3$ patent. By 1975, the toy was completed, intriguing the world. The Cube, due to its sizeable number of permutations of states, as well as its inherent group properties, has potential to be used for cryptography. The individual squares of the face, called "cubies", are perfect locations for the placement of plaintext letters. With enough twists and turns of the well-known cube, ciphertext can be created.

The Rubik's Cube continues to have application in group theory due to its puzzle-like properties. The faces, each with a different color, are broken up into nine different "cubies", one central cubie which serves as a rotational axis, and eight outer cubies that can be moved and manipulated to alter the cube's state from the solved, or "start" position.

1.7 Counting the Permutations of the Cube

Initially, it appears that the counting of the cube can be a simple multiplication of factorials, using the fact that each cubie may have a certain orientation and possible position. However, we will see this not to be the case. There are eight corner cubies. Selecting one out of the eight for one corner, and then one out of the remaining seven in the next corner, and so on until the corners are all filled, will have an initial counting of 8! arrangements of the corners. Moreover, each of the corner cubies can be oriented in one of three ways. The reason that there are not six orientations of the corners is due to the fact that the cubies cannot have the stickers removed to create more permutations. Any

Figure 1.5: The Rubik's Cube in Its Solved State

rotation around the axis will have an order of three. Using the corner cubie below, if one were to rotate the cubie clockwise, the yellow face would move to the spot where the red face is located, and the red to the blue, etc. There are only three rotations possible.

From the basic counting strategy, using this factor of three orientations for eight cubies, giving an additional multiplier of 3^8. On top of that, the permutations of the 12 edge cubies, $12!$, and then the two orientations for each of the 12 edge cubies (those in between the corners) would multiply the number of permutations by 2^{12}. The total number of permutations using these calculations would be: $8! \times 3^8 \times 12! \times 2^{12} \approx 5.1902 \times 10^{20}$. Because the Cube is a closed system, we cannot assume that every orientation can be attained with the given cubies. There is a relationship between the rest of the cubies that require the number of permutations to be re-evaluated, and even reduced further.

When orienting the cube and selecting the states, one can position seven of the eight corners in any orientation. However, the last cubie is determined in one position, with one orientation. This reduces the estimate by a factor of three. The edge cubies operate similarly: 11 out of the 12 edge pieces can be positioned with either orientation, but this determines the orientation of the final cubie. This reduces the estimate by a factor of two. Lastly, "there is one final constraint on the permutations of cubies (disregarding their orientations) that says you can place all but two of them wherever you want but the last two are forced" [3]. This further reduces the estimate by a final factor of two. Dividing the initial number, 5.1902×10^{20} by 12, we get the final number of permutations of the $3 \times 3 \times 3$ Rubik's Cube: 43,252,003,274,489,856,000. If the every cube state was represented in a column of cubes, this collection would span 250 light years. So at first glance, 43 quintillion might seem like a large number, but in actuality in the world or cryptography and computer technology, this figure is quite insignificant. Even brute force would have the capability of cracking codes with a keyspace of this size.

Practically, the Cube is perfect. At an airport, would security suspect the cube of being anything but a toy? The Cube is not suspicious, as it comes across as merely a puzzle, not a spy tool. Due to this, transporting it as a cryptic device can be easily disguised. There is some merit in using the Cube for cryptography, even with the relatively small number of permutations possible.

Chapter 2

The Vigenère and A One-Time Pad

2.1 The Vigenère Cipher

The Vigenère Cipher dates back to the 16th century. This particular system is also called a "polyalphabetic shift cipher" due to the fact that the plaintext characters can be mapped to several different ciphertext letters by means of a specific shift, or keyword. A *keyword* is an grouping of characters, often an actual word, that is employed in the system to alter the plaintext to an unreadable ciphertext. Instead of a single letter altering the plaintext, as in the regular shift cipher, the keyword changes the message in the Vigenère cipher. For example, a given plaintext could read: "GREETINGSMATHEMATICIANS". A keyword of a certain length would be "added" to the plaintext to create a new message. If the keyword was "HELLO," for instance, we would add this to the given plaintext, repeating if necessary until all the letters were altered to create a ciphertext. Below is the resulting ciphertext using this example.

G	R	E	E	T	I	N	G	S	M	A	T	H	E	M	A	T	I	C	I	A	N	S
H	E	L	L	O	H	E	L	L	O	H	E	L	L	O	H	E	L	L	O	H	E	L
O	W	Q	Q	I	Q	S	S	E	B	I	Y	T	Q	B	I	Y	U	O	X	I	S	E

Table 2.1: Plaintext; Keyword Added; Ciphertext

The keyword is represented as a vector. It is customary to have the vector with numbers corresponding to the letters of the alphabet. With the numbers 0 to 25, $(a = 0, b = 1 \ldots)$, the keyword k for this example would be the vector $(7, 4, 11, 11, 14)$. The choice of word and length should be known only by those exchanging the ciphertext. If the security of these two pieces of information was ever compromised, the system would easily be broken.

2.2 Cracking the Vigenère Cipher

Cracking of the Vigenère cipher is not quite trivial, especially if the keyword is particularly long. After several hundred years of being considered an "unbreakable cipher," the Vigenère cipher was finally broken.

Breaking the Vigenère cipher can be somewhat complicated, but certain details about the system facilitate the cryptanalytic attack. Often, a cryptanalyst will only have the ciphertext to read. The first step in deciphering requires the knowledge of the key length. The process involves first taking two copies of the ciphertext and displacing both (one above the other) by a specified number of places. Below is an example of a displacement of three:

			O	W	Q	Q	I	Q	S	S	E	B	I	Y	T	Q	B	I	...
O	W	Q	Q	I	Q	S	S	E	B	I	Y	T	Q	B	I	Y	U	O	...
					*														

Table 2.2: A Displacement of Three

We notice that there is a point where the corresponding letters in a column match. This is called a *coincidence*. In order to determine the key length, we count the number of coincidences at every displacement. The message here is not quite long enough to provide us with a substantially different number of coincidences at varying displacements, but with a long enough message this soon becomes apparent. The maximum number of coincidences will occur at a particular shift, and multiples thereof. This information will yield the key length.

Once the key length is known, we can implement a method to break the Vigenère cipher. Using the above example and the assumption that the key length is 5 – without knowledge of the actual characters which comprise the key itself – we can determine the actual key. Frequency analysis helps determine the ciphertext.

Trappe highlights the process. The first step in finding the keyword is to place the frequencies of English letters into a vector, $\mathbf{A}_0 = [P(\mathsf{A}),\ P(\mathsf{B}),\ \ldots]$ where $P(\alpha)$ is the probablility of the occurence of a letter, α:

$$\mathbf{A}_0 = (0.082,\ 0.015,\ 0.028,\ \ldots,\ 0.020,\ 0.001).$$

We can let $\mathbf{A}_i$ be a new vector when $\mathbf{A}_0$ shifts i spaces to the right. For instance,

$$\mathbf{A}_1 = (0.001,\ 0.082,\ 0.015,\ \ldots,\ 0.020).$$

We can take the dot product of $\mathbf{A}_0$ with itself. This yields:

$$\mathbf{A}_0 \cdot \mathbf{A}_0 = (0.082)^2 + (0.015)^2 + \ldots (0.001)^2 = 0.066.$$

Any $\mathbf{A}_i \cdot \mathbf{A}_i$ will be equal to 0.066, since the sum of the products is identical, with just a different starting term. When $i \neq j$ for $\mathbf{A}_i \cdot \mathbf{A}_j$, the dot products are much lower. The dot product depends on only $|i-j|$, and so it is only necessary to compute up to $|i-j| = 13$ (half the distance between 0 and 26, the number of letters in the English alphabet).

When we have shifts i and j, the probablility that we have a coincidence (from Table 2.2) is equal to $\mathbf{A}_i \cdot \mathbf{A}_j$. In particular, we will select a displacement of 5, so $i = 0$, and $j = 5$. The probability that the letters are the same when $|i-j| = 5$ is found by the dot product $\mathbf{A}_0 \cdot \mathbf{A}_5$, which is equal to:

$$\mathbf{A}_0 \cdot \mathbf{A}_5 = [P(\text{A})][P(\text{F})] + [P(\text{B})][P(\text{G})] + \ldots + [P(\text{Z})][P(\text{E})].$$

When $i = j$, the shift of each letter is the same amount during encryption. This occurs when the displacement is equal to key length, k. The dot product will be 0.066 in this case. Multiplying the number of comparisons (the actual key length displacement subtracted from the total number of letters in the ciphertext) by the dot product 0.066, e.g. with key length 5, we calculate $23 \times 0.066 \approx 1.5$ coincidences. The actual number of coincidences is 4 for our message example, but with longer messages, the margin of error is much less. As in the table, the calculation of coincidences will be close to the number of '*'s.

We can create a vector, $\mathbf{W} = [P(\text{A}),\ P(\text{B}),\ \ldots,\ P(\text{Z})]$, where the probabilities of a particular α are equal to the number of occurrences of a letter in the i^{th} position, the $(i+k)^{\text{th}}$ position, the $(i+2k)^{\text{th}}$ position, etc divided by the total number of counted letters. The frequencies of the letters are given by this vector, $\mathbf{W}$, which approximates $\mathbf{A}_i$, i being the shift of the first element of the key.

The dot products can be calculated for $\mathbf{W} \cdot \mathbf{A}_j$ for $0 \leq j \leq 25$. The maximum dot product will be when $j = i$. The largest value will be equal to $\mathbf{W} \cdot \mathbf{A}_j$. From above, we can summarize the steps to finding the key, k with length, n:

1. Compute the letter frequencies in positions i mod n, and form the vector, $\mathbf{W}$.
2. Compute $\mathbf{W} \cdot \mathbf{A}_j$.
3. Have $k_i = j_0$ provide the $max\{\mathbf{W} \cdot \mathbf{A}_j\}$.

The key will most likely be $\{k_1, \ldots, k_n\}$.

2.3 Keystream for One-Time Pad with the Cube

2.3.1 One is the Loneliest Number

The *one-time pad* or *OTP*, patented as Vernam's Cipher in 1917, can obtain perfect secrecy. This cipher contains a keylength that is as long as the plaintext message being encoded. In 1917, perfect secrecy was an unknown concept, but this changed when C.E.

Shannon demonstrated 25 years later that a one-time pad can indeed achieve the level of perfect security.

A one-time pad can be used for any message, but the length of the key must be as long as the message itself. To explain the concept of the one-time-pad, we take the *bitwise exclusive-or* (XOR) $a \oplus b$, where a and b are binary strings. If $a = a_1, \ldots, a_k$ and $b = b_1, \ldots, b_k$, we would then have $a \oplus b = a_1 \oplus b_1, \ldots, a_k \oplus b_k$. We can then define the one-time pad with the following information:

1. Let $l \in \mathbb{Z}^+$, where $l > 0$. A message space, $\mathcal{M}$, a keyspace $\mathcal{K}$, and ciphertext space $\mathcal{C}$ are all equal to $\{0,1\}^l$, this being the set of binary strings with length, l.

2. The algorithm $\mathbb{G}$en for generating keys works by selecting a string from $\mathcal{K} = \{0,1\}^l$ according to a uniform distribution. Each of the 2^l strings in the space is selected as a key with probablility of 2^{-l}.

3. Given a key $k \in \{0,1\}^l$ and message $m \in \{0,1\}^l$, encryption creates the output $c := k \oplus m$.

4. Given a key $k \in \{0,1\}^l$ and ciphertext $c \in \{0,1\}^l$,, decryption creates original message $m := k \oplus c$.

When a binary stream of 1's and 0's represents the message, using the iterated XOR scheme will create a new message. Analagously, we can use a one-time pad with letters *mod* 26 instead of binary bits *mod* 2. The Caesar cipher and the Vigenère cipher can be viewed as a version of an OTP, but one with a non-random keystream.

With perfect security comes drawbacks. The limitation around the keylength of the one-time pad makes this system–and any other "perfectly-secret" system–virtually unusable. An additional issue is the fact that generating truly random keystreams is nearly impossible using a computer. Some creative ways to attempt at generating random numbers include connecting a geiger counter to a computer to transcribe motion of tectonic plates into numerical representation. Nevertheless, in most settings, including commercial ones, a one-time pad is not an option.

2.3.2 The Cube's Keystream

With the 4.3 quintillion permutations of the $3 \times 3 \times 3$ Cube, it is natural to think that there is quite some potential for its use as a keystream generator for possible use with the Vigenère cryptosystem. While there is indeed merit in the size of the keystream that could be generated using the cubies, that is length 54, the keystream will not be able to be completely random. The Cube is held to the properties of group theory, and there is the finite number of permutations to generate a keystream for a one-time-pad. A one-time pad follows the rules of the Vigenère algorithm, but there is no repetitious pattern to the keyword or keystream as in the Vigenère system. If one were to create a keystream by mixing up the Cube with the various letters written on the cubies, it would require that there is some adjustment of the Cube's state at each 54-character block. Otherwise, the keystream mimics the Vigenère cipher completely with the keyword being of length 54.

One weakness of using the Cube for a generator of a one-time-pad is the consistency of the letters of the center faces on the Cube. If a keystream for the first 54-character block looked like the following: "AHEO*F*MECUGNWP*L*TY...," the letters in italics would not change as long as the keystream blocks were read off in the same order every time during the encoding process.

OTP's present practical disadvantages, and are therefore unreliable in most cases. The length of the key must span the entire message to be completely secure, and using the key a second time is out of the question. Additionally, the weakness of key negotiation between the two communicating parties presents another concern; transferring the key over great distances must involve cryptological remedies to avoid interception.

2.3.3 Size of Keyspace

For a cryptosystem to be secure and efficient, it is often important to have a large keyspace. This means that the number of different keys in a given system, without overlap, will be a sufficiently large number. Fast computers do have capabilities of solving the ciphertext and attacking the system when certain pieces of information are known, following Kerckhoffs' Principle. From what we know about the permutations of the Rubik's Cube, it is obvious that the $3 \times 3 \times 3$ Cube will not provide an adequately sized keyspace. 43 quintillion keys would not be large enough for a computer to attack with much difficulty. Therefore, a new algorithm must be devised if the Cube is to hold merit as an integral part to a cryptosystem.

Chapter 3

The Mitchell Cryptosystem

Douglas W. Mitchell, a professor of Economics at West Virginia University submitted a proposal cipher system utilizing the Rubik's Cube toy to the mathematical magazine, Cryptologia, in 1992. Known primarily for research in theoretical macroeconomics and monetary economics, Mitchell diverted to cryptography. His interest in creating a transposition cipher inspired the article. The system, according to the article's abstract, "is secure against brute force attacks; since it permits a different scrambled ordering of letters for each letter block enciphered, it is also secure against multiple anagramming" [6].

Substitution and transposition encipherment are the two most basic forms for encoding and decoding messages. Mitchell explains that brute force attacks to find the "correct" ordering of the plaintext will be insufficient and unsuccessful. Those who are interested in creating a cipher system with the Cube should take into consideration the weaknesses presented with the shuffle algorithm when the Cube is not shuffled adequately. A cryptanalyst, when seeing a particular-length block of letters might think to use multiple anagramming in attempts to decrypt the message, but as explained later, these efforts would be in vain.

In general, using mechanisms for transposition ciphertext creation are not efficient in generating letters in a manner necessary for encoding and decoding. Such systems are designed to apply *polyalphabetic substitution*, the permutation of the alphabet, rather than *polygraphic substitution*, or the permutation of position. Mitchell's paper proposes and outlines an encryption device with the Rubik's Cube that hopefully provides fast encryption and decryption. Of course the meaning of "fast" is relative in this context, as computers have the capabilities of encoding and decoding much more quickly than human hands, and with the assistance of a program, Mitchell's system could work at a faster rate.

3.1 The Mechanics that Apply to Mitchell's System

The original Rubik's Cube, a 3×3 arrangement of 27 mini-cube faces, has six sides, with nine cubies each. The rotation of the cube allows for movement about the three axes such that a turn of a face by a multiple of $\pi/2$ will set the Cube back to its original shape, but with an alteration of the cubies. The Cube's portability aids in the encipherment process as well. As a game, the Cube is a puzzle. The solved state is unique in that the color of each face is solid. "From this initial configuration, various rotations are randomly performed so as to jumble the colors; the object of the game is to restore the cube to the initial configuration with one color per cube face" [6]. Although the colors are not drastically important in the ciphersystem for Mitchell's purposes, it helps to keep the sides distinguished. The ciphersystem suggested by Mitchell with the Rubik's Cube has immediate appeal due to the shuffling potential. Undoubtedly, the obvious qualities that stand out attract the eye of cryptologists.

3.2 The System and How it Works

Mitchell's ciphersystem involves some sneaky steps in order to fix the plaintext. He suggests coating the Rubik's Cube in order to use ink on the faces for the transposition algorithm iterated later. The first step in the process requires one to write the numeral "1" on the upper left square of a cube face. The number "2" can be on an arbitrarily chosen square on another face, and so on until all six sides have a characteristic numeral as its representative identifier. Taking the top row, one may then write the plaintext (the first 48-letter chunk) on the remainder of the cube faces starting at the top row and writing left to right. Below is the first cube that would be written with the quote "IFMATHEMATICSISTHEWAYOFLIFEDONTFORGETTHETHEOREMS".

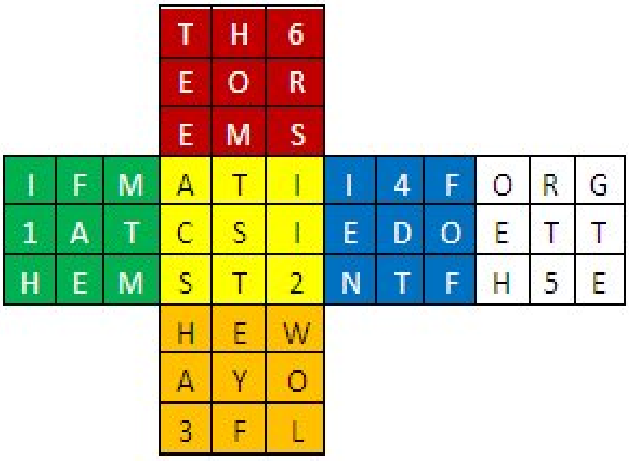


Figure 3.1: Mitchell Initialization

An important feature of the Cube is that the message chosen can be encrypted in a large number of ways. Because of this, the system "defends against multiple anagramming attacks" [6]. After the first side is labeled with a "1", there are $5!$ ways to select the order of the remaining faces with the numbering system stated above. The locations of the remaining plaintext letters (for the $3 \times 3 \times 3$ cube, the 9^{th} through 48^{th} letters) depends on which of the 120 options are picked to complete the ordering of the Cube. The next aspect of the ordering is the ordering of the nine individual cubies, and the placement of the remaining 8 letters after the number for orientation has been placed. After that, the orientation of the number on the square will change the orientation of the remaining letters distributed on the cube face. Figure 3.1 demonstrates this.

The general solution to transposition algorithms according to Kahn is indeed multiple anagramming. This technique encorporates utilizing multiple ciphertext blocks of the same length, enciphered by the same key. Multiple anagramming can be utilized by placing both messages one on top of the other on paper, then the strips are cut and placed side by side in a new order until both messages show plaintext. This technique relies on a one-to-one mapping from the location in the original plaintext to the ciphertext. The system proposed does not have this feature. Mitchell notes that his system allows for a plaintext encryption using any of the 120 orderings of the faces of the Cube, with the orientation being one of four possibilities. Both messages would have to use the same cube face ordering and the same set of Cube face orientation for multiple anagramming to even be successful. Mitchell states that "multiple anagramming can be thwarted by simply varying the plaintext initialization from one block of 48 letters to the next" [6]. These changes are "self-keyed" so no communication of the initialization is required, and the rotation key may be held constant. By keeping a record of all initializations and limiting the use of similar initialization, one can maintain a decent level of security.

The rotation sequence performed must have the "1" in the top left hand corner of the face selected, knowing that the color does not matter for the first cube face selected. This will serve as the reference point for the rotation sequence. Due to the point mentioned above, we can count that there are a total of "$120 \times 9^5 \times 4^5 > 7.25$ billion different ways of initializing a given plaintext" [6]. A single rotation key has more than 7.25 billion different encryptions due to the ways the plaintext can be initialized. Successful attacks cannot occur because of a lack of frequently repeated plaintext passages. Multiple anagramming as an attack would also be unreliable because this attack relies on the repeated use of the same ordering of the transposition. Different initializations of the plaintext correspond to various transpositions, and this in turn thwarts multiple anagramming.

Mitchell proposes an additional technique to increase the possibilities of the initial plaintext created: "the use of arbitrary numbers of nulls [or blank spaces]" [6]. Mitchell describes how much change occurs when just a single null is substituted into one of the 9^{th} through 48^{th} spaces of the Cube: this "will change the column of every plaintext letter, change the row of about one third of them, and change the cube face of one eighth of them" [6]. Apparently, the changes will appear to be random with the addition of a single null. With even more nulls, this could multiply the number of permutations of the keys created. The locations of these nulls could also vary by key.

Rotation keys are used to create the generated “mixing.” In essence, this scrambles the Cube in such a way that the letters and numbers are mixed up even more and create a “new” state of the Cube. Holding the number “1” in place (towards the user) three categories of rotations, by row, by column, and by level, can be twisted to count as a move in the mixing up of the Cube. The outside one or two rows, columns, or levels are able to be moved by $\pi/2$, π, or $3\pi/2$ (2π results in the identity, as if the cube was never moved at all). The row, column, and level rotations follow Figure 3.2.

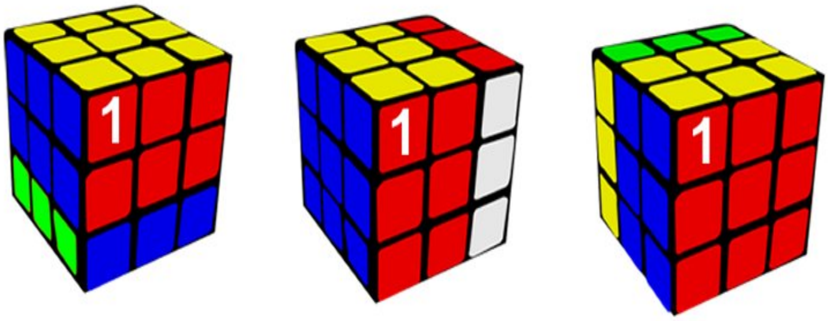


Figure 3.2: Row, Column, Level

The number of possible rotations can be counted by which column, row, or level is turned. Since one can twist one or two of these layers, and there are three sizes of rotations, and three axes on which one can twist, the total number of different twists is $2 \times 3 \times 3 = 18$ possibilities of moves.

When making a rotation key, the axis upon which the twist is performed must change at every iteration in order to provide a key that is unique. For example, a row twist by $\pi/2$ and then a second row twist by π will be equivalent to a single row twist of $3\pi/2$. After the first twist, then, only 15 options of twists remain. Mitchell explains that the creation of a rotation key can take less than five minutes with thirty, ten-second twists.

3.3 Deciphering Mitchell’s Cryptosystem

Using the ciphertext, one can read the letters off the cube in a prearranged order, starting with the initial “1” on the Cube. To provide some extra “randomness”, the encryption key can have information about how to read off the squares at the beginning of the ciphertext. According to a fixed system, one could orient the Cube in such a way, all the while denoting each face as a particular letter (with A = top face, B = bottom face, etc), that a *key extension*, or set of extra letters placed at the beginning of the exchanged rotation key, would hint at the order of reading off the ciphertext. Mitchell describes a six-letter extension such as “AFDEBC” which would tell the receiving party to read off the text on the Cube top face first, then the face denoted ‘F’, etc until it was copied down onto paper.

Additionally, Mitchell proposed having the order of letters being read to be signaled by an additional letter in the key extension: “suppose X stands for reading left-to-right, top row first, middle row second, bottom row last, while Y stands for reading top-to-bottom, left column first, then center column, then right column, and so forth” [6]. After this is created, the decipherment is simple. The decoder would copy the ciphertext onto a cube in the order of the key extension. The key for the rotation would be read backwards, which would involve substitution of the right for left and down for up. The plaintext can be read off of the face labeled “1”, then ”2”, etc. Mitchell points out that the orientation of the faces after the first will be one of four possibilities, the correct one reading off intelligible plaintext. For the rotation key, we can use the following standard:

R = Row, C = Column, and L = Level.

1, 2, 3 will represent the clockwise rotation of a face by a muliple of $\pi/2$.

4, 5, 6 will represent the outside two layers being twisted a multiple of $\pi/2$. So with the above standards, a possible rotation key, with an extension would look like the following:

ADCEBFX-R3-L2-C4-L6-R5-C1-R3-L4-R1

From this rotation key, the sender would twist the cube like such: the bottom outer row $3\pi/2$ radians clockwise, π radians for the outer level, a $\pi/2$ twist for the outer right two columns, $3\pi/2$ radian twist for the outside two levels, etc until the process was complete. At this point, the decoder would read off the ciphertext in the order of faces, “ADCEBF” and follow the top-to-bottom, left-to-right pattern and write these down in that order. This method, however, does not necessarily address the orientation of the face when recording the resulting ciphertext.

Mitchell claims that “if the technique is used properly-i.e. with different message blocks enciphered using different random drawings from these billions of possibilities-attacks based on frequently-occurring phrases should be unsuccessful” [6]. In attempts to try all the possible 48 letter chunks, there would be 48! transpositions to check. This equates to a huge 1.2×10^{61} possibilities. Using brute force, one would have to check through $18 \times 15^{(k-1)}$ keys, where k is the number of moves per rotation sequence. Having the addition of the key extension would multiply the number of options by 720 (6!). If $k = 30$, as in the example suggested by Mitchell, we see there is about 2.3×10^{35} different keys. The actual length of k is unknown, and thus a large k value would make the search even that much more difficult.

Although this may be the case, Mitchell did not take Kerckhoffs’ Principle into account. With knowledge of the Cube’s properties, and the fact that the Cube is responsible for the algorithms, not every permutation of the 48 characters is possible using rotations of the Cube. Also, since there are only 26 letters of the alphabet, there are bound to be repeated characters in the ciphertext, which further decreases the practical size of the keyspace. Computers have calculated that every state can be reached in fewer than 20 moves: “The median optimal solution length appears to be 18 moves” [5].

When two types of systems are combined, there is additional strength and security present. According to Mitchell, "Pre-ciphering the plaintext with a simple (and thus fast) substitution cipher should make cryptanalysis immeasurably harder, since for instance a brute force computer attack could no longer be designed to terminate when words from a computer's dictionary are encountered" [6]. Adding a substitution cipher is problematic because a simple substituition can be stripped using basic frequency analysis. In order to avoid using a second layer which can easily be stripped, it might be helpful to consider a cipher that will strengthen the system rather than provide no additional security.

Mitchell's system seems to have some merit. However, if the technique was known and anything was discovered around the actual system, such as the key with its rotations, further efforts in attempting to secure the messages would be necessary. If the message is discovered, or somehow multiple anagramming proves to work, the cryptanalysis would prove effective and the interception would ruin all hopes of secrecy.

Chapter 4

The Trans-Composite Cryptosystem Revealed

With several cryptosystems and ciphers at our fingertips, there is potential for the creation of a new system. A lesson that Johnathan Katz, a cryptography author, warns us is that "Designing secure ciphers is a hard task." Complexity does not necessarily imply security. According to Katz, "it is very hard to design a secure encryption scheme, and such design should be left to experts." Most cryptographers find that "No one these days uses the Vigenère for secure communications" [2]. Alone, this system does have weakness, but can it be applied to another system, like Mitchell advises, in order to strengthen the cryptosystem? This sounds like a challenge worth pursuing.

4.1 Vigenère Finds a Friend

To compensate for the weaknesses of both the Vigenère and the Mitchell Cryptosystem, we must attempt to find a new cryptosystem that will prevent the attacks of multiple anagramming and frequency analysis. Often when an algorithm for encoding is repeated, there is no additional security. Repeating an encryption step often produces a ciphertext equivalent to a single encryption with a different key.

Combining two different algorithms, the Mitchell and the Vigenère, we have a more complex algorithm with the weaknesses balanced out. The proposed system will incorporate both ciphers to create a system that is more difficult to attack.

The first step of the process is to employ the Vigenère cipher. A keyword must be selected to change the plaintext to a preliminary ciphertext. For our purposes, this is the first layer of the algorithm. Figure 4.1 is an example with the keyword "RUBIKS".

Encoding a plaintext message will require a two-step process to create the ciphertext message. For the second step, the text will undergo the algorithm of the Mitchell system,

I	F	M	A	T	H	E	M	A	T	I	C	S	I	S	T	H	E	W	A	Y...
R	U	B	I	K	S	R	U	B	I	K	S	R	U	B	I	K	S	R	U	B...
A	A	O	J	E	A	W	H	C	C	T	V	K	D	U	C	S	X	O	V	A...

Table 4.1: First Step of the Combined System

but instead of a readable English text placed on the cubies, the individual encoding would place the seemingly-unreadable letters onto the cube and then proceed with the algorithm.

With the ciphertext on the Cube, the encoding party must begin by writing down the characters from the Cube, top to bottom, and left to right. For this particular system, instead of following the Mitchell key extension, we have decided to utilize a fixed pattern of reading the ciphertext letters prior to writing them down. A specific example of this would be the sender selecting a face, writing down the nine characters, rotating the Cube to the right for the next nine characters, then down for the next nine characters following a right-down-right-down rotation until all are copied. The ciphertext written on paper will include both letters and the six numbers placed on the Cube in the initialization process.

An example would look like the following:
Keyword: "RUBIKS"
Plaintext: IFMATHEMATICSISTHEWAYOFLIFE...
Vigenère Layer Ciphertext: AAOJEAWHCCTVKDUCSXOVAXQEAAG...
Mitchell Rotation Key: R3-L2-C4-L6-R5-C1-R3-L4-R1
Final Ciphertext: 5VCXJOQXGH6AZVAE2WLEAKAACJDN...

4.2 Decoding With the Ciphertext

The information that must be transferred between the encoder and decoder is the keyword for the Vigenère step and the rotation key. With these pieces, the individual decoding the ciphertext may proceed through the algorithm and reach the original plaintext message.

The first step for the decoder will be to select an arbitrary face and copy the ciphertext onto the Cube following the right-down-right-down pattern. The next step is to hold the Cube with the '1' in the upper left hand corner oriented properly, and following the reverse sequence of the rotation key. This requires maintaining the clockwise rotation of all the rows, columns, and levels. Additionally, we see that the amount of rotation per move of every Cube face will "add up" to 2π. For example, if the last part of the rotation key was **L4**, the first move of the reverse rotation key would be **L6** because $\pi/2 + 3\pi/2 = 2\pi$, or a complete rotation.

After we obtain the Vigenère layer on the Cube, every 8-letter partition of the 48-character block must be oriented correctly. With the original Mitchell system, the orientation of the Cube face was determined by the intelligible plaintext which could be read. However, this layer will not be readable by the decoder, and thus all four orientations must pass through the Vigenère step of the algorithm. With the length of the keyword known, and the order of the faces still following the 1 through 6 progression, the Vigenère step may be utilized for each of the orientations until intelligible plaintext results.

4.3 Addressing the Weaknesses

As isolated systems, the Vigenère system and Mitchell system still have weaknesses. The Vigenère cipher system can be broken with frequency analysis, as mentioned earlier. Frequency analysis relies on the fact that corresponding shifts are in known positions. The Mitchell system added to this does not allow for the shifting of characters in the same positions due to the transposition element. Although the Mitchell system is safe against multiple anagramming, the fact remains that all the plaintext letters are present in the ciphertext. With the knowledge that the 54-character blocks contain 48 plaintext letters, the Mitchell system alone can potentially be exploited through a genetic algorithm implemented by a fast computer. An algorithm like this rearranges the letters of the ciphertext looking for a word and checking whether or not the remaining letters can be rearranged to form intelligible plaintext. With the extra layer, however, the genetic algorithm would be unsuccessful since the original plaintext letters are no longer present.

For the combined system proposed, the greatest weakness based on the research appears to be efficiency. Without a quick means to encode and decode, this presents a problem for the communicating parties. On the other hand, using the system with merely a Rubik's Cube, a marker, and a piece of paper is a desirable advantage.

4.4 Conclusion to the Crypto-Cube

Cryptanalysis of the proposed system is an interesting prospect. Working on the side of the implementation of the system itself, rather than the cryptanalytic perspective creates bias. Therefore in going forward, we must ask several questions to probe the possibilities of breaking the system. Charles Babbage said in 1864, "Deciphering is an affair of time, ingenuity and patience" [1]. He was right. The amount of creativity and mental fortitude necessary to figure out some sort of system takes effort, skill, and plenty of hours.

Pure cryptanalysis, in which a computer is implemented as a means of deciphering the cryptotext, may mathematically prove to succeed, but this requires further investigation. In the meantime, what can be done to prevent successful cryptanalysis? Bauer suggests that "[T]he most important weapon seems to be imagination" [1]. Does the Rubik's Cube combined with the Vigenère algorithm hold enough secrecy in itself? What weaknesses does this three-dimensional puzzle possess that prevents it from being a legitimate tool for cryptography? Further research may provide more insight.

Bibliography

[1] Bauer, Friedrich L. (2007). *Decrypted secrets: Methods and maxims of cryptology*. New York: Springer Science+Business Media.

[2] Hoffstein, Jeffrey, Jill Pipher, and Joseph H. Silverman. (2008). *An introduction to mathematical cryptography*. New York: Springer Science+Business Media.

[3] Hofstadter, Douglas R. (1996). *Metamagical themas: The magic cube's cubies are twiddled by cubists and solved by cubemeisters*. Scientific American.

[4] Katz, Jonathan and Yehuda Lindell. (2008). *Introduction to modern cryptography*. New York: Chapman & Hall.

[5] Korf, Richard E. (1997). *Finding optimal solutions to rubik's cube using pattern databases*. Los Angeles: University of California Press.

[6] Mitchell, Douglas W. (1992). *"Rubik's cube": As a transposition device*. Cryptologia. Volume 16, Number 3. Morgantown: Taylor & Francis.

[7] Trappe, Wade and Lawrence C. Washington. (2006). *Introduction to cryptography with coding theory second edition*. New Jersey: Prentice Hall.