UNIVERSITY OF CALIFORNIA
Santa Barbara

Cryptanalysis of the Schlüsselzusatz

A Thesis submitted in partial satisfaction of the
requirements for the degree Master of Science
in Computer Science

by

Nitesh Saxena

Committee in charge:
Professor Alan G. Konheim, Chair
Professor Teofilo F. Gonzalez
Professor Giovanni Vigna
Doctor David Kahn

March 2002

The Thesis of Nitesh Saxena is approved.

Teofilo F. Gonzalez

Giovanni Vigna

David Kahn

Alan G. Konheim, Committee Chair

March 2002

Cryptanalysis of the Schlüsselzusatz

ABSTRACT

Cryptanalysis of the Schlüsselzusatz

by

Nitesh Saxena

Telecipher machines were developed in the early part of this century. Their function was to encipher messages - to *secret* or *mask* their contents - prior to transmission over media which could be monitored by unauthorized parties. It was believed that encipherment of messages might prevent an enemy learning the information in a message.

Germany developed three such machines - the Schlüsselzusatz, the Enigma and the Geheimfernschreiber. The Schlüsselzusatz (key attachment in German) was developed by Lorenz in the year 1940 and thus called SZ40. It became an essential part of the German Forces during the World War II.

The Thesis consists of two parts : SZ40 encipherment and the cryptanalysis. A set of equations are formulated to carry out the key generation and encipherment. A statistical model based on cribbing is presented that cryptanalyzes the machine.

Contents

List of Figures

List of Tables

1 The Taxonomy of Codes

The use of encipherment is detected as early as 480 BC in Greece. Often, the means of secreting messages was only to physically conceal them.

An alphabet $\mathcal{A} = \{a_0, a_1, \cdots, a_{m-1}\}$ is a finite set of symbols appropriately referred to as letters; examples include

1. $m = 2^r$: (0,1)-sequences of fixed length r

   $Z_{r,2} = \{\underline{x} = (x_0, x_1, \cdots, x_{r-1}) : x_i = 0, 1, 0 \leq i < r\}$;

2. $m = 2^7$: the ASCII character alphabet;

3. $m = 26$: the alphabet of upper case latin letters : $\{\mathtt{A}, \mathtt{B}, \cdots, \mathtt{Z}\}$.

Text is formed by concatenating letters from $\mathcal{A}$; an n-gram $(a_0, a_1, \cdots, a_{n-1})$ with $a_i \in \mathcal{A}$ is a concatenation of n letters. We do not insist that the text be *understandable* or to be grammatically correct in a natural *language*; thus

```
Good_Morning   vUI*_9Uiing8
```

are both examples of text.

Encipherment or encryption is a transformation process; T transforms the plaintext $\underline{x} = (x_0, x_1, \cdots, x_{n-1})$ into ciphertext $\underline{y} = (y_0, y_1, \cdots, y_{n-1})$

It is not necessary that the plaintext and ciphertext be written using the same alphabet of symbols nor that encipherment preserve the length of text. The only requirement on T is the obvious one; it must be possible to reverse the process of encipherment - termed decipherment or decryption - by means of a transformation T^{-1} to recover the plaintext $\underline{x} = (x_0, x_1, \cdots, x_{n-1})$ from the *ciphertext* $\underline{y} = (y_0, y_1, \cdots, y_{n-1})$.

When a pair of communicating users wants to hide the content of their transmissions, the encipherment transformation must be specific to the users. A cryptographic system is a family $\mathcal{T} = \{T_k : k \in \mathbf{K}\}$ of cryptographic transformations. The key space $\mathbf{K}$ is the totality of key values; a key k is a label identifying a transformation in the family $\mathcal{T}$. The *sender* and *receiver* agree on a particular k and encipher their messages using the transformation T_k.

David Kahn's book [KAH67] provides a thorough description of the evolution of codes and the role of cryptography in history.

Originally encipherment and decipherment involved pen-and-pencil calculations. Giovanni Battista Porta (1535-1615) made contributions to astrology, optics, meteorology, magic and cryptography. His four volume work *Magiae Naturalis* was published in 1585 but he is renowned for the sequel, his one volume work divided into twenty books *magnus opus* published four years later. His book *De Furtivis Literarum Notis* published in 1563 described digraphic substitution and transposition and is considered one of the first serious work in cryptography. Porta's book *Traicté des Chiffres* published in 1586 described a variety of encryption systems.

World War I and II forced the creation of new and faster automated methods to secret messages.

A code is essentially a synthetic language invented to conceal the meaning of a message. One type of code uses a codebook, a dictionary of artificial codewords which is used to replace the plaintext message.

A cipher is another type of cryptosystem enciphering the plaintext by letter replacement - substitution - and rearranging the order of the letters - transposition.

In the 1918 a cipher system with perfect secrecy was invented. The one-time pad or one-time tape enciphers using a key stream, a sequence of (alphabetic) symbols. These are produced by independent random trials using the uniform distribution. The key stream is combined additively with the plaintext symbols; one key symbol per plaintext symbol. The term *one time* is applied since each (key) symbol is used to encipher only *one* (plaintext) letter. When this encryption recipe is strictly followed, the ciphertext symbols completely hides the plaintext and encipherment achieves perfect secrecy.

If on the other hand, the key stream symbols are re-used, the perfect secrecy property will fail to hold, as the USSR learned in World War II.

Unfortunately, the *sender* and *receiver* must both have a copy of the one-time pad. The delivery of the key stream, say from the sender to receiver is the key exchange process, a basic aspect in the successful use of cryptography.

Often, a government distributes the key on a CD or a tape to a consulate or embassy by an *alternate secure path*, for example, a courier. In some military operations, this is not feasible and a list of keys for a fixed period is often issued in advance to each military group. The logistics of updating these key tables is formidable.

The Enigma machine[1] used a protocol wherebye the key tables were used only to exchange a session key on-the-fly.

The Key to Rebecca by Ken Follett is spy novel with a cryptographic subplot. The *spy* Cicero uses a book code to transmit secret messages; the triplet (P,L,W) points to a page #, line # (L) and word # (W) in some

[1] Enigma was initially invented in 1918 for commercial applications, until it was adopted by the German Army for military encipherment.

book. Encipherment is like the one-time system; the ciphertext is the XOR of the plaintext with the key stream derived from the book's text starting at the point (P,L,W) in the (secret) book.

Claude Shannon's basic paper [SHA49] set forth the foundations of cryptographic system design identifying two building blocks of secrecy systems:

- *Substitution*

  Deriving ciphertext by substituting for the plaintext letters

  $\underline{x} = (x_0, x_1, \cdots, x_{n-1})$ with letters in a ciphertext alphabet

  $(x_0, x_1, \cdots, x_{n-1}) \to (y_0, y_1, \cdots, y_{n-1})$.

  Shannon referred to substitution as *confusion*.

- *Transposition*

  Deriving ciphertext by rearranging (or transposing) the positions of the letters in the plaintext

  $T : (x_0, x_1, \cdots, x_{n-1}) \to (x_{\pi_0}, x_{\pi_1}, \cdots, x_{\pi_{n-1}})$ where $\underline{\pi} = (\pi_0, \pi_1, \cdots, \pi_{n-1})$

  is a permutation of $0, 1, \cdots, n-1$.

Shannon referred to transposition as *diffusion*. He suggested that an effective encipherment system might be built by interleaving the operations of diffusion and confusion.

The science of codes is cryptology or cryptography, the latter term is derived from the Greek words *kryptos* (hidden) and *graphien* (to write).

The effectiveness of an encipherment process T_k with $k \in \mathbf{K}$ depends on the three factors:

1. the size of the key space $\mathbf{K}$,
2. secrecy of the key, and
3. the complexity of the transformation T_k from plaintext to ciphertext.

The term cryptanalysis is applied to the variety of techniques used to

- recover the plaintext from the ciphertext without the key, and
- recover the key from the ciphertext

It is assumed that the cryptosystem $\mathcal{T}$ is known and that additional (side) information may be available, for example,

- the subject of the plaintext;
- some of the plaintext.

Cryptography is a contest between two adversaries; the designer of the code and the cryptanalyst.

1.1 Cipher Machines

As attractive as the one-time system is cryptographically, its use is limited due to the need for large amounts of keying material. In its place, cryptographic system designers have opted for systems in which a *small* amount of keying material is used to generate a *larger* operational key. The designers of cryptographic systems in the first half of the twentieth century used mechanical means to generate the key stream. Under general conditions, the output of a *finite-state machine* is periodic; the designers sought mechanical devices to generate key streams with gigantic periods in order to emulate the one-time system. Starting with a key of N symbols, a mechanical device generates a periodic sequence of symbols of length $M >> N$. The designer hoped that key expansion $N \rightarrow M$ would offer secrecy comparable to that of the one-time system.

This process accelerated in World War I, with combatants devising devious ciphers and their adversaries cryptanalyzing them. Codebreaking proved to have a strong influence on the course of the war. The need to speedup and automate the processes of encoding and decoding led to the invention of cipher machines. They were designed to speed-up encipherment while using complex mechanisms so that the encipherment process would be *impossible* for the adversary to break. These cipher machines became an important part of the defense programs of the countries using them. Germany developed three cipher machines for its Army, Air Force and Navy; the Enigma (1918), Geheimfernschreiber[2](1930) and Schlüsselzusatz (1940/42). The Japanese also introduced mechanical devices in the 1930s given the color names RED, PURPLE and JADE (by Americans).

We can regard a cipher machine of a given design or architecture as a device that converts the input plaintext (X) into ciphertext (Y) using a key (K) by means of a function f such that Y = f(X,K).

1.2 The Vernam System

An important additive cipher system, often referred to as Vernam cipher, was invented during the First World War by two American cryptographers, Gilbert Vernam of AT&T and Colonel Joseph O. Mauborgne/.

Teleciphers were typewriters connected to a transmission system (radios, telex and telephone); they could *(i)* encode and transmit messages and *(ii)* receive, decode and print them. They were natural extensions of the nineteenth-century telegraph technology introduced after the turn of the century. Teleciphers used a digital transmission scheme, with two different electrical signals sent to represent a binary value of ”1” or ”0”.

[2] Geheimfernschreiber means secret telegraph in German

Teleciphers encoded plaintext often using the 5-bit Baudot code[3], to represent different characters. This coding scheme was a natural successor to the Morse code which uses a variable number of bits to encode some alphabet.

Although five bits are able to only represent 32 states, two of the Baudot-codes are reserved for *shift* between two sets of characters, much like the modern *Caps Lock* key on a typewriter keyboard. This allows the Baudot code to encode more than 32 characters. The Baudot code allows plaintext written with

1. upper-case characters `A B` $\cdots$ `Z`
2. digits `0 1` $\cdots$ `9`
3. punctation characters `, " ; ? !   .`
4. special characters `# $ + &` BS (space), Bell and
5. teletype control characters `LF` (line feed) `CR` (carriage return).

Table 1.1 lists the Baudot Code; two plaintext characters are listed for each 5-bit Baudot code;

1. the left-most character is in the letter group;

[3] Baudot Code was invented by French scientist Jean Maurice Emile Baudot in 1880

00011	A	-	11000	O	9
11001	B	?	10110	P	0
01110	C	:	10111	Q	1
01001	D	$	01010	R	4
00001	E	3	00101	S	BELL
01101	F	!	10000	T	5
11010	G	&	00111	U	7
10100	H	#	11110	V	;
00110	I	8	10011	W	2
01011	J	’	11011	*	*
01111	K	(	11101	X	/
10010	L	)	10101	Y	6
11100	M	.	10001	Z	”
01100	N	,	00010	LF	LF
00000	@	@	01000	CR	CR
00100	SP	SP	11111	+	+
CF :	carriage return		SP :	white space	
LF :	line feed		BELL :	bell	
+ :	letters-numbers		* :	numbers-letters	
@ :	undefined				

Table 1.1: Baudot Codes

2. the right-most character is in the digit group.

The Baudot code is not very efficient and has largely been replaced by the 7-bit ASCII (American Standard Code For Information Interchange).

A telecipher encipers binary codes using a key stream; when the Baudot encoding of plaintext characters is used, a key stream consisting of 5-bit key (B-bytes is XORed to the 5-bit plaintext. Encipherment is a two step process:

1. *encode* : each plaintext character is replaced by its 5-bit Baudot code;
2. *encipher* : combine the encoded plaintext character by a bit-by-bit XOR with a 5-bit key B-byte.

For example

```
plaintext  :     10011
XOR key    :     00110
----------------------
ciphertext :     10101
```

The plaintext is recovered from ciphertext if the key B-byte is XORed to the ciphertext.

Vernam's key stream was written on a paper tape in blocks of 5-bits with the ends of the paper tape glued together to form a loop yielding a periodic running key $\underline{k} = (k_0, k_1, \cdots)$ of period r (say). See Figure 1.1.

Realizing that the reuse of the key $k_i = k_{i\ (modulo\ r)}$ might weaken the encipherment, Vernam suggested combining several tapes whose periods $\{r_i\}$ were relatively prime to each other. The composite tape thereby has a period equal to the product of $r = \prod_i r_i$. By this process, Vernam converted a total of $\sum_i r_i$ key values into a periodic key with a potentially much larger period. See Figure 1.2.

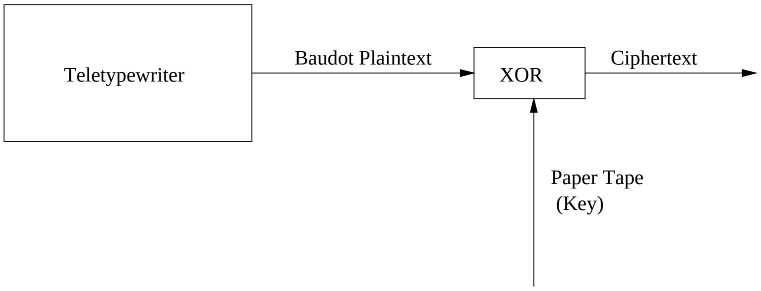

Figure 1.1: Vernam Encipherment Technique (using one paper tape)

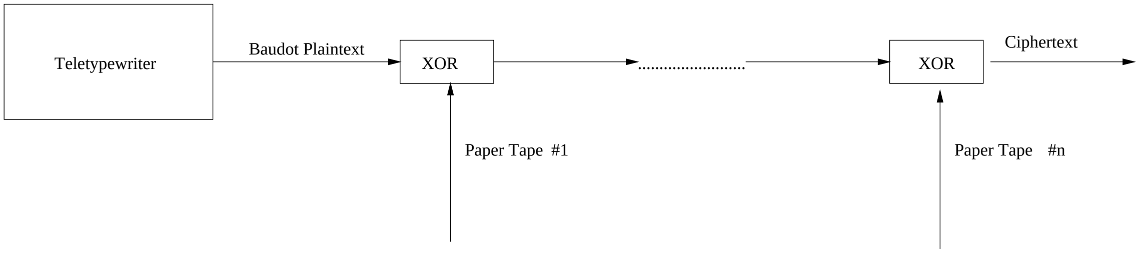

Figure 1.2: Vernam Encipherment Technique (using n paper tapes)

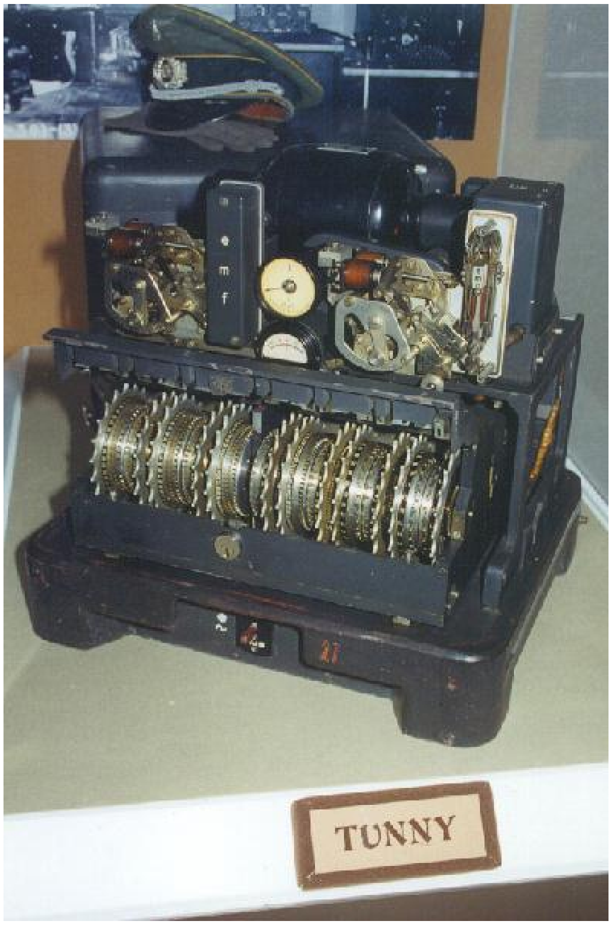

Figure 2.1: SZ40 : The Actual Machine

2 Description of the Schlüsselzusatz

2.1 The Pin-Wheels

The architecture of the SZ40 is described in the book by F. Hinsley and A. Stripp [HiSt01].

The SZ40 generated a key stream using pin-wheels in place of tapes as in the Vernam system. A pin-wheel is a mechanical implementation

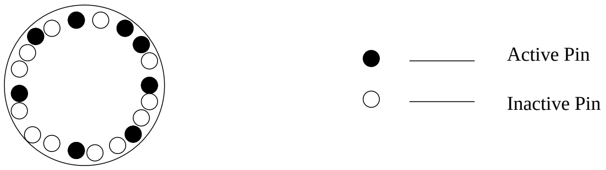

Figure 2.2: Pin Wheel

of a tape, generating a sequence of 0's and 1's. A pin-wheel of length L contains L pins equally spaced arranged around its circumference; a pin is either active or inactive.

- when a pin is active (present), the pin-wheel output is 1;
- when a pin is inactive (absent/folded down), the pin-wheel utput is 0.

The SZ40 had 12 pin-wheels divided into three sets:

1. K1-K5 : five χ pin-wheels $\{\chi_i : 1 \leq i \leq 5\}$ of lengths $\{T_i\}$.

χ_i-wheel					
i	1	2	3	4	5
T_i	41	31	29	26	23

2. S1-S5 : five ψ pin-wheels $\{\psi_{:}1 \leq i \leq 5\}$ of lengths $\{S_i\}$.

ψ_i-wheel					
i	1	2	3	4	5
S_i	43	47	51	53	59

3. M1-M2 : two motor pin-wheels

- a μ pin-wheel of length 37;
- a π pin-wheel of length 61.

The χ and ψ pin-wheels XORed bits to the plaintext just as in the 2-tape Vernam system. In the two step process

1. the output of the χ pin-wheels is first XORed (bit-by-bit) to the 5-bit Baudot plaintext;
2. the output of the ψ pin-wheels is next XORed (bit-by-bit) to the 5-bit output of the χ-wheels.

If this was the complete description of the SZ40-encipherment, it would provide little in the way of secrecy. The *German Cipher Bureau* understood the limitations of Vernam-Vigenére encipherment. Even with multiple tapes, a cryptanalysis is possible as described in [TUC70] . This was proved in 1918 by W. F. Friedman in the US, who solved such a system. By introducing key-dependent irregular motion in the second tape, the Germans believed that a significant increase in secrecy would result.

2.2 The Motion of the Pin-Wheels

The pin-wheels *either*

- rotated one position (counterclockwise) after the encipherment of each plaintext character *or*
- remained stationary - did not rotate.

The rules governing rotation of the pin-wheels are:

1. each of the five χ pin-wheels rotated one position (counterclockwise) after the encipherment of each plaintext character;
2. each of the five ψ pin-wheels rotated (counterclockwise) one position after the encipherment of each plaintext character *if and only if* the current μ pin-wheel output was 1;
3. the μ pin-wheel rotated (counterclockwise) one position after the encipherment of each plaintext *if and only if* the current π pin-wheel output was 1;
4. the π pin-wheel rotated (counterclockwise) one position after the encipherment of each plaintext character.

2.3 SZ40 Keys

The SZ40 key had two components;

1. the 501 bits determining the active pins on the pin-wheels;

2. the initial positions of the 5 χ, 5 ψ and 2 motor pin-wheels μ and π.

The first component was originally changed each month; it was intended that the second component was to be changed with each message. Initially, an SZ40 message began with an indicator transmitted *in the clear* consisting of 12 alphabetic characters, for example `HQIBPEXEZMUG`. A character translated into a 12-tuple of integers in $\{0, 1, \cdots, 25\}$ specifying the initial settings of the 12 pin-wheels so that not all initial settings were possible. Subsequently, the indicator was replaced by an entry in a codebook which was translated into initial wheel settings.

2.4 Key Generation

We introduce the following notation:

$X(j)$: the j^{th} plaintext character;

$X_i(j)$: i^{th} bit of the j^{th} 5-bit Baudot-coded plaintext-block;

$Y_i(j)$: i^{th} bit of the j^{th} 5-bit SZ40 ciphertext-block;

$K_i(j)$: i^{th} bit of the j^{th} 5-bit SZ40 key-block.

The variables $X_i(j), Y_i(j)$ and $K_i(j)$ are related by

$$Y_i(j) = (X_i(j) + K_i(j)) \ (modulo\ 2); \quad 1 \leq i \leq 5; \quad 0 \leq j < N$$

where N is the length of the plaintext.

The key generated is the XOR of the output signals from χ and ψ pin-wheels.

$K_i(j) = (\chi_i(P_i[j]) + \psi_i(Q_i[j]))\ (modulo\ 2)\ 1 \leq i \leq 5\ j = 0, 1, 2, \cdots, N\text{–}1$

where

- $P_i[j]$: position of the i^{th} χ pin-wheel for the j^{th}-plaintext-block;
- $Q_i[j]$: position of the i^{th} ψ pin-wheel for the j^{th}-plaintext-block;
- $U[j]$: position of the μ pin-wheel for the j^{th}-plaintext-block;
- $V[j]$: position of the ψ pin-wheel for the j^{th}-plaintext-block.

2.5 SZ40 Encipherment Steps

1. encode the j^{th}-plaintext letter $X(j)$ using the 5-bit Baudot code to $(X_1(j), X_2(j), \cdots, X_5(j))$;
2. XOR bit-by-bit $(X_1(j), X_2(j), \cdots, X_5(j))$ with the current χ pin-wheel

   output $(\chi_1(P_1[j]), \chi_2(P_2[j]), \cdots, \chi_5(P_5[j]))$ producing the intermediate

   ciphertext-block $(\widetilde{X}_1(j), \widetilde{X}_2(j), \cdots, \widetilde{X}_5(j))$;

3. XOR bit-by-bit the intermediate ciphertext $(\widetilde{X}_1(j), \widetilde{X}_2(j), \cdots, \widetilde{X}_5(j))$; with the current ψ pin-wheel output $(\psi_1(Q_1[j]), \psi_2(Q_2[j]), \cdots, \psi_5(Q_5[j]))$ producing the 5-bit ciphertext-block $(Y_1(j), Y_2(j), \cdots, Y_5(j))$;

4. Move the pin-wheels according to the following equations :

   $P_i[j+1] = (P_i[j]+1)\ (modulo\ T_i)\ 1 \leq i \leq 5\ j = 1, 2 \cdots$;

   $Q_i[j+1] = (Q_i[(j) + \mu(U[j]))\ (modulo\ S_i)\ 1 \leq i \leq 5\ j = 1, 2 \cdots$;

   $U[j+1] = (U[(j) + \pi(V[j]))\ (modulo\ 37)\ j = 1, 2 \cdots$;

   $V[j+1] = (V[j]+1)\ (modulo\ 61)\ j = 1, 2 \cdots$.

The above steps are repeated until all the letters in the plaintext have been processed. Figure 2.3 shows the operation of SZ40.

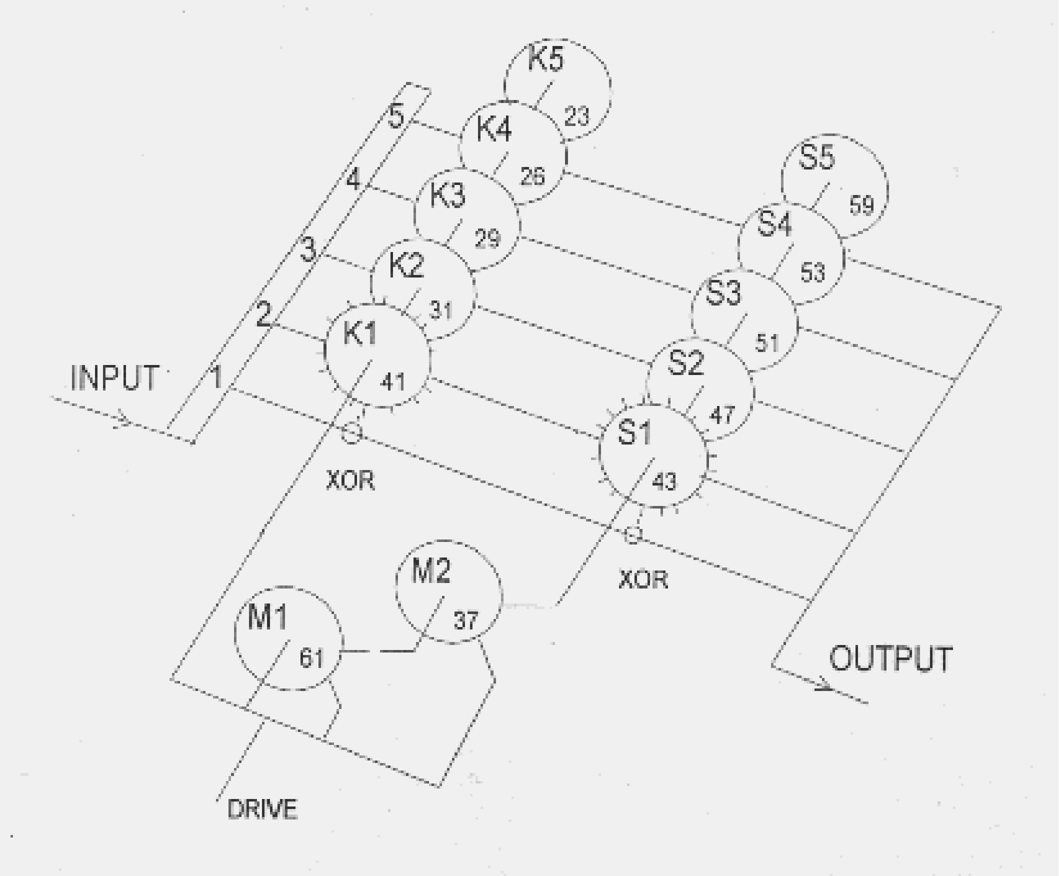

Figure 2.3: Operation of SZ

2.6 Initial Positions

To further complicate the SZ40 encipherment process, the initial positions of the pin-wheels could be set.

2.7 SZ40 Encipherment Example

Initial pin-wheel settings are all 0.

χ_1-Wheel							
1	0	0	1	0	0	1	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1							

χ_2-Wheel							
0	1	0	1	0	1	0	0
1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	

χ_3-Wheel							
1	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1			

χ_4-Wheel							
0	1	1	0	0	0	0	1
1	1	1	1	0	0	1	1
1	1	1	1	1	1	1	1
1	1						

χ_5-Wheel							
1	0	0	0	1	1	0	0
1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	

ψ_1-Wheel							
0	0	1	0	0	1	1	1
0	1	1	0	1	0	1	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1					

ψ_2-Wheel							
0	1	1	1	0	0	1	1
1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	

ψ_3-Wheel							
1	0	0	0	1	0	0	0
1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1					

ψ_4-Wheel							
1	0	0	1	0	0	0	0
0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1			

ψ_5-Wheel							
0	1	0	1	1	1	1	1
0	0	1	0	1	1	0	1
1	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1					

μ-Wheel							
1	1	1	1	1	1	1	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	1	0
1	1	1	1	1	1	1	0
1	1	1	1	1			

π-Wheel							
1	1	1	0	1	1	1	0
1	1	0	1	1	1	0	0
1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1
0	1	1	1	1			

Plaintext : `MERRY CHRISTMAS`
11100 00001 01010 10101 10101 00100 01110 10100
01010 00110 00101 10000 11100 00011 00101

Operation :

Character	Baudot Code	χ-Wheel Outputs	Input to ψ-Wheels	ψ-Wheel Outputs	Output
M	11100	10101	01001	00110	01111
E	00001	01010	01011	01001	00010
R	01010	00110	01100	11000	10100
R	01010	11100	10110	01011	11101
Y	10101	00101	10000	00101	10101
	00100	01101	01001	10001	11000
C	01110	10100	11010	11001	00011
H	10100	00110	10010	11001	01011
R	01010	11111	10101	01100	11001
I	00110	10010	10100	11000	01100
S	00101	11111	11010	11111	00101
T	10000	11111	01111	11111	10000
M	11100	11101	00001	11111	11110
A	00011	11101	11110	01110	10000
S	00101	11111	11010	01110	10100

Ciphertext : 01111 00000 10100 11101 10101 11000 00011 01011
11001 01100 00101 10000 11110 10000 10100

3 Chronology of the Fish Traffic

SZ40 ciphertext traffic was referred to as fish as described in [TUT98] . Both the cryptographic device and the special processors built to carry out the cryptanalysis of the SZ40 were referred to as tunny as in [SAT01]; these *first* generation processors were designed by the British *General Communications Headquarters* (GCHQ) located in *Bletchley Park* outside of London where the SZ40 cryptanalysis activities took place.

- *1941* : The first regular transmission for the fish ciphertext were intercepted on an experimental German Army link between Vienna and Athens in the middle of *1941*.

- *1942* : The fish traffic proliferated steadily from the middle of 1942. The first success against the fish keys, the solution of the Athens-Vienna experimental link in the spring of 1942, was made by hand methods and it was by these methods that overcoming various cryptographic and procedural improvements made by the Germans, *Global Command and Control System* GC&CS kept abreast of the still limited number of fish links until May 1943. By December 1942, it was clear that the German program for increasing the security of the ciphers would win unless high speed processing devices were developed.

The first stage was the *Heath Robinson*[4], a mechanical device using two tapes driven by a pulley system. It was used to evaluate the Boolean functions. A complete description of the machine can be found in [SAH01].

- *1943* : By July 1943, there were six links, by autumn 1943 ten links. In May 1943, the first prototype of the Heath Robinson machine became available and was then superseded by *Collossus*[5] *Mark I* described in [AND01] , which came into use from the end of 1943. Alan Turing was the architect of the Collussi series; the machines were built by the UK Post Office Research Station Dollis Hills (London) for for the *Government Code and Cipher School.* Mark I contained 1500 *thermionic vales* (electronic tubes); each character was coded with the 5-bit Baudot teleprinter code, read by an optical character reader and punched on a paper moving at the rate of 5000 characters/s. It began analyzing ciphertext at Bletchley Park in December 1943. Its successor, *Colossus Mark II* (1944) contained 2500 valves allowed conditional branching but did not implement the *internal program store* central to the concept of a computer.

[4] Heath Robinson is the name of a British cartoonist who was renowned for his drawings of outlandish machines, like Rube Goldberg in the United States.

[5] It was primarily to break the cipher of SZ40 that the British devised what is now considered the world's first electronic computing machine, the Collossus.

- *1944-45* : From early in 1944 the fish traffic comprised twenty six links, each using different settings, between Berlin and the chief Army Commands. The decryption was also increasingly successful; with two serious interruptions, in February 1944 and from June to October 1944, GC&CS solved a growing proportion of the ciphers and decrypted more or less regularly the growing volume of traffic passing in them. On June 1, 1944 *Collossus Mark II* was brought into commission. The number of decrypted transmissions rose from an average of only three hundred a month in 1943, so that more were decrypted in the six months from October 1944 to March 1945 than in all the period from the summer of 1942 to September 1944.

A more detailed description of above events can be found in [HiSt01] and [SAF01]

4 Cryptanalysis

Once the structure of the SZ40 has been diagnosed by GCHQ at Bletchley Park, the cryptanalysis could be undertaken. Various versions of the cryptanalysis problem exist of which the following is the most challenging:

- *Problem #1*

  Given a ciphertext Y, determine the plaintext X. A solution involves determining all pin-wheel parameters - active pins and initial positions.

This is an *ill-posed* problem since an initial setting together with a pin-wheel determines an equivalent pin-wheel with zero initial setting. Additional complications include *(i)* the unknown density of active pins and *(ii)* the key management employed by the Germans. [CAR97] describes the method used by the British in cryptanalyzing SZ40.

4.1 Key Determination by Cribbing

Depth occurs when two or more SZ40 ciphertexts $\underline{Y}_i$ $(i = 1, 2, \cdots)$ were intercepted in a period

- during which the pin-wheels were unchanged and

- both messages were identified the <u>same</u> indicator.

In this case $\underline{Y}_i = \underline{X}_i + \underline{K}$ for $i = 1, 2, \cdots$. Computing the <u>differences</u> $\underline{Y}_1 + \underline{Y}_2 \equiv \Delta\underline{Y}_{1,2}$ gives $\Delta\underline{X}_{1,2} \equiv \underline{X}_1 + \underline{X}_2$ since the *differenced* key is $\underline{0}$.

The differenced plaintext can be searched for probable words (cribs); for example,

- German cipher-clerks often prefaced their messages with `SPRUCHNUMMER`[6], and
- messages might contains references to various organizations `LUFTWAFFE`, `WEHRMACHT`. `OBERKOMMANDO`.

To search for the <u>crib</u> `SPRUCHNUMMER`, the crib is *slid* across the differenced ciphertext. In each position j, the crib is XORed with the differenced ciphertext

$$\begin{array}{rclllllll}
\underline{\mathbf{X}}_{\mathbf{1}} & : & \cdots & \mathtt{S} & \mathtt{P} & \cdots & \mathtt{M} & \mathtt{E} & \cdots \\
\underline{\mathbf{X}}_{\mathbf{2}} & : & \cdots & X_2(j) & X_2(j{+}1) & \cdots & X_2(j{+}8) & X_2(j{+}9) & \cdots \\
\hline
\boldsymbol{\Delta}\underline{\mathbf{X}}_{\mathbf{1,2}} & : & \cdots & \mathtt{S}{+}X_2(j) & \mathtt{P}{+}X_2(j{+}1) & \cdots & \mathtt{M}{+}X_2(j{+}8) & \mathtt{E}{+}X_2(j{+}9) & \cdots
\end{array}$$

The result of the XOR of the crib and the differenced plaintext at position j give *putative* plaintext

[6] Spruchnummer means message number in German.

$X_2(j)$, $X_2(j+1)$, $\cdots$, $X_2(j+8)$, $X_2(j+9)$, $X_2(j+10)$

If the putative plaintext is *readable text*, a hit is obtained, which generally reveals additional plaintext. With good luck, both plaintexts $\underline{X}_1$ and $\underline{X}_2$ may be discovered and from (either) the (common) key $\underline{K}$.

Early in the SZ40-cryptanalysis, an interception of the near-repeat of message of 4000 characters using the same indicator (and pin-wheel settings) was received providing the entire key stream.

When cribbing is successful, a segment of the (common) key stream $\{\underline{K}(j)\ j = 0, 1, \cdots, N-1\}$ is uncovered.

We will illustrate the solution of a second cryptanalysis problem:

- *Problem #2*

  Given a key stream K generated by the machine, determine the active pins on all pin-wheels.

Problem #2 does not have a unique solution since complementing the χ and ψ pin-wheel values leads to the same key stream.

4.2 A Statistical Model of Pin Motion

In this section we present a solution to the Problem #2.

We define the SZ40 parameters

- q : the *averaged* density of active pins on ψ pin-wheels;
- ν : the *average* probability that a ψ pin-wheel rotates.

The values of q and ν are unknown and must be guessed and later refined as a result of the cryptanalysis.

For the pin-wheel parameters in the §2.7 example, $\nu \simeq 0.8$ and $q \simeq 0.8$. The graph in Figure 4.1 shows the motion of the ψ pin-wheel rotates for several initial positions.

The parameters q and ν can be used to define a statistical model of the ψ_1 pin-wheel rotation. Let $\delta_{(i,j)}$ be the probability that $\psi_1(j, j+1) \equiv (\psi_1(Q_1(j)), \psi_1(Q_1(j+1))) = (a, b)$ with $(a, b) \in \{(0,0), (0,1), (1,0), (1,1)\}$. Assuming that the motion of the pin-wheels at all positions are approximately independent and identicially distributed leads to the formulae

- $\delta_{(0,0)} = \nu(1-q)^2 + (1-\nu)(1-q) = 0.072;$
- $\delta_{(1,1)} = \nu q^2 + (1-\nu)q = 0.672;$
- $\delta_{(0,1)} = \nu(1-q)q = 0.128$
- $\delta_{(1,0)} = \nu(1-q)q = 0.128.$

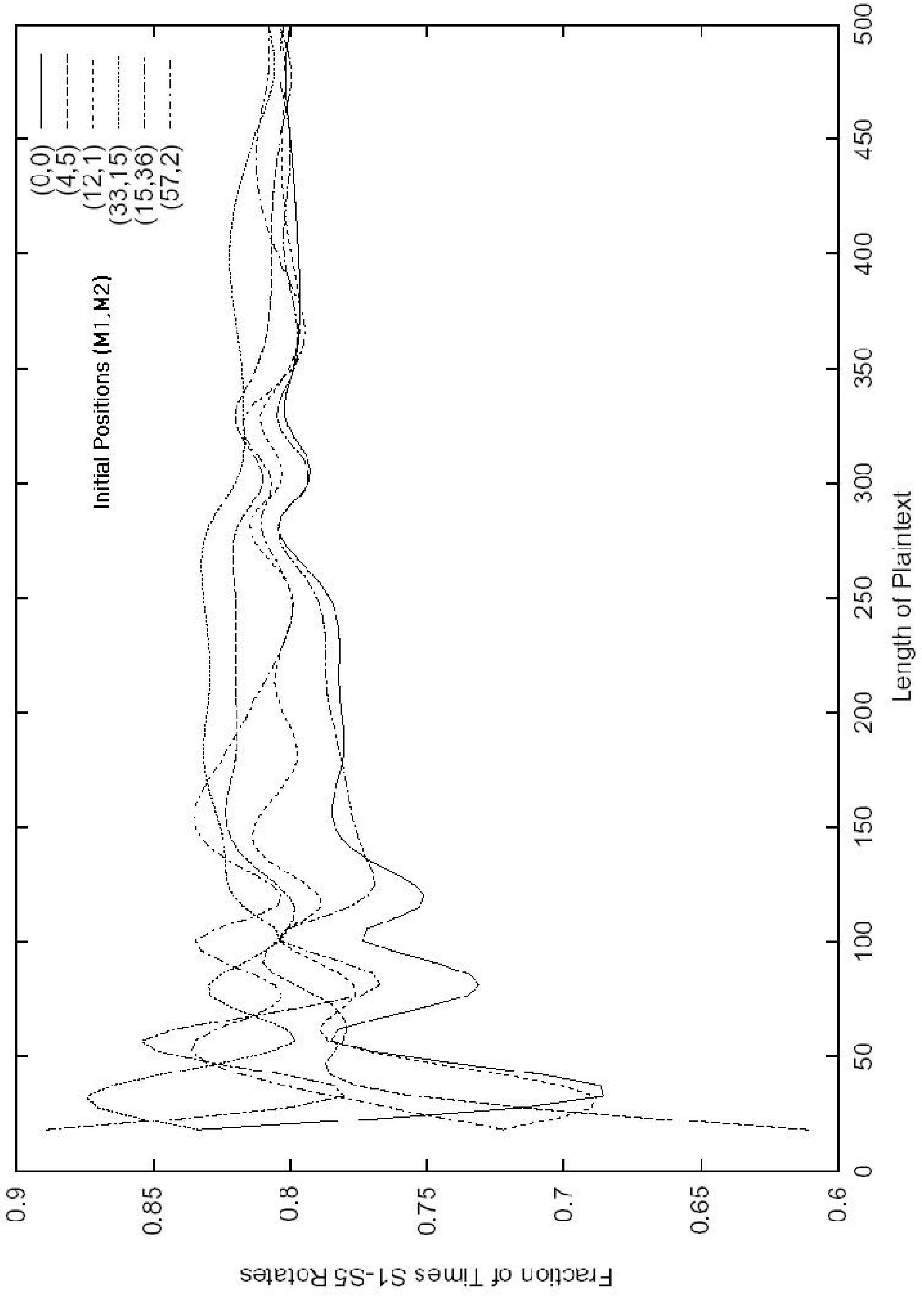

Figure 4.1: Frequency of Rotation for Different Initial Positions

This statistical model of pin-wheel motion implies that in a *large* sample of R positions, there will be $\simeq Rq_{(a,b)}$ values of j for which $\psi_1(j, j+1) = (a, b)$.

4.3 Finding Active Pins on χ Wheels

We make use of the above observation and the periodicity of the χ wheels to identify the pins on all pin-wheels from the key stream.

The notation $K_i(j, j+1) \equiv (K_i(j), K(j+1))$ is used below.

```
for (each position) $j$ do
        begin
for (each wheel) $i$ do
        begin
```

Count the number of times $\mathrm{KCount}_{(\mathrm{i,j})}[\mathrm{a, b}]$, the key bits $K_i(j, j+1)$ is equal to (a, b) for each of the four pairs (a,b) $\in \{(0,0),(0,1),(1,0),(1,1)\}$ for the key stream at positions $j, j+T_i, j+2T_i, \cdots, j+(k_i-1)T_i$ where k_i depends on the length of the known key stream.

```
        end
        end
```

4.4 Example

The key is generated using the same parameters as in §2.8. In the tables that follow for $j = 0(1)4$ the <u>un</u>known values of $\psi\mathrm{Count}_{(i,j)}[a, b]$, defined as the the number of times the pair $\psi_i(j, j+1$ is equal to (a, b) for each of the four pairs (a,b) $\in \{(0,0),(0,1),(1,0),(1,1)\}$ for the key stream at positions $j, j+T_i, j+2T_i, \cdots, j+(k_i-1)T_i$ where k_i depends on the length of the known key stream is tabulated.

Testing Position 0				
KCount$_{(1,0)}$	[0,0] 1	[0,1] 11	[1,0] 1	[1,1] 0
ψCount$_{(1,0)}$	[0,0] 1	[0,1] 0	[1,0] 1	[1,1] 11
KCount$_{(2,0)}$	[0,0] 4	[0,1] 1	[1,0] 12	[1,1] 0
ψCount$_{(2,0)}$	[0,0] 1	[0,1] 4	[1,0] 0	[1,1] 12
KCount$_{(3,0)}$	[0,0] 1	[0,1] 15	[1,0] 2	[1,1] 0
ψCount$_{(3,0)}$	[0,0] 2	[0,1] 0	[1,0] 1	[1,1]] 15
KCount$_{(4,0)}$	[0,0] 1	[0,1] 3	[1,0] 14	[1,1] 2
ψCount$_{(4,0)}$	[0,0] 3	[0,1] 1	[1,0] 2	[1,1] 14
KCount$_{(5,0)}$	[0,0] 1	[0,1] 14	[1,0] 4	[1,1] 3
ψCount$_{(5,0)}$	[0,0] 4	[0,1] 3	[1,0] 1	[1,1] 14

Testing Position 1				
KCount$_{(1,1)}$	[0,0] 0	[0,1] 2	[1,0] 1	[1,1] 10
ψCount$_{(1,1)}$	[0,0] 0	[0,1] 2	[1,0] 1	[1,1] 10
KCount$_{(2,1)}$	[0,0] 2	[0,1] 14	[1,0] 1	[1,1] 0
ψCount$_{(2,1)}$	[0,0] 1	[0,1] 0	[1,0] 2	[1,1] 14
KCount$_{(3,1)}$	[0,0] 2	[0,1] 1	[1,0] 14	[1,1] 1
ψCount$_{(3,1)}$	[0,0] 1	[0,1] 2	[1,0] 1	[1,1] 14
KCount$_{(4,1)}$	[0,0] 14	[0,1] 1	[1,0] 0	[1,1] 5
ψCount$_{(4,1)}$	[0,0] 5	[0,1] 0	[1,0] 1	[1,1] 14
KCount$_{(5,1)}$	[0,0] 3	[0,1] 2	[1,0] 2	[1,1] 15
ψCount$_{(5,1)}$	[0,0] 3	[0,1] 2	[1,0] 2	[1,1] 15

Testing Position 2				
$\text{KCount}_{(1,2)}$	[0,0] 0	[0,1] 1	[1,0] 9	[1,1] 3
$\psi\text{Count}_{(1,2)}$	[0,0] 1	[0,1] 0	[1,0] 3	[1,1] 9
$\text{KCount}_{(2,2)}$	[0,0] 3	[0,1] 0	[1,0] 13	[1,1] 1
$\psi\text{Count}_{(2,2)}$	[0,0] 0	[0,1] 3	[1,0] 1	[1,1] 13
$\text{KCount}_{(3,2)}$	[0,0] 15	[0,1] 1	[1,0] 0	[1,1] 2
$\psi\text{Count}_{(3,2)}$	[0,0] 2	[0,1] 0	[1,0] 1	[1,1] 15
$\text{KCount}_{(4,2)}$	[0,0] 0	[0,1] 14	[1,0] 4	[1,1] 2
$\psi\text{Count}_{(4,2)}$	[0,0] 4	[0,1] 2	[1,0] 0	[1,1] 14
$\text{KCount}_{(5,2)}$	[0,0] 2	[0,1] 3	[1,0] 2	[1,1] 15
$\psi\text{Count}_{(5,2)}$	[0,0] 2	[0,1] 3	[1,0] 2	[1,1] 15

Testing Position 3				
$\text{KCount}_{(1,3)}$	[0,0] 1	[0,1] 8	[1,0] 1	[1,1] 3
$\psi\text{Count}_{(1,3)}$	[0,0] 1	[0,1] 3	[1,0] 1	[1,1] 8
$\text{KCount}_{(2,3)}$	[0,0] 3	[0,1] 13	[1,0] 0	[1,1] 1
$\psi\text{Count}_{(2,3)}$	[0,0] 0	[0,1] 1	[1,0] 3	[1,1] 13
$\text{KCount}_{(3,3)}$	[0,0] 13	[0,1] 2	[1,0] 1	[1,1] 2
$\psi\text{Count}_{(3,3)}$	[0,0] 2	[0,1] 1	[1,0] 2	[1,1] 13
$\text{KCount}_{(4,3)}$	[0,0] 4	[0,1] 0	[1,0] 1	[1,1] 15
$\psi\text{Count}_{(4,3)}$	[0,0] 4	[0,1] 0	[1,0] 1	[1,1] 15
$\text{KCount}_{(5,3)}$	[0,0] 1	[0,1] 3	[1,0] 17	[1,1] 1
$\psi\text{Count}_{(5,3)}$	[0,0] 3	[0,1] 1	[1,0] 1	[1,1] 17

Testing Position 4				
$\text{KCount}_{(1,4)}$	[0,0] 0	[0,1] 2	[1,0] 1	[1,1] 10
$\psi\text{Count}_{(1,4)}$	[0,0] 0	[0,1] 2	[1,0] 1	[1,1] 10
$\text{KCount}_{(2,4)}$	[0,0] 0	[0,1] 2	[1,0] 13	[1,1] 1
$\psi\text{Count}_{(2,4)}$	[0,0] 2	[0,1] 0	[1,0] 1	[1,1] 13
$\text{KCount}_{(3,4)}$	[0,0] 13	[0,1] 1	[1,0] 2	[1,1] 2
$\psi\text{Count}_{(3,4)}$	[0,0] 2	[0,1] 2	[1,0] 1	[1,1] 13
$\text{KCount}_{(4,4)}$	[0,0] 4	[0,1] 1	[1,0] 0	[1,1] 15
$\psi\text{Count}_{(4,4)}$	[0,0] 4	[0,1] 1	[1,0] 0	[1,1] 15
$\text{KCount}_{(5,4)}$	[0,0] 17	[0,1] 1	[1,0] 4	[1,1] 0
$\psi\text{Count}_{(5,4)}$	[0,0] 0	[0,1] 4	[1,0] 1	[1,1] 17

4.4.1 Step 1 : Inference of the $\chi_1(j, j+1)$-Value

The *hypothesis* $\chi_1(j, j+1) = (A, B)$ is tested as follows;

– define [a,b] by $\text{KCount}_{(1,j)}[\text{a, b}] = \max_{[c,d]} \text{KCount}_{(1,j)}[\text{c, d}]$

– [A,B] is uniquely determined by the condition [A,B] + [a,b] = [1,1].

How do we reconcile the uniqueness of $\chi_1(j, j+1)$ with the asserted <u>non</u>uniqueness of the solution to Problem #2?

With the parameters in §4.1.3 ($q \simeq 0.8$ and $\nu \simeq 0.8$), we have $\delta_{(1,1)} = \max_{(r,s)} \delta_{(r,s)}$. When the χ and ψ pin-wheel values are complemented $q \simeq 0.2$ and $\nu \simeq 0.8$ so that $\widetilde{\delta}_{(0,0)} = \max_{(r,s)} \widetilde{\delta}_{(r,s)}$ where the *tilde* denotes computation with $q \simeq 0.2$. Note that $\delta_{(1,1)} = \widetilde{\delta}_{(0,0)}$. The correct value of (A,B) will be defined by (A,B) + (a,b) = (0,0).

Note that [1,1] is the most frequently occurring pair. If $j = 4$, then

– $\text{KCount}_{(2,4)}[1, 0]$ is the maximum of $\text{KCount}_{(2,4)}[\text{a, b}]$;

– (0,1) + (1,0) = (1,1)

The process just described recovers the $\chi_1(j)$ pin-wheel values. It remains to find the $\psi_1(j)$ pin-wheel values. These values are partially obscured by the action of the motor pin-wheels.

4.4.2 Step 2 : Inference of the $\psi_1(Q[j])$-Value

Tables 4.1-4.5 that follow list for $j = 0(1)199$

- the *unknown* move indicator (MI(j)) with values (M/N) specifying whether or not the ψ pin-wheels moved;
  - equal to M if $\mu(Q[j]) = 1$;
  - equal to N if $\mu(Q[j]) = 0$;
- the unknown true position of the ψ_1 pin-wheel;
- the inferred $\underline{\chi}(j, j+1)$ and $\underline{\psi}(Q_i[j], Q_i[j+1])$;
- the 5-bit known key obtained from cribbing;
- an inference of the unknown move indicator (M/M?);
  - equal to M if for *at least* one index i, we have $\psi_i(Q_i[j]) \neq \psi_i(Q_i[j+1])$;
  - equal to M? if for *all* indices i, we have $\psi_i(Q_i[j]) = \psi_i(Q_i[j+1])$.

j	**MI(j)**	$\underline{\chi}$	$\underline{\psi}$	$\underline{K}$	**M?**
0	M(0)	10101	00110	10011	M
1	M(1)	01010	01001	00011	M
2	M(2)	00110	11000	11110	M
3	M(3)	11100	01011	10111	M
4	M(4)	00101	00101	00000	M
5	M(5)	01101	10001	11100	M
6	M(6)	10100	11001	01101	M?
7	M(7)	00110	11001	11111	M
8	M(8)	11111	01100	10011	M
9	M(9)	10010	11000	01010	M
10	N(10)	11111	11111	00000	M?
11	N(10)	11111	11111	00000	M?
12	M(10)	11101	11111	00010	M
13	N(11)	11101	01110	10011	M?
14	M(11)	11111	01110	10001	M
15	M(12)	11111	11111	00000	M
16	M(13)	11111	01111	10000	M
17	M(14)	11111	10110	01001	M
18	M(15)	11111	01111	10000	M
19	N(16)	11111	11111	00000	M?
20	M(16)	11111	11111	00000	M
21	N(17)	11111	10100	01011	M?
22	M(17)	11111	10100	01011	M
23	N(18)	11111	11110	00001	M?
24	N(18)	11110	11110	00000	M?
25	N(18)	11110	11110	00000	M?
26	N(18)	11100	11110	00010	M?
27	M(18)	11111	11110	00001	M
28	M(19)	11111	10111	01000	M
29	M(20)	11100	11111	00011	M?
30	M(21)	11000	11111	00111	M?
31	N(22)	10001	11111	01110	M?
32	M(22)	11100	11111	00011	M?
33	M(23)	10111	11111	01000	M?
34	M(24)	11111	11111	00000	M?
35	M(25)	10111	11111	01000	M?
36	M(26)	11111	11111	00000	M?
37	M(27)	10111	11111	01000	M?
38	M(28)	10001	11111	01110	M?
39	N(29)	11101	11111	00010	M?

Table 4.1: Infering ψ Wheels (continued)

j	**MI(j)**	$\underline{\chi}$	$\underline{\psi}$	$\underline{K}$	**M?**
40	N(29)	10111	11111	01000	M?
41	M(29)	11111	11111	00000	M?
42	M(30)	01111	11111	10000	M?
43	M(31)	01111	11111	10000	M?
44	M(32)	11111	11111	00000	M?
45	M(33)	01111	11111	10000	M?
46	M(34)	01111	11111	10000	M?
47	M(35)	11110	11111	00001	M?
48	M(36)	01110	11111	10001	M?
49	M(37)	11110	11111	00001	M?
50	M(38)	11111	11111	00000	M?
51	M(39)	11111	11111	00000	M?
52	M(40)	11100	11111	00011	M?
53	M(41)	11110	11111	00001	M?
54	M(42)	11111	11111	00000	M
55	M(0)	11100	01111	10011	M?
56	N(1)	11101	01111	10010	M?
57	N(1)	11101	01111	10010	M?
58	M(1)	11101	01111	10010	M
59	N(2)	11011	11111	00100	M?
60	M(2)	11011	11111	00100	M
61	M(3)	11111	01111	10000	M
62	M(4)	10111	00111	10000	M
63	N(5)	11111	11111	00000	M?
64	M(5)	10101	11111	01010	M
65	M(6)	11101	10111	01010	M
66	N(7)	10111	11111	01000	M?
67	M(7)	11011	11111	00100	M
68	N(8)	10111	00111	10000	M?
69	N(8)	10111	00111	10000	M?
70	M(8)	11110	00111	11001	M
71	M(9)	10110	10011	00101	M
72	M(10)	11110	11011	00101	M
73	M(11)	11111	01001	10110	M
74	M(12)	11111	11101	00010	M
75	N(13)	11110	01011	10101	M?
76	N(13)	11110	01011	10101	M?
77	N(13)	11111	01011	10100	M?
78	M(13)	11100	01011	10111	M
79	M(14)	11111	11001	00110	M

Table 4.2: Infering ψ Wheels (continued)

j	**MI(j)**	$\underline{\chi}$	$\underline{\psi}$	$\underline{K}$	**M?**
80	M(15)	11101	01001	10100	M
81	M(16)	11101	11100	00001	M
82	M(17)	11101	11001	00100	M
83	M(18)	01101	10100	11001	M
84	M(19)	01101	11101	10000	M
85	M(20)	11111	11111	00000	M
86	M(21)	01111	10111	11000	M
87	M(22)	01111	11111	10000	M
88	N(23)	11011	10111	01100	M?
89	M(23)	01011	10111	11100	M
90	M(24)	11101	11110	00011	M?
91	M(25)	11101	11110	00011	M
92	M(26)	11111	11111	00000	M
93	M(27)	10110	11100	01010	M
94	M(28)	11110	11111	00001	M?
95	M(29)	10110	11111	01001	M
96	M(30)	11011	11110	00101	M
97	M(31)	10111	11111	01000	M?
98	M(32)	11110	11111	00001	M
99	M(33)	10110	11110	01000	M?
100	M(34)	10111	11110	01001	M
101	M(35)	11110	11111	00001	M?
102	M(36)	10111	11111	01000	M?
103	N(37)	11111	11111	00000	M?
104	M(37)	11101	11111	00010	M?
105	N(38)	11111	11111	00000	M?
106	N(38)	11111	11111	00000	M?
107	M(38)	11101	11111	00010	M?
108	M(39)	11101	11111	00010	M?
109	M(40)	11101	11111	00010	M?
110	N(41)	11101	11111	00010	M?
111	M(41)	11111	11111	00000	M?
112	M(42)	11111	11111	00000	M
113	N(0)	11111	01111	10000	M?
114	M(0)	11111	01111	10000	M?
115	N(1)	11111	01111	10000	M?
116	M(1)	11100	01111	10011	M
117	M(2)	11000	11111	00111	M
118	M(3)	11110	01111	10001	M?
119	M(4)	11111	01111	10000	M

Table 4.3: Infering ψ Wheels (continued)

j	**MI(j)**	$\underline{\chi}$	$\underline{\psi}$	$\underline{K}$	**M?**
120	M(5)	11111	11111	00000	M?
121	N(6)	11110	11111	00001	M?
122	M(6)	11110	11111	00001	M?
123	M(7)	11111	11111	00000	M
124	M(8)	00110	00111	00001	M
125	M(9)	01011	11111	10100	M?
126	M(10)	10111	11111	01000	M
127	M(11)	01111	01111	00000	M
128	M(12)	00111	10111	10000	M
129	M(13)	11111	00111	11000	M
130	M(14)	00101	11111	11010	M
131	N(15)	10111	01111	11000	M?
132	M(15)	11111	01111	10000	M
133	M(16)	10101	11111	01010	M
134	M(17)	11101	11011	00110	M?
135	M(18)	11101	11011	00110	M?
136	M(19)	11101	11011	00110	M
137	M(20)	11111	11111	00000	M
138	M(21)	11111	11001	00110	M
139	M(22)	11110	10001	01111	M
140	M(23)	11110	11011	00101	M
141	M(24)	11110	11101	00011	M
142	M(25)	11101	10001	01100	M
143	M(26)	11101	11101	00000	M
144	M(27)	11110	10101	01011	M
145	M(28)	11110	11101	00011	M?
146	M(29)	11011	11101	00110	M
147	M(30)	11110	11111	00001	M?
148	M(31)	11111	11111	00000	M
149	M(32)	11111	11110	00001	M
150	M(33)	11111	11111	00000	M
151	N(34)	11111	11110	00001	M?
152	M(34)	11111	11110	00001	M
153	N(35)	11111	11111	00000	M?
154	M(35)	11011	11111	00100	M?
155	M(36)	10111	11111	01000	M
156	M(37)	11101	11101	00000	M
157	N(38)	10111	11111	01000	M?
158	M(38)	11111	11111	00000	M?
159	N(39)	10101	11111	01010	M?

Table 4.4: Infering ψ Wheels (continued)

j	**MI(j)**	$\underline{\chi}$	$\underline{\psi}$	$\underline{K}$	**M?**
160	M(39)	11101	11111	00010	M
161	N(40)	10101	11110	01011	M?
162	N(40)	10100	11110	01010	M?
163	M(40)	11110	11110	00000	M?
164	M(41)	10110	11110	01000	M
165	M(42)	01111	11111	10000	M
166	M(0)	01111	01110	00001	M
167	M(1)	11110	01111	10001	M
168	N(2)	01100	11111	10011	M?
169	M(2)	01101	11111	10010	M
170	M(3)	11110	01110	10000	M
171	M(4)	01111	01111	00000	M
172	M(5)	11111	11111	00000	M
173	M(6)	11111	11110	00001	M?
174	M(7)	11111	11110	00001	M
175	M(8)	11011	01111	10100	M
176	M(9)	11111	11111	00000	M?
177	N(10)	11111	11111	00000	M?
178	M(10)	11111	11111	00000	M
179	M(11)	11111	01111	10000	M
180	M(12)	11111	10111	01000	M
181	M(13)	11111	01111	10000	M
182	M(14)	11101	11111	00010	M
183	M(15)	11011	01111	10100	M
184	M(16)	11111	10111	01000	M?
185	M(17)	11100	10111	01011	M
186	M(18)	10100	11111	01011	M?
187	M(19)	11100	11111	00011	M?
188	M(20)	10101	11111	01010	M?
189	M(21)	11111	11111	00000	M?
190	M(22)	10110	11111	01001	M?
191	M(23)	11110	11111	00001	M?
192	M(24)	10111	11111	01000	M
193	M(25)	10110	11011	01101	M
194	M(26)	11101	10011	01110	M
195	N(27)	10101	11011	01110	M?
196	M(27)	11111	11011	00100	M
197	N(28)	11111	11111	00000	M?
198	N(28)	11111	11111	00000	M?
199	N(28)	11111	11111	00000	M?

Table 4.5: Infering ψ Wheels (continued)

4.4.3 Step 3 : Inference of the $\psi(j)$ Pin-Wheel Values

Whenever the inferred move indicator is M, a value of $\psi_i(Q_i[j])$ is determined. The j^{th}-M-block $\mathcal{B}_j$

- starts when the inferred move indicator is equal to M and

- ends when the inferred indicator is equal to M?

We list the blocks and ψ-values in the next three tables (Table 4.6 t0 4.8).

M-Blocks			
j	$\mathbf{P}_j$	$\mathbf{L_j}$	$\mathcal{B}_j$
1	0	7	0 0 1 0 0 1 1
2	7	4	1 0 1 1
3	12	2	1 0
4	14	6	0 1 0 1 0 1
5	20	2	1 1
6	22	2	1 1
7	27	3	1 1 1
8	54	2	1 0
9	58	2	0 1
10	60	4	1 0 0 1
11	67	2	1 0
12	70	6	0 1 1 0 1 0
13	78	11	0 1 0 1 1 1 1 1 1 1 1
14	89	2	1 1
15	91	4	1 1 1 1
16	95	3	1 1 1
17	98	2	1 1
18	100	2	1 1
19	112	2	1 0
20	116	3	0 1 0
21	119	2	0 1
22	123	3	1 0 1
23	126	6	1 0 1 0 1 0
24	132	3	0 1 1

Table 4.6: M-Blocks

M-Blocks			
j	$\mathbf{P_j}$	$\mathbf{L_j}$	$\mathcal{B}_j$
25	136	10	1 1 1 1 1 1 1 1 1 1
26	146	2	1 1
27	148	4	1 1 1 1
28	152	2	1 1
29	155	3	1 1 1
30	160	2	1 1
31	164	5	1 1 0 0 1
32	169	5	1 0 0 1 1
33	174	3	1 0 1
34	178	7	1 0 1 0 1 0 1
35	185	2	1 1
36	192	4	1 1 1 1
37	196	2	1 1
38	200	4	1 1 1 1
39	204	2	1 1
40	206	3	1 1 1
41	214	2	1 1
42	218	2	1 0
43	220	5	0 1 0 0 1
44	225	5	1 1 1 0 1
45	230	8	1 0 1 0 1 0 1 1
46	238	3	1 1 1
47	242	2	1 1
48	245	2	1 1
49	255	3	1 1 1

Table 4.7: M-Blocks (continued)

M-Blocks			
j	$\mathbf{P_j}$	$\mathbf{L_j}$	$\mathcal{B}_j$
50	258	4	1 1 1 1
51	262	3	1 1 1
52	266	7	1 1 1 1 0 0 1
53	273	2	1 0
54	275	2	0 1
55	277	4	1 1 0 1
56	281	7	1 0 1 0 1 0 1
57	292	3	1 1 1
58	296	2	1 1
59	298	2	1 1
60	300	2	1 1
61	302	2	1 1
62	306	2	1 1
63	308	8	1 1 1 1 1 1 1 1
64	317	5	1 1 1 1 1
65	322	5	1 0 0 1 0
66	327	9	0 1 1 1 0 1 1 0 1
67	336	2	1 0
68	338	4	0 1 0 1
69	342	2	1 1
70	351	2	1 1
71	353	2	1 1
72	356	2	1 1
73	358	2	1 1
74	367	13	1 1 1 1 1 1 1 0 0 1 0 0 1

Table 4.8: M-Blocks (continued)

To carry out the inference of the $\psi_1(Q_1[j])$ pin-wheel values, the results in the previous tables are placed in a more revealing tabular format; In the tables that follow:

1. the first row lists the blocks $\mathcal{B}_0, \mathcal{B}_1, \cdots$ separated by a ?;
2. the starting position P_j of the j^{th}-block $\mathcal{B}_j$ is in the second row;
3. the length L_j of the j^{th}-block $\mathcal{B}_j$ is in the third row;
4. the bound $M_{(j,j+1)} \equiv P_{j+1} + L_{j+1} - (P_j + L_j - 1)$ in row 4.

Note that

$m_{(j,j+1)} \equiv Q[P_{j+1} + L_{j+1}] - Q[P_j + L_j + 1] \leq M_{(j,j+1)}$

0	0	1	0	0	1	1	?	1	0	1	1	?	1	0	?	0	1	0	1	0	1	?	1	1	?	1	1
0								7					12			14							20			22	
7								4					2			6							2			2	
							0					1			0							0			0		

The unknown values ?

- between blocks $\mathcal{B}_0$ and $\mathcal{B}_1$
- between blocks $\mathcal{B}_1$ and $\mathcal{B}_2$

gives rise to the following ways of concatenating these blocks:

$m_{(0,1)} = -1: \quad \underbrace{001001\mathrlap{\overbrace{1011}^{\mathcal{B}_1}}}_{\mathcal{B}_0}$

$$m_{(0,1)} = 0 : \quad \underbrace{0010011}_{\mathcal{B}_0}\overbrace{1011}^{\mathcal{B}_1}$$

$$m_{(1,2)} = -1 : \quad \underbrace{101\rlap{$\overbrace{}^{\mathcal{B}_2}$}1}_{\mathcal{B}_2}0$$

$$m_{(1,2)} = 0 : \quad \underbrace{1011}_{\mathcal{B}_1}\overbrace{10}^{\mathcal{B}_2}$$

$$m_{(1,2)} = 1 : \quad \underbrace{1011}_{\mathcal{B}_1}?\overbrace{10}^{\mathcal{B}_2} \qquad ? \in \{0,1\}$$

```
?      1   1   1     ?    1   0   ?     0   1   ?     1   0   0   1   ?     1   0   ?      0   1   1   0   1   0    ?
    27                  54            58            60                    67           70
      3                   2            2              4                    2             6
3                 24              2             0                     3             1                              2
```

```
   0   1   0   1   1   1   1   1   1   1   1   ?     1   1   ?     1   1   1   1   ?     1   1   1   ?      1   1   ?
78                                                 89           91                    95                98
11                                                  2             4                     3                 2
                                               0                  0           0                     0              0
```

```
    1   1     ?       1   0   ?   0       1   0   ?      0   1   ?       1   0   1   ?        1   0   1   0   1   0   ?
100               112                116             119             123                126
    2                2                   3              2               3                   6
            10               2                   0              2                   0                                0
```

```
    0   1   0   ?       1   1   1   1   1   1   1   1   1   1   ?      1   1   ?       1   1   1   1   ?        1   1   ?
132                 136                                             146             148                     152
    3                 10                                                2              4                       2
                1                                               0               0                       0               1
```

```
    1   1   1   1   ?      1   1   ?       1   1   0   0   1   ?       1   0   0   1   1   ?        1   0   1   ?
155                     160            164                         169                         174
    4                      2               5                           5                           3
                    1              2                           0                           0                    1
```

1	0	1	0	1	0	1	?	1	1	?	1	1	1	1	?
178								185			192				
7								2			4				
							0			5					

4.4.4 M-Block Concatenation : Finding $\psi(j)$

The problem is to concatenate the M-blocks and by doing so to identify the unknown ? $\psi_1(Q_1[j])$ values.

A program tests all possible values of the unknown bits ?? in an attempt to find the best match between pairs of blocks. The matching program yields the following results:

$\underline{\text{0-54}}$: $\underbrace{00\,\overbrace{100111011010101111111111111111111111}\,11111}$

$\underline{\text{54-115}}$: $1\underbrace{0010011101101010111111111111111111111111111}001$

$\underline{\text{115-150}}$: $\overbrace{10011101101010111111111111111111111111}$

From these results, the values of $\psi_1(j)$ can be seen.

4.4.5 Step 4 : Inference of $\pi(j)$

When the ψ_1 pin-wheel is determined, the values in the move indicator column in Tables 4.1 to 4.5 are known and as well as $\underline{\text{all}}$ $\psi_i(j)$ pin-wheels values.

To infer the $\pi(j)$ pin-wheel values, we begin by observing

$\mu(U[j]) = 1 \Longrightarrow MI(j) = \mathrm{M}$

$\mu(U[j]) = 0 \Longrightarrow MI(j) = \mathrm{N}$

$U[j] = (U[j-1] + \pi(V[j])$ $(\mathit{modulo}\ 37)$

Thus $MI(j) = M$ and $MI(j+1) = N \Longrightarrow \pi(V[j]) = 1$

$MI(j) = N$ and $MI(j+1) = M \Longrightarrow \pi(V[j]) = 1$

The values of j for which $1 = \pi(j\ (\mathit{modulo}\ 41))$ inferred by this algorithm with $0 \leq j \leq 65$ are given in Table 4.9.

Continuing this process a sufficient number of steps will reveal *all* values of j with $0 \leq j < 61$ for which $\pi(j) = 1$; the remaining values of $\pi(j)$ are 0. If $\pi(j)$ is <u>in</u>correctly inferred, an <u>in</u>consistency will result.

j	**MI(j)**	**MI(j+1)**	**j** (**mod 61**)	π(**j** (**mod 61**))
9	M	N	9	1
11	N	M	11	1
12	M	N	12	1
13	N	M	13	1
18	M	N	18	1
19	N	M	19	1
20	M	N	20	1
21	N	M	21	1
22	M	N	22	1
26	N	M	26	1
30	M	N	30	1
31	N	M	31	1
38	M	N	38	1
40	N	M	40	1
54	M	N	54	1
57	M	N	57	1
58	M	N	58	1
59	M	N	59	1
60	N	M	60	1
62	M	N	1	1
63	M	N	2	1
65	M	N	4	1

Table 4.9: Infering π Wheel

4.4.6 Step 5 : Inference of $\mu(j)$

We again start with the idea leading to $\pi(j)$; with complete (?) knowledge of $\pi(j)$, inferences of the values of $\mu(U[j])$ and $U[j]$ may be made.

$$\mu(U[j]) = 1 \Longrightarrow MI(j) = \mathrm{M}$$

$$\mu(U[j]) = 0 \Longrightarrow MI(j) = \mathrm{N}$$

$$U[j] = (U[j-1] + \pi(V[j]) \; (modulo\, 37)$$

This leads to the $\mu(j)$-inference rules in which $U[j+1] = (U[j] + S[j]) \; (modulo\, 37)$:

MI(j)	**MI(j+1)**	$\pi(V[(j])$	$\mu(U[j])$	$\mu(U[j+1])$	**S(j)**
M	N	1	1	0	1
M	N	0	Impossible		
M	M	1	1	1	1
M	M	0	1	1	0
N	M	1	0	1	1
N	M	0	Impossible		
N	N	1	0	0	1
N	N	0	0	0	0

Table 4.10: Infering μ Wheel

5 Conclusion

We have presented a method to solve the cryptanalysis problem #2 for Schlüsselzusatz; given the key, find the active pins on all the wheels. The key can be obtained using cribbing as described. The method can be used to decrypt all the messages generated using the same key.

The method is based on making good guesses for the values of the model parameters q and ν. This involves prior knowledge of the type of messages usually sent by the sender. Also, the method is supposed to work well when $\delta_(i, j)$ is much higher for one pair of (i,j) than the rest. Moreover, to get the pin wheel settings on all the wheels, a sufficiently long key is required otherwise we will have inconsistencies in our solution. As stated before, the problem #2 doesn't have a unique solution. The method gives one of the solutions; the other solution can be similarly found by complementing the values of ψ and χ wheels.

References

[KAH67] David Kahn, "The Codebreakers", MacMillan, 1967.

[SHA49] Claude E. Shannon, "Communication Theory of Secrecy Systems", *Bell Systems Technical Journal*, pp. 28, pp. 656-715, 1949.

[HiSt01] F. H. Hinsley and Alan Stripp, "Code Breakers", Oxford University Press, 2001.

[TUC70] Bryant Tuckerman, "A Study of the Vigenére-Vernam Single and Multiple Loop Enciphering Systems", IBM Research Report RC 2879, May 14, 1970.

[TUT98] William F. Tutte, "The Fish and I", University of Waterloo lecture, June 18, 1998.

[SAH01] Tony Sale, "Rebuilding of Heath Robinson", Bletchley Park, June 2001.

[AND01] Scott L. Andresen, "Donald Michie: Secrets of Collossus revealed", IEEE Intelligent Systems, November 2001.

[SAT01] Tony Sale, "Part of the General Report on Tunny", March 2001.

[SAF01] Tony Sale, "The Special Fish Report by Albert W. Small (December 1944) reformatted by Tony Sale", March 2001.

[CAR97] F. L. Carter,"Cryptography and Coding", the proceedings of the 6th IMA International Conference, December 1997