



Cryptology for IoT

Modules M4, M6, M8
Session of 24th May, 2022.

M6.1 Briefing of the session

M6.2 Friedman Test

M6.3 Hill Climbing

M6.4 Final Exercise using Hill Climbing

Prof.: Guillermo Botella



Cryptology for IoT

Modules M4, M6, M8
Session of 24th May, 2022.

M6.1 Briefing of the session

M6.2 Friedman Test

M6.3 Hill Climbing

M6.4 Final Exercise using Hill Climbing

Prof.: Guillermo Botella



M6.1 Briefing of today

- Keeping going with Cryptography and Cryptoanalysis (Crypto lab v1 and v2)
 - Slides and supplementary videos
 - Deal with Unknown cipher
 - Friedman Test
 - Hill Climbing
- Create and break our own Homophonic Substitution code
- I warn you for tomorrow 25th May. We will go to the Socrative. Second quiz (continuation of First quiz)
 - Please study the slides!



Cryptology for IoT

Modules M4, M6, M8
Session of 24th May, 2022.

M6.1 Briefing of the session

M6.2 Friedman Test

M6.3 Hill Climbing

M6.4 Final Exercise using Hill Climbing

Prof.: Guillermo Botella



Friedman Test for classical and modern crypto

Cryptanalysis

Cost / Fitness Function

Where Do We Need and Use Them?

- . Entropy
- . Language Models
- . Index of Coincidence



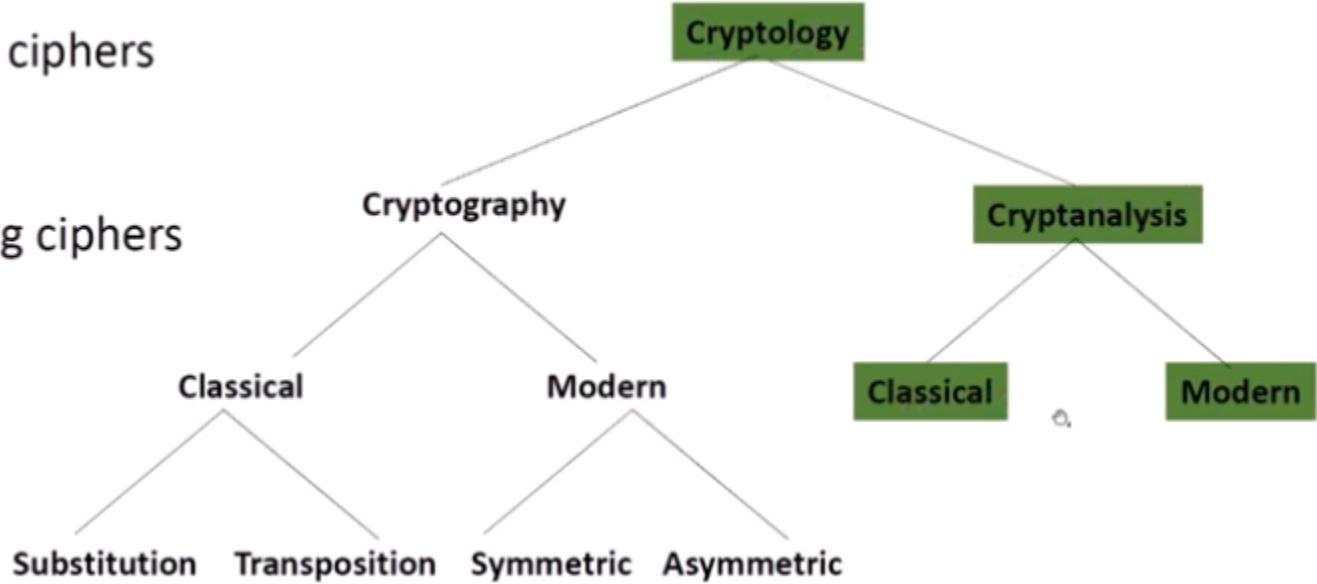
Friedman Test for classical and modern crypto

Cryptography

Art of making ciphers

Cryptanalysis

Art of breaking ciphers





Friedman Test for classical and modern crypto

- The Index of Coincidence (IoC) is a **statistical value**. It is the **probability** that **two letters randomly drawn** (from **different positions** of a given text) **are the same**
- The inventor of the Index of Coincidence was **William Friedman**
- It is defined as

$$IoC = \sum_{i=A}^Z \frac{n_i \cdot (n_i - 1)}{N \cdot (N - 1)}$$

- The sum of the number of each letter n_i multiplied with the sum of each letter minus one ($n_i - 1$) divided by the sum of all letters N multiplied by the sum of all letters minus one ($N - 1$)
- English texts have an IoC of about 0.066 and German texts have an IoC of about 0.078
- Random texts have an IoC of about $\frac{1}{\#(\text{letters in alphabet})} \rightarrow \frac{1}{26} = 0.038$



Friedman Test for classical and modern crypto

- The IoC was used by Friedman in the **Friedman Test** to determine the keylength of a polyalphabetic substitution cipher, e.g. the Vigenère cipher (not part of this video)
- We use the IoC as **cost or fitness function** in **modern heuristics** to **improve our (putative) key** during e.g. hill climbing
- The **closer the IoC** of the current decrypted plaintext is to the **IoC of the assumed language**, the “better” is our key
- The IoC can be used when other statistical values can't be used (e.g. ADFGVX, Enigma rotors)



Friedman Test for classical and modern crypto

- Example Calculation:

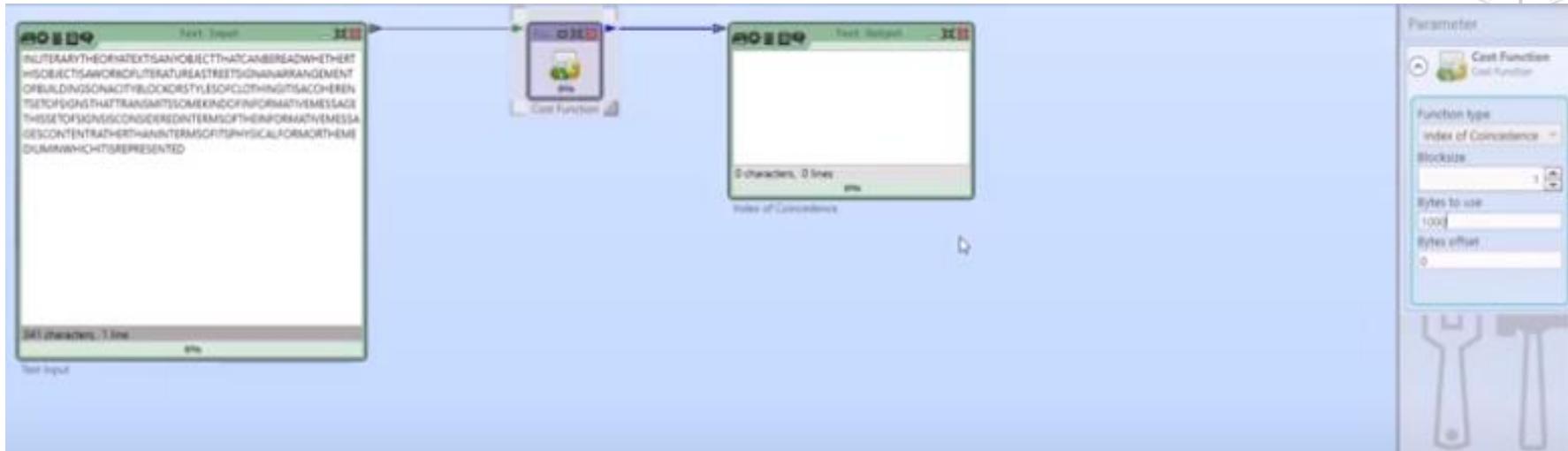
IN LITERARY THEORY A TEXT IS ANY OBJECT THAT CAN BE READ WHETHER THIS OBJECT IS A WORK OF LITERATURE A STREET SIGN AN ARRANGEMENT OF BUILDINGS ON A CITY BLOCK OR STYLES OF CLOTHING IT IS A COHERENT SET OF SIGNS THAT TRANSMITS SOME KIND OF INFORMATIVE MESSAGE THIS SET OF SIGNS IS CONSIDERED IN TERMS OF THE INFORMATIVE MESSAGES CONTENT rather than in terms of its physical form or the medium in which it is represented

$$N = 341$$

$$\begin{aligned} n_A &= 23, n_B = 5, n_C = 11, n_D = 7, n_E = 40, n_F = 11, n_G = 8, n_H = 16, n_I = 33, \\ n_J &= 2, n_K = 3, n_L = 7, n_M = 12, n_N = 25, n_O = 24, n_P = 2, n_R = 25, n_S = 32, \\ n_T &= 40, n_U = 3, n_V = 2, n_W = 3, n_X = 1, n_Y = 6 \end{aligned}$$

$$IoC = \frac{23 \cdot 22}{341 \cdot 340} + \frac{5 \cdot 4}{341 \cdot 340} + \frac{11 \cdot 10}{341 \cdot 340} + \dots + \frac{6 \cdot 5}{341 \cdot 340} = 0.071002243$$

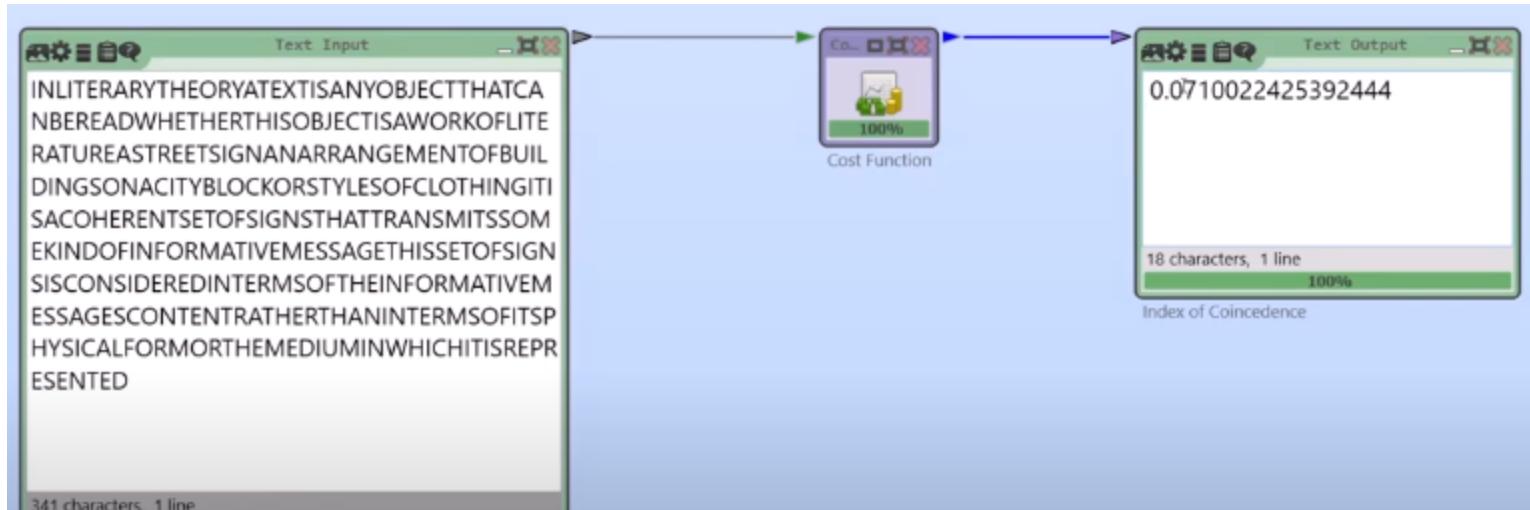
Friedman Test for classical and modern crypto



- Blocksize = 1
- 1000 bytes to use!



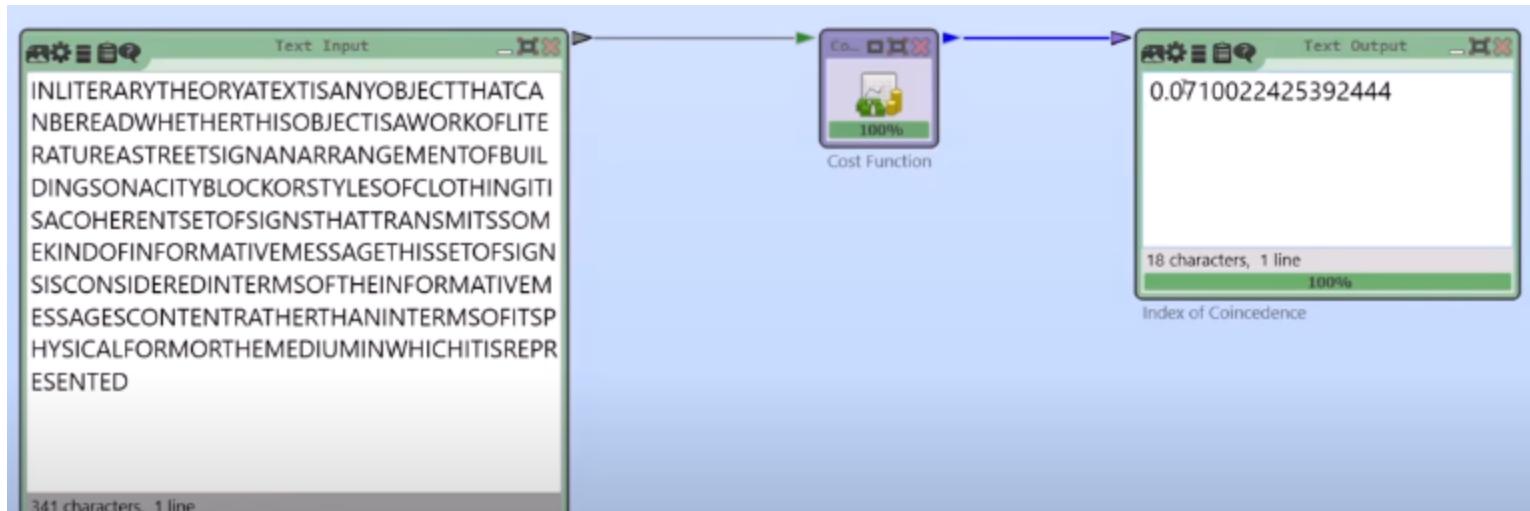
Friedman Test for classical and modern crypto



- Blocksize = 1
- 1000 bytes to use!



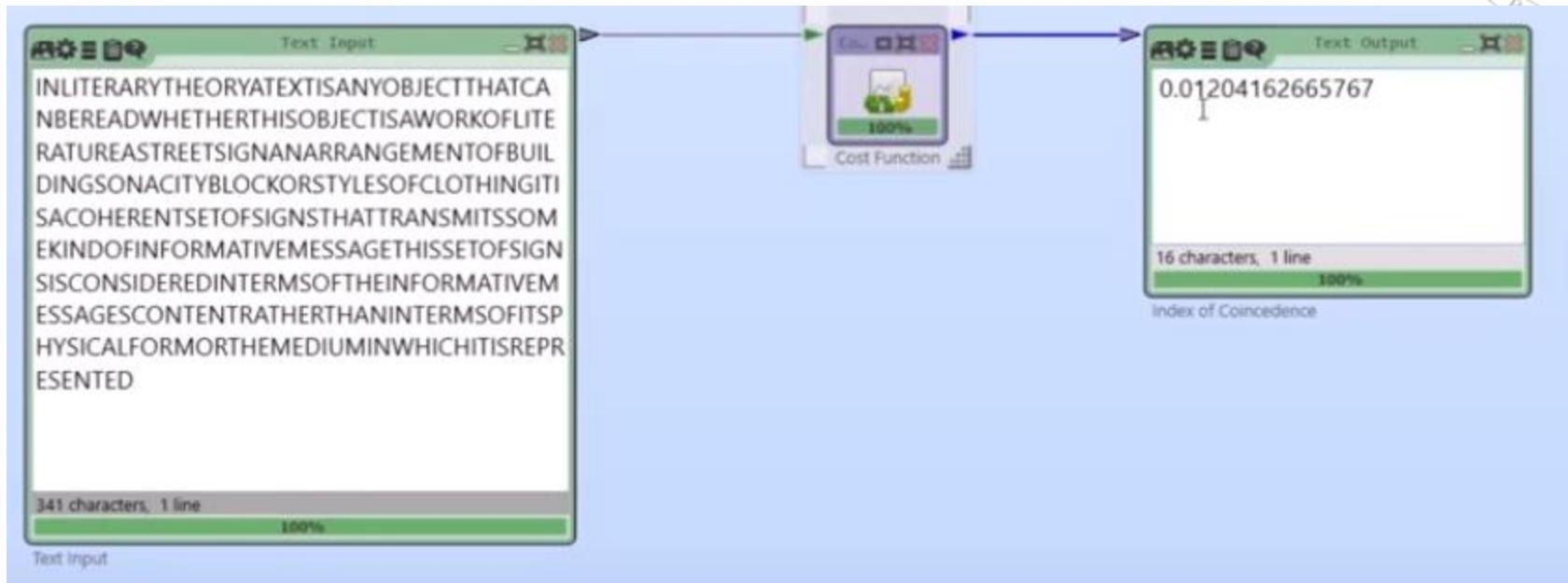
Friedman Test for classical and modern crypto



- Blocksize = 1
- 1000 bytes to use!



Friedman Test for classical and modern crypto



- Blocksize = 2
- Probability goes down!



Friedman Test for classical and modern crypto

Welcome to CrypTool 2. There are two ways for a quick start: Click on the wizard button in the section "Main Functions" to get a guided tour, or load one of the pre-defined workflows in the section "Templates" which dem...

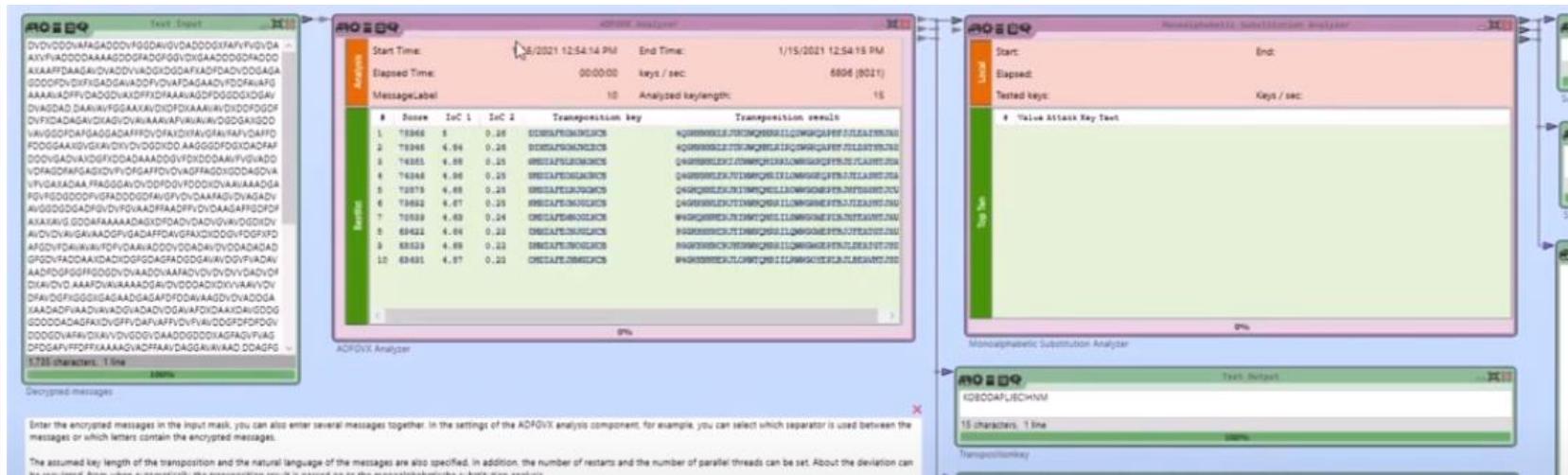
Main functions

- View a tutorial video about CrypTool 2 on YouTube
- Create a new workspace with the graphical editor
- Use the wizard to easily try some CrypTool 2 features
- Open the CrypTool Book

Templates (double click to open)

- adfgvxl
ADFGVX Cipher
- adfgvxl
ADFGVX Dictionary Attack
- adfgvxl
ADFGVX Heuristic Analysis

Friedman Test for classical and modern crypto



- Blocksize = 2
 - Probability goes down!



Friedman Test for classical and modern crypto

Analysis		Start Time:	1/15/2021 12:54:14 PM		End Time:	1/15/2021 12:54:17 PM	
		Elapsed Time:	00:00:03		keys / sec:	9426 (30101)	
		MessageLabel	10		Analyzed keylength:	15	
Bestlist	#	Score	IoC 1	IoC 2	Transposition key	Transposition result	
	1	93752	5.54	0.34	NHBIAFEDCJKOGLM	QGGKWNNEKJRKUIHQHQRЛИOVKGAQGEFTJHRDEGHTJKU	
	2	93033	5.75	0.33	BHNIAFEDCJKOGLM	KMGKWNNKFKJRKUINKHQRЛИPUKGAQGEXBJHRDEHGTJKU	
	3	83665	5.23	0.29	BHNIAFEDCLMJKG	KMGKTKNQFKJRIKUNKHQORLJUKGAEQGXBHQQLDBGTJKA	
	4	83304	5.4	0.28	BINHAFELMJKGDC	GQGHKNQWLEJIKURQHHORLKPIWGEQGATFJKLDBNSHJAU	
	5	83067	5.42	0.28	BHNIAFELMJKGDC	KMGHKNQWFKJIKURNKHORLKPUKGEQGAXBKQLDBNGTJAU	
	6	82310	5.31	0.28	BHNIAFEJKLMOGDC	KMGKNHQWFKJKIURNKHIRKPUKGQEGAXBJLEJBNGTJUA	
	7	81538	5.31	0.28	BINHAFEJKLMOGDC	GQGKNHQWLEJKIURQHHRIRKPIWQEGATFJLEJBNSHJUA	
	8	80481	5.3	0.27	BINHAFEJMLKGDC	GQGKHNQWLEJIKURQHHROLKPIWQEGATFJLKDBNSHJSC	
	9	76574	5.16	0.25	BINHAFEOMJKLGDC	GQGHKNQWLEJUKIRQHHRRRIKPIWQKQAATFJJLEBNSHJAU	
	10	75966	5	0.26	DINHAFEOMJKLGCB	4QGHKNNEJUKIWIQHHRRILQIWGKQAPBFJJLEATSHJAU	

- See difference between (IoC) unigrams and bigrams

Friedman Test for classical and modern crypto



The screenshot shows a software interface for cryptanalysis, specifically for monoalphabetic substitution ciphers. It consists of several windows and panels:

- Left Panel:** Shows a large text area containing a long string of encrypted text: "ACDDODXAFVIVGDA...". Below it is a message label input field.
- Top Left Window:** Titled "ROB 09", it displays analysis results for "ADFOXVX" messages. It includes fields for "Start Time", "End Time", "Elapsed Time", and "Analyzed keylength". A table lists 10 rows of data with columns: #, Row, Inv 1, Inv 2, Transposition key, and Transposition result.
- Top Middle Window:** Titled "ROB 09", it shows the "Monoalphabetic Substitution Analyzer" results. It includes fields for "Start", "End", "Elapsed", and "Tasted key". A table lists 1 row of data with columns: #, Value, Attack, Key.
- Top Right Window:** Titled "ROB 09", it shows the "Text Output" of the monoalphabetic substitution analysis, displaying the decrypted text: "SUTTONACDFGHJKLMNPQRUVWXYZÖÄ".
- Middle Right Window:** Titled "ROB 09", it shows the "Text Output" of the transposition analysis, displaying the decrypted text: "ABCDEFOHJKLMNPQRSTUVWXYZÖÄ".
- Bottom Right Window:** Titled "ROB 09", it shows the "Text Output" of the fraction thread analysis, displaying the decrypted text: "GENERALFELDMARSCHALLMACKENSENVRDAMOONAUDELTSCHMITWEITERSNTRUPPENVEREINENUNDVORSTOSSENNOVISALUFUNRENSTEKMPANIERERBITTEWEITERLEITERSTOFLEIERUNGUNFERKOMMENDEWICHMONTAGRÜCKDONAUWURDEHOFFENDERSTORTZQUERUNDNICHMOGLICHWEREERLUSTTEBBORGRECHTVORBLGRASZTADTWEISTERINTERBELADUNGDEKAESERHABENDEBUNDUNGSLOSENKAMPFENSYDIDENDEUTSCHENOLDATENBEPOHLINZWITRANDEERLIEHTATTLEMEDENVABUREEINGEDEINENACHRICHTENANDERWEITERLEITENUMMUTERNEINICHTANOSTFRONTZBRINGENZDESERTURENSISDORTVERSCHIESSENUNITIONAVSTOLZANIDOVERRAUCHZERBITTENSFORTGENACHSCHULEERUNGHAUPTMANVIERZUEINTEDOMPAKIZHEUTUMDREUHRDSEFALLZELTNANTMUELLERWURDEUMLPTMERNANTUNDERSTETTMERORNSEINHEITENVONRITTERKOMPANIENAUSGEFALLENZNACHRICHTENABSOFTDIRKTUBERBERESTKOMMELITENWEITERNACHWEIORSCHLECHTZVORSTOSSEDAHERNICHTMOGLUCHZWARTENAUFBESSERUNG".

Below the main windows, there are several descriptive text blocks:

- "Input mask, you can also enter several messages together. In the settings of the ADFOXVX analysis component, for example, you can select which separator is used between the encrypted messages."
- "Position and the natural language of the messages are also specified. In addition, the number of restarts and the number of parallel threads can be set. About the deviation of the transposition result is passed on to the monoalphabetic substitution analysis."
- "The ADFOXVX analysis component and simulated annealing reverses the transposition. The result is passed to the monoalphabetic substitution analysis, which completely decrypts the message."



Cryptology for IoT

Modules M4, M6, M8
Session of 24th May, 2022.

M6.1 Briefing of the session

M6.2 Friedman Test

M6.3 Hill Climbing

M6.4 Final Exercise using Hill Climbing

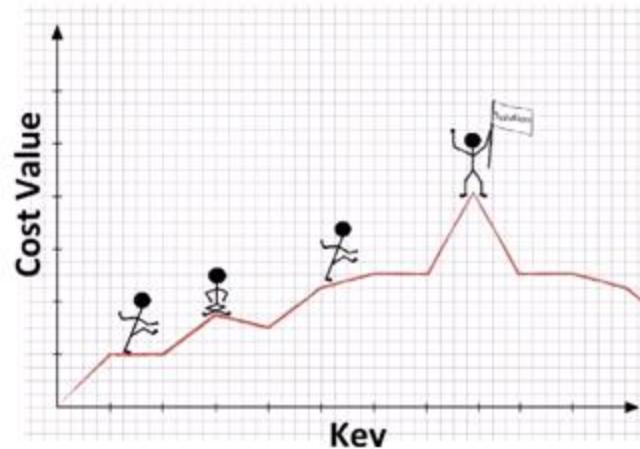
Prof.: Guillermo Botella



Hill climbing for classical and modern crypto

Cryptanalysis of Classic Ciphers Using Heuristics

- . What is a Heuristic?
- . Hill Climbing – Basics
- . Genetic Algorithm
- . Simulated Annealing – Improved Hill Climbing





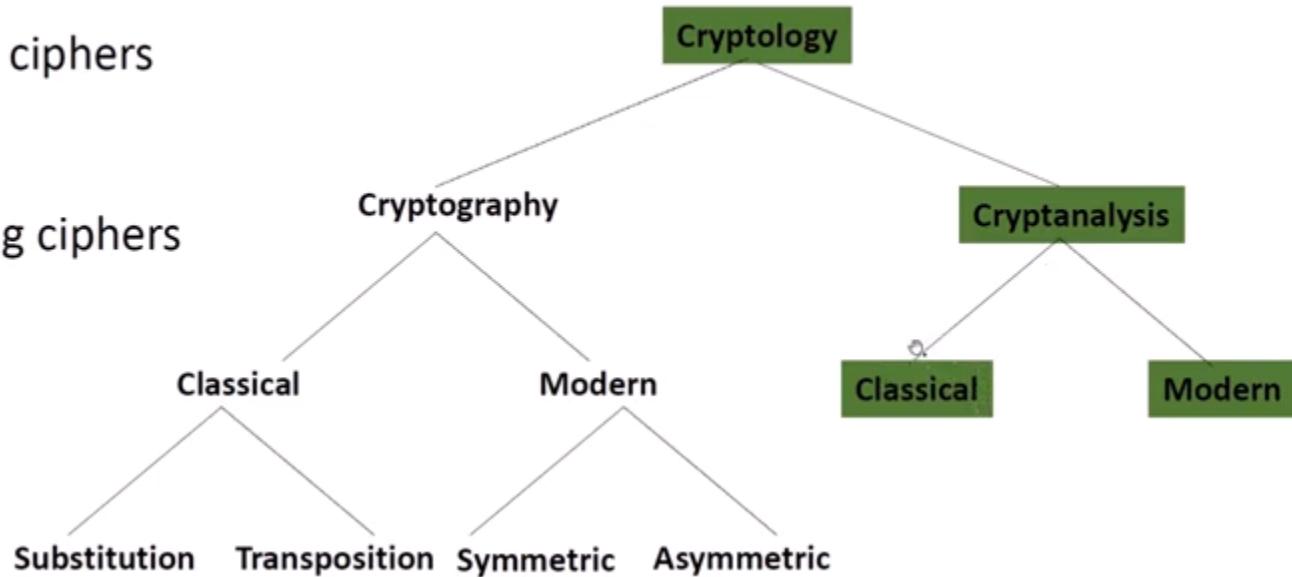
Hill climbing for classical and modern crypto

Cryptography

Art of making ciphers

Cryptanalysis

Art of breaking ciphers





Hill climbing for classical and modern crypto

- A **heuristic technique** (/hjʊə'rɪstɪk/; Ancient Greek: εύρίσκω, "find" or "discover"), or a **heuristic**, is any approach to **problem solving** or self-discovery that employs a **practical method** that is **not guaranteed to be optimal**, perfect or rational, but which is nevertheless sufficient for **reaching an immediate, short-term goal**. Where **finding an optimal solution** is **impossible** or **impractical**, heuristic methods can be used to speed up the process of finding a **satisfactory solution**. (Englisch Wikipedia)
- Our problem: Finding the key of a classical cipher with brute-force is (mostly) **impractical**
- Can we **speed up the search** for the key using a heuristic, → Yes, of course ☺ making the search **practical**?
- Hill climbing is a well suited heuristic to do so. Why? → On next slides ☺



Hill climbing for classical and modern crypto

- First, we need to know the **vulnerabilities** of **classical ciphers** we exploit with **hill climbing**
- I focus now on substitution ciphers, but transposition ciphers also work
 1. **Letter frequencies** are **still visible** in ciphertext
 2. **Low diffusion** after changing key or plaintext
 3. We can measure "**how good (even a wrong) key is**"
 4. **Inventors** of (historical/classical) ciphers **didn't know computers** ☺



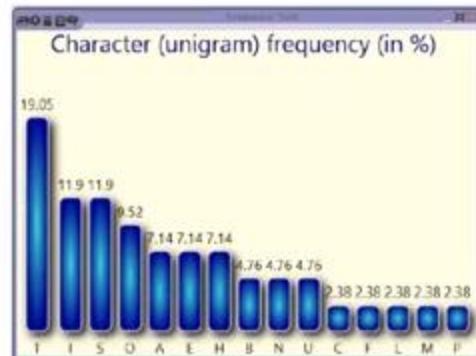
Hill climbing for classical and modern crypto

1. Letter frequencies are **still visible** in ciphertext

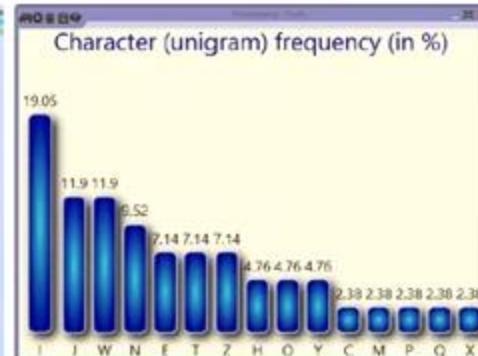
Example: Simple monoalphabetic substitution cipher

THIS IS A TEST OF THE MONOALPHABETIC SUBSTITUTION.

ITWJ WJ Z IEJI NC ITE PNONZQMTZYEIWX JHYJIWIHIWNO



Plaintext Frequencies



Ciphertext Frequencies



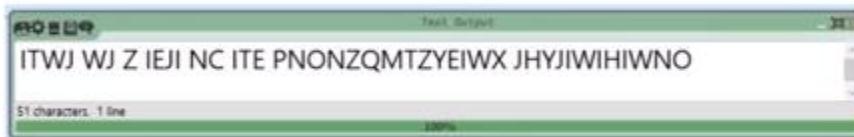
Hill climbing for classical and modern crypto

2. Low diffusion after changing key or plaintext

THIS IS A TEST OF THE MONOALPHABETIC SUBSTITUTION

Key:

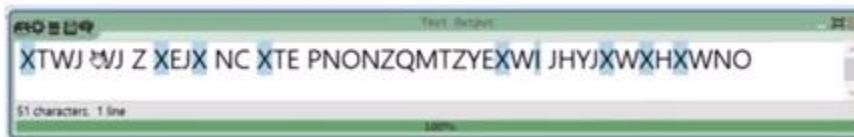
ZYXSECRTWVUQPONMLKJIHGFDBA



Ciphertext

Changed Key:

ZYISECRTWVUQPONMLKJXHGFDBA



Ciphertext

In this example, only nine letters changed in the ciphertext. This is about 17% of the letters.
Modern ciphers change on average about 50% of the bits when their inputs are changed.



Hill climbing for classical and modern crypto

3. We can measure “how good (even a wrong) key is”

- “The better a key the higher is its fitness”
- Example (bigram log2 cost function of CrypTool 2; performed on decrypted text)

EYISZWRTCVUQOPNMLKJAHGFBX	193.24	<- start key
EYISZCRTWVUQOPNMLKJAHGFBX	214.27	
EYISZCRTWVUQPONMLKJAHGFBX	218.16	
ZYISECRTWVUQPONMLKJAHGFBX	231.14	
ZYISECRTWVUQPONMLKJXHGFBX	271.28	<- final key



Hill climbing for classical and modern crypto

4. Inventors of (historical/classical) ciphers didn't know computers

- 120,000 keys/sec with Vigenère analyzer (on a single CPU core)
- Enables optimization techniques (heuristics) to use vulnerabilities mentioned before
 - Hill climbing
 - Simulated annealing
 - Genetic algorithms
 - ...
- Parallelization and distribution additionally speeds up the analysis ☺



Hill climbing for classical and modern crypto

- Why not just test every key (search for the correct one via a brute-force attack) ? (by rating all keys and the best should have the highest rating)
- Simplest cipher: Simple monoalphabetic substitution
We can achieve ~120,000 keys/sec
- Computation time to search through all keys

$$26! = 2^{88.4} \text{ key}$$

$$\text{Search time} = \frac{2^{88.4} \text{ keys}}{120,000 \text{ keys/sec}} \approx 10^{21.5} \text{ sec} \approx 10^{14} \text{ years}$$

- Maybe, brute-force is not a good idea... 😞



Hill climbing for classical and modern crypto

Step 1: Create an **initial random key**

Step 2: **Decrypt ciphertext using the initial key and compute fitness (e.g. trigram frequencies sum)**

Step 4: **Modify key (e.g. randomly swap letters)**

Step 5: **Decrypt ciphertext using the modified key and compute fitness**

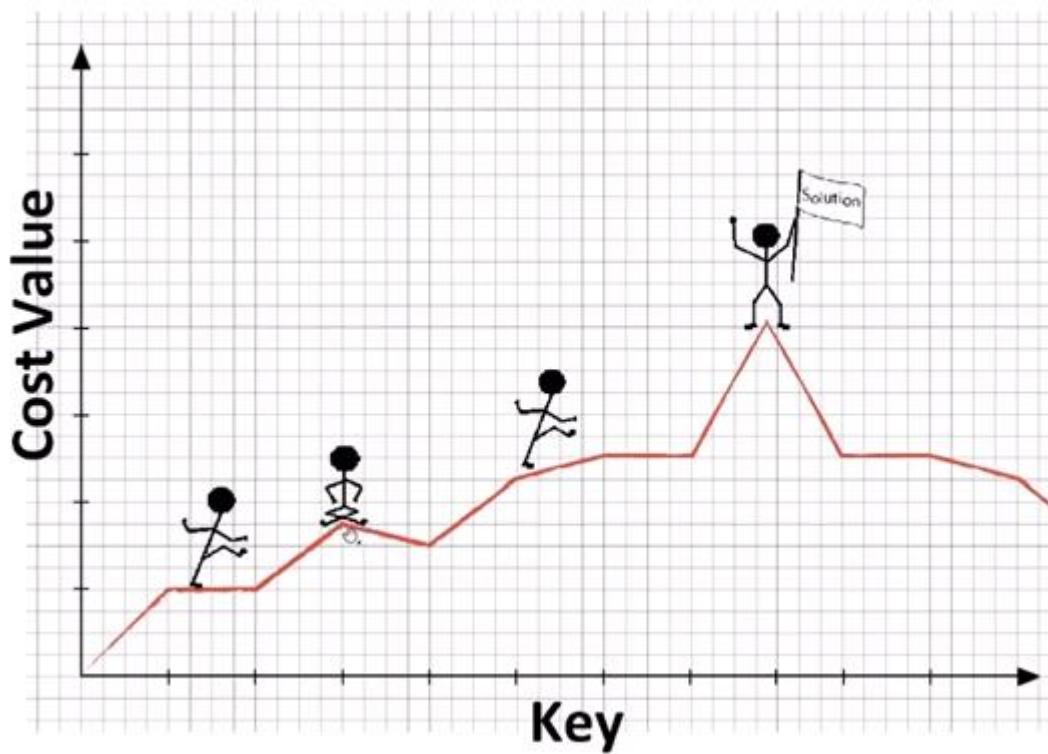
Step 6: If fitness **is worse than before revert modifications**

Step 7: Increment a **counter**;
If counter is above a defined value, we stop algorithm

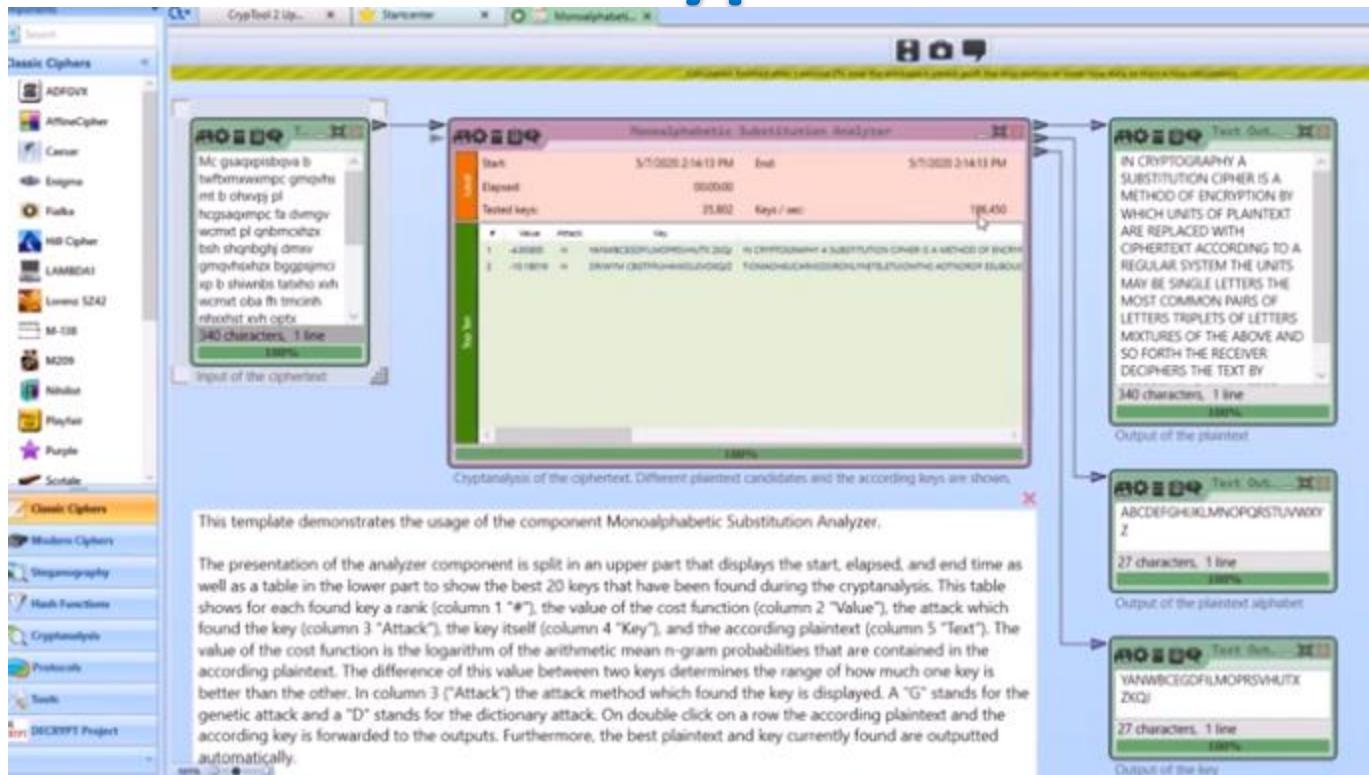
Step 8: Jump to Step 4



Hill climbing for classical and modern crypto



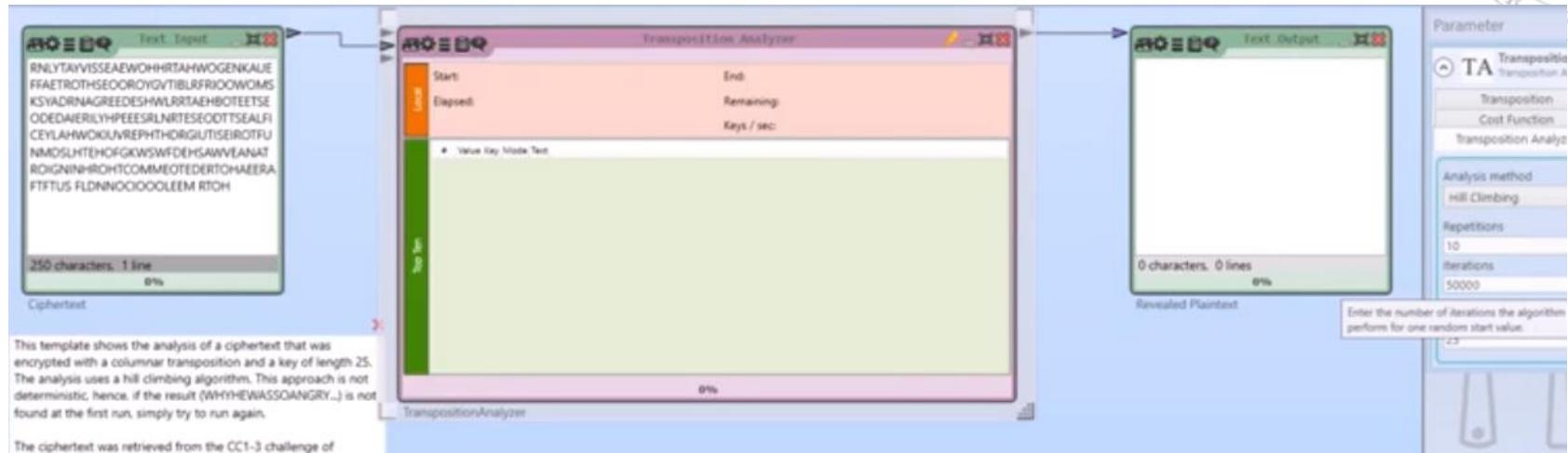
Hill climbing for classical and modern crypto



- Different uses of hill climbing
 - Monoalphabetic substitution analyzer



Hill climbing for classical and modern crypto



- Different uses of hill climbing
 - Transposition analyzer

Hill climbing for classical and modern crypto



This template shows the analysis of a ciphertext that was encrypted with a columnar transposition and a key of length 25. The analysis uses a hill climbing algorithm. This approach is not deterministic, hence, if the result (WHYHEWASSOANGRY...) is not found at the first run, simply try to run again.

The ciphertext was retrieved from the CC1-3 challenge of MysteryTwister L, where you can find many more ciphertexts that were encrypted with a columnar transposition. You can then use

Text Input

```
RNLTYAYVISSEAEOWHRTAHWOGENKAUE  
FFAETROTSEOROYGVIBLRFRIOWOMS  
KSYADRNAGREEDESHWLRRTAEBOTEETSE  
ODEDAIERILYHPEEESRNLRTESOOTTSEALFI  
CEYLAHWOKIUVREPHTHDRIUTIESIROTFU  
NIMDSLHTEHOFGKWSWDFEHSANVEANAT  
ROIGNINHRHOHTCOMMEOTEDTOHAEEERA  
FTFTUSFLDNNOOCIOOLEEM RTOH
```

250 characters, 1 line 100% Ciphertext

Transposition Analyzer

#	Value	Key	Mode	Text
1	721.01758	[15, 17, 7, 18, 12, 23, 13, 11, 10, 24, 6, 2, 25, 19, 4, 16, 3, 9, 8, 14, 1, 20, 22, 21, 8]	R-C-R	WHYHEWASSOANGRYOUVE
2	689.16332	[15, 17, 7, 18, 12, 11, 10, 24, 23, 13, 6, 2, 25, 19, 4, 16, 3, 9, 8, 14, 1, 20, 22, 21, 5]	R-C-R	WHYHESSOWAANGRYOLIVE
3	684.78152	[15, 17, 7, 18, 12, 23, 13, 2, 25, 19, 11, 10, 24, 6, 4, 16, 3, 9, 8, 14, 1, 20, 22, 21, 8]	R-C-R	WHYHEWANGRSOAYOUVE
4	678.56052	[15, 6, 2, 25, 19, 4, 16, 17, 7, 18, 12, 23, 13, 11, 10, 24, 3, 9, 8, 14, 1, 20, 22, 21, 8]	R-C-R	WANGRYOHYHEWASSOLIVE
5	675.44185	[15, 6, 2, 25, 19, 4, 16, 3, 9, 8, 14, 1, 17, 7, 18, 12, 23, 13, 11, 10, 24, 20, 22, 21, 8]	R-C-R	WANGRYOUVERHYHEWASSO
6	675.40764	[15, 6, 2, 25, 19, 4, 17, 7, 18, 12, 23, 13, 11, 10, 24, 16, 3, 9, 8, 14, 1, 20, 22, 21, 5]	R-C-R	WANGRYHYHEWASSOLIVE
7	673.59066	[15, 3, 9, 8, 14, 1, 20, 22, 21, 5, 11, 10, 24, 6, 2, 25, 19, 4, 15, 17, 7, 18, 12, 23, 13]	R-C-R	OLIVERTHATSANGRWYWHY
8	671.21302	[15, 6, 2, 25, 19, 4, 16, 3, 9, 8, 14, 17, 7, 18, 12, 23, 13, 11, 10, 24, 1, 20, 22, 21, 8]	R-C-R	WANGRYOUVERHYHEWASSO
9	668.94254	[15, 6, 2, 25, 19, 11, 10, 24, 4, 17, 7, 18, 12, 23, 13, 16, 3, 9, 8, 14, 1, 20, 22, 21, 8]	R-C-R	WANGRSSOYHYHEWASSOLIVE
10	667.14966	[15, 6, 2, 25, 19, 4, 16, 3, 9, 17, 7, 18, 12, 23, 13, 11, 10, 24, 8, 14, 1, 20, 22, 21, 8]	R-C-R	WANGRYOUHYHEWASSOBE

Top Ten 1 100% TranspositionAnalyzer

Text Output

```
WHYHEWASSOANGRYOLIVERTHATHEFOR  
GOTEVENTOASKAFTERYOUADMRSOWE  
RBERRYLOOKINGAFTERTHEBEADLEASHES  
TRODEDOWNTHESTREETYESIRREPLIEDO  
LIVERWHOADCAREFULLYKEPTHIMSELF  
UTOFSIGHTDURINGTHEINTERVIEWANDW  
HOWASSHAKINGFROMHEADTOFOOTATT  
HEMERERECOLLECTIONOFTHESOUNDOF  
MR.
```

250 characters, 1 line 100% Revealed Plaintext

- Different uses of hill climbing
 - Transposition analyzer

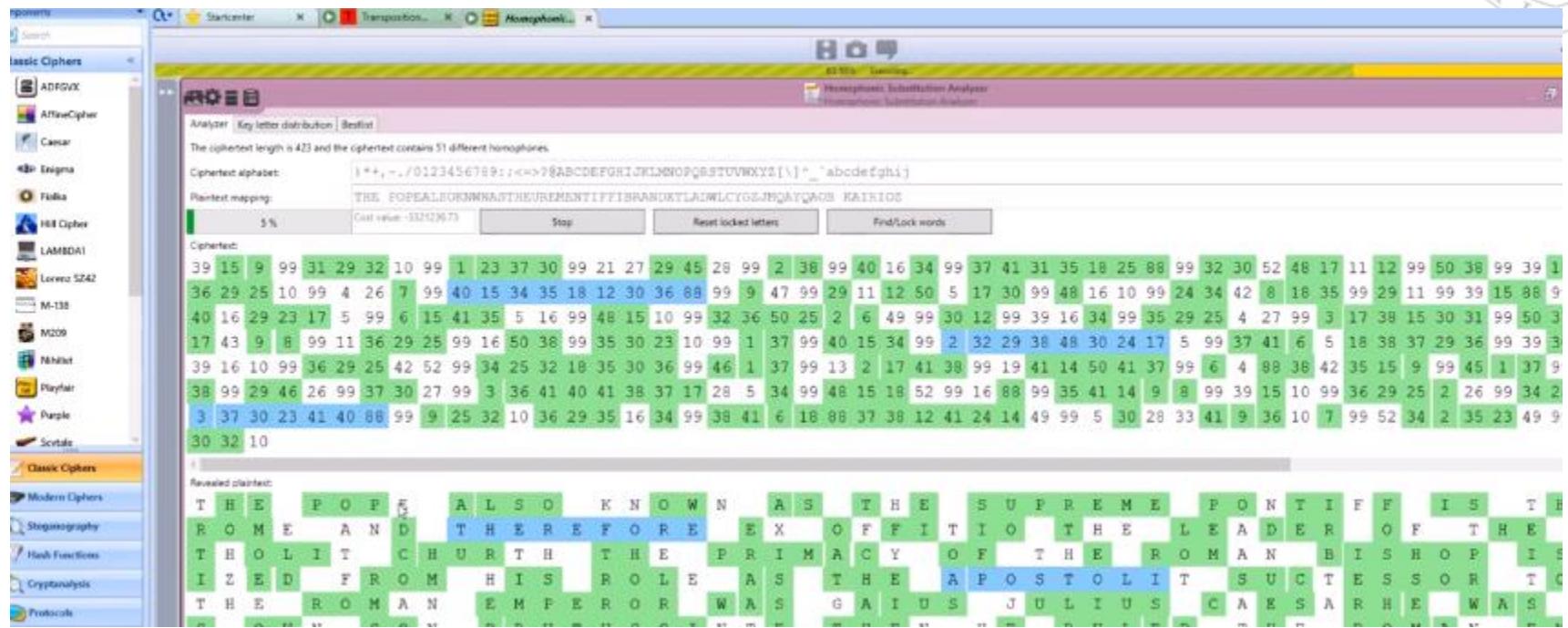
Hill climbing for classical and modern crypto



The screenshot shows the ROBEX Homophonic Substitution Analyzer interface. The main window displays a ciphertext of 423 characters. Below the ciphertext, a detailed cryptanalysis section includes a table for 'Key letter distribution' and 'Cipher mapping'. To the right, a 'Parameter' panel shows a dictionary with 41,238 entries. Three smaller windows on the right show the 'Revealed plaintext' (empty), 'Revealed key' (empty), and a 'ROBEX' logo.

- Different uses of hill climbing
 - Homophonic Substitution analyzer

Hill climbing for classical and modern crypto



- Different uses of hill climbing
 - Homophonic Substitution analyzer

Hill climbing for classical and modern crypto

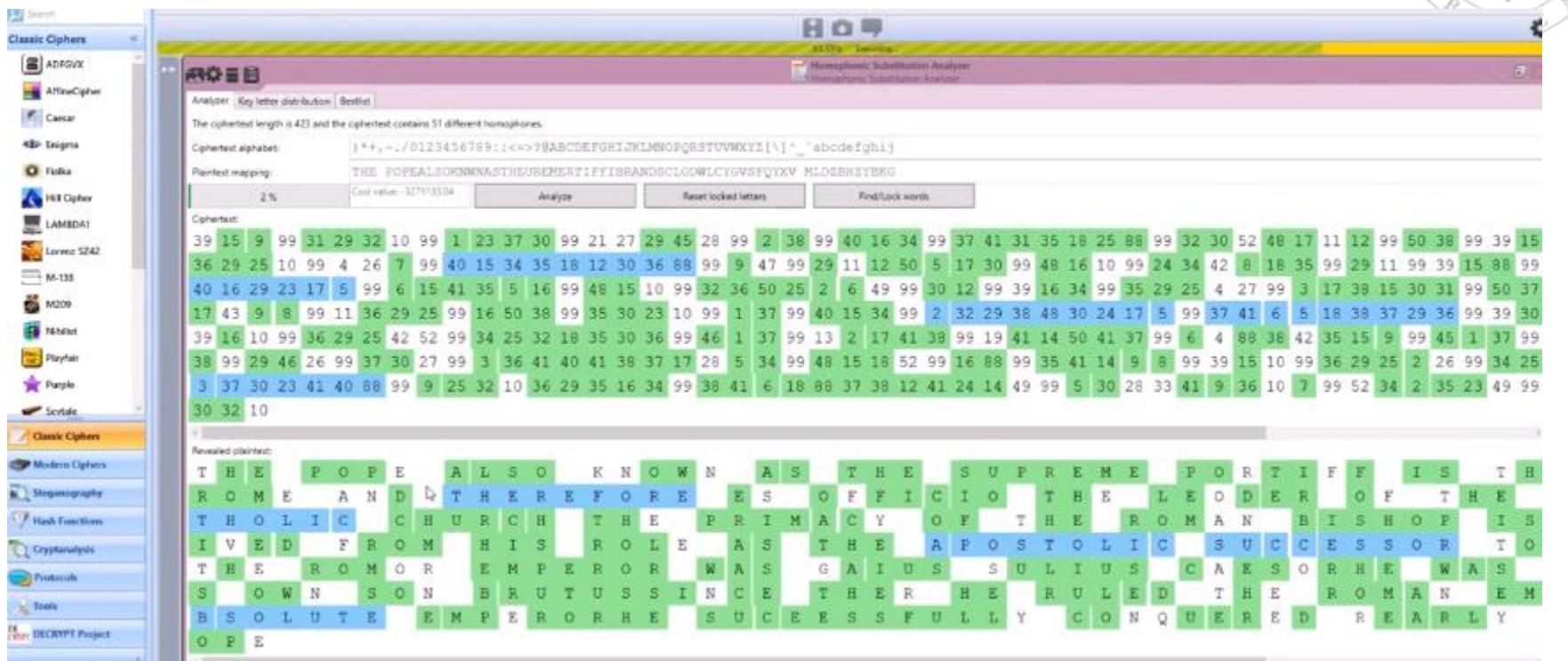


The screenshot shows the Homophonic Substitution Analyzer interface. At the top, there are icons for file operations (New, Open, Save, Print, Exit) and a progress bar indicating '85.7% Executing'. The title bar reads 'Homophonic Substitution Analyzer' and 'Homophonic Substitution Analyzer'. Below the title bar, there are tabs for 'Analyzer', 'Key letter distribution', and 'Bestlist'. The main area displays the following information:

- Ciphertext length: 423
- Different homophones: 51
- Ciphertext alphabet:) *-,./0123456789;:<>?@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
- Plaintext mapping: THE POPE ALSO ANWNASTHEUREMENTIFFBRANDLCLADWL/CYGTJFQUTAOBQNXFAZYRG
- Status bar: 15 % Cost value: -3174629.62
- Buttons: Analyze, Reset locked letters, Find/Lock words
- Ciphertext grid: A large grid of numbers representing the analyzed ciphertext.

- Different uses of hill climbing
 - Homophonic Substitution analyzer

Hill climbing for classical and modern crypto



- Different uses of hill climbing
 - Homophonic Substitution analyzer



Cryptology for IoT

Modules M4, M6, M8
Session of 24th May, 2022.

M6.1 Briefing of the session

M6.2 Friedman Test

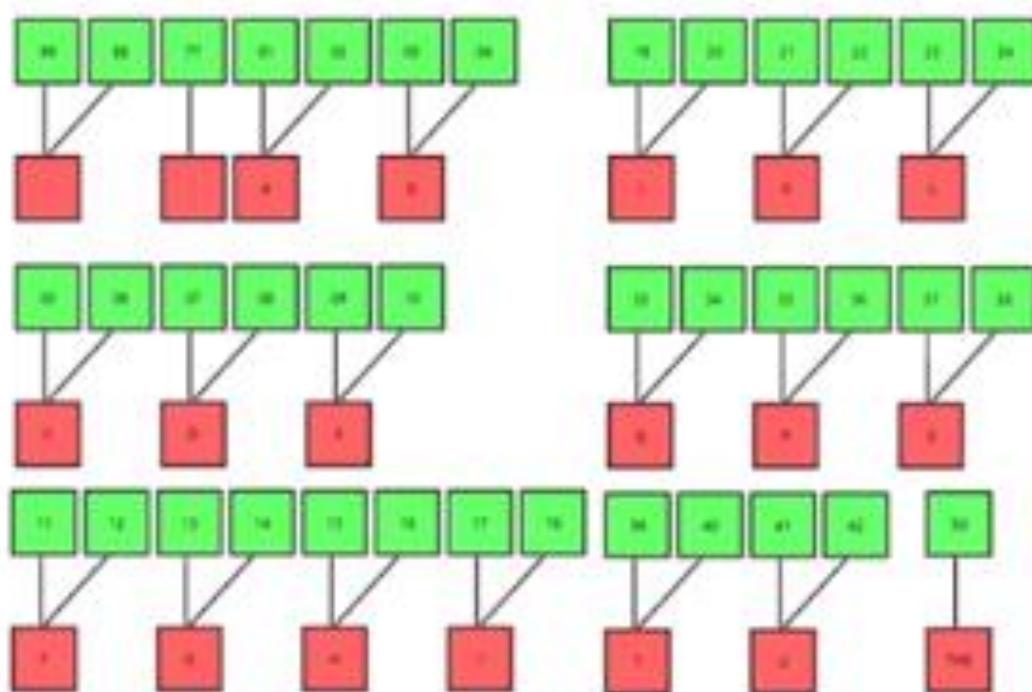
M6.3 Hill Climbing

M6.4 Final Exercise using Hill Climbing

Prof.: Guillermo Botella



Homophonic Substitution



Visualization of a homophonic substitution cipher

Green boxes = ciphertext symbols

Red boxes = plaintext symbols



Homophonic Substitution

- A **simple substitution** replaces **single letters** by single letters, digit groups, or (in general) symbols
 - Examples: **Caesar cipher**, **Simple MASC**, **Pigpen cipher**
 - Simple substitution ciphers can be broken easily by hand (with frequency analysis)

Example encryption with a simple MASC*:

Key: H=01,E=02,L=03,O=04,W=05,R=06,D=07

HELLO WORLD → 01 02 03 03 04 05 04 06 03 07

- To make ciphers more secure, cryptographers invented **homophonic substitution** ciphers
- A homophonic substitution has more choices (**homophones**) for encrypting letters (or pieces of text)

Example encryption with a homophonic substitution cipher*:

Key: H=01,E=02,L=03|04,O=05|06,W=07,R=08,D=09

HELLO WORLD → 01 02 03 04 05 07 06 08 03 09

* To make it easy to “produce” many ciphertext symbols, we use digits



Homophonic Substitution

- **Example plaintext:**

IN CRYPTOGRAPHY A SUBSTITUTION CIPHER IS A METHOD OF ENCRYPTING IN WHICH UNITS OF PLAINTEXT ARE REPLACED WITH THE CIPHERTEXT IN A DEFINED MANNER WITH THE HELP OF A KEY THE UNITS MAY BE SINGLE LETTERS THE MOST COMMON PAIRS OF LETTERS TRIPLETS OF LETTERS MIXTURES OF THE ABOVE AND SO FORTH. THE RECEIVER DECIPHERS THE TEXT BY PERFORMING THE INVERSE SUBSTITUTION PROCESS TO EXTRACT THE ORIGINAL MESSAGE.

- Encrypted ciphertext using **simple MASC** (A=01, B=02, C=03, ..., Z=26):

09 14 00 03 18 25 16 20 15 07 18 01 16 08 25 00 01 00 19 21 02 19 20 09 20 21 20 09 15 14 00 03 09 16 08 05 18 00 09 19 00 01 00 13 05 20 08 15 04 00
15 06 00 05 14 03 18 25 16 20 09 14 07 00 09 14 00 23 08 09 03 08 00 21 14 09 20 19 00 15 06 00 16 12 01 09 14 20 05 24 20 00 01 18 05 00 18 05 16 12
01 03 05 04 00 23 09 20 08 00 20 08 05 00 03 09 16 08 05 18 20 05 24 20 00 09 14 00 01 00 04 05 06 09 14 05 04 00 13 01 14 14 05 18 00 23 09 20 08 00
20 08 05 00 08 05 12 16 00 15 06 00 01 00 11 05 25 00 20 08 05 00 21 14 09 20 19 00 13 01 25 00 02 05 00 19 09 14 07 12 05 00 12 05 20 20 05 18 19 00
20 08 05 00 13 15 19 20 00 03 15 13 13 15 14 00 16 01 09 18 19 00 15 06 00 12 05 20 20 05 18 19 00 20 18 09 16 12 05 20 19 00 15 06 00 12 05 20 20 05
18 19 00 13 09 24 20 21 18 05 19 00 15 06 00 20 08 05 00 01 02 15 22 05 00 01 14 04 00 19 15 00 06 15 18 20 08 00 20 08 05 00 18 05 03 05 09 22 05 18
00 04 05 03 09 16 08 05 18 19 00 20 08 05 00 20 05 24 20 00 02 25 00 16 05 18 06 15 18 13 09 14 07 00 20 08 05 00 09 14 22 05 18 19 05 00 19 21 02 19
20 09 20 21 20 09 15 14 00 16 18 15 03 05 19 19 00 20 15 00 05 24 20 18 01 03 20 00 20 08 05 00 15 18 09 07 09 14 01 12 00 13 05 19 19 01 07 05 00

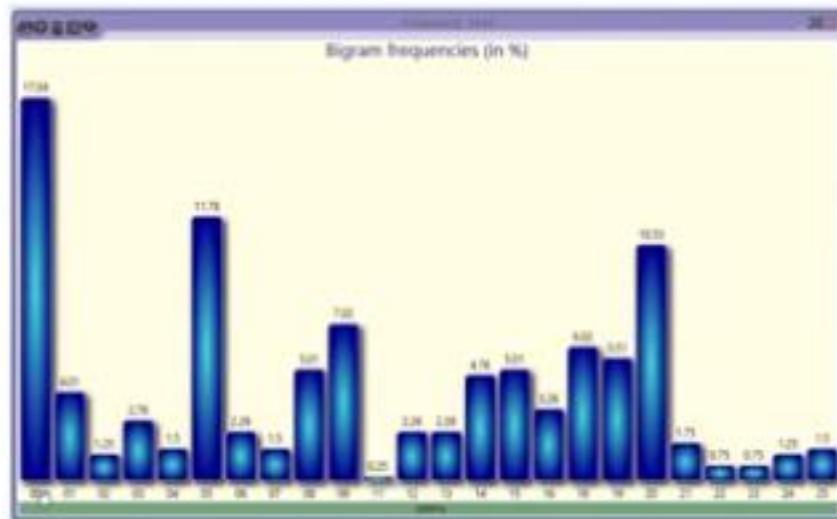
- Encrypted ciphertext using **homophonic substitution cipher** (homophonicity = 3):

09 14 00 03 18 25 16 20 15 07 44 01 42 08 51 88 27 99 19 21 02 45 46 35 72 47 20 61 41 40 00 29 09 68 34 05 70 88 35 71 99 53 00 13 31 46 60 67 04 88
15 06 99 57 66 55 18 77 16 72 61 14 33 00 09 40 88 23 08 35 03 34 99 73 66 61 20 19 00 41 32 88 42 12 01 09 14 46 05 24 72 99 27 44 31 00 70 57 68 38
53 29 05 30 88 49 35 20 60 99 46 08 31 00 55 61 16 34 57 18 72 05 50 20 88 09 40 99 01 00 56 31 58 35 66 57 04 88 39 27 14 40 05 44 99 75 61 46 60 00
72 08 31 88 34 57 64 42 99 67 06 00 53 88 11 05 25 99 20 60 31 00 21 66 09 46 45 88 65 01 51 99 28 57 00 71 35 14 59 12 05 88 38 31 72 20 57 70 19 99
46 08 05 00 13 15 45 72 88 03 41 39 65 67 40 99 68 27 61 18 71 00 15 32 88 64 31 20 46 57 44 19 99 72 70 09 16 12 05 20 45 00 41 58 88 38 31 46 72 57
18 71 99 13 35 76 20 47 44 05 19 00 67 06 88 46 34 31 99 53 54 15 22 57 00 01 66 30 88 45 41 99 32 67 70 72 60 00 20 08 05 88 18 31 29 57 61 48 05 44
99 56 31 55 09 42 34 57 70 71 00 46 60 05 88 72 31 24 20 99 02 77 00 68 57 18 58 15 44 39 35 14 07 88 46 08 05 99 61 40 74 31 70 19 57 00 45 44 99 75
72 09 20 21 46 35 41 66 88 16 18 67 03 05 19 45 99 72 15 00 31 50 20 44 27 29 46 88 72 34 57 99 41 70 61 33 09 14 53 64 00 65 05 71 19 01 59 31 88

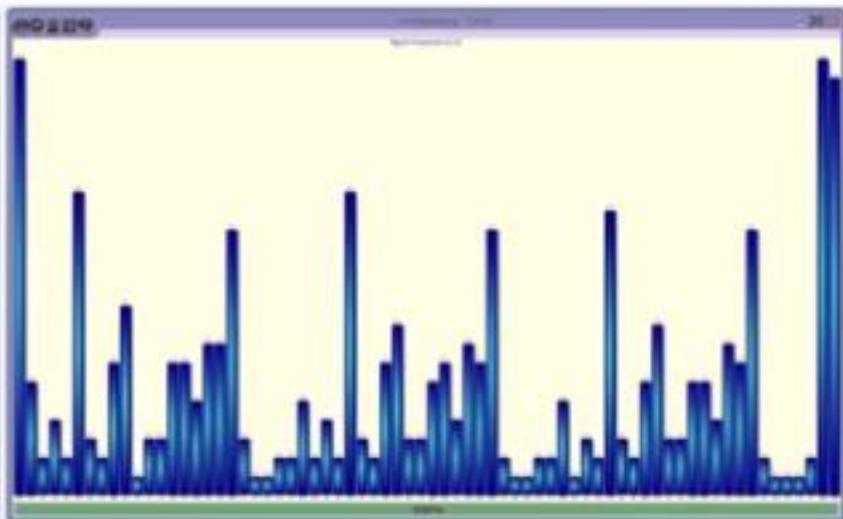


Homophonic Substitution

Frequencies of two digit combinations



Simple substitution cipher
27 different ciphertext symbols in total

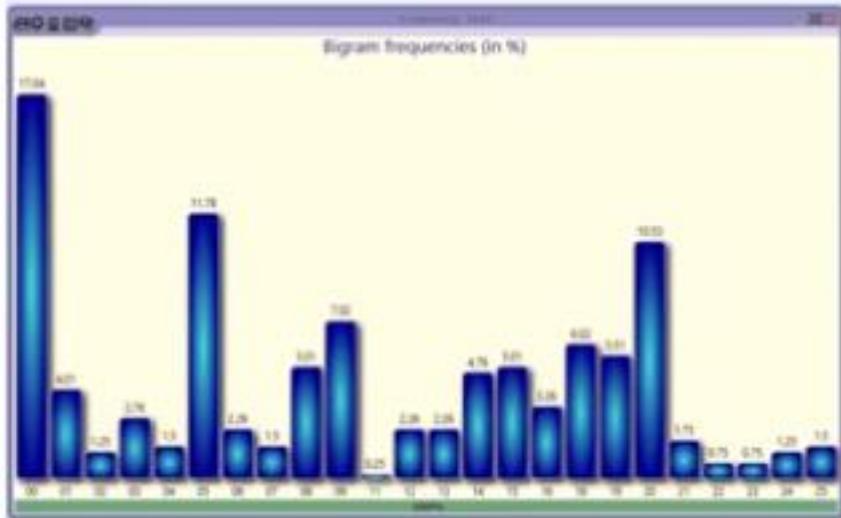


Homophonic substitution cipher – homophonicity = 3
70 different ciphertext symbols in total

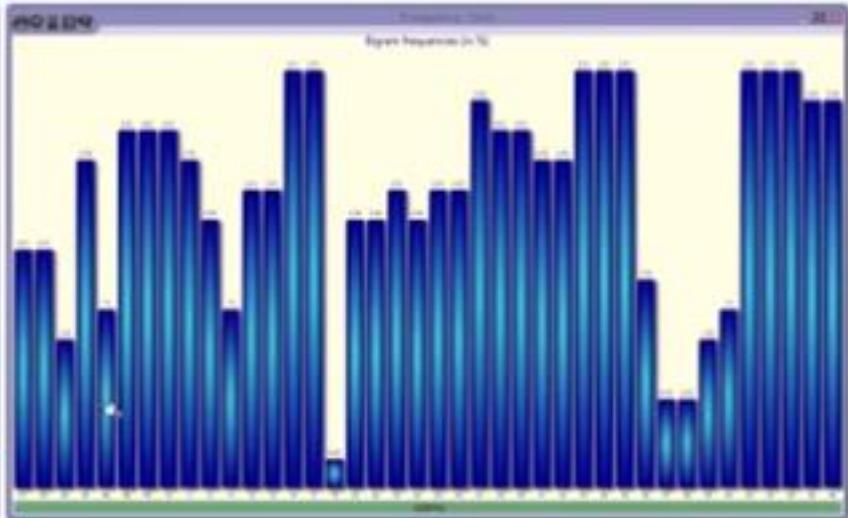
Homophonic Substitution



Frequencies of two digit combinations



Simple substitution cipher
27 different ciphertext symbols in total



Homophonic substitution cipher – homophones based on language frequencies
41 different ciphertext symbols in total



Homophonic Substitution

- Homophonic substitution ciphers were not only built based on single letters
- To improve security, additional cipher elements were introduced

1. Doubles / double letters	AA, NN, TT, LL, SS, ...
2. Other frequently used bigrams (or higher order n-grams)	TH, HE, IN, EN, NT, RE, ..., ING, THE,
3. Nulls → Cipher elements that encode nothing to	
a) Further change frequencies	
b) Confuse attackers	
c) Mark "special" cipher elements, e.g. nomenclature elements (= code words)	
4. Complete Words	THE, AND, WITH, TO, BE, OF, FROM, IN, ...
5. Code Words (in nomenclators)	The King, the Pope, London, Maximilian, ...
• Encrypting (or even showing unencrypted) word separators (spaces) helps the cryptanalyst to break a cipher	
• Creating a code/cipher which shows separations of ciphertext symbols helps the cryptanalyst to break a code/cipher	
• Ordering in a code/cipher helps the cryptanalyst to break a code/cipher	



Homophonic Substitution

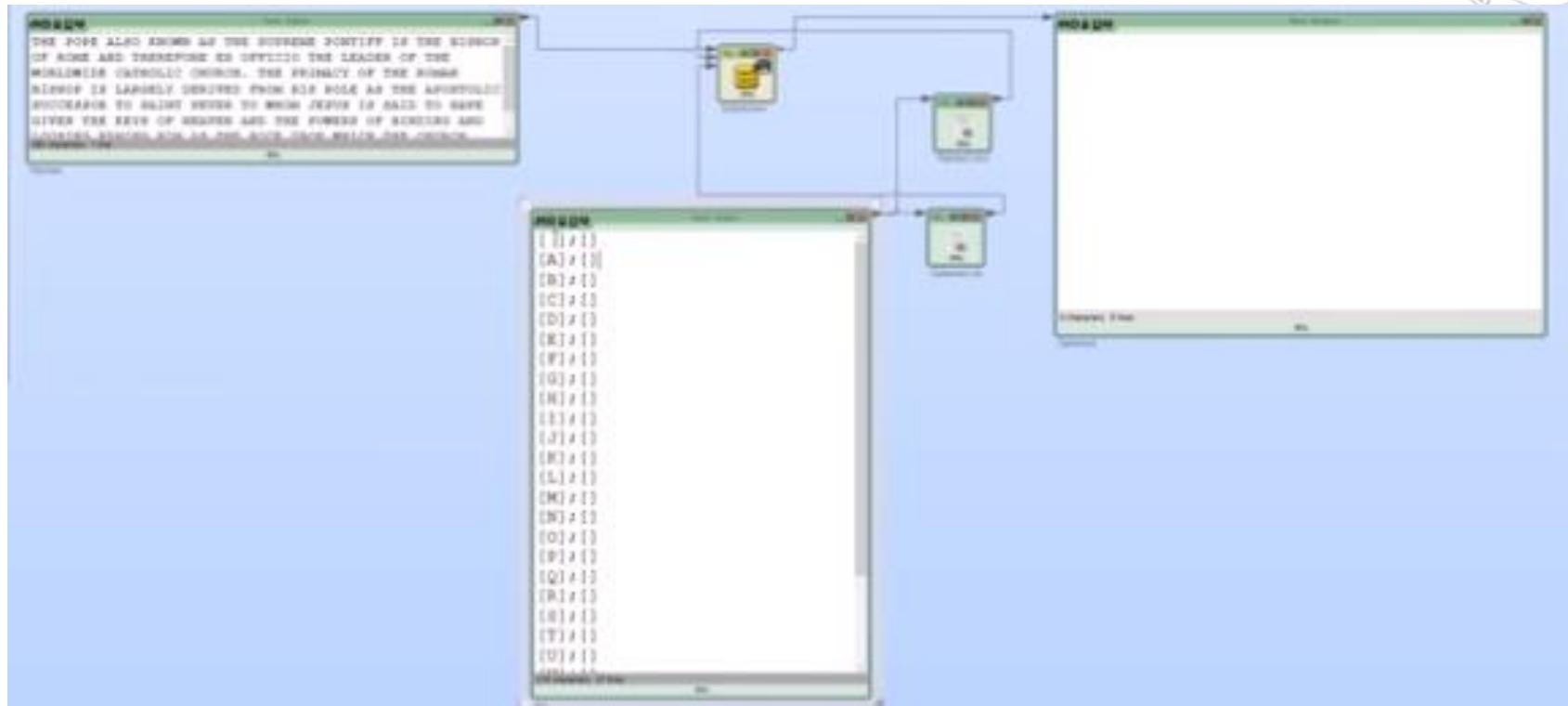
- Nomenclator (ciphers) contain a **nomenclature** (= code word table)

Example:

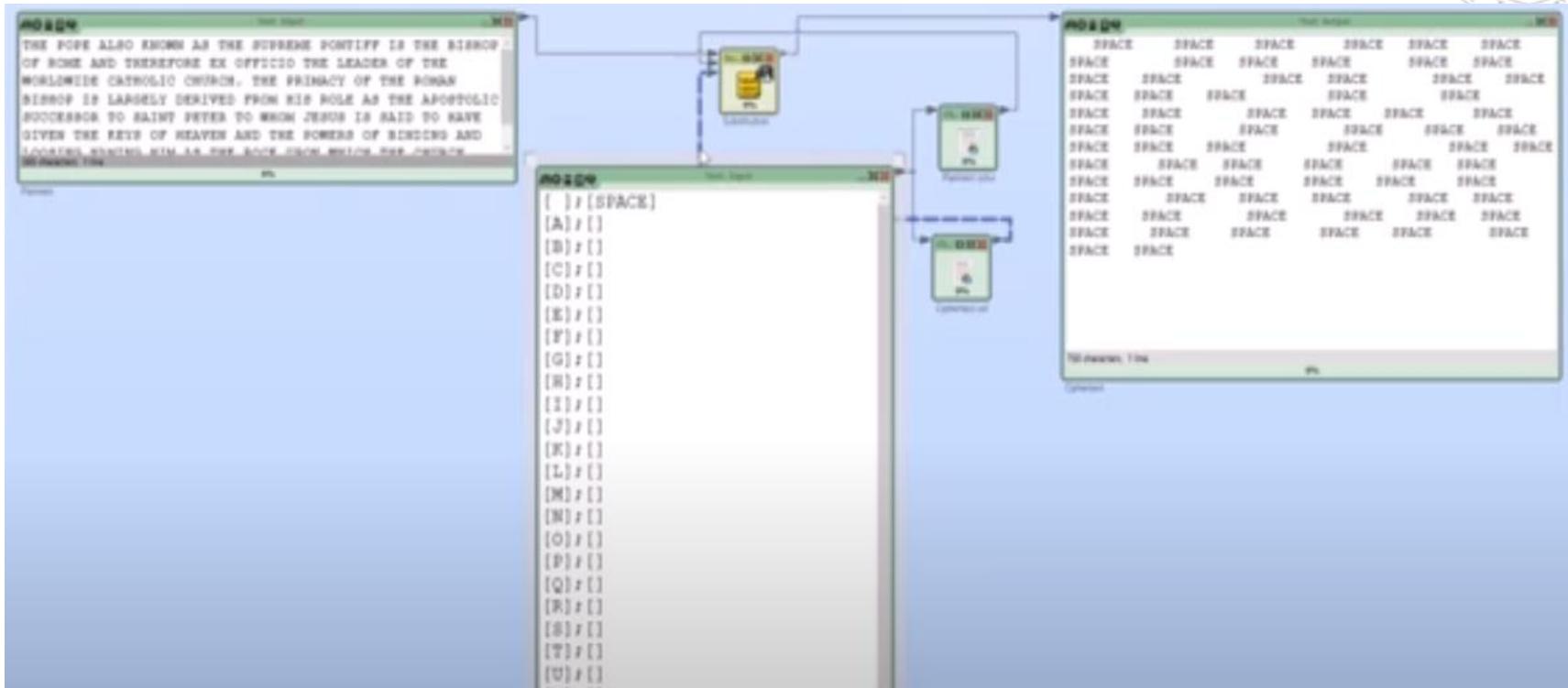
Cipher:						
A	01 02 03	B	03 04 05	C	06 07 08	...
AA	11 12	LL	13 14	TT	15 16	
Nulls:						
70 71 72 73 74 75 76 77 78 79						
Nomenclature:						
THE	840 841	AND	842 843	WITH	844 845	...
9100	The King	9200	One		We need help	9400
9101	The Pope	9201	Two		We attack at	9401
9102	Germany	9202	Three		The enemy is at	9402
9103	France	9203	Four		We will meet at	9403
9104	England	9204	Five		We need urgently	9404
9105	The Netherlands	9206	Six		We provide cover	9405



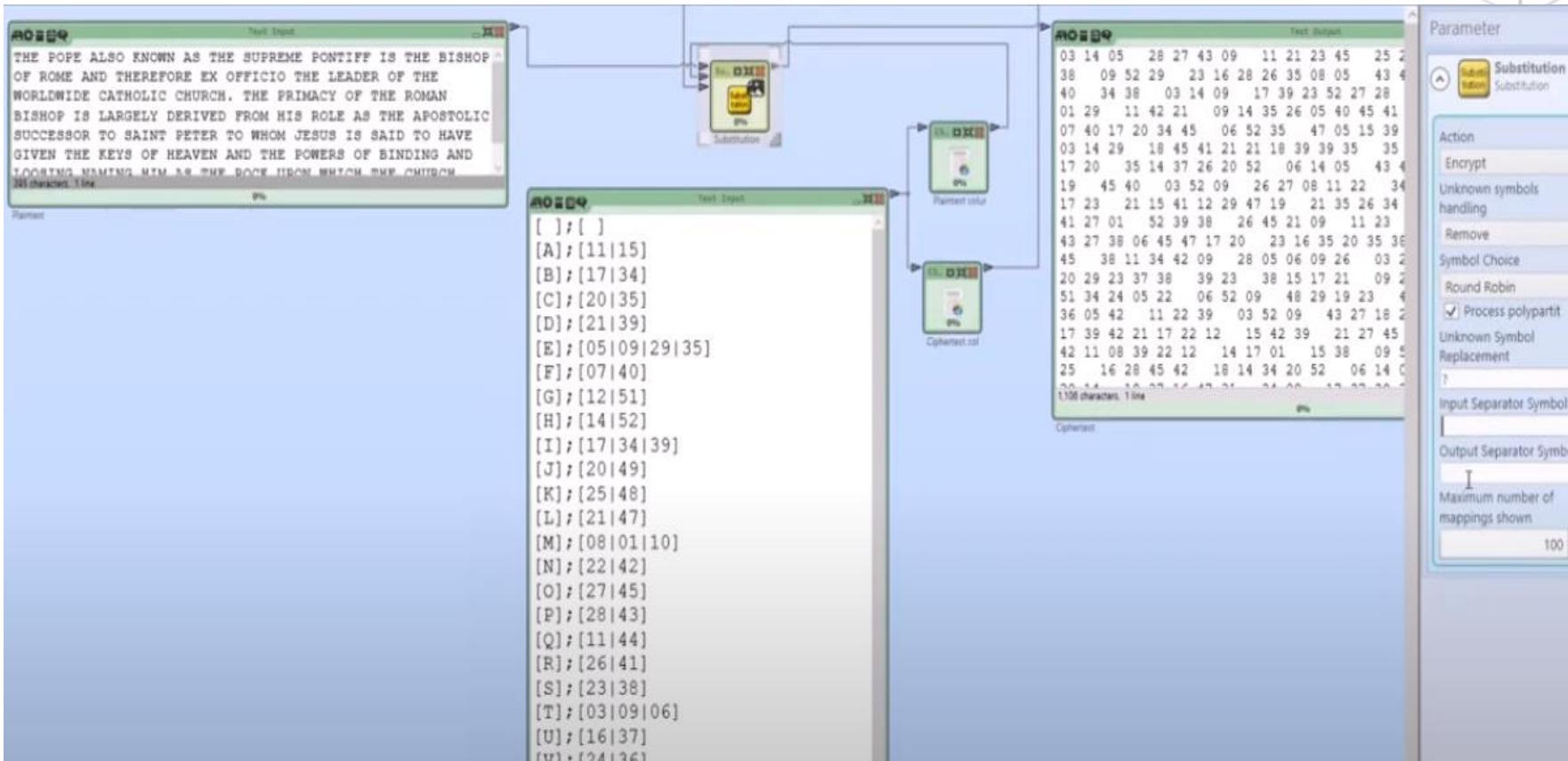
Homophonic Substitution



Homophonic Substitution



Homophonic Substitution



Homophonic Substitution



[] # [00]

Text		Output	
03	14	05	00
28	27	43	09
00	11	21	23
45	00	25	22
22	27	18	42
00	15	38	00
09	52	29	00
23	16	28	26
35	08	05	00
43	45	22	
06	17	07	40
00	34	38	00
03	14	09	00
17	39	23	52
27	28	00	
45	07	00	41
27	01	29	00
11	42	21	00
09	14	35	26
05	40	45	
41	09	00	29
15	00	27	07
40	17	20	34
45	00	06	52
35	52	35	00
47			
05	15	39	09
26	00	27	07
00	03	14	29
00	18	45	41
21	21	18	
39	39	35	00
00	35	11	09
52	27	47	17
17	20	00	35
14	37	14	37
26	20	52	22
52	00	35	26
06	14	43	41
34	10	15	35
19	00	45	40
00	00	03	52
09			09
00	26	27	08
11	22	00	34
39	38	14	45
45	28	00	17
00	23	00	21
15			
41	12	29	47
19	00	21	35
26	34	24	05
39	00	07	41
00	27	01	00
52	39	38	00
00	26	45	21
21	09	00	11
23	00	14	29
00	00	00	15
43	27	18	29
41	38	00	45
07	00	17	39
42	42	21	17
22	17	22	12
12	00	00	15
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11	08	39	22
12	00	14	
42	39	00	21
27	45	23	34
22	51	00	42
11</td			



Homophonic Substitution

The screenshot shows a software interface for performing homophonic substitution. On the left, a sidebar titled "Substitution" contains the following options:

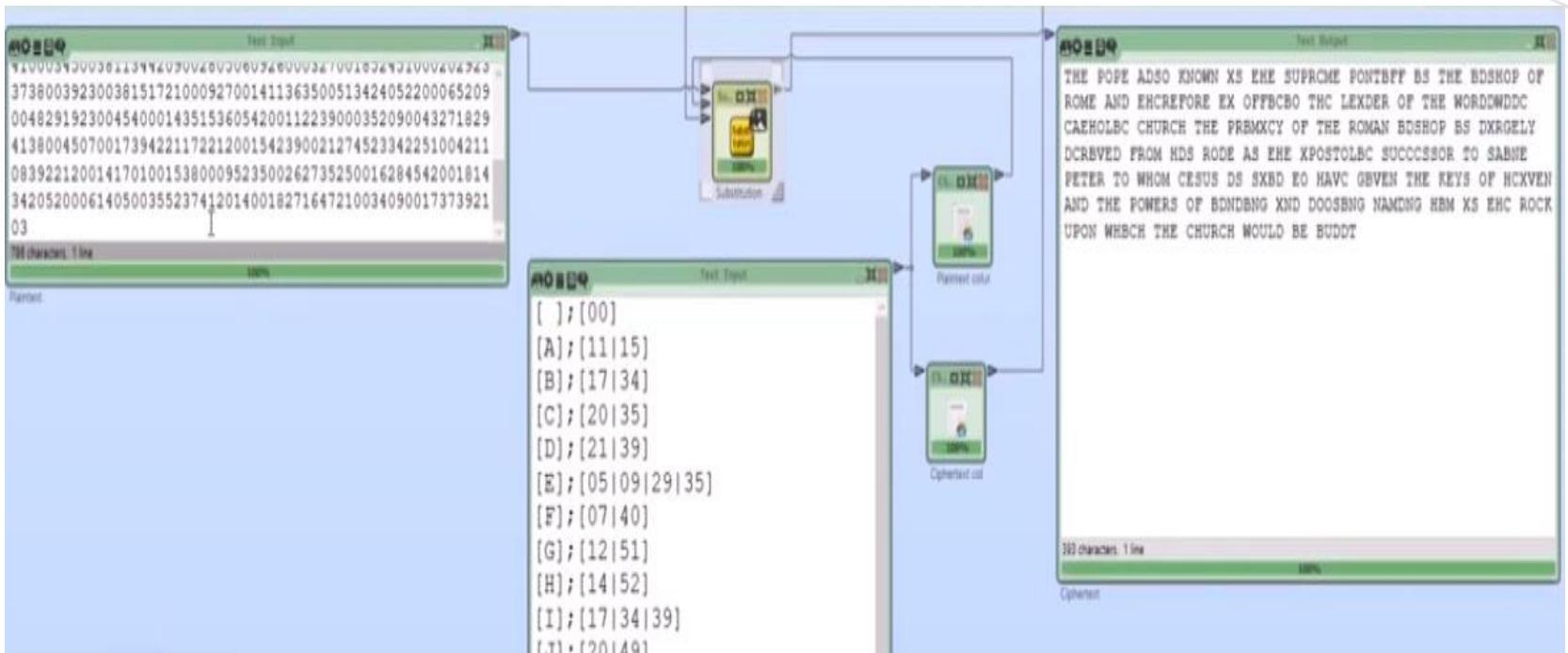
- Action: Encrypt
- Unknown symbols handling
- Remove
- Symbol Choice
- Round Robin
- Process polypartite
- Unknown Symbol Replacement
- ?
- Input Separator Symbol: A dropdown menu showing a list of symbols.
- Output Separator Symbol: A dropdown menu showing a list of symbols, currently set to a vertical bar (|).
- Maximum number of mappings shown: A text input field set to 100.

On the right, the main window displays the "Text" tab with the title "MOEISQ". The text area contains a large amount of encoded data, consisting of a sequence of digits and symbols. Below the text area, it says "786 characters, 1 line". At the bottom, there is a progress bar at 100% completion and a status bar with the word "Ciphertext".

```
031405002827430900112123450025222718420015380009522900231628  
263508050043452206170740003438000314090017392352272800450700  
412701290011422100091435260540454109002915002707401720344500  
065235004705153909260027070003142900184541212116393935003511  
095227471720003514372620520006140500434134101535190045400003  
520900262708112200343938144528001723002115411229471900213526  
342405390007412701005239380026452109001123000914290015432738  
064547172000231635203538232741000345003811344209002805060926  
000327001852451000202923373800392300381517210009270014113635  
005134240522000652090048291923004540001435153605420011223900  
035209004327182941380045070017394221172212001542390021274523  
342251004211083922120014170100153800095235002627352500162845  
420018143420520006140500355237412014001827164721003409001737  
392103
```



Homophonic Substitution



Homophonic Substitution

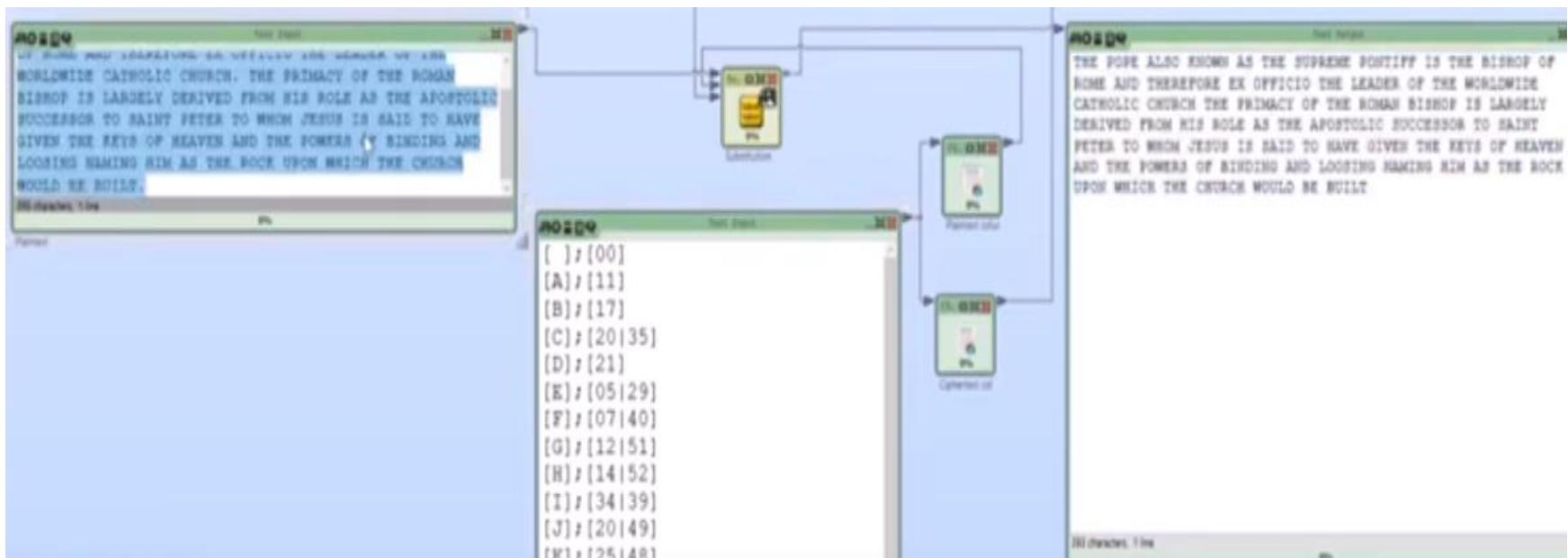


```
ROSEB  
Text Input  
[ ] ; [00]  
[A] ; [11]  
[B] ; [17]  
[C] ; [20|35]  
[D] ; [21]  
[E] ; [05|29]  
[F] ; [07|40]  
[G] ; [12|51]  
[H] ; [14|52]  
[I] ; [34|39]  
[J] ; [20|49]  
[K] ; [25|48]  
[L] ; [47]  
[M] ; [08|01|10]  
[N] ; [22|42]  
[O] ; [27|45]  
[P] ; [28|43]  
[Q] ; [44]  
[R] ; [26|41]  
[S] ; [23|38]  
[T] ; [03|09|06]  
[U] ; [16|37]  
[V] ; [24|36]  
[W] ; [18]  
[X] ; [15]  
[Y] ; [19]  
[Z] ; [13]
```

```
ROSEB  
Text Input  
[I] ; [34|39]  
[J] ; [20|49]  
[K] ; [25|48]  
[L] ; [47]  
[M] ; [08|01|10]  
[N] ; [22|42]  
[O] ; [27|45]  
[P] ; [28|43]  
[Q] ; [44]  
[R] ; [26|41]  
[S] ; [23|38]  
[T] ; [03|09|06]  
[U] ; [16|37]  
[V] ; [24|36]  
[W] ; [18]  
[X] ; [15]  
[Y] ; [19]  
[Z] ; [13]  
  
[THE] ; [601]  
[AND] ; [602]  
[TO] ; [603]  
[OF] ; [604]  
  
[BISHOP] ; [7001]  
[ROME] ; [7002]  
[CHURCH] ; [7003]  
[JESUS] ; [7004]
```

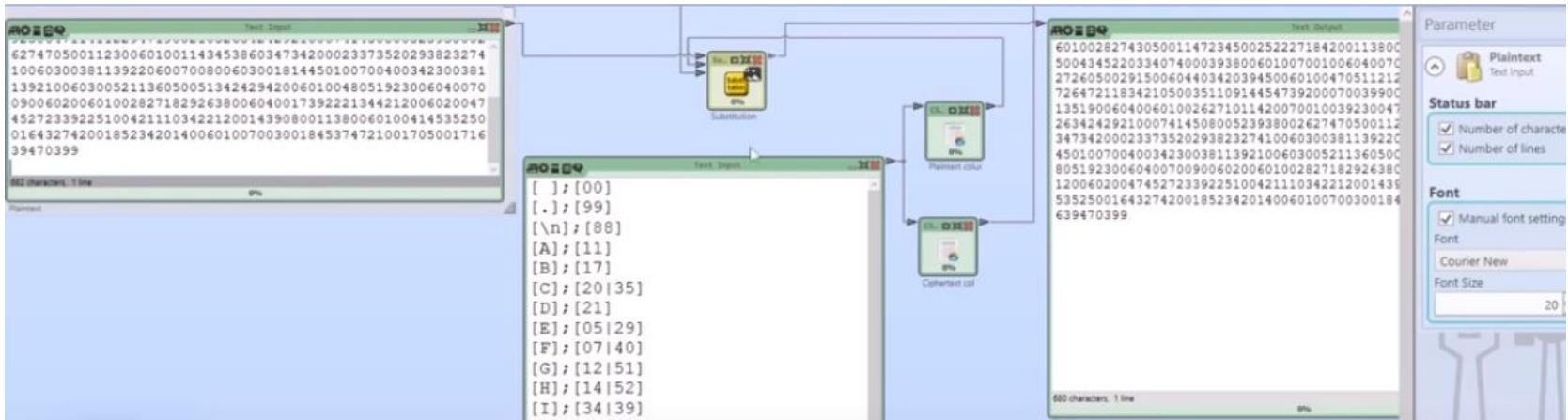


Homophonic Substitution





Homophonic Substitution



- Try to break your own code by using the Hill Climbing technique !



Cryptology for IoT

Modules M4, M6, M8
Session of 24th May, 2022.

M6.1 Briefing of the session

M6.2 Friedman Test

M6.3 Hill Climbing

M6.4 Final Exercise using Hill Climbing

Prof.: Guillermo Botella