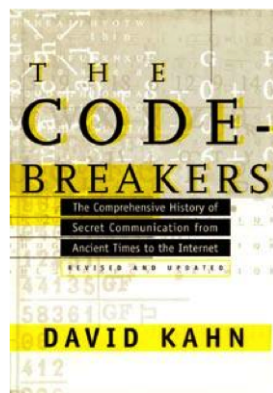


## 2 Historical Ciphers

### Historical Ciphers

- [Shift Cipher](#)
- [Substitution Cipher](#)
- [Vigenère Cipher](#)
- [Vernam Cipher](#)
- [Rotor Machines](#)



### 2.1 Shift Cipher

#### Shift Cipher

Identify each letter with a number

- $A = 0$
- $B = 1$
- $C = 2$
- $\vdots$
- $Z = 25$

The key  $k$  is a number in the range 0-25

- Encryption: add  $k$  to each letter (modulo 26).

Julius Caesar used the key  $k = 3$ .

- **ATTACK AT DAWN** becomes **DWWDFN DW GDZQ**

We break a shift cipher by using the [statistics](#) of the underlying language.

Geoff Hamilton

### English Letter Frequencies

A	8.05	J	0.10	S	6.59
B	1.62	K	0.52	T	9.59
C	3.20	L	4.03	U	3.10
D	3.65	M	2.25	V	0.93
E	12.31	N	7.19	W	2.03
F	2.28	O	7.94	X	0.20
G	1.61	P	2.29	Y	1.88
H	5.14	Q	0.20	Z	0.09
I	7.18	R	6.03		

Most common bigrams: TH,HE,IN,ER,AN,RE,ED,ON,ES,ST,EN,AT,TO,NT,HA

Most common trigrams: THE,ING,AND,HER,ERE,ENT,THA,NTH,WAS,ETH,FOR

### Shift Cipher

Take the following example cipher text:

BPMZM WVKM EIA IV COTG LCKSTQVO EQBP NMIBPMZA ITT ABCJJG IVL  
JZWEV IVL BPM WBPMZ JQZLA AIQL QV AW UIVG EWZLA OMB WCB WN  
BWEV OMB WCB, OMB WCB, OMB WCB WN BWEV IVL PM EMVB EQBP  
I YCIKS IVL I EILLTM IVL I YCIKS QV I NTCZZG WN MQLMZLWEV BPIB  
XWWZ TQBBTM COTG LCKSTQVO EMVB EIVLMZQVO NIZ IVL VMIZ JCB  
IB MDMZG XTIKM BPMG AIQL BW PQA NIKM VWE OMB WCB, OMB WCB,  
OMB WCB WN PMZM IVL PM EMVB EQBP I YCIKS IVL I EILLTM IVL I YCIKS  
IVL I DMZG CVPIXXG BMIZ

We need to compare the frequency distribution of this text with standard English

### Letter Frequencies

A	2.59	J	1.44	S	1.73
B	10.37	K	2.59	T	3.46
C	5.48	L	6.63	U	0.29
D	0.58	M	10.09	V	8.36
E	4.61	N	2.31	W	6.63
F	0.00	O	3.46	X	1.15
G	2.59	P	4.03	Y	1.15
H	0.00	Q	4.03	Z	4.90
I	11.53	R	0.00		

### Cracking the Cipher

The shift of **E** seems to be either **4**, **8**, **17**, **18** or **23**

The shift of **A** seems to be either **1**, **8**, **12**, **21** or **22**

Hence the key is probably equal to **8**

We can now decrypt the cipher text to reveal:

THERE ONCE WAS AN UGLY DUCKLING WITH FEATHERS ALL STUBBY AND BROWN AND THE OTHER BIRDS SAID IN SO MANY WORDS GET OUT OF TOWN GET OUT, GET OUT, GET OUT OF TOWN AND HE WENT WITH A QUACK AND A WADDLE AND A QUACK IN A FLURRY OF EIDERDOWN THAT POOR LITTLE UGLY DUCKLING WENT WANDERING FAR AND NEAR BUT AT EVERY PLACE THEY SAID TO HIS FACE NOW GET OUT, GET OUT, GET OUT OF HERE AND HE WENT WITH A QUACK AND A WADDLE AND A QUACK AND A VERY UNHAPPY TEAR

## 2.2 Substitution Cipher

### Substitution Cipher

The problem with the shift cipher is that the number of keys is too [small](#).

- We only have 26 possible keys

To increase the number of keys a [substitution cipher](#) was invented.

[Encryption](#) involves replacing each letter by its permuted version.

[Decryption](#) involves use of the inverse permutation.

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

Hence [ATTACK AT DAWN](#) encrypts to [TXXTKE TX GTHA](#)

### Substitution Cipher

Number of keys is  $26! \approx 4.03 \times 10^{26} \approx 2^{88}$

- Feasible to only run a computer on a problem which takes under  $2^{80}$  steps.

This is far too [large](#) a number to brute force search using modern computers.

Still we can break these ciphers using [statistics](#) of the underlying plaintext language.

### Example

XSO MJIWXVL JODIVA STW VAO VY OZJVCO'W LTJDOWX KVAKOAXJTXI-  
VAW VY SIDS XOKSAVLVDQ IAGZWXJQ. KVUCZXOJW, KVUUZAIKTXIVAW  
TAG UIKJVOLOKXJVAIKW TJO HOLL JOCJOWOAXOG, TLVADWIGO GIDIXTL  
UOGIT, KVUCZXOJ DTUOW TAG OLOKXJVAIK KVUUOJKO. TW HOLL TW  
SVWXIAD UTAQ JOWOTJKS TAG CJVGZKX GONOLVCUOAX KOAXJOW VY  
UTPVJ DLVMTL KVUCTAIOW, XSO JODIVA STW T JTCIGLQ DJVHIAD AZU-  
MOJ VY IAAVNTXINO AOH KVUCTAIOW. XSO KVUCZXOJ WKIOAKO GOC-  
TJXUOAX STW KLVWO JOLTXIVAWSICW HIXS UTAQ VY XSOWO VJDTAI-  
WTXIVAW NIT KVLLTMVJTXINO CJVPOKXW, WXTYY WOKVAGUOAXW TAG  
NIWIXIAD IAGZWXJITL WXTYY. IX STW JOKOAXLQ IAXJVGZKOG WONO-  
JTL UOKSTAIWUW YVJ GONOLVCIAID TAG WZCCVJXIAD OAXJOCJOAOZJITL  
WXZGOAXW TAG WXTYY, TAG TIUW XV CLTQ T WIDAIYIKTAX JVLO IA  
XSO GONOLVCUOAX VY SIDS-XOKSAVLVDQ IAGZWXJQ IA XSO JODIVA.

**English Letter Frequencies**

A	8.99	J	6.51	S	3.26
B	0.00	K	4.81	T	7.60
C	2.95	L	4.34	U	3.57
D	3.10	M	0.62	V	8.06
E	0.00	N	1.40	W	7.13
F	0.00	O	11.63	X	7.75
G	3.72	P	0.31	Y	2.17
H	0.78	Q	1.40	Z	2.17
I	7.75	R	0.00		

Most common bigrams: TA,AX,IA,VA,WX,XS,AG,OA,JO,JV

Most common trigrams: OAX,TAG,IVA,XSO,KVU,TXI,UOA,AXS

**Analysis**

Since O occurs with frequency 11.63 we can guess O = E

Common trigrams are:

- OAX = E\*\*
- XSO = \*\*E

Common similar trigrams in English are:

- ENT, ETH
- THE

Hence likely to have:

- X = T
- S = H
- A = N

**Analysis**

From now on we only look at the first two sentences:

THE MJIVTVL JEDIVN HTW VNE VY EZJVCE'W LTJDEWT KVNKENTJTTIV  
 NW VY HIDH TEKHNVLVDQ INGZWTJQ. KVUCZTEJW, KVVU ZNIKTIVNW  
 TNG UIKJVELEKTJVNIKW TJE HELL JECJEWENTEG, TLVNDWIGE GIDITTL  
 UEGIT, KVUCZTEJ DTUEW TNG ELEKTJVNIK KVVUEJKE.

This was after the changes:

- O = E
- X = T
- S = H
- A = N

**Analysis**

Since T occurs as a single letter we must have:

- $T = I$ , or
- $T = A$

T has probability of 8.07, which is the highest probability left  
Therefore, more likely to have:

- $T = A$

The most frequent

- bigram is  $TA = AN$
- trigram is  $TAG = AN^*$

Therefore highly likely that:

- $G = D$

**Analysis**

THE MJIWTVL JEDIVN HAW VNE VY EZJVCE'W LAJDEWT KVNKENTJATIV  
NW VY HIDH TEKHNVLVDQ INDZWTJQ. KVUCZTEJW, KUUZNIKATIVNW  
AND UIKJVELEKTJVNIKW AJE HELL JECJEWENTED, ALVNDWIDE DIDITAL  
UEDIA, KVUCZTEJ DAUEW AND ELEKTJVNIK KUUUEJKE.

This was after the additional changes:

- $T = A$
- $G = D$

**Analysis**

We now look at two letter words:

- $IX = *T$
- Therefore  $I$  must be one of  $A, I$  due to English plaintext
- Already have  $A$

Hence:

- $I = I$
- $XV = T^*$
- Must have, due to English,  $V = O$

**Analysis**

More two letter words:

- $VY = O^*$
- Hence  $Y$  must be one of  $F, N, R$  due to English
- Already have  $N$
- $Y$  has probability 1.61
- $F$  has probability 2.28
- $R$  has probability 6.03

Hence  $Y = F$

We also have:

- $IW = I^*$
- Therefore  $W$  must be one of  $F, N, S, T$
- Already have  $F, N, T$

Hence  $W = S$

**Analysis**

THE MJISTOL JEDION HAS ONE OF EZJOCE'S LAJDEST KONKENTJATIONS  
OF HIDH TEKHNOLODQ INDZSTJQ. KOUCZTEJS, KOUUZNIKATIONS AND  
UIKJOELEKTJONIKS AJE HELL JECJESANTED, ALONDSIDE DIDITAL UEDIA,  
KOUCZTEJ DAUES AND ELEKTJONIK KOUUEJKE.

This was after the additional changes:

- $I = I$
- $V = O$
- $Y = F$
- $W = S$

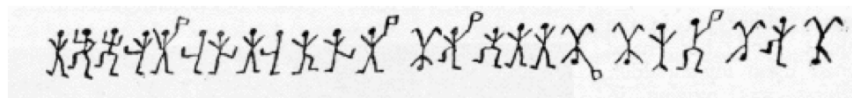
It is then easy to see what the underlying plaintext is.

**Example Substitution Ciphers**

Edgar Allan Poe's "The Gold Bug":

53++!305))6\*;4826)4+.)4+);806\*;48!8'60))85;]8\*;;+\*8!83(88)5\*!; 46(;88\*96\*?;8)\*+(;485);5\*!2:\*+(;4956\*2(5\*-  
4)8'8\*;4069285);)6 !8)4++;1(+9;48081;8:8+1;48!85;4)485!528806\*81(+9;48;(88;4(+?3  
4;48)4+;161;;188;+?;

Sir Arthur Conan Doyle's "Adventure of the Dancing Men":



## 2.3 Vigenère Cipher

### Vigenère Cipher

The problem with the shift and substitution ciphers was that each plaintext letter always encrypted to the **same** ciphertext letter.

Hence underlying **statistics** of the language could be used to break the cipher.

From the early 1800s onwards cipher designers tried to **break** this link between the plain and cipher texts.

The most famous cipher used during the 1800s is the **Vigenère** Cipher

- Invented in 1553 by Giovan Batista Belaso.
- Believed to be **unbreakable** for a number of years.
- Actually **easy** to break.

### Vigenère Cipher

The **Vigenère cipher** again identifies letters with the numbers 0-25

The **secret key** is a short sequence of letters (e.g. a word).

Encryption involves **adding** the plaintext letter to a key letter, with the key letters used in rotation.

Thus if the key is **SESAME**, encryption works as follows:

T	H	I	S	I	S	A	T	E	S	T	M	E	S	S	A	G	E	Message
S	E	S	A	M	E	S	E	S	A	M	E	S	E	S	A	M	E	Keystream
L	L	A	S	U	W	S	X	W	S	F	Q	W	W	K	A	S	I	Ciphertext

This is a **polyalphabetic substitution cipher**

- **A** will encrypt to a different letter depending on where it is in the message.

### Vigenère Cipher

Ciphertext:

```

CI UT WFCN LTFE VF AAHGKEE DNH VYC IPSKGT MV EVLINF A, NC HXS SLGIX QNVY GEM VYYI MVG ZLUHFORRXHB-RIMRXGUZLV TBF KCAXQQD-
KJGWERRXHB U ICKHZWKG DGG PFU JGRGIUPR KKCJ RHBVZLJX JKXMGHIUCW XGHQ KFT MKGERN-YWTJR. LX WPKCGTQV RLS MFCEQPVH DP
BXKSEKGCZ TNFAZL CH UGVBHCC NPVYGKQ IHKCIBH XO EY MIAS T KFGHIY ANUSTJNPVS, ERPGRWXP JDOS PFRTL, RKXGITZ ERQW, TBF JCRKSV
TMGICTRTT WCELKTGHU. FSG ISTJMTZ CEB TVCPFKXV ZKMCH KSNP KDKS CEB BHFG FL DNF CSGABHA KM AXH ULAW XHJVPTTZ ERPG-
BST GGVXCPJ KTWWC KC PM O FZQITBEV UWTH YV SHXR VF BD PWVY DPVS-VF-DPVS OVCIBIJ, NPIST UMRNAGERH, TBF R DXKA JRLSLVCBC.
JGTQIRJGOVJN, MVG KCRABKTYA PWBRPSKM GEYQEWXP PTF CVV ADEZCSMGTHKFLH BG HFSCWSF FL QKCCUAPLHKEE TOSTPRWBBI RQ HX-
EVLRLXG QW XTKCU RLS HBGJ RWTH QECH HKP UMV PCWCBCM FGT MVGWBY. UWTH KJ RD WWUKGCZIKJF P WWIZRPE RQCJPK KJVL XM
WU RQ TTGKCW GXDTFBJVWDCC PL HJV QEHYGE UDKR? JFU SH KG TMCOSTJC EKWXRRTEM YYCC XJGIW HRZNRZAX WU SMJGQGU MUY O
URRTEZKKC PGR UDCPKSF FTTK OP VLIBFG TCMWVPVLI?

```

### Vigenère Cipher

First we need to look for **repeated** sequences of characters.

- Recall English has a large repetition of certain **bigrams**
- These are likely to match up to the **same** two letters in the key every so often

- By looking for the **distance** between two repeated sequences we can guess the length of the keyword.

Each distance should be a **multiple** of the keyword.

- Taking the **gcd** of all distances between sequences should give the keyword length.

### Vigenère Cipher

First sentence is:

CJ UT WFCN LTTF VF AAHGKEE DNH VYC IPSP**KG**TMV EVLINFA, NC HXS  
SLGIX QNVY GEM VYYI MVG ZLUHFO**RR**XHB-RIMRXGUZLV TBF KCAXQQD-  
KJGWER**RR**XHBU ICKHZW**KG**DGG PFU JGRGIUPR KKCJ RHBVZLJX JKXMGHI-  
UCW XGHQ KFT **MKG**ERN-YWTJR.

Distance between occurrences of **RR**: 30

Distance between occurrences of **KG**: 96, 46

We have:

- $\gcd(30,96)=6$ ,  $\gcd(30,46)=2$ ,  $\gcd(96,46)=2$

Unlikely to have a keyword of length **2**

- Guess keyword is of length **6**

### Vigenère Cipher

Now we take every **sixth** letter and look at the statistics just as we did for a shift cipher to deduce the first letter of the keyword.

This gives us the following:

A	1.49	J	3.73	S	0.75
B	1.49	K	8.96	T	7.46
C	8.96	L	0.00	U	8.21
D	1.49	M	0.00	V	8.21
E	6.72	N	2.99	W	2.24
F	4.48	O	1.49	X	0.75
G	11.19	P	8.21	Y	1.49
H	1.49	Q	4.48	Z	0.00
I	2.99	R	0.75		

We look for a low value and then three high ones, corresponding to **Q, R, S, T**  $\Rightarrow$  Key is **2** or **C**

### Vigenère Cipher

Then we take every **sixth** letter starting from the **second** one and repeat to find the second letter of the keyword:



A	0.00	J	8.21	S	2.24
B	0.75	K	9.70	T	3.73
C	5.22	L	2.24	U	3.73
D	1.49	M	0.75	V	10.44
E	7.46	N	1.49	W	0.75
F	11.19	O	0.00	X	2.99
G	0.75	P	2.24	Y	4.48
H	0.00	Q	0.00	Z	3.73
I	4.48	R	11.94		

We look for a low value and then three high ones, corresponding to **Q, R, S, T**  $\Rightarrow$  Key is **17** or **R**

And so on to determine the six letters of the keyword: **CRYPTO**

### Vigenère Cipher

The underlying plaintext is then found to be:

AS WE DRAW NEAR TO CLOSING OUT THE TWENTIETH CENTURY, WE SEE QUITE CLEARLY THAT THE INFORMATION-PROCESSING AND TELECOMMUNICATIONS REVOLUTIONS NOW UNDERWAY WILL CONTINUE VIGOROUSLY INTO THE TWENTY-FIRST. WE INTERACT AND TRANSACT BY DIRECTING FLOCKS OF DIGITAL PACKETS TOWARDS EACH OTHER THROUGH CYBERSPACE, CARRYING LOVE NOTES, DIGITAL CASH, AND SECRET CORPORATE DOCUMENTS. OUR PERSONAL AND ECONOMIC LIVES RELY MORE AND MORE ON OUR ABILITY TO LET SUCH ETHEREAL CARRIER PIGEONS MEDIATE AT A DISTANCE WHAT WE USED TO DO WITH FACE -TO-FACE MEETINGS, PAPER DOCUMENTS, AND A FIRM HANDSHAKE. UNFORTUNATELY, THE TECHNICAL WIZARDRY ENABLING REMOTE COLLABORATIONS IS FOUNDED ON BROADCASTING EVERYTHING AS SEQUENCES OF ZEROS AND ONES THAT ONES OWN DOG WOULDNT RECOGNIZE. WHAT IS TO DISTINGUISH A DIGITAL DOLLAR WHEN IT IS A S EASILY REPRODUCIBLE AS THE SPOKEN WORD? HOW DO WE CONVERSE PRIVATELY WHEN EVERY SYLLABLE IS BOUNCED OFF A SATELLITE AND SMEARED OVER AN ENTIRE CONTINENT?

## 2.4 Vernam Cipher

### Vernam Cipher (One-Time Pad)

- Gilbert Vernam patented this cipher in 1917 for encryption and decryption of telegraph messages.
- Used extensively during the first world war.
- To send a binary string  $m$  you need a key  $k$  as long as the message.
- Each key  $k$  is uniformly selected at random and can be used only once - hence **one-time pad**.
- Encryption:  $c = m \oplus k$ .
- Decryption:  $m = c \oplus k$ .
- $\oplus$  is **exclusive-or (XOR)**:

$\oplus$	0	1
0	0	1
1	1	0

**Vernam Cipher (One-Time Pad)**

Plaintext:	o	n	e	t	i
In binary:	01101111	01101110	01100101	01110100	01101001
Key:	01011100	01010001	11100000	01101001	01111010
Ciphertext:	00110011	00111111	10000101	00011101	00010011

Plaintext:	m	e	p	a	d
In binary:	01101101	01100101	01110000	01100001	01100100
Key:	11111001	11000110	01011010	10110001	01110011
Ciphertext:	10010100	10100011	00101010	11010001	00010111

**Vernam Cipher (One-Time Pad)**

Gives perfect secrecy, provided:

- Key is truly random.
- Key is at least as big as plaintext (not practical).
- Key is not reused e.g. consider a ‘two-time’ pad:
  - First encryption:  $c_1 = m_1 \oplus k$ .
  - Second encryption:  $c_2 = m_2 \oplus k$ .
  - $c_1 \oplus c_2 = m_1 \oplus m_2$
  - Vulnerable to frequency analysis.

**2.5 Rotor Machines****Rotor Machines**

With the advent of the 1920s people saw the need for a mechanical encryption device. Taking a substitution cipher and then rotating it became seen as the ideal solution.

- This had actually been used during the American Civil War.
- But now this could be done more efficiently.

The rotors could be implemented using wires and then encryption can be done mechanically using an electrical circuit.

- By rotating the rotor we obtain a new substitution cipher.
- Transmission of message still done manually using Morse Code.

### Rotor Machines

To encrypt the first letter we use the substitutions:

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

To encrypt the second letter we use the substitutions:

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

To encrypt the third letter we use the substitutions:

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

and so on.

### Enigma

The most famous of these machines was the **Enigma**.



### Enigma

We shall describe the most simple version of Enigma.

Used three such rotors chosen from a set of five:

- **EKMFLGDQVZNTOWYHXUSPAIBRCJ** Rotor One
- **AJDKSIRUXBLHWTMCQGZNPYFVOE** Rotor Two
- **BDFHJLCPRTXVZNYEIWGAKMUSQO** Rotor Three

Geoff Hamilton

- **ESOVZJAYQUIRHXLNFTGKDCMWB** Rotor Four
- **VZBRGITYUPSDNHLXAWMJQOFECK** Rotor Five

The order of the rotors in the machine is important.

Number of ways of choosing the rotors is  $5 \times 4 \times 3 = 60$

### Enigma

Each rotor has an initial starting position

- Number of starting positions is  $26^3 = 17576$

The first rotor would move on the second rotor on each full iteration under the control of a **ring** hitting a **notch**.

- The notches were fixed for each rotor but the rings were movable.
- Similarly the second rotor would move the third rotor.
- Number of positions of the rings is  $26^2 = 676$

The second rotor also had a kick associated to it

- Cycle length of three rotors was then  $26 \times 25 \times 26 = 16900$

A given plaintext letter would encrypt to a different ciphertext letter on each press of the keyboard.

### Enigma

A **plugboard** swapped letters twice in each decryption/encryption operation.

- This increased the complexity and gave another  $10^{14}$  keys.

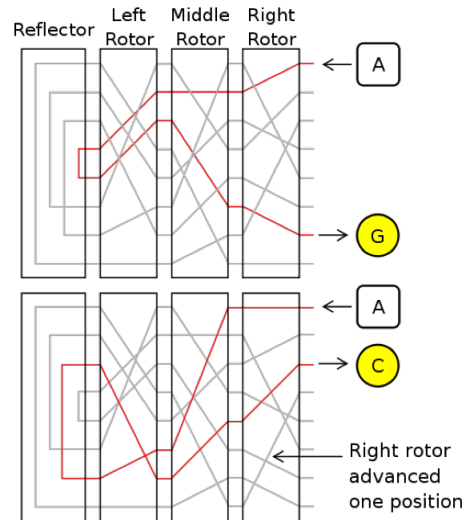
Total number of keys was then around  $10^{23}$

To make sure encryption and decryption were the same operation a **reflector** was used:

- This was a fixed substitution:

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

### Enigma



### Enigma

To use Enigma the Germans had a day setting of:

- **Plugs** to use
- **Rotors** to use, and their order
- **Ring settings**
- Initial **rotor settings**

However, we really need to **change** the key on each message.

- This is where the **flaw** in Enigma resulted.
- Not in the design, but in the **use** of the system

### Enigma

So sender of message would choose a new set of rotor positions for this message (**A**, **F**, **G** say) and then encrypt these twice using the day settings (**G**, **H**, **K**, **L**, **P**, **T** say).

He would then set the machine to the message setting and **encrypt** the message.

The receiver would **decrypt** the message key, reset his machine and then decrypt the rest of the message.

It is the repeating of the message rotor settings which led to the Polish and British cryptographers being able to **break** Enigma in WWII.

### Enigma

However, it is not only **bad usage** which led to the breaking of Enigma.

The main problem with Enigma is that the plugboard and the rotors are **orthogonal** in some sense.

- In particular the plugboard acts on the permutation given by the rotors as **conjugation**.
- This means that many of the underlying statistics of the rotor settings are **not disguised** by the plugboard
- Once the rotor settings are found the plugboard settings can be found **easily**.

### Enigma

Reflector **weakens** Enigma: no difference between encryption and decryption.

- Problem 1: encryption becomes **involutary**, i.e. if  $K \rightarrow T$ , then  $T \rightarrow K$
- Problem 2: no letter is encrypted to **itself** (electricity can't go same way back)

Heavy reduction of encryption alphabet - violation of **Kerckhoff's principle**:

- Security of Enigma depended on **wiring** of rotors
- Wiring was part of **algorithm**, not part of key
- Wiring **never changed** from 1920s until 1945

## 2.6 Summary

### Historical Ciphers

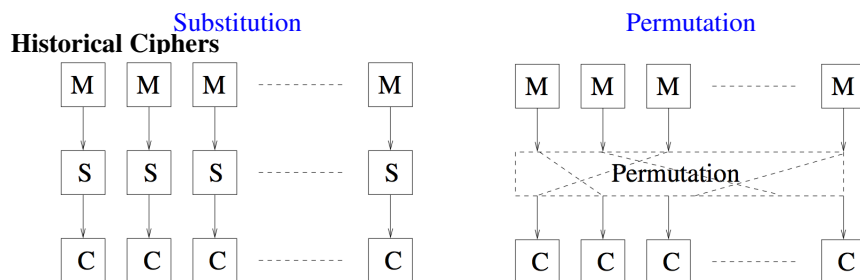
The historical ciphers we have studied can be categorised as follows:

- **monoalphabetic**: shift cipher, substitution cipher
- **polyalphabetic**: Vigenère, Enigma

Simplest ciphers could be easily broken unless encryption rule was kept **secret** (security by obscurity).

Simple ciphers could easily be broken because they did not conceal the **language characteristics** (encryption functions were not complicated enough).

Enigma was the first step towards more **mature** designs (secure and efficient).



Geoff Hamilton

- **Substitution**: provides **confusion**, i.e. it makes the relationship between key and ciphertext complex.
- **Permutation**: provides **diffusion**, i.e. each bit (symbol) of the ciphertext depends on many (if possible all) bits (symbols) of the plaintext.

### Modern Ciphers

Modern ciphers use **both** substitution and permutation.

They typically encrypt **small blocks** of plaintext at a time.

They can be **efficiently** implemented on different types of platforms.

Ciphers that apply an encryption function to small blocks of plaintext at a time are called **block ciphers**.

Ciphers that use substitution and permutation are called **substitution-permutation networks**.

## 2.7 Security

### Computational Security

A system is **computationally** secure if the **best** algorithm for breaking it requires  **$N$**  operations.

- Where  **$N$**  is a very big number
- No **practical** system can be proved secure under this definition.

In practice we say a system is computationally secure if the **best known** algorithm for breaking it requires an **unreasonably large** amount of computer time.

### Computational Security

Another practical approach is to **reduce** a well studied **hard problem** to the problem of breaking the system.

- For example, the system is secure if a given integer  **$n$**  cannot be factored.

Systems of this form are often called **provably secure**.

- However, we only have a proof **relative** to some hard problem.
- Not an **absolute** proof.

Essentially **bounding** the computational power of the adversary.

- Even if the adversary has limited (but large) resources they still will not break the system.

### Computational Security

When considering schemes which are computationally secure:

- We need to be careful about the **key sizes**.
- We need to keep abreast of current **algorithmic developments**.
- At some point in the future we should **expect** our system to be **broken** (may be many millennia hence though).

**Most** schemes in use today are computationally secure.

### Unconditional Security

For unconditional security we place **no bound on the computational power** of the adversary.

In other words, a system is unconditionally secure if it cannot be broken even with **infinite computing power**.

- Some systems are unconditionally secure.

Other names for unconditionally secure are:

- **Perfectly secure**
- **Information-theoretically secure**
- **Semantically secure**

### Unconditional Security

A cryptosystem has unconditional security iff:

$$p(P = m \mid C = c) = p(P = m)$$

for all  $m \in P$  and  $c \in C$ .

That is, the probability that the plaintext is  $m$  given that the ciphertext  $c$  is observed is the same as the probability that the plaintext is  $m$  without seeing  $c$ .

In other words knowing  $c$  reveals **no information** about  $m$ .

Unconditional security implies that:  $|keys| \geq |messages|$ .

The use of such large key spaces is **impractical** in practice.

### Key Distribution

Perfect secrecy implies length of key is at least length of plaintext.

**Key distribution** becomes the major problem.

Aim of modern cryptography is to design systems where:

- **one key** can be used **many times**
- a **short key** can encrypt a **long message**.

Such systems will not be unconditionally secure, but should be at least **computationally secure**.



**Examples**

Of the ciphers we have seen or will see later on, the following are **not** computationally secure:

- Caesar cipher
- Substitution cipher
- Vigenère cipher

The following are computationally secure but **not** unconditionally secure:

- DES(?) - AES
- RSA

The Vernam cipher (one-time pad) is unconditionally secure if used correctly.