Bahan kuliah IF4020 Kriptografi

04 - Kriptografi Klasik

(Bagian 2)

Oleh: Rinaldi M

Pogram Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2026

A. Cipher Abjad-Tunggal (*monoalphabetic cipher*)

- *Monoalpabhetic cipher*: kelompok cipher substitusi yang mengganti satu huruf dengan satu huruf yang lain

- *Caesar cipher* adalah salah satu *cipher* abjad-tunggal dengan melakukan substitusi huruf berdasarkan hasil pergeseran huruf-huruf alfabet sejauh *k* huruf. Namun *Caesar Cipher* bukan satu-satunya *cipher* abjad-tunggal.

- Secara umum, kita dapat membentuk tabel subtitusi sembarang. Jumlah kemungkinan tabel substitusi yang dapat dibuat pada sembarang *cipher* abjad-tunggal adalah sebanyak

  26! = 403.291.461.126.605.635.584.000.000

  karena ada 26! cara mempermutasikan 26 huruf alfabet.

- Tabel substitusi dapat dibentuk secara acak seperti contoh berikut:

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

- Atau berdasarkan kalimat kunci yang mudah diingat:

  Contoh: `di bawah sinar bulan purnama hati resah jadi senang`

  Buang duplikasi huruf menjadi: `dibawhsnrulpmtejg`

  Sambung dengan huruf lain yang belum ada:

  `dibawhsnrulpmtejgcfkoqvwxyz`

  Tabel substitusi yang dihasilkan:

  `Plainteks :` 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

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

Kriptanalisis Cipher Abjad-Tunggal

- Kelemahan *cipher* abjad-tunggal: tidak dapat menyembunyikan hubungan statistik antara plainteks dengan cipherteks.
  - Huruf yang sama dienkripsi menjadi huruf cipherteks yang sama
  - Huruf yang sering muncul di dalam plainteks, juga sering muncul di dalam huruf cipherteksnya yang berkoresponden.
  - artinya struktur statistik bahasa masih muncul di dalam cipherteks

- Oleh karena itu, cipherteks dapat didekripsi tanpa mengetahui kuncinya sekalipun dengan menggunakan **teknik analisis frekuensi**.

Teknik Analisis Frekuensi

- Pada *cipher* abjad-tunggal, perulangan huruf di dalam plainteks tercermin pula pada perulangan huruf cipherteksnya.

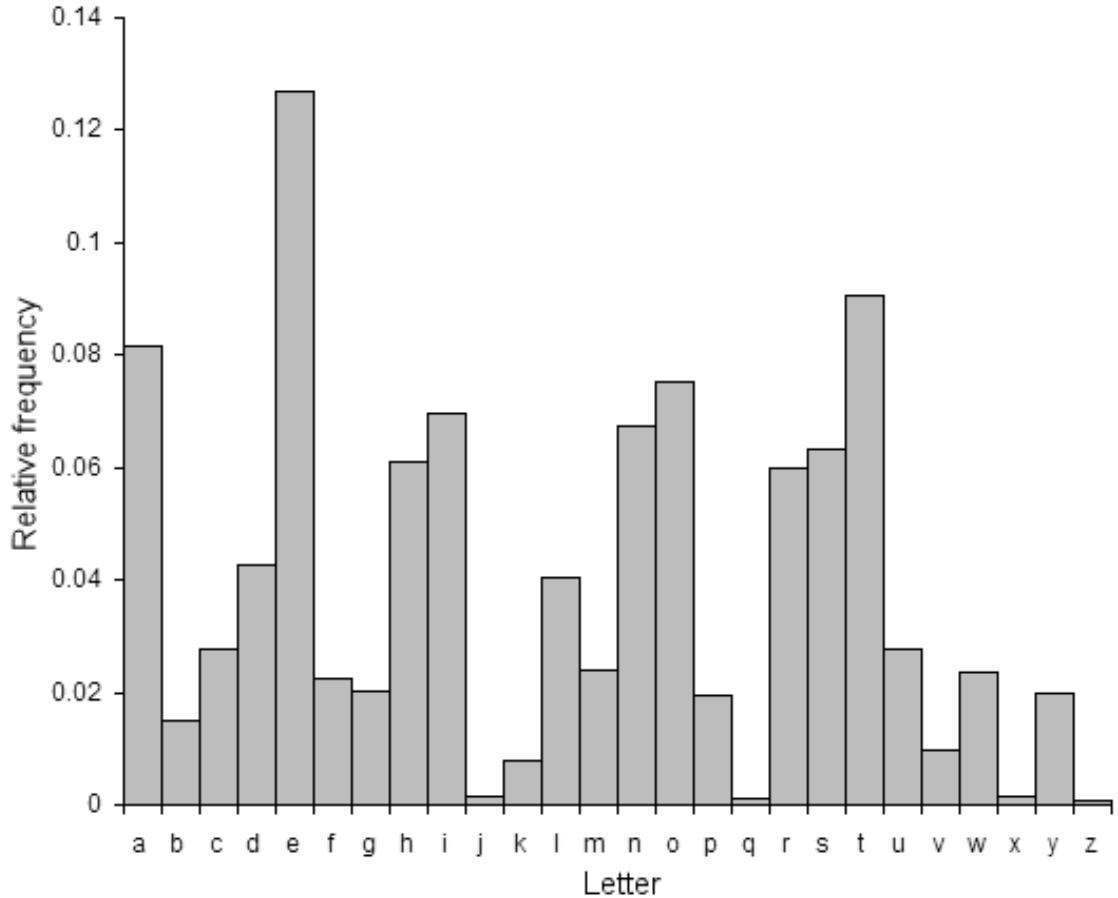
- Artinya, huruf yang sering muncul di dalam plainteks, maka huruf cipherteksnya juga sering muncul.

- Hubungan statistik antara huruf-huruf di dalam plainteks dengan huruf-huruf di dalam cipherteks menjadi peluang bagi kriptanalis untuk memecahkan cipherteks tanpa mengetahui kunci yang digunakan.

- Dengan memanfaatkan tabel frekuensi kemunculan huruf, pasangan huruf (bigram), dan tiga huruf (trigram) di dalam suatu bahasa natural, kriptanalis dapat dapat mencari tabel substitusi huruf plainteks menjadi huruf cipherteks dengan

Tabel Frekuensi kemunculan (relatif) huruf-huruf dalam teks Bahasa Inggris (sampel mencapai 300.000 karakter di dalam sejumlah novel dan suratkabar

Huruf	%	Huruf	%
A	8,2	N	6,7
B	1,5	O	7,5
C	2,8	P	1,9
D	4,2	Q	0,1
E	12,7	R	6,0
F	2,2	S	6,3
G	2,0	T	9,0
H	6,1	U	2,8
I	7,0	V	1,0
J	0,1	W	2,4
K	0,8	X	2,0
L	4,0	Y	0,1
M	2,4	Z	0,1



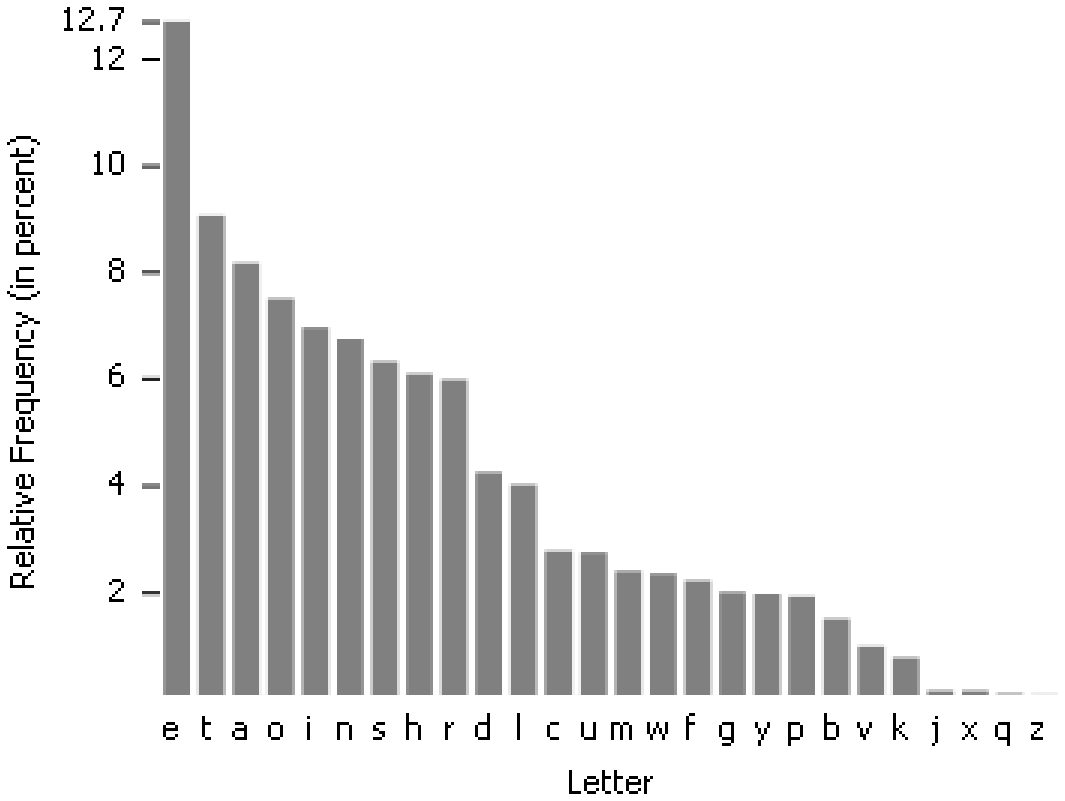
- *Top* 10 huruf yang sering muncul dalam teks Bahasa Inggris: E, T, A, O, I, N, S, H, R, D, L, U

- Top 10 huruf *bigram* yang sering muncul dalam teks B. Inggris: TH, HE, IN, EN, NT, RE, ER, AN, TI, dan ES

- Top 10 huruf *trigram* yang sering muncul dalam teks B. Inggris: THE, AND, THA, ENT, ING, ION, TIO, FOR, NDE, dan HAS

- Kriptanalis menggunakan tabel frekuensi kemunculan huruf dalam B. Inggris sebagai kakas bantu melakukan dekripsi.

- Misalnya, jika huruf “R” paling sering muncul di dalam cipherteks, maka kemungkinan besar itu adalah huruf “E” di dalam plainteksnya.



Bigram Frequencies

TH : 2.71	EN : 1.13	NG : 0.89
HE : 2.33	AT : 1.12	AL : 0.88
IN : 2.03	ED : 1.08	IT : 0.88
ER : 1.78	ND : 1.07	AS : 0.87
AN : 1.61	TO : 1.07	IS : 0.86
RE : 1.41	OR : 1.06	HA : 0.83
ES : 1.32	EA : 1.00	ET : 0.76
ON : 1.32	TI : 0.99	SE : 0.73
ST : 1.25	AR : 0.98	OU : 0.72
NT : 1.17	TE : 0.98	OF : 0.71

Trigram Frequencies

THE : 1.81	ERE : 0.31	HES : 0.24
AND : 0.73	TIO : 0.31	VER : 0.24
ING : 0.72	TER : 0.30	HIS : 0.24
ENT : 0.42	EST : 0.28	OFT : 0.22
ION : 0.42	ERS : 0.28	ITH : 0.21
HER : 0.36	ATI : 0.26	FTH : 0.21
FOR : 0.34	HAT : 0.26	STH : 0.21
THA : 0.33	ATE : 0.25	OTH : 0.21
NTH : 0.33	ALL : 0.25	RES : 0.21
INT : 0.32	ETH : 0.24	ONT : 0.20

- Perbandingan: top 10 huruf yang paling sering muncul dalam Bahasa Indonesia:

Huruf	Peluang (%)
A	17,50
N	10,30
I	8,70
E	7,50
K	5,65
T	5,10
R	4,60
D	4,50
S	4,50
M	4,50

Langkah-langkah kriptanalisis dengan teknik analisis frekuensi adalah sbb:

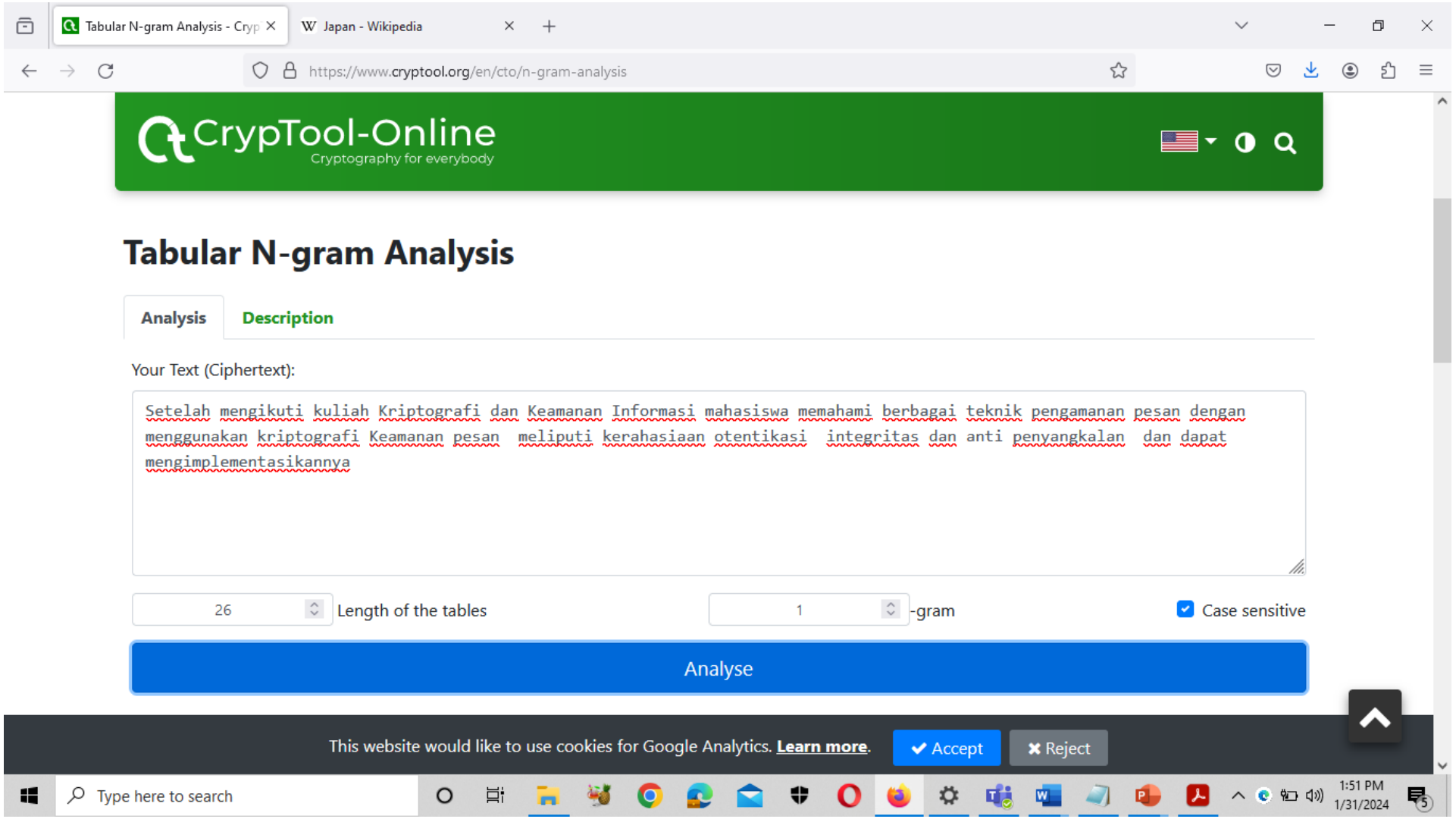
1. Hitung frekuensi kemunculan relatif huruf-huruf di dalam cipherteks.

2. Bandingkan hasil langkah 1 dengan tabel frekuensi kemunculan huruf. Mengingat huruf yang paling sering muncul dalam teks Bahasa Inggris adalah huruf E, maka huruf yang paling sering muncul di dalam cipherteks kemungkinan besar berpadanan dengan huruf E di dalam plainteksnya.

3. Langkah 2 diulangi untuk huruf dengan frekuensi terbanyak berikutnya. (biasanya hanya terpakai untuk 2 sampai 3 huruf pertama di dalam tabel frekuensi).

4. Ulangi langah 1 dan 2 dengan menggunakan bigram, trigram, dst, yang sering muncul di dalam cipherteks.

- Kakas online untuk menghitung frekuensi kemunculan huruf, bigram, trigram dsb: https://www.cryptool.org/en/cto/n-gram-analysis



N-gram tables

Rank	1-gram	Abs.	Rel.
1	a	44	19.298
2	n	31	13.596
3	i	22	9.649
4	e	21	9.211
5	m	14	6.140
6	t	13	5.702
7	g	11	4.825
8	k	10	4.386
9	p	9	3.947
10	s	9	3.947
11	r	8	3.509

Contoh 1: Diberikan cipherteks berikut ini (Stalling, 2011), spasi tidak dibuang:

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
```

Kita akan malakukan kriptanalisis dengan metode analisis frekuensi untuk memperoleh plainteks.

Asumsi: bahasa yang digunakan adalah Bahasa Inggris dan *cipher* yang digunakan adalah *cipher* abjad-tunggal.

Hitung frekuensi kemunculan huruf di dalam cipherteks tersebut sbb:

Rank	1-gram	Abs.	Rel.
1	P	16	13.333
2	Z	14	11.667
3	U	10	8.333
4	S	10	8.333
5	O	9	7.500
6	M	8	6.667
7	H	7	5.833
8	E	6	5.000
9	D	6	5.000
10	V	5	4.167
11	X	5	4.167

Rank	1-gram	Abs.	Rel.
12	W	4	3.333
13	F	4	3.333
14	Q	3	2.500
15	T	3	2.500
16	G	2	1.667
17	B	2	1.667
18	A	2	1.667
19	Y	2	1.667
20	I	1	0.833
21	J	1	0.833

Jumlah total huruf, N = 120

$$CI = \frac{\sum_{i=1}^{21} f_i(f_i-1)}{N(N-1)} = \frac{916}{120(119)} = 0{,}0641$$

Nilai CI sangat dekat dengan 0,065 (teks bahasa Inggris), jauh dari 0,038 (teks acak/polyalphabetic cipher), maka kemungkinan besar ini adalah cipherteks dari *monoalphabetic cipher*

Rank	1-gram	Abs.	Rel.
1	P	16	13.333
2	Z	14	11.667
3	U	10	8.333
4	S	10	8.333
5	O	9	7.500
6	M	8	6.667
7	H	7	5.833
8	E	6	5.000
9	D	6	5.000
10	V	5	4.167
11	X	5	4.167

- Dua huruf yang paling sering muncul di dalam cipherteks: huruf P dan Z.
- Dua huruf yang paling sering muncul di dalam B. Inggris: huruf E dan T.
- Kemungkinan besar,
  - P adalah pemetaan dari e
  - Z adalah pemetaan dari t

- Tetapi kita belum dapat memastikannya sebab masih diperlukan cara *trial and error* dan pengetahuan tentang Bahasa Inggris.

- Tetapi ini adalah langkah awal yang bagus.

Iterasi 1:

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
 t           e   e te    t  t  e e             t

VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
   e t    t  t    e  ee    e   t     t

EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
 e  e e t t  e    t e   et
```

- ZWP dan ZWSZ dipetakan menjadi t*e dan t**t
- Kemungkinan besar W adalah pemetataan dari H sehingga kata yang mungkin untuk ZWP dan ZWSZ adalah the dan that

- Diperoleh pemetaan (cipherteks → plainteks):

  P → e
  Z → t
  W → h
  S → a

- **Iterasi 2:**

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
 t  a         e   e te  a  that  e e a        a    t

VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
   e t    ta t  ha e  ee   a e   th     t  a

EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
 e  e e tat  e     the   et
```

- WSFP dipetakan menjadi ha*e.

- Dalam Bahasa Inggris, kata yang mungkin untuk ha*e hanyalah have, hate, hale, dan haze

- Dengan mencoba mengganti semua F di dalam cipherteks dengan v, t, l, dan z, maka huruf yang cocok adalah v sehingga WSFP dipetakan menjadi have

- Dengan mengganti F menjadi v pada kriptogram EPYEPOPDZSZUFPO sehingga menjadi *e*e*e*tat*ve*, maka kata yang cocok untuk ini adalah representatives

- Diperoleh pemetaan:

E → r Y → p

U → I O → s

D → n

- Hasil akhir bila diselesaikan seluruhnya:

It was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in Moscow

- Tabel substitusi yang dihasilkan:

Plainteks: 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

Cipherteks: **S A H V P B J W U - - X T D M Y Z E O Z I F Q – G -**

Contoh 2: Diberikan cipherteks berikut ini, yang dihasilkan dari cipher abjad-Tunggal. Lakukan dekripsi untuk mengungkap plainteksnya.

```
LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHKVSTYLXZIX
LIKIIXPIJVSZEYPERRGERIMWQLMGLMXQERIWGPSRIHMXQEREKIETXMJT
PRGEVEKEITREWHEXXLEXXMZITWAWSQWXSWEXTVEPMRXRSJGSTVRIEYVI
EXCVMUIMWERGMIWXMJMGCSMWXSJOMIQXLIVIQIVIXQSVSTWHKPEGARCS
XRWIEVSWIIBXVIZMXFSJXLIKEGAEWHEPSWYSWIWIEVXLISXLIVXLIRGE
PIRQIVIIBGIIHMWYPFLEVHEWHYPSRRFQMXLEPPXLIECCIEVEWGISJKTV
WMRLIHYSPHXLIQIMYLXSJXLIMWRIGXQEROIVFVIZEVAEKPIEWHXEAMWY
EPPXLMWYRMWXSGSWRMHIVEXMSWMGSTPHLEVHPFKPEZINTCMXIVJSVLMR
SCMWMSWVIRCIGXMWYMX
```

- Hasil perhitungan frekuensi kemunculan huruf, bigram, dan trigram:
  - huruf `I` paling sering muncul,
  - `XL` adalah bigram yang paling sering muncul,
  - `XLI` adalah trigram yang paling sering muncul.

Ketiga data terbanyak ini menghasilkan dugaan bahwa

`I` berkoresponden dengan huruf plainteks `e`,

`XLI` berkoresponden dengan `the`,

`XL` berkoresponden dengan `th`

Pemetaan:

`I` → `e`

`X` → `t`

`L` → `h`

- XLEX dipetakan menjadi th*t.
- Kata yang cocok untuk th*t. adalah that.
- Jadi kita memperoleh: E → a
- Hasil iterasi pertama:

heVeTCSWPeYVaWHaVSReQMthaYVaOeaWHRtatePFaMVaWHKVSTYhtZe
theKeetPeJVSZaYPaRRGaReMWQhMGhMtQaReWGPSReHMtQaRaKeaTtM
JTPRGaVaKaeTRaWHatthattMZeTWAWSQWtSWatTVaPMRtRSJGSTVRea
YVeatCVMUeMWaRGMeWtMJMGCSMWtSJOMeQtheVeQeVetQSVSTWHKPaG
ARCStRWeaVSWeeBtVeZMtFSJtheKaGAaWHaPSWYSWeWeaVtheStheVt
heRGaPeRQeVeeBGeeHMWYPFhaVHaWHYPSRRFQMthaPPtheaCCeaVaWG
eSJKTVWMRheHYSPHtheQeMYhtSJtheMWReGtQaROeVFVeZaVAaKPeaW
HtaAMWYaPPthMWYRMWtSGSWRMHeVatMSWMGSTPHhaVHPFKPaZeNTCMt
eVJSVhMRSCMWMSWVeRCeGtMWYMt

- Selanjutnya,

  `Rtate` mungkin adalah `state`,

  `atthattMZE` mungkin adalah `atthattime`,

  `heVe` mungkin adalah `here`.

- Jadi, kita memperoleh pemetaan baru:

  `R` → `s`

  `M` → `i`

  `Z` → `m`

  `V` → `r`

- Hasil iterasi ke-2:

```
hereTCSWPeYraWHarSseQithaYraOeaWHstatePFairaWHKrSTYhtm
etheKeetPeJrSmaYPassGaseiWQhiGhitQaseWGPSseHitQasaKeaT
tiJTPsGaraKaeTsaWHatthattimeTWAWSQWtSWatTraPistsSJGSTr
seaYreatCriUeiWasGieWtiJiGCSiWtSJOieQthereQeretQSrSTWH
KPaGAsCStsWearSWeeBtremitFSJtheKaGAaWHaPSWYSWeWeartheS
therthesGaPesQereeBGeeHiWYPFharHaWHYPSssFQithaPPtheaCC
earaWGeSJKTrWisheHYSPHtheQeiYhtSJtheiWseGtQasOerFremar
AaKPeaWHtaAiWYaPPthiWYsiWtSGSWsiHeratiSWiGSTPHharHPFKP
ameNTCiterJSrhisSCiWiSWresCeGtiWYit
```

- Teruskan, dengan menerka kata-kata yang sudah dikenal, misalnya `remarA` mungkin `remark`, dsb

- Hasil iterasi 3:

```
hereuponlegrandarosewithagraveandstatelyairandbr
oughtmethebeetlefromaglasscaseinwhichitwasenclos
editwasabeautifulscarabaeusandatthattimeunknownt
onaturalistsofcourseagreatprizeinascientificpoin
tofviewthereweretworoundblackspotsnearoneextremi
tyofthebackandalongoneneartheotherthescalesweree
xceedinglyhardandglossywithalltheappearanceofbur
nishedgoldtheweightoftheinsectwasveryremarkablea
ndtakingallthingsintoconsiderationicouldhardlybl
amejupiterfor*hisopinionrespectingit
```

- Tambahkan spasi, tanda baca, dll

Here upon Legrand arose, with a grave and stately air, and brought me the beetle from a glass case in which it was enclosed. It was a beautiful scarabaeus, and, at that time, unknown to naturalists—of course a great prize in a scientific point of view. There were two round black spots near one extremity of the back, and a long one near the other. The scales were exceedingly hard and glossy, with all the appearance of burnished gold. The weight of the insect was very remarkable, and, taking all things into consideration, I could hardly blame Jupiter for his opinion respecting it.

B. Vigènere Cipher

- Vigenere Cipher termasuk ke dalam kelompok *cipher* abjad-majemuk (*polyalpabetic cipher*). Ini berbeda dengan cipher abjad-tunggal (*monolphabetic cipher*).

- Perbedaan *polyalphabetic cipher* dengan *monoalphabetic cipher*:
  - *Cipher* abjad-tunggal: satu kunci untuk semua huruf plainteks

    Contoh: Caesar cipher
  - *Cipher* abjad-majemuk: setiap huruf menggunakan kunci berbeda.

    Contoh: Vigenere cipher

- *Cipher* abjad-majemuk dibuat untuk mengatasi kelemahan *cipher* abjad-Tunggal yang mudah dipecahkan dengan teknik analisis frekuensi

Bentuk umum *cipher* abjad-majemuk

- Kunci:

  $K = k_1 k_2 \ldots k_m$ (ket: m = panjang kunci)

  Karena m lebih pendek daripada panjang plainteks, maka kunci diulang secara periodik

- Plainteks:

  $P = p_1 p_2 \ldots p_m p_{m+1} \ldots p_{2m} \ldots$

- Cipherteks:

  $$C = E_k(P) = f_{k_1}(p_1)\, f_{k_2}(p_2) \ldots f_{k_m}(p_m) f_{k_1}(p_{m+1}) f_{k_2}(p_{m+2}) \ldots f_{k_m}(p_{2m}) \ldots$$

  f(.) adalah fungsi enkripsi pada *cipher* abjad-tunggal

- Untuk m = 1, *cipher*-nya ekivalen dengan *cipher* abjad-tunggal.

- Misalkan panjang kunci *m* = 10, maka 10 huruf pertama plainteks dienkripsi dengan kunci K, setiap huruf ke-*i* menggunakan kunci k_i.

- Untuk 10 karakter berikutnya, kembali menggunakan pola enkripsi yang sama.

Contoh: kunci = `sony` `(m = 4)`

Plainteks: `thisplaintextissecret`

Kunci: `sonysonysonysonysonys`

Setiap huruf plainteks dienkripsi dengan huruf kunci di bawahnya dengan *cipher* abjad tunggal (misalnya menggunakan *Caesar Cipher*):

(t+s) mod 26

(h + o) mod 26

(i + n) mod 26

dst…

Sejarah Vigenere Cipher

- Penemu *cipher* ini sebenarnya adalah Giovan Batista Belaso, karena ia menggambarkan pertama kali pada tahun 1553 seperti ditulis di dalam bukunya *La Cifra del Sig. Giovan Batista Belaso.*

- Namun, *cipher* ini disempurnakan dan dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Vigènere pada abad 16 (tahun 1586).

- Pada abat ke-19, banyak orang yang mengira Vigenère adalah penemu *cipher* ini, sehingga dikenal luas sebagai *Vigenère Cipher*.

Blaise de Vigenère

23 languages

Article Talk Read Edit View history

From Wikipedia, the free encyclopedia

Blaise de Vigenère (French: [viʒnɛːʁ]; 5 April 1523 – 19 February 1596)[1] was a French diplomat, cryptographer, translator and alchemist.

Biography [edit]

Vigenère was born into a respectable family in the village of Saint-Pourçain in Bourbonnais. When he was 12, his father, Jehan (modern spelling Jean) de Vigenère, arranged for him to have a classical education in Paris. Registered at the university at 14, he quit after three years without a known degree.[2]

From 1539 to around 1545, he worked under Gilbert Bayard, a first secretary to King Francis I, who had fiefs in Bourbonnais.[2]

In 1545, he accompanied the French envoy Louis Adhémar de Monteil, Count of Grignan, to the Diet of Worms as a junior secretary.[3][4] After the diet's rupture, he traveled in Europe.[5]

In 1547, he quit the court and entered the service of the House of Nevers. He would remain associated with it until at least a year before

Blaise de Vigenère

Born	5 April 1523 Saint-Pourçain-sur-Sioule, Kingdom of France
Died	19 February 1596 (aged 72) Paris, Kingdom of France
Occupations	Diplomat, cryptographer, alchemist
Known for	Vigenère cipher

Sumber: Wikipedia

- *Cipher* ini berhasil dipecahkan oleh Babbage dan Kasiski pada pertengahan Abad 19 (akan dijelaskan pada materi selanjutnya).

- Kasiski menguraikan langkah-Langkah untuk menemukan panjang kunci (bukan huruf-huruf kunci kunci).

- *Vigènere Cipher* digunakan oleh Tentara Konfiderasi (*Confederate Army*) pada Perang Sipil Amerika (*American Civil war*).

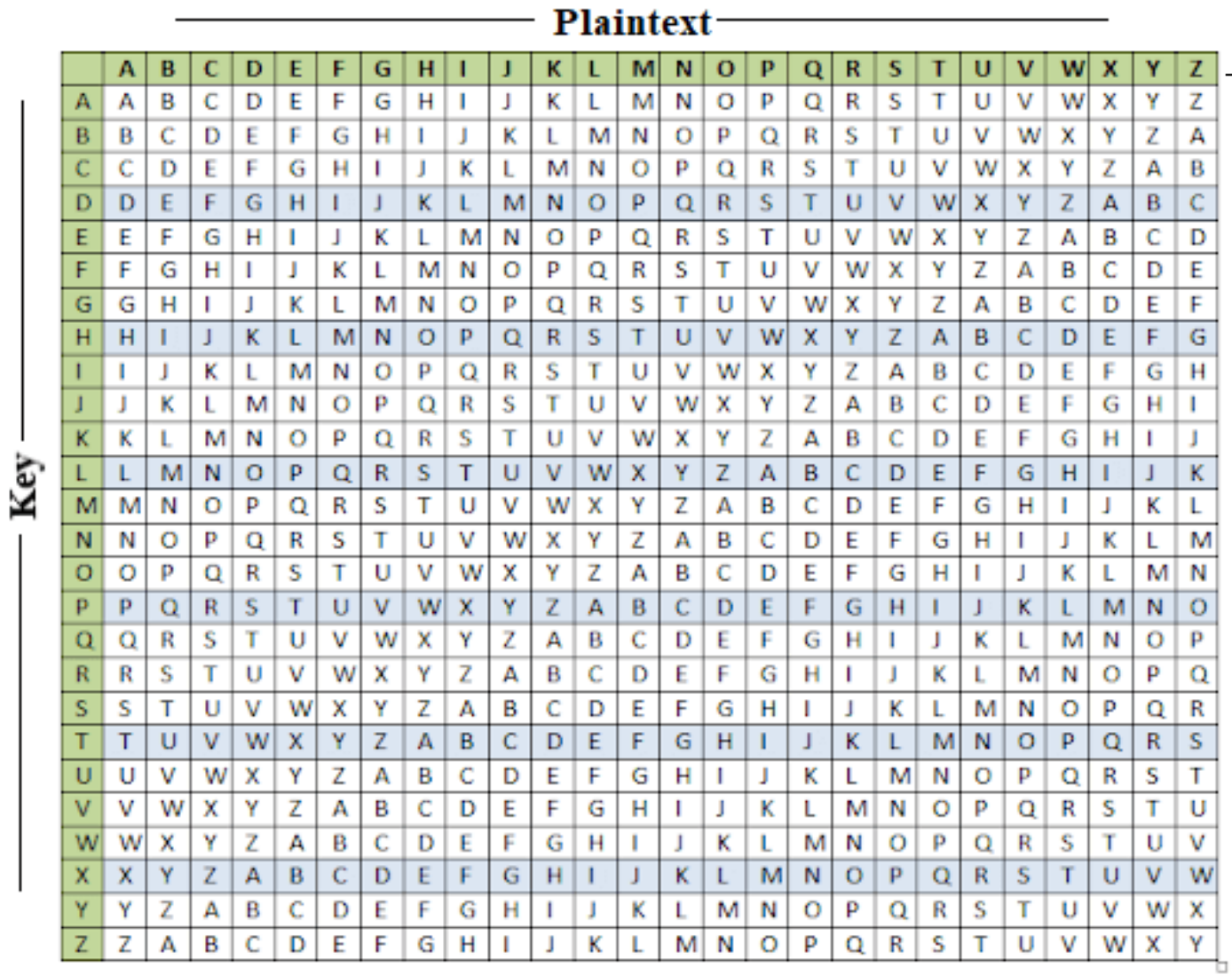
- Perang Sipil terjadi setelah *Vigènere Cipher* berhasil dipecahkan.

- *Vigènere Cipher* menggunakan tabel *Vigènere* – dinamakan *Vigenere square,* untuk melakukan enkripsi dan dekripsi.

- Setiap baris *i* di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh menggunakan *Caesar Cipher* dengan kunci *k* = *i*.

- Artinya, setiap baris *i* merupakan pergeseran huruf alfabet sejauh *i* ke kanan

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



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

→ baris ke-0

→ baris ke-25

A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8, J = 9, K = 10
L = 11, M = 12, N = 13, O = 14, P = 15, Q = 16, R = 17, S = 18, T = 19, U = 20
V = 21, W = 22, X = 23, Y = 24, Z = 25

- Kunci adalah string: $K = k_1 k_2 \ldots k_m$

  k_i untuk $1 \leq i \leq m$ menyatakan huruf-huruf alfabet

- Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci diulang secara periodik.

- Misalkan panjang kunci m = 10, maka 10 huruf pertama plainteks dienkripsi dengan kunci K, setiap huruf ke-i menggunakan kunci k_i.

Contoh: kunci = `soreang` (dalam angka: 18, 14, 17, 4, 0, 13, 6)

Plainteks: `terimapesanangulaikambing`

Kunci: `soreangsoreangsoreangsore`

Ini artinya setiap huruf plainteks dienkripsi menggunakan *Caesar Cipher* dengan kunci *k* yang berbeda-beda

Untuk 8 karakter berikutnya, kembali menggunakan pola enkripsi yang sama.

- Enkripsi dilakukan dengan mencari titik potong huruf plainteks dengan huruf kunci:

Plainteks : terimapesanangulaikambing
Kunci : soreangsoreangsoreangsore
Cipherteks: LSIMMNVWGRRAAMMZRMKNSTWEK

Plainteks

KUNCI	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
a	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
b	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	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	**L**	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	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

Gambar 4.3 Enkripsi huruf T dengan kunci s

A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8, J = 9, K = 10
L = 11, M = 12, N = 13, O = 14, P = 15, Q = 16, R = 17, S = 18, T = 19, U = 20
V = 21, W = 22, X = 23, Y = 24, Z = 25

- Hasil enkripsi seluruhnya adalah sebagai berikut:

Plainteks : `terimapesanangulaikambing`
Kunci : `soreangsoreangsoreangsore`
Cipherteks : `LSIMMNVWGRRAAMMZRMKNSTWEK`

- Tanpa menggunakan *Vigenere Square* pun enkripsi tetap dapat dihitung secara *Caesar Cipher* dengan menjumlahkan plainteks p_j dengan kunci k_i dalam modulus 26:

Enkripsi: $c_j = E(p_j) = (p_j + k_i) \bmod 26$ (1)

Dekripsi: $p_j = D(c_j) = (c_j - k_i) \bmod 26$ (2)

Contoh:

(t + s) mod 26 = (19 + 18) mod 26 = 37 mod 26 = 11 = L

(e + o) mod 26 = (4 + 14) mod 26 = 18 mod 26 = 18 = S, dst

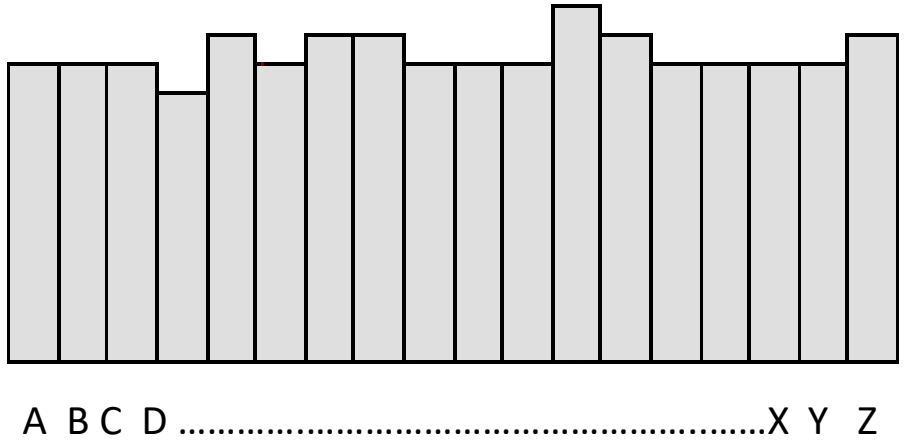
- Kelebihan Vigenere Cipher: huruf plainteks yang sama tidak selalu dienkripsi menjadi huruf cipheteks yang sama pula, bergantung huruf kunci yang digunakan.

  Contoh: pada contoh di atas, huruf plainteks `T` dapat dienkripsi menjadi **`L`** atau **`H`**, dan huruf cipherteks **`V`** dapat merepresentasikan huruf plainteks `H`, `I`, dan `X`

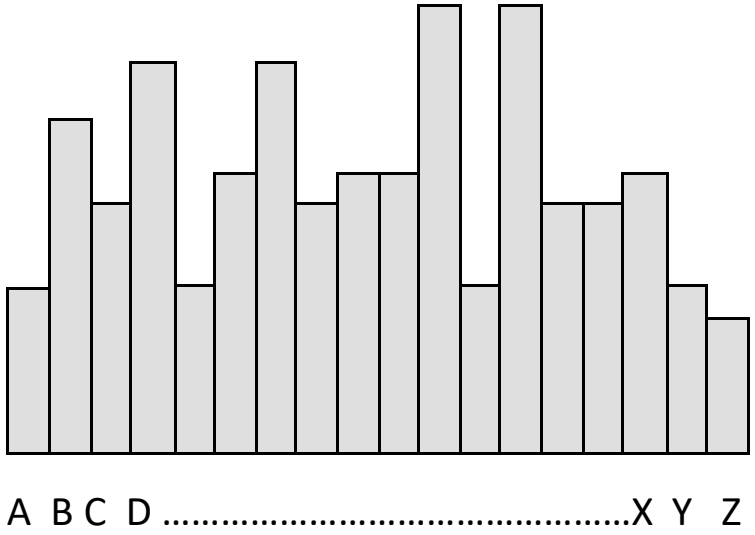
- Hal di atas merupakan karakteristik dari *cipher* abjad-majemuk: setiap huruf cipherteks dapat memiliki kemungkinan banyak huruf plainteks.

- Bandingkan dengan *cipher* abjad-tunggal, setiap huruf cipherteks selalu menggantikan huruf plainteks tertentu.

- Dengan karakteristik seperti terebut, maka distribusi kemunculan huruf di dalam cipherteks relatif mendekati seragam (*uniform*) dan histogramnya cenderung *flat* (datar)
- Histogram yang flat akan menyulitkan kriptanalisis memecahkan cipherteks dengan metode analisis frekuensi



Flat histogram



Bukan flat histogram

Plainteks:

Dinas Pendidikan Kota Ternate meminta kepada pihak sekolah dan orang tua siswa untuk jenjang pendidikan SD dan SMP se-Kota Ternate untuk melarang para siswa membawa permainan lato-lato yang sedang tren itu ke sekolah, karena akan mengganggu kegiatan belajar mengajar yang dinilai berbahaya sehingga mengantisipasi kecelakaan bagi anak di daerah itu.

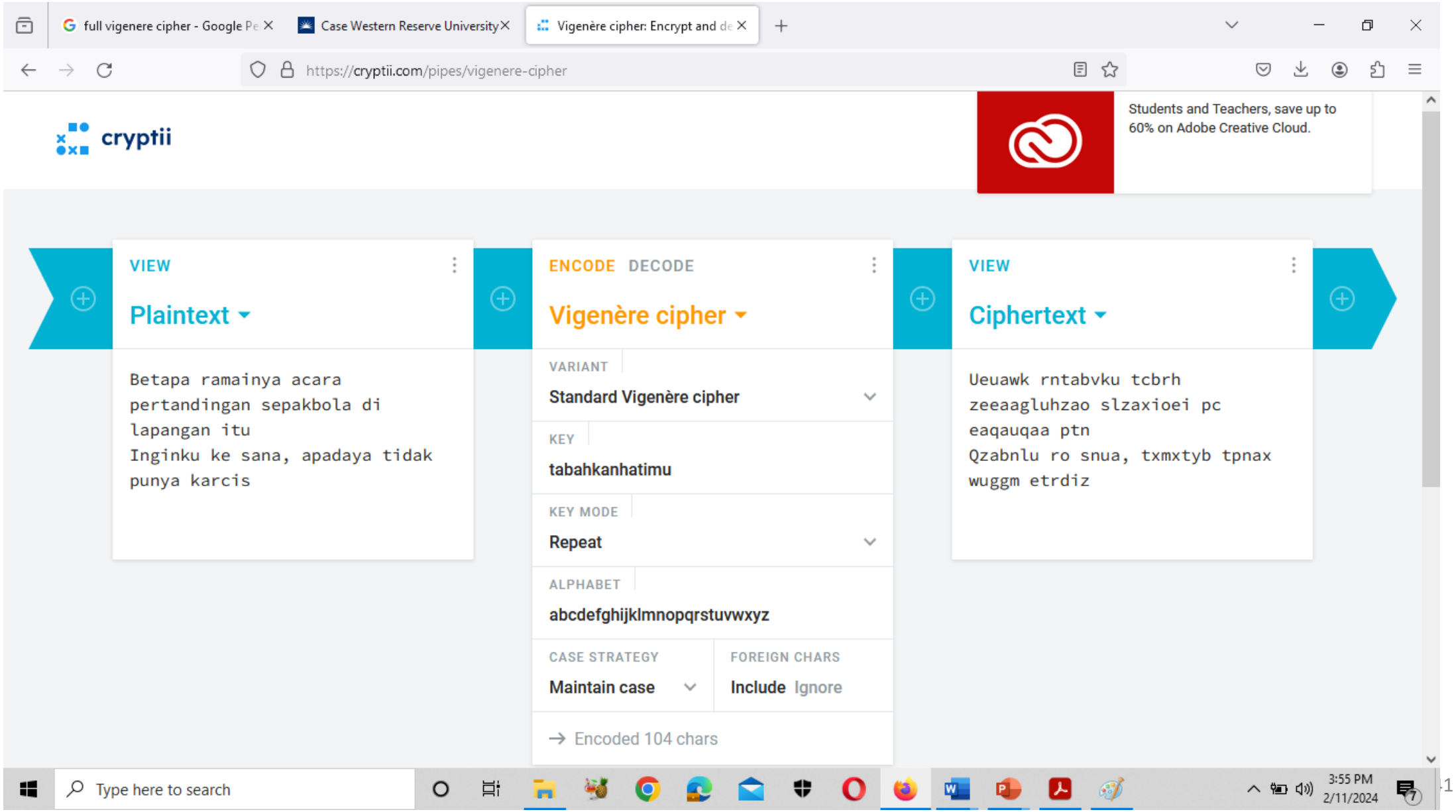
Kunci: selatsunda

Cipherteks:

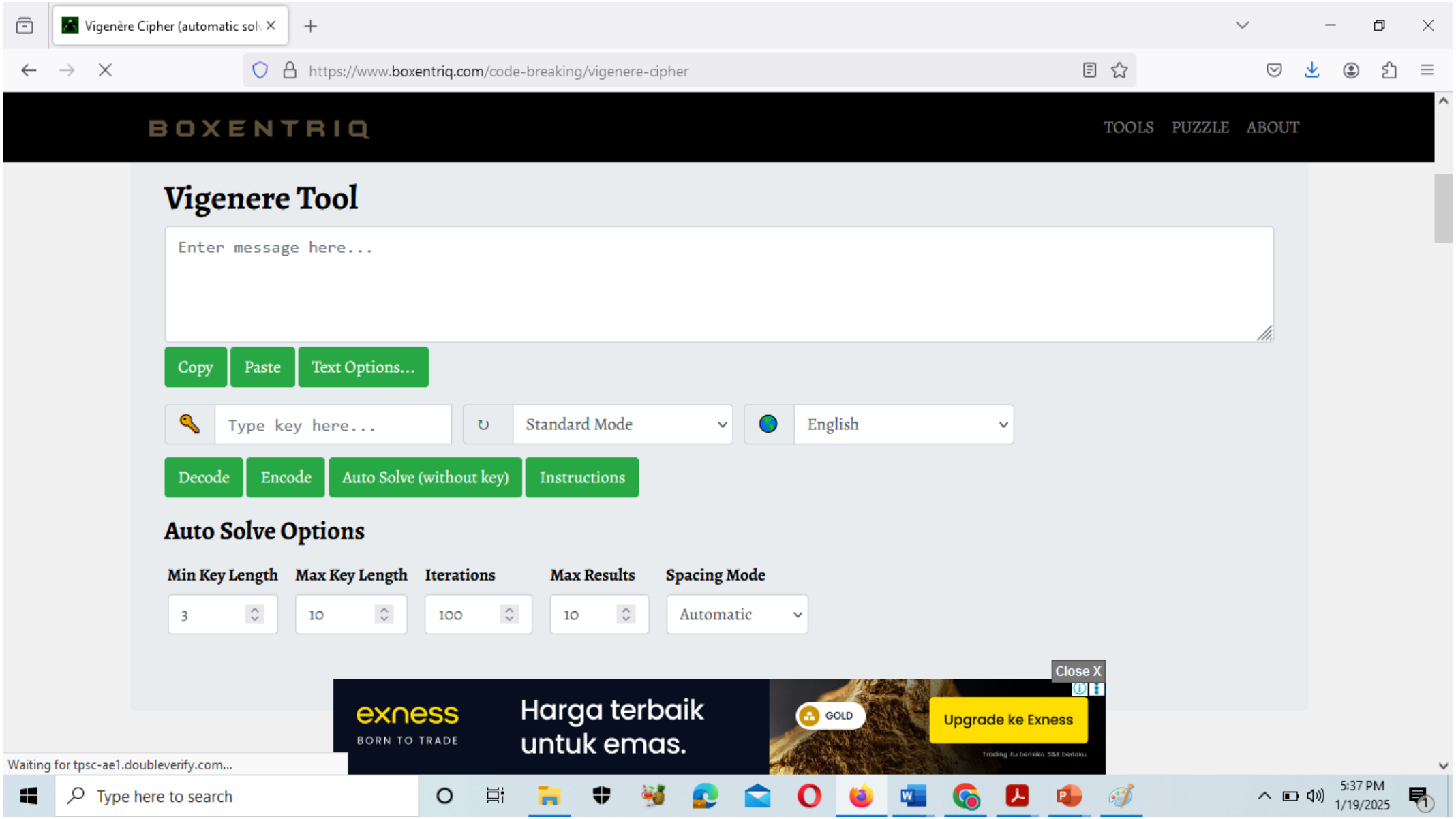
(dikelompokkan 4-huruf)

VMYAL HYAGI VMVAG CIGDT WVYAM WGRPI FXLKX HUQDP ALLKL WEBOA
ZHLNH JUAJT MEDIL OUHQT MOUEG BUAJP WROIW AENQS VHLNL EJFHK
GXLTX JHNWE MREUD EYYDR SRRPT JUFLS OEXEF TUJDP WVXAB FUAOA
LSWAM GSNQG KIOAG YNEHN AXFKX KYXRL SLVAK WHNDK SRXEG YANQG
YYVEZ AUGDN TIWAC SLZHN YEUAK QUAJD ARTLT AVRUB SLLYT KYULN YKLMX
FANQT AWTPT KCXHC WPLKT SHODG AEYAD VCQDE JESIM M

- Demo Vigenere Cipher online: https://cryptii.com/pipes/vigenere-cipher



https://www.boxentriq.com/code-breaking/vigenere-cipher



[7]:
```python
def Vigenere_cipher_encrypt(plainteks, kunci):
    cipherteks = ""
    kunci = kunci.lower()
    indeks_kunci = 0
    for char in plainteks:
        if char.isalpha():           # Hanya memproses huruf alfabet saja
            start = ord('A') if char.isupper() else ord('a')
            k = ord(kunci[indeks_kunci]) -  ord('a')
            c = (ord(char) - start + k) % 26   # Kodekan huruf ke angka 0 s/d 25, lalu enkripsi dengan Caesar Cipher
            c = c + start            # Kembalikan ke posisi semula
            cipherteks = cipherteks + chr(c)      # sambung setiap huruf cipherteks
            indeks_kunci = (indeks_kunci + 1) % len(kunci)
        else:
            cipherteks = cipherteks + char     # Pesan yang bukan huruf tidak dienkripsi, dibiarkan saja
    return cipherteks
```

[23]:
```python
def Vigenere_cipher_decrypt(cipherteks, kunci):
    plainteks = ""
    kunci = kunci.lower()
    indeks_kunci = 0
    for char in cipherteks:
        if char.isalpha():           # Hanya memproses huruf alfabet saja
            start = ord('A') if char.isupper() else ord('a')
            k = ord(kunci[indeks_kunci]) -  ord('a')
            c = (ord(char) - start - k) % 26   # Kodekan huruf ke angka 0 s/d 25, lalu enkripsi dengan Caesar Cipher
            c = c + start            # Kembalikan ke posisi semula
            plainteks = plainteks + chr(c)      # sambung setiap huruf plainteks
            indeks_kunci = (indeks_kunci + 1) % len(kunci)
        else:
            plainteks = plainteks + char     # Pesan yang bukan huruf tidak didekripsi, dibiarkan saja
    return plainteks
```

Vigenere Cipher dalam Python

[13]:
```
pesan = input("ketikkan pesan anda: ")
```

```
ketikkan pesan anda:  Sekarang tahun 2025, semoga kita sehat selalu, aamin!!
```

[5]:
```
kunci = input("kunci: ")
```

```
kunci:  apakabar
```

[21]:
```
cipherteks = Vigenere_cipher_encrypt(pesan, kunci)
```

[22]:
```
print(f"Pesan terenkripsi: {cipherteks}")
```

```
Pesan terenkripsi: Stkkrbnx tphen 2025, tedova uiua jewad sflrlj, akmjn!!
```

[24]:
```
plainteks = Vigenere_cipher_decrypt(cipherteks, kunci)
```

[25]:
```
print(f"Pesan hasil dekripsi: {plainteks}")
```

```
Pesan hasil dekripsi: Sekarang tahun 2025, semoga kita sehat selalu, aamin!!
```

Extended Vigenere Cipher

- Vigenere cipher untuk 26 huruf dapat dikembangkan untuk enkripsi 256 karakter ASCII

Enkripsi: $c_j = E(p_j) = (p_j + k_i) \bmod 256$

Dekripsi: $p_j = D(c_j) = (c_j - k_i) \bmod 256$

Latihan

- Ubahlah program Vigenere Cipher di atas sehingga dapat melakukan enkripsi pesan berupa file teks.

Kriptanalisis Vigenere Cipher

- Friedrich Kasiski adalah orang yang pertama kali memecahkan *Vigènere cipher* pada Tahun 1863.

> Friedrich Kasiski
> Born: November 29, 1805 @ Schlochau, Kingdom of Prussia
> Died: May 22, 1881 (aged 75) @ Neustettin, German Empire
> Nationality: German

- Metodenya dinamakan metode Kasiski

Metode Kasiski

- Metode Kasiski tidak secara langsung menemukan kunci Vigenere Cipher, tetapi membantu menemukan estimasi panjang kunci *Vigenere cipher*.

- Metode Kasiski memanfaatkan keuntungan bahwa bahasa Inggris tidak hanya mengandung perulangan huruf,

- tetapi juga perulangan pasangan huruf atau tripel huruf, seperti `TH`, `THE`, `EN`, dsb.

- Perulangan kelompok huruf ini ada kemungkinan menghasilkan kriptogram yang berulang.

Contoh 3:

Plainteks : **crypto**isshortfor**crypto**graphy

Kunci : abcdabcdabcdabcdabcdabcdabcdabcd

Cipherteks : **CSASTP**KVSIQUTGQU**CSASTP**IUAQJB

- Pada contoh ini, crypto dienkripsi menjadi kriptogram yang sama, yaitu **CSATP**.

- Tetapi kasus seperti ini tidak selalu demikian, misalnya pada contoh berikut ini….

Contoh 4:

Plainteks : **crypto**isshortfor**crypto**graphy

Kunci : abcdefabcdefabcdefabcdefabcd

Cipherteks : **CSASXT**ITUKWSTGQU**CWYQVR**KWAQJB

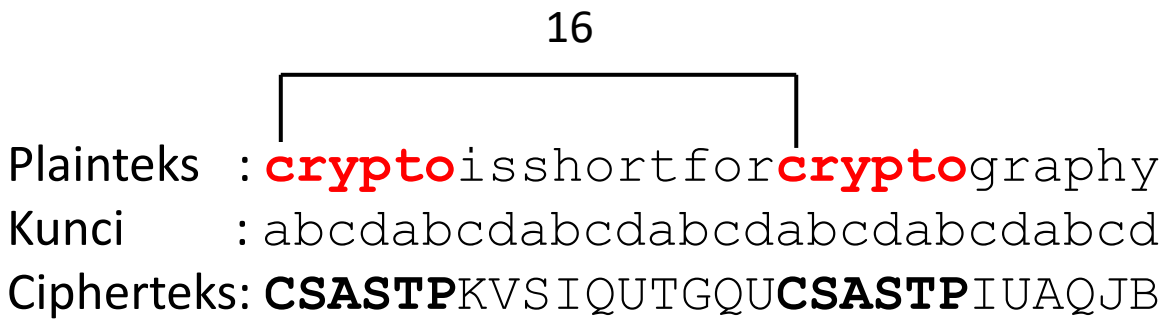
- Pada contoh di atas, `crypto` tidak dienkripsi menjadi kriptogram yang sama.

- Mengapa bisa demikian?

- Secara intuitif: jika jarak antara dua buah *string* yang berulang di dalam plainteks merupakan kelipatan dari panjang kunci,

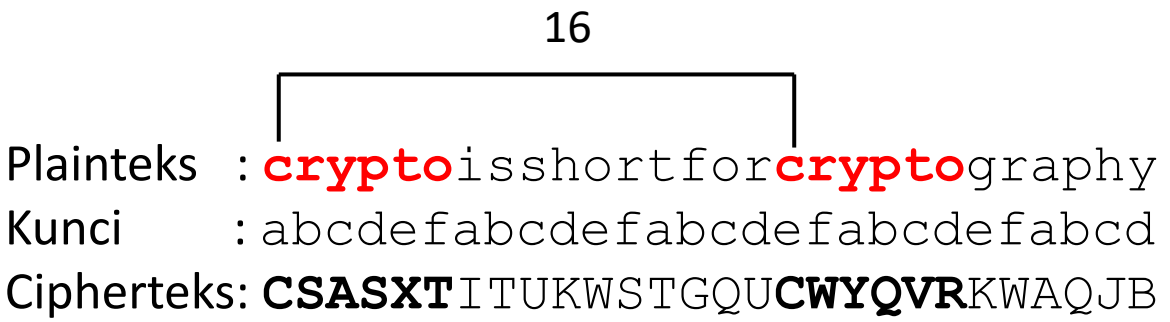
- maka *string* yang sama tersebut akan muncul menjadi kriptogram yang sama pula di dalam cipherteks.

- Pada Contoh 1,



  - kunci = `abcd`
  - panjang kunci = 4
  - jarak antara dua `crypto` yang berulang = 16
  - 16 = kelipatan 4

∴ `crypto` dienkripsi menjadi kriptogram yang sama



- Pada Contoh 2,
  - kunci = `abcdef`
  - panjang kunci = 6
  - jarak antara dua `crypto` yang berulang = 16
  - 16 bukan kelipatan 6

∴ `crypto` tidak dienkripsi menjadi kriptogram yang sama

- Goal metode Kasiski: mencari dua atau lebih kriptogram yang berulang untuk menentukan panjang kunci.

Langkah-langkah metode Kasiski:

1. Temukan semua kriptogram yang berulang di dalam cipherteks (pesan yang panjang biasanya mengandung kriptogram yang berulang).
2. Hitung jarak antara kriptogram yang berulang
3. Hitung semua faktor (pembagi) dari jarak tersebut (faktor pembagi menyatakan panjang kunci yang mungkin).
4. Tentukan irisan dari himpunan faktor pembagi tersebut. Nilai yang muncul di dalam irisan menyatakan angka yang muncul pada semua faktor pembagi dari jarak-jarak tersebut . Nilai tersebut mungkin adalah panjang kunci.

- Contoh:

  DYDUXRMHTVDV**NQD**QNW**DYDUXRMH**ARTJGW**NQD**

  Kriptogram yang berulang: **DYUDUXRMH** dan **NQD**.

  Jarak antara dua buah perulangan **DYUDUXRMH** = 18.

  Semua faktor pembagi 18 : {18, 9, 6, 3, 2}

  Jarak antara dua buah perulangan **NQD** =20.

  Semua faktor pembagi 20 : {20, 10, 5, 4, 2}.

  Irisan dari kedua buah himpunan tersebut adalah 2

  Panjang kunci kemungkinan besar adalah 2.

- Setelah panjang kunci diketahui, maka langkah berikutnya menentukan huruf-huruf kunci

- Huruf-huruf kunci dapat ditentukan dengan menggunakan *exhaustive key search*

- Jika panjang kunci = *p*, maka jumlah kunci yang harus dicoba sampai menemukan kunci yang benar adalah adalah maksimal 26^p kali.

- Namun lebih sangkil menemukan huruf-huruf kunci dengan menggunakan teknik analisis frekuensi.

Langkah-langkahnya sbb:

1. Misalkan panjang kunci yang sudah berhasil dideduksi adalah *n.* Setiap huruf berjarak *n* pasti dienkripsi dengan huruf kunci yang sama. Kelompokkan setiap huruf berjarak *n* bersama-sama sehingga kriptanalis memiliki n buah "pesan", masing-masing "pesan" dienkripsi dengan huruf kunci yang sama dengan *cipher* abjad-tunggal (dalam hal ini *Caesar cipher*).

2. Tiap-tiap pesan dari hasil langkah 1 dapat dipecahkan dengan metode analisis frekuensi.

3. Dari hasil langkah 2 kriptanalis dapat menyusun huruf-huruf kunci. Atau, kriptanalis dapat menerka kata yang membantu untuk memecahkan cipherteks

- Contoh:

1 2 3 4 5

LJVBQSTNEZLQMED**LJV**MAMPKAUFAVAT**LJV**DAYYVNFJQLNP**LJV**HKVTRNF**LJV**CMLKETA
LJVHUYJVSFKRFTTWEFUXVHZNP

6

Kriptogram yang berulang adalah **LJV**

Jarak **LJV** ke-1 dengan **LJV** ke-2 = 15

Jarak **LJV** ke-2 dengan **LJV** ke-3 = 15

Jarak **LJV** ke-3 dengan **LJV** ke-4 = 15

Jarak **LJV** ke-4 dengan **LJV** ke-5 = 10

Jarak **LJV** ke-5 dengan **LJV** ke-6 = 10

Faktor pembagi 15 = {3, 5, 15}

Faktor pembagi 10 = {2, 5, 10}

Irisan kedua himpunan ini = 5. Jadi, panjang kunci diperkirakan = 5

- Kelompokkan “pesan” setiap berjarak 5, dimulai dari huruf cipherteks pertama, kedua, dan seterusnya. Setiap huruf pada posisi berjarak 5 pasti dienkripsi dengan huruf kunci yang sama.

123456789$_{10}$11…

LJVBQSTNEZLQMEDLJVMAMPKAUFAVATLJVDAYYVNFJQLNPLJVHKVTRNFLJVCM
LKETALJVHUYJVSFKRFTTWEFUXVHZNP

Kelompok	Pesan	Huruf paling sering muncul
1	LSLLMFLYJLVLLLYKWV	L
2	JTQJPAJYQJTJKJJREH	J
3	VNMVKVVVLVRVEVVFFZ	V
4	BEEMAADNNHNCTHSTUN	N
5	QZDAUTAFPKFMAUFTXP	A

- Dalam Bahasa Inggris, 10 huruf yang yang paling sering muncul adalah E, T, A, O, I, N, S, H, R, dan D,

- Triplet yang paling sering muncul adalah THE. Karena **LJV** paling sering muncul di dalam cipherteks, maka dari 10 huruf tsb semua kemungkinan kata 3-huruf dibentuk dan kata yang yang cocok untuk **LJV** adalah `THE`.

- Jadi, kita dapat menerka bahwa **LJV** mungkin adalah `THE`.

- Huruf-huruf kunci lainnya dicoba dengan menerka dan menguji coba.

- Dari sini kita buat tabel yang memetakan huruf plainteks dengan cipherteks dan huruf-huruf kuncinya (ingatlah bahwa setiap nilai numerik dari huruf kunci menyatakan jumlah pergeseran huruf pada *Caesar cipher*):

Kelompok	Huruf plainteks	Huruf cipherteks	Huruf kunci
1	T	L	S (=18)
2	H	J	C (=2)
3	E	V	R (=17)
4	N	N	A (=0)
5	O	A	M (=12)

Jadi, kuncinya adalah SCRAM

- Dengan menggunakan kunci SCRAM cipherteks berhasil didekripsi menjadi:

```
THEBE ARWEN TOVER THEMO UNTAI NYEAH
THEDO GWENT ROUND THEHY DRANT THECA
TINTO THEHI GHEST SPOTH ECOUL DFIND
```

- Lakukan post-processing untuk menghasilkan kalimat yang lebih jelas:

```
THE BEAR WENT OVER THE MOUNTAIN YEAH
THE DOG WENT ROUND THE HYDRANT
THE CAT INTO THE HIGHEST SPOT HE COULD FIND
```

Varian *Vigenere Cipher*

Untuk mengatasi serangan dengan metode Kasiski, maka dibuat varian Vigenere Cipher sebagai berikut:

1. *Full Vigènere cipher*
   - Setiap baris di dalam tabel tidak menyatakan pergeseran huruf, tetapi merupakan permutasi huruf-huruf alfabet (lihat contoh table pada halaman berikut).
   - Tabel tersebut harus dirahasiakan.

   - Proses enkripsi dan dekripsi tetap sama seperti Vigenere standard:

Contoh sebuah full Vigènere square

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

2. *Auto-Key Vigènere cipher*

- Jika panjang kunci lebih kecil dari panjang plainteks, maka kunci disambung dengan plainteks tersebut.

- Misalnya,

  Pesan: `negara penghasil minyak mentah di dunia`

  Kunci: `INDO`

  maka kunci tersebut disambung dengan plainteks semula sehingga panjang kunci menjadi sama dengan panjang plainteks:

  Plainteks: `negarapenghasilminyakmentahdidunia`

  Kunci: `INDONEGARAPENGHASILMINYAKMENTAHDID`

  Cipherteks: `VRJOEEVEEGWEFOSMAVJMSZCNDMLQBDBQQD`

3. Running-Key Vigènere cipher

- Kunci adalah string yang sangat panjang yang diambil dari teks bermakna (misalnya naskah proklamasi, naskah Pembukaan UUD 1945, terjemahan ayat di dalam kitab suci, dan lain-lain).

- Misalnya,

  Pesan: `negarapenghasilminyakmentahdidunia`

  Kunci: `KERAKYATANYANGDIPIMPINOLEHHIKMATPE`

- Selanjutnya enkripsi dan dekripsi dilakukan seperti Vigenere cipher biasa.

C. Playfair Cipher

- Ditemukan oleh Sir Charles Wheatstone namun dipromosikan oleh Baron Lyon Playfair pada tahun 1854.

Sir Charles Wheatstone

Baron Lyon Playfair

Lyon Playfair, 1st Baron Playfair

8 languages

Article Talk Read Edit View history

From Wikipedia, the free encyclopedia

(Redirected from Lord Playfair)

Lyon Playfair, 1st Baron Playfair (1 May 1818 – 29 May 1898) was a British scientist and Liberal politician who was Postmaster-General from 1873 to 1874.

Early life [edit]

Playfair was born at Chunar, Bengal, the son of George Playfair (1781–1845), the chief inspector-general of hospitals in that region, and Janet Ross (1795–1862), daughter of John Ross.[1] The family was fairly middle class with strong academic roots in University of St Andrews, his grandfather being Rev Prof James Playfair, Principal of the University of St Andrews. All of Playfair's siblings were sent back to Scotland to avoid the hazards of an Indian upbringing.[1] Playfair was named after his uncle, Sir Hugh Lyon Playfair, and was educated at the University of St Andrews, the Andersonian Institute in Glasgow, and the University of Edinburgh. After going to Calcutta at the end of 1837, he became private laboratory assistant to Thomas Graham at University College London and in 1839 went to work under Justus Liebig at the University of Giessen.

The Right Honourable
The Lord Playfair
GCB PC FRS FRSE

Postmaster General

In office
18 November 1873 – 17 February 1874

Monarch	Queen Victoria
Prime Minister	William Ewart Gladstone
Preceded by	William Monsell

- *Playfair cipher* termasuk ke dalam kelompok *polygram cipher*

- Pada *polygram cipher*, satu blok huruf plainteks disubstitusi dengan satu blok huruf cipherteks.

- Jika satu blok panjangnya 2 huruf, maka ia disebut digram (*bigram*), jika 3 huruf disebut trigram, dst. *Playfair cipher* melakukan enkripsi/dekripsi dalam bentuk bigram.

  Misalnya, bigram `AS` diganti dengan **`RT`**, `BY` diganti dengan **`SL`**

- Kunci enkripsi/dekripsi pada Playfair cipher disusun berbentuk matriks berukuran 5 x 5
- Setiap elemen matriks adalah huruf-huruf alfabet A..Z
- Karena ada 25 elemen matriks, dan ada 26 huruf alfabet, maka ada satu huruf alfabet dibuang, yaitu huruf J

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

- Jumlah kemungkinan matriks kunci yang bisa dibentuk:

  25!=15.511.210.043.330.985.984.000.000

Alasan mengapa huruf J dibuang:

1. Matriks Playfair hanya 5×5

Playfair cipher menggunakan tabel 5×5, jadi hanya bisa memuat 25 huruf. Sementara alfabet latin ada 26 huruf (A–Z). Artinya satu huruf harus “dikorbankan”.

2. I dan J dianggap mirip secara historis

Secara historis, dalam alfabet Latin lama, I dan J tidak dibedakan. J dianggap hanya variasi dari I. Karena itu, Playfair menggabungkan I dan J. Huruf J dihilangkan. Semua J dalam plaintext diganti menjadi I

Contoh: JALAN → IALAN

3. Dampaknya ke proses enkripsi/dekripsi

Saat enkripsi: setiap J diperlakukan sebagai I. Saat dekripsi: huruf I bisa berarti I atau J, tergantung konteks bahasa. Ini tidak mengurangi keamanan, karena Playfair sudah bersifat substitusi digraf

Ambiguitas kecil ini masih bisa diselesaikan lewat konteks

4. Apakah bisa buang huruf lain?

Bisa saja secara teori, tapi I/J paling masuk akal secara linguistik

Sudah jadi standar konvensi Playfair. Memudahkan implementasi dan pembacaan hasil

Sumber: ChatGPT

Kunci dapat dipilih dari sebuah kalimat yang mudah diingat, misalnya:

JALAN GANESHA SEPULUH

Buang huruf yang berulang dan huruf J jika ada:

ALNGESHPU

Lalu tambahkan huruf-huruf yang belum ada (kecuali J):

ALNGESHPUBCDFIKMOQRTVWXYZ

Masukkan ke dalam bujursangkar:

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

Pesan yang akan dienkripsi diatur terlebih dahulu sebagai berikut (langkah *preprocessing*):

1. Buang semua spasi di dalam pesan
2. Ganti huruf `j` (bila ada) dengan `i`. Contoh: jalan → ialan
3. Tulis pesan dalam pasangan huruf (*bigram*).
4. Jangan sampai ada bigram dengan pasangan huruf yang sama. Jika ada, sisipkan `x` di tengahnya
5. Jika jumlah huruf di dalam pesan ganjil, tambahkan huruf `x` pada bigram terakhir

Contoh plainteks: `temui ibu nanti malam`

→ Tidak ada huruf `j`, maka langsung tulis pesan dalam pasangan huruf (setelah semua spasi dibuang):

`te mu ii bu na nt im al am`

→ Ada bigram dengan huruf yang berulang (`ii`), sisipkan huruf `x` di tengahnya:

`te mu ix ib un an ti ma la m`

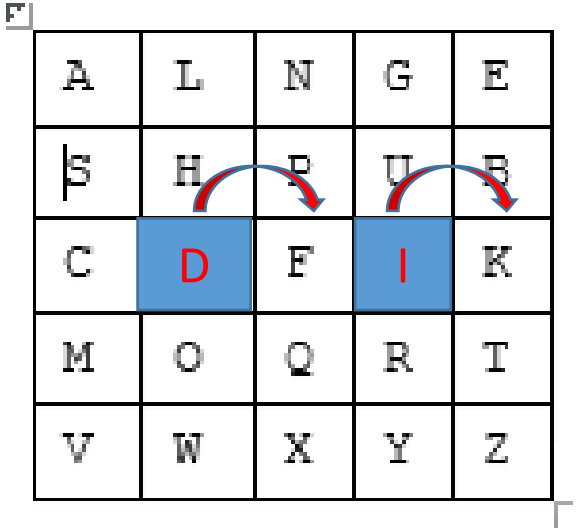
→ Tambahkan huruf `x` jika bigram terakhir hanya satu huruf:

`te mu ix ib un an ti ma la mx`

Algoritma enkripsi:

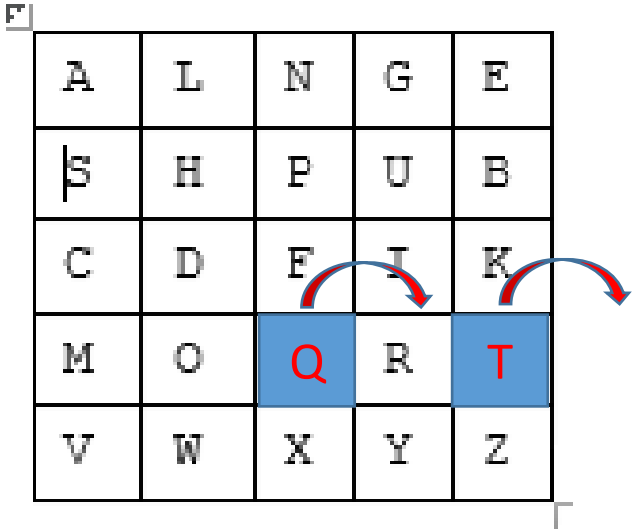
1. Jika dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kanannya (bersifat siklik).

Bigram: `di`



Cipherteks: `FK`

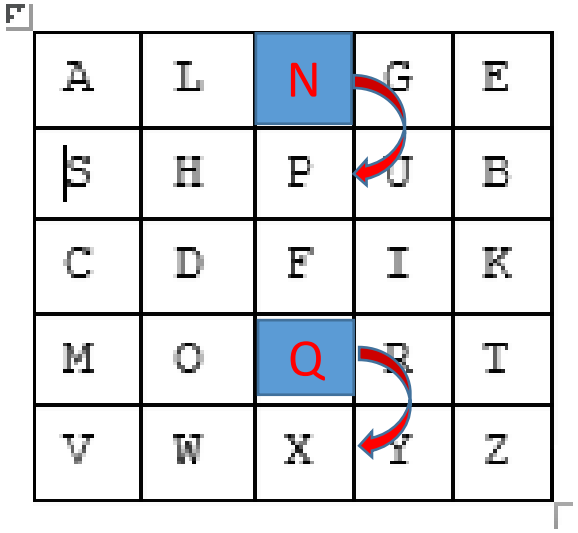
Bigram: `qt`



Cipherteks: `RM`

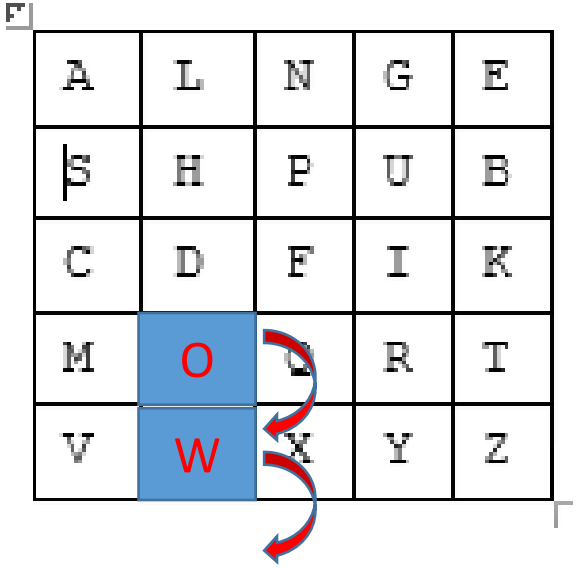
2. Jika dua huruf terdapat pada kolom kunci yang sama maka tiap huruf diganti dengan huruf di bawahnya (bersifat siklik).

Bigram: `nq`



Cipherteks: `PX`

Bigram: `ow`

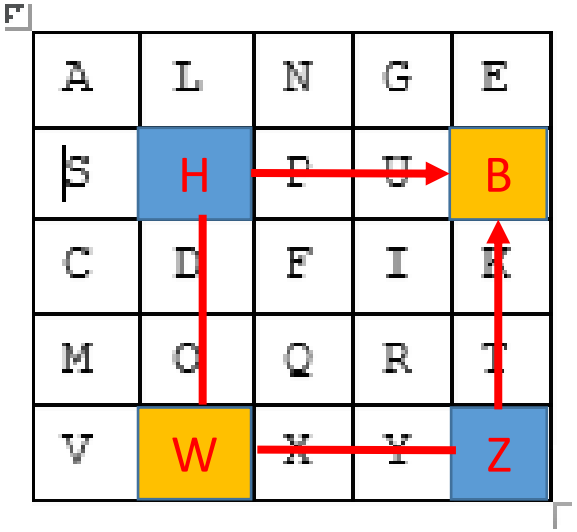


Cipherteks: `WL`

3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka:

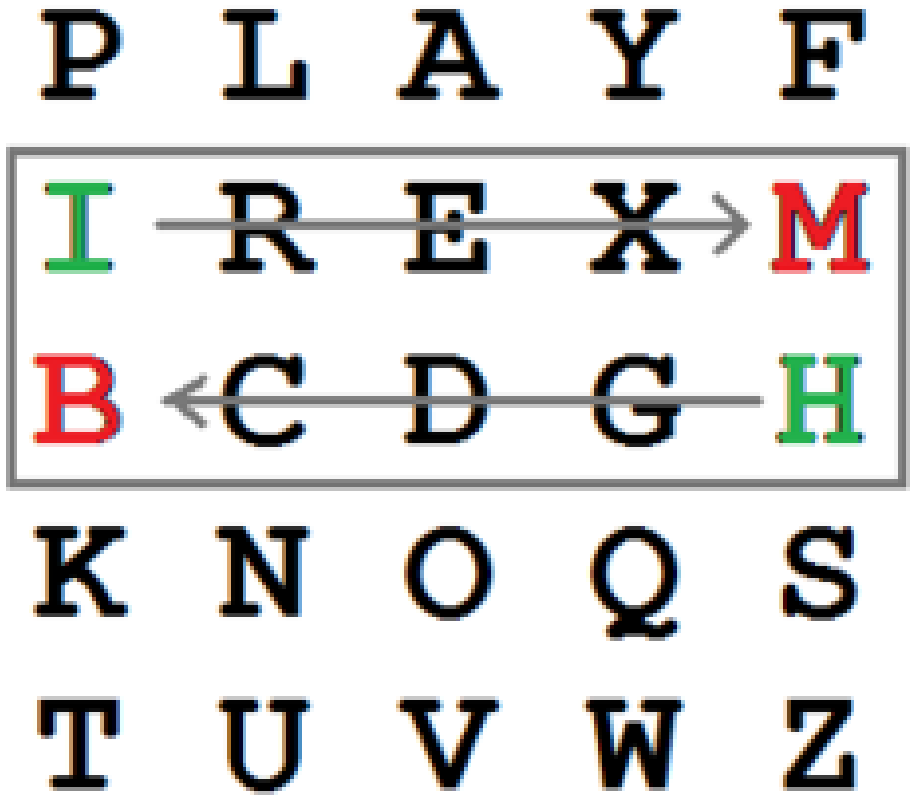
- huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua.
- huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari tiga huruf yang digunakan sampai sejauh ini.

Bigram: `hz`



Cipherteks: `BW`

Bigram IH diganti menjadi MB



Sumber gambar: Wikipedia

Plainteks: temui ibu nanti malam

Bigram: te mu ix ib un an ti ma la mx

Kunci:

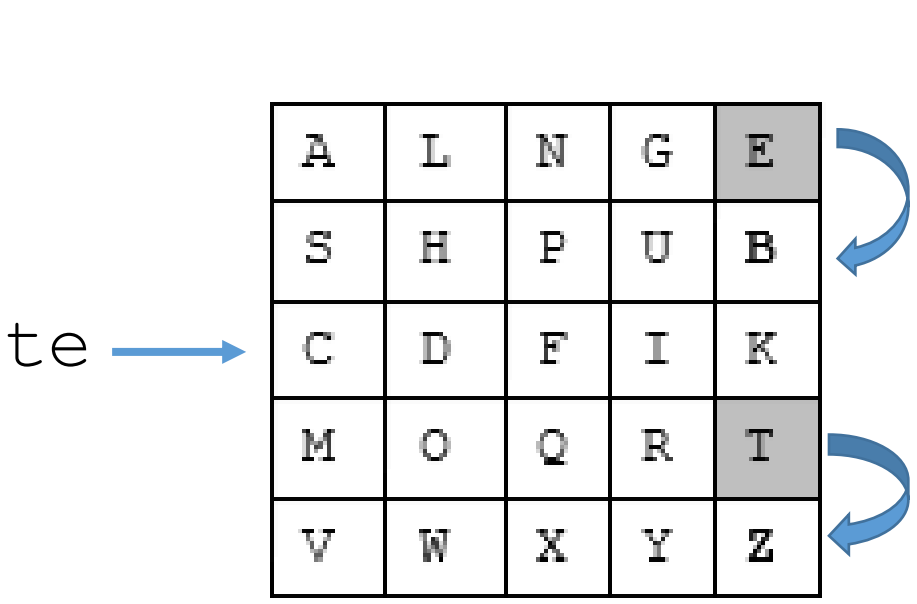
A	L	N	G	E
S	H	P	U	B
C	D	F	I	K
M	O	Q	R	T
V	W	X	Y	Z

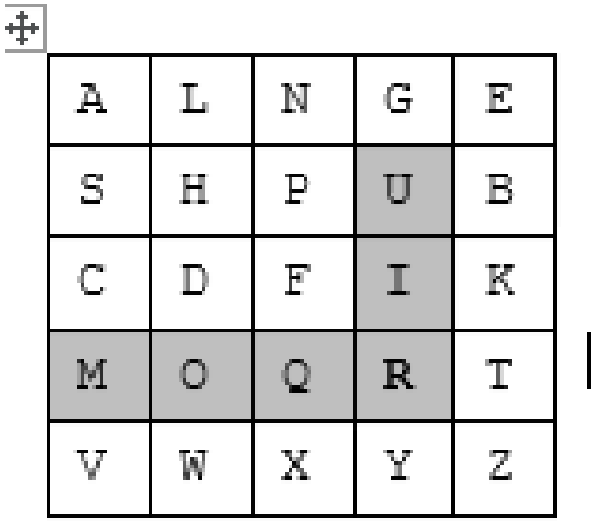
Cipherteks: ZB RS FY KU PG LG RK VS NL QV

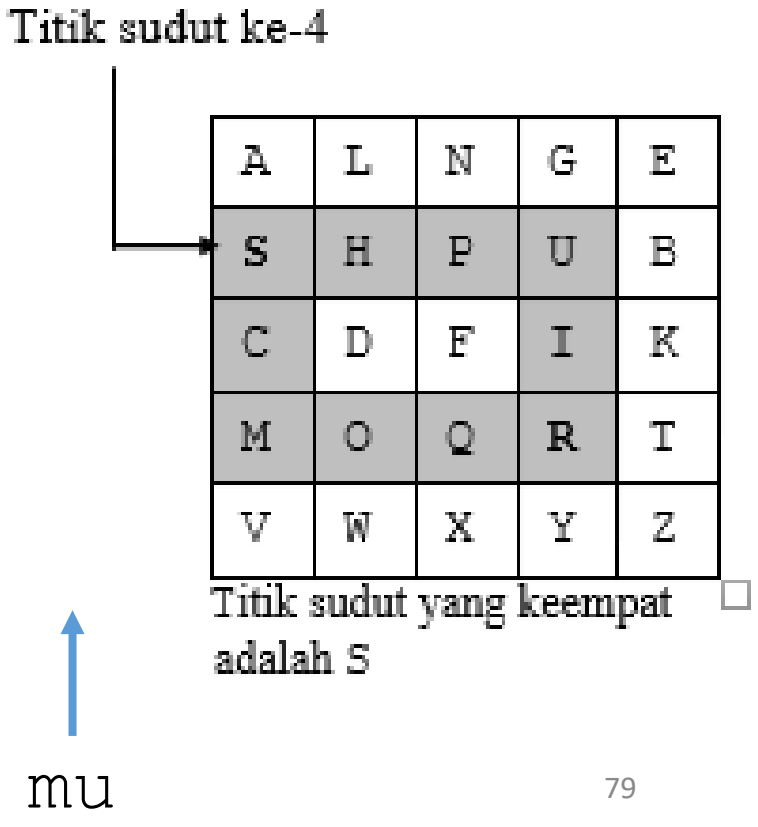
Cara enkripsinya sebagai berkut:

Bigram: te mu ix ib un an ti ma la mx

Cipherteks: ZB RS FY KU PG LG RK VS NL QV



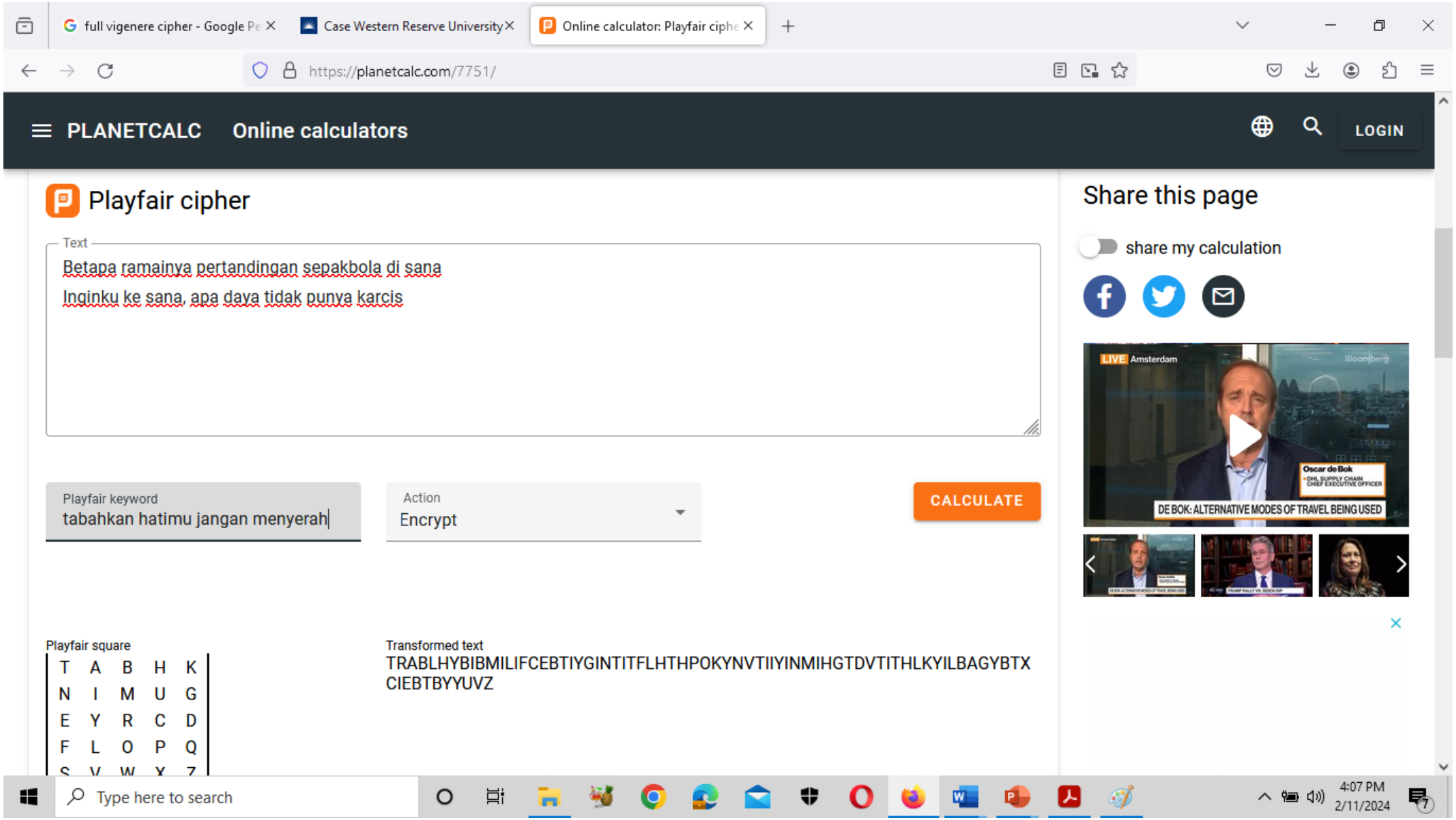




Algoritma dekripsi kebalikan dari algoritma enkripsi. Langkah-langkahnya adalah sebagai berikut:

1. Jika dua huruf terdapat pada baris bujursangkar yang sama maka tiap huruf diganti dengan huruf di kirinya.
2. Jika dua huruf terdapat pada kolom bujursangkar yang sama maka tiap huruf diganti dengan huruf di atasnya.
3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua. Huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari tiga huruf yang digunakan sampai sejauh ini.
4. Buanglah huruf X yang tidak mengandung makna.

Demo Playfair cipher online: https://planetcalc.com/7751/



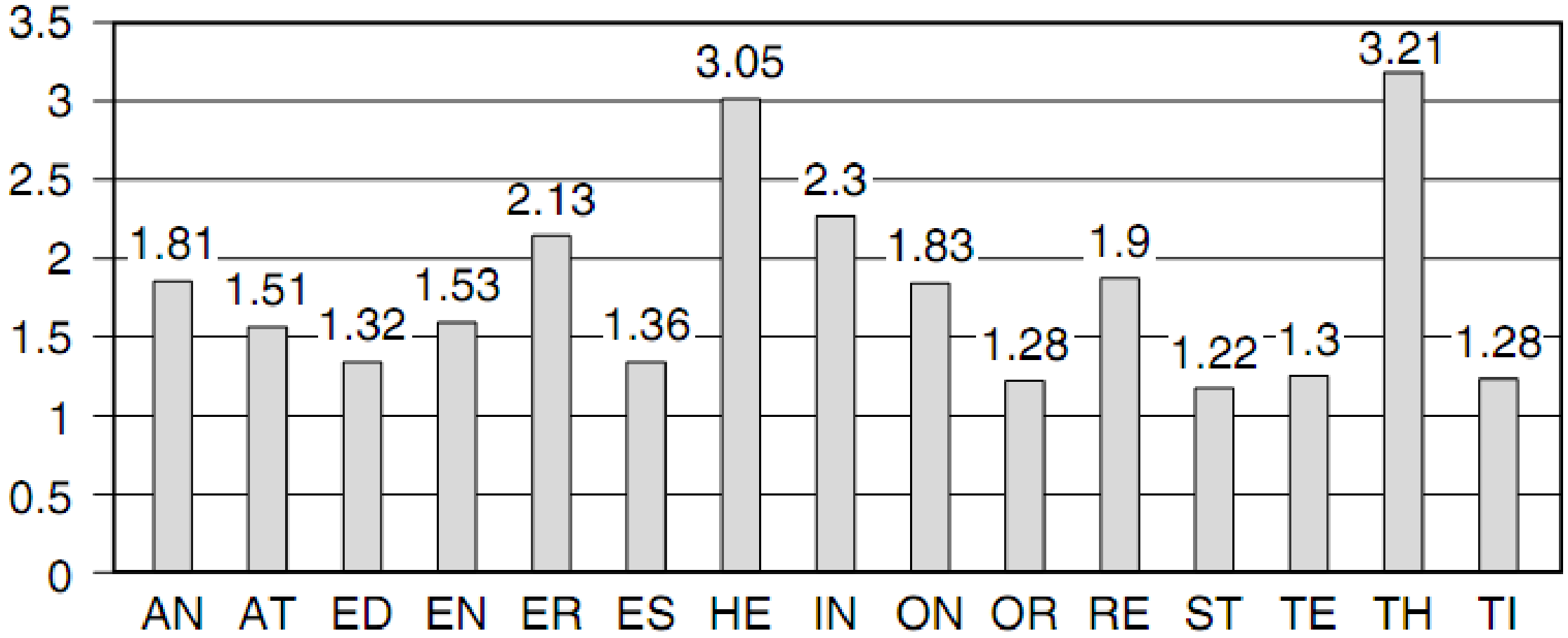
Kriptanalisis Playfair Cipher

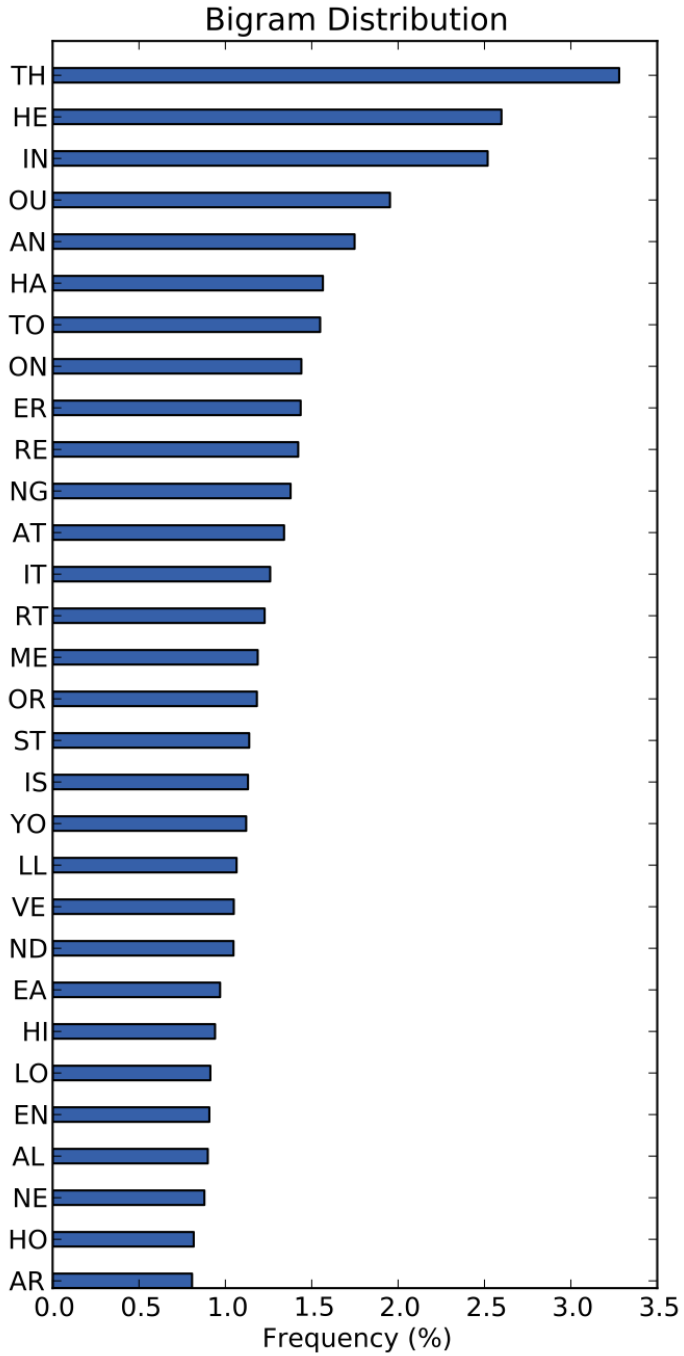
- Karena ada 26 huruf abjad, maka terdapat 26 x 26 = 677 bigram, sehingga identifikasi bigram individual lebih sukar.

- Sayangnya ukuran poligram di dalam *Playfair cipher* tidak cukup besar, hanya dua huruf sehingga *Playfair cipher* tetap tidak aman.

- Meskipun *Playfair cipher* sulit dipecahkan dengan analisis frekuensi relatif huruf-huruf, namun ia dapat dipecahkan dengan analisis frekuensi pasangan huruf.

- Dalam Bahasa Inggris kita bisa mempunyai frekuensi kemunculan pasangan huruf, misalnya pasangan huruf TH dan HE paling sering muncul.


3.5
3
2.5
2
1.5
1
0.5
0
1.81
1.51
1.32
1.53
2.13
1.36
3.05
2.3
1.83
1.28
1.9
1.22
1.3
3.21
1.28
AN
AT
ED
EN
ER
ES
HE
IN
ON
OR
RE
ST
TE
TH
TI



- Dengan menggunakan tabel frekuensi kemunculan pasangan huruf di dalam Bahasa Inggris dan cipherteks yang cukup banyak, *Playfair cipher* dapat dipecahkan.

- Kelemahan lainnya, bigram dan kebalikannya (misal AB dan BA) akan didekripsi menjadi pola huruf plainteks yang sama (misal RE dan ER). Di dalam bahasa Inggris terdapat banyak kata yang mengandung bigram terbalik seperti REceivER dan DEpartED.

- Contoh:

  - misalkan bigram yang sering muncul di dalam cipherteks adalah BM.

  - bigram yang sering muncul di dalam bahasa Inggris adalah TH

  - jadi, kita duga TH dipetakan menjadi BM

  - misalkan TH dienkripsi dengan aturan segiempat, maka jika T dan H membentuk segiempat dengan B dan M, maka posisi mereka harus konsisten dalam matriks.

  Artinya: T satu baris dengan B, dan H satu baris dengan M

  T -------- **B**

  | |

  M-------- H

  - Jika TH dienkripsi dengan aturan kolom, maka B di bawah T, M di bawah H

  - Jika TH dienkripsi dengan aturan baris, maka B di kanan T, M di kanan H

  - Mulai bangun matriks parsial 5 x 5 dengan hipotesis seperti di atas

C. Affine Cipher

- Merupakan perluasan *Caesar cipher*

  Enkripsi: $C \equiv mP + b \pmod{n}$

  Dekripsi: $P \equiv m^{-1}(C - b) \pmod{n}$

- Kunci: *m* dan *b*

Keterangan:

1. *n* adalah ukuran alfabet
2. *m* bilangan bulat yang relatif prima dengan *n*
3. *b* adalah jumlah pergeseran
4. *Caesar cipher* adalah khusus dari *affine cipher* dengan $m = 1$
5. m^{-1} adalah inversi *m* (mod *n*), yaitu $m \cdot m^{-1} \equiv 1 \pmod{n}$

- Contoh:

  Plainteks: `kripto` (10 17 8 15 19 14)

  n = 26, ambil m = 7 (7 relatif prima dengan 26)

  Enkripsi: $C \equiv 7P + 10 \pmod{26}$

  $p_1 = 10 \rightarrow c_1 \equiv 7 \cdot 10 + 10 \equiv 80 \equiv 2 \pmod{26}$ (huruf 'C')

  $p_2 = 17 \rightarrow c_2 \equiv 7 \cdot 17 + 10 \equiv 129 \equiv 25 \pmod{26}$ (huruf 'Z')

  $p_3 = 8 \rightarrow c_3 \equiv 7 \cdot 8 + 10 \equiv 66 \equiv 14 \pmod{26}$ (huruf 'O')

  $p_4 = 15 \rightarrow c_4 \equiv 7 \cdot 15 + 10 \equiv 115 \equiv 11 \pmod{26}$ (huruf 'L')

  $p_5 = 19 \rightarrow c_5 \equiv 7 \cdot 19 + 10 \equiv 143 \equiv 13 \pmod{26}$ (huruf 'N')

  $p_6 = 14 \rightarrow c_6 \equiv 7 \cdot 14 + 10 \equiv 108 \equiv 4 \pmod{26}$ (huruf 'E')

  Cipherteks: `CZOLNE`

- Dekripsi:

  - Mula-mula hitung m^{-1} yaitu $7^{-1} \pmod{26}$ dengan memecahkan $7x \equiv 1 \pmod{26}$

    Solusinya: $x \equiv 15 \pmod{26}$ sebab $7 \cdot 15 = 105 \equiv 1 \pmod{26}$.

  - Jadi, $P \equiv 15\,(C - 10) \pmod{26}$

$c_1 = 2 \quad \rightarrow p_1 \equiv 15 \cdot (2 - 10) = -120 \equiv 10 \pmod{26}$ (huruf 'k')

$c_2 = 25 \rightarrow p_2 \equiv 15 \cdot (25 - 10) = 225 \equiv 17 \pmod{26}$ (huruf 'r')

$c_3 = 14 \rightarrow p_3 \equiv 15 \cdot (14 - 10) = 60 \equiv 8 \pmod{26}$ (huruf 'i')

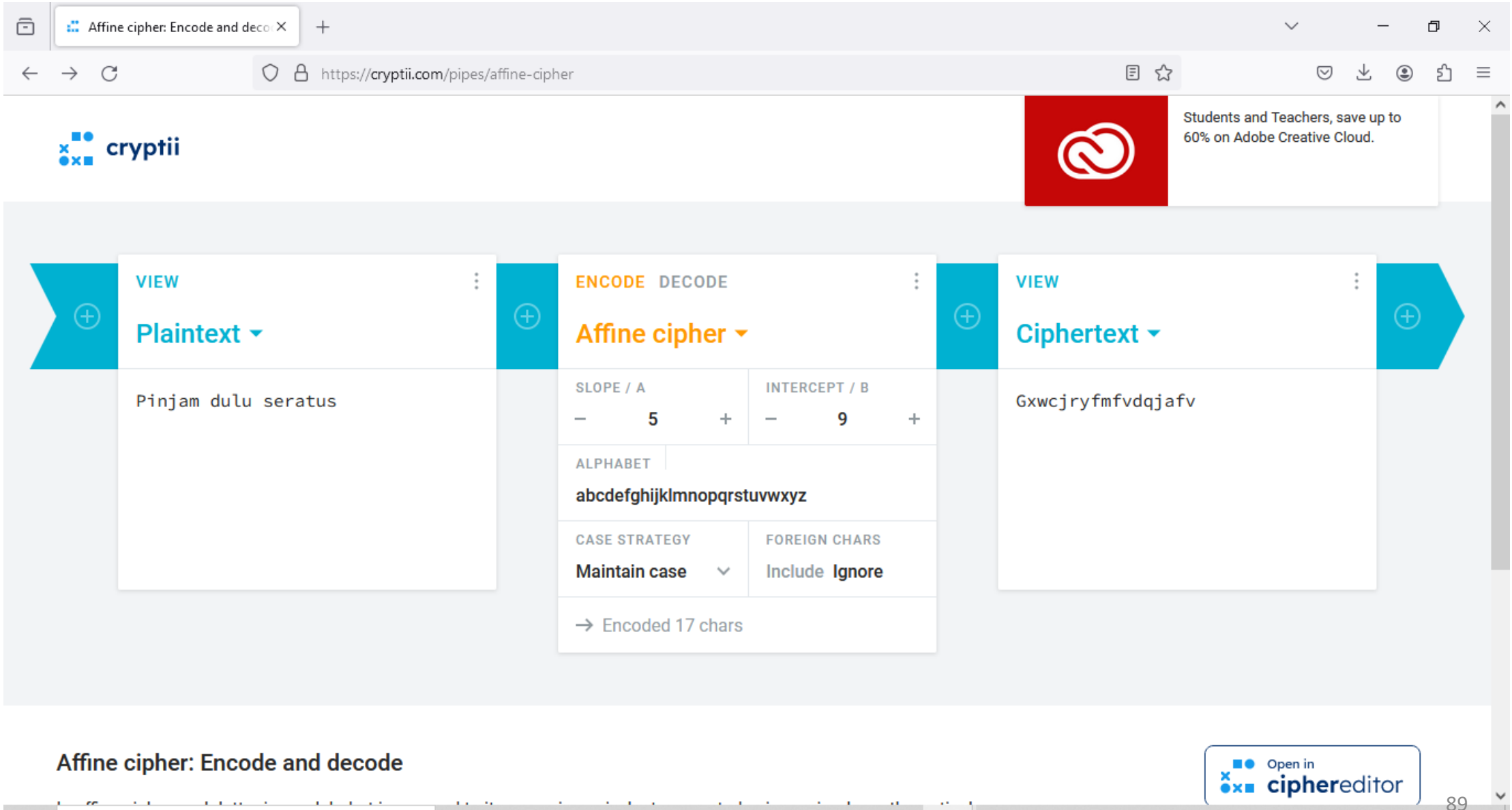
$c_4 = 11 \rightarrow p_4 \equiv 15 \cdot (11 - 10) = 15 \equiv 15 \pmod{26}$ (huruf 'p')

$c_5 = 13 \rightarrow p_5 \equiv 15 \cdot (13 - 10) = 45 \equiv 19 \pmod{26}$ (huruf 't')

$c_6 = 4 \quad \rightarrow p_6 \equiv 15 \cdot (4 - 10) = -90 \equiv 14 \pmod{26}$ (huruf 'o')

Plainteks yang diungkap kembali: `kripto`

Demo affine cipher online: https://cryptii.com/pipes/affine-cipher



Kriptanalisis Affine Cipher

- *Affine Cipher:*

  Enkripsi: $C \equiv mP + b \pmod{n}$

  Dekripsi: $P \equiv m^{-1}(C - b) \pmod{n}$

- *Affine cipher* mudah diserang dengan *known-plaintext attack* (sebagian plainteks dari pesan diketahui)

- Misalkan kriptanalis mempunyai dua buah plainteks, P_1 dan P_2, yang berkoresponden dengan cipherteks C_1 dan C_2,

- maka m dan b mudah dihitung dari buah kekongruenan simultan berikut ini:

  $C_1 \equiv mP_1 + b \pmod{n}$

  $C_2 \equiv mP_2 + b \pmod{n}$

- Contoh: Misalkan kriptanalis menemukan

  cipherteks **C** dan plainteks berkorepsonden K

  cipherteks **E** dan plainteks berkoresponden O.

- Kriptanalis m dan n dari kekongruenan berikut:

  $2 \equiv 10m + b \pmod{26}$ (i)

  $4 \equiv 14m + b \pmod{26}$ (ii)

- Kurangkan (ii) dengan (i), menghasilkan

  $2 \equiv 4m \pmod{26}$ (iii)

  Solusi: $m = 7$

  Substitusi $m = 7$ ke dalam (i),

  $2 \equiv 70 + b \pmod{26}$ (iv)

  Solusi: $b = 10$.

- Jika pasangan plainteks dan cipherteks tidak tersedia, maka Affine Cipher masih bisa diserang dengan *exhaustive key search.*

- sebab ada 25 pilihan untuk *b* dan 12 buah nilai *m* yang relatif prima dengan 26 (yaitu 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, dan 25).

- Dengan mencoba sebanyak 25 x 12 = 300 susunan *b* dan *m*, maka nilai *b* dan *m* yang benar dapat ditemukan dengan mudah, karena 300 susunan *b* dan *m* adalah jumlah yang sangat sedikit.

- Salah satu cara memperbesar faktor kerja untuk *exhaustive key search:* adalah dengan cara enkripsi tidak dilakukan terhadap huruf individual, tetapi dalam blok huruf.

- Misal, pesan `kriptografi` dipecah menjadi kelompok 4-huruf:

  `krip togr afi`

  (ekivalen dengan 10170815 19140617 000508, dengan memisalkan

  'A' = 0, 'B' = 1, …, 'Z' = 25)

- Nilai terbesar yang dapat muncul untuk merepresentasikan blok: 25252525 (ZZZZ), maka 25252525 dapat digunakan sebagai modulus *n*.
- Nilai *m* yang relatif prima dengan 25252525, misalnya 21035433,
- *b* dipilih antara 1 dan 25252525, misalnya 23210025.

- Fungsi enkripsi menjadi:

$$C \equiv 21035433P + 23210025 \pmod{25252525}$$

- Fungsi dekripsi, setelah dihitung, menjadi

$$P \equiv 5174971\,(C - 23210025) \pmod{25252525}$$

D. Hill Cipher

- Dikembangkan oleh Lester Hill (1929), termasuk ke dalam polygram cipher yang berbasis aljabar linier
- Menggunakan *m* buah persamaan linier
- Untuk *m* = 3 (enkripsi setiap 3 huruf), sistem persamaan linernya menjadi:

$$C_1 = (k_{11}\, p_1 + k_{12} p_2 + k_{13}\, p_3) \bmod 26$$
$$C_2 = (k_{21}\, p_1 + k_{22} p_2 + k_{23}\, p_3) \bmod 26$$
$$C_3 = (k_{31}\, p_1 + k_{32} p_2 + k_{33}\, p_3) \bmod 26$$

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

C = KP

Lester S. Hill

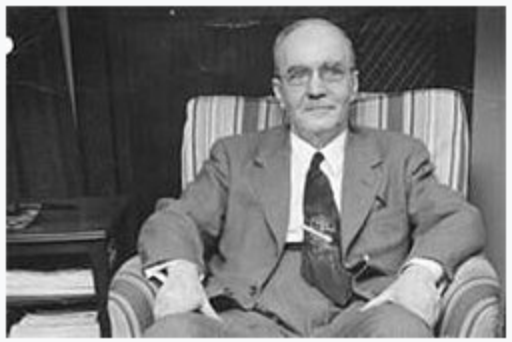
6 languages

Article Talk Read Edit View history

From Wikipedia, the free encyclopedia

Lester S. Hill (1891–1961) was an American mathematician and educator who was interested in applications of mathematics to communications. He received a bachelor's degree (1911) and a master's degree (1913) from Columbia College and a Ph.D. from Yale University (1926). He taught at the University of Montana, Princeton University, the University of Maine, Yale University, and Hunter College. Among his notable contributions was the Hill cipher. He also developed methods for detecting errors in telegraphed code numbers and wrote two books.

Dr.
Lester S. Hill



Lester S. Hill on May 16, 1956

Born	Lester Sanders Hil[1] January 18, 1891 New York City
Died	January 9, 1961 (aged 69)[2][3] Bronxville, New York[2]
Occupations	mathematician and cryptographer
Known for	the Hill cipher (1929)
Notable work	Cryptography in an Algebraic Alphabet (1929)[4]

References [edit]

1. ^ Christensen, Chris (2014). "Lester Hill Revisited". *Cryptologia*. **38** (4): 300–301. doi:10.1080/01611194.2014.915260. S2CID 45603857. Retrieved November 9, 2018 – via ResearchGate.
2. ^ *a b* "Dr. Lester S. Hill". *Chicago Tribune*. January 10, 1961. p. 10. "Dr. Lester S. Hill, 70, mathematician and cryptographer, died today in Lawrence hospital after a long illness. Hill was commended for application, of higher mathematics to the construction of secret codes."

Letter	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
Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- Contoh:

$$\mathbf{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Plainteks: `paymoremoney`

Enkripsi tiga huruf pertama: `pay = (15, 0, 24)`

$$\text{Cipherteks: } C = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \texttt{LNS}$$

Cipherteks selengkapnya: `LNSHDLEWMTRW`

- Dekripsi perlu menghitung $\mathbf{K}^{-1}$ sedemikian sehingga $\mathbf{K}\mathbf{K}^{-1} = \mathbf{I}$ (**I** matriks identitas).

$$\mathbf{K}^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

sebab

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Cara menghitung matriks invers 2 x 2:

$$K = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad K^{-1} = \frac{1}{\det(K)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$= \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Contoh: $K = \begin{pmatrix} 3 & 10 \\ 15 & 9 \end{pmatrix}$

det(K) = (3)(9) – (15)(10) = 27 – 150 = –123 mod 26 = 7

$$\mathbf{K}^{-1} = \frac{1}{7}\begin{pmatrix} 3 & 10 \\ 15 & 9 \end{pmatrix} = 7^{-1}\begin{pmatrix} 9 & -10 \\ -15 & 3 \end{pmatrix}$$

$$= 15\begin{pmatrix} 9 & -10 \\ -15 & 3 \end{pmatrix} = 15\begin{pmatrix} 9 & 16 \\ 11 & 3 \end{pmatrix} = \begin{pmatrix} 135 & 240 \\ 165 & 45 \end{pmatrix} \bmod 26 = \begin{pmatrix} 5 & 6 \\ 9 & 19 \end{pmatrix}$$

Keterangan (ingat kembali teori bilangan di dalam Matematika Diskrit):

(i) $7^{-1} \pmod{26} \equiv 15$, karena (7)(15) = 105 mod 26 = 1

(ii) $-10 \equiv 16 \pmod{26}$

(iii) $-15 \equiv 11 \pmod{26}$)

Periksa bahwa:

$$\mathbf{K}\mathbf{K}^{-1} = \begin{pmatrix} 3 & 10 \\ 15 & 9 \end{pmatrix}\begin{pmatrix} 5 & 6 \\ 9 & 19 \end{pmatrix} = \begin{pmatrix} 3\cdot 5+10\cdot 9 & 3\cdot 6+10\cdot 19 \\ 15\cdot 5+9\cdot 9 & 15\cdot 6+9\cdot 19 \end{pmatrix}$$

$$= \begin{pmatrix} 105 & 208 \\ 156 & 261 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Untuk matriks 3 x 3, matriks dapat dicari dengan metode Eliminasi Gauss-Jordan, atau dihitung langsung dengan rumus berikut::

$$\mathbf{K} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}, \quad \mathbf{K}^{-1} = \frac{1}{\det(\mathbf{K})} \begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix}^T = \frac{1}{\det(\mathbf{K})} \begin{pmatrix} A & D & G \\ B & E & H \\ C & F & I \end{pmatrix}$$

- yang dalam hal ini,

$A = (ei - hf)$ $B = -(di - fg)$ $C = (dh - eg)$

$D = -(bi - hc)$ $E = (ai - cg)$ $F = -(ah - bg)$

$G = (bf - ec)$ $H = -(af - cd)$ $I = (ae - bd)$

dan

$$\det(\mathbf{K}) = aA + bB + cC$$

- Dekripsi:

$$\mathbf{P} = \mathbf{K}^{-1}\,\mathbf{C}$$

Cipherteks: LNS atau C = (11, 13, 18)

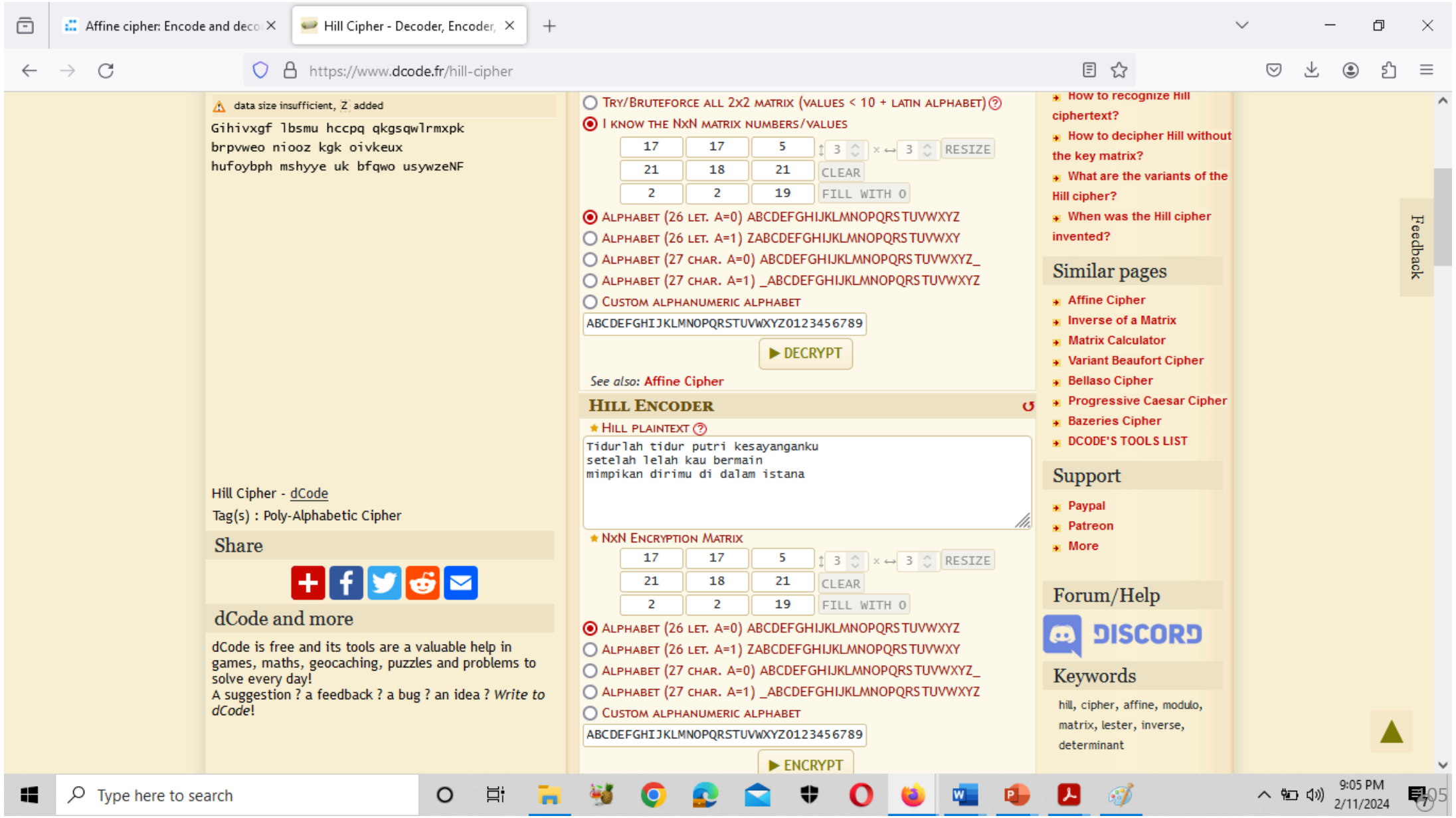
Plainteks:
$$\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \begin{pmatrix} 431 \\ 494 \\ 570 \end{pmatrix} \bmod 26 = \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix}$$

C = (15, 0, 24) = (P, A, Y)

- Kekuatan Hill cipher terletak pada penyembunyian frekuensi huruf tunggal

- Huruf plainteks yang sama belum tentu dienkripsi menjadi huruf cipherteks yang sama.

Demo Hill Cipher online: https://www.dcode.fr/hill-cipher



Affine cipher: Encode and deco | Hill Cipher - Decoder, Encoder, | https://www.dcode.fr/hill-cipher

Search for a tool

★ Search a tool on dCode by keywords:

e.g. type 'sudoku'

★ Browse the full dCode tools' list

Results

GIHIVXGFLBSM…ENF

17	17	5
21	18	21
2	2	19

=ABCDEFGHIJKLMNOPQRSTUVWXYZ

Tidurlah tidur putri kesayanganku
setelah lelah kau bermain
mimpikan dirimu di dalam istanaZZ

Hill Cipher

Cryptography › Poly-Alphabetic Cipher › Hill Cipher

Hill Decoder

★ Hill ciphertext

Gihivxgf lbsmu hccpq qkgsqwlrmxpk
brpvweo niooz kgk oivkeux
hufoybph mshyye uk bfqwo usywzeNF

- Try/Bruteforce all 2x2 matrix (values < 10 + latin alphabet)
- I know the NxN matrix numbers/values

17	17	5
21	18	21
2	2	19

3 × 3 RESIZE | CLEAR | FILL WITH 0

- Alphabet (26 let. A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Alphabet (26 let. A=1) ZABCDEFGHIJKLMNOPQRSTUVWXY
- Alphabet (27 char. A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ_
- Alphabet (27 char. A=1) _ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Custom alphanumeric alphabet

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

▶ DECRYPT

See also: Affine Cipher

Summary

Similar pages

- Affine Cipher
- Inverse of a Matrix
- Matrix Calculator
- Variant Beaufort Cipher
- Bellaso Cipher

Feedback

Type here to search

9:06 PM 2/11/2024

Kriptanalisis Hill Cipher

Enkripsi: C = KP mod 26
Dekripsi: P = K^{-1}C mod 26

- *Hill cipher* mudah dipecahkan dengan *known-plaintext attack*.
- Misalkan untuk *Hill cipher* dengan *m* = 2 diketahui:
  - P = (19, 7) → *C* = (0, 23)
  - P = (4, 17) → *C* = (12, 6)
  - Jadi, **K**(19, 7) = (0, 23) dan **K**(4, 17) = (12, 6)

$$\begin{pmatrix} 0 & 12 \\ 23 & 6 \end{pmatrix} = K \begin{pmatrix} 19 & 4 \\ 7 & 17 \end{pmatrix} \bmod 26 \qquad \rightarrow \mathbf{K} = \mathbf{CP}^{-1} \bmod 26$$

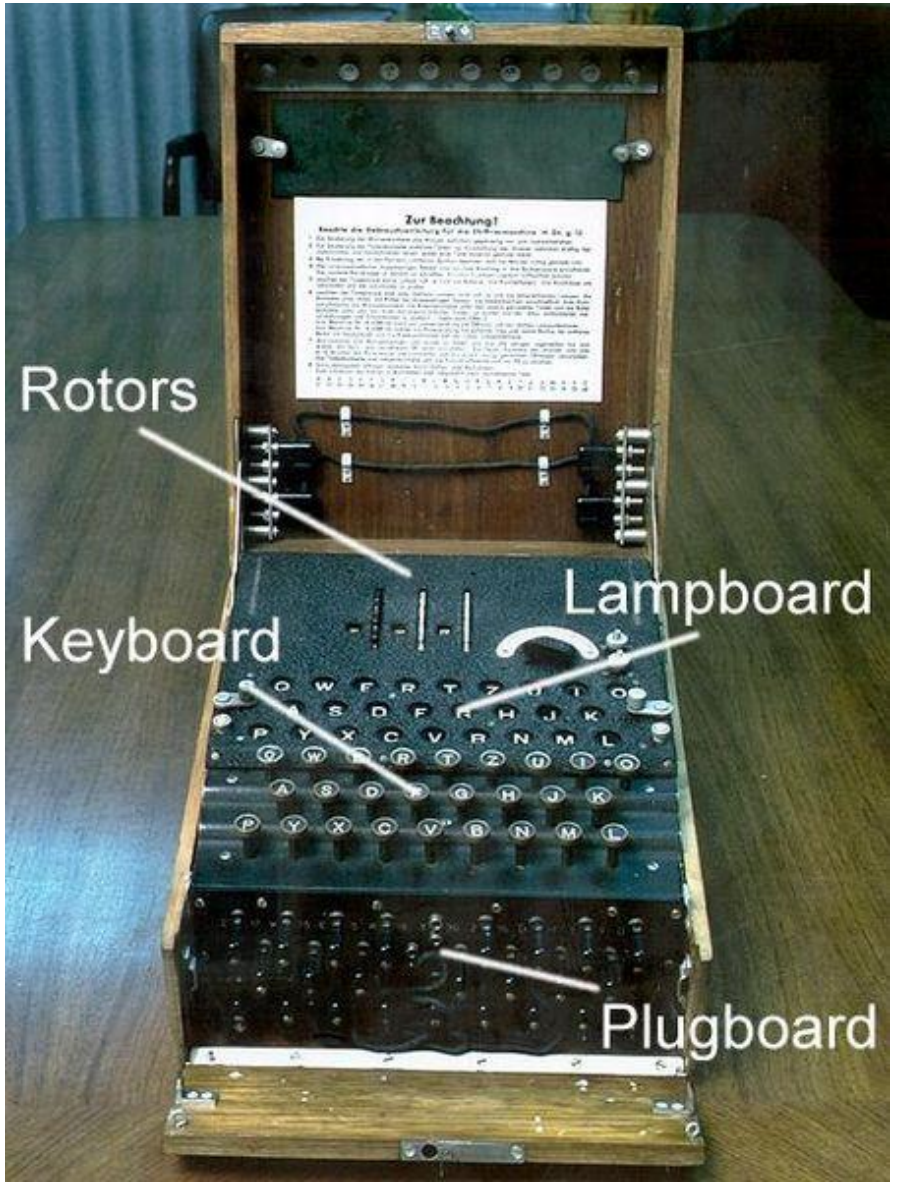
$\leftarrow C \rightarrow \qquad \leftarrow P \rightarrow$

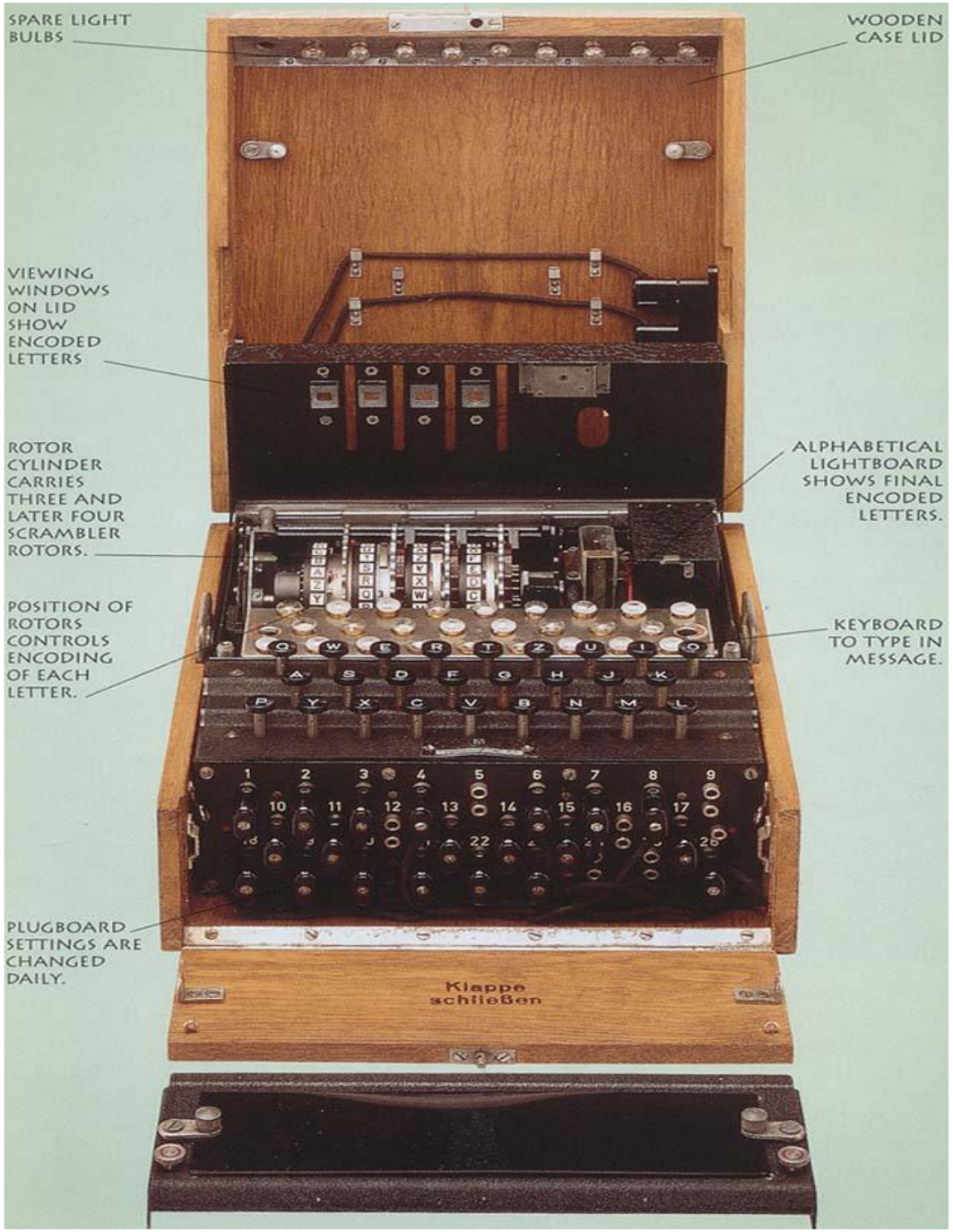
- Matriks balikan dari *P* adalah P^{-1} = $\begin{pmatrix} 19 & 4 \\ 7 & 17 \end{pmatrix}^{-1} \bmod 26 = \begin{pmatrix} 25 & 14 \\ 5 & 5 \end{pmatrix}$
- Sehingga

$$\mathbf{K} = \mathbf{CP}^{-1} \bmod 26 = \begin{pmatrix} 0 & 12 \\ 23 & 6 \end{pmatrix} \begin{pmatrix} 25 & 14 \\ 5 & 5 \end{pmatrix} \bmod 26 = \begin{pmatrix} 8 & 8 \\ 7 & 14 \end{pmatrix}$$

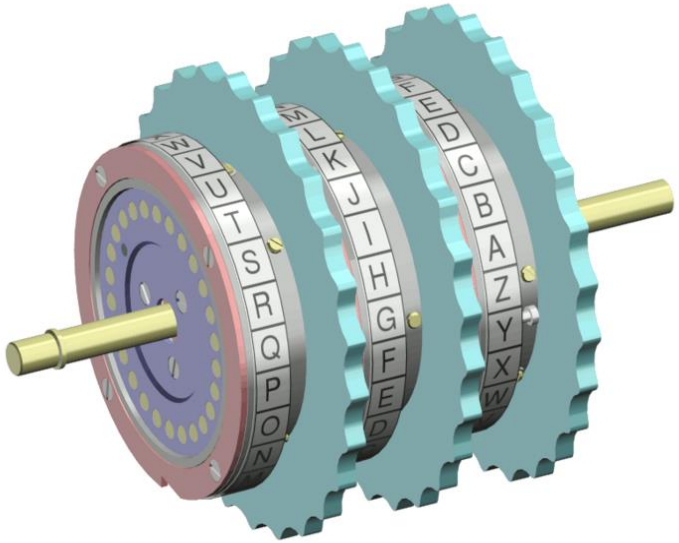
E. Enigma Cipher

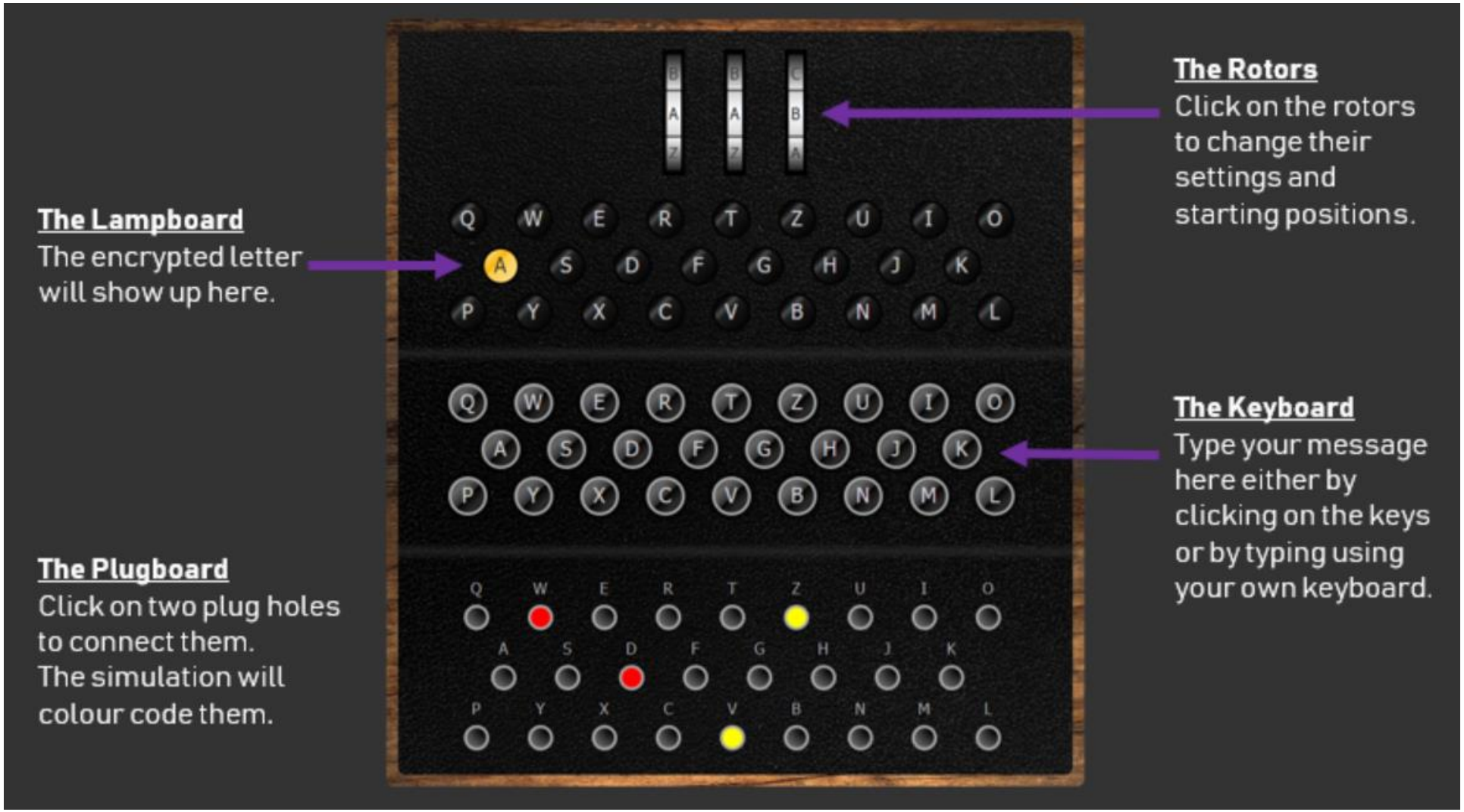
- Enigma adalah mesin enkripsi elektromekanik untuk melakukan enkripsi dan dekripsi.
- Ditemukan dan dipatenkan oleh insinyur Jerman, Arthur Scherbius, untuk tujuan komersil, diplomatik, dan militer
- Menjadi terkenal karena digunakan oleh tantara Nazi Jerman selama Perang Dunia II untuk mengenkripsi/dekripsi pesan-pesan militer.
- Enigma berasal dari bahasa latin, *enigmae*, yang artinya teka-teki.





Enigma Rotors





Sumber gambar: https://www.101computing.net/enigma/enigma-instructions.html

Operator mengetik huruf plainteks pada keyboard, lalu menyalin ulang huruf cipherteks yang menyala pada *lampboard*. Cipherteks dikirim ke penerima pesan

- Enigma menggunakan sistem *rotor* (roda berputar) untuk membentuk huruf cipherteks yang berubah-ubah.

- Setiap rotor melakukan substitusi abjad-tunggal (*monoalphabetic cipher*).

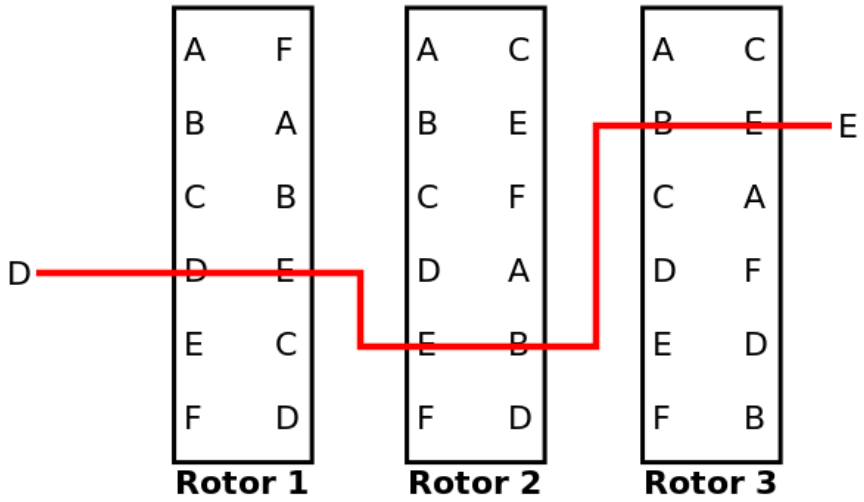
- Hasil substitusi oleh suatu rotor menjadi huruf input untuk operasi substitusi rotor selanjutnya.

- Hasil substitusi oleh rotor terakhir menjadi huruf cipherteks.

- Setiap kali sebuah huruf dienkripsi oleh sebuah rotor, *rotor* berputar satu huruf untuk membentuk huruf cipherteks baru bagi huruf plainteks berikutnya.

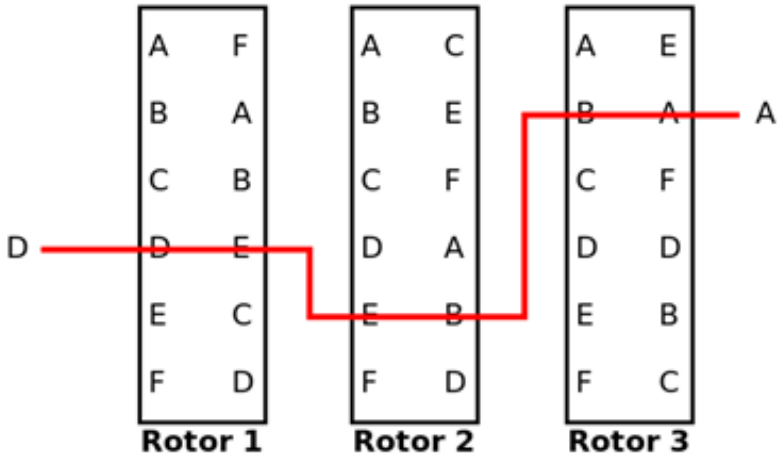
- Setelah berputar 26 huruf rotor kembali pada posisi semula. Jadi, diperoleh cipher abjad-majemuk dengan periode 26.

- Sebagai contoh, tinjau 3 rotor yang disederhanakan menjadi hanya 6 huruf alfabet:



- Misalkan huruf plainteks D ditekan pada keyboard
- Huruf D dienkripsi oleh roto pertama menjadi E
- Huruf E menjadi input untuk rotor kedua, dienkripsi menjadi B
- Huruf B menjadi input untuk rotor ketiga, dienkripsi menjadi E
- Jadi, huruf D dienkripsi menjadi E

- Setelah D dienkripsi menjadi E, rotor ketiga bergeser satu huruf.
- Jika D dienkripsi kembali, maka hasilnya adalah A



Sumber: https://www.usna.edu/Users/cs/nchamber/courses/ic211/s19/lab/l04/realenigma.html

- Model mesin enigma ada yang menggunakan 3 rotor atau 4 rotor, setiap rotor melakukan operasi substitusi cipher abjad-tunggal.

- Untuk mesin enigma 4-rotor, berarti terdapat $26 \times 26 \times 26 \times 26 = 456.976$ kemungkinan huruf cipherteks sebagai pengganti huruf plainteks sebelum terjadi perulangan urutan cipherteks.

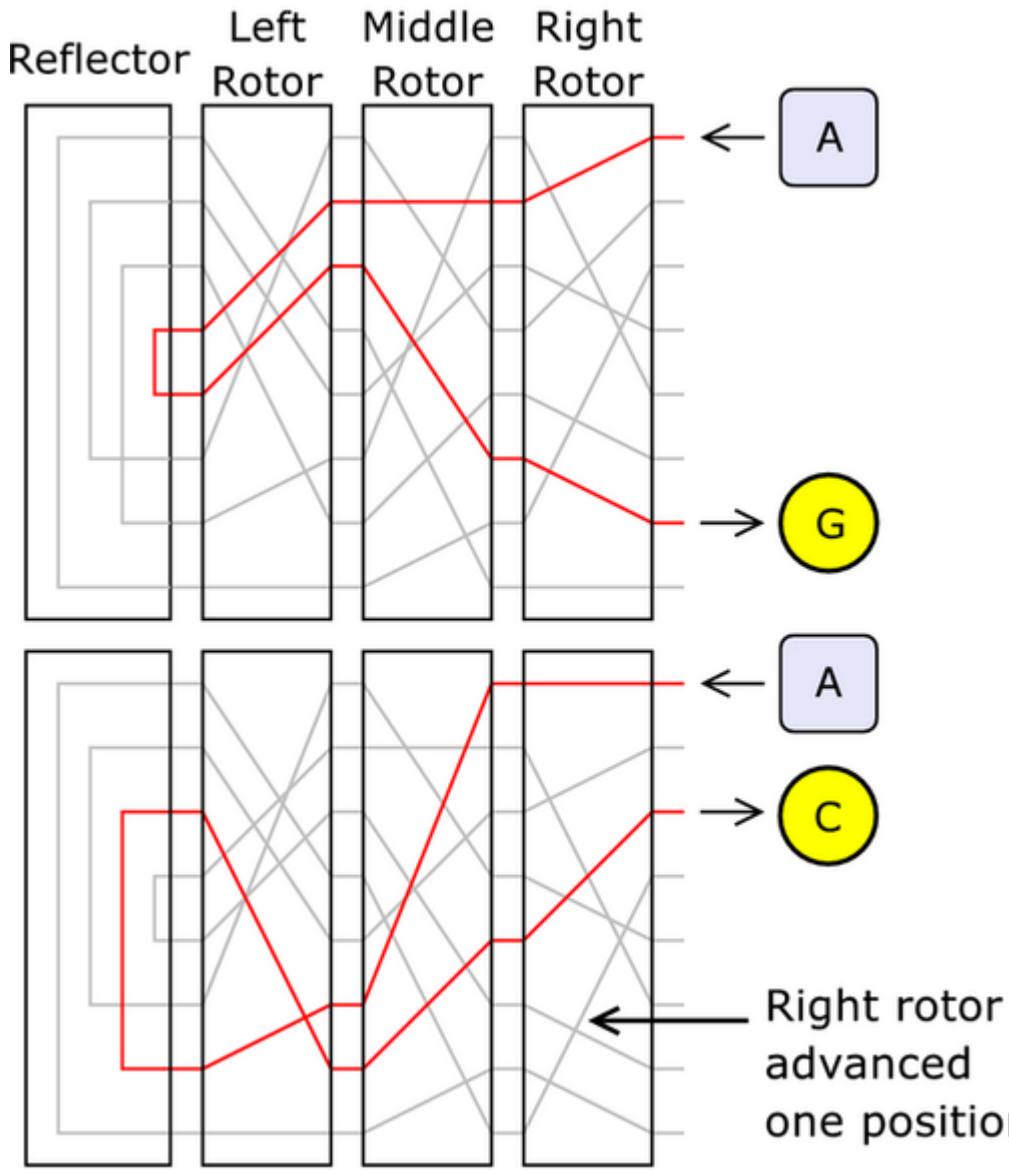
- Setiap kali sebuah huruf selesai disubstitusi, *rotor* pertama bergeser satu huruf.

- Setiap kali *rotor* pertama selesai bergeser 26 kali, rotor kedua bergeser satu huruf. Setelah rotor kedua bergeser 26 kali, rotor ketiga bergeser satu huruf. Setelah rotor ketiga bergeser 26 kali, rotor keempat bergeser satu huruf.

- Posisi awal keempat *rotor* dapat di-*set*; dan posisi awal ini menyatakan kunci dari Enigma.

- Kriptanalisis mesin Enigma pertama kali ditemukan pada tahun 1932 oleh kriptografer Polandia, yaitu Marian Rejewski, Jerzy Różycki dan Henryk Zygalski.

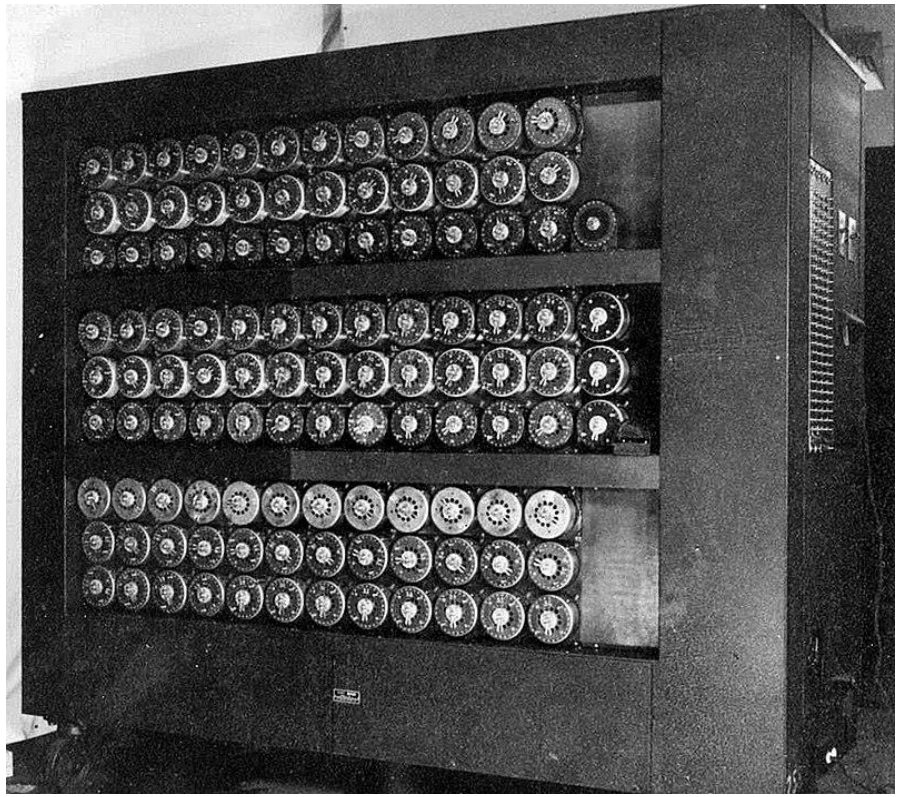
- Pemerintahan Nazi Jerman kemudian mendesain ulang mesin Enigma pada tahun 1939 dengan menambahkan *plugboard* dan *refelector*, sehingga proses enkripsi menjadi lebih kompleks. Metode kriptanalisis Enigma sebelumnya tidak dapat digunakan lagi.



- Jerman sangat percaya diri bahwa Enigma tidak akan dapat dipecahkan.

- Namun, dengan bantuan Polandia, Perancis dan Inggris kemudian membuat mesin pemecah Enigma baru ini, yang diberi nama *bombe*.

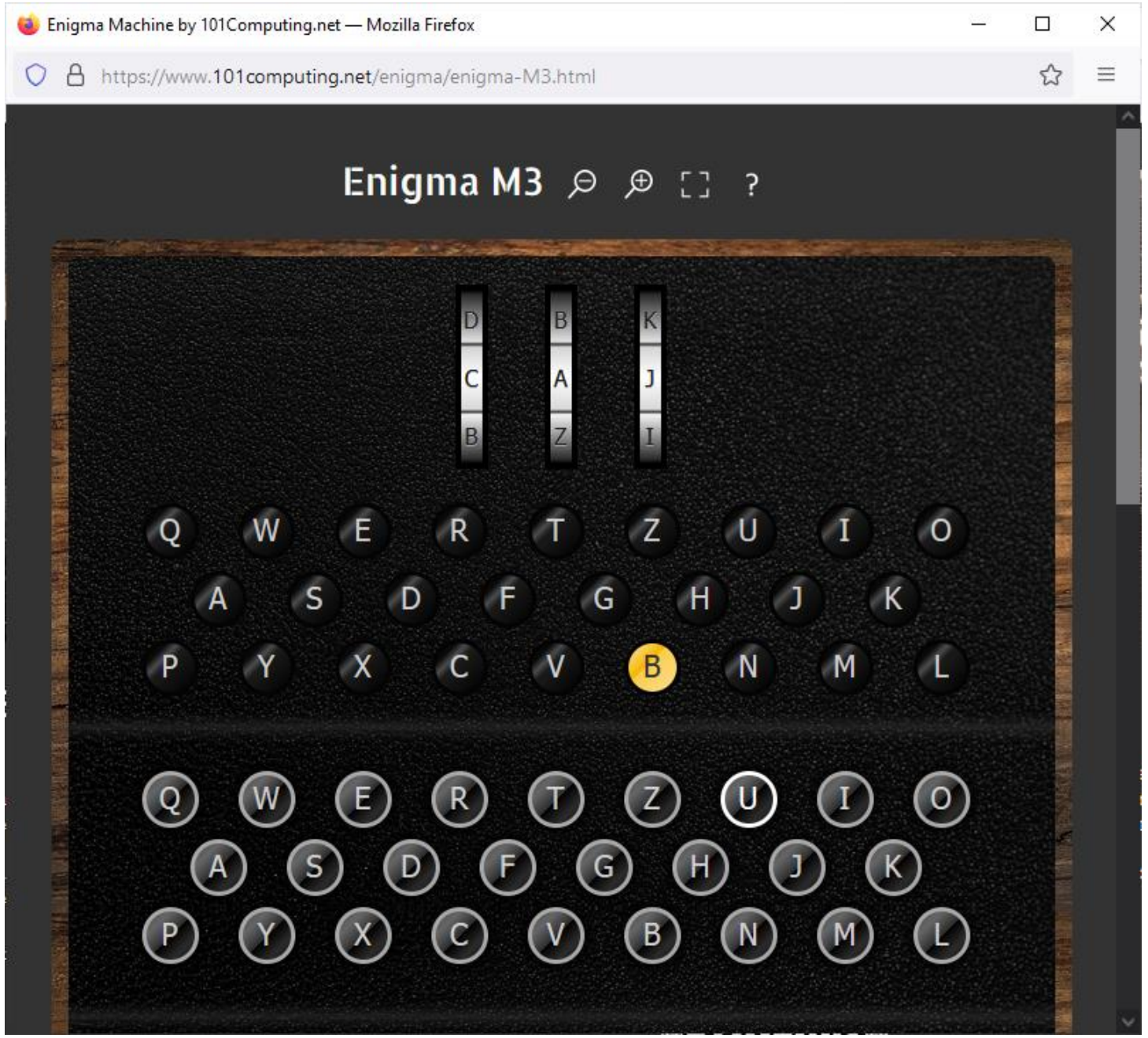
- *Bombe* dirancang oleh Alan Turing.

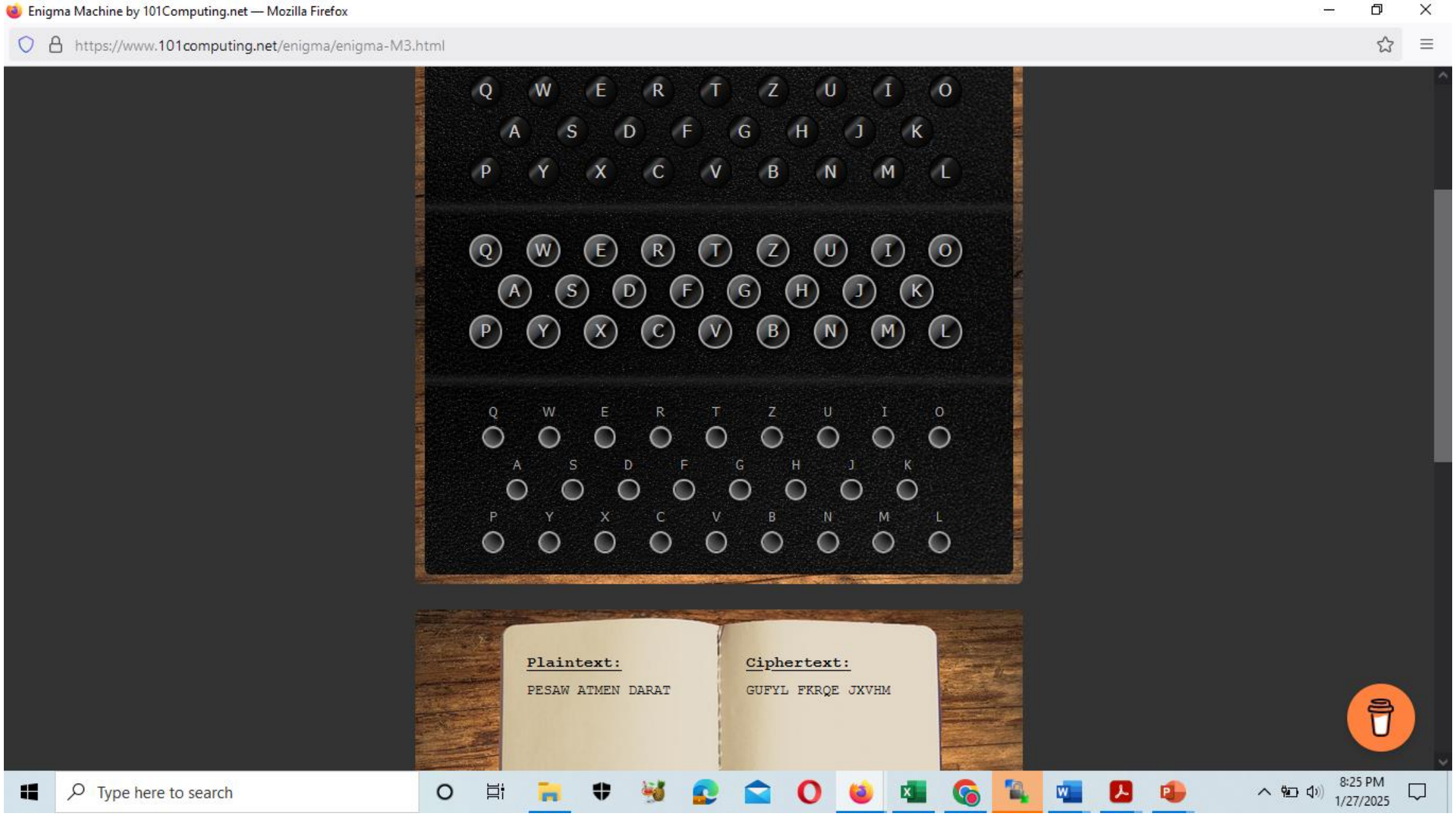


- *Bombe* berhasil memecahkan Enigma Cipher buatan Jerman.

- Keberhasilan memecahkan Enigma Cipher dianggap sebagai faktor yang memperpendek perang dunia kedua menjadi hanya dua tahun.

Coba simulator online Enimga di: https://www.101computing.net/enigma/enigma-instructions.html


Enigma Machine by 101Computing.net — Mozilla Firefox
https://www.101computing.net/enigma/enigma-M3.html
Enigma M3
D C B
B A Z
K J I
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L


Enigma Machine by 101Computing.net — Mozilla Firefox
https://www.101computing.net/enigma/enigma-M3.html
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L
Plaintext:
PESAW ATMEN DARAT
Ciphertext:
GUFYL FKRQE JXVHM
Type here to search
8:25 PM
1/27/2025

Encryption Steps:

Keyboard Input: P
Rotors Position: ABE
Plugboard Encryption: P
Wheel 3 Encryption: W
Wheel 2 Encryption: U
Wheel 1 Encryption: A
Reflector Encryption: Y
Wheel 1 Encryption: O
Wheel 2 Encryption: T
Wheel 3 Encryption: G
Plugboard Encryption: G
Output (Lampboard): G

Keyboard Input: E
Rotors Position: ABM
Plugboard Encryption: E
Wheel 3 Encryption: W
Wheel 2 Encryption: U
Wheel 1 Encryption: A
Reflector Encryption: Y
Wheel 1 Encryption: O
Wheel 2 Encryption: T
Wheel 3 Encryption: Q
Plugboard Encryption: Q
Output (Lampboard): Q

Keyboard Input: S
Rotors Position: ABG
Plugboard Encryption: S
Wheel 3 Encryption: K
Wheel 2 Encryption: G
Wheel 1 Encryption: D
Reflector Encryption: H
Wheel 1 Encryption: P
Wheel 2 Encryption: P
Wheel 3 Encryption: F
Plugboard Encryption: F
Output (Lampboard): F

TAMAT