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

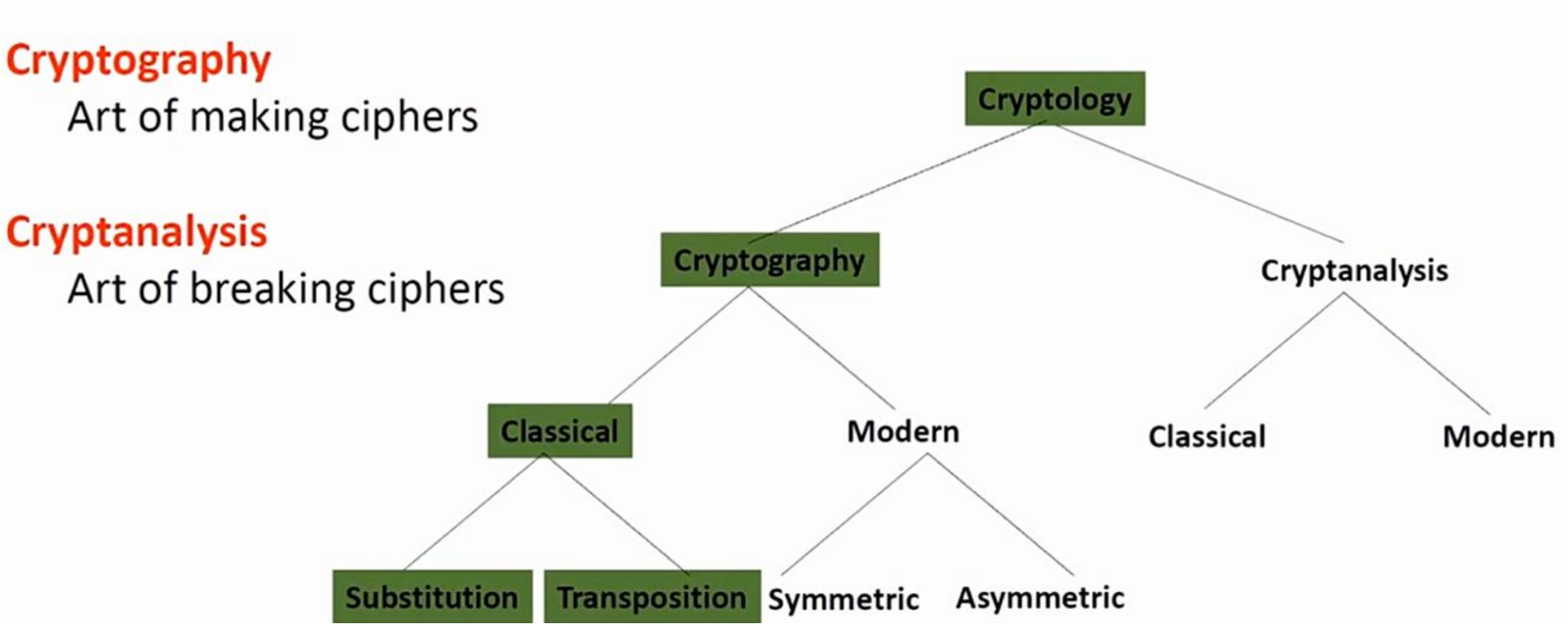
Prof.: Guillermo Botella

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



Basic Crypto I

- **Cryptography using Cryptool (video+slides)**
  - **Terms**

Cipher

- Encryption method/algorithm

Plaintext

- Non-encrypted text

Ciphertext

- Encrypted text

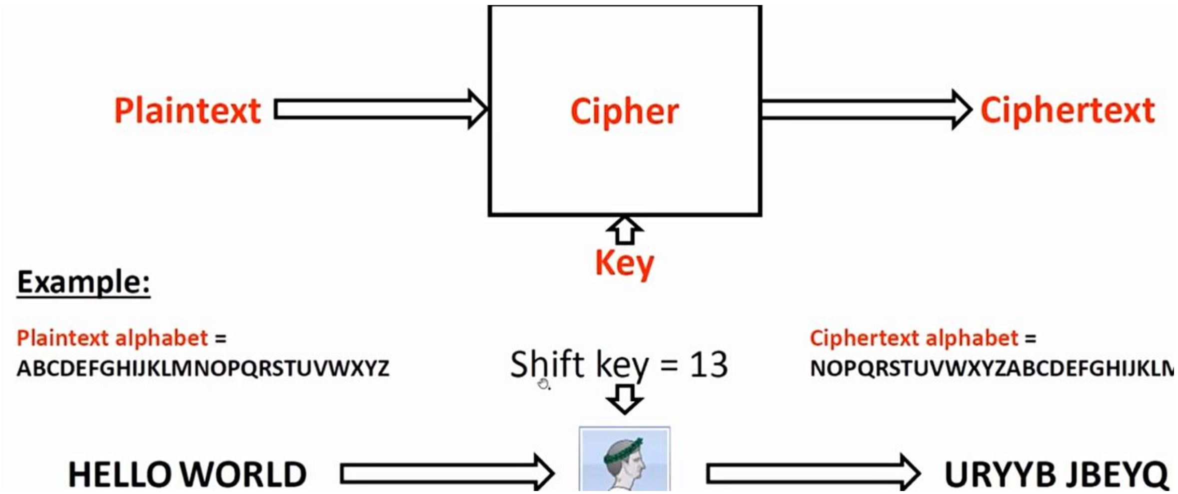
Key

- 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

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

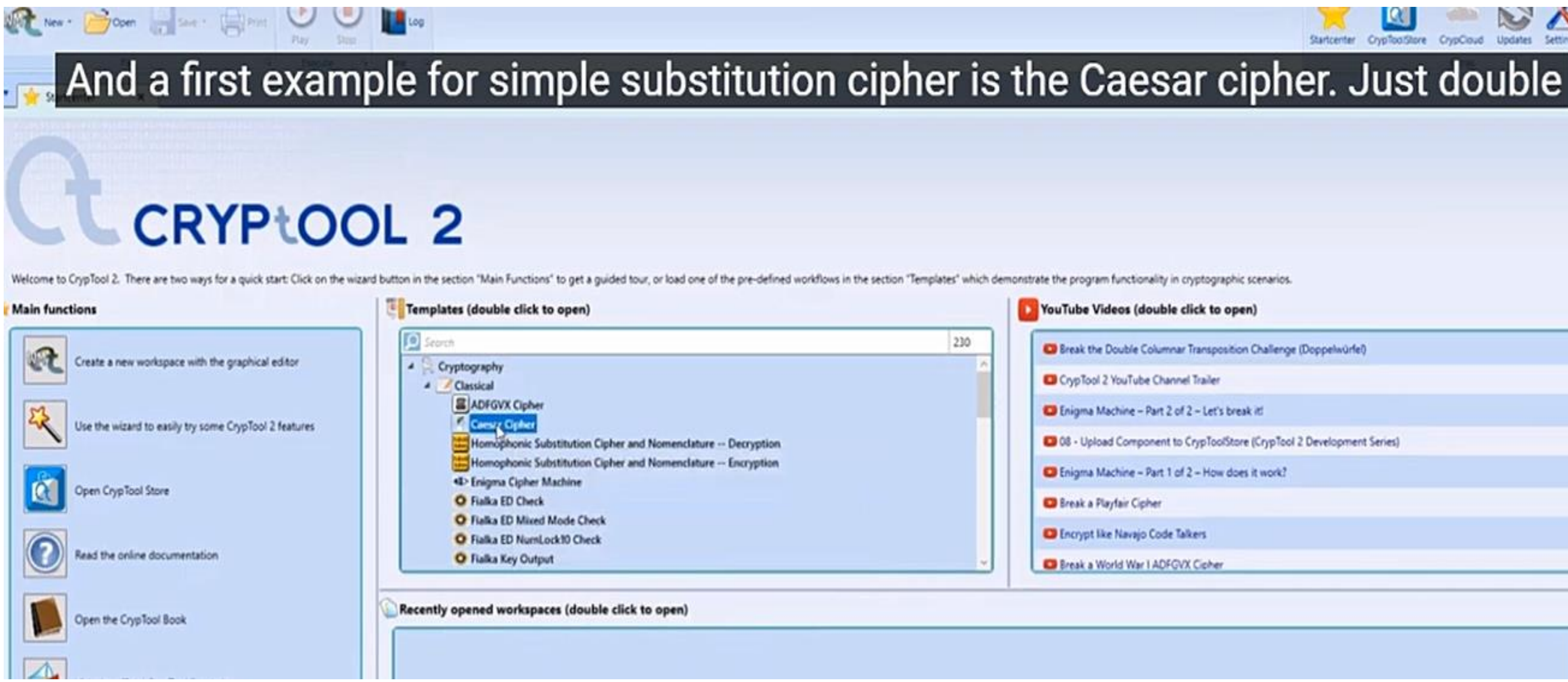
- Replacement are two letters

Example:

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

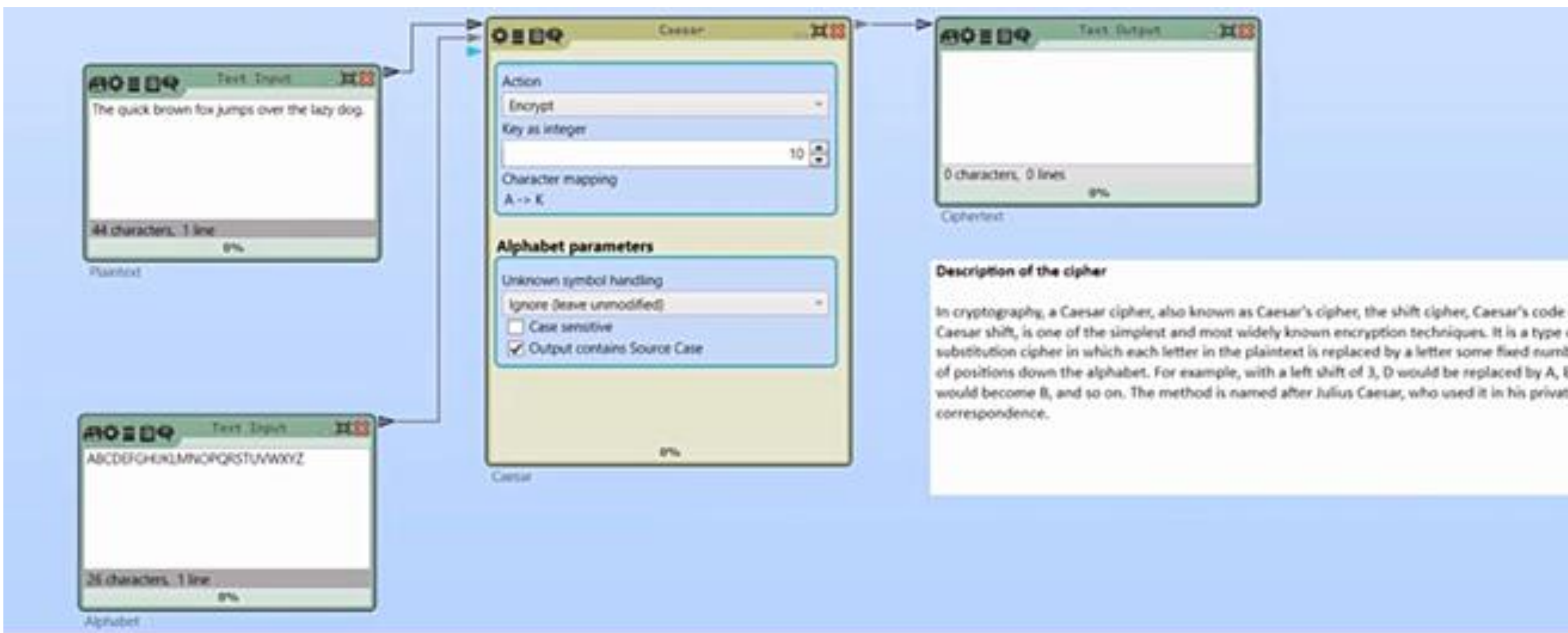
Basic Crypto I

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



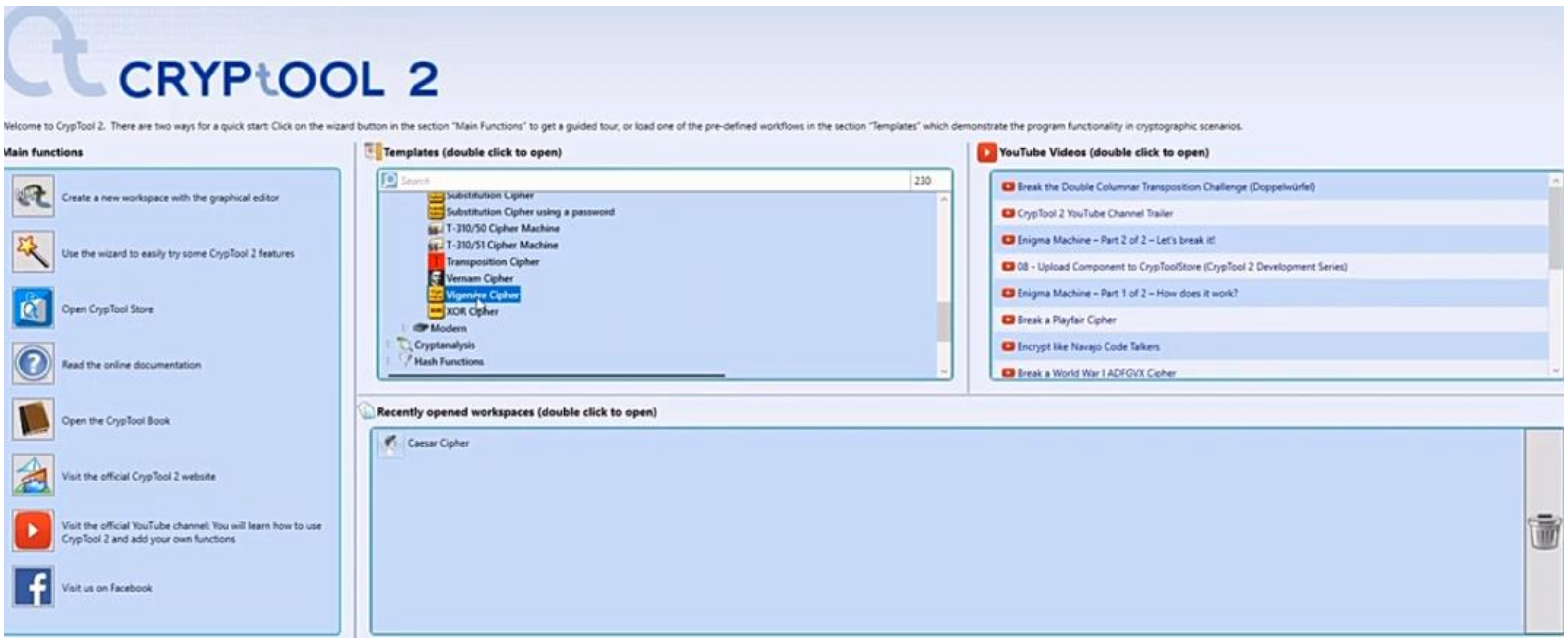
Basic Crypto I

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



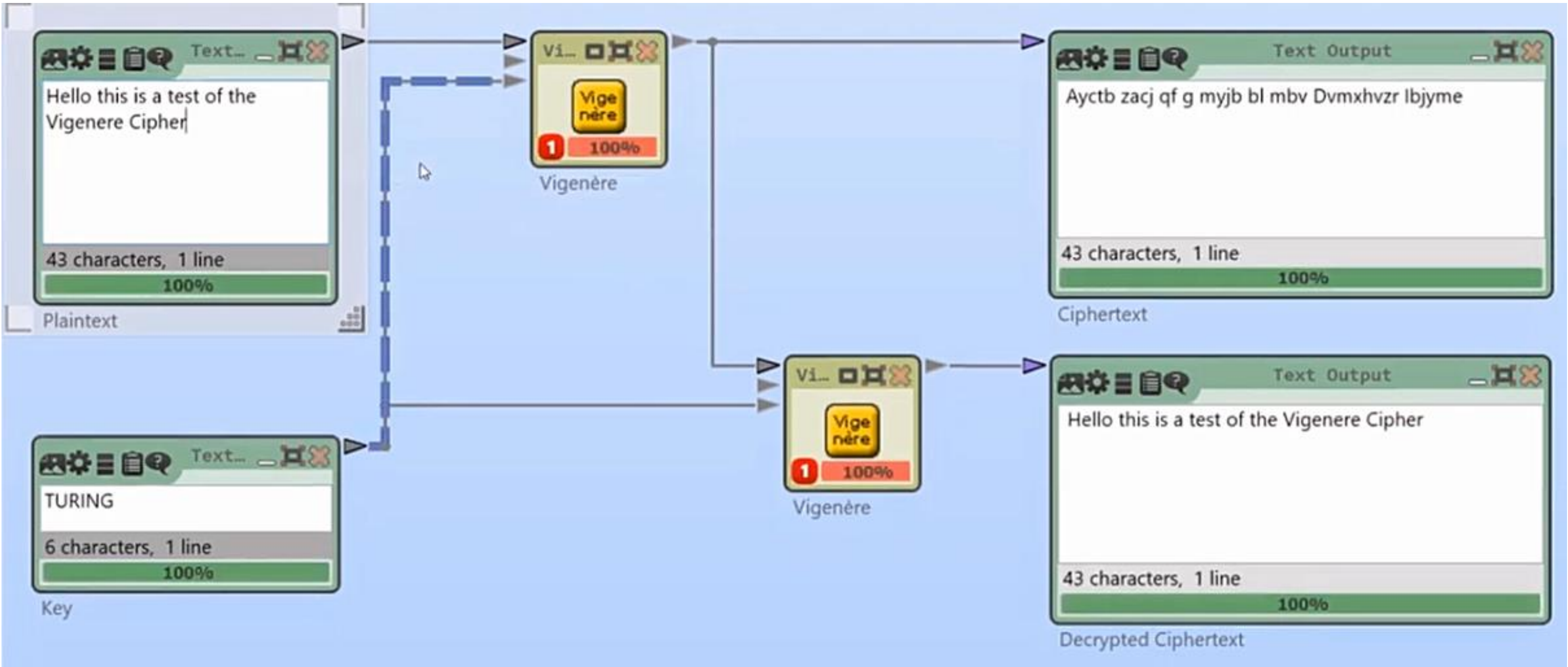
Basic Crypto I

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



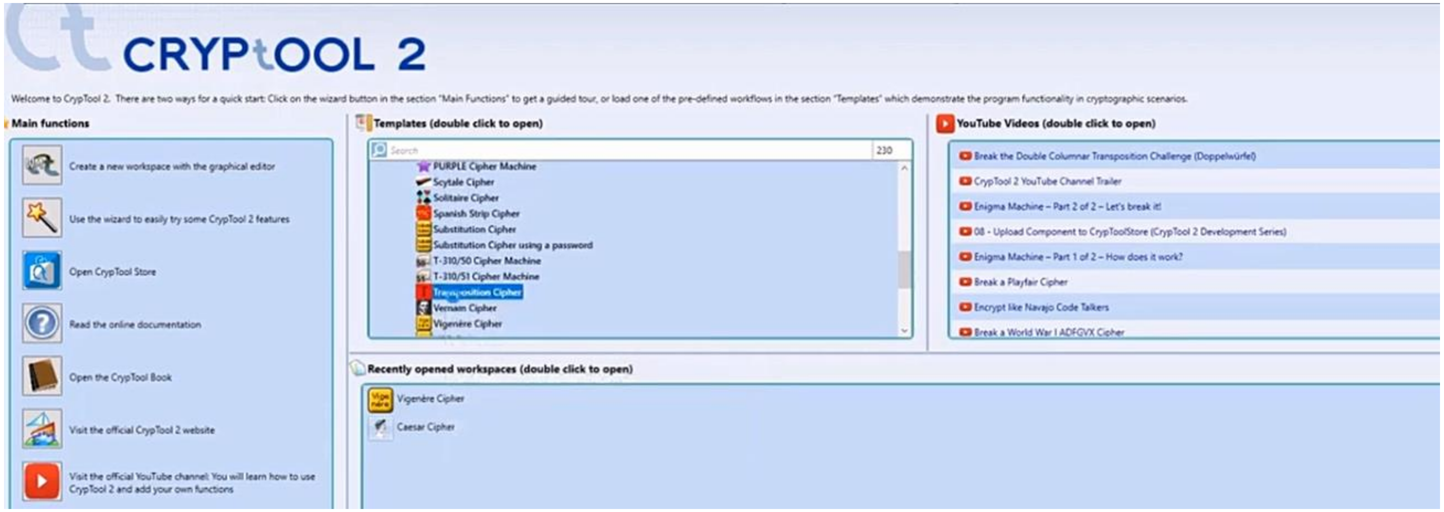
Basic Crypto I

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



Basic Crypto I

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

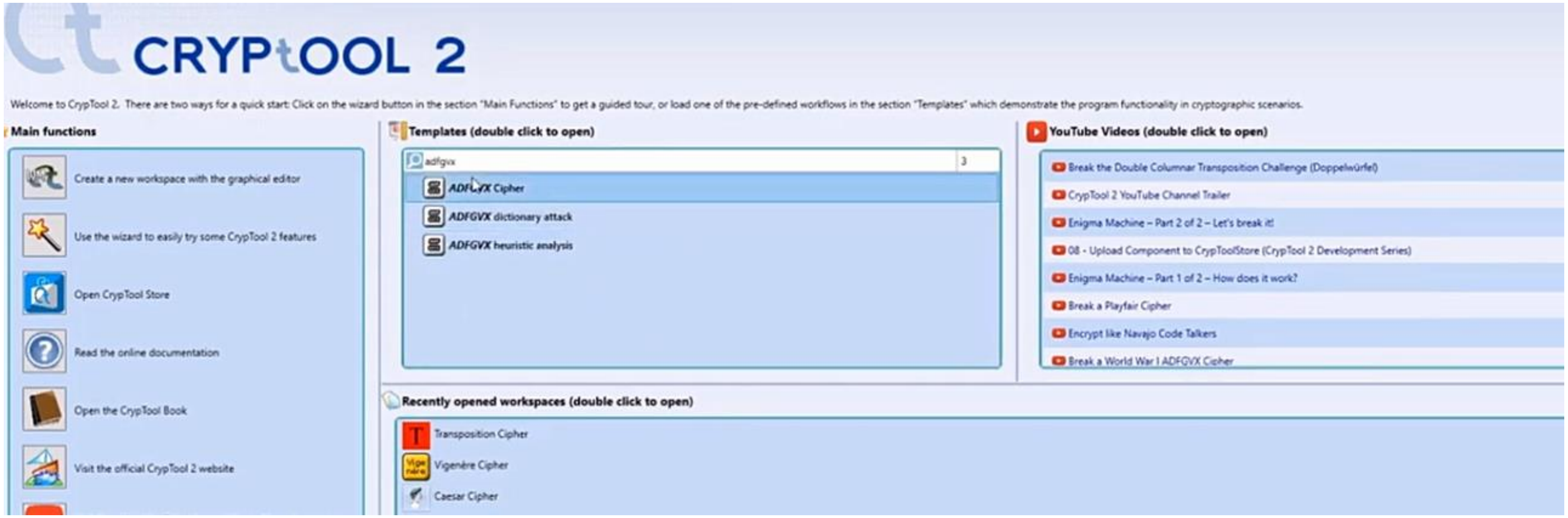


Basic Crypto I

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

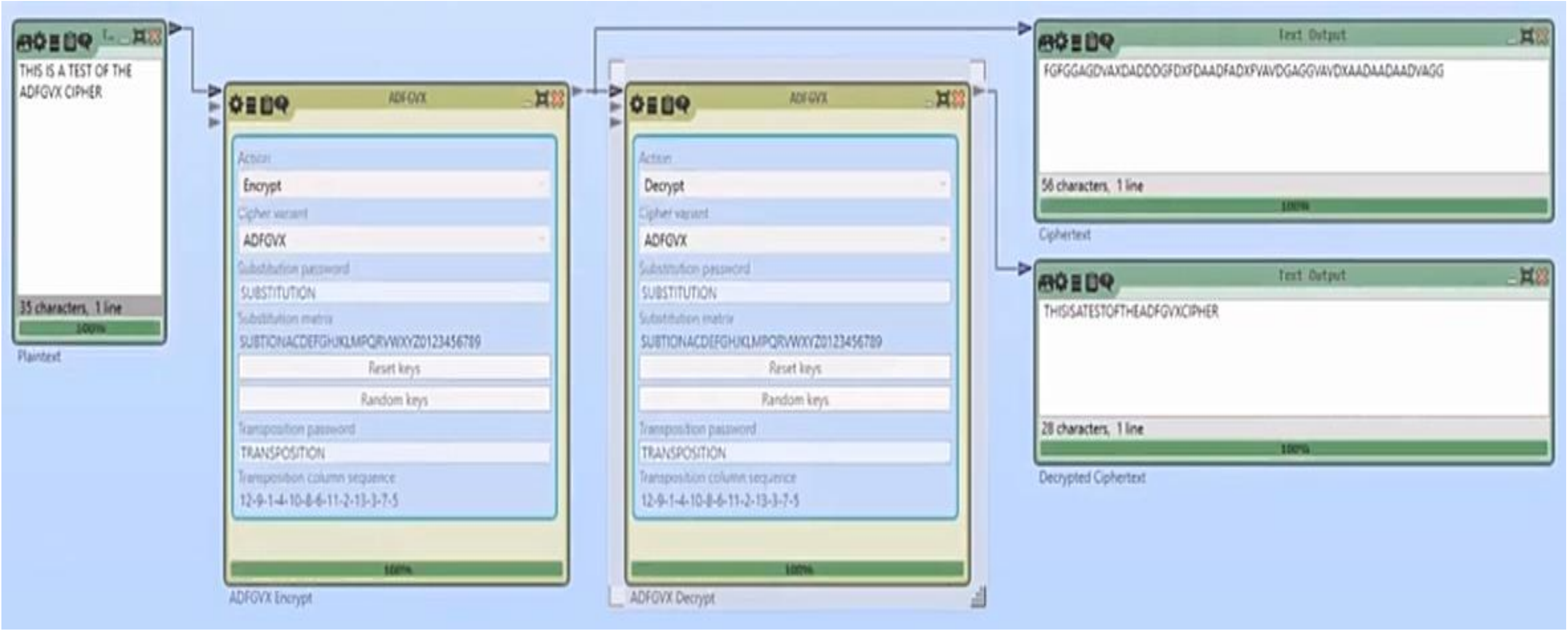
Basic Crypto I

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



Basic Crypto I

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

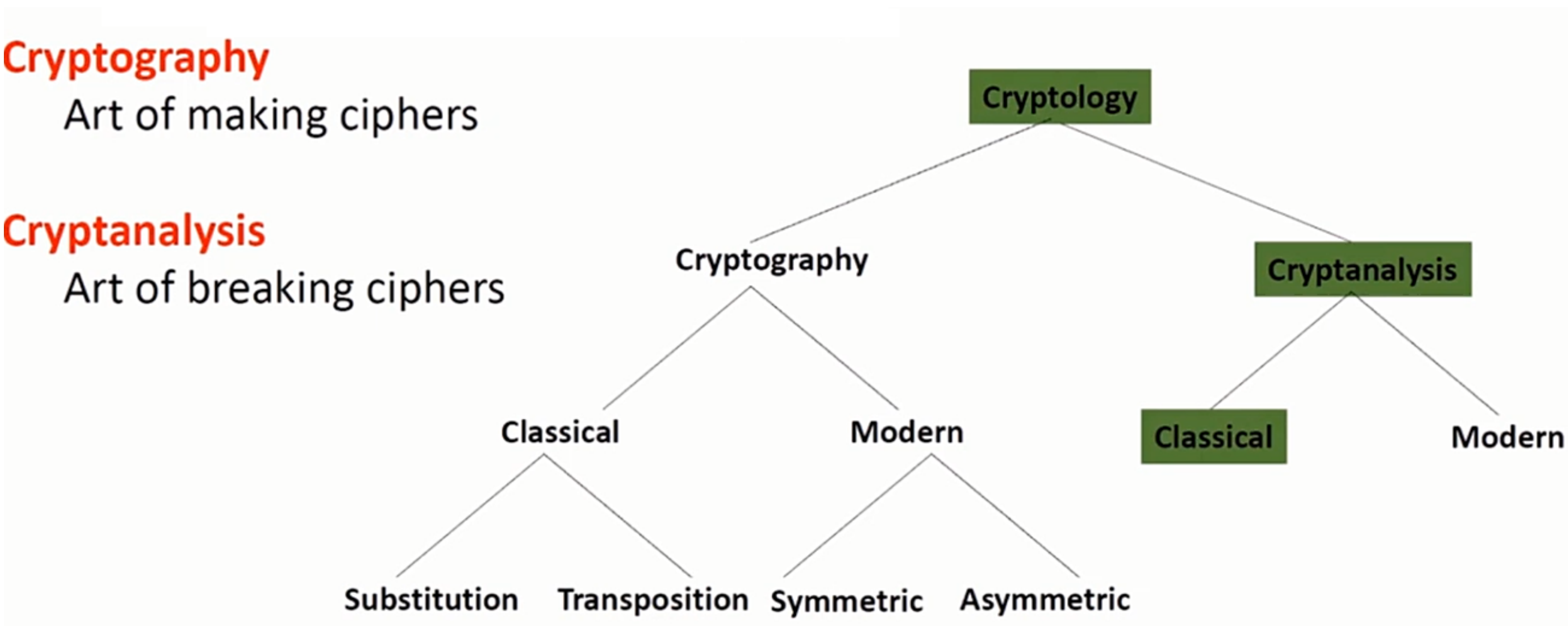


Basic Crypto I

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

Basic Crypto II

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



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)

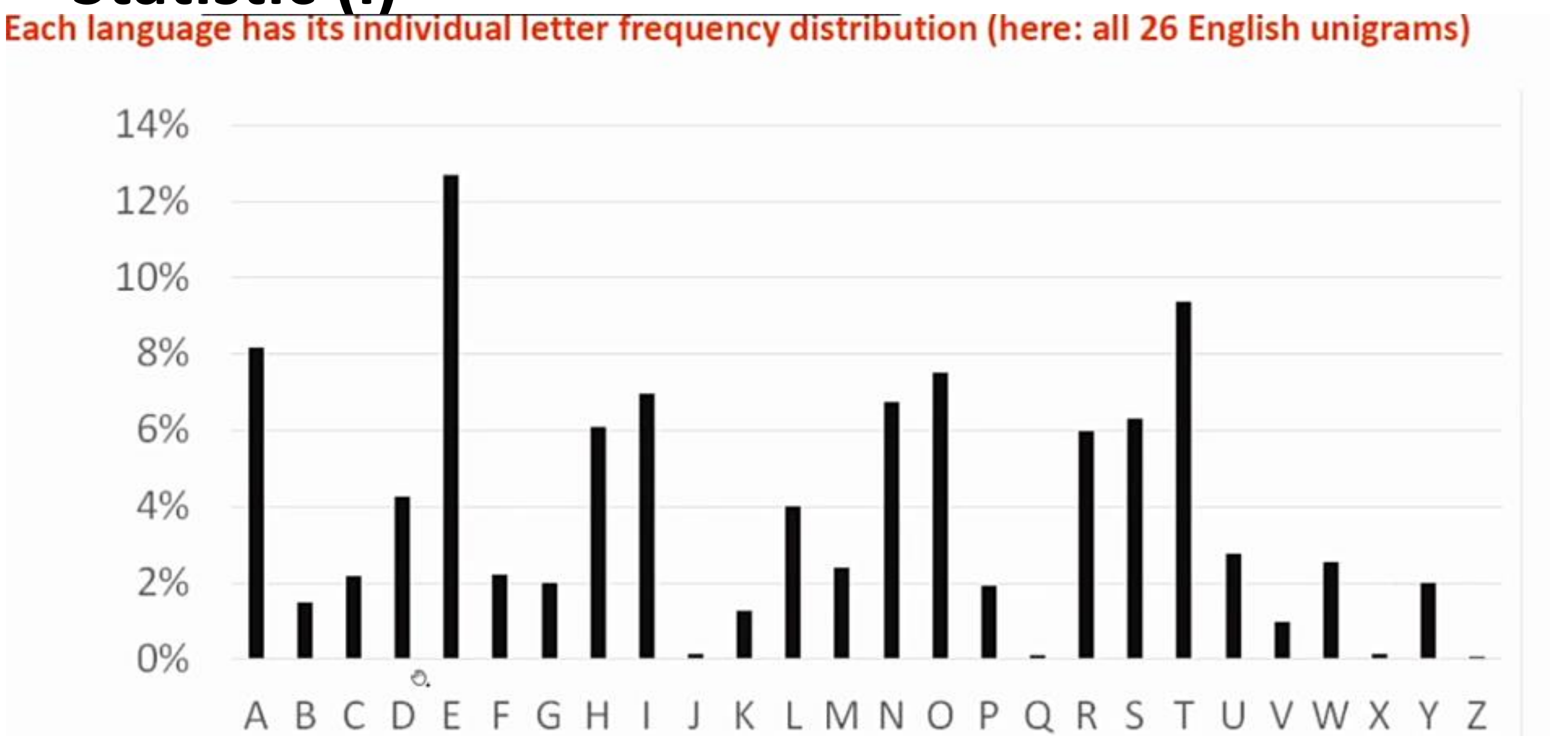
- E.g. break a MASC by hand using the knowledge of letter frequency distribution
- E.g. cut transposition ciphertext into paper strips and rearrange them

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

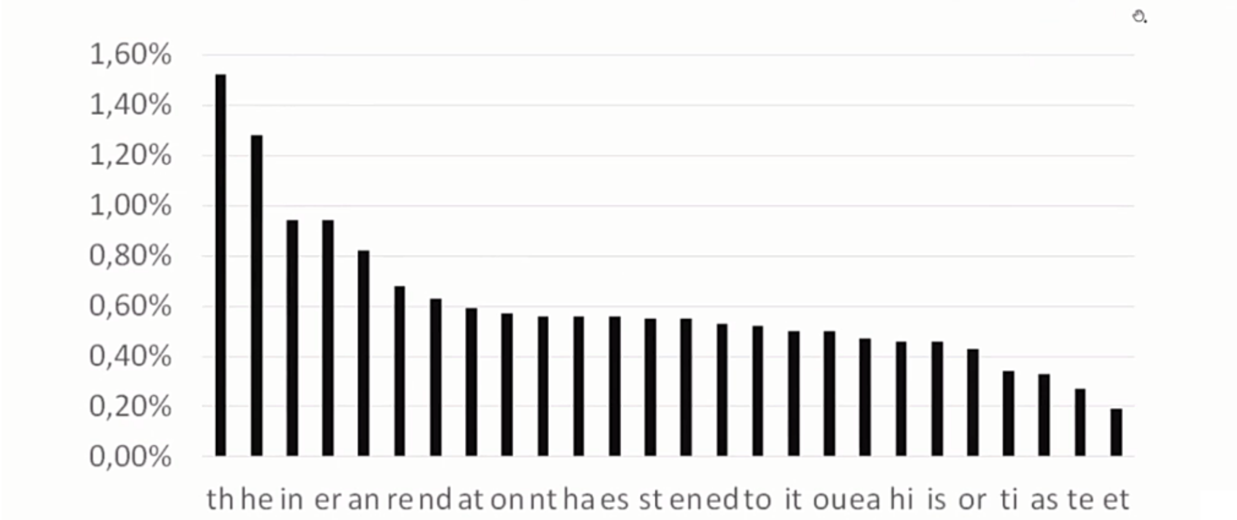
- **Cryptography using Cryptool (video+slides)**
  - **Statistic (i)**



Basic Crypto II

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

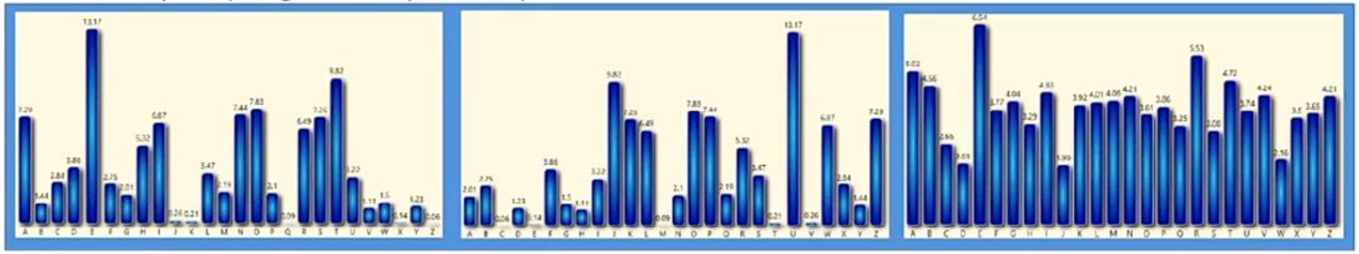
Each language has its individual letter frequency distribution (here: 39 most frequent English bigrams)



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):



Plaintext / transposition cipher   Monoalphabetic substitution   Polyalphabetic substitution

Basic Crypto II

- Cryptography using Cryptool (video+slides)
  - Statistic (iv)

Ciphertext (26 letters) = BUUBDL UIF FOFNZ JO UIF FWFOJOH

Count unigrams

B = 2	D = 1	F = 6	H = 1	I = 2	J = 2
L = 1	N = 1	O = 4	U = 4	W = 1	Z = 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 colums 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“

AKEI
ECHN
TTVG
TAEN
WATE

→

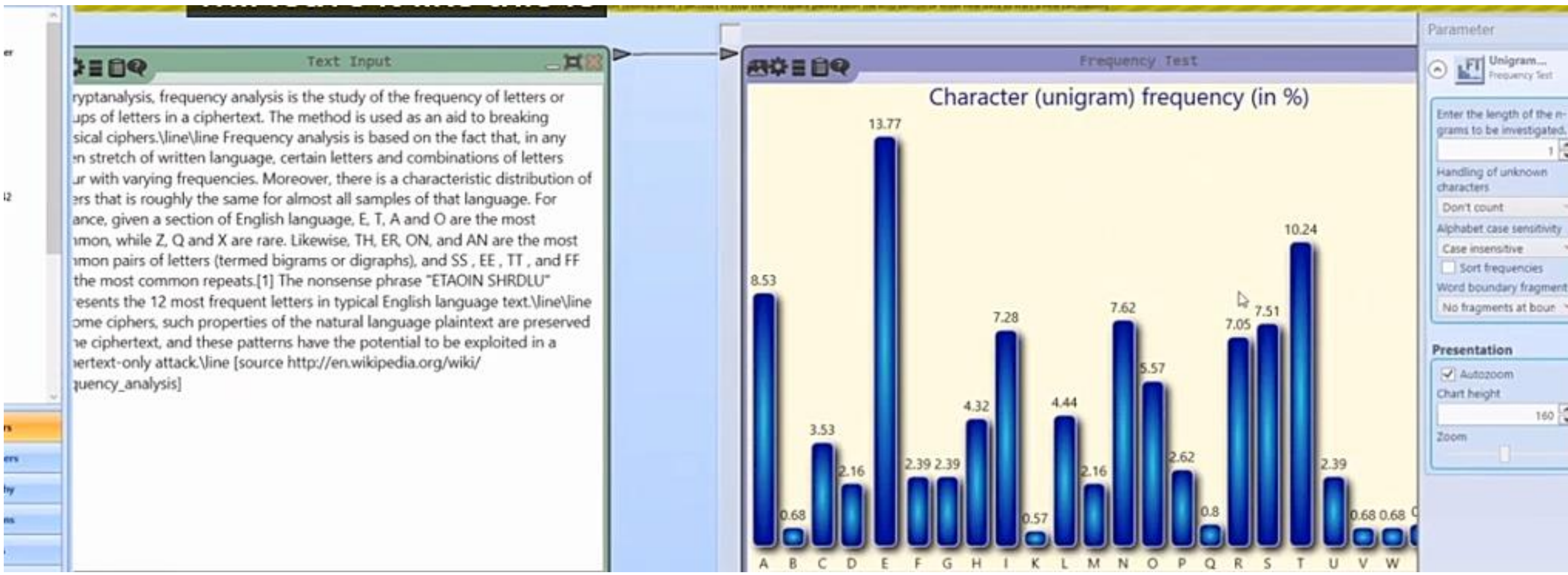
WATE
ECHN
AKEI
TAEN
TTVG

→

WE ATTACK AT THE EVENING

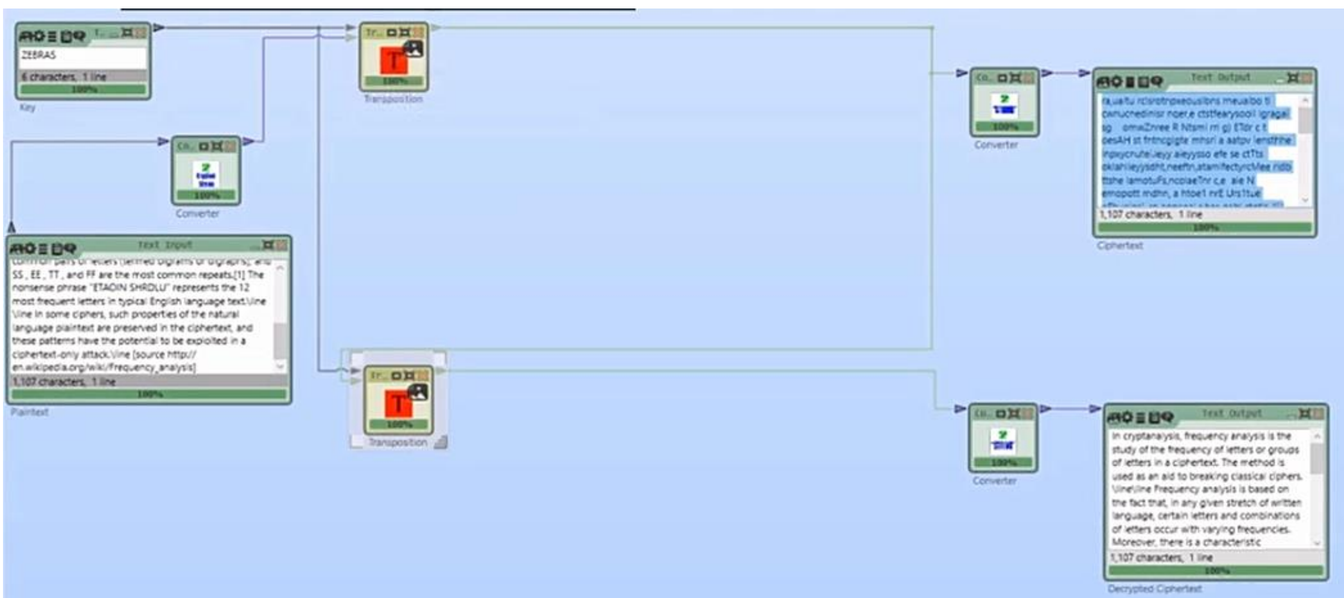
Basic Crypto II

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



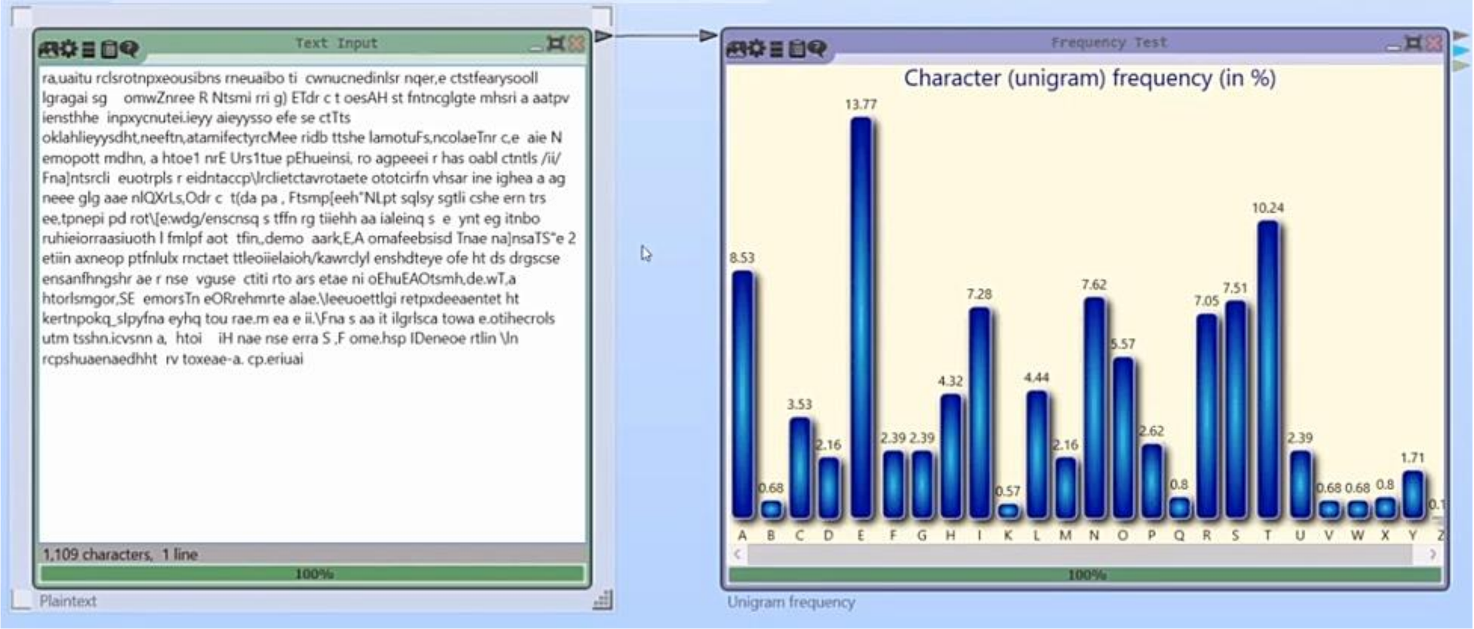
Basic Crypto II

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



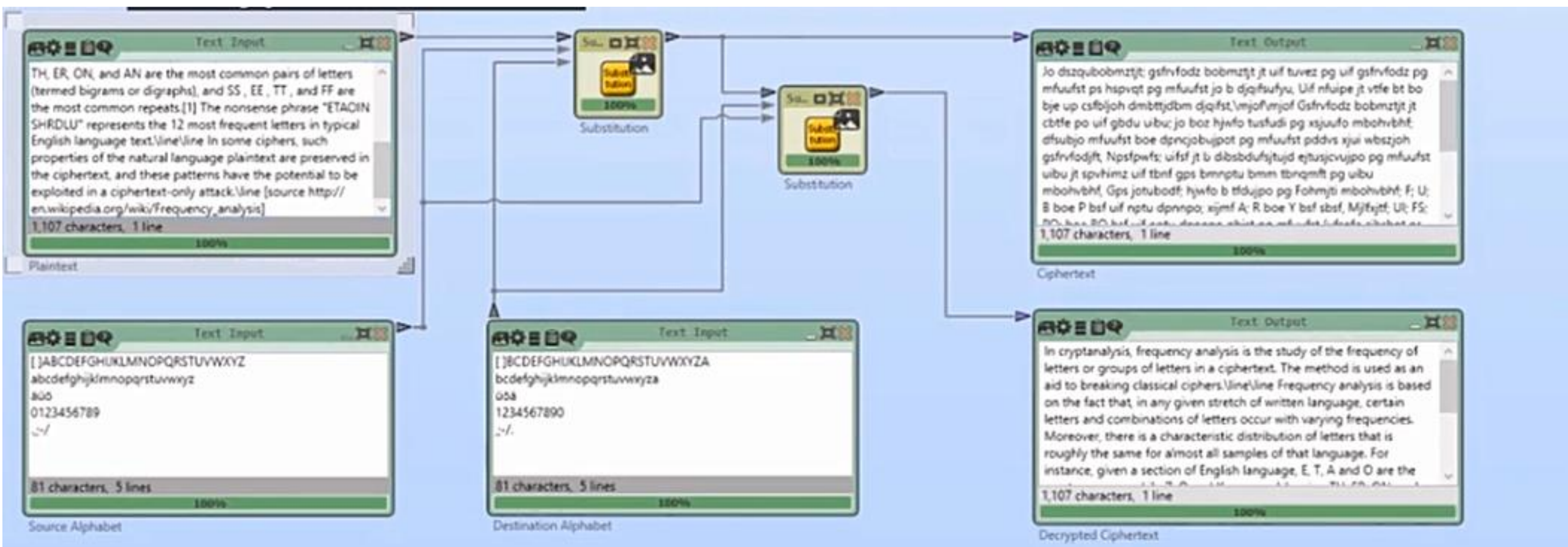
Basic Crypto II

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



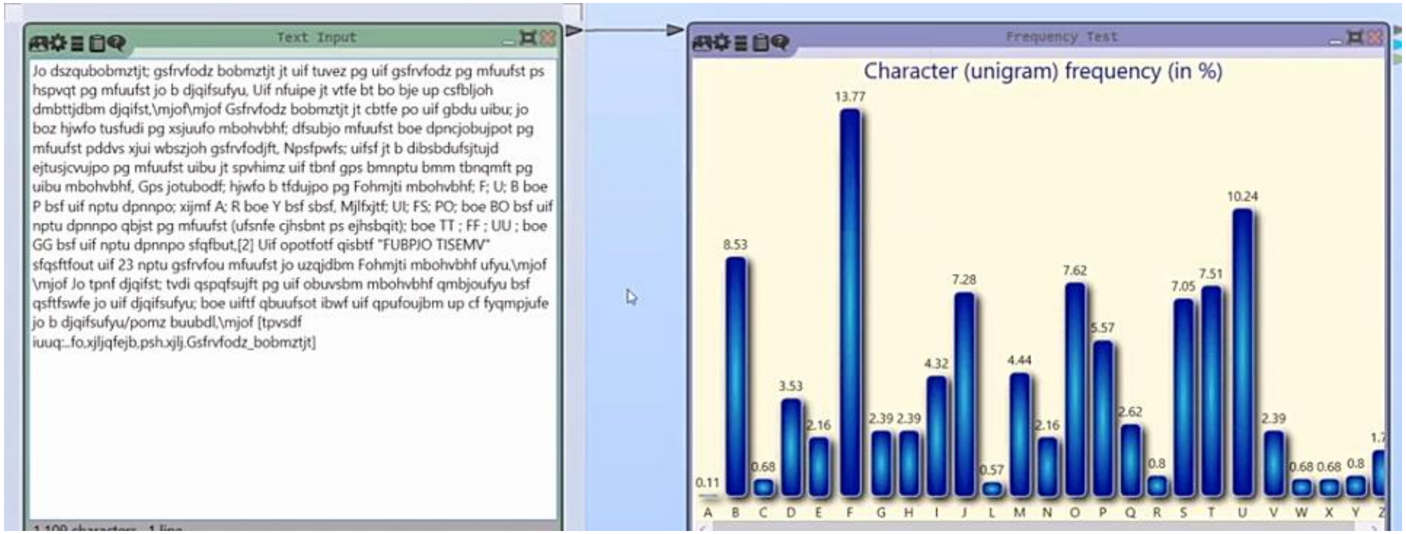
Basic Crypto II

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



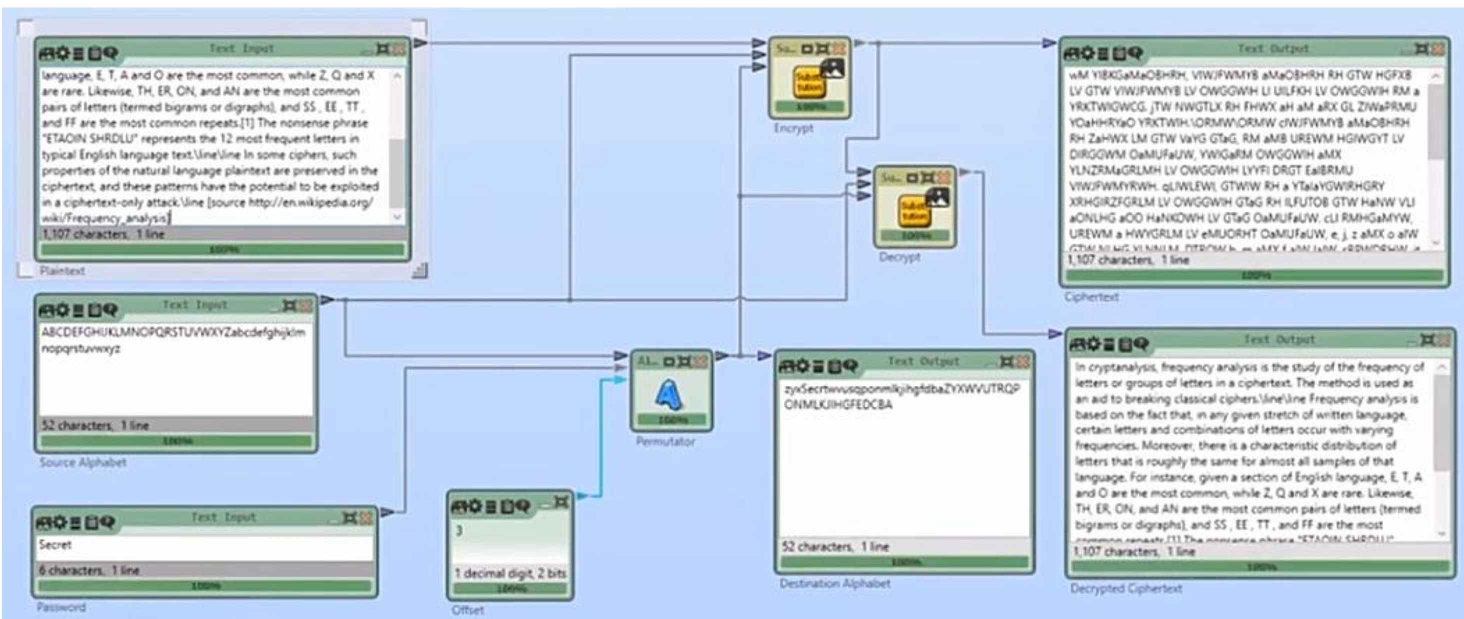
Basic Crypto II

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



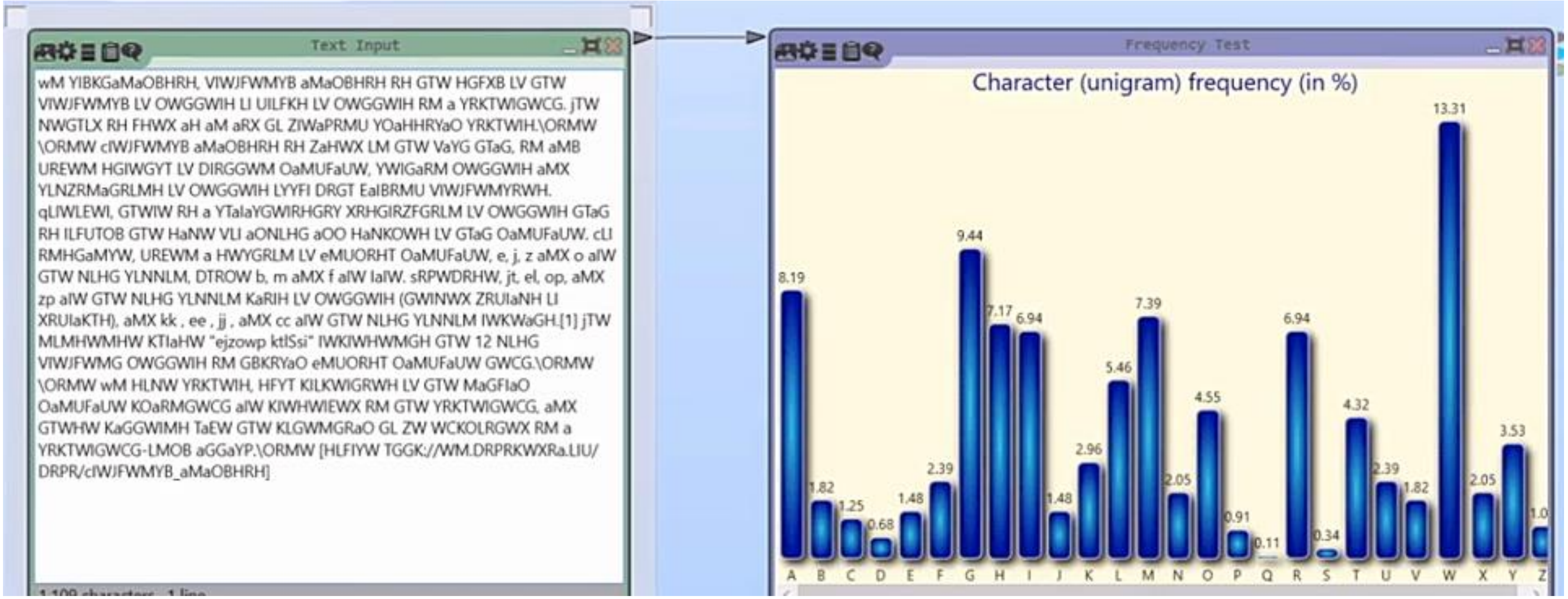
Basic Crypto II

- Cryptography using Cryptool (video+slides)
  - Letter Frequency of ciphers (vi)
  - Substitution (iii), password



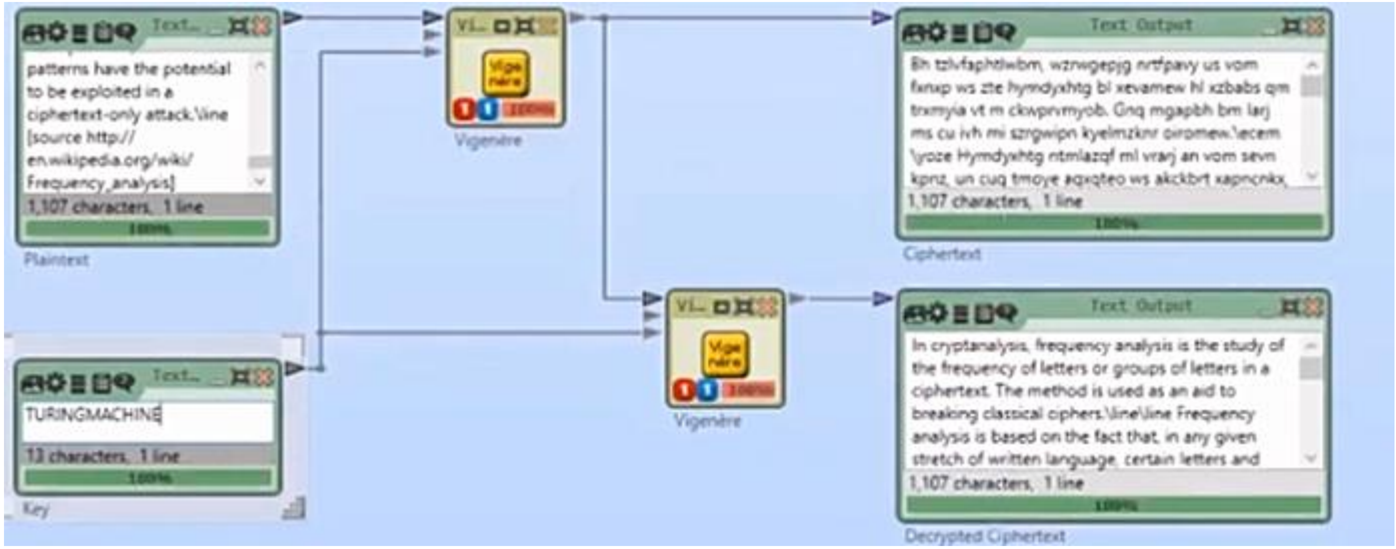
Basic Crypto II

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



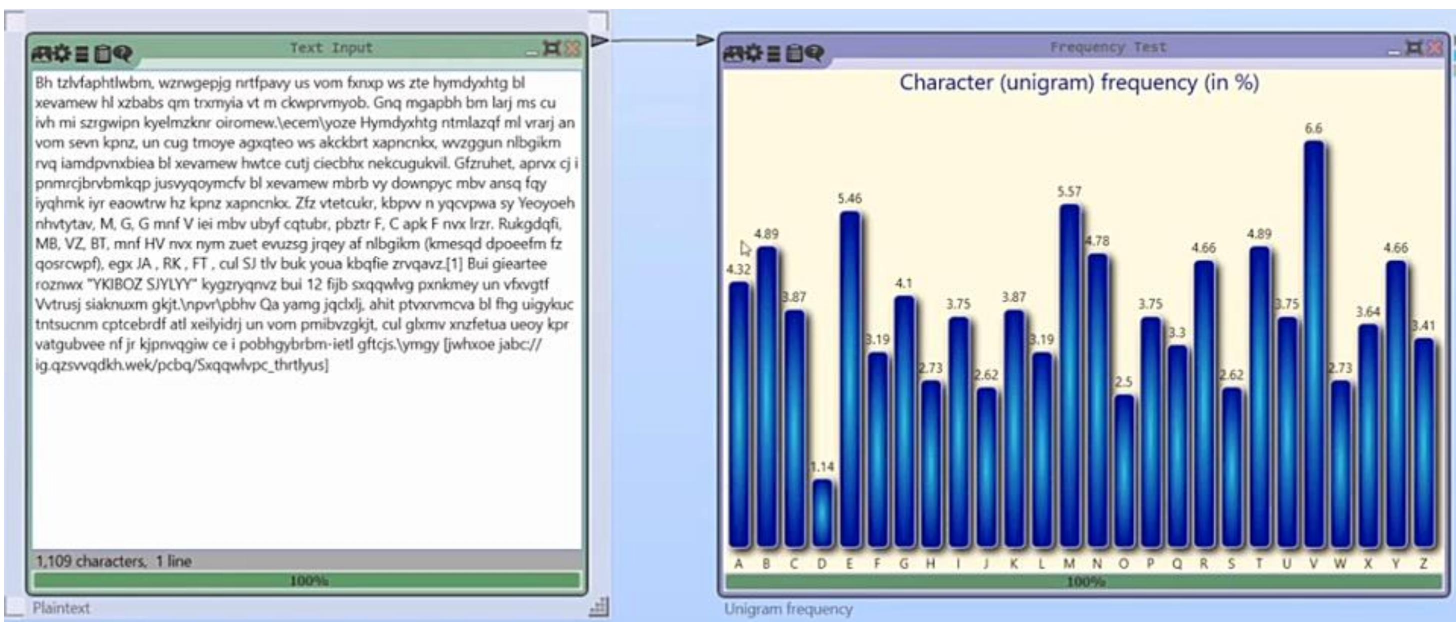
Basic Crypto II

- 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)

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

Breaking cipher I

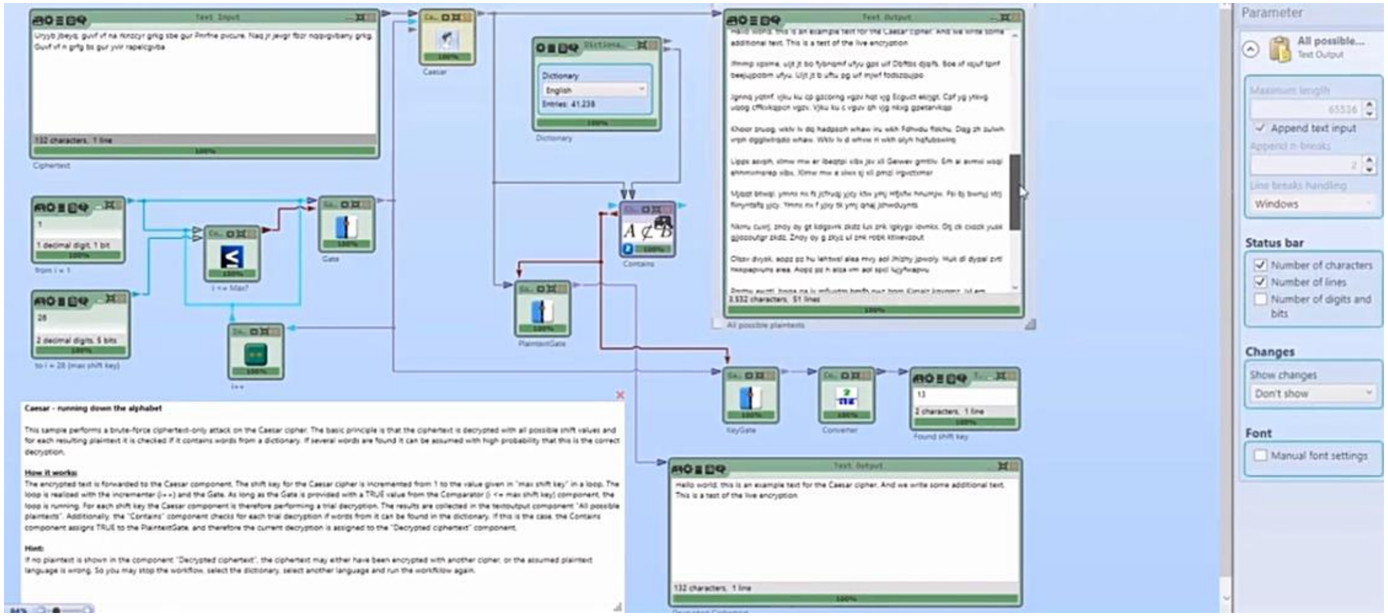
- **Breaking Caesar (video+slides) (i)**
  - **Shift of 13**

Breaking cipher I

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

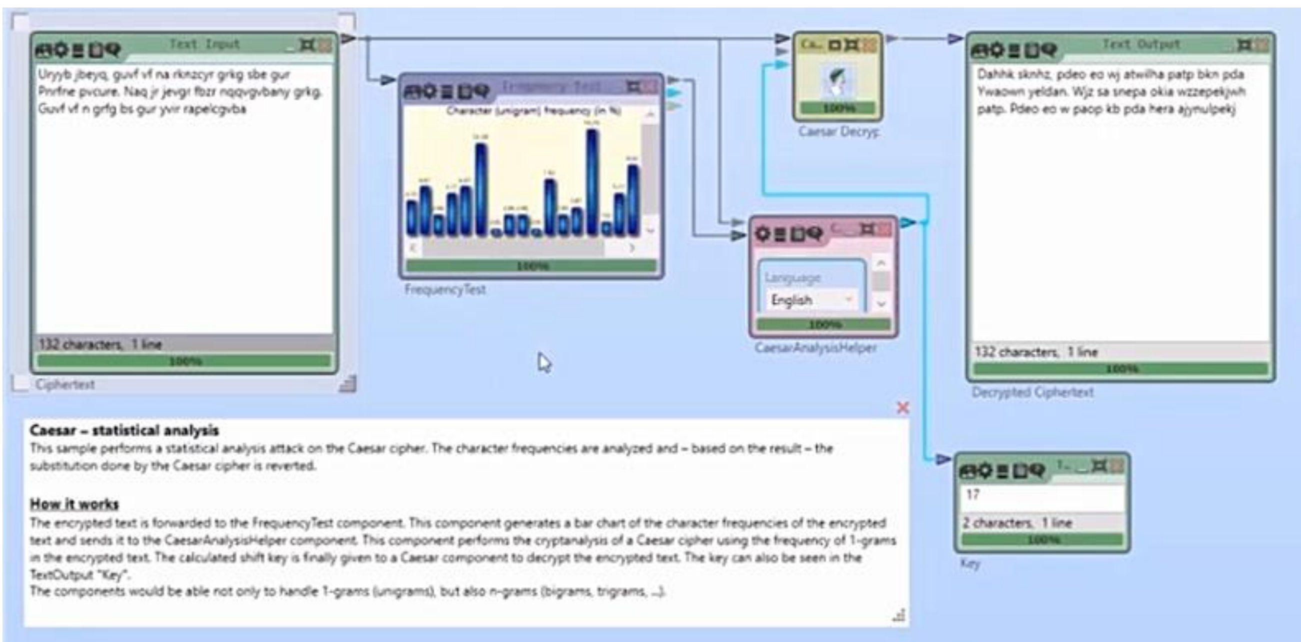
Breaking cipher I

- Breaking Caesar (video+slides) (iii)
  - Brute Force analysis



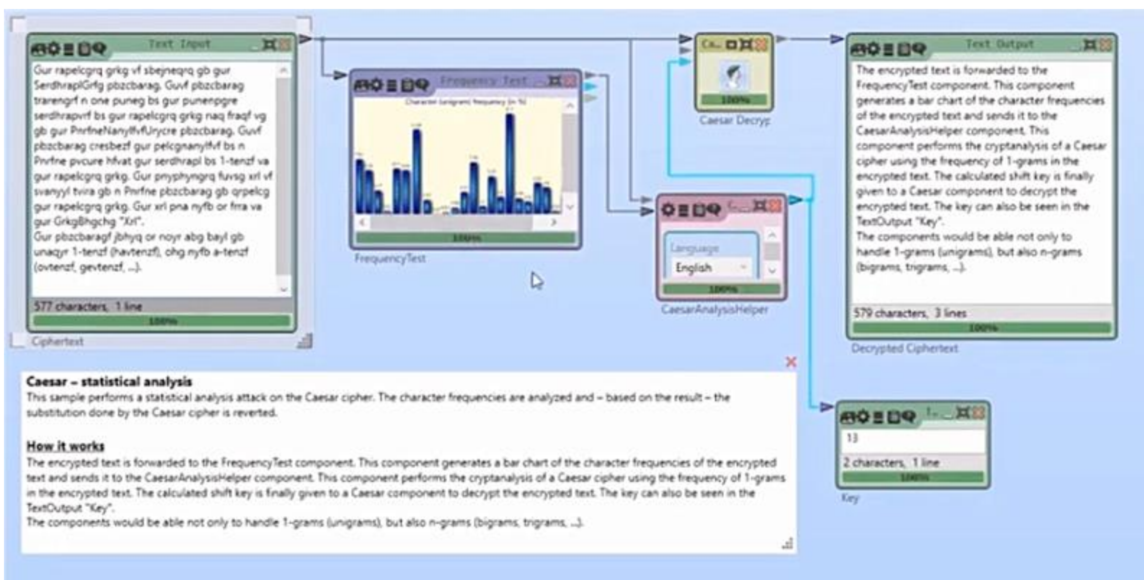
Breaking cipher I

- 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

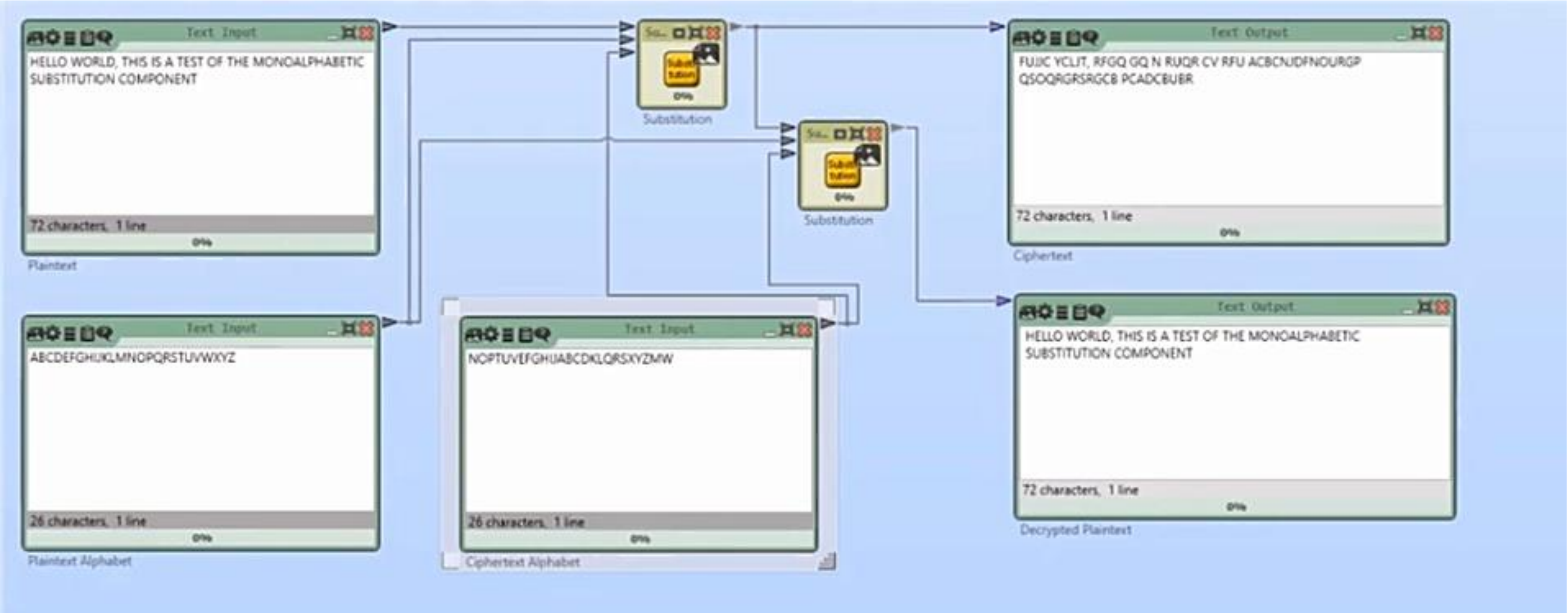
Breaking cipher II

■ 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 (=> 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
```

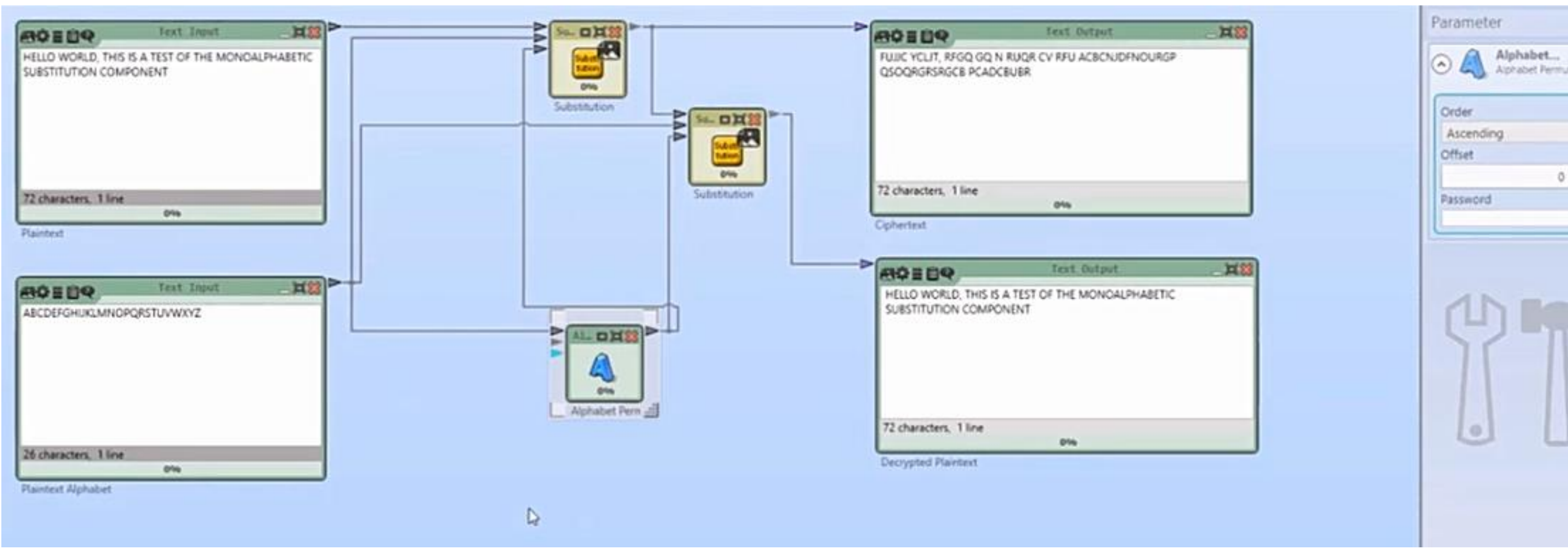
Breaking cipher II

- **Breaking Monoalphabetic substitution (ii)**



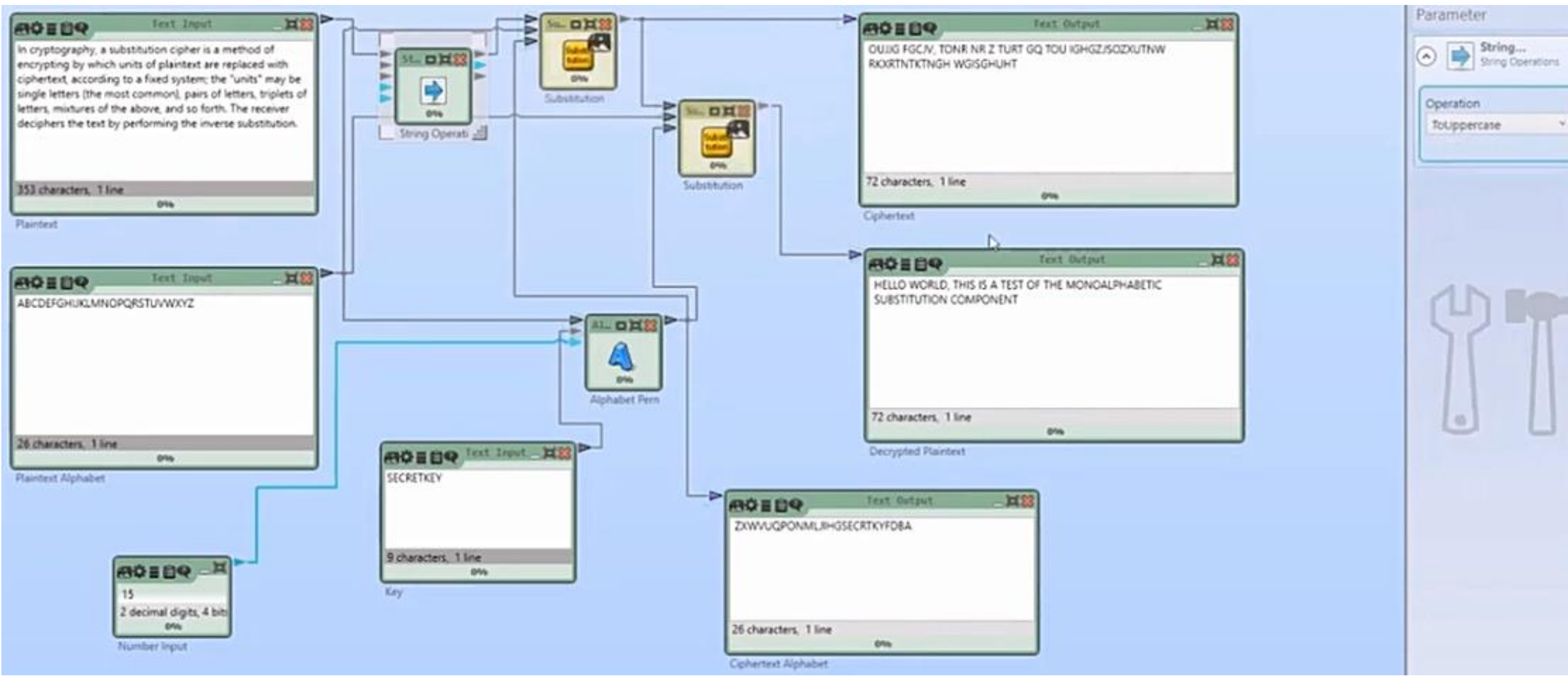
Breaking cipher II

- **Breaking Monoalphabetic substitution (iii)**



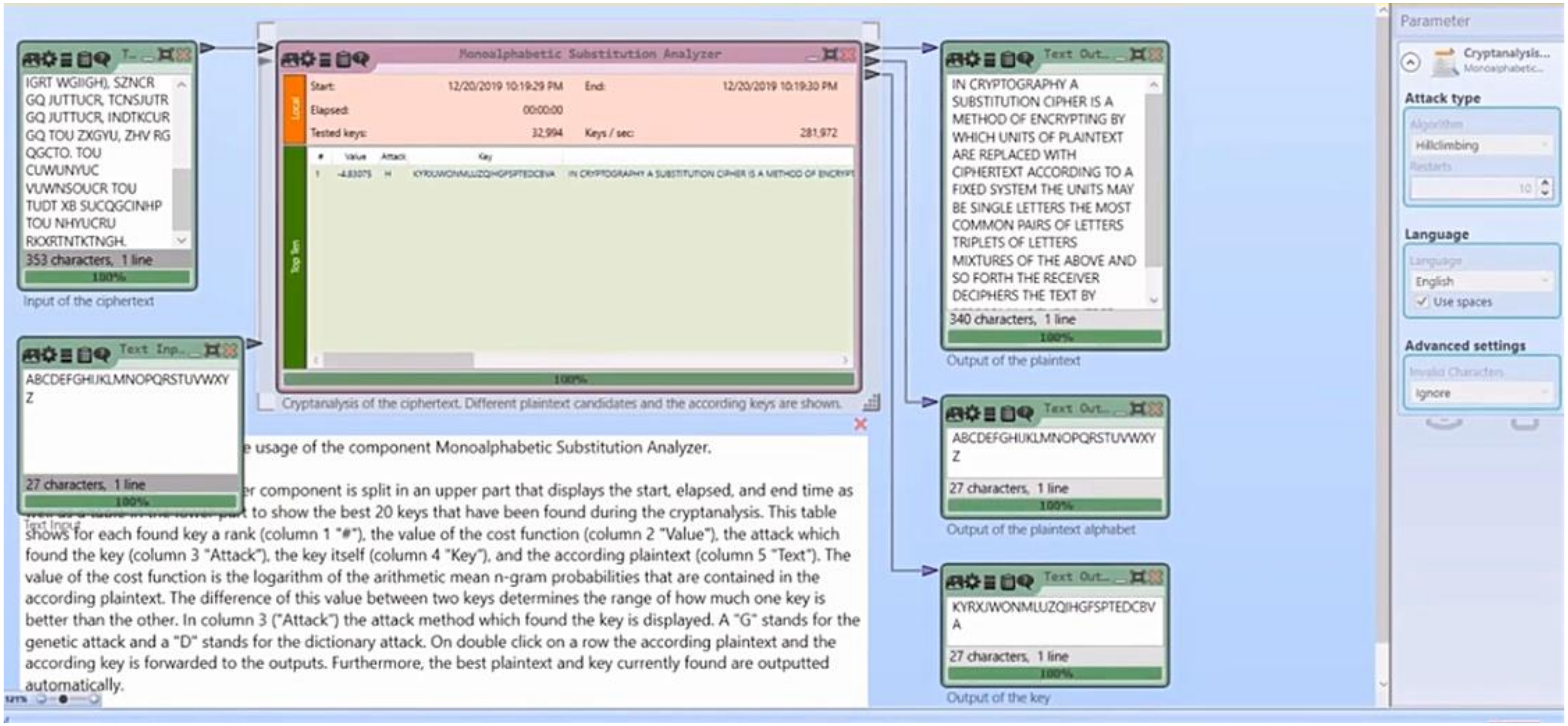
Breaking cipher II

- **Breaking Monoalphabetic substitution (iv)**



Breaking cipher II

- **Breaking Monoalphabetic substitution (v)**



Basic Crypto II (LAB II)

- **<u>Task II. Reproduce the analysis at lab (20 MINS)</u>**
- **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)

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

Module 4: Cryptology for IoT

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: $C_i = (K_i + P_i) \ mod\ 26$
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 $A = 0,\ B = 1,\ C = 2, \ldots,\ Z = 25$

	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)

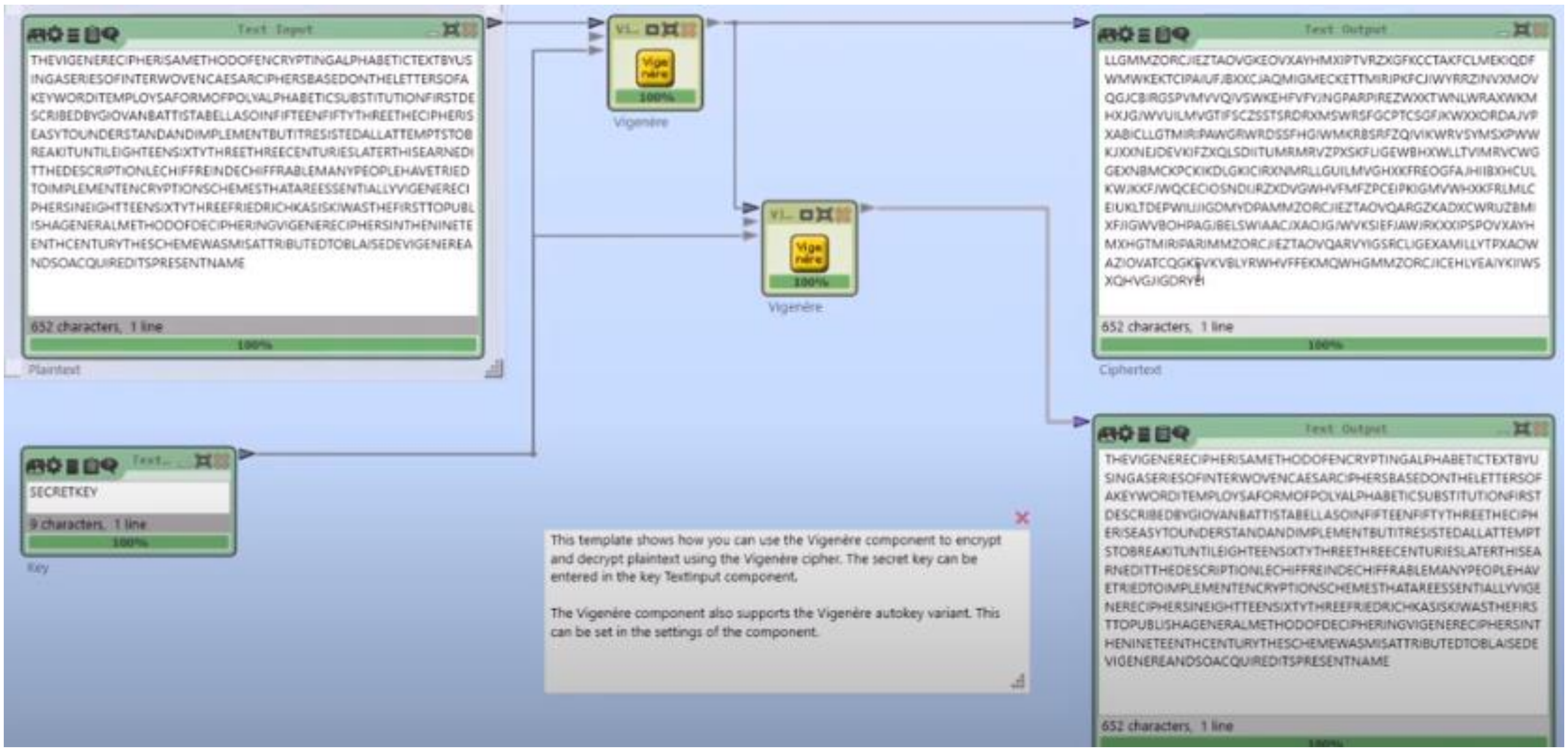
- The Vigenère Cipher can be implemented with **different ciphertext alphabets**

1.	Latin Alphabet (today):	ABCDEFGHIJKLMNOPQRSTUVWXYZ	(26 letters)
2.	Latin Alphabet ("historic"):	ABCDEFGHIKLMNOPQRSTUWXYZ	(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

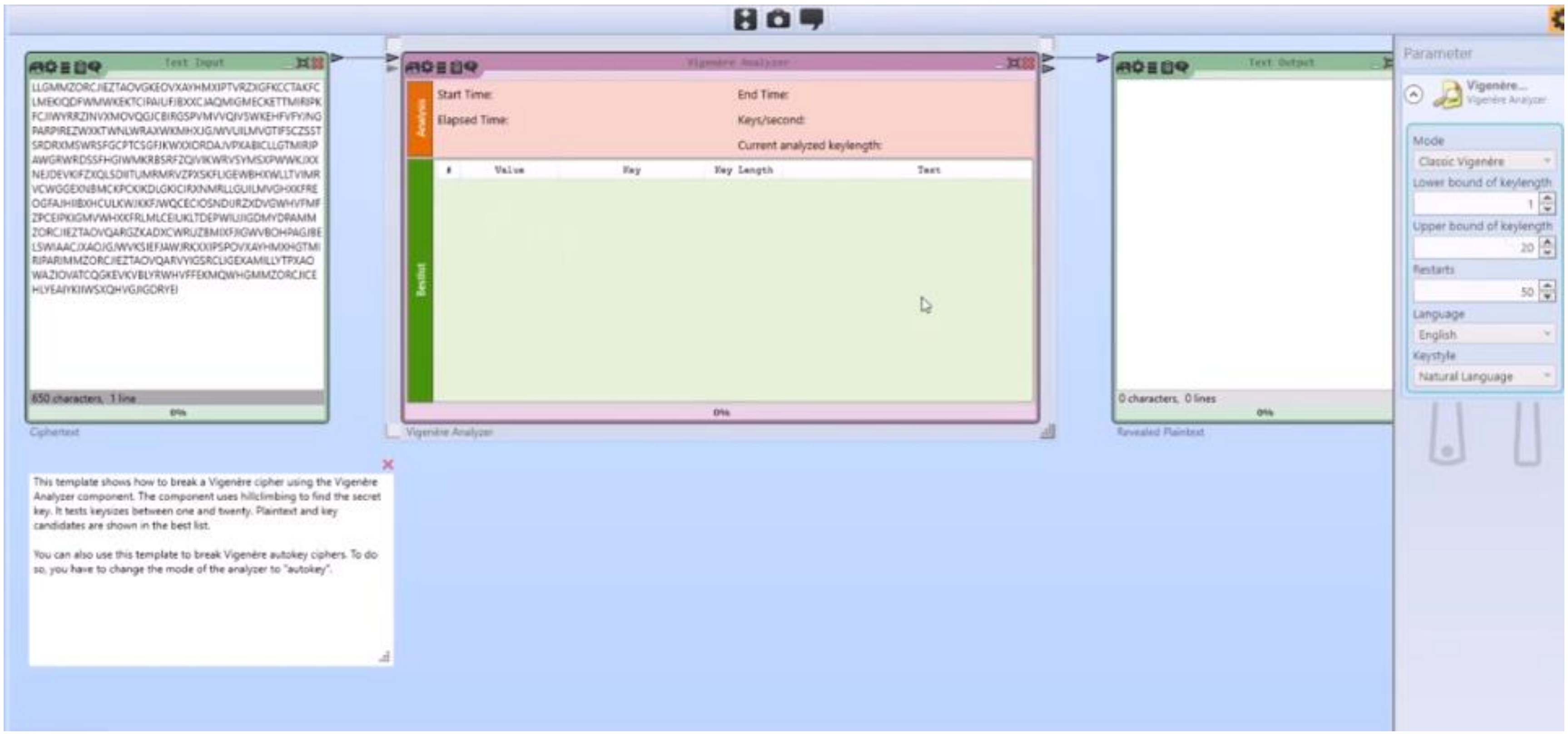
Breaking cipher III

- **Breaking Polyalphabetic substitution (vi)**



Breaking cipher III

- **Breaking Polyalphabetic substitution (vii)**

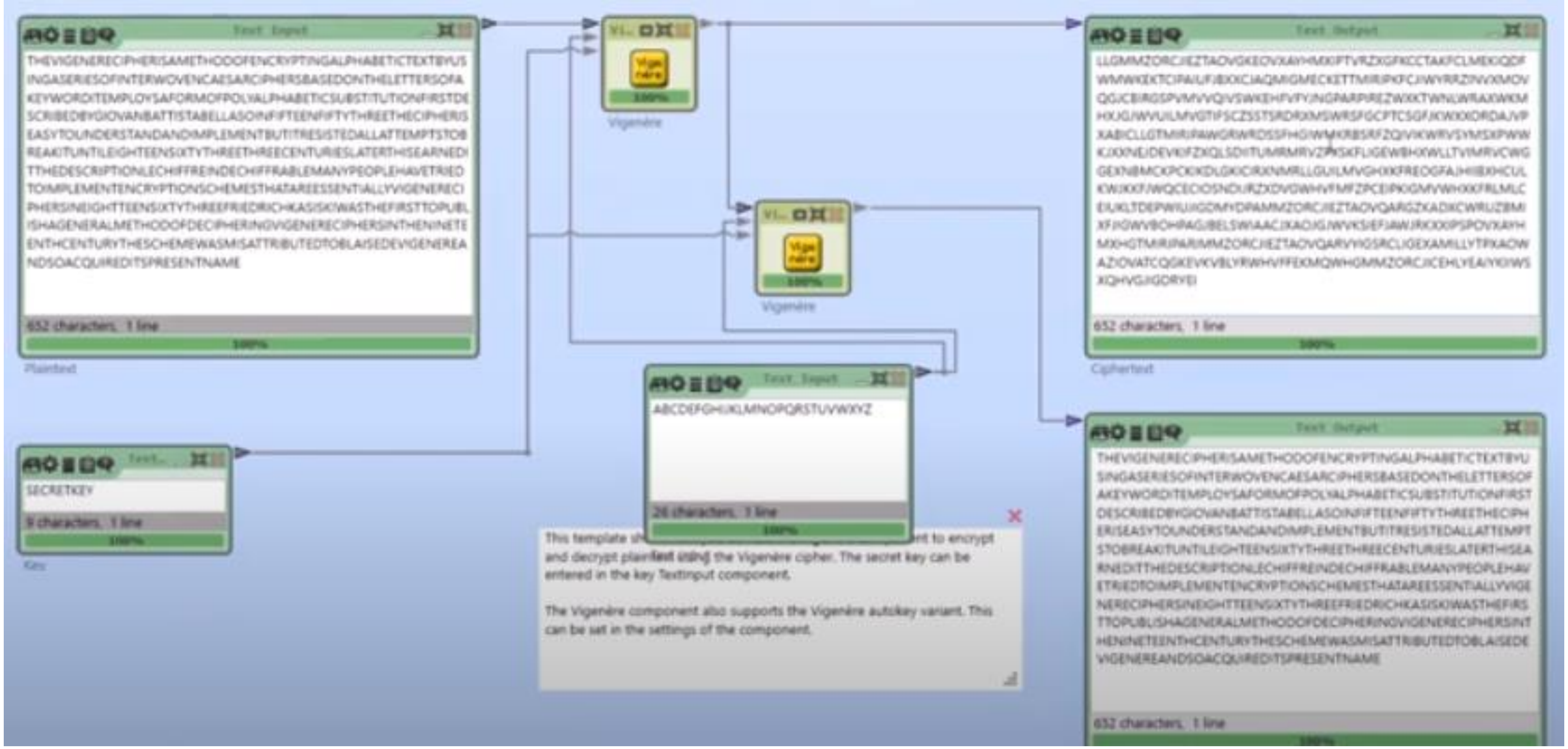


Breaking cipher III

- **Breaking Polyalphabetic substitution (viii)**

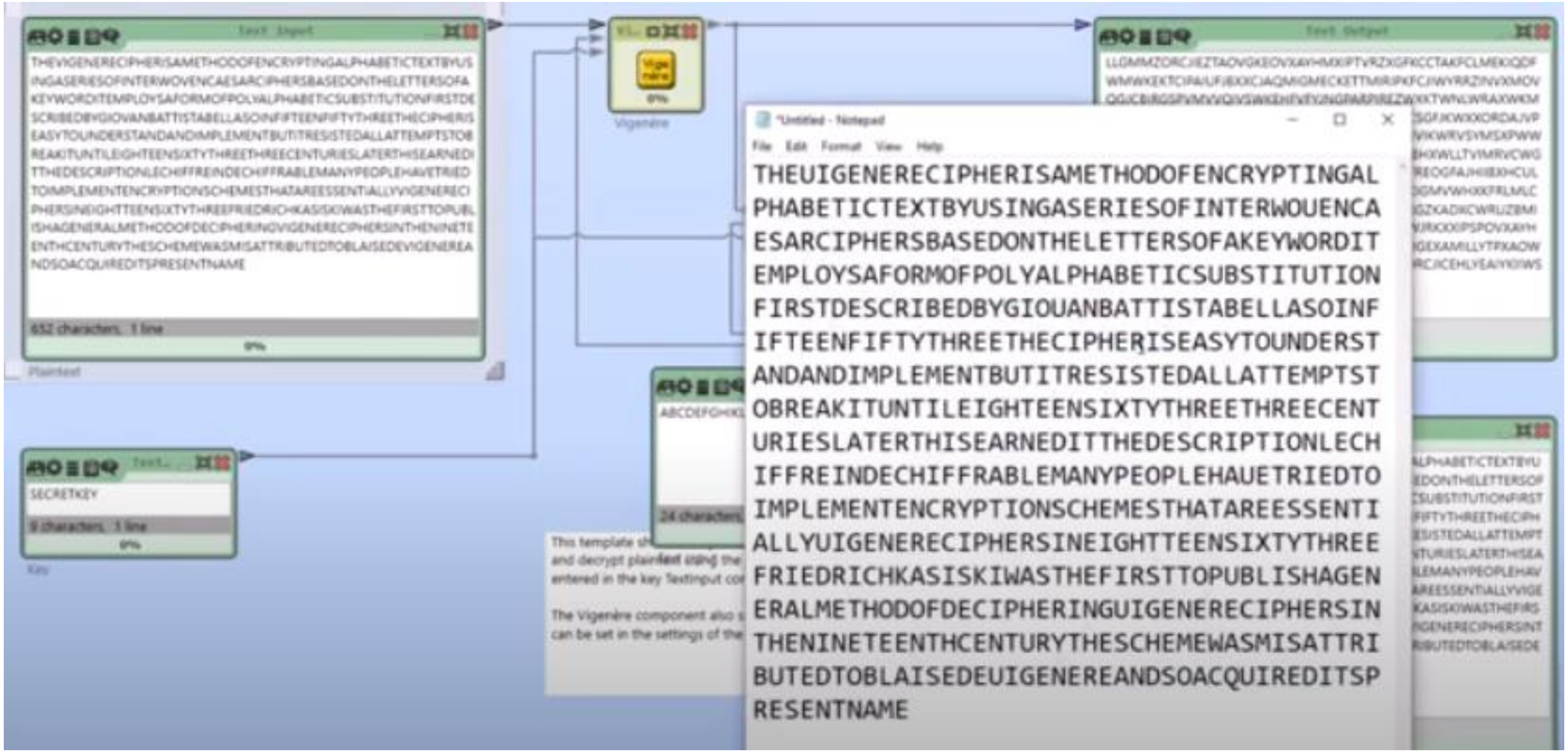
Breaking cipher III

- **Breaking Polyalphabetic substitution (ix)**



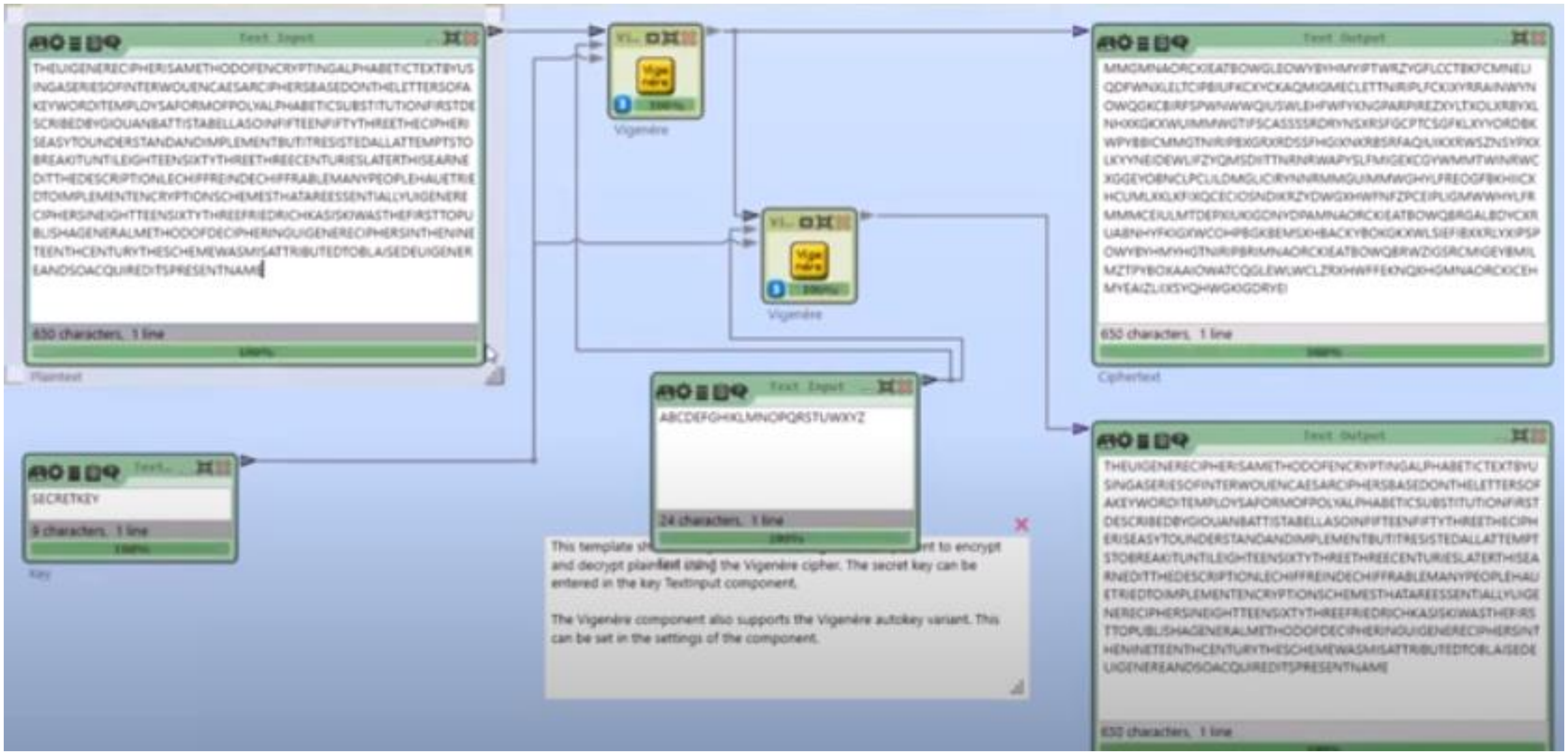
Breaking cipher III

- **Breaking Polyalphabetic substitution (x)**



Breaking cipher III

- **Breaking Polyalphabetic substitution (xi)**



Breaking cipher III

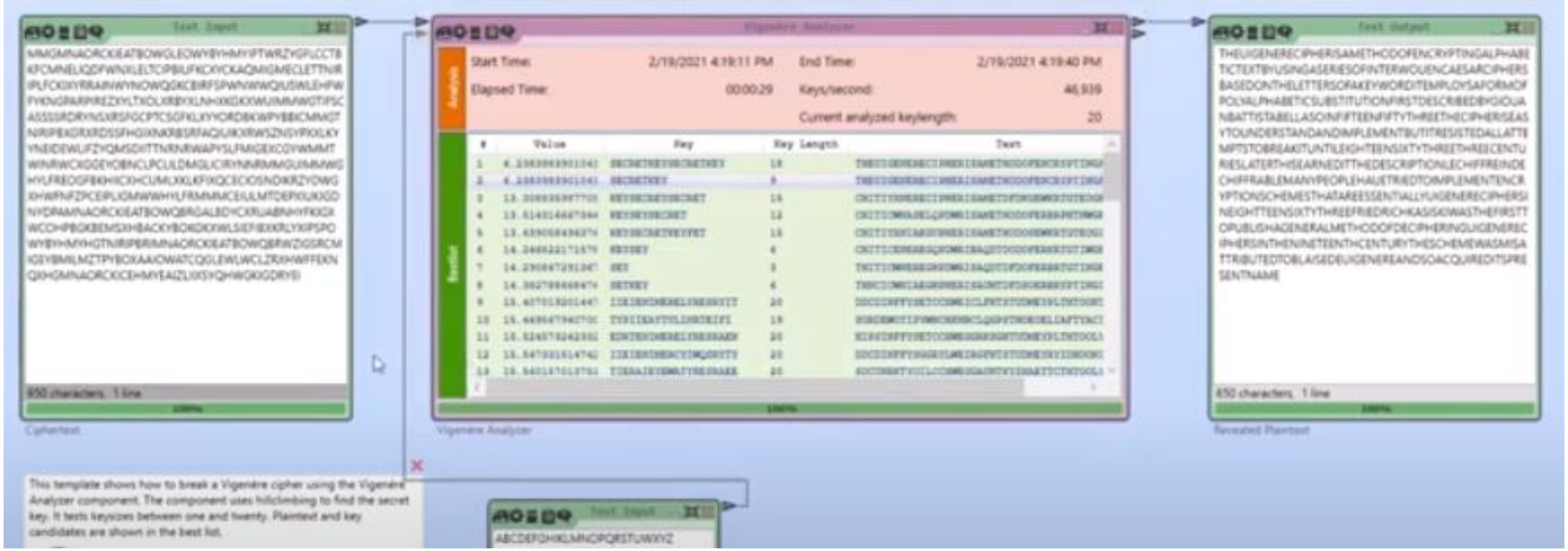
- **Breaking Polyalphabetic substitution (xii)**

Breaking cipher III

- **Breaking Polyalphabetic substitution (xiii)**

Breaking cipher III

- **Breaking Polyalphabetic substitution (xiv)**



Basic Crypto II (LAB III)

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

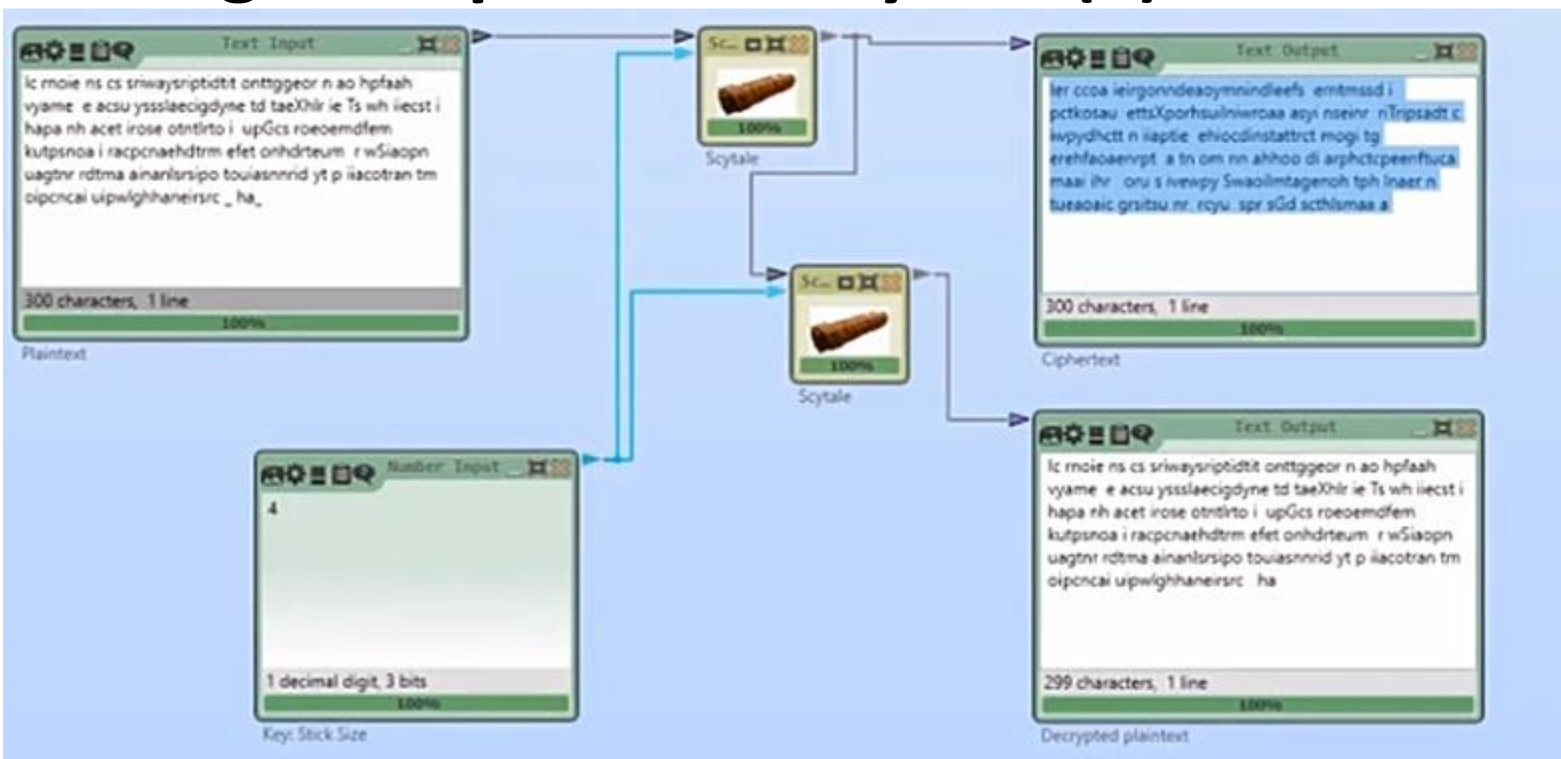
Breaking cipher IV

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 diamter 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
```

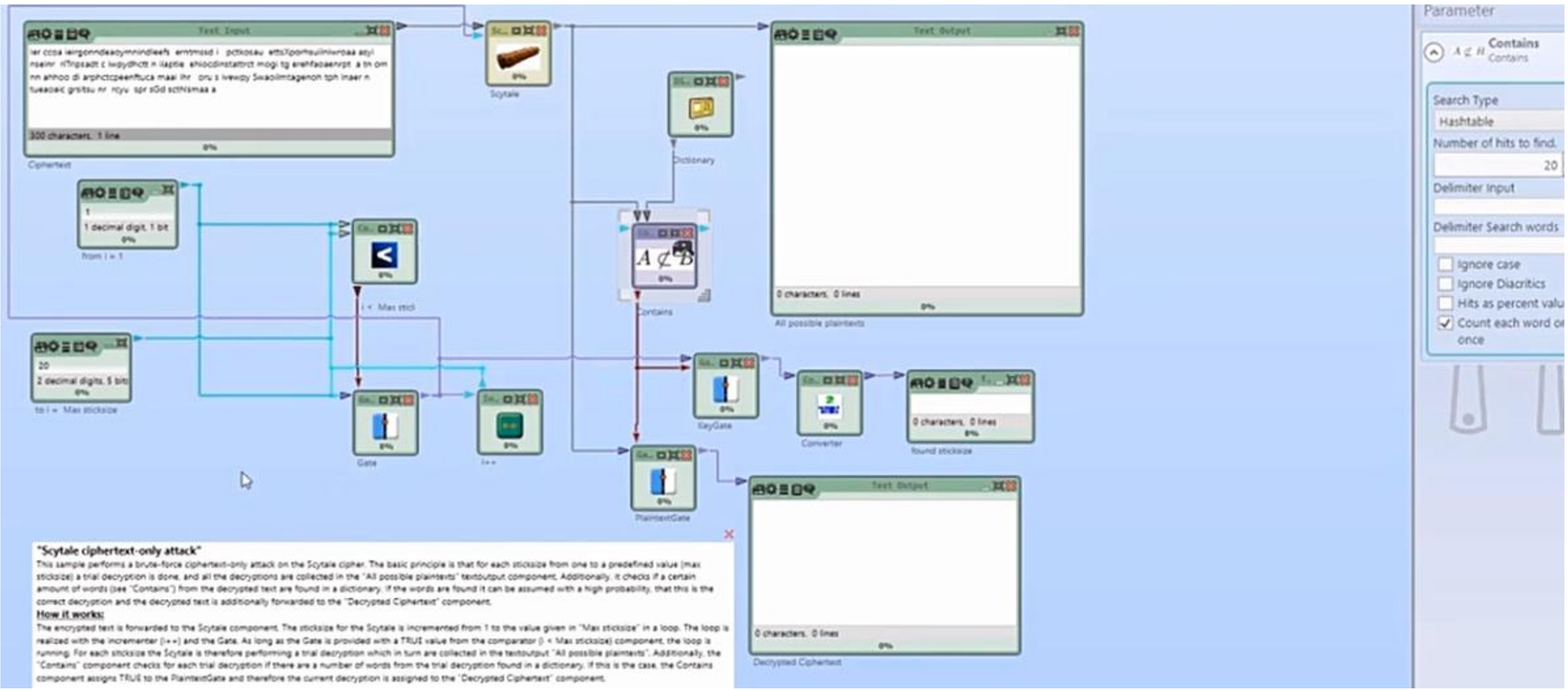
Breaking cipher IV

- **Breaking Transposition. Scytale (ii)**



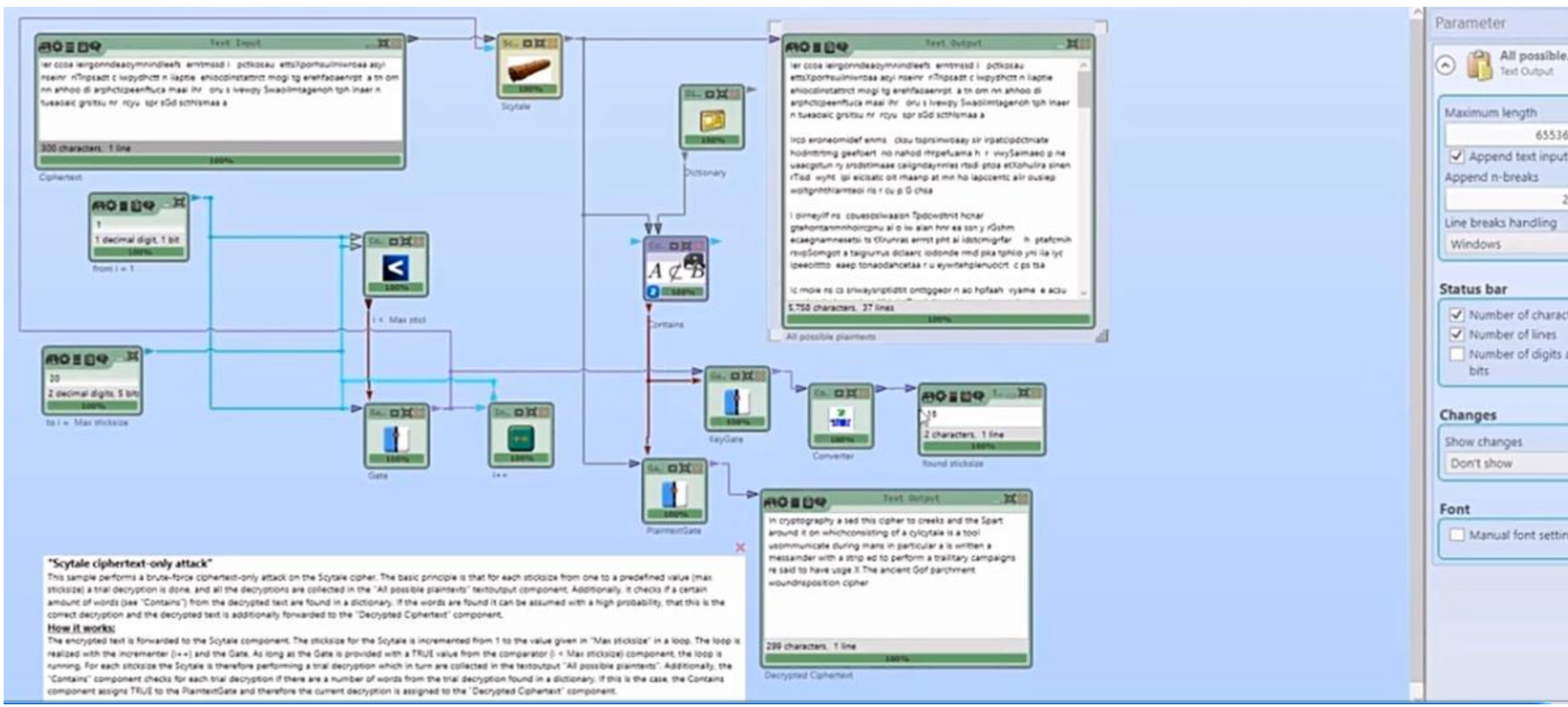
Breaking cipher IV

- **Breaking Transposition. Scytale (iii)**



Breaking cipher IV

- **Breaking Transposition. Scytale (iv)**



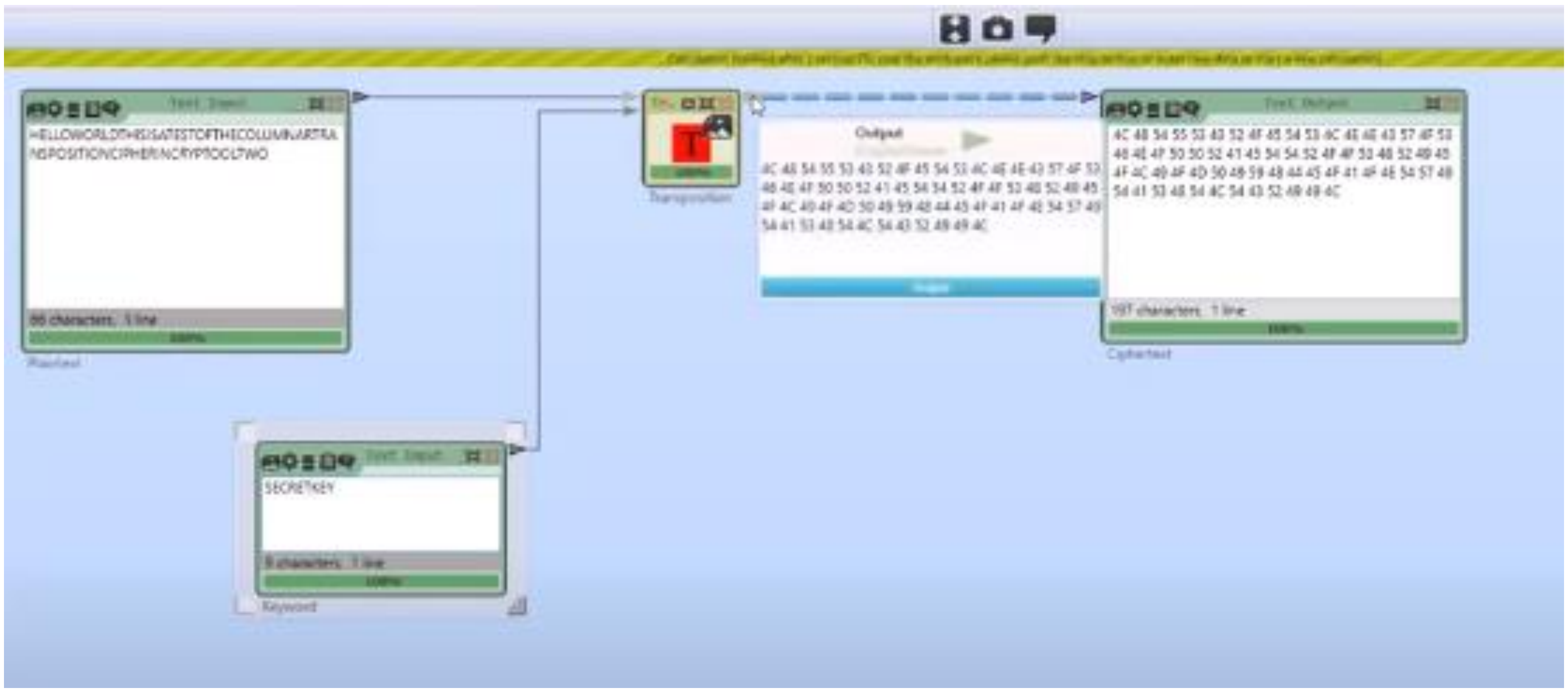
Breaking cipher V

- **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
```

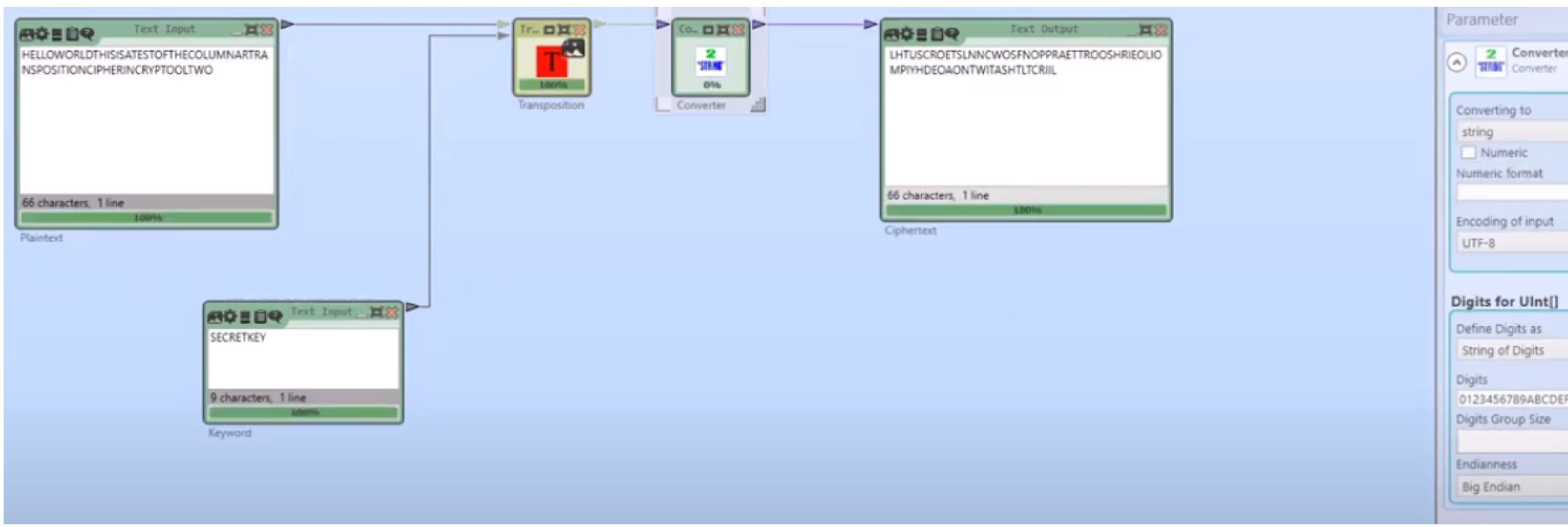
Breaking cipher V

- **Breaking Columnar Transposition (ii)**



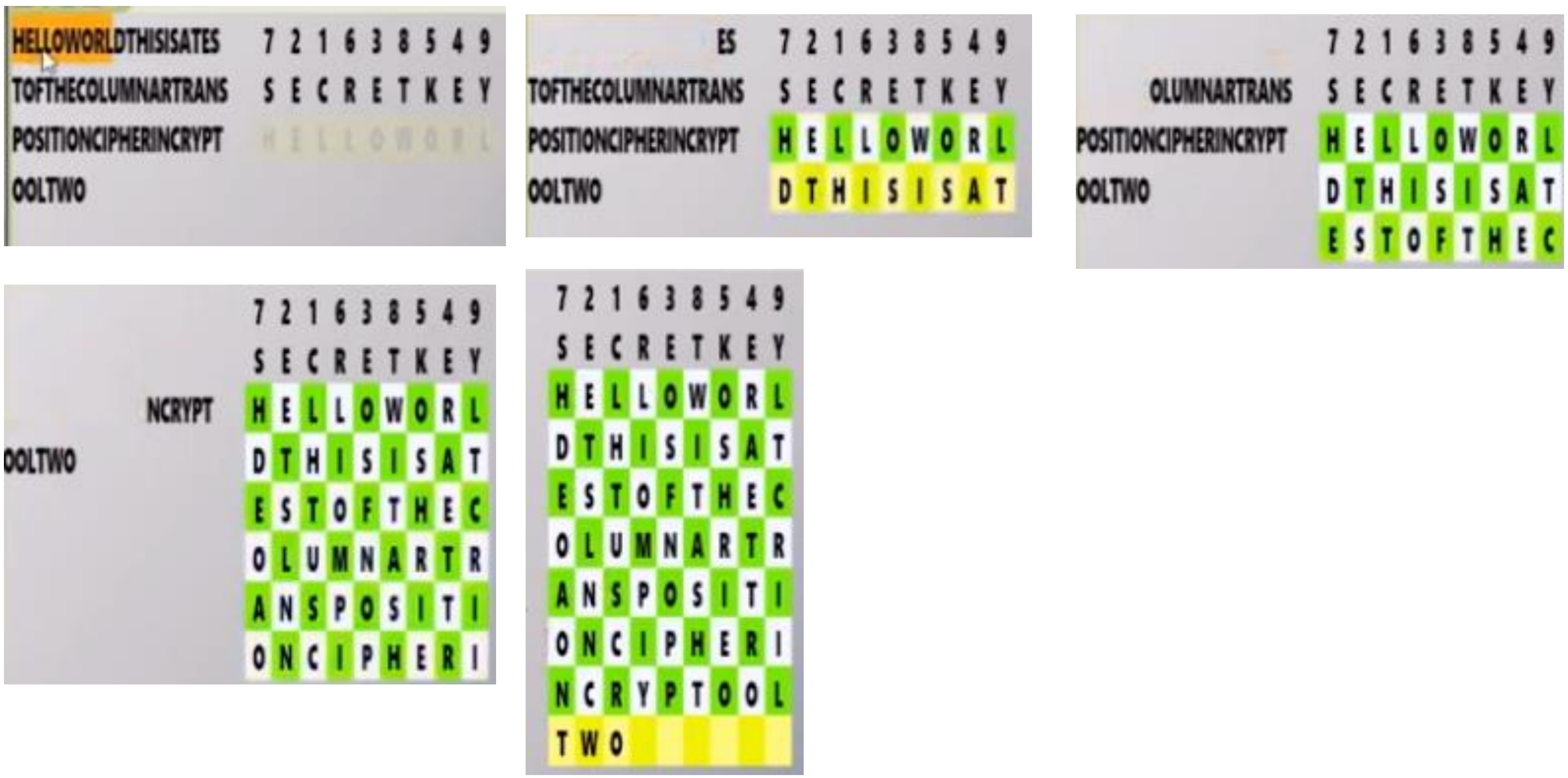
Breaking cipher V

- **Breaking Columnar Transposition (iii)**



Breaking cipher V

- **Breaking Columnar Transposition (iv)**

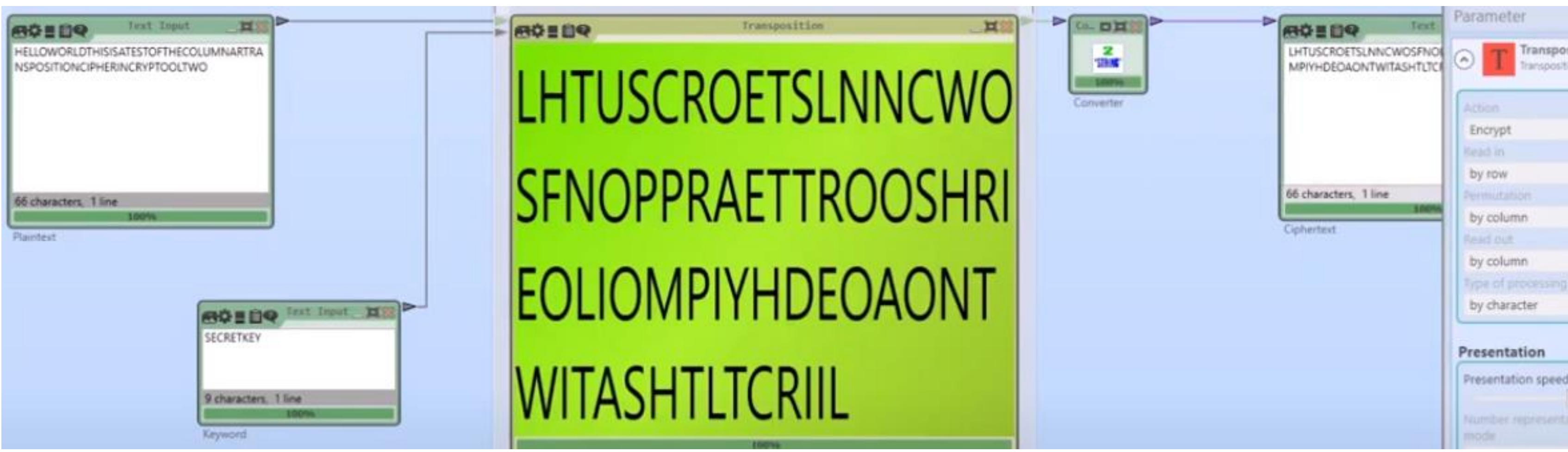


Breaking cipher V

- **Breaking Columnar Transposition (v)**

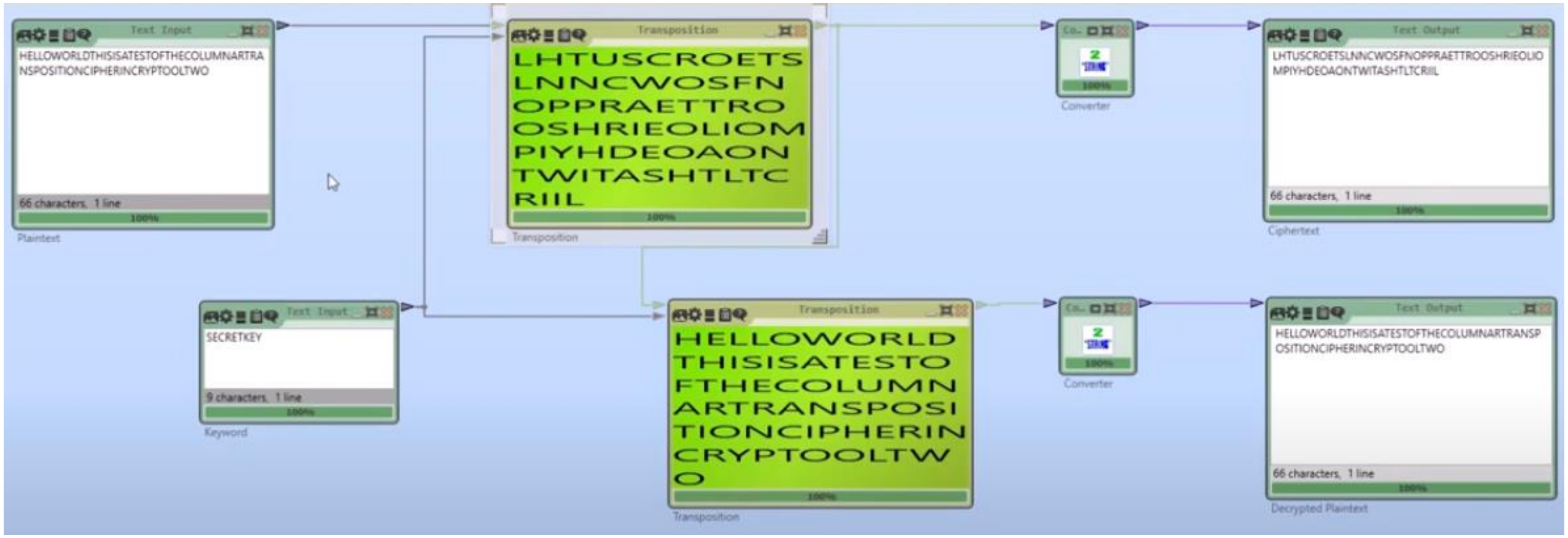
Breaking cipher V

- **Breaking Columnar Transposition (vi)**



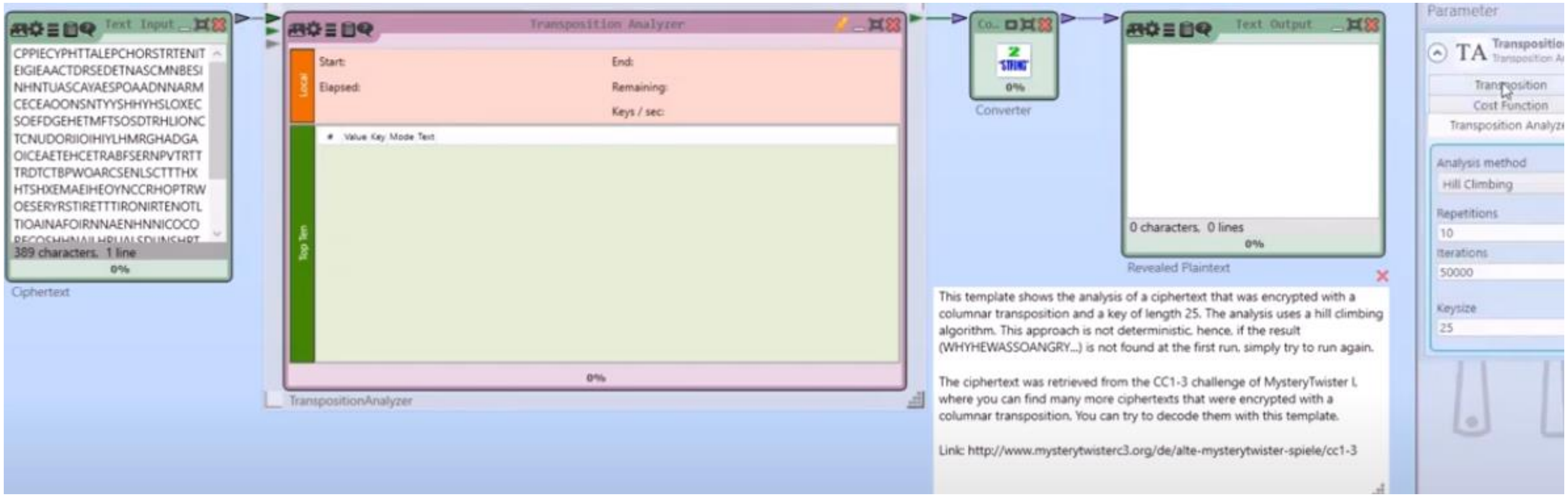
Breaking cipher V

- **Breaking Columnar Transposition (vii)**



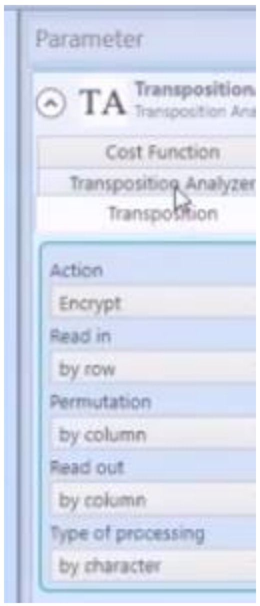
Breaking cipher V

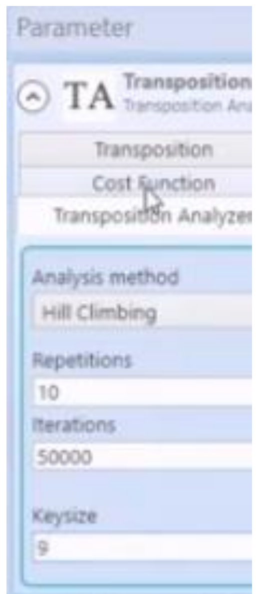
- **Breaking Columnar Transposition (viii)**

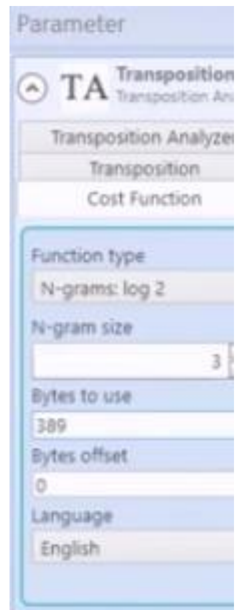


Breaking cipher V

- **Breaking Columnar Transposition (ix)**

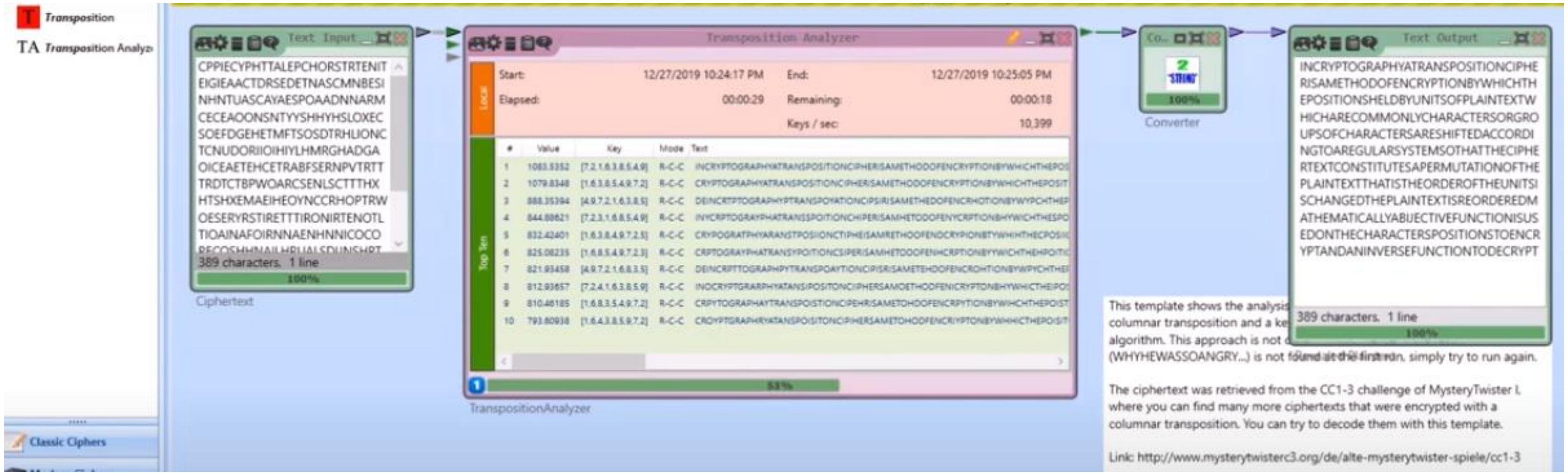






Breaking cipher V

- **Breaking Columnar Transposition (x)**



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:**

    ADFGX $= 25!$ **ADFGVX** $= 36!$
  - **2. Transposition keyspace size (n = 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:**

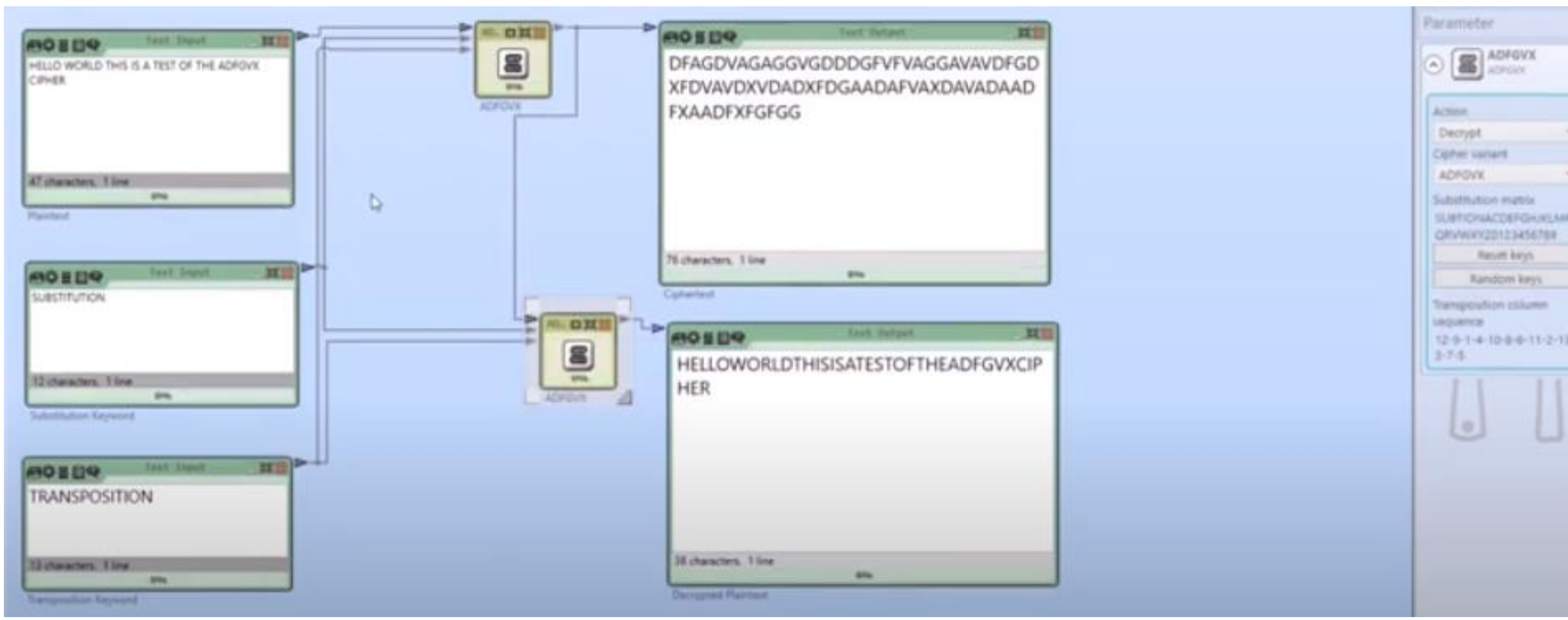
    ADFGX $= 25!$ **ADFGVX** $= 36!$
  - **2. Transposition keyspace size (n = max key length):**

    $= \sum_{i=1}^{n} n!$
  - **Example: transposition key length up to 15 (ADFGVX):**

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

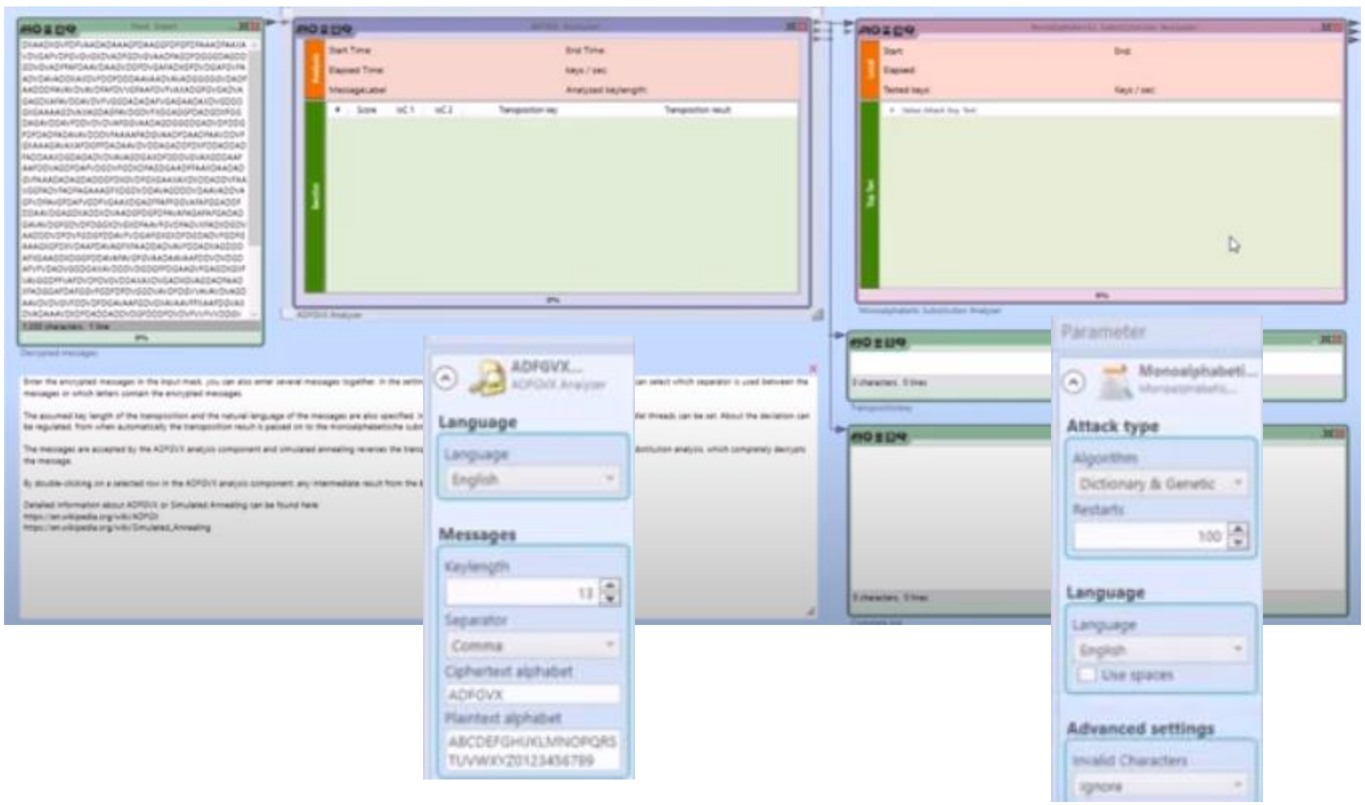
Breaking cipher VI

- **Breaking Mixed cipher (v)**



Breaking cipher VI

- **Breaking Mixed cipher (vi)**



Breaking cipher VI

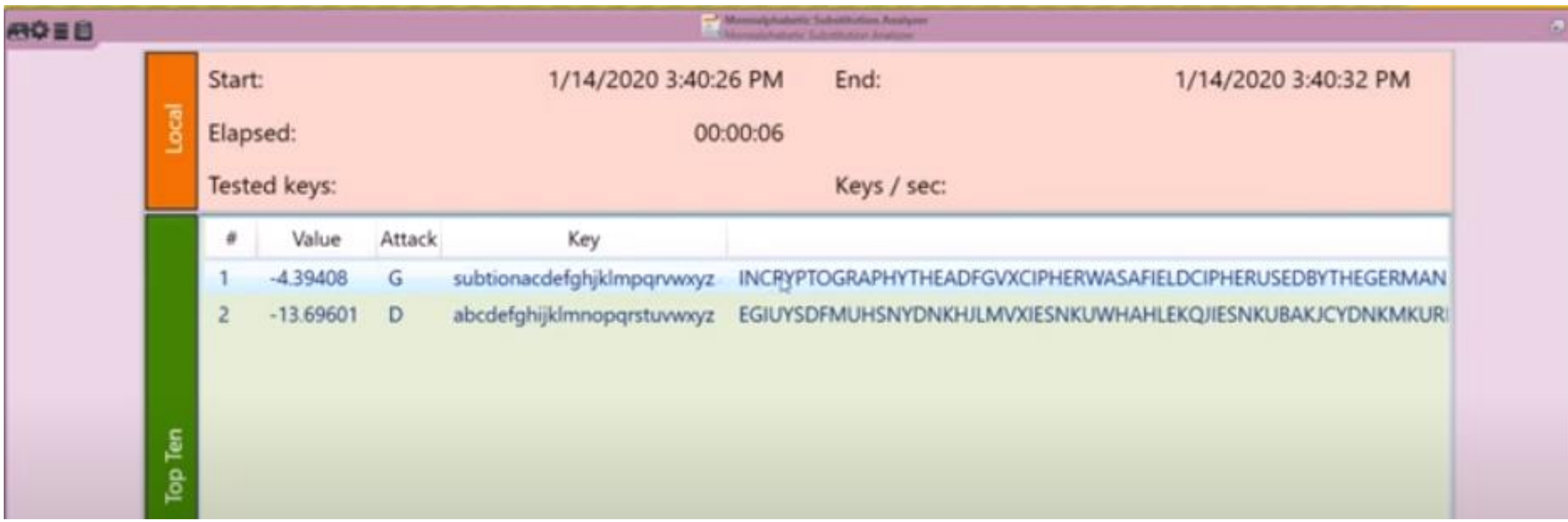
- **Breaking Mixed cipher (vii)**

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

#	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	LIADJHFMBKCGE	EGIS0SDFMUTGNYDNHKJLMVXCKSNKUTKAHLEKKPIESNIV
4	125062	5.93	0.5	LIADJHFMCKBGE	EGISUYDFMUSHNYDNHKJLMVXEISNKUTEGHLEKJQIESNIE
5	113683	5.89	0.44	LIADJMCHFKBGE	EGGUUYDFMSUHNYDNHKJLMVEXISNKTCWGHLEJKQIESOI
6	111494	5.84	0.42	LIAMCDJHFKBGE	EGSIUYDFSMUHNYBJNKJLMEVXISNHEUWGHLJEKQIEUANI
7	99648	5.49	0.37	DELIAMJHFKBGC	ZEGIUYSDFSUHMNYBNKJJLMDXIWSNHUWGEHLKKQJCEUI
8	96528	5.35	0.36	LEDMAIJHFKBGC	BYKIUYVAXAUHMZNANKJLJAPXIWMTHUWGBKLKKQJCCW
9	95314	5.31	0.35	FELMDIJHAKBGC	TA2IIYUDSAXHMHZMNEJLLDPSIWSNTUKGEHKKLQJICENW
10	94710	5.51	0.34	GFLIJHAKBMDEC	NEIIYZSVAXHSAOYNEHNJLPSIDKXNUKHSATKLQKJKENWU

Breaking cipher VI

- **Breaking Mixed cipher (viii)**



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