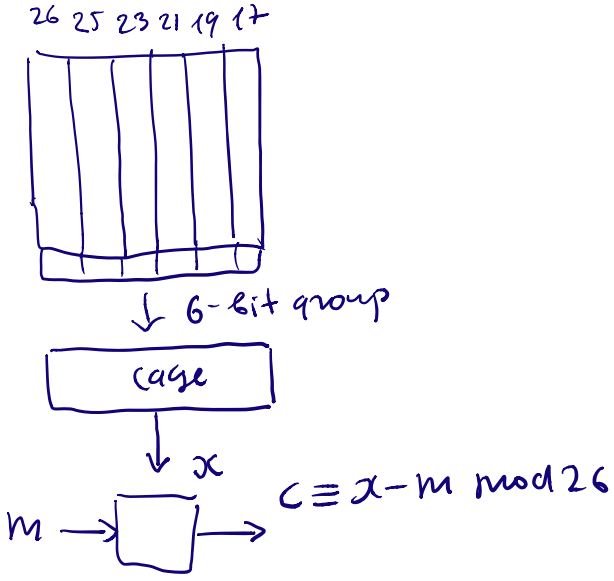
Hagelin



Example. 1) cipher-text may not be uniform.

cage

000000 100000 111111

↓ ↓ ↓

A A Z

AAA BBB ... LLL MM NN ... ZZ

12 3 times 14 ch. 2 times.

$12 \cdot 3 + 14 \cdot 2 = 64$ memory locations

distribution on the key-stream

$$X = \left(\frac{3}{64}, \frac{3}{64}, \ldots, \frac{3}{64}, \frac{2}{64}, \ldots, \frac{2}{64}\right)$$ non-uniform

compute distribution on the cipher-text

$$\left(P_Z(c=0), P_Z(c=1), \ldots, P_Z(c=25)\right) =$$

(A, B, Z)

$$= (0.039, 0.037, \ldots, 0.036, \ldots, 0.040, \ldots, 0.037)$$

$\frac{1}{26} \approx 0.038$

$$P_Z(c=j) = \sum_{i=0}^{25} P_Z(c=j, x=i) =$$

↑ cipher-text ch. ↑ key-stream ch.

$C = X - M$ pl.-text char.

$$= \sum_{i=0}^{25} P_r(X-M=j, X=i)$$

$$= \sum_{i=0}^{25} P_r(M=i-j, X=i) =$$

$$= \sum_{i=0}^{25} P_r(M=i-j) \cdot P_r(X=i) =$$

$$= \sum_{i=0}^{25} q_{i-j \bmod 26} \cdot P_r(X=i)$$

$P_r(M=i-j) = q_{i-j}$

$q_{i-j \bmod 26}$: language constants

$P_r(X=i)$: known.

compute distribution on the cipher-text

1) right most wheel position has value 0

2) ——— " ——— 1

1) key-stream distribution

$$\left(\frac{3}{32}, \frac{3}{32}, \ldots \underset{A}{\frac{3}{32}}, \underset{B}{\frac{2}{32}}, \underset{J}{0}, \ldots, \underset{K}{0}\right)$$ by cage definition

$\Rightarrow$ cipher-text distribution

$(0.042, 0.042, 0.042, 0.035, 0.033, \ldots)$

2) key-stream distribution is

$$\left(\underset{A}{0}, \ldots, \underset{J}{0}, \underset{K}{1/32}, \underset{L}{3/32}, \underset{M}{2/32}, \ldots, \underset{Z}{2/32}\right)$$

different.

$\Rightarrow$ cipher-text distribution.

$(0.035, 0.034, 0.035, 0.040, 0.041, \ldots)$

At-depth Cryptanalysis.

Germans read $\nearrow 10\%$ of M-209 cipher-texts

$X = X_1 \dots X_n$ key-stream by M-209

$M = m_1 \dots m_n$ pl.-text

$C = c_1 \dots c_n$ cipher-text

$$c_i \equiv x_i - m_i \bmod 26$$

$\Rightarrow$ write this $C \equiv X - M \bmod 26$ (entrywise).

Assume that several pl.-texts $M_1, \dots, M_s$ were encrypted with the same $X \Rightarrow$ system of equations:

$$\begin{cases} C_1 \equiv X - M_1 \\ C_2 \equiv X - M_2 \\ \dots \\ C_s \equiv X - M_s \end{cases} \quad (*)$$

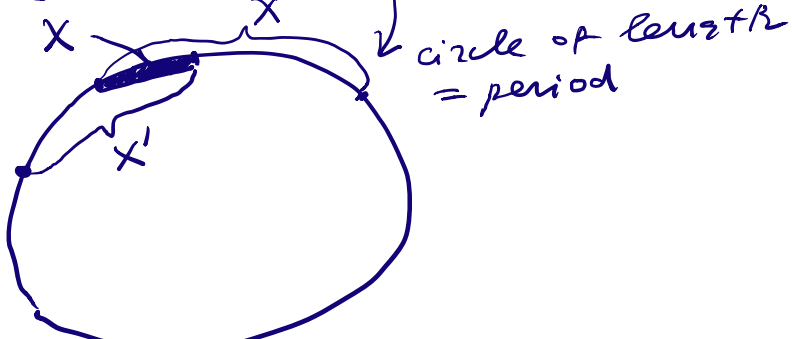
s called depth.

goal recover all $M_1, M_2, \dots, M_s$ from $(*)$

Why that may happen for M-209.

period of the key-stream

$26 \cdot 25 \cdot 23 \cdot 21 \cdot 19 \cdot 17 \approx 10^8$



M'

X' used to

encrypt M'

X'' used to encrypt M''

some part M_1 of M'
M_2 of M''
were encrypted with the same key-stream X

$$C_1 = X - M_1$$
$$C_2 = X - M_2$$

depth is 2.

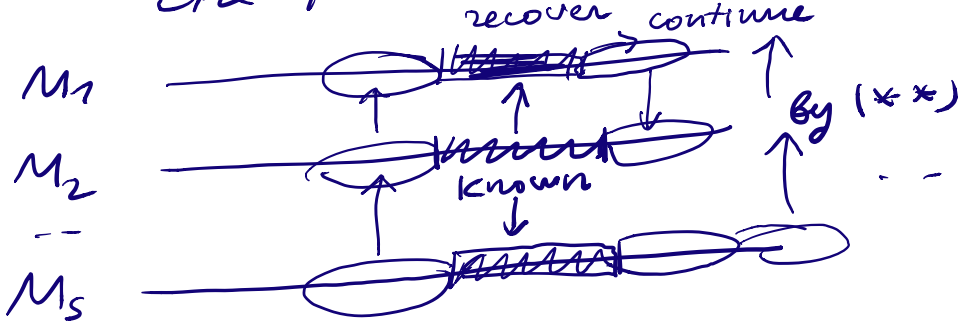
How at-depth cryptanalysis works.

Eliminate X from (*).

$$\begin{cases} C_2 - C_1 \equiv M_1 - M_2 & (**) \\ \dots \\ C_s - C_1 \equiv M_1 - M_s \end{cases}$$

observation.

one knows part of some of $M_1, \dots, M_s$
$\Rightarrow$ one recovers parts of the rest of the plain-text in those positions.



for $s \geq 3$ works very well
$s = 2$ still works

Probable words.

1) general content of the communication

military — "general" "generalstaff" "Headquarters" ---

diplomatic — "embassy" "envoy" ---

scientific — "theory", "nuclear bomb", ---

2) English probable words are
"ofthe"
"which"
"therearе"

Algorithm

1) try a probable word at all locations in the plain-texts $M_1, \dots, M_s$

2) recover relevant portions of the other pl.-text
   a) portion is sensible ⇒ guess was correct
   b) non sensible ⇒ guess was wrong

↖ (return to step 1)

↙ try another location or another probable word.

3) if stopped, then make new guesses with new probable words ----

---

Example.

3 cipher-texts from Lecture Notes

C_1 363 cR.
C_2 249
C_3 234

C_1 = gcijq / RxjRz| ---

C_2 = rsqox / vqdzm| ---

C_3 = Project Gutenberg ...

eliminate key-stream X : $C_i \equiv X - M_i$

$M_{31} \equiv C_1 - C_3 \equiv M_3 - M_1$ = zxvqe/Adqgm/...

$M_{32} \equiv C_2 - C_3 \equiv M_3 - M_2$ = kbcfl/vgkez/...

M_{31} 234-ch. string

M_{32} 245-ch. string.

assume

the discrete logarithm problem

on one of M_1, M_2, M_3.

trying the prob. word in M_3 at position 35

H	AVEMER	GEDAS	APROM	ISING	NEWAR	EAINP	UB	LICKEY
...	PROTOC	OLALS	OCALL	EDEXP	ONENT	IALKE	YA	GREEM...
	BASEDON	THEDI	SCRET	ELOGA	RITHM	PROBL	EM	ANDIS...

guess correct

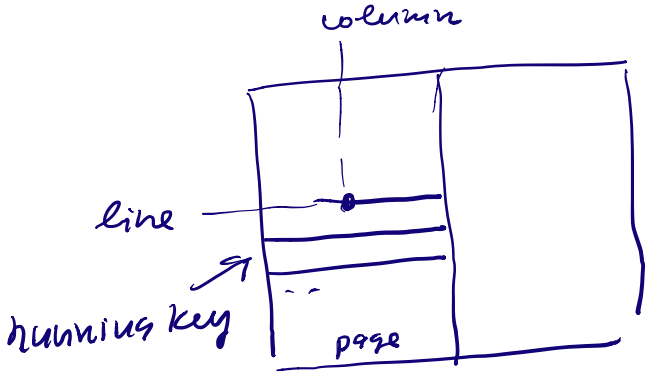
recover all pl.-texts.

Venona project.

Running key cipher.

aperiodic key-stream from a book.

key : name of a book
page number
line number
column number.



without spaces, punctuation, in lower case.

$X = x_1 \cdots x_n$

to encrypt

$M = m_1 \cdots m_n$

$C = c_1 \cdots c_n$

$$c_i \equiv x_i - m_i \bmod 26$$

or $\equiv x_i + m_i \bmod 26$

cipher text $C \equiv X + M \bmod 26$

given C find M,

easy because X, M are two pl-texts

at depth cryptanalysis with depth = 2.

Running key cipher is broken

because English is 1.50% redundant.

$X, M \xrightarrow{\text{encoding}} C$

of length $2 \cdot n$ of length n.

$\xleftarrow{\text{decoding}}$

Half of this text is redundant.

But it is known English is 75% redundant.

$X_1 X_2 X_3 X_4$ (length $4 \cdot n$) $\xrightarrow{\text{encoding}}$ Code length n

$\xleftarrow{\text{decoding.}}$

$3n$ characters are redundant.

$\Rightarrow$ Improve running key cipher.

$$C \equiv M + X_1 + X_2 + X_3 + X_4 \mod 26$$

C: cipher-text, M: pl.-text, X_1, X_2, X_3, X_4: running keys.

if this is not secure.

given C recover M.

by symmetry recover X_1, X_2, X_3, X_4.

$\Rightarrow$ English is 80% redundant.

not the case.

this encryption is perfect.

even if $C \equiv M + X_1 + X_2 \mod 26$ very very diff. to recover M.

$C_1 = X - M_1$

$C_2 = X - M_2$ depth $= 3$

$C_3 = X - M_3$