



# Cryptology for IoT

**Modules M4, M6, M8**  
**Session of 10th May, 2022.**

M4.6 Briefing of the session  
M4.7 Tasks to do in the lab

Prof.: Guillermo Botella



# Cryptology for IoT

**Modules M4, M6, M8**  
**Session of 10th May, 2022.**

**M4.6 Briefing of the session**  
**M4.7 Tasks to do in the lab**

Prof.: Guillermo Botella



# M4.6 Briefing of today

- Cryptography and Cryptoanalysis
  - Slides and supplementary videos
- We go to the rooms. Practical Session I.
  - Assignments
    - (They will be specified when we start)
  - Work in groups
    - (Same than usual)



# Cryptology for IoT

**Modules M4, M6, M8**  
**Session of 10th May, 2022.**

M4.6 Briefing of the session  
**M4.7 Tasks to do in the lab**

Prof.: Guillermo Botella



# Slides and videos

- Cryptography using Cryptool
- Cryptoanalysis using Cryptool
- Substitution ciphers lab
  - Caesar (trivial case)
    - Monoalphabetic Substitution
    - Polyalphabetic Substitution
- Transposition Ciphers lab
  - Scytale (basic case)
    - Columnar Transposition
- Mixed Ciphers lab
  - ADVGX Cipher



# Basic Crypto I

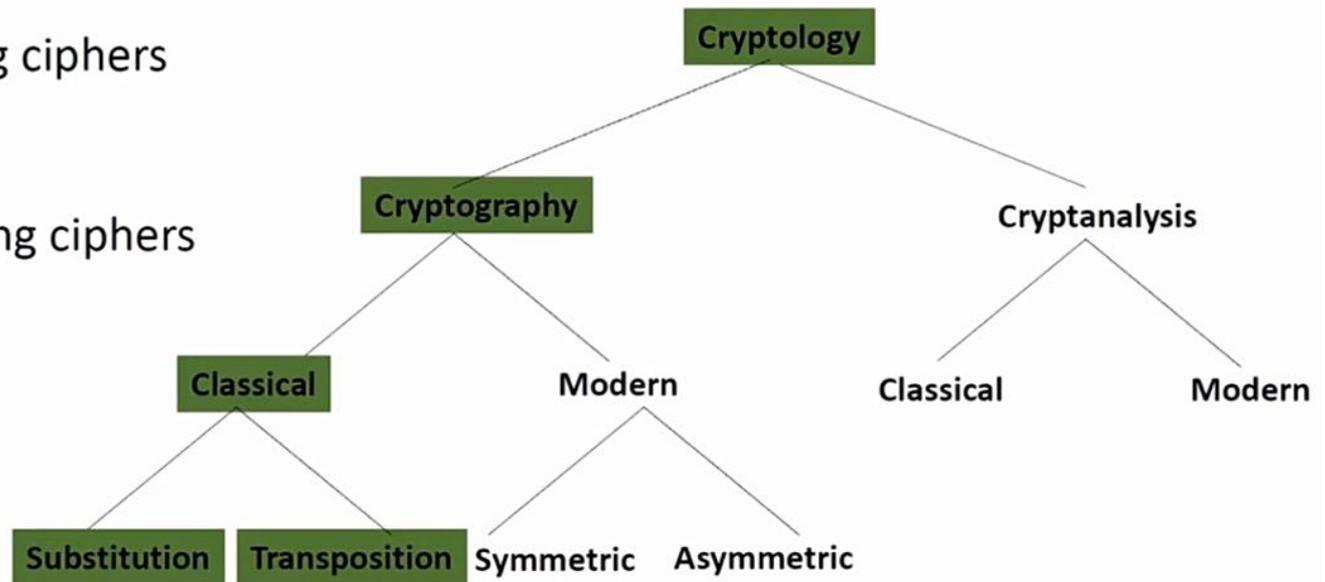
- **Cryptography using Cryptool (video+slides)**
  - Family ciphers
  - Classical

## Cryptography

Art of making ciphers

## Cryptanalysis

Art of breaking ciphers





# Basic Crypto I

## ■ Cryptography using Cryptool (video+slides)

### – Terms

#### **Cipher**

- Encryption method/algorithm

#### **Plaintext**

- Non-encrypted text

#### **Ciphertext**

- Encrypted text

#### **Key**

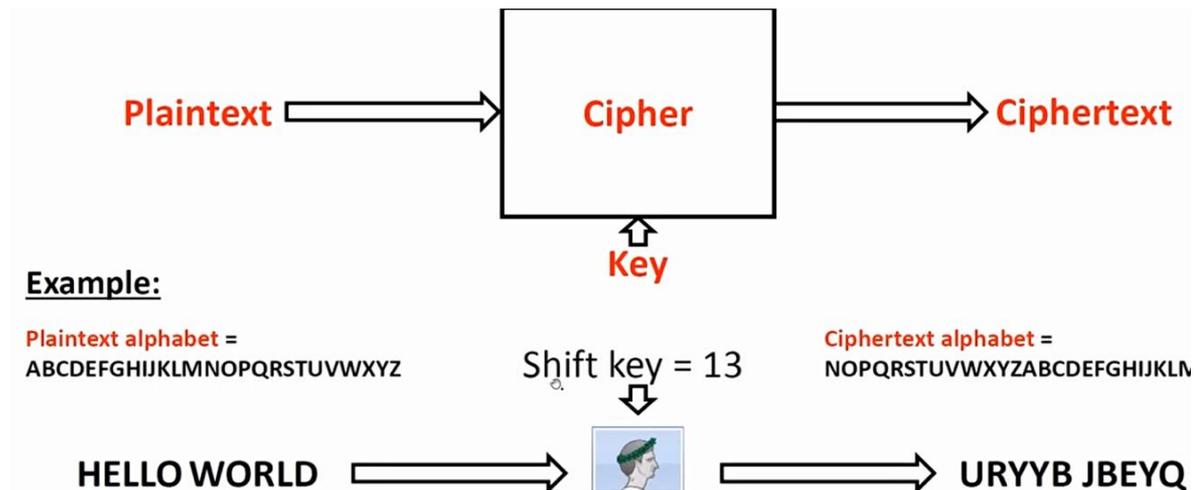
- Secret information used for encryption/needed for decryption

#### **Alphabet (plaintext alphabet & ciphertext alphabet)**



# Basic Crypto I

- **Cryptography using Cryptool (video+slides)**
  - Caesar's Scheme





# Basic Crypto I

## ■ Cryptography using Cryptool (video+slides)

### – Types of classical ciphers

Three types of (classical) ciphers. Two main types (1 & 2)

#### 1. Substitution ciphers

- Replace letters by other letters (or symbols)
- Examples: Caesar, simple MASC, Vigenère

#### 2. Transposition ciphers

- Change the order of the plaintext letters
- Examples: Scytale, columnar transposition

#### 3. Composed ciphers

- Combination of substitution and transposition
- Examples: ADFGVX, Granite



# Basic Crypto I

## ■ Cryptography using Cryptool (video+slides) – Terms (ii)

### Keyspace

- Set of all possible keys of a cipher

### Keyspace size

- Size of the set of all possible keys of a cipher
- Usually given as (rounded up) power of 2

26

### Example: Caesar

$$\text{Keyspace size} = 26 \approx 2^5$$

Keyspace = { 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 }  
→ all possible shift keys, including the identity (shift key = 0)



# Basic Crypto I

## ■ Cryptography using Cryptool (video+slides) – Terms (iii)

### **Monoalphabetic Substitution**

- Only one ciphertext alphabet is used
- Examples: Caesar cipher, simple MASC

### **Polyalphabetic Substitution**

- The ciphertext alphabet is changed during encryption
- Examples: Vigenère cipher, Enigma machine

### **Homophonic Substitution**

- A letter is encrypted by more than one letter/symbol
- Examples: Zodiac killer ciphers, historic ciphers of the Vatican

### **Polyphonic Substitution**

- Different plaintext letters are encrypted by the same ciphertext
- Non-deterministic. Decryption ambiguous



# Basic Crypto I

## ■ Cryptography using Cryptool (video+slides) – Terms (iv)

### Monographic cipher

- One letter is encrypted at the same time

### Bigraphic cipher

- Letter pairs are encrypted at the same time

### Monopartite cipher

- Replacement is a single letter

### Bipartite cipher

- Replacement are two letters

### Example:

The simple monoalphabetic substitution cipher (simple MASC) is a monoalphabetic monographic monopartite substitution cipher



# Basic Crypto I

- **Cryptography using Cryptool (video+slides)**
  - Substitution cipher → Caesar

And a first example for simple substitution cipher is the Caesar cipher. Just double

CRYPTOOL 2

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 demonstrate the program functionality in cryptographic scenarios.

**Main functions**

- Create a new workspace with the graphical editor
- Use the wizard to easily try some Cryptool 2 features
- Open Cryptool Store
- Read the online documentation
- Open the Cryptool Book

**Templates (double click to open)**

- Cryptography
  - Classical
    - ADFGVX Cipher
    - Caesar Cipher
    - Homophonic Substitution Cipher and Nomenclature -- Decryption
    - Homophonic Substitution Cipher and Nomenclature -- Encryption
    - Enigma Cipher Machine
      - Fialka ED Check
      - Fialka ED Mixed Mode Check
      - Fialka ED NumLock10 Check
      - Fialka Key Output

**YouTube Videos (double click to open)**

- Break the Double Columnar Transposition Challenge (Doppelwürfel)
- Cryptool 2 YouTube Channel Trailer
- Enigma Machine - Part 2 of 2 - Let's break it!
- 08 - Upload Component to CryptoolStore (Cryptool 2 Development Series)
- Enigma Machine - Part 1 of 2 - How does it work?
- Break a Playfair Cipher
- Encrypt like Navajo Code Talkers
- Break a World War I ADFGVX Cipher

**Recently opened workspaces (double click to open)**



# Basic Crypto I

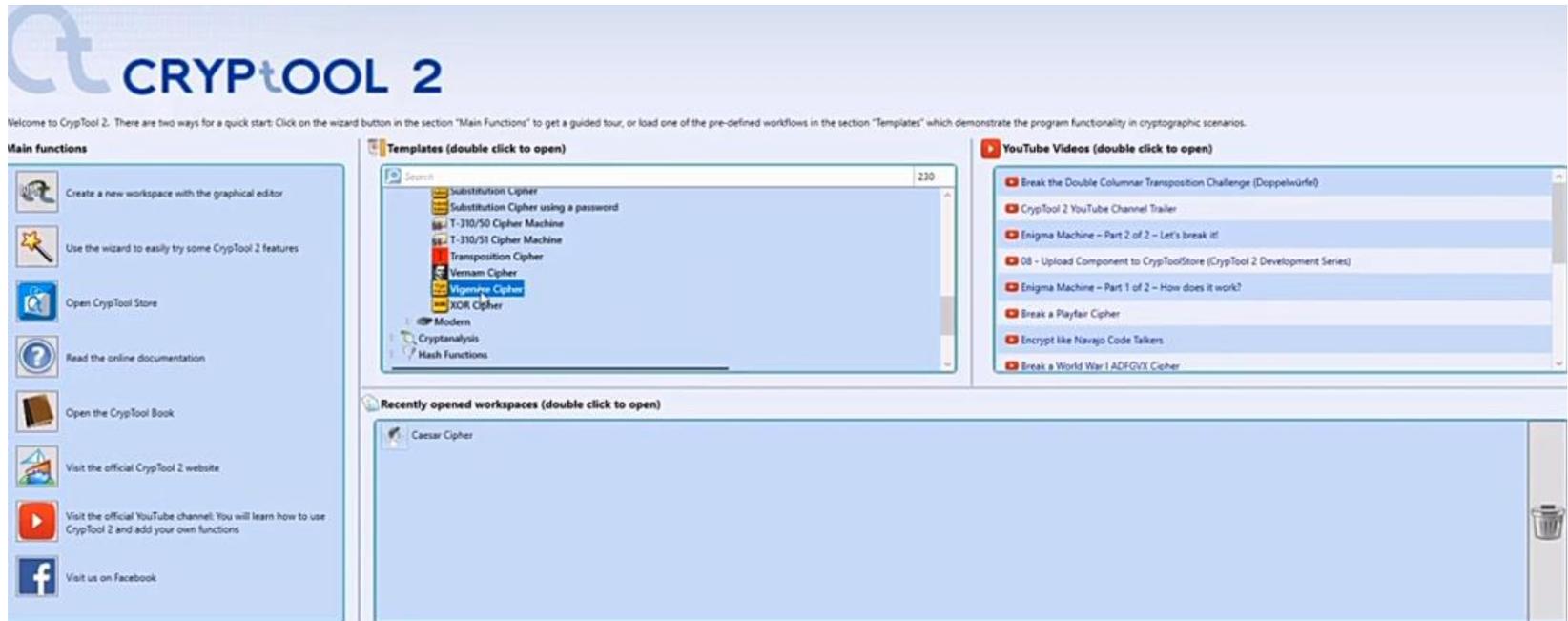
- **Cryptography using Cryptool (video+slides)**
  - Substitution cipher → Caesar

**Description of the cipher**

In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

# Basic Crypto I

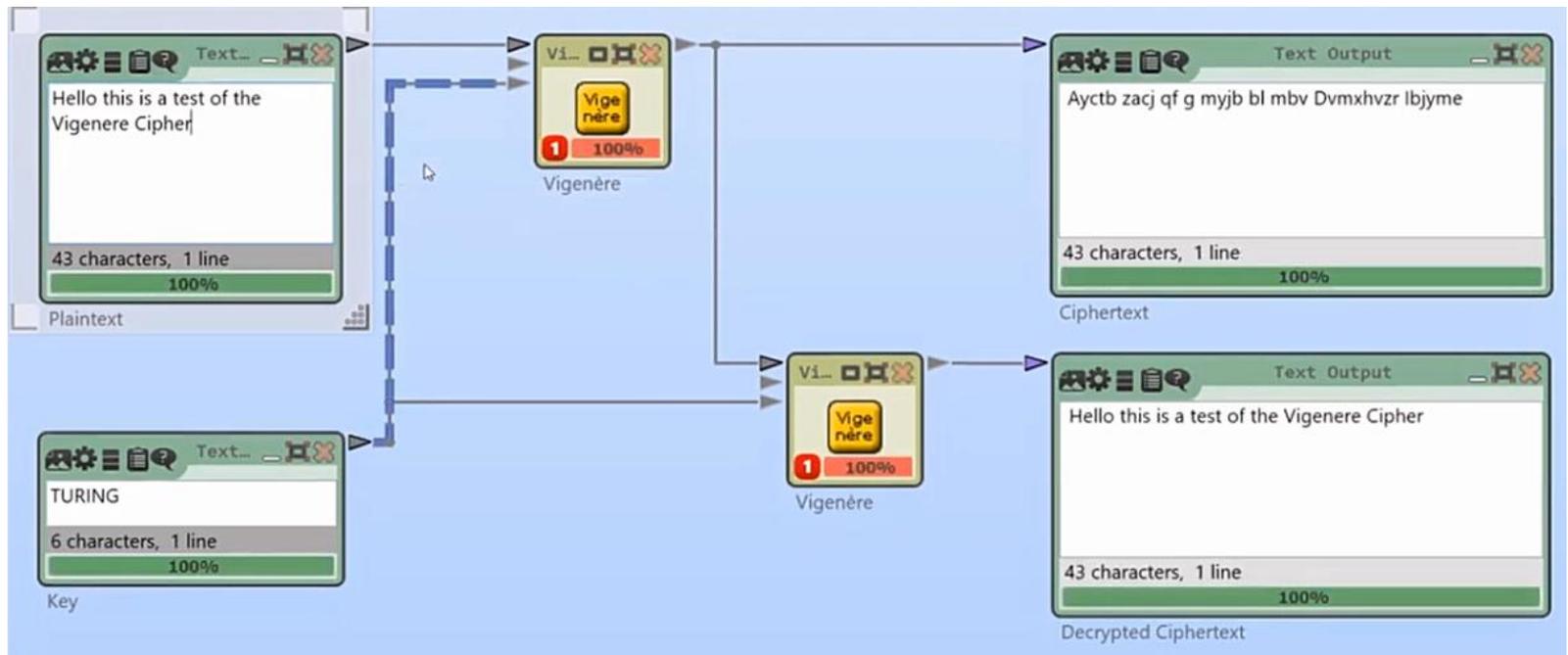
- **Cryptography using Cryptool (video+slides)**
  - Substitution cipher → Vigenere





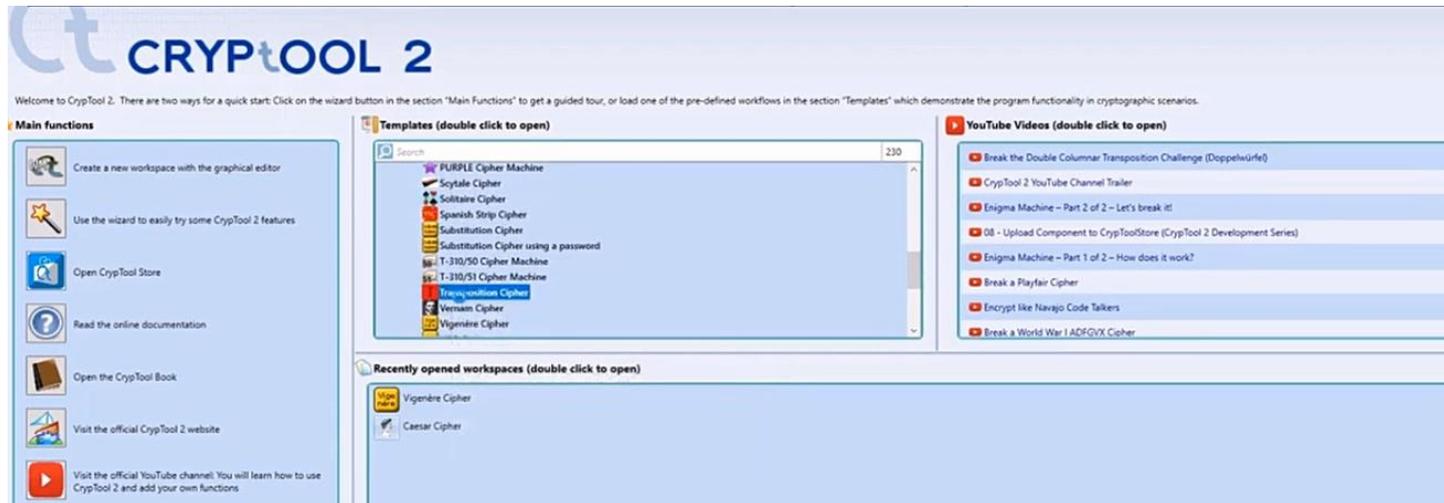
# Basic Crypto I

- **Cryptography using Cryptool (video+slides)**
  - Substitution cipher → Vigenere



# Basic Crypto I

- **Cryptography using Cryptool (video+slides)**
  - **Transposition cipher**





# Basic Crypto I

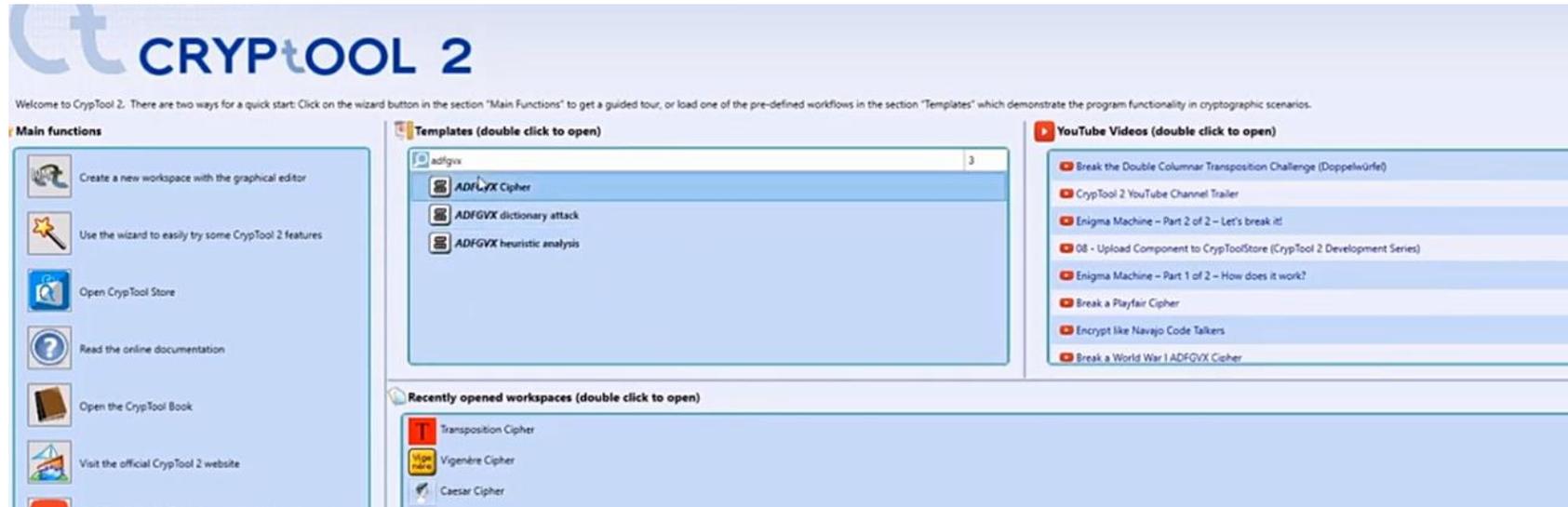
- Cryptography using Cryptool (video+slides)
  - Transposition cipher





# Basic Crypto I

- **Cryptography using Cryptool (video+slides)**
  - **Composed Cipher**





# Basic Crypto I

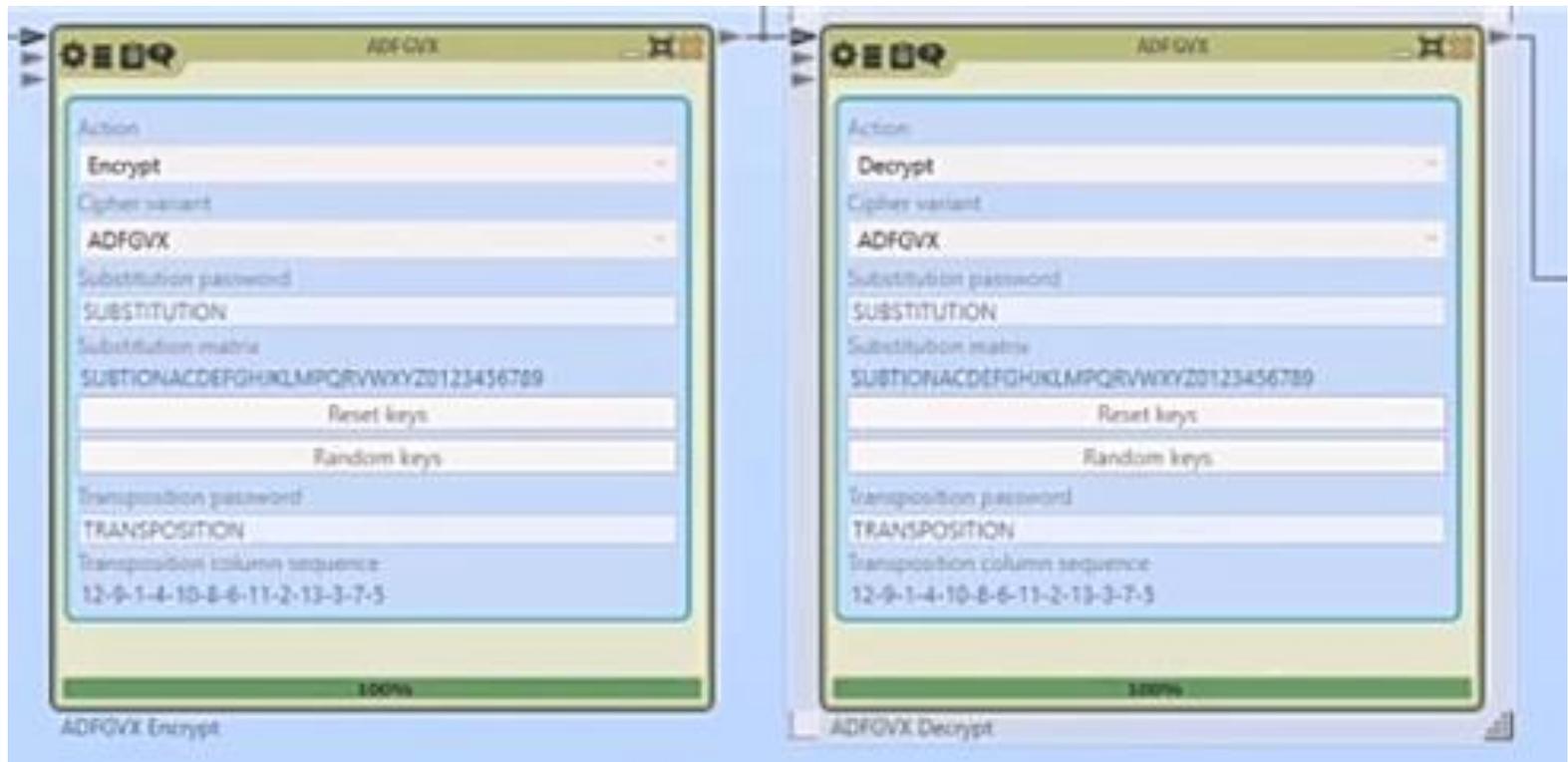
- Cryptography using Cryptool (video+slides)
  - Composed Cipher





# Basic Crypto I

- Cryptography using Cryptool (video+slides)
  - Composed Cipher





# Basic Crypto II

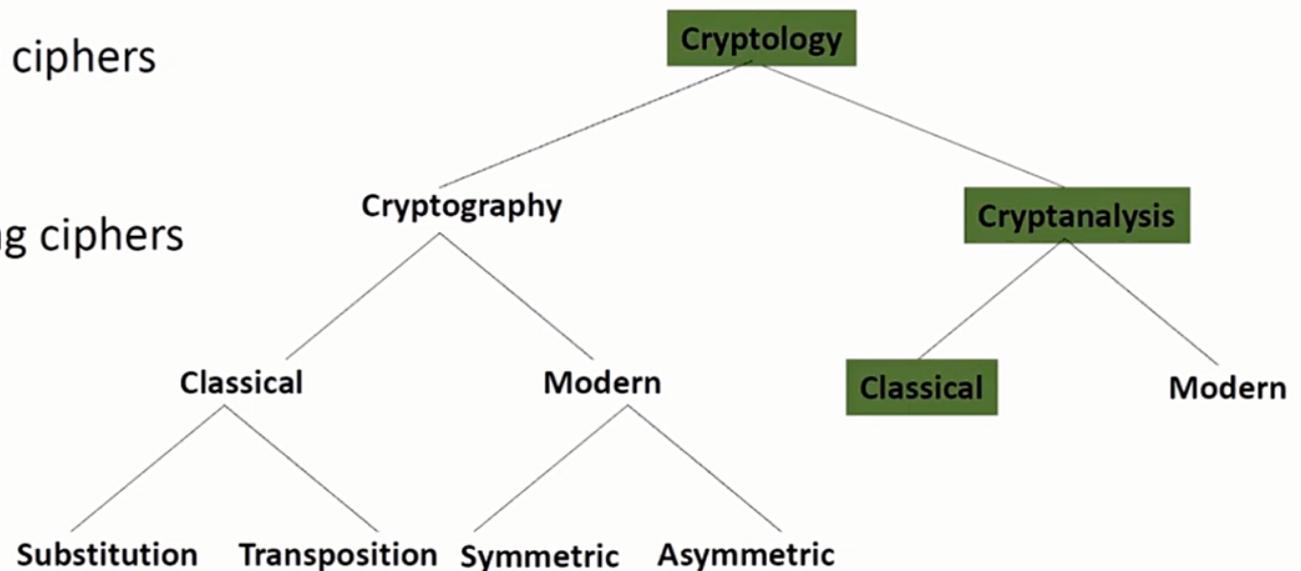
- **Cryptoanalysis using Cryptool (video+slides)**
  - Family ciphers
  - **Classical**

## Cryptography

Art of making ciphers

## Cryptanalysis

Art of breaking ciphers





# Basic Crypto II

## ■ Cryptography using Cryptool (video+slides) – Terms (i)

### Cryptanalyst

- Someone who analyzes a cipher/ciphertext to break it

### Attack

- Method to revert the key/plaintext of a ciphertext

### Breaking a ciphertext

- Successfully performed attack on a single ciphertext

### Breaking a cipher

- Finding an attack on a cipher that works reproducibly on ciphertexts encrypted with that particular cipher



# Basic Crypto II

## ■ Cryptography using Cryptool (video+slides) – Terms (ii)

### Assumption with each attack type

- “Attacker knows the system” (i.e. the used cipher; no security through obscurity)

### Chosen-plaintext attack

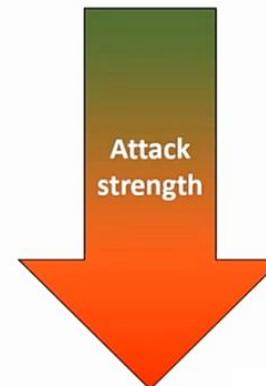
- Goal: revert the key
- Attacker is able to produce (arbitrary) plaintext-ciphertext pairs

### (Partially) Known-plaintext attack

- Goal: revert the key; revert the rest of unknown plaintext
- Attacker has (parts of) the plaintext of a ciphertext

### Ciphertext-only attack

- Goal: revert the key; revert the plaintext
- Attacker only is in possession of the ciphertext





# Basic Crypto II

## ■ Cryptography using Cryptool (video+slides)

### – Terms (iii)

#### Brute-force attack (aka exhaustive key search)

- Attack that works with every cipher (except perfect ciphers, e.g. the one-time pad)
- Attacker tests every key of the cipher
- Only suitable, if it's practical to search through the keyspace

#### Manual attacks (this video)

- E.g. break a MASC by hand using the knowledge of letter frequency distribution
- E.g. cut transposition ciphertext into paper strips and rearrange them

#### Computerized attacks (later videos)

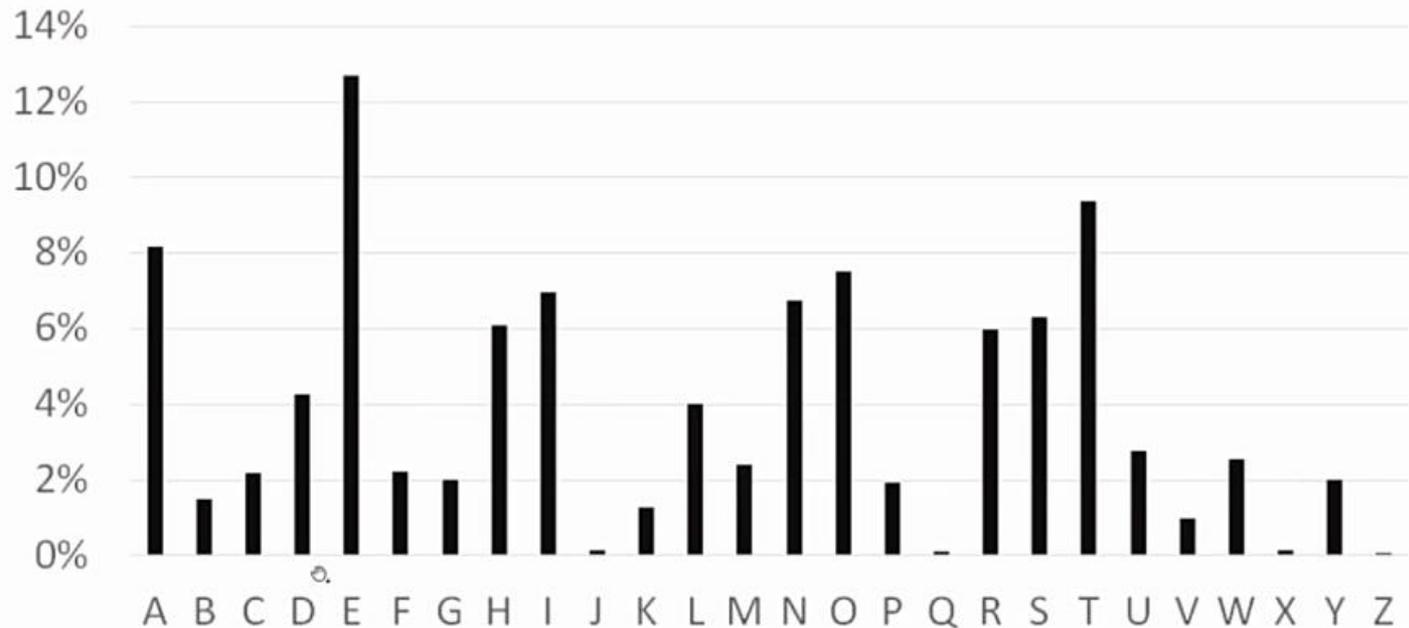
- Implementation of manual attacks, e.g. automated frequency analysis
- Heuristic attacks work on many classical ciphers, e.g. MASC, transposition, Enigma, ...



# Basic Crypto II

- **Cryptography using Cryptool (video+slides)**
  - **Statistic (i)**

Each language has its individual letter frequency distribution (here: all 26 English unigrams)

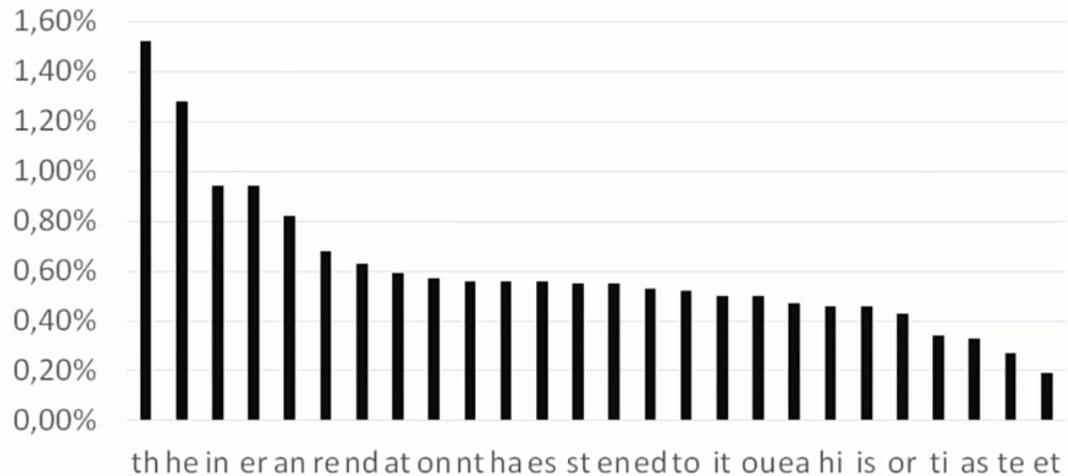




# Basic Crypto II

- **Cryptography using Cryptool (video+slides)**
  - **Statistic (ii)**

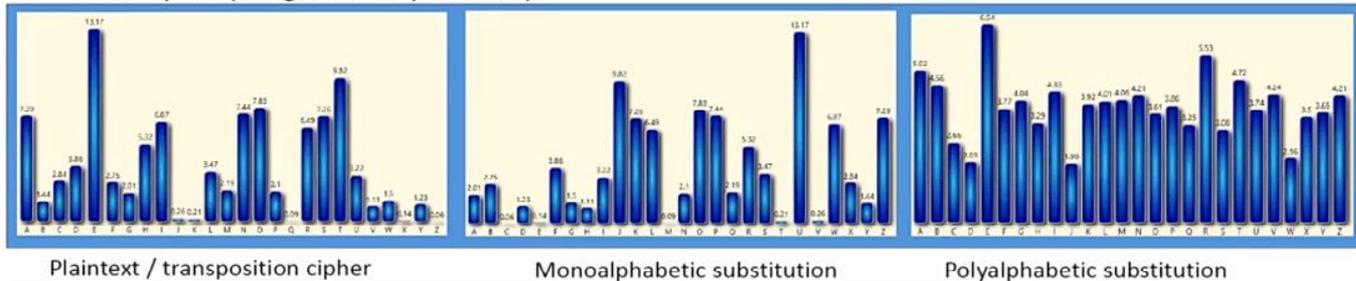
Each language has its individual letter frequency distribution (here: 39 most frequent English bigrams)



# Basic Crypto II

## ■ Cryptography using Cryptool (video+slides) – Statistic (iii)

- Ciphers try to flat the letter frequencies of the text
  - Substitution ciphers flat unigrams, bigrams, trigrams, etc.
  - Transposition ciphers **do not** flat unigrams, but flat bigrams, trigrams, etc.
- The flatter the frequencies, the more difficult is the analysis of a cipher
- Examples (unigram frequencies):





# Basic Crypto II

## ■ Cryptography using Cryptool (video+slides) – Statistic (iv)

Ciphertext (26 letters) = **B****U****U****B****D****L** **U****I****F** **F****O****F****N****Z** **J****O** **U****I****F** **F****W****F****O****J****O****H**

Count unigrams

<b>B</b> = 2	<b>D</b> = 1	<b>F</b> = 6	<b>H</b> = 1	<b>I</b> = 2	<b>J</b> = 2
<b>L</b> = 1	<b>N</b> = 1	<b>O</b> = 4	<b>U</b> = 4	<b>W</b> = 1	<b>Z</b> = 1

- Most frequent letter is “F”; assumption that “E” is encrypted to “F”

Look at bigrams, trigrams, and words

- Double letters “UU” may be “NN”, “LL”, or “TT”
- “JO” may be “IN”, “ON”, “AT”
- Word “UIF” may be “THE”; then, “UU” would be “TT”
- If “UU” is “TT”, then “BUUBDL UIF” may be “ATTACK THE”
- Following, “FOFNZ” may be “ENEMEY”
- Final solution: plaintext = “ATTACK THE ENEMY IN THE EVENING”

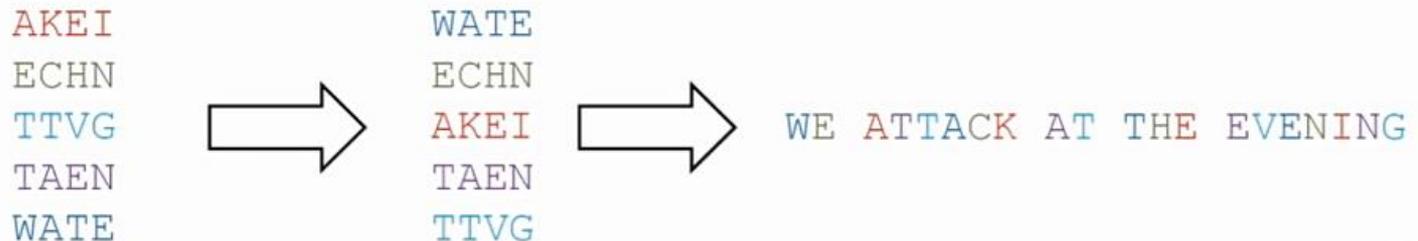


# Basic Crypto II

## ■ Cryptography using Cryptool (video+slides) – Transposition

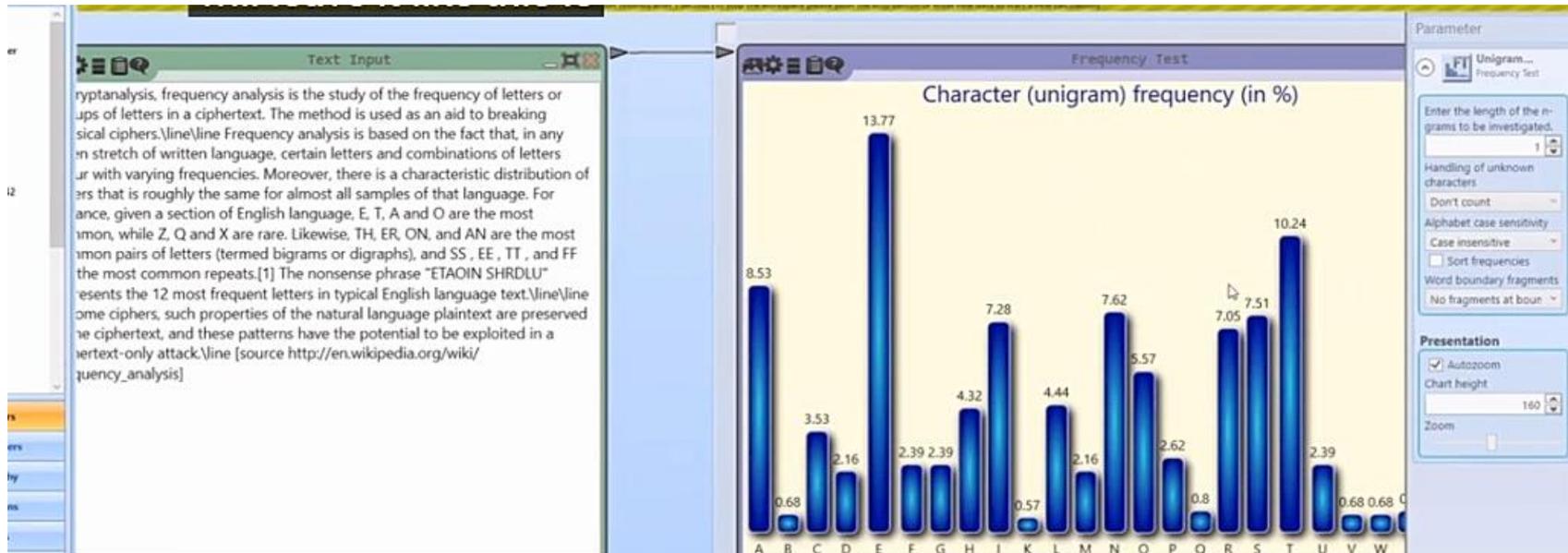
Ciphertext (20 letters) = AKEIECHNTTVGTAENWATE

1. Determine/assume key length; we assume key length = 5  
**Hint:** we have a regular transposition with key length = 5
2. Divide text into columns with length 5; i.e. row size = 4
3. Rearrange the rows to break the ciphertext  
we see „A“, „T“ and „T“; assumption „ATT(ack)“  
Also, „W“ and „E“ may be „WE“



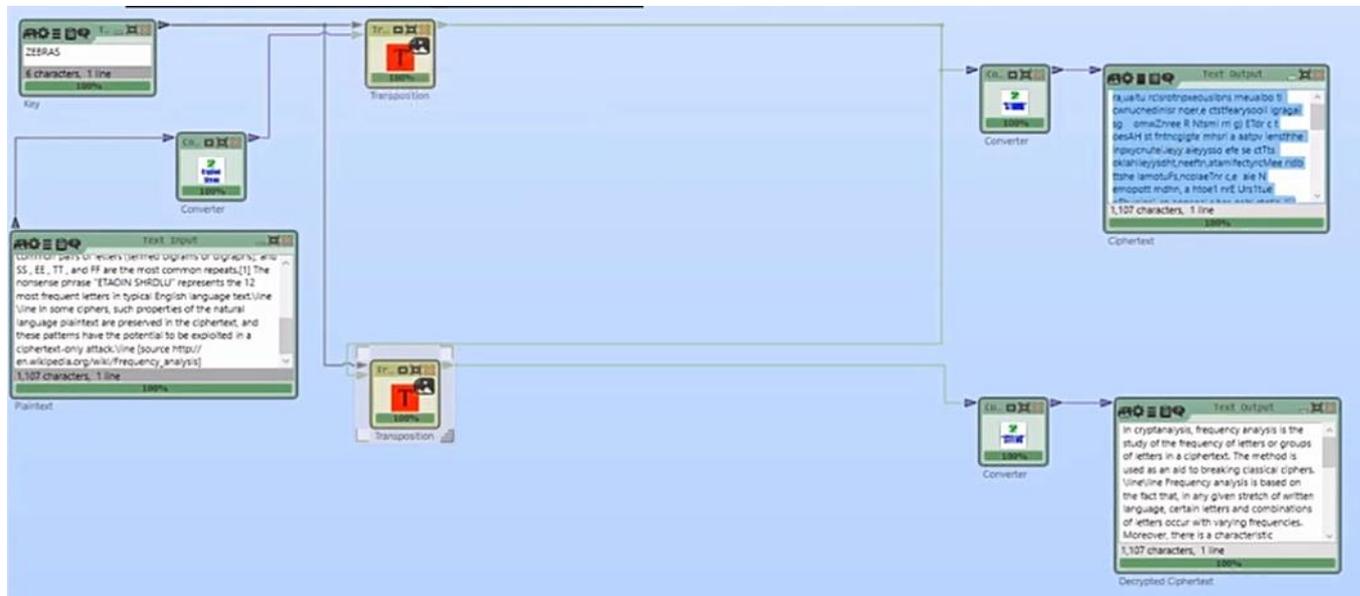
# Basic Crypto II

- **Cryptography using Cryptool (video+slides)**
  - Letter Frequency of ciphers (i)
  - Plain text



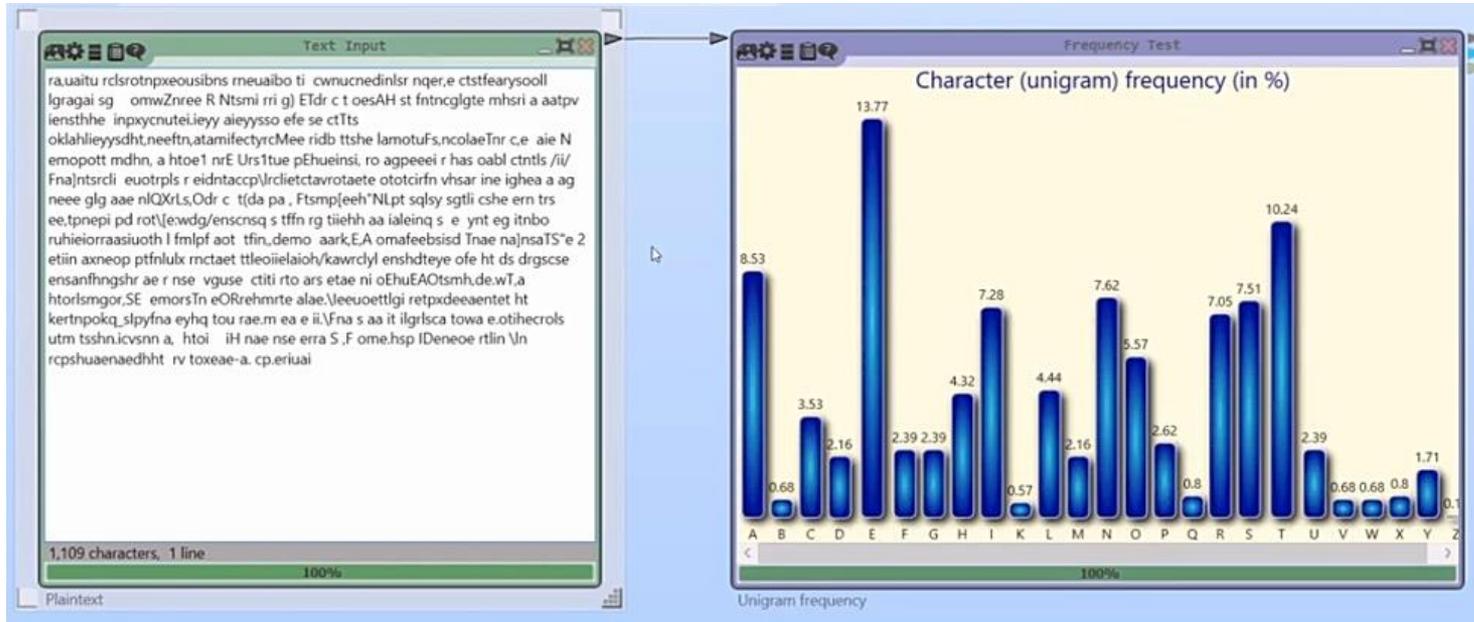
# Basic Crypto II

- Cryptography using Cryptool (video+slides)
  - Letter Frequency of ciphers (ii)
  - Transposition (i)



# Basic Crypto II

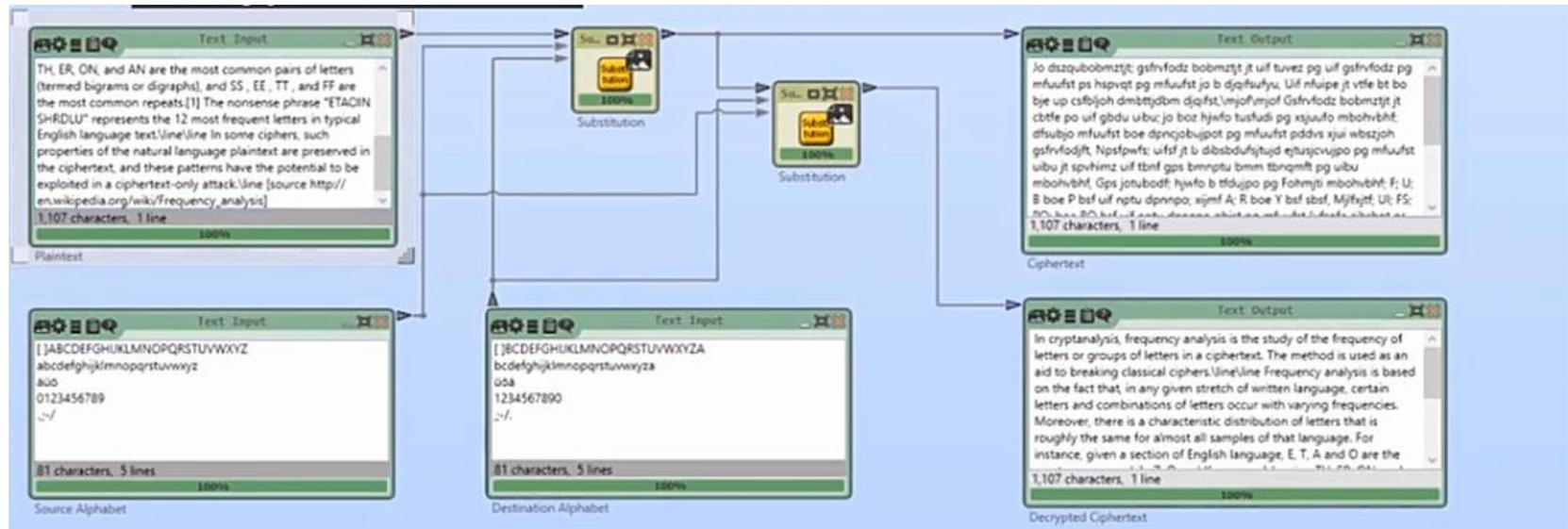
- **Cryptography using Cryptool (video+slides)**
  - Letter Frequency of ciphers (iii)
  - Transposition (ii)





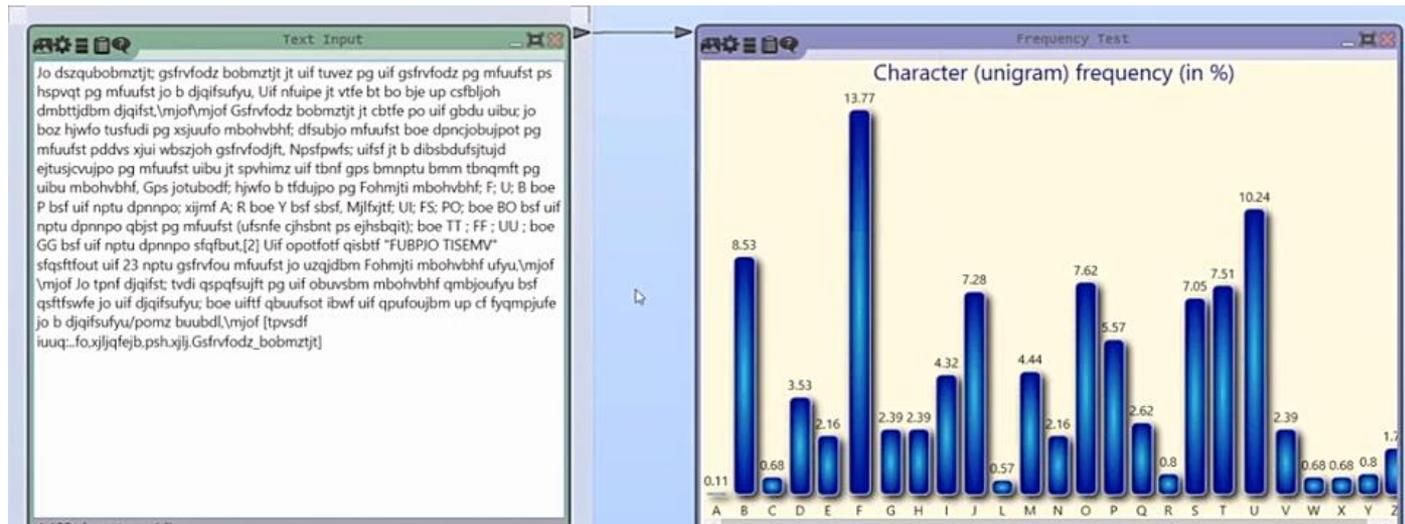
# Basic Crypto II

- Cryptography using Cryptool (video+slides)
  - Letter Frequency of ciphers (iv)
  - Substitution (i), no password



# Basic Crypto II

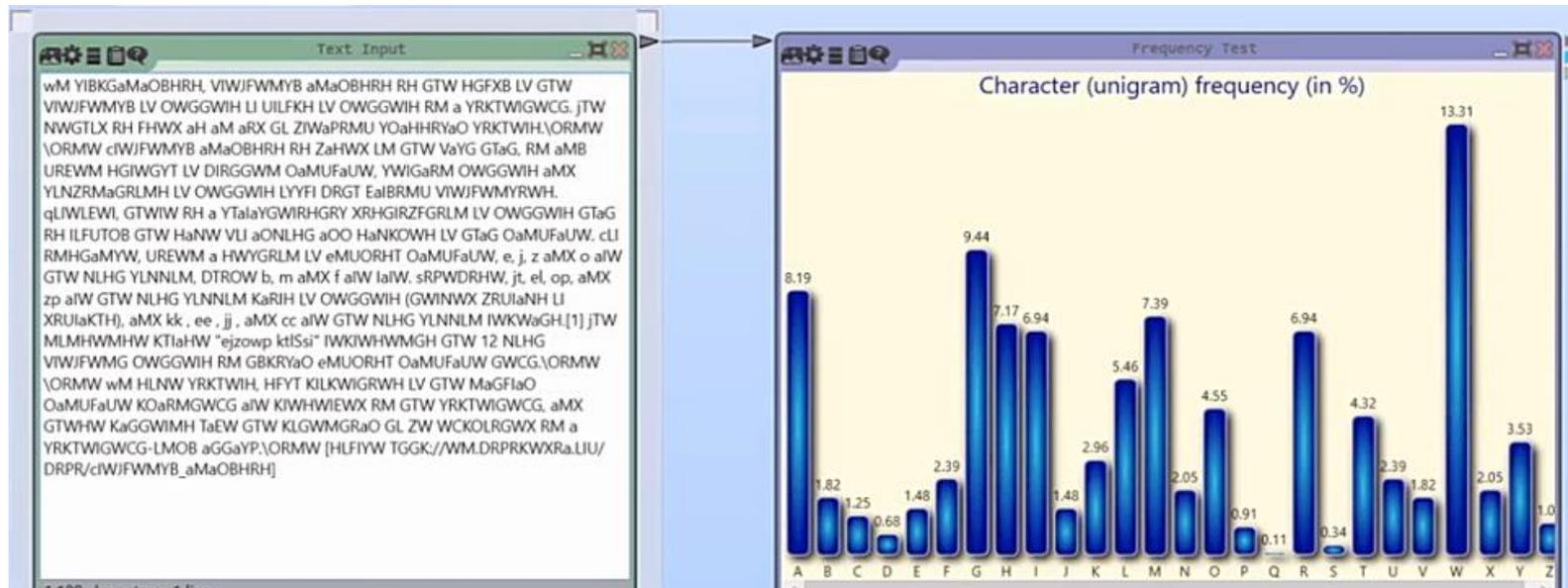
- **Cryptography using Cryptool (video+slides)**
  - Letter Frequency of ciphers (v)
  - Substitution (ii), no password





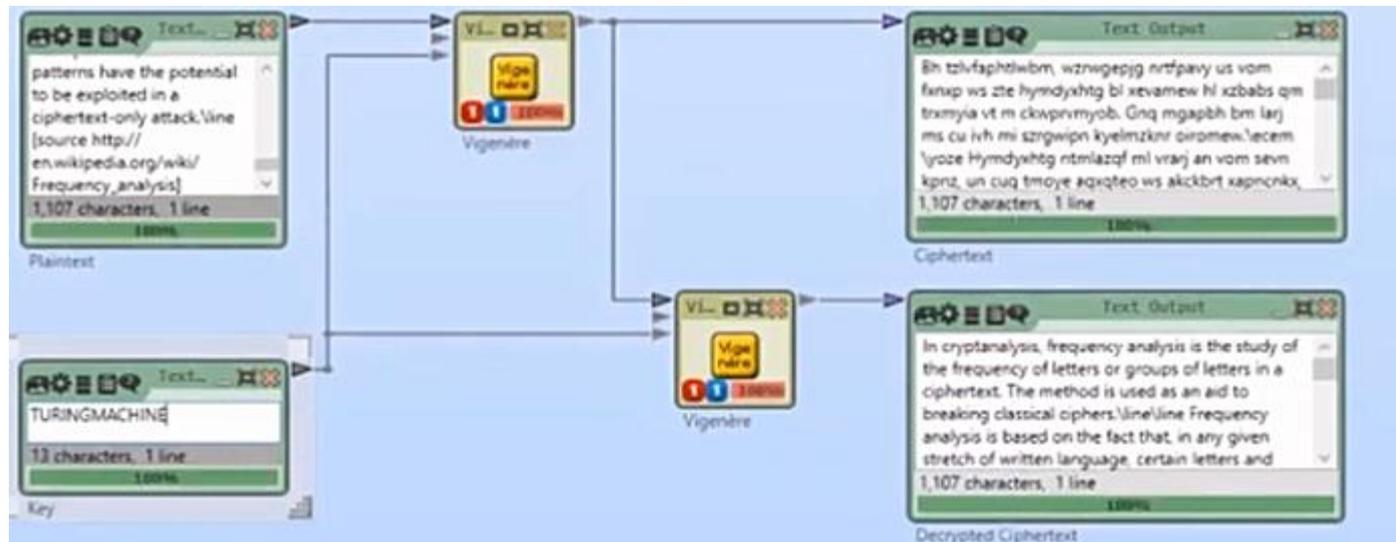
# Basic Crypto II

- **Cryptography using Cryptool (video+slides)**
  - Letter Frequency of ciphers (vii)
  - Substitution (iv), password



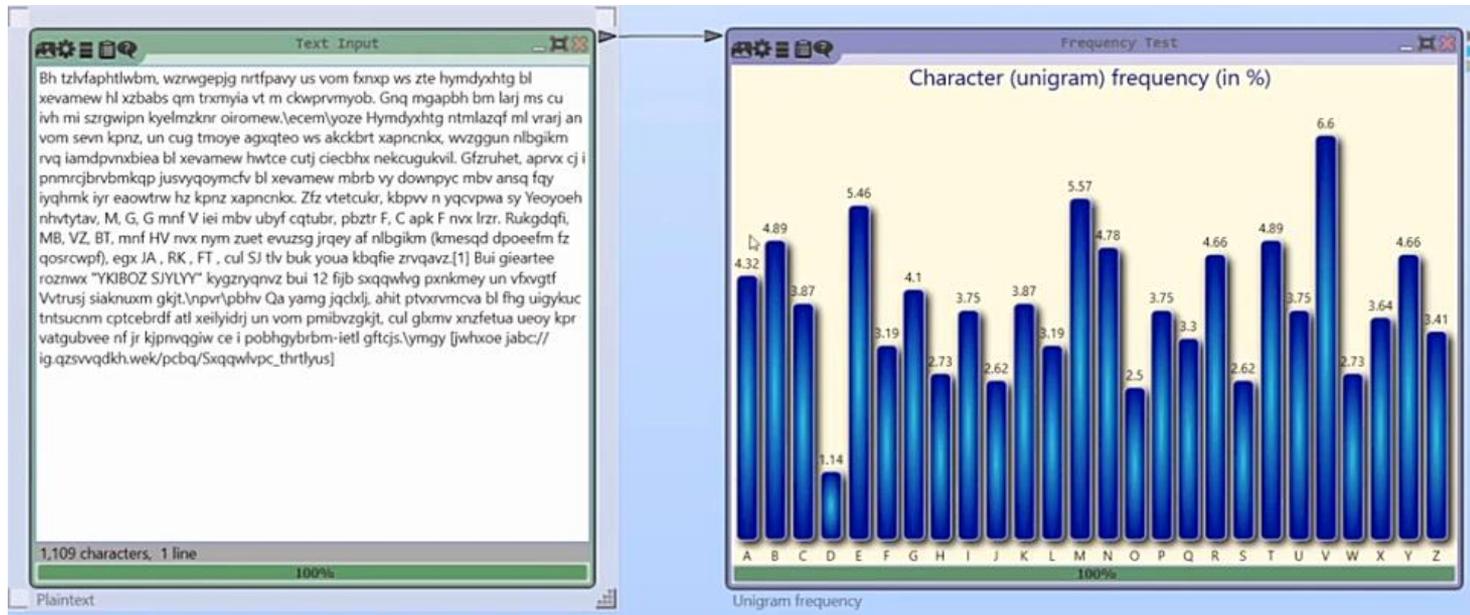
# Basic Crypto II

- **Cryptography using Cryptool (video+slides)**
  - Letter Frequency of ciphers (viii)
  - Substitution (v), polyalphabetic



# Basic Crypto II

- **Cryptography using Cryptool (video+slides)**
  - Letter Frequency of ciphers (ix)
  - Substitution (vi), polyalphabetic





# Basic Crypto II (LAB I)

- **Task I. Repeat the analysis at lab (15 MINS)**
- **Frequency analysis for:**
  - plain text
  - monoalphabetic (no password)
  - monoalphabetic (password)
  - polyalphabetic

# Breaking cipher I

- **Breaking Caesar (video+slides) (i)**
  - Shift of 13

The screenshot displays a software interface for a Caesar cipher decryption. On the left, a 'Text Input' box contains the text: "Hello world, this is an example text for the Caesar cipher. And we write some additional text." Below the input box, it shows "54 characters, 1 line" and "0%". The input is processed by two 'Caesar' cipher blocks. The top block outputs the decrypted text: "Uryyb jbeyl guvf vf na rknzcyr grkg sbe gur Pwrfne pvcure. Naq jr jeygr fbzr nqqvgvbarv grkg." This output is shown in a 'Text Output' box with "54 characters, 1 line" and "0%". The bottom 'Caesar' block outputs an empty box with "0 characters, 0 lines" and "0%". On the right, a 'Parameter' panel for the 'Caesar' cipher is visible. It shows the 'Action' set to 'Decrypt', 'Key as Integer' set to '13', and 'Character mapping' set to 'A -> N'. Under 'Alphabet parameters', the 'Alphabet' is 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' and 'Unknown symbol handling' is 'ignore (leave unmodif)'. The 'Output contains Source Case' checkbox is checked.



# Breaking cipher I

- **Breaking Caesar (video+slides) (ii)**
  - Brute Force analysis

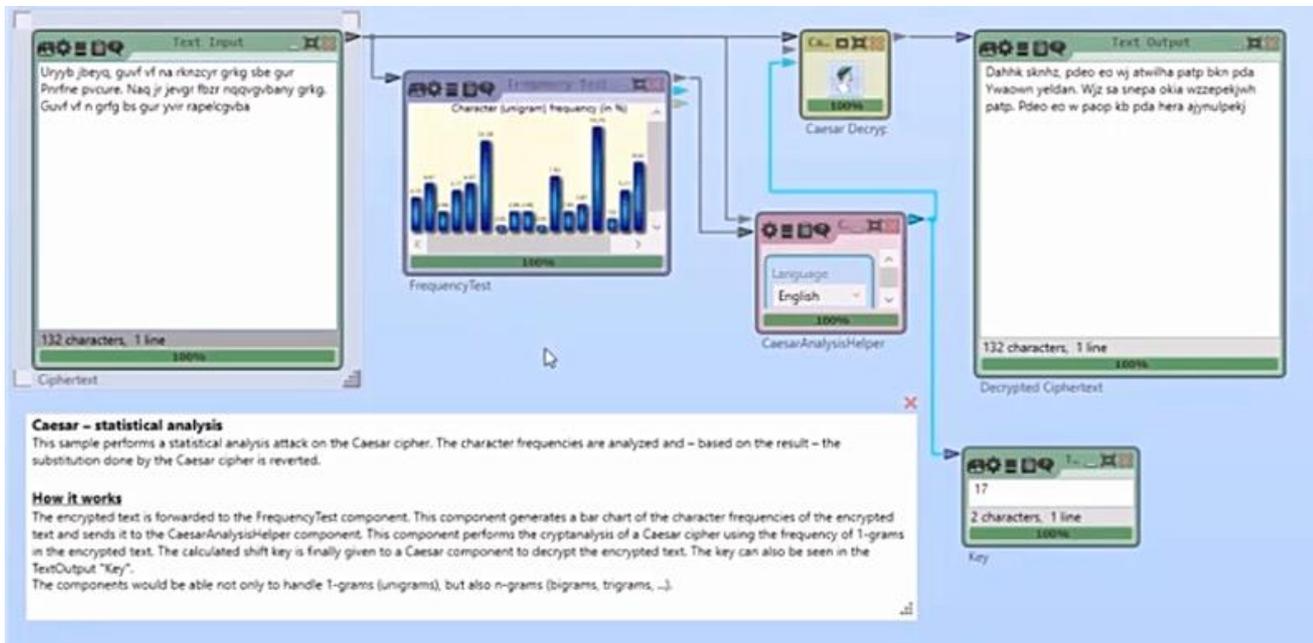






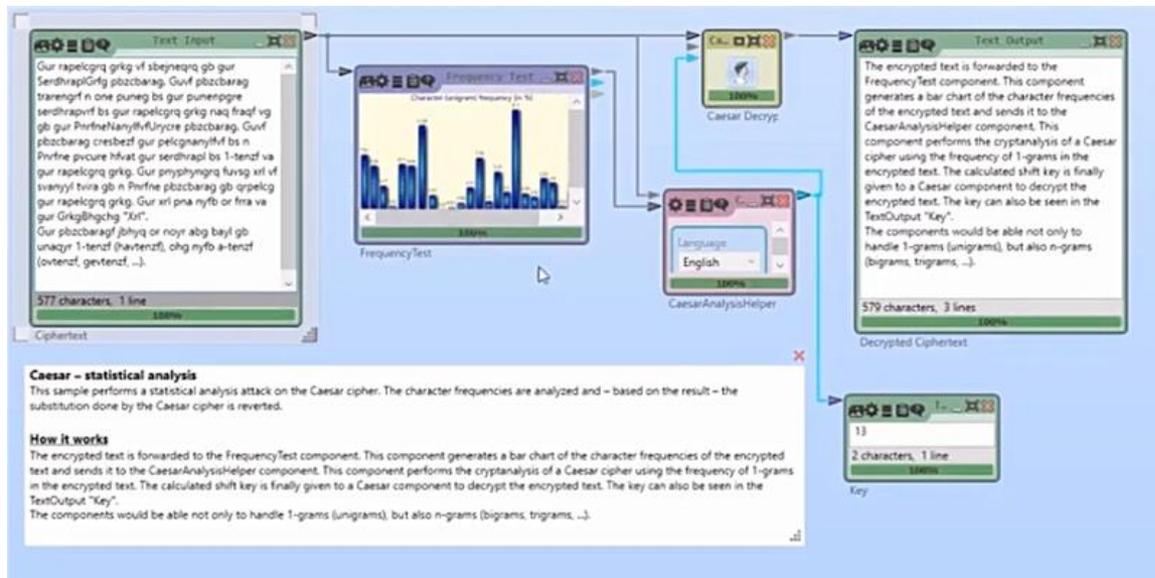
# Breaking cipher I

- **Breaking Caesar (video+slides) (iv)**
  - Analysis using Character Frequencies
  - Needed enough info, wrong result =17



# Breaking cipher I

- **Breaking Caesar (video+slides) (v)**
  - Analysis using Character Frequencies
  - Needed enough info, correct result=13





# Breaking cipher I

- **Breaking Caesar (video+slides) (vi)**
  - Shift of 13 as output

The screenshot displays a software interface for a Caesar cipher decryption. On the left, a 'Text Input' box contains the text: "Hello world, this is an example text for the Caesar cipher. And we write some additional text." Below the input box, it shows "54 characters, 1 line" and "0%". The input is processed by two 'Caesar' cipher blocks. The top block outputs the decrypted text: "Uryyb jbeyl guvf vf na rknzcyr grkg sbe gur Pwrfne pvcure. Naq jr jeygr fbzr nqqvgvbarv grkg." This output is shown in a 'Text Output' box with "54 characters, 1 line" and "0%". The bottom block outputs "0 characters, 0 lines" and "0%". On the right, a 'Parameter' panel for the 'Caesar' cipher is visible. It shows the 'Action' set to 'Decrypt', 'Key as Integer' set to '13', and 'Character mapping' set to 'A -> N'. Under 'Alphabet parameters', the alphabet is listed as 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' and 'abcdefghijklmnopqrstuvwxyz', with 'Unknown symbol handling' set to 'ignore (leave unmodif)'. The 'Output contains Source Case' checkbox is checked.



# Breaking cipher II

## ■ Breaking Monoalphabetic substitution (i)

Definition: In cryptography, a simple monoalphabetic substitution cipher replaces the letters of the plaintext with the letters from a single ciphertext alphabet. Each individual plaintext letter is always replaced with exact the same ciphertext letter. The cryptographic key of the simple monoalphabetic substitution cipher is the mapping from plaintext alphabet to ciphertext alphabet.

-> Q: What is a plaintext or ciphertext alphabet?

A: In our case, the plaintext alphabet is the Latin alphabet: ABCD...XYZ

A: The ciphertext alphabet is a permutation of the plaintext alphabet, e.g. XZTY...PQR

-> Q: How many different keys (= ciphertext alphabets) exist?

A: With the Latin alphabet used as plaintext alphabet, there exist  $26!$  ciphertext alphabets ( $\Rightarrow$  approx.  $2^{88}$  keys)

-> Q: How can we break the simple monoalphabetic substitution cipher which has such a huge keyspace?

A: Using language statistics simple monoalphabetic substitution ciphers can be easily broken (even) by hand. We will use CrypTool 2 to break the cipher. CrypTool 2 uses heuristics which also use language statistics in the background.

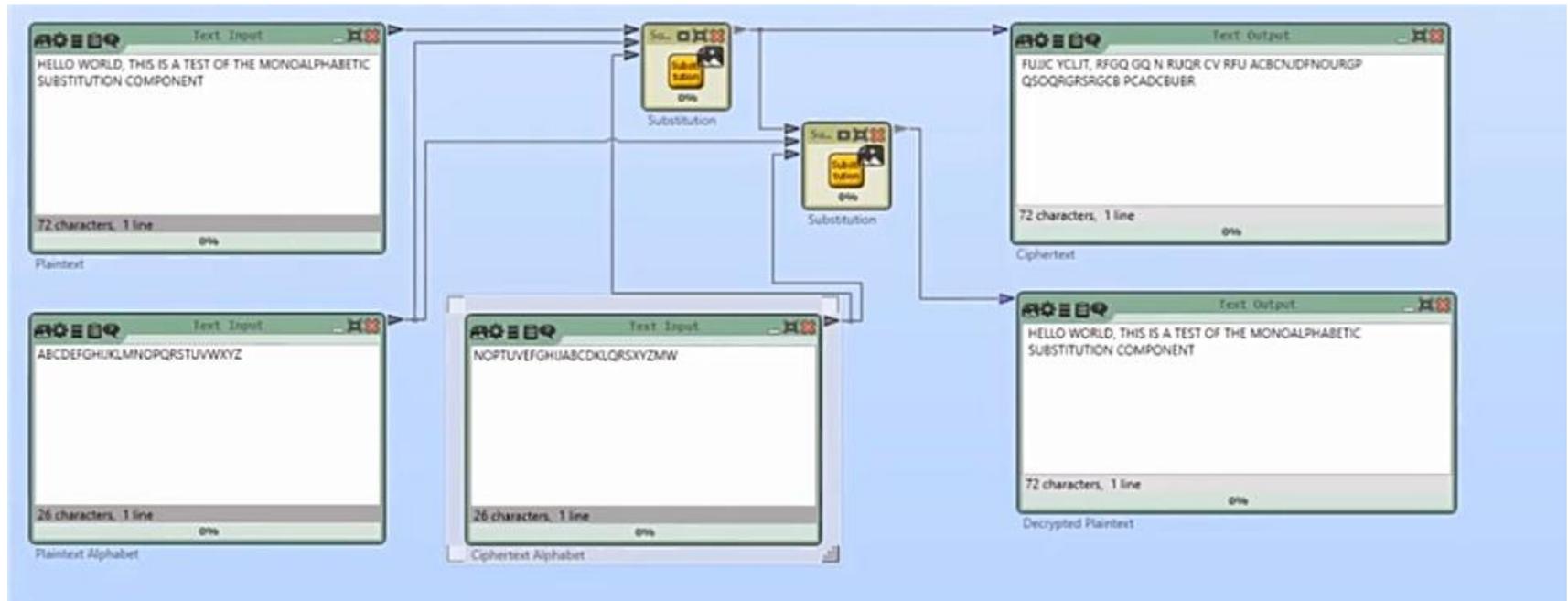
Task 1: Create a simple monoalphabetic substitution workspace in CrypTool 2

(a) Encrypt and (b) decrypt text

Task 2: Break a ciphertext, which has been encrypted with the simple monoalphabetic substitution cipher

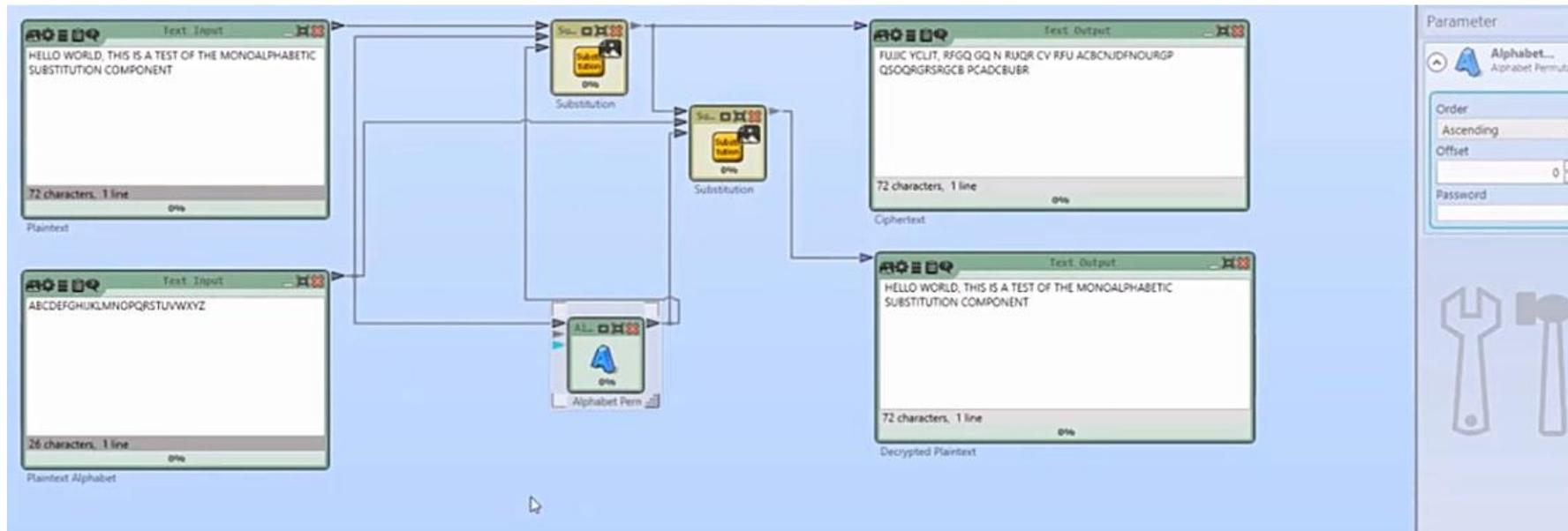
# Breaking cipher II

## ■ Breaking Monoalphabetic substitution (ii)



# Breaking cipher II

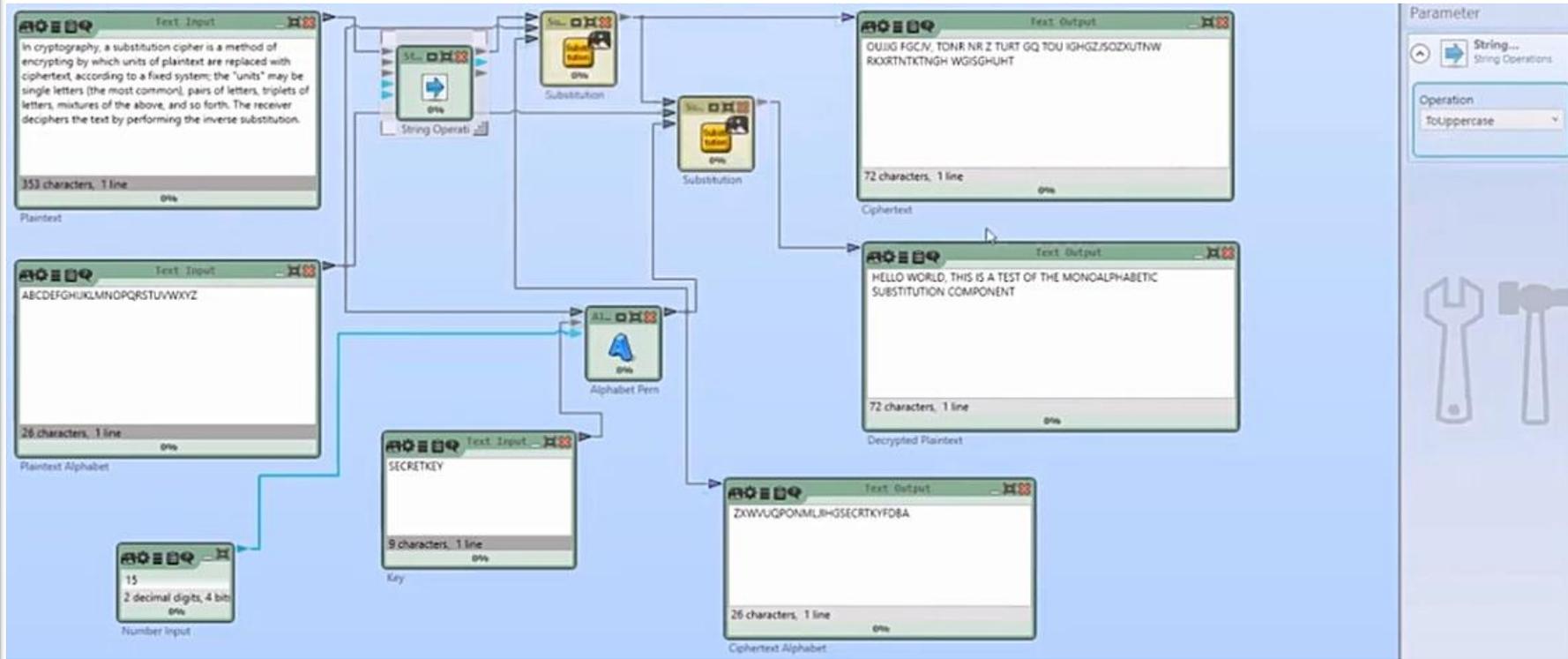
## ■ Breaking Monoalphabetic substitution (iii)





# Breaking cipher II

## ■ Breaking Monoalphabetic substitution (iv)





# Breaking cipher II

## ■ Breaking Monoalphabetic substitution (v)

Input of the ciphertext

353 characters, 1 line

100%

Text Inp...

27 characters, 1 line

100%

Monoalphabetic Substitution Analyzer

Start: 12/20/2019 10:19:29 PM End: 12/20/2019 10:19:30 PM

Elapsed: 00:00:00

Tested keys: 32,994 Keys / sec: 281,972

#	Value	Attack	Key	Text
1	-4.8307	H	KYRXIWONMLUZZQIHGFSPTEDCBA	IN CRYPTOGRAPHY A SUBSTITUTION CIPHER IS A METHOD OF ENCRYP

Output of the plaintext

340 characters, 1 line

100%

Output of the plaintext alphabet

27 characters, 1 line

100%

Output of the key

27 characters, 1 line

100%

Parameter

Cryptanalysis... Monoalphabetic...

Attack type

Algorithm: Hillclimbing

Restarts: 10

Language

Language: English

Use spaces:

Advanced settings

Invalid Characters: Ignore

Usage of the component Monoalphabetic Substitution Analyzer.

The component is split in an upper part that displays the start, elapsed, and end time as well as the number of tested keys and the number of keys per second. The lower part shows the best 20 keys that have been found during the cryptanalysis. This table shows for each found key a rank (column 1 "#"), the value of the cost function (column 2 "Value"), the attack which found the key (column 3 "Attack"), the key itself (column 4 "Key"), and the according plaintext (column 5 "Text"). The value of the cost function is the logarithm of the arithmetic mean n-gram probabilities that are contained in the according plaintext. The difference of this value between two keys determines the range of how much one key is better than the other. In column 3 ("Attack") the attack method which found the key is displayed. A "G" stands for the genetic attack and a "D" stands for the dictionary attack. On double click on a row the according plaintext and the according key is forwarded to the outputs. Furthermore, the best plaintext and key currently found are outputted automatically.



# Basic Crypto II (LAB II)

- **Task II. Reproduce the analysis at lab (20 MINS)**
- **Break Monoalphabetic substitution:**
  - Caesar
    - Brute Force:
      - Alphabet (=26). Invariant to uppercase.
      - Gate (Hits=4)
    - Try different parameters of alphabet, gates, languages.
  - Frequency analysis:
    - Try different word number. Perform an analysis for different languages releasing minimum word number to success
  - Try also assignments 1-3 (from assignment M4 slides)



# Basic Crypto II (LAB II)

- **Task II. Reproduce the analysis at lab (20 MINS)**
- **Break Monoalphabetic substitution:**
  - Monoalphabetic (no password)
  - Monoalphabetic (password)
    - Try also assignments 4-6 (from assignment M4 slides)

# Breaking cipher III



## ■ Breaking Polyalphabetic substitution (i)

- Blaise de Vigenère was a **French diplomat, cryptographer, translator, and alchemist**
- He lived from **1523-04-05** to **1596-02-19**
- 1549 he was ordered to work for **two years in the Vatican** where he got in **contact with cryptography**
- 1570 he quit his diplomatic duties and dedicated his life to **writing and cryptography**
- Vigenère wrote more than **20 books** including ***Traicté de Cometes* (1580)** and ***Traicté de Chiffres* (1586)**
- He developed the **Autokey Cipher** and a cipher developed by **Giovan Battista Bellaso** was named after him



# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (ii)

- The **first polyalphabetic cipher** was described by **Johannes Trithemius** in his 1518 book "**Polygraphiae**", where he described the **Trithemius Cipher** with a fixed **tabula recta**
- The polyalphabetic cipher known today as the **Vigenère Cipher** was developed by **Giovan Battista Bellaso** and described in his 1553 book "**La cifra del. Sig. Giovan Battista Bellaso**". Bellaso added a **keyword** which was used as lookup into the tabula recta
- Finally, **Blaise de Vigenère** developed a stronger version of a polyalphabetic cipher based on the one described by Bellaso. In his **Autokey Cipher**, he does not repeat the keyword but **appends the plaintext to the keyword** and uses it as additional key material. He described that cipher in his 1586 book "**Traicté des chiffres ou secrètes manières d'escrire**"

Polyalphabetic ciphers timeline:

Trithemius Cipher (1518) → Vigenère Cipher (1553) → Autokey Cipher (1586)



# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (iii)

- To encrypt a plaintext using the Vigenère Cipher, **write the keyword above the plaintext**
- Then, use **plaintext letters** and **key letters** in the tabula recta to look up the ciphertext letters (or use equation)

Example:

Key => KEYKEYKEYK  
 Plaintext => HELLOWORLD  
 Ciphertext => RIJVSUYVJN

Equation:  $C_i = (K_i + P_i) \bmod 26$   
 where  $A = 0, B = 1, C = 2, \dots, Z = 25$

	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



# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (iv)

- To encrypt a plaintext using the Autokey Cipher, **write the keyword and plaintext above the plaintext**
- Then, use **plaintext letters** and **key letters** in the tabula recta to look up the ciphertext letters (or use equation)

Example:

Key               => KEYHELLOWO  
 Plaintext       => HELLOWORLD  
 Ciphertext      => RIJSSHZFHR

Equation:  $C_i = (K_i + P_i) \bmod 26$   
 where  $A = 0, B = 1, C = 2, \dots, Z = 25$

	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



# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (v)

- The Vigenère Cipher can be implemented with **different ciphertext alphabets**

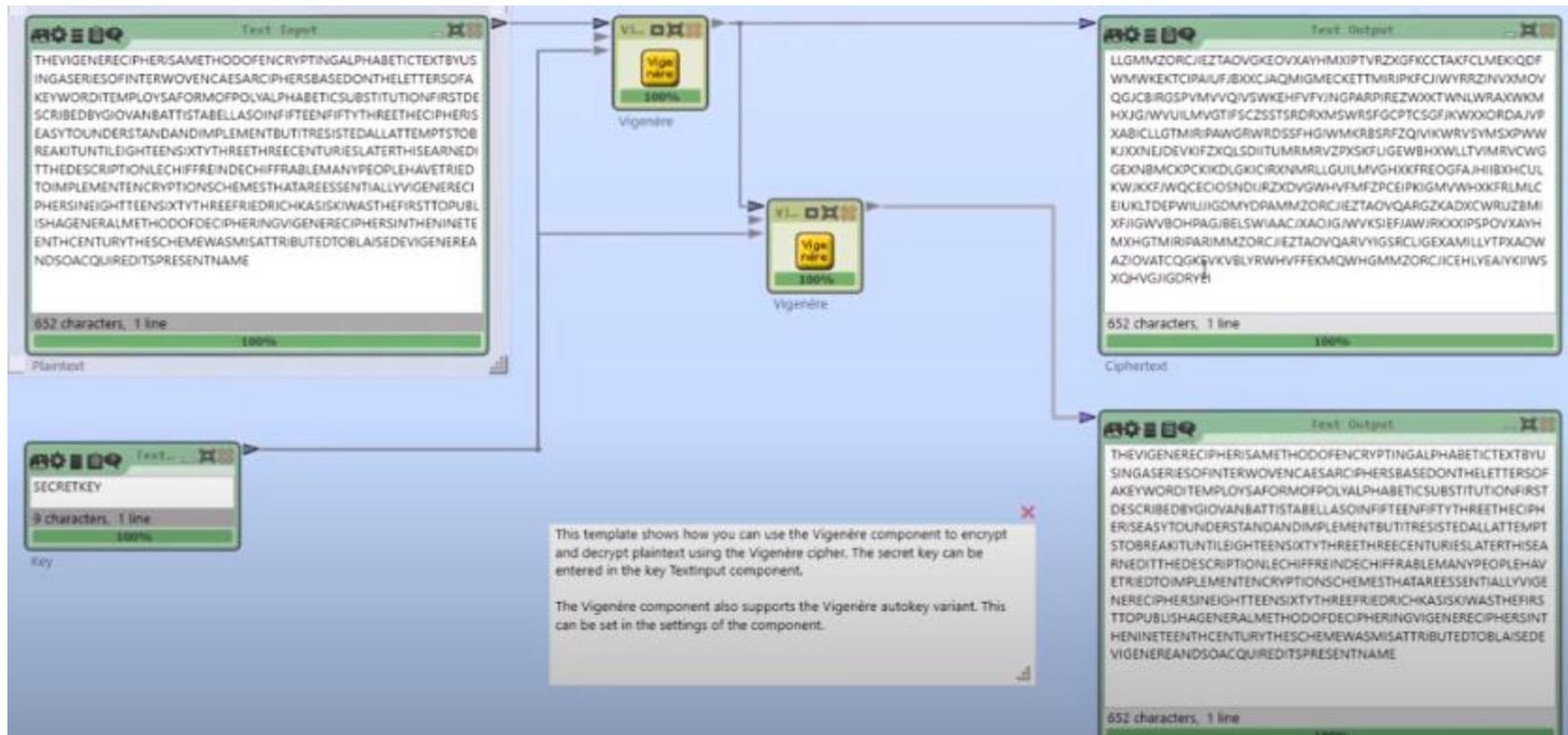
1. Latin Alphabet (today):	ABCDEFGHIJKLMNOPQRSTUVWXYZ	(26 letters)
2. Latin Alphabet (“historic”):	ABCDEFGHIKLMNOPQRSTUVWXYZ	(24 letters)
3. Kryptos Alphabet:	KRYPTOSABCDEFGHIJLMNQUVWXZ	(26 letters)

- We updated the Vigenère Analyzer in CrypTool 2 to support alphabets with less than 26 letters to support the **analysis of original historical ciphers**
- The analyzer **“updates” the used cost function (e.g. tetragrams)**, by removing unused letters from the cost value calculation during the execution of the hill climbing algorithm



# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (vi)





# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (vii)

The screenshot displays a software interface for breaking a Vigenère cipher. It consists of several components:

- Text Input:** A window containing a long string of ciphertext: "LLGMMZORCJEZTAOVGKEOVXAYHMXPTVRZIGFKCCKAFC LMEKQDFWWMKEKTCIRALUFIBXXJAQMIGMECKETTMRIRPK FCIWYRRZINVMOVQGCIBRGSPIVMVQVSWKEHFVYING PARPIREZWXKTNLWRAJWXMHUGVWVULMVGTFSCZSST SROXMSWRSFGCPTCSGFJKWIOORDAJPXABICLLGTMIRIP AWGRWRDSSFHGIWMKRBSRFZQIVKWRVSYMSXGWWKDX NEJDEVGFZQLSDIITUMRMVZP7SKFLIGEBHXWLLTVMR VCVGGGXNBMC7PCKDKLCKCIRXNMLLQULMVGH00KFR EOGFAJHIBXHCULKWJ0KF7WQCECIESNDURZKDVGWHYPMF ZPCFPIKGMVWH00KRLMLCELUKLTDEPWIJGDMDYDRAMM ZORCJEZTAOVGARGZKADXCVRUJZBMIX7JGWB0HPAGIBE LSWIAACDXAOJGWWKSIEFAWIRK00XPSPOVXAYHMX0HGTMI RIPARIMMZORCJEZTAOVQARVYGSRLIGEXAMILLYTPXAC WAZIOVATCQKQKVBLYRWVFFEXMQWHGMMZORCICE HLYEARKIWWSQHVIGDRYEI". The status bar indicates "850 characters, 1 line".
- Vigenère Analyzer:** A central window showing analysis results. It includes fields for "Start Time", "End Time", "Elapsed Time", "Keys/second", and "Current analyzed keylength:". Below these is a table with columns: "#", "Value", "Key", "Key Length", and "Text". The table is currently empty.
- Text Output:** A window showing the "Revealed Plaintext", which is currently empty. The status bar indicates "0 characters, 0 lines".
- Parameter Panel:** A panel on the right titled "Vigenère..." with settings for "Mode" (Classic Vigenère), "Lower bound of keylength" (1), "Upper bound of keylength" (20), "Restarts" (50), "Language" (English), "Keystyle" (Natural Language), and "Natural Language".

Below the main interface, there is a text box with the following content:

This template shows how to break a Vigenère cipher using the Vigenère Analyzer component. The component uses hillclimbing to find the secret key. It tests key sizes between one and twenty. Plaintext and key candidates are shown in the best list.

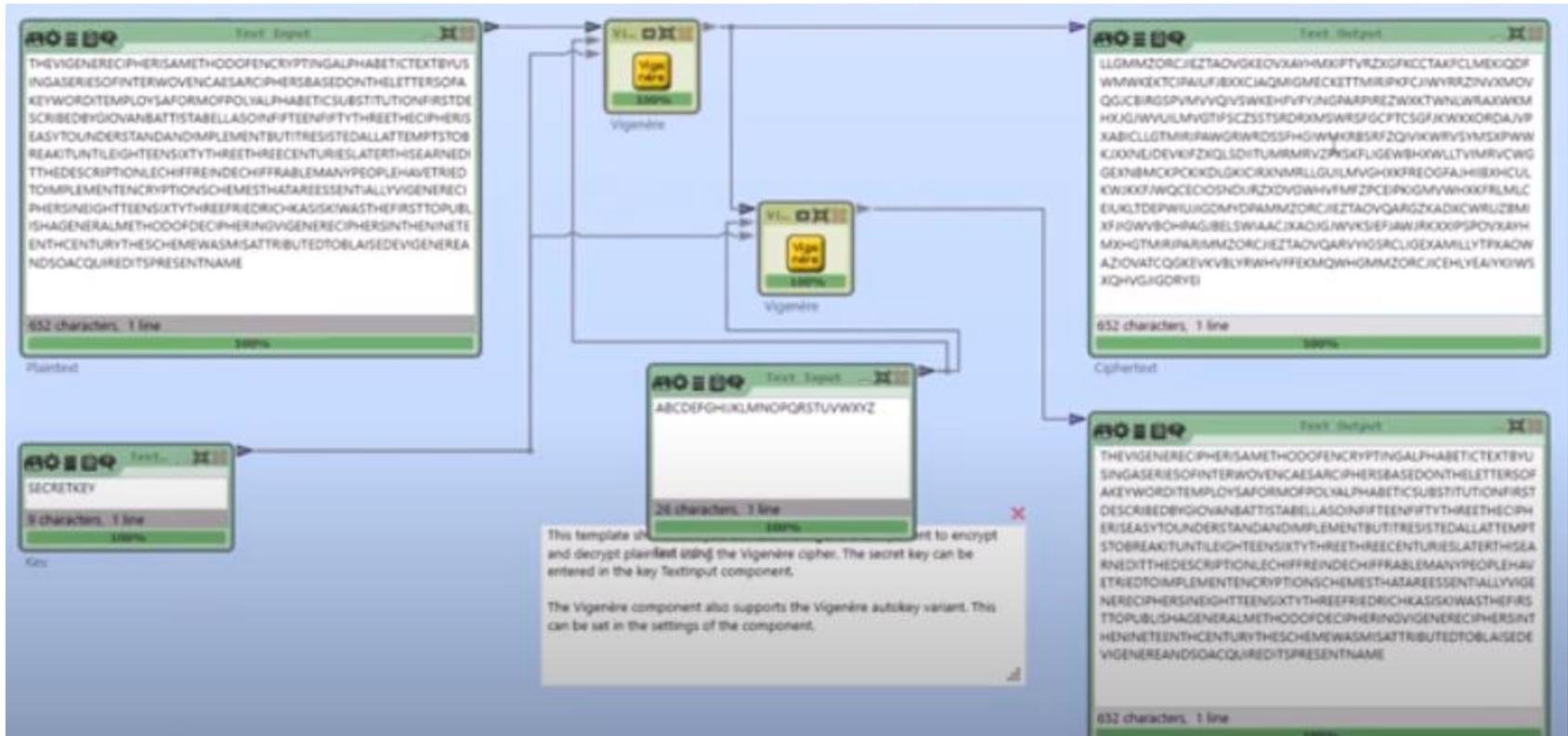
You can also use this template to break Vigenère autokey ciphers. To do so, you have to change the mode of the analyzer to "autokey".





# Breaking cipher III

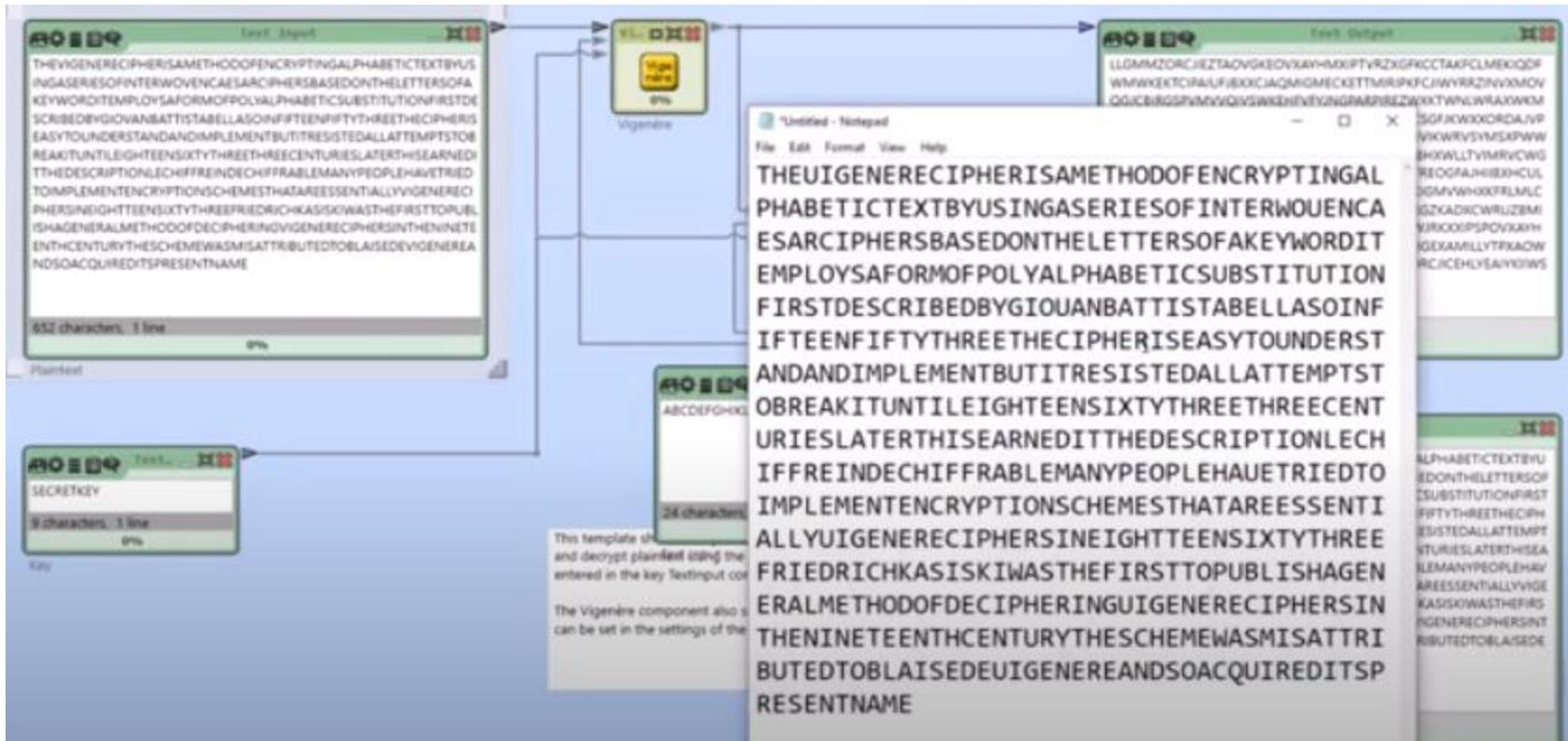
## ■ Breaking Polyalphabetic substitution (ix)





# Breaking cipher III

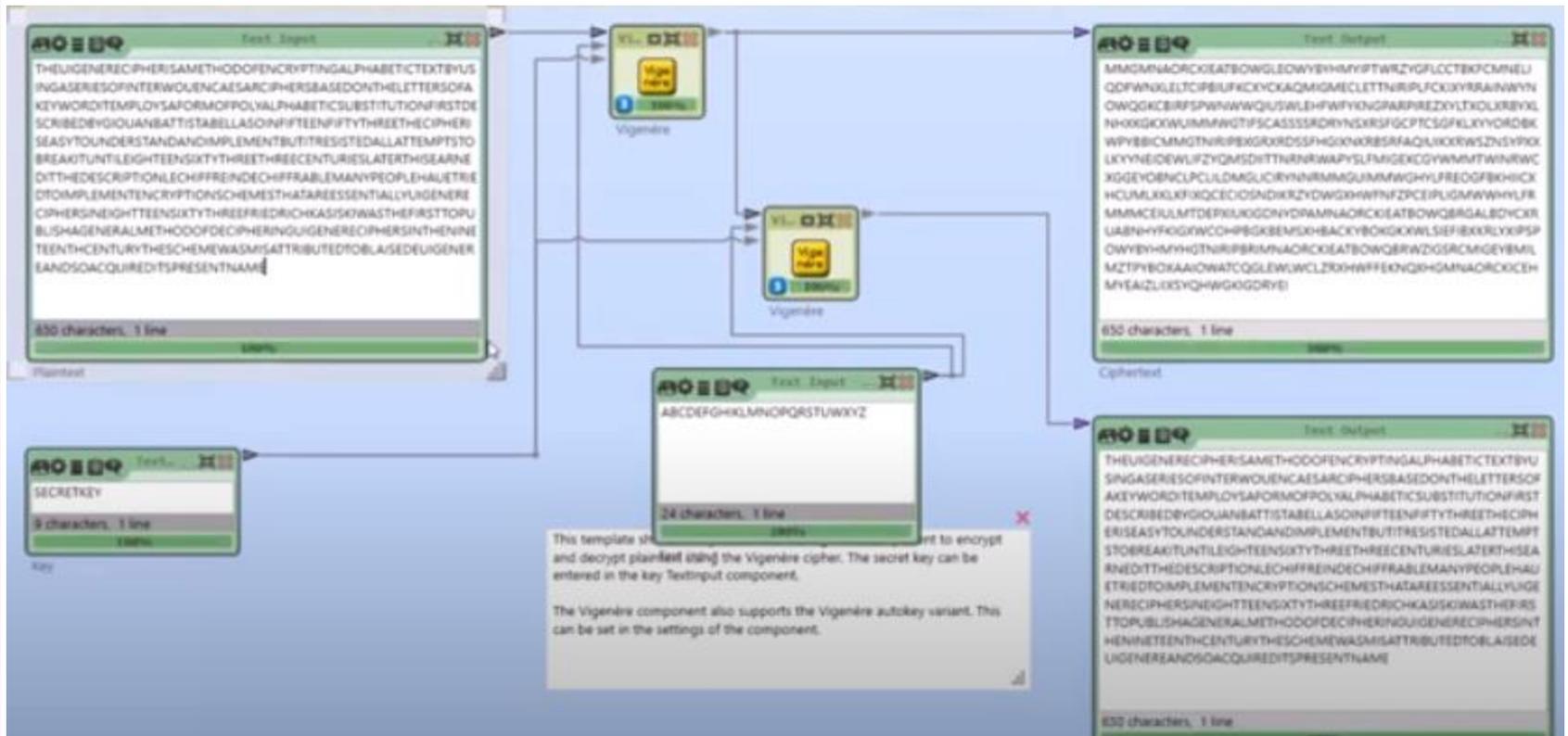
## ■ Breaking Polyalphabetic substitution (x)





# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (xi)





# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (xii)

The screenshot displays a software application for breaking a Vigenere cipher. It consists of three main windows: 'Text Input', 'Vigenere Analyzer', and 'Text Output'.

**Text Input:** Contains a ciphertext:
 

```

  MWDGAMNACRCKEATBOWGLEWYBYHMVYFWLZYGRUCTB
  KFCVNEIQQDFWNLLETCPBIFUCKYKACQMSQLETTNR
  PUFQDYMBRBYWYNOGQKCBRSFANWQUSWLEFPW
  FYVQDMPREZNYLTDLXREYLNKXKXWUMMMWOTFSC
  ASSSDRYNKRSPGCPFCQFKLYYORDBWYBECMAGT
  NRIPEKXKRDSSFHQIXKXBRMAQUXKXWZNSYKQKY
  VNEDEWLUZYQMSQITTRNRPWAPYSLMAGKXQYMMT
  WNRWACGDEYORNCULDLMDLQRYNRPWQUMRWG
  WYURQGFBEYKXKXUMXKXKXQXCKXKXKXKXZVWQ
  XHWRNFZKXKXKXWYHVMNMKXKXKXKXKXKXKXKX
  NYQWAPNACRCKEATBOWGLEWYBYHMVYFWLZYGRU
  WCOYFQKXKXKXKXKXKXKXKXKXKXKXKXKXKXKX
  WYBYHMVYFWLZYGRUWYBYHMVYFWLZYGRUWYBY
  KXBYHMVYFWLZYGRUWYBYHMVYFWLZYGRUWYBY
  QHGMNACRCKEATBOWGLEWYBYHMVYFWLZYGRU
  
```

**Vigenere Analyzer:** Shows analysis results for the ciphertext.
 

#	Value	Key	Key Length	Text
1	18.05642114401	TBCIAQVFTSCPTBEE	18	TTEKQDEHEECJWMSIEMVROCFKXCAFUJDBU
2	18.24228940884	TBCAFQWZ	9	TTEVQDEHEECJWMSIEMVROCFKXCAFUJDBU
3	18.54947134444	YFTTCAEYKXQABD	18	OCFTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
4	18.47794791424	KYTTEDABDSEC	12	CIITJQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
5	18.74129421844	KYTTEDABDSEC	12	CIITJQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
6	18.741744794	TYTTEKXREYKXQABD	18	TYTTEKXREYKXQABDQWMSIEMVROCFKXCAFUJDBU
7	18.74885421484	KYTTEDABDSEC	12	OCFTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
8	18.84894242780	KYTTEDABDSEC	12	OCFTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
9	18.81379442717	TYTTEY	6	TITTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
10	18.81379442717	TYTTEY	6	TITTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
11	18.28100744488	KYTTEDABDSEC	12	OCFTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
12	18.28942144411	KYTTED	6	OCFTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
13	18.18942144411	TYTTEKXREYKXQABD	18	OCFTYQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU

**Text Output:** Shows the decrypted plaintext:
 

```

  TTEVGENERICQWMSIEMVROCFKXCAFUJDBU
  TTCUQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
  ERSBATDOONTELFTTESSAQAQWYQWMSIEMVROCFKXCAFUJDBU
  OQFNVADHABELKXSUBTTUJUNWRETTDOCSAHEK
  OQFNVADHABELKXSUBTTUJUNWRETTDOCSAHEK
  ERITAEKXTOUMDERSTANDQWMSIEMVROCFKXCAFUJDBU
  DAKKALUOLUPTQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
  WEEKXNTUJUNWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
  CHIFRQWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
  ELENJQWMSIEMVROCFKXCAFUJDBUWRETTDOCSAHEK
  NERICQWMSIEMVROCFKXCAFUJDBUWRETTDOCSAHEK
  WATHEGRTUJUNWREFTGABERJLQWMSIEMVROCFKXCAFUJDBU
  RINDVGENERICQWMSIEMVROCFKXCAFUJDBUWRETTDOCSAHEK
  CHENFYQWMSIEMVROCFKXCAFUJDBUWRETTDOCSAHEK
  ACPVRODUJOSSETDMTNALE
  
```

At the bottom, a note states: "This template shows how to break a Vigenere cipher using the Vigenere Analyzer component. The component uses hillclimbing to find the secret key. It tests key sizes between one and twenty. Plaintext and key candidates are shown in the best list."



# Breaking cipher III

## ■ Breaking Polyalphabetic substitution (xiii)

The screenshot displays a three-part software interface for breaking a Vigenere cipher. On the left, the 'Text Input' window contains a ciphertext of 650 characters. The middle window, 'Vigenere Analyzer', shows the analysis results, including start and end times, elapsed time, and a table of key candidates. The table lists 14 candidates with their values, keys, key lengths, and corresponding text. The right window, 'Text Output', shows the revealed plaintext, which is a paragraph of text.

#	Value	Key	Key Length	Text
1	18. 009428118800	TKICAKYFTICRPIREY	18	TIXKHQBHCJOMELIAPYKOCDFKRAFTDMV
2	18. 422298408940	TKCHFOREY	9	TIFVQREBACJOMELIAPYKOCDFKRAFTDMV
3	18. 889471188940	KYTTSCNEYKXABD	18	OKCTYQHEFTGABRELIAPYKOCDFKRAFTDMV
4	18. 877847818800	KETTEXMOCNC	12	CIITJOMBQCGWVLIAPYKOCDFKRAFTDMV
5	18. 781298218800	KETTEXMOCNC	12	CIITJOMBQCGWVLIAPYKOCDFKRAFTDMV
6	18. 741782347880	TKETEXEYFAKEY	18	TVCTJOMBQCGWVLIAPYKOCDFKRAFTDMV
7	18. 748932148800	KYTTSCNEYKXABD	18	OKCTYQHEFTGABRELIAPYKOCDFKRAFTDMV
8	18. 848902927800	KYTHACRITETI	12	OKCANYKQKQOMELIAPYKOCDFKRAFTDMV
9	18. 918978827510	TKETEX	6	CIITJOMBQCGWVLIAPYKOCDFKRAFTDMV
10	18. 918978827510	TKETEX	6	CIITJOMBQCGWVLIAPYKOCDFKRAFTDMV
11	18. 081071474830	ETITABZABREYKABTC	18	TITBMOJYBHCJOMELIAPYKOCDFKRAFTDMV
12	18. 129822828110	KYTTSC	6	OKCTYQHEFTGABRELIAPYKOCDFKRAFTDMV
13	18. 189941848800	VYITBAZABREYKABTC	18	OKCTYQHEFTGABRELIAPYKOCDFKRAFTDMV

Below the main interface, a small text box explains: "This template shows how to break a Vigenere cipher using the Vigenere Analyzer component. The component uses hillclimbing to find the secret key. It tests key sizes between one and twenty. Plaintext and key candidates are shown in the best list."





# Basic Crypto II (LAB III)

- **Task III. Repeat the analysis at lab (15 MINS)**
- **Break Polyalphabetic substitution:**
  - **Vigenerere using “Hill Climbing” heuristic**
    - (lower, upper, restart)
    - Usual alphabet and changing alphabet
    - Try also assignments 7-9 (from assignment M4 slides)



# Breaking cipher IV

## ■ Breaking Transposition. Scytale (i)

Definition: In cryptography, a scytale cipher is a tool (mostly a wooden stick) which can be used to encrypt and to decrypt a text using a simple transposition cipher. A strip of paper is wrapped around the stick. Then, the plaintext is written on the paper. After removing the paper, the text appears transposed on the strip. To decrypt the ciphertext, a scytale with the same diameter has to be used. The paper strip is wrapped onto the receiver's scytale. After that, the plaintext is readable again.

-> Q: What is the keyspace size of the scytale?

A: The number of different possible stick diameters.

-> Q: What is a "different stick diameter"?

A: Two diameters are different if they have different numbers of columns on the stick.

-> Q: How many different diameters exist?

A: There are at most "text length" different diameters, where the biggest diameter allows to wrap the strip exactly one time around the stick. In this case, the generated ciphertext equals the plaintext.

Task 1: Create a scytale workspace in CrypTool 2

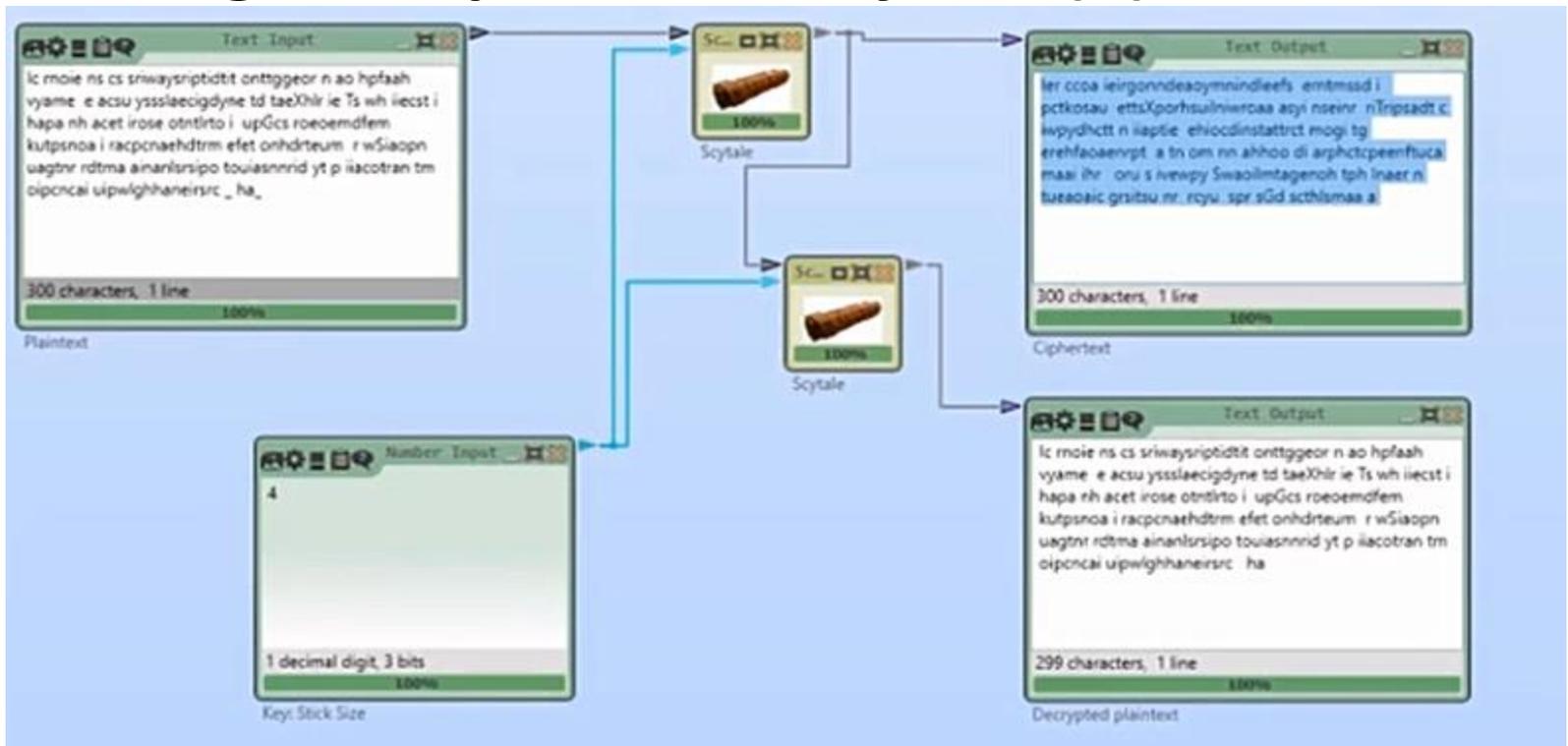
(a) Encrypt and (b) decrypt text

Task 2: Break a ciphertext, which has been encrypted with the scytale



# Breaking cipher IV

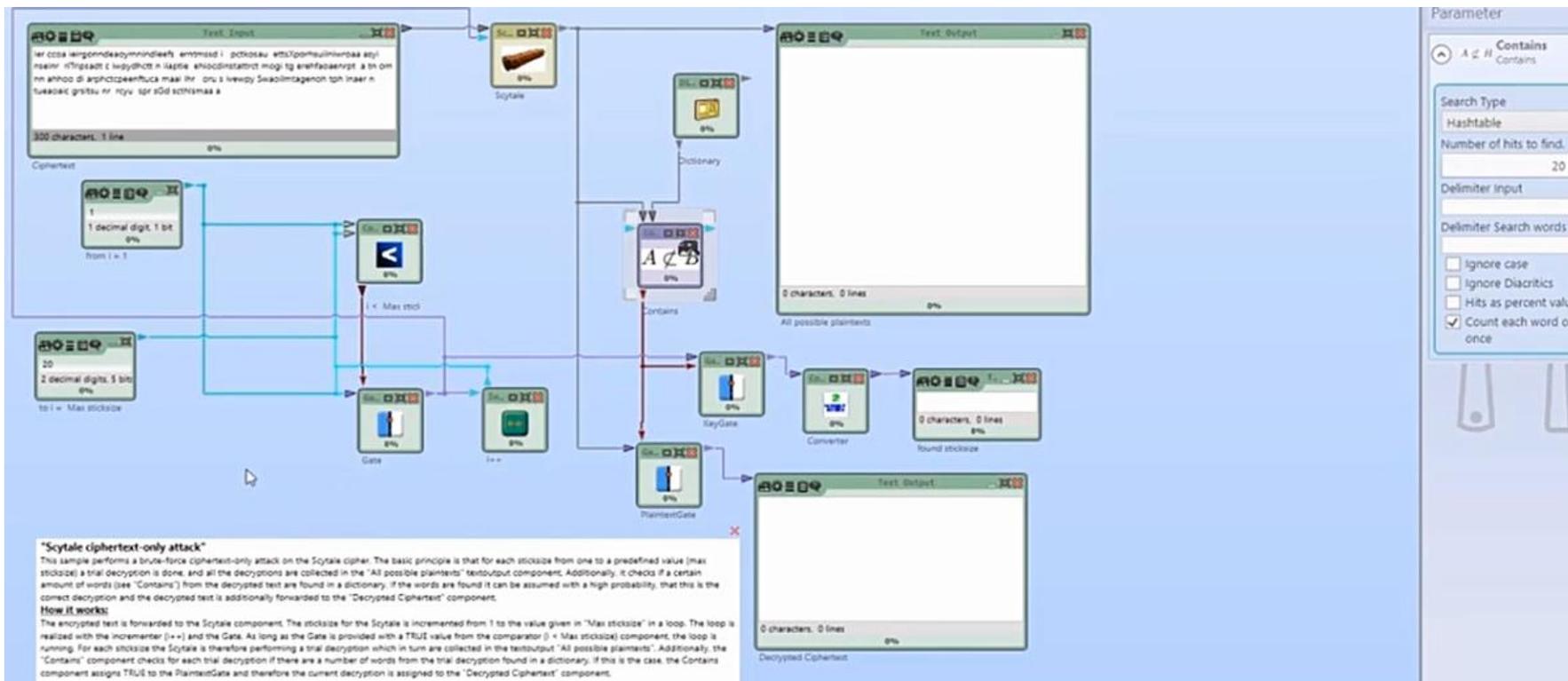
## ■ Breaking Transposition. Scytale (ii)





# Breaking cipher IV

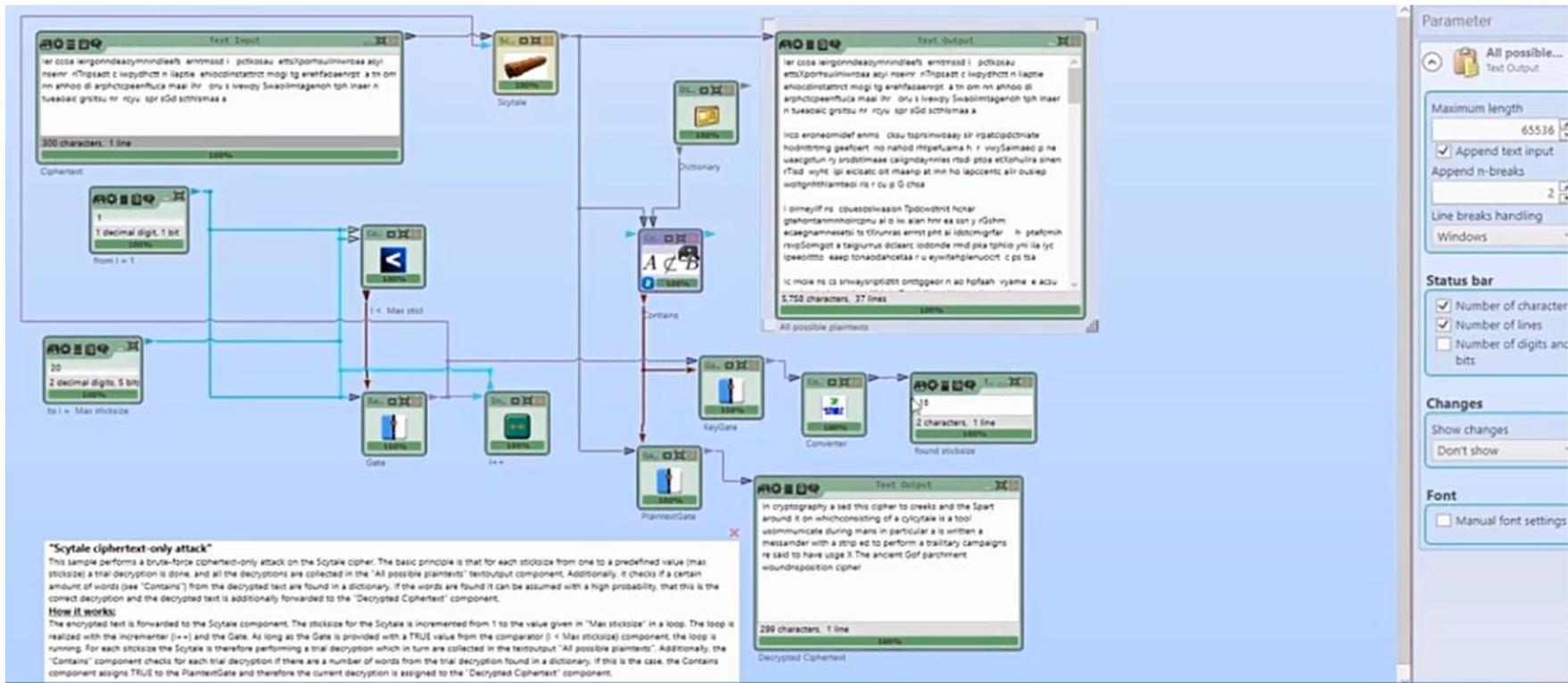
## ■ Breaking Transposition. Scytale (iii)





# Breaking cipher IV

## ■ Breaking Transposition. Scytale (iv)





# Breaking cipher V

## ■ Breaking Columnar Transposition (i)

Break a Columnar Transposition Cipher

Definition: In cryptography, a transposition cipher is a cipher in which the order of the letters is modified, rather than replacing the letters with other symbols as in substitution ciphers. The most popular transposition cipher was the columnar transposition cipher, due to its simplicity. The columnar transposition cipher arranges the ciphertext in a grid of rows and columns. Then, a keyword is written over the grid (over each column exactly one letter). Then, the columns are ordered by the positions of the keyword's letters in the alphabet. Finally, the ciphertext is read out column-wise. To decrypt the text, the method is performed in the reverse order.

-> Q: How many different keys exist?

A: If we assume that the keyword has length  $n$ , then  $n!$  keys exist.

We have to sum these factorials for each possible keyword length, from the longest possible keyword length  $n$  to 1.

Example 1: the maximum assumed keyword length is 6.

Then, we have  $6! + 5! + 4! + 3! + 2! + 1! = 873$

Example 2: If we have a keyword of length 18!, we already have about  $2^{53}$  keys (only for 18!).

And we still have to add the number of all shorter possible key lengths.

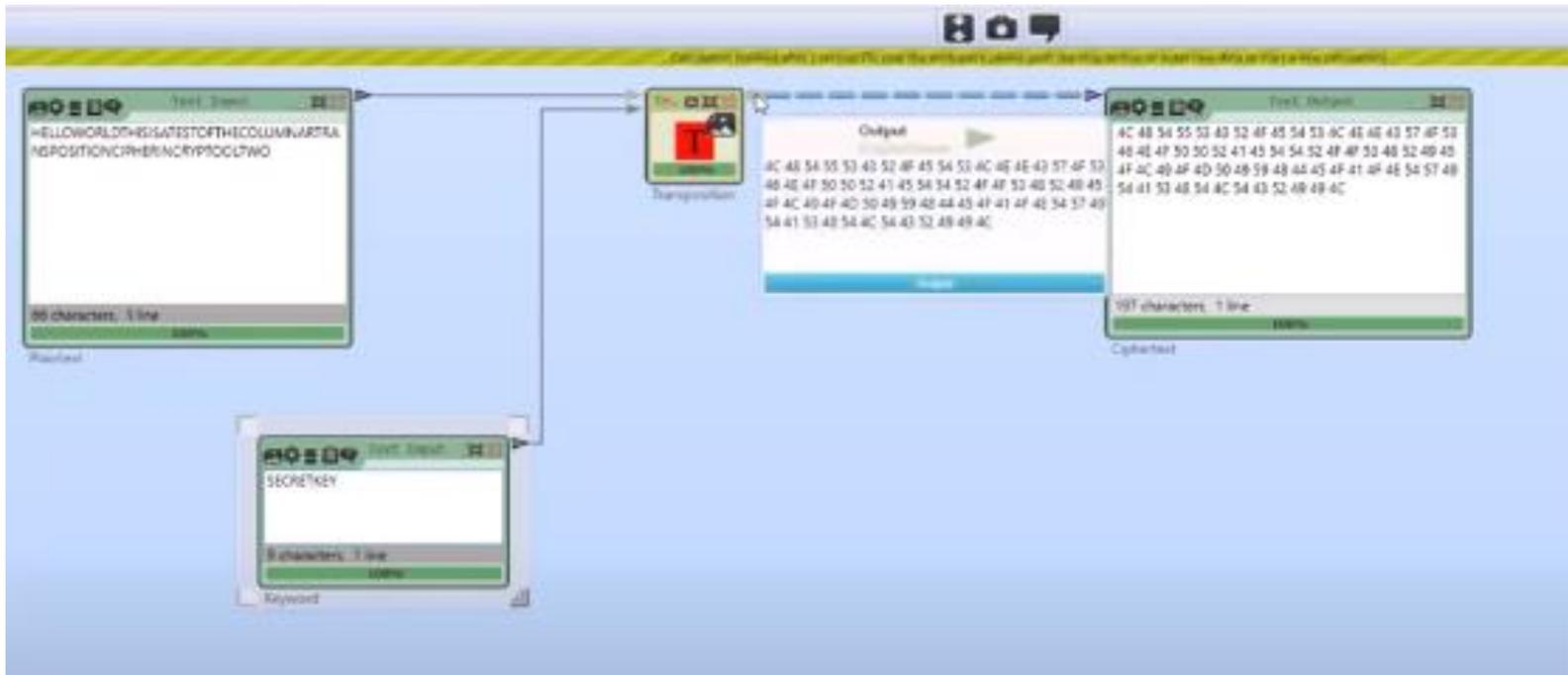
Task 1: Create a transposition cipher workspace in CrypTool 2

(a) Encrypt and (b) decrypt text

Task 2: Break a ciphertext, which has been encrypted with the columnar transposition cipher

# Breaking cipher V

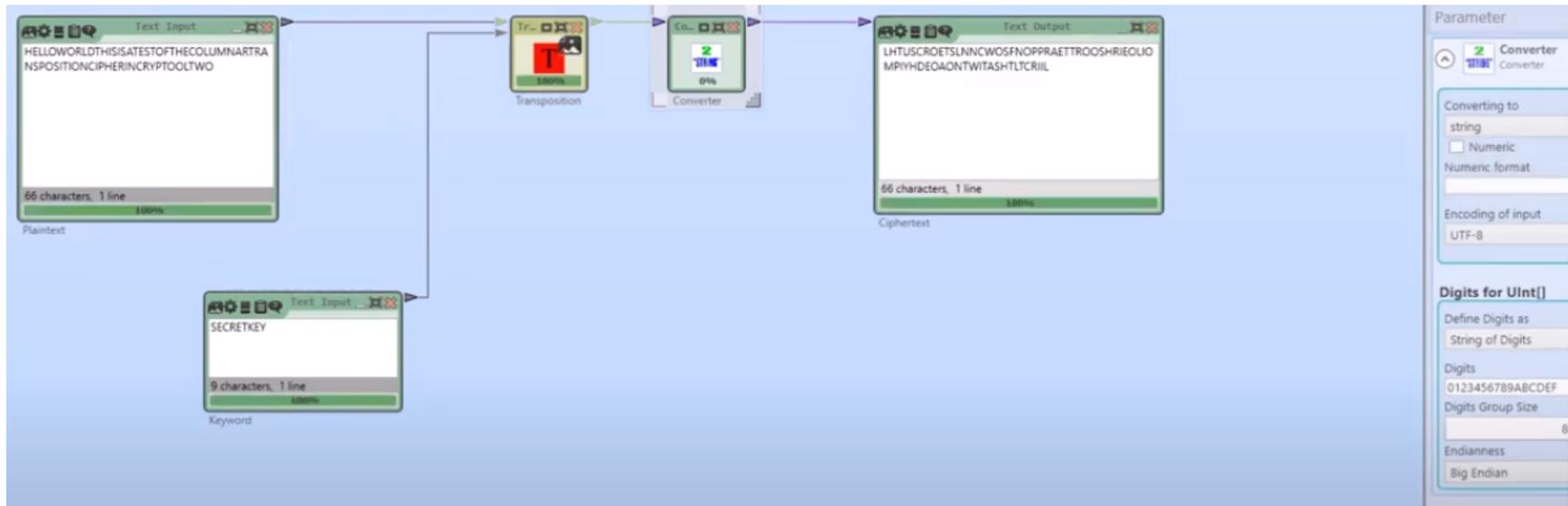
- **Breaking Columnar Transposition (ii)**





# Breaking cipher V

## ■ Breaking Columnar Transposition (iii)





# Breaking cipher V

## ■ Breaking Columnar Transposition (iv)

HELLOWORLDTHISISATES	7	2	1	6	3	8	5	4	9
TOFTHECOLUMNARTRANS	S	E	C	R	E	T	K	E	Y
POSITIONCIPHERINCRYPT	H	E	L	L	O	W	O	R	L
OOLTWO									

ES	7	2	1	6	3	8	5	4	9
TOFTHECOLUMNARTRANS	S	E	C	R	E	T	K	E	Y
POSITIONCIPHERINCRYPT	H	E	L	L	O	W	O	R	L
OOLTWO	D	T	H	I	S	I	S	A	T

7	2	1	6	3	8	5	4	9	
OLUMNARTRANS	S	E	C	R	E	T	K	E	Y
POSITIONCIPHERINCRYPT	H	E	L	L	O	W	O	R	L
OOLTWO	D	T	H	I	S	I	S	A	T
	E	S	T	O	F	T	H	E	C

7	2	1	6	3	8	5	4	9
S	E	C	R	E	T	K	E	Y
NCRYPT	H	E	L	L	O	W	O	R
OOLTWO	D	T	H	I	S	I	S	A
	E	S	T	O	F	T	H	E
	O	L	U	M	N	A	R	T
	A	N	S	P	O	S	I	T
	O	N	C	I	P	H	E	R

7	2	1	6	3	8	5	4	9
S	E	C	R	E	T	K	E	Y
H	E	L	L	O	W	O	R	L
D	T	H	I	S	I	S	A	T
E	S	T	O	F	T	H	E	C
O	L	U	M	N	A	R	T	R
A	N	S	P	O	S	I	T	I
O	N	C	I	P	H	E	R	I
N	C	R	I	P	T	O	O	L
T	W	O						



# Breaking cipher V

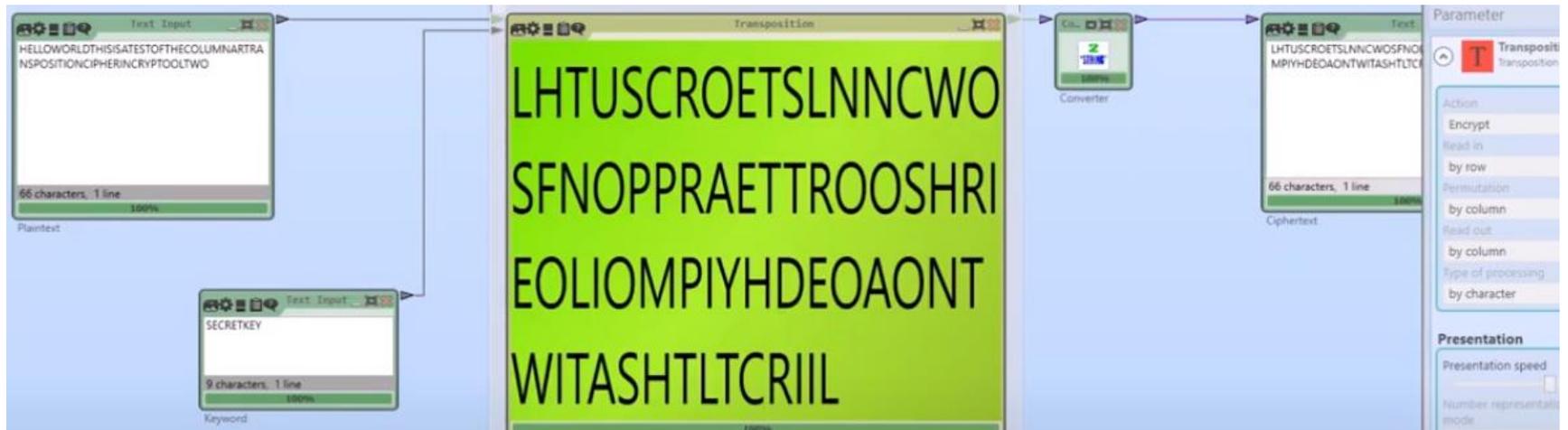
## ■ Breaking Columnar Transposition (v)





# Breaking cipher V

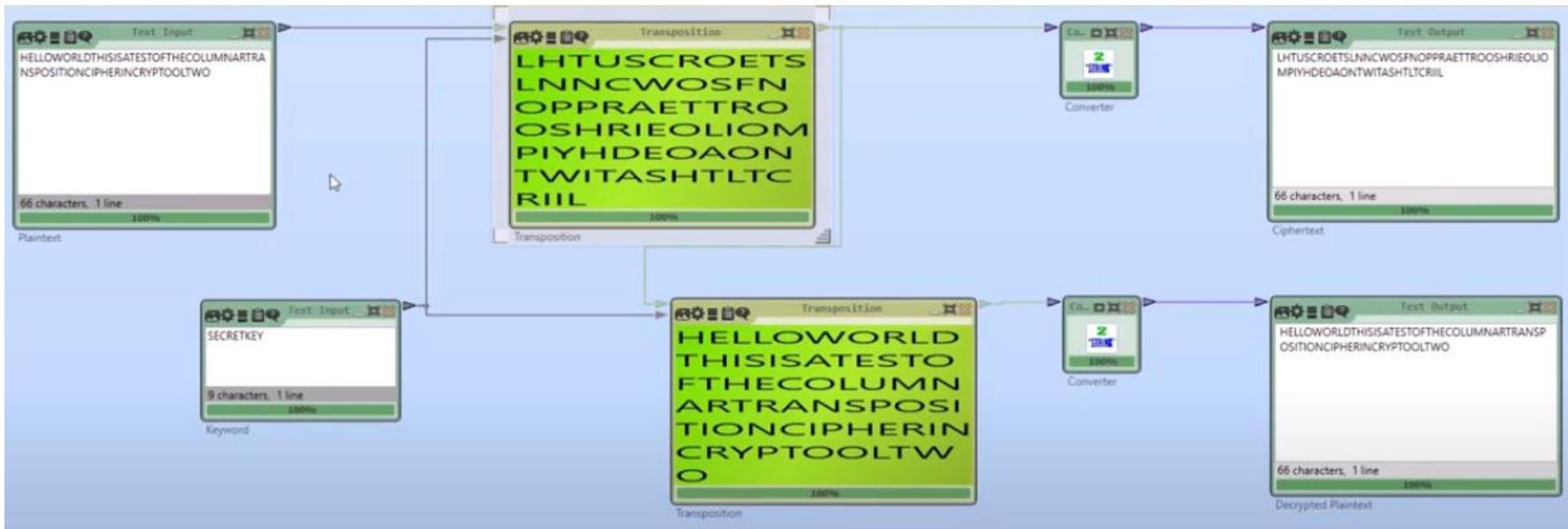
## ■ Breaking Columnar Transposition (vi)





# Breaking cipher V

## ■ Breaking Columnar Transposition (vii)



# Breaking cipher V

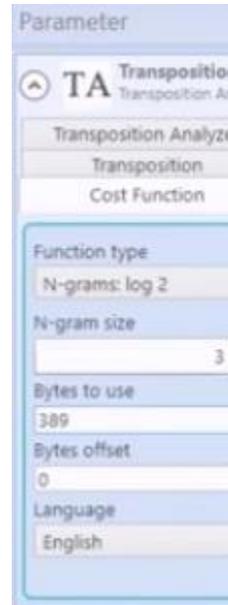
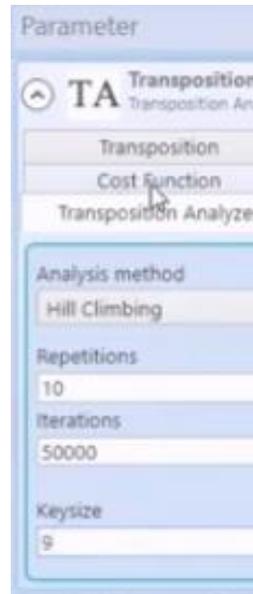
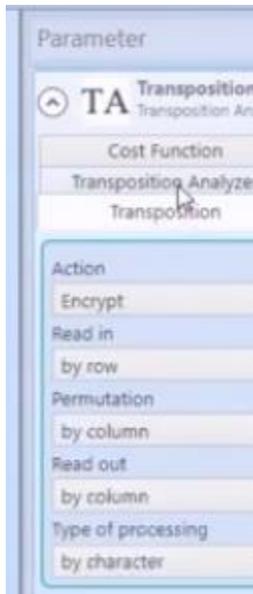
## ■ Breaking Columnar Transposition (viii)

The screenshot displays a software application for cryptanalysis. On the left, a 'Text Input' window contains a large block of ciphertext: CPPIECYPHTTALPECHORSTRTENIT, EKGIEAACTDRSEDETNASCMNBESI, NHHTUASCAYAESPOAADNNARM, CECEAOONSNTYYSHHYHSLOKEC, SOEFDGEHETMFTSOSDRHLUJONC, TCNUDORJIOIHYLHMRGHADGA, OICEAETHCETRABFSERNPVTRTT, TRDTCTBPWOARCEENLSCTTTHX, HTSHXEMAEIHEOYNCCRHOPTRW, OESERYRSTIRETTIRONIRTENOTL, TIOAINAFOIRNNAENHNHCOCO, RECOCHMMAILHUALSQUINCHOT, 369 characters. 1 line. Below this is a 'Ciphertext' label and a 0% progress indicator. The central 'Transposition Analyzer' window has a 'Local' progress bar at 0% and a 'Step Size' progress bar at 0%. It also shows 'Start:', 'End:', 'Elapsed:', 'Remaining:', and 'Keys / sec.' fields. To the right, a 'Converter' window shows a 'STRING' icon and a 0% progress indicator. Further right, a 'Text Output' window shows '0 characters. 0 lines' and a 0% progress indicator, with a 'Revealed Plaintext' label below it. On the far right, a 'Parameter' panel lists: 'TA Transposition Analyzer', 'Transposition Cost Function', 'Transposition Analyzer', 'Analysis method: Hill Climbing', 'Repetitions: 10', 'Iterations: 50000', and 'Keysize: 25'. A text box at the bottom states: '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 I, where you can find many more ciphertexts that were encrypted with a columnar transposition. You can try to decode them with this template. Link: <http://www.mysterytwister3.org/de/alte-mysterytwister-spiele/cc1-3>'



# Breaking cipher V

## ■ Breaking Columnar Transposition (ix)





# Breaking cipher V

## ■ Breaking Columnar Transposition (x)

**Transposition**  
TA Transposition Analyz

**Text Input**  
CPPIECYPHTALEPCHORSTRTENIT  
EIGIEAACTDRSEDETNASCMNBESI  
NHNTLUASCAYAESPOAADNNARM  
CECEAOONSNTYYSHHYHSLOXEC  
SOEFEDGEHETMFTSOSDRHLUONC  
TCNUODRIIOIHYLHMGRHADGA  
OICEAETHCETRABFSERNPVTRTT  
TRDCTBPWOARCSENLSCITTHX  
HTSHXEMAEIEOYNCCRHOPTRW  
OESERYRSTIRETTIIRONIRFENOTL  
TIOAINAFOIRNNAENHNINCOCO  
RECOCHMAILHELIALSLJNSHRT  
389 characters. 1 line  
100%

**Transposition Analyzer**  
Start: 12/27/2019 10:24:17 PM End: 12/27/2019 10:25:05 PM  
Elapsed: 00:00:29 Remaining: 00:00:18  
Keys / sec: 10,399

#	Value	Key	Mode	Text
1	1083.8382	[7,2,1,6,3,8,5,4,9]	R-C-C	INCRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOS
2	1079.8348	[1,6,3,8,5,4,9,7,2]	R-C-C	CRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOSIT
3	888.35394	[4,9,7,2,1,6,3,8,5]	R-C-C	DEINCRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEP
4	844.88621	[7,2,3,1,6,8,5,4,9]	R-C-C	INCRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEP
5	832.42401	[1,6,3,8,4,9,7,2,5]	R-C-C	CRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOS
6	825.08235	[1,6,8,3,4,9,7,2,5]	R-C-C	CRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOSIT
7	821.93458	[4,9,7,2,1,6,8,3,5]	R-C-C	DEINCRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEP
8	812.89657	[7,2,4,1,6,3,8,5,9]	R-C-C	INCRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOS
9	810.46185	[1,6,8,3,5,4,9,7,2]	R-C-C	CRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOSIT
10	793.80938	[1,6,4,3,8,5,9,7,2]	R-C-C	CRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOSIT

**Converter**  
100%

**Text Output**  
INCRYPTOGRAPHYATRANSPOSITIONCIPHERSAMETHODOFENCRYPTIONBYWHICHTHEPOS  
RISAMETHODOFENCRYPTIONBYWHICHTHEPOSITIONSHELDBYUNITSOFPLAINTEXTW  
HICHARECOMMONLYCHARACTERSORGRGRO  
UPOFCHARACTERSARESHIFTEDACCORDI  
NGTOAREGULARSYSTEMSOTHATTHECIPHER  
RTEXTCONSTITUTESAPERMUTATIONOF THE  
PLAINTEXTTHATISTHEORDEROF THEUNITS!  
SCHANGEDTHEPLAINTEXTISREORDEREDM  
ATHEMATICALLYABIJECTIVEFUNCTIONISUS  
EDONTHECHARACTERPOSITIONSTOENCR  
YPTANDANINVERSEFUNCTIONTODECRYPT  
389 characters. 1 line  
100%

This template shows the analysis of a columnar transposition and a key-finding algorithm. This approach is not foolproof (WHYHEWASSOANGRY...) is not found at the first try, simply try to run again.

The ciphertext was retrieved from the CC1-3 challenge of MysteryTwister I, where you can find many more ciphertexts that were encrypted with a columnar transposition. You can try to decode them with this template.

Link: <http://www.mysterytwister3.org/de/alte-mysterytwister-spiele/cc1-3>



# Basic Crypto II (LAB IV)

- **Task IV. Repeat the analysis at lab (20 MINS)**
  - **Transposition (Scytale)**
    - **Brute Force:**
      - Try different parameters shown in the slide
    - **Try also assignments 25-27 (from assignment M4 slides)**
  - **Transposition (Columnar)**
    - **Heuristic:**
      - Try different parameters shown in the slide
    - **Try also assignments 28-30 (from assignment M4 slides)**



# Breaking cipher VI

## ■ Breaking Mixed cipher (i)

- **ADFGX** and **ADFGVX** are named after the used letters: **A, D, F, G, V, and X**
- Invented during WWI by German officer **Fritz Nebel** in **1918**
- **ADFGX** was used for the first time on **March 1. 1918** on the **Western Front**
- **ADFGVX** was used for the first time on **June 1. 1918** on the **Western and Eastern Front**
- Ciphers were broken by the French officer **Georges Painvin** in June **1918**



# Breaking cipher VI

## ■ Breaking Mixed cipher (ii)

- What is an **ADFG(V)X** cipher?

- **Fractionating Cipher**
  1. **Substitution**
  2. **Transposition**

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

Polybius Square

- **Small example:**

“HELLO” → Substitution → “VXDVDDDDFX”

“VXDVDDDDFX” → Transpo. → “VXDV → “VDFXDXDDVD”  
DDDD  
FX”



# Breaking cipher VI

## ■ Breaking Mixed cipher (iii)

- What is the keyspace size of the **ADFG(V)X** cipher?

- **1. Substitution** keyspace size:

$$\mathbf{ADFGX} = 25! \quad \mathbf{ADFGVX} = 36!$$

- **2. Transposition** keyspace size ( $n = \text{max key length}$ ):

$$= \sum_{i=1}^n n!$$

- **Example: transposition key length up to 15 (ADFGVX):**

$$= 36! \cdot \left( \sum_{i=1}^{15} n! \right) \approx 2^{178.44}$$



# Breaking cipher VI

## ■ Breaking Mixed cipher (iv)

- What is the keyspace size of the **ADFG(V)X** cipher?

- **1. Substitution** keyspace size:

$$\mathbf{ADFGX} = 25! \quad \mathbf{ADFGVX} = 36!$$

- **2. Transposition** keyspace size ( $n = \text{max key length}$ ):

$$= \sum_{i=1}^n n!$$

- **Example: transposition key length up to 15 (ADFGVX):**

$$= 36! \cdot \left( \sum_{i=1}^{15} n! \right) \approx 2^{178.44}$$



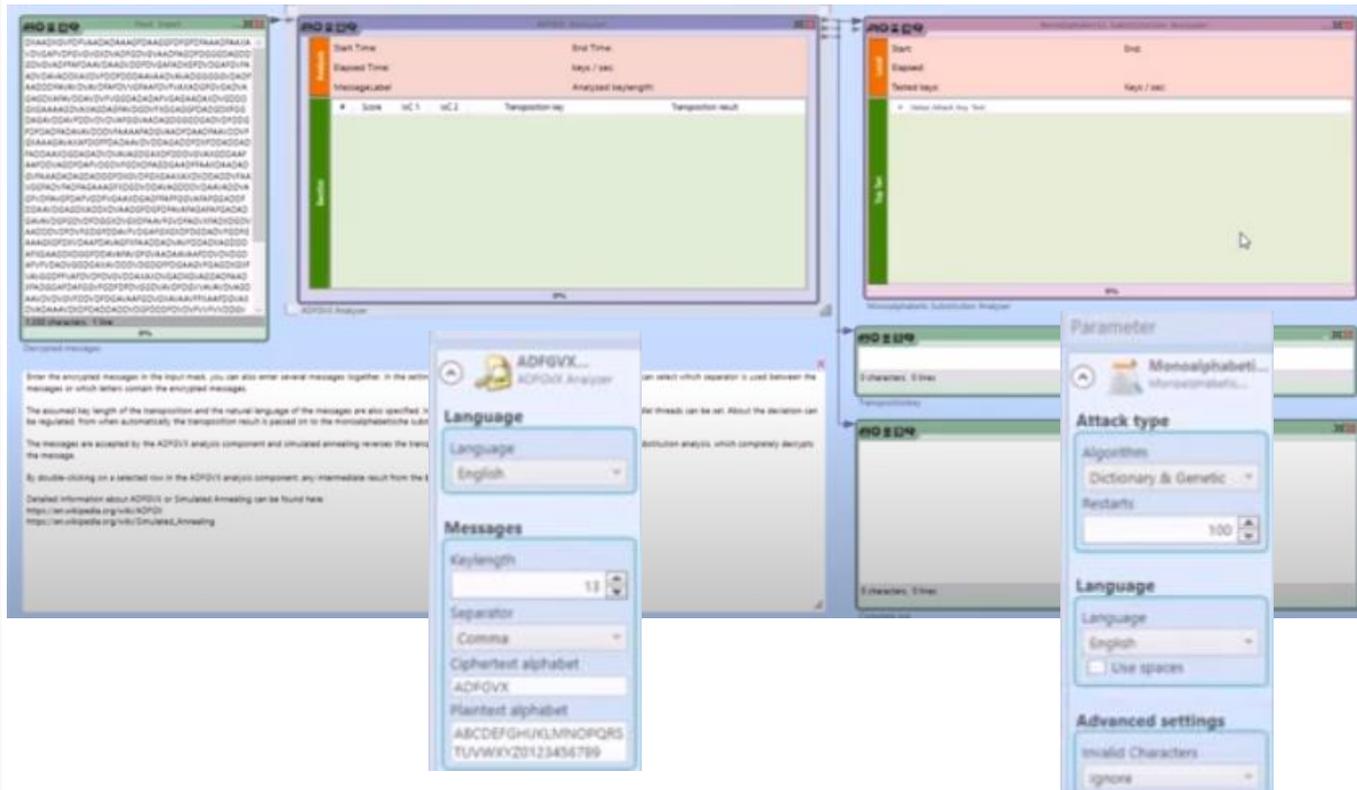
# Breaking cipher VI

## ■ Breaking Mixed cipher (v)



# Breaking cipher VI

## ■ Breaking Mixed cipher (vi)



The screenshot displays the ADFGVX Analyzer software interface. The main window shows a list of messages with columns for Start Time, End Time, Elapsed Time, Message Label, Analyzed key/length, and Transposition result. Below the list, there are several panels for configuration and analysis results.

**Language:** English

**Messages:** Keylength: 13, Separator: Comma, Ciphertext alphabet: ADFGVX, Plaintext alphabet: ABCDEFGHIJKLMNOPQRS TUVWXYZ0123456789

**Parameter:** Monoalphabetic Substitution Analyzer

**Attack type:** Algorithm: Dictionary & Genetic, Restarts: 100

**Language:** English, Use spaces:

**Advanced settings:** Invalid Characters: ignore



# Breaking cipher VI

## ■ Breaking Mixed cipher (vii)

Analysis		Start Time:	1/14/2020 3:40:10 PM	End Time:	1/14/2020 3:40:32 PM	
		Elapsed Time:	00:00:22	keys / sec:	10309 (235014)	
		MessageLabel	1	Analyzed keylength:	13	
Bestlist	#	Score	IoC 1	IoC 2	Transposition key	Transposition result
	1	200899	6.5	0.9	LIADJHFKBMCGE	EGIUYSDFMUHSNYDNKHJLMVXIESNKUWHAHLEKQJIESNK
	2	134214	5.21	0.57	LIADJHFKMBCGE	EGIUESDFMUJGNYDNKHJLMVXGQSNKUWHAHLEKN1IESN
	3	129685	5.71	0.53	LIADJHFM8KCGE	EGIS0SDFMUTGNVDNHKJLMVXCKSNKUTKAHLEKKPIESNIV
	4	125062	5.93	0.5	LIADJHFMCKBGE	EGISUYDFMUSHNYDNHKJLMVXEISNKUTEHGLEKJQIESNIE?
	5	113683	5.89	0.44	LIADJMCHFKBGE	EGGUUYDFMSUHNVDNHKJLMVEXISNKTGWGHLEJKQIESOI
	6	111494	5.84	0.42	LIAMCDJHFKBGE	EGSIUYDFSMUHNYBJNKJLMEVXISNHEUWGHLEJKQIEUANI
	7	99648	5.49	0.37	DELIAMJHFKBGC	ZEGIUYSDFSUHMNYBNKJLMDXIWSNHUWGEHLKKQJCEUI
	8	96528	5.35	0.36	LEDMAIJHFKBGC	BYKIUYVAXAUHMZLNANKJLJAPXIWMTHUWGBKLLKKQJCCW
	9	95314	5.31	0.35	FELMDUHAKBGC	TA2IUYDSAXHMHZMNEJLLDPSIWSNTUKGEHKKLQJICENW
10	94710	5.51	0.34	GFLIJHAKBMDEC	NEIYZSVAXHSAOYNEHNJLPSIDKXNUKHSATKLQKJENWU	



# Breaking cipher VI

## ■ Breaking Mixed cipher (viii)

Start: 1/14/2020 3:40:26 PM End: 1/14/2020 3:40:32 PM  
Elapsed: 00:00:06  
Tested keys: Keys / sec:

#	Value	Attack	Key
1	-4.39408	G	subtionacdefghjklmpqrwxyz INCRYPTOGRAPHYTHEADFGVXCIPHERWASAFIELDCIPHERUSEDBYTHEGERMAN
2	-13.69601	D	ABCDEFGHIJKLMNopqrstuvwxyz EGIUYSDFMUHSNYDNKHJLMVXIESNKUWHHAHLEKQJIESNKUBAKJCYDNKMKUR



# Basic Crypto II (LAB V)

- **Task V. Repeat the analysis at lab (10 MINS)**
  - **Mixed**
    - Try different parameters shown in the slide
    - Try also assignments 31-36 (from assignment M4 slides)



# Cryptology for IoT

**Modules M4, M6, M8**  
**Session of 10th May, 2022.**

M4.6 Briefing of the session  
M4.7 Tasks to do in the lab

Prof.: Guillermo Botella