BOOK CIPHER, RUNNING KEY CIPHER, VIC CIPHER AND SECOM CIPHER

A **book cipher** is a cipher in which the key is some aspect of a book or other piece of text; books being common and widely available in modern times, users of book ciphers take the position that the details of the key is sufficiently well hidden from attackers in practice. This is in some ways an example of security by obscurity. It is typically essential that both correspondents not only have the same book, but the same edition.[1]

Traditionally book ciphers work by replacing words in the plaintext of a message with the location of words from the book being used. In this mode, book ciphers are more properly called codes.

This can have problems; if a word appears in the plaintext but not in the book, it cannot be encoded. An alternative approach which gets around this problem is to replace individual letters rather than words. One such method, used in the second Beale cipher, substitutes the first letter of a word in the book with that word's position. In this case, the book cipher is properly a cipher — specifically, a homophonic substitution cipher. However, if used often, this technique has the side effect of creating a larger ciphertext (typically 4 to 6 digits being required to encipher each letter or syllable) and increases the time and effort required to decode the message.

Choosing the Key []

The main strength of a book cipher is the key. The sender and receiver of encoded messages can agree to use any book or other publication available to both of them as the key to their cipher. Someone intercepting the message and attempting to decode it, unless they are a skilled cryptographer (see Security below), must somehow identify the key from a huge number of possibilities available. In the context of espionage, a book cipher has a considerable advantage for an agent in enemy territory. A conventional codebook, if discovered by the local authorities, instantly incriminates the holder as a spy and gives the authorities the chance of deciphering the code and sending false messages impersonating the agent. On the other hand a book, if chosen carefully to fit with the spy's cover story, would seem entirely innocuous. The drawback to a book cipher is that both parties have to possess an identical copy of the key. The book must not be of the sort that would look out of place in the possession of those using it and it must be of a type likely to contain any words required. Thus, for example, a spy wishing to send information about troop movements and numbers of armaments would be unlikely to find a cookery book or a romantic novel useful keys.

Using Widely Available Publications []

Dictionary []

Another approach is to use a dictionary as the codebook. This guarantees that nearly all words will be found, and also makes it much easier to find a word when encoding. This approach was used by George Scovell for the Duke of Wellington's army in some campaigns of the Peninsular War. In Scovell's method, a codeword would consist of a number (indicating the page of the dictionary), a letter (indicating the column on the page), and finally a number indicating which entry of the column was meant. However, this approach also has a disadvantage: because entries are arranged in alphabetical order, so are the code numbers. This can give strong hints to the cryptanalyst unless the message is superenciphered. The wide distribution and availability of dictionaries also present a problem; it is likely that anyone trying to break such a code is also in possession of the dictionary which can be used to read the message.

Bible Cipher []

The Bible is a widely available book that is almost always printed with chapter and verse markings making it easy to find a specific string of text within it, making it particularly useful for this purpose.

Security []

Essentially, the code version of a "book cipher" is just like any other code, but one in which the trouble of preparing and distributing the codebook has been eliminated by using an existing text. However this means, as well as being attacked by all the usual means employed against other codes or ciphers, partial solutions may help the cryptanalyst to guess other codewords, or even to completely break the code by identifying the key text. This is, however, not the only way a book cipher may be broken. It is still susceptible to other methods of cryptanalysis, and as such is quite easily broken, even without sophisticated means, without the cryptanalist having any idea what book the cipher is keyed to.[2][*page needed*]

If used carefully, the cipher version is probably much stronger, because it acts as a homophonic cipher with an extremely large number of equivalents. However, this is at the cost of a very large ciphertext expansion.

Examples []

- A famous use of a book cipher is in the Beale ciphers, of which document no. 2 uses a (variant printing of) the United States Declaration of Independence as the key text.
- Richard Sorge's spy ring in Japan used a book cipher which the Japanese were unable to cryptanalyze even after capturing both Sorge and his radio operator / code clerk. It used an ion of a statistical handbook of Germany as the key text.
- In the American Revolution, Benedict Arnold used a book cipher, sometimes known as the Arnold Cipher, which used Sir William Blackstone's *Commentaries on the Laws of England* as a key text.

In classical cryptography, the **running key cipher** is a type of polyalphabetic substitution cipher in which a text, typically from a book, is used to provide a very long keystream. Usually, the book to be used would be agreed ahead of time, while the passage to use would be chosen randomly for each message and secretly indicated somewhere in the message.

Example []

Suppose we have agreed to use *The C Programming Language* (1978 edition) as our text, and we are using the *tabula recta* as our tableau. We need to send the message 'Flee at once'.

First, we choose a starting point. Let us choose page 63, line 1:

> errors can occur in several places. A label has...

We write out the running key under our plaintext:

Plaintext: f l e e a t o n c e
Running key: E R R O R S C A N O
Ciphertext: J C V S R L Q N P S

And send the message 'JCVSR LQNPS'. However, unlike a Vigenère cipher, if we have to extend our message, we don't repeat the key; we just continue on from the key text. So suppose we need a longer message, like: 'Flee at once. We are discovered'. Then we just continue as before:

Plaintext: f l e e a t o n c e w e a r e d i s c o v e r e d
Running key: E R R O R S C A N O C C U R I N S E V E R A L P L
Ciphertext: J C V S R L Q N P S Y G U I M Q A W X S M E C T O

Next we need to tell the recipient where to find the running key for this message. In this case, we've decided to make up a fake block of five ciphertext characters, with three denoting the page number, and two the line number, using A=0, B=1 etc. to encode digits. Such a block is called an **indicator block**. The indicator block will be inserted as the second last of each message. (Of course, many other schemes are possible for hiding indicator blocks). Thus page 63, line 1 encodes as 'AGDAB' (06301).

Finally we can send the message 'JCVSR LQNPS YGUIM QAWXS AGDAB MECTO'.

Variants []

Modern variants of the running key cipher often replace the traditional *tabula recta* with bitwise exclusive or, operate on whole bytes rather than alphabetic letters, and

derive their running keys from large files. Apart from possibly greater entropy density of the files, and the ease of automation, there is little practical difference between such variants and traditional methods.

Permutation generated running keys []

A more compact running key can be used if one combinatorially generates text using several start pointers (or combination rules). For example, rather than start at one place (a single pointer), one could use several start pointers and xor together the streams to form a new running key, similarly skip rules can be used. What is exchanged then is a series of pointers to the running key book and/or a series of rules for generating the new permuted running key from the initial key text. (These may be exchanged via public key encryption or in person. They may also be changed frequently without changing the running key book).

Cyphertext appearing to be plaintext []

Traditional cyphertext appears to be quite different than plaintext. To address this problem, one variant outputs "plaintext" words instead of "plaintext" letters as the cyphertext output. This is done by creating an "alphabet" of words (in practice multiple words can correspond to each cypher-text output character). The result is a cyphertext output which looks like a long sequence of plaintext words (the process can be nested). Theoretically, this is no different than using standard cyphertext characters as output. However, plaintext-looking cyphertext may result in a "human in the loop" to try to mistakenly interpret it as decoded plaintext.

An example would be BDA (Berkhoff deflater algorithm), each cyphertext output character has at least one noun, verb, adjective and adverb associated with it. (E.g. (at least) one of each for every ASCII character). Grammatically plausible sentences are generated as cyphertext output. Decryption requires mapping the words back to ASCII, and then decrypting the characters to the real plaintext using the running key. Nested-BDA will run the output through the reencryption process several times, producing several layers of "plaintext-looking" cyphertext - each one potentially requiring "human-in-the-loop" to try to interpret its non-existent semantic meaning.

Security []

If the running key is truly random, never reused, and kept secret, the result is a one-time pad, a method that provides perfect secrecy (reveals no information about the plaintext). However, if (as usual) the running key is a block of text in a natural language, security actually becomes fairly poor, since that text will have non-random characteristics which can be used to aid cryptanalysis. As a result, the entropy per character of both plaintext and running key is low, and the combining operation is easily inverted.

To attack the cipher, a cryptanalyst runs guessed probable plaintexts along the ciphertext, subtracting them out from each possible position. When the result is a chunk

of something intelligible, there is a high probability that the guessed plain text is correct for that position (as either actual plaintext, or part of the running key). The 'chunk of something intelligible' can then often be extended at either end, thus providing even more probable plaintext - which can in turn be extended, and so on. Eventually it is likely that the source of the running key will be identified, and the jig is up.

There are several ways to improve the security. The first and most obvious is to use a secret mixed alphabet tableau instead of a *tabula recta*. This does indeed greatly complicate matters but it is not a complete solution. Pairs of plaintext and running key characters are far more likely to be high frequency pairs such as 'EE' rather than, say, 'QQ'. The skew this causes to the output frequency distribution is smeared by the fact that it is quite possible that 'EE' and 'QQ' map to the same ciphertext character, but nevertheless the distribution is not flat. This may enable the cryptanalyst to deduce part of the tableau, then proceed as before (but with gaps where there are sections missing from the reconstructed tableau).

Another possibility is to use a key text that has more entropy per character than typical English. For this purpose, the KGB advised agents to use documents like almanacs and trade reports, which often contain long lists of random-looking numbers.

Another problem is that the keyspace is surprisingly small. Suppose that there are 100 million key texts that might plausibly be used, and that on average each has 11 thousand possible starting positions. To an opponent with a massive collection of possible key texts, this leaves possible a brute force search of the order of 2^{40}, which by computer cryptography standards is a relatively easy target. (See permutation generated running keys above for an approach to this problem).

Confusion []

Because both ciphers classically employed novels as part of their key material, many sources confuse the book cipher and the running key cipher. They are really only very distantly related. The running key cipher is a polyalphabetic substitution, the book cipher is a homophonic substitution. Perhaps the distinction is most clearly made by the fact that a running cipher would work best of all with a book of random numbers, whereas such a book (containing no text) would be useless for a book cipher.

VIC cipher

The **VIC cipher** was a pencil and paper cipher used by the Soviet spy Reino Häyhänen, codenamed "VICTOR".

If the cipher were to be given a modern technical name, it would be known as a "straddling bipartite monoalphabetic substitution superenciphered by modified double transposition." However, by general classification it is part of the Nihilist family of ciphers.

It was arguably the most complex hand-operated cipher ever seen, when it was first discovered. The initial analysis done by the NSA in 1953 did not absolutely conclude that it was a hand cipher, but its placement in a hollowed out 5c coin implied it could be broken by pencil and paper. The VIC cipher remained unbroken until more information about its structure was available.

Although certainly not as complex or secure as modern computer operated stream ciphers or block ciphers, in practice messages protected by it resisted all attempts at cryptanalysis by at least NSA from its discovery in 1953 until Häyhänen's defection in 1957.

A revolutionary leap []

The VIC cipher can be regarded as the evolutionary pinnacle of the Nihilist cipher family.

The VIC cipher has several important integrated components, including mod 10 chain addition, a lagged Fibonacci generator (a recursive formula used to generate a sequence of pseudorandom digits), a straddling checkerboard, and a disrupted double transposition.

Until the discovery of VIC, it was generally thought that a double transposition alone was the most complex cipher an agent, as a practical matter, could use as a field cipher.

History []

During World War II, several Soviet spy rings communicated to Moscow Centre using two ciphers which are essentially evolutionary improvements on the basic Nihilist cipher. A very strong version was used by Max Clausen in Richard Sorge's network in Japan, and by Alexander Foote in the Lucy spy ring in Switzerland. A slightly weaker version was used by the *Rote Kapelle* network.

In both versions, the plaintext was first converted to digits by use of a straddling checkerboard rather than a Polybius square. This has the advantage of slightly compressing the plaintext, thus raising its unicity distance and also allowing radio operators to complete their transmissions quicker and shut down sooner. Shutting down sooner reduces the risk of the operator being found by enemy radio direction finders. Increasing the unicity distance increases strength against statistical attacks.

Clausen and Foote both wrote their plaintext in English, and memorized the 8 most frequent letters of English (to fill the top row of the checkerboard) through the mnemonic (and slightly menacing) phrase "a sin to err" (dropping the second "r").[1] The

standard English straddling checkerboard has 28 character slots and in this cipher the extra two became "full stop" and "numbers shift". Numbers were sent by a numbers shift, followed by the actual plaintext digits in repeated pairs, followed by another shift. Then, similarly to the basic Nihilist, a digital additive was added in, which was called "closing". However a different additive was used each time, so finally a concealed "indicator group" had to be inserted to indicate what additive was used.

Unlike basic Nihilist, the additive was added by non-carrying addition (digit-wise addition modulo 10), thus producing a more uniform output which doesn't leak as much information. More importantly, the additive was generated not through a keyword, but by selecting lines at random from almanacs of industrial statistics. Such books were deemed dull enough to not arouse suspicion if an agent was searched (particularly as the agents' cover stories were as businessmen), and to have such high entropy density as to provide a very secure additive. Of course the figures from such a book are not actually uniformly distributed (there is an excess of "0" and "1" (see Benford's Law), and sequential numbers are likely to be somewhat similar), but nevertheless they have much higher entropy density than passphrases and the like; at any rate, in practice they seem never to have been successfully cryptanalysed.

The weaker version generated the additive from the text of a novel or similar book (at least one *Rote Kapelle* member actually used *The Good Soldier Schweik*, which may not have been a good choice if one expected to be searched by Nazis!) This text was converted to a digital additive using a technique similar to a straddling checkerboard.

The ultimate development along these lines was the VIC cipher, used in the 1950s by Reino Häyhänen. By this time, most Soviet agents were instead using one-time pads. However, despite the theoretical perfection of the one-time pad, in practice they *were* broken, while VIC was not.

Internal mechanics []

Straddling checkerboard []

A straddling checkerboard is a device for converting an alphabetic plaintext into digits whilst simultaneously achieving fractionation (a simple form of information diffusion) and data compression relative to other schemes using digits. It also is known as a monôme-binôme cipher.

A straddling checkerboard is set up something like this:

	0	1	2	3	4	5	6	7	8	9
	E	T		A	O	N		R	I	S
2	B	C	D	F	G	H	J	K	L	M

6	P	Q	/	U	V	W	X	Y	Z	.

The first row is populated with the ten digits, 0-9. They can be presented in order, as in the above table, or scrambled for additional security. The second row is typically set up with high-frequency letters (mnemonic ESTONIA-R), leaving two blank spots. It has no row label. The remaining rows are labeled with each digit that was not assigned a letter in the second row, and then filled out with the rest of the alphabet.

Much like the ordering of the digits in the top row, the alphabet can be presented in order (as it is here), or scrambled with a keyword or other technique. Since there are 30 slots in our grid, and we skipped two letters in the first row, there will be two spare cells in the other rows. We have filled these cells with a period '.', and a slash '/' to be used as a numeric escape character (indicating that a numeral follows). It doesn't matter where these spares go, so long as the sender and receiver use the same system.

To encipher, a letter in the second row is simply replaced by the number labeling its column. Letters in the third and fourth rows are replaced by a two-digit number representing their row and column numbers. Mapping one-digit numbers to common letters reduces the length of the ciphertext, while also concealing the identities of the two-digit numbers by reducing the frequency of their first digits. Here is an example:[2]

A T T A C K A T D A W N

3 1 1 3 21 27 3 1 22 3 65 5

The resulting message, 3113212731223655, may be sent directly (if the table is scrambled), but is usually processed through a second cipher stage, such as transposition or substitution. As a simple example, we will add a secret key number (say, 0452) using modular (non-carrying) arithmetic:

3 1 1 3 2 1 2 7 3 1 2 2 3 6 5 5

\+ 0 4 5 2 0 4 5 2 0 4 5 2 0 4 5 2

= 3 5 6 5 2 5 7 9 3 5 7 4 3 0 0 7

Optionally, we could then use the same straddling checkerboard to convert the ciphertext back into letters:

3 5 65 25 7 9 3 5 7 4 3 0 0 7

A N W H R S A N R O A E E R

Deciphering is simply the reverse of these processes. Although the size of groups can vary, deciphering is unambiguous because whenever the next element to be deciphered starts with a 2 or a 6, it is a pair; otherwise, it is a singleton.

Disrupted transposition []

In a disrupted transposition, certain positions in a grid are blanked out, and not used when filling in the plaintext. This breaks up regular patterns and makes the cryptanalyst's job more difficult.

Fractionation []

Transposition is particularly effective when employed with fractionation - that is, a preliminary stage that divides each plaintext symbol into several ciphertext symbols. For example, the plaintext alphabet could be written out in a grid, then every letter in the message replaced by its co-ordinates (see Polybius square). Another method of fractionation is to simply convert the message to Morse code, with a symbol for spaces as well as dots and dashes.

When such a fractionated message is transposed, the components of individual letters become widely separated in the message, thus achieving Claude E. Shannon's diffusion. Examples of ciphers that combine fractionation and transposition include the bifid cipher, the trifid cipher, the ADFGVX cipher and the VIC cipher.

Another choice would be to replace each letter with its binary representation, transpose that, and then convert the new binary string into the corresponding ASCII characters. Looping the scrambling process on the binary string multiple times before changing it into ASCII characters would likely make it harder to break. Many modern block ciphers use more complex forms of transposition related to this simple idea.

The VIC Cipher

The VIC cipher is an intricate cipher issued by the Soviet Union to at least one of its spies. It is of interest because it seems highly secure, despite being a pencil-and-paper cipher. It was the cipher in which a message was written which was found on a piece of microfilm inside a hollowed-out nickel by a newspaper boy in 1953. The workings of this cipher were explained by Hayhaynen to FBI agents shortly after his defection to the United States in 1957.

David Kahn described that cipher briefly in an article in Scientific American, and in full detail in a talk at the 1960 annual convention of the American Cryptogram Association which was later reprinted in his book *Kahn on Codes*.

The VIC cipher, which I will demonstrate here adapted to the sending of English-language messages, begins with an involved procedure to produce ten pseudorandom digits. The agent must have memorized six digits (which were in the form of a date), and the first 20 letters of a key phrase (which was the beginning of a popular song) and must think of five random digits for use as a message indicator.

Let the date be July 4, 1776, to give the digits 741776. (Actually, the Russians used their customary form of dates, with the month second.) And let the random indicator group be 77651.

The first step is to perform digit by digit subtraction (without carries) of the first five digits of the date from the indicator group:

```
  77651
(-) 74177
 ---------
  03584
```

The second step is to take the 20-letter keyphrase, and turn it into 20 digits by dividing it into two halves, and within each half, assigning 1 to the letter earliest in the alphabet, and so on, treating 0 as the last number, and assigning digits in order to identical letters. Thus, if our keyphrase is "I dream of Jeannie with t", that step proceeds:

```
I D R E A M O F J E  A N N I E W I T H T
6 2 0 3 1 8 9 5 7 4  1 6 7 4 2 0 5 8 3 9
```

The result of the first step is then expanded to ten digits through a process called *chain addition*. This is a decimal analog of the way a linear-feedback shift register works: starting with a group of a certain number of digits (in this case five, and later we will do the same thing with a group of ten digits), add the first two digits in the group together, take only the last digit of the result and append it to the end of the group, then ignore the first digit, and repeat the process.

The 10 digit result is then added, digit by digit, ignoring carries, to the first 10 digits produced from the keyphrase to produce a ten-digit result, as follows:

```
  6 2 0 3 1 8 9 5 7 4
(+) 0 3 5 8 4 3 8 3 2 7
 ----------------------
  6 5 5 1 5 1 7 8 9 1
```

And these 10 digits are then encoded by encoding 1 as the first of the 10 digits produced from the second half of the keyphrase, 2 as the second, up to 0 as the tenth.

```
using code: 1 2 3 4 5 6 7 8 9 0
        1 6 7 4 2 0 5 8 3 9

        6 5 5 1 5 1 7 8 9 1
becomes 0 2 2 1 2 1 5 8 3 1
```

This ten digit number is used by chain addition to generate 50 pseudorandom digits for use in encipherment:

```
 0 2 2 1 2 1 5 8 3 1
--------------------
 2 4 3 3 3 6 3 1 4 3
 6 7 6 6 9 9 4 5 7 9
 3 3 2 5 8 3 9 2 6 2
 6 5 7 3 1 2 1 8 8 8
 1 2 0 4 3 3 9 6 6 9
```

The last row of these digits (which will still be used again) is used like the letters in a keyword for transposition to produce a permutation of the digits 1 through 9 (with 0 last again):

```
 1 2 0 4 3 3 9 6 6 9
--------------------
 1 2 0 5 3 4 8 6 7 9
```

and those digits are used as the top row of numbers for a straddling checkerboard:

```
 1 2 0 5 3 4 8 6 7 9
 -------------------
 A T  O N E  S I R
 -------------------
0 B C D F G H J K L M
8 P Q U V W X Y Z . /
```

One detail omitted is that the checkerboard actually used had the letters in the bottom part written in vertical columns with some columns left until the end. That doesn't work as well in an English example, as there are only two left-over spaces after the alphabet.

With the straddling checkerboard in place, we can begin enciphering a message.

Let our message be:

We are pleased to hear of your success in establishing your false identity. You will be sent some money to cover expenses within a month.

Converting this to numbers, we proceed:

W EAREP L EASED TOH EAROF Y OU RSU C C ESSINESTAB L ISH ING
8341948107416400250441950588580968002024667346210107760473 03

Y OU RF AL SEID ENTITY Y OU W IL L B ESENTSOM EM ONEY TOC O
88580905107647004327288885808370707014643265094095348825025

V EREX P ENSESW ITH INAM ONTH
85494848143646837204731095320 4

For the sake of our example, we will give our agent a small *personal number* of 8. This number is used to work out the widths of the two transposition tableaux used to transpose the numbers obtained above. The last two unequal digits, which in this case are the last two digits (6 and 9) of the last row of the 50 numbers generated above, are added to the personal number with the result that the two transpositions will involve 8+6, or 14, and 8+9, or 17, columns respectively.

The keys for those two transpositions are taken by reading out the 50 numbers by columns, using the 10 digits used to generate them as a transposition key. Again, 0 is last, so given the table above:

```
 0 2 2 1 2 1 5 8 3 1
 --------------------
 2 4 3 3 3 6 3 1 4 3
 6 7 6 6 9 9 4 5 7 9
 3 3 2 5 8 3 9 2 6 2
 6 5 7 3 1 2 1 8 8 8
 1 2 0 4 3 3 9 6 6 9
```

we read out the digits in order:

36534 69323 39289 47352 36270 39813 4

stopping when we have the 31 digits we need.

Our first transposition uses the first 14 digits as the key of a conventional simple columnar transposition:

```
 36534693233928
 --------------
 83419481074164
 00250441950588
 58096800202466
 73462101077604
 73038858090510
 76470043272888
 85808370707014
 64326509409534
 88250258549484
 81436468372047
 3109532049
```

Since our message consisted of ten rows of 14 digits, plus one extra row of 9 digits, it is 149 digits long. At this initial stage, one null digit is appended to the message, making it 150 digits long, so that it will fill a whole number of 5-digit groups.

Thus, with the null digit added, it gives us the intermediate form of the message:

```
09200274534 6860181384 80577786883 15963702539 11018309880
75079700479 4027027992 90628086065 42040483240 30833654811
44818035243 4864084447 84005470562 1546580540
```

The fact that our message is 150 digits long was important to note, since the next step in the encipherment, although it is also a columnar transposition, includes an extra complexity to make the transposition irregular, and so it is necessary to lay out in advance the space that will be used in that transposition.

The remaining 17 digits of the 31 we read out above, 9 47352 36270 39813 4, are the key for this second transposition. The numbers, in addition to indicating the order in which the columns are to be read out, indicate where triangular areas start which will be filled in last.

The first triangular area starts at the top of the column which will be read out first, and extends to the end of the first row. It continues in the next row, starting one column later, and so on until it includes only the digit in the last column. Then, after one space, the second triangular area starts, this time in the column which will be read out second.

Since we know that our message is 150 digits long, we know that it will fill 8 rows of 17 digits, with 14 digits in the final row. This lets us fill in the transposition block, first avoiding the triangular areas:

```
 94735236270398134
 -----------------
 09200274534686
 018138480577786
 8831596370253911
 01830988075079700
 47940
 270279
 9290628
 08606542
 040483240
```

and then with them filled in as well:

```
 94735236270398134
 -----------------
 09200274534686308
 01813848057778633
 88315963702539116
 01830988075079700
 47940548114481803
 27027952434864084
 92906284478400547
 08606542056215465
 04048324080540
```

from which the fully encrypted message can be read out:

36178054 289959253 507014400 011342004 746845842 675048425
03100846 918177284 83603475 035007668 483882424 283890960
350713758 689914050 008042900 873786014 472544860

The last digit, 6, in the date shows that the indicator group is to be inserted in the final message as the sixth group from the end, so the message in the form in which it will be transmitted becomes:

36178 05428 99592 53507 01440 00113 42004 74684 58426 75048
42503 10084 69181 77284 83603 47503 50076 68483 88242 42838
90960 35071 37586 89914 05000 77651 80429 00873 78601 44725
44860

Two Trigraphic Ciphers, and a Heptagraphic One

Playfair for Three

Based on the Playfair cipher, I once thought of a way to make a cipher that worked on groups of three letters.

Using a square, as with Playfair:

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

encipher with the following rules:

- If all three letters are the same, replace them by three repetitions of the letter diagonally below and to the right of that letter. Thus: MMM becomes JJJ in the square above. Below, and to the right, are always interpreted cyclically, so DDD becomes RRR, and PPP becomes NNN, and even III becomes TTT.
- If two of the letters are the same, encipher the two letters as if they were part of a digraph to be enciphered with Playfair.
    - If the two letters are in the same row, replace each one with the letter to its right. Thus: PKK becomes LMM, NON becomes ZJZ.
    - If the two letters are in the same column, replace each one with the letter below it. Thus: HUH becomes UJU, ZZB becomes GGX.
    - If the two letters are neither in the same row nor the same column, replace each letter with the letter that is in its own row, but in the column of the other letter. Thus: BOO becomes SZZ, MIM becomes PSP.
- When all three letters are different, follow these rules:
    - If all three letters are in the same row, replace each one with the letter to its right. Thus, CYG becomes GAW, ZEN becomes ONZ.
    - If all three letters are in the same column, replace each one with the letter below it. Thus, MOW becomes OWS, KGB becomes ZBX.
    - If two letters are in the same row, and one of those two is in the same column as the third letter, replace the letter that is in the same row as one other letter and the same column as the other other letter with the letter that is in the same

column as the letter with which it shares a row, and in the same row as the letter with which it shares a column. Replace the two other letters by each other. Thus, YUK becomes KGY, WVY becomes HYV, POE becomes OPM, GAP becomes PKG.

- If two letters are in the same column, but the third letter is neither in that column nor in the same row as either of those two letters, replace each letter by the letter which is in its own row, but in the other column used by the three letters. Thus, TCO becomes VWN, TAN becomes RCE, HUG becomes XKY.
- If two letters are in the same row, but the third letter is neither in that row nor in the same column as either of those two letters, replace each letter by the letter which is in its own column, but in the other one of the two rows used by the letters. Thus, NED becomes FIJ, GAS becomes BIW, LOP becomes NME.
- If no two letters share either a row or column, each letter is replaced by the letter in its own row, but in the column of the next letter of the trigram, the first letter being the 'next letter' for the last one. Thus, TOY becomes VJC, LOB becomes MZF, GET becomes ANX.

Note that since a trigram with repeated letters always enciphers to a trigram with repeated letters, one could use a separate square for each of the three possibilities, or even just use an arbitrary substitution alphabet for the case of three identical letters.

Trigraphic from Fractionation

If one uses a substitution where each letter of a 27-letter alphabet is replaced by three digits from 1 to 3, then the obvious method of constructing a trigraphic cipher from this is to write the equivalents of the three letters in by columns and take them out by rows; thus, with the alphabet

W 111 M 121 Z 131 N 211 O 221 L 231 C 311 T 321 U 331
A 112 & 122 Y 132 E 212 V 222 P 232 X 312 J 322 G 332
K 113 B 123 H 133 Q 213 R 223 S 233 I 313 F 323 D 333

we encipher like this:

T H E
3 1 2 X
2 3 1 L
1 3 2 Y

For 26 Letters

But how can we adapt these two ciphers to a 26-letter alphabet?

Let's imagine that we want to have a method that doesn't require, as the original Playfair did, inserting a letter like X into the plaintext when a double letter occurs; we want something that can be applied mechanically to any arbitrary input text. This would make it suitable for use as a step in encryption performed by a computer.

For the cipher derived from Playfair, the structure of the rules provides a clue. When the extra letter turns up, ignore it for encryption, but place it in the ciphertext without alteration, and treat two remaining letters, if they are different, as in regular Playfair, and a single remaining letter (or two identical remaining letters) as if they were three identical letters.

How, though, can we possibly make the cipher which requires a 27-letter alphabet work with only 26 letters?

First, choose a substitution table such that the unused letter, &, is represented by the code 333.

U 111 F 121 P 131 Q 211 D 221 W 231 B 311 R 321 G 331
X 112 J 122 I 132 C 212 A 222 K 232 Y 312 T 322 V 332
H 113 M 123 O 133 S 213 Z 223 L 233 E 313 N 323 & 333

Now then, any combination that does not contain an ampersand, but which produces a combination that does contain one, will have produced a combination that, when enciphered again, doesn't contain an ampersand.

L O G S & G
2 1 3 S 2 3 3 L
3 3 3 & 1 3 3 O
3 3 1 G 3 3 1 G

That appears to be a trivial consequence of the fact that this cipher is reciprocal.

Since an ampersand is represented by the code 333, however, that means that whether or not a square produces an ampersand depends only on the positions of the 3s in that square; the other two digits, 1 and 2, are irrelevant. Thus, we can do better than leaving trigrams which encipher to combinations including an ampersand unenciphered.

Between the two encipherments, we can apply a substitution to the letters of the first result, as long as that substitution leaves the 3s unchanged. Since this substitution operates perpendicular to the plaintext and the ciphertext, the cipher still mixes the letters of the trigram together in this case.

Such a substitution might look like this:

111 212 113 123 131 232 311 321 133 233
112 122 123 223 132 231 312 311 233 133
121 211 213 113 231 131 321 322 313 313
122 221 223 213 232 132 322 312 323 323
211 121 331 332
212 222 332 331
221 111
222 112 333 333

With such a substitution, our encipherment would become:

L O G H & V
2 1 3 S 213 -> 113 (H) 1 3 3 O
3 3 3 & 333 -> 333 (&) 1 3 3 O
3 3 1 G 331 -> 332 (V) 3 3 2 V

Heptagraphic encryption

Having now obtained two trigraphic ciphers which both operate on trigrams of the 26-alphabet, but which operate on different principles, one is immediately tempted to combine them to create a cipher which will be much stronger than either one alone.

One way to do this is simply to apply both in sequence. However, inspired by recently encountering the polymorphic block ciphers of Kostadin Bajalcaliev, I have thought of a more elaborate way of doing this.

Let us encipher a block of seven letters at a time. Three letters are enciphered trigraphically by one of the two systems given above, and the next three are enciphered using the other system. The seventh letter is used to indicate which system is used.

Then, the letters are rearranged according to the permutation

from 1 2 3 4 5 6 7
to 4 7 1 2 3 5 6

and the process is repeated.

Since it seems wasteful to leave a letter unenciphered just to use it as the source of one bit of information, that letter could also be used to choose between twelve possibilities: there could be three different sets of tables for one of the block ciphers, and four different sets of tables for the other. For the one based on Playfair, it is not even necessary that the omitted letter be the same for each set of tables.

It was a pleasant evening in Brooklyn that summer of 1953 and Jimmy Bozart, a delivery boy for the ‘Brooklyn Eagle’ newspaper, was collecting from his customers. One lady had only a one dollar bill and Jimmy didn’t have enough change. He went across the hall and found two ladies that together were able to give him change for the dollar. After he finished collecting, he noticed that one of the coins seemed to be different from the others. The coin, a nickel, felt lighter than usual. He dropped it on the sidewalk and it split apart. Inside the hollow nickel was a piece of microfilm about one-half inch square. The small square of film was filled with rows of numbers.

Jimmy, realizing it was probably a secret code of some sort, turned it over to the police. The police suspected that the small piece of film with numbers on it had something to do with spies, so they sent it to the FBI, the government agency whose job it is to catch spies. The film was turned over to FBI cryptographers in Washington and agents began interviewing people in Jimmy's neighborhood to see if they could find out where the hollow nickel had come from. Neither the cryptographers in Washington, nor the agents conducting the search in New York, were able to make any headway in determining what message had been placed on the microfilm or the origin of the hollow nickel. On and off from 1953 to 1957, many worked on the case but could make no headway. The only fact that could be determined was that the numbers had probably been typed using a typewriter of foreign manufacture. Several foreign intelligence agents, who had defected to the United States and other free countries were contacted. Not one had any idea about the microfilm or the nickel.

Page 17 from "SECRET CODE BREAKER II – A Cryptanalyst's Handbook "

Straddle Checkerboard Cipher

Introduction §

The straddling checkerboard is a substitution cipher, except that the substitutions are of variable length. It has formed a component of several impotant field ciphers, the most notable being the VIC cipher used by russian spies during the cold war.

When combined with other methods of encryption, as shown in the example, the straddling checkerboard can be quite strong.

The Algorithm §

The key for a straddling checkerboard is a permutation of the alphabet e.g. 'fkmcpdyehbigqrosazlutjnwvx', along with 2 numbers e.g. 3 and 7. A straddling checkerboard is set up something like this (using the key information above):

```
0 1 2 3 4 5 6 7 8 9

f k m  c p d  y e

3: h b i g q r o s a z

7: l u t j n w v x
```

The first row is set up with the first eight key letters, leaving two blank spots. It has no row label. The second and third rows are labelled with whichever two digits didn't get a letter in the top row, and then filled out with the rest of the key letters. Since there are 30 slots in our grid, and we missed two letters in the first row, there will end up being two spare in the other rows. It doesn't matter where these spares go, so long as sender and receiver use the same system.

To encipher, a letter on the top row is simply replaced by the number labelling its column. Letters on the other rows are replaced by their row number, then column number:

```
D E F E N  D T  H  E E A  S T  W A L L O  F T  H  E C A  S T L E

6 9 0 9 74 6 72 30 9 9 38 37 72 75 38 70 70 36 0 72 30 9 4 38 37 72 70 9
```

So, DEFEND THE EAST WALL OF THE CASTLE becomes 690975672309938377275387070360723094383772709. This may be sent directly, but usually is first input into a second cipher stage, such as a substitution or transposition step. As a simple example, we will add a secret key number (say, 83729) using non-carrying addition:

```
837298372983729837298372983729837298372983729

+ 690974672309938377275387070360723094383772709

= 427162944282657104463659953089550282655655428
```

Then use the same straddling checkerboard to turn it back into letters:

427162944282657104463659953089550282655655428

CMU DMECCMYMDPU FCCDO PEEPH YEPPFMYMDPPDPPCMY

The final ciphertext is then CMUDMECCMYMDPUFCCDOPEEPHYEPPFMYMDPPDPPCMY. Note that it is a different length compared to the original plaintext. Deciphering is simply the reverse of this process. Although the size of groups can vary, deciphering is unambiguous because whenever the next element to be deciphered starts with a 3 or a 7, it is a pair; otherwise, it is a single letter.

- [2] Kahn, D (1973) The CodeBreakers. Macmillan: New York

The SECOM Cipher

The SECOM cipher is a handcipher to encrypt a message, containing letters, numbers and spaces. It is one of the most secure field ciphers. SECOM uses four steps: calculating the key phrase digits, the straddling checkerboard, and two columnar transpositions, of which one disrupted. The combination of fractioning by the checkerboard and the double disrupted transposition makes SECOM a very powerful encryption methode.

We will demonstrate the encryption technique by the following example:

Plain text: RV TOMORROW AT 1400PM TO COMPLETE TRANSACTION USE DEADDROP AS USUAL
Key Phrase: MAKE NEW FRIENDS BUT KEEP THE OLD

Take the first 20 letters of the key phrase and divide it into two halves. Within each half, we assign 1 to the letter earliest in the alphabet, and so on, treating 0 as the last number, and assigning digits in order to identical letters.

MAKENEWFRI ENDSBUTKEE
7162830495 3728109645

The two 10 digit results are then added, digit by digit, ignoring carries.

```
 7162830495
+3728109645
 ----------
 0880939030
```

The result of the addition is then expanded to another 50 pseudorandom digits through chain addition. Add the first two digits in the group together, take only the last digit of the result and append it to the end of the group, then ignore the first digit, and repeat the process.

0880939030

8689229338
4471412612
8185538730
9930815039
8238965327

Take the last row, assign 1 to the smallest digit, and so on, treating 0 as the last number, and assigning digits in order to identical digits. Those 10 digits are used as the top row of numbers for a straddling checkerboard.

8238965327
8139065427

The second row of the checkerboard contains the highest frequency letters ESTONIA, with a blank in the 3th , 6 th and 9 th square. Write the digits, located above an empty square, downwards in the first column. Complete the checkerboard with the following letters and numbers:

B C D F G H J K L M
P Q R U V W X Y Z *
1 2 3 4 5 6 7 8 9 0

However, we start the filling of the rows in the column, pointed to by the digit at the left of that row. In our example, the start positions are underlined. Complete until the end of that row and proceed at the beginning of the row.

| 8 1 3 9 0 6 5 4 2 7
+----------
| E S T O N I A
3| L M B C D F G H J K
6| W X Y Z * P Q R U V
2| 0 1 2 3 4 5 6 7 8 9

We convert the plain text into numbers according to the straddling checkerboard:

R V * TOM OR R OW * AT* 1 4 0 0 P M * TO* C OM P L ETE*
64676090310646406860796021202828663160906039031663889860

TR ANSAC TION* U SE* D EAD D R OP * AS* U SU AL
964751739940560621860308730306406660716062162738

In our example, several spaces are used. However, it is recommended to omit all spaces where legibility preserved.

To determine the number of columns for the two transpositions, we take the unequal digits, one by one, starting at the end of the last row of the 50 generated numbers, from right to left, and add them until the result is more than 9. The result is the width of the first columnar transposition. We continue with reading off unequal digits, from right to left, and add them to retrieve the width of the second columnar transposition:

....815039
8238965327

1st transposition: 7 + 2 + 3 = 12 columns
2nd transposition: 5 + 6 = 11 columns

Take the 10 digits from the second half of the key phrase, and the 10 digits used for the straddling checkerboard and add them, digit by digit, ignoring carries:

```
 3728109645 second half of key phrase
+8139065427 Checkerboard
 ----------
 1857164062 Result
```

The keys for the two transpositions are taken by reading out the 50 numbers by columns, using the 10 digits as a transposition key. Again, 0 is last.

```
 1857164062
 ----------
 8689229338
 4471412612
 8185538730
 9930815039
 8238965327
```

The digits for the transpositions after reading off the 23 required digits:

848982458982 09792855878

The first transposition is a simple columnar transposition. We use the first 12 of the 23 digits as transposition key, and fill the transposition block with the numbers, obtained by the checkerboard conversion. At this stage, null digits are appended to the message, so that it will fill a whole number of 5-digit groups. In our example, we add one null digit.

```
848982458982
------------
646760903106
464068607960
212028286631
609060390316
638898609647
517399405606
218603087303
064066607160
621627380
```

The message is then read off in columns, using the top row digits as transposition key:

088089367 60167630 461031162 962364063 008900808 642665206
642987841 662699062 376095770 06314006 700083606 19636631

The second transposition is a disrupted columnar transposition. We use the last 11 of the 23 digits as transposition key. The first triangular area starts at the top of the column which will be read out first, and extends to the end of the first row. It continues in the next row, starting one column later, and so on until

it includes only the digit in the last column. Then, if possible, after one full row, a second triangular area starts, this time in the column which will be read out second. A third, fourth, and more triangular areas can be added, if enough rows available.

Since we know that the message is 105 digits long, we know that we have to fill 9 rows with 11 digits, and 1 row with 6 digits. First, we fill the transposition block row by row with the numbers from the first transposition, first avoiding the triangular areas:

09792855878

0880
89367
601676
3046103
11629623
640630089
0080864266
52066429878
416626
990623XXXXX

Next, we fill in the triangular areas, row by row as well:

09792855878

0880**7609**577
8936**7006314**
601676**00670**
304610**30083**
116296**23606**
640630089**19**
008086426**66**
52066429878
41662**636631**
990623XXXXX

Again, the message is read off in columns, using the top row digits as transposition key:

7771938622 000320423 960038296 8314608060 717801673 6060606463
536069686 740369681 8900140219 0666260666 0863160549

Finally, the digits are divided in groups of 5 to get the fully encrypted message:

77719 38622 00032 04239 60038 29683
14608 06071 78016 73606 06064 63536
06968 67403 69681 89001 40219 06662
60666 08631 60549

Decrypting a message:

To decrypt a message, we use the key phrase to calculate the digits for the checkerboard and the two transpositions. Next, we apply the transpositions in reversed order.

We create the block for the second -disrupted - transposition, with the appropriate column lengths and triangular areas. We fill in the encrypted message column by column, according to the 2nd transposition key. First, we read of the message row by row, avoiding the triangular areas. Next, we read off the triangular areas, also row by row.

The result is filled in the first - simple - transposition block, also created with the appropriate column lengths, column by column according to the 1st transposition key. Again, we read off the digits row by row.

The resulting sequence of digits is converted to plain text, using the checkerboard. Note that, at the end of the sequence, up to four null digits could be added to complete a block of five, and should be disregarded during conversion.