MysteryTwister

THE CRYPTO CHALLENGE CONTEST

# Bigram Substitution – Part 2

Authors: Lillin Modi, BE

October 2025

# Bigram Substitution – Part 2

Bigram substitution [1] is a monoalphabetic substitution in which the plaintext alphabet consists of 676 letter pairs. Many more details about this method can be found in Part 1 of this challenge series. Further basics on the cryptography used are described in the CrypTool book [3].

## Challenge

In this challenge you are given the ciphertext (CT) and three small pieces of the plaintext (PT) – i.e., you are to carry out a partial known-plaintext attack (KPA).

In reality this happened quite often, because one could assume that certain keywords (cribs) would occur in specific parts of the CT.

You receive the CT in the file:

- `bgs_ciphertext.txt`

The three plaintext cribs are the following strings:

- studying societies
- sociology of knowledge
- patterns

Using these, conduct a partial KPA (known-plaintext attack).

The idea for the English plaintext is taken from the sociology book “The Social Construction of Reality” by Berger and Luckmann (1966) together with a modern complement.

Your task is to find the key and submit it in the single-line representation (CT alphabet only). If you cannot map a PT bigram to a CT bigram, write `??` .

More precisely: submit the list of 676 CT bigrams in the order of the PT bigrams `AA..ZZ` , using `??` for unknowns.

An example submission might look like: `DE GH IK ZR ?? HG ?? ?? SI PQ …`

You may use a space, a tab, or a comma as a separator.

Before submitting, please verify that you list 676 bigram tokens including the `??` , and that—apart from the `??` placeholders—no bigram appears twice.

## Tool

A Python program for bigram substitution is included:

- `2-gram-subst.py` Tasks: generate key, encrypt, decrypt, simple KPA

Use the following command to show options for the “kpa” subcommand:

```
python 2-gram-subst.py kpa --help
```

Since you do not have the full plaintext, you will need to extend the provided tool substantially, or write your own tool.

### Files provided for solvers

- `2-gram-subst.py` Python program for the bigram substitution cipher (encrypt/decrypt, simple KPA, utils)
- `bgs_ciphertext.txt` Ciphertext file (the result of encryption)

## References

[1] Schmeh, K.: Bigram substitution: An old and simple encryption algorithm that is hard to break *Cipherbrain* (2017)

[2] Esslinger, B.: *The CrypTool book: Learning and applying cryptography with CrypTool and SageMath, 2nd edition* (Lehmanns Media, Berlin, 2025).

[3] Esslinger, B.: *Learning and experiencing cryptography with CrypTool and SageMath* (Artech House, Norwood, 2024).