

Lab TCP Student Handbook

Six online cryptanalysis laboratories

<https://lab-tcp-cryptanalysis-classroom.kbtu-6877.chatgpt.site>

How to start

Open the classroom link in a browser. Select a laboratory and your assigned variant, choose an exercise and click Start / resume variant. All ciphertexts are already on the site; you do not need to download or upload anything.

What is saved

Inputs, parameters, experiment results and your conclusion save automatically in this browser on this device. Changing exercises keeps separate attempts. Reopening the same site restores the last attempt. Clearing browser data or using another device will not preserve it.

How to use this handbook

Each chapter contains a small, fully explained demonstration and a procedure for your own variant. Demonstration messages and keys are teaching examples, not answers to assessed variants. Use the same reasoning with your assigned ciphertext.

Alphabet

The English alphabet has 27 symbols: A=0, B=1, ..., Z=25 and space=26. Spaces are encrypted. Use uppercase A-Z and spaces. Normalize text only for your own plaintext. Position controls start at 1; technical offset and step fields in JSON start at 0.

Variants

Labs 1-5 have variants 1-30. Lab 4 has two exercise files; Lab 5 has three. Lab 6 has variants 1-25. Your instructor assigns the number. All 265 exercise files are available on the website.

An experiment cycle

State a hypothesis, choose an action, inspect the result, accept or reject the hypothesis with a reason, then record the conclusion. Review experiment log shows the parameters and results without downloading a file.

Chapters

Lab	Topic
1	Monoalphabetic substitution
2	Polyalphabetic substitution
3	Multiple-key cipher
4	Probable-word analysis
5	Transposition ciphers
6	Additive stream ciphers

Monoalphabetic substitution

Objective. Recover a one-to-one substitution by combining symbol counts and word patterns.

The method

Every occurrence of a plaintext symbol is replaced by the same ciphertext symbol. Counts stay unchanged. For a symbol with n occurrences in a message of length N , its relative frequency is n/N . Compare frequencies as evidence, not as a guaranteed dictionary.

Worked example

Plaintext

MEET ME AT THE LIBRARY

Ciphertext

DTTZ DT QZ ZIT SOWKQKN

A useful observation

There are 22 symbols. T occurs 4 times: $4/22 = 18.18\%$. Try $T=E$. Z occurs 3 times; the short word ZIT suggests THE, so try $Z=T$ and $I=H$.

Partial mapping

$D=M$; $T=E$; $Z=T$; $I=H$; $S=L$; $O=I$; $W=B$; $K=R$; $Q=A$; $N=Y$; [space]=[space]

Recovered message

MEET ME AT THE LIBRARY

Your variant - procedure

1. Select Lab 1 and your assigned variant. Click Start / resume variant. Look at the two Statistics columns. They are ranked independently; rows are not established symbol pairs.
2. Choose a ciphertext symbol and a candidate under Replace with. Click Replace symbol. Unknown symbols are shown as dots. For a space use the space option; in the mapping field it is written as the visible symbol shown in the interface.
3. Expand the mapping using likely short words and repeated patterns. Record two hypotheses that fail and explain the contradictions. The application rejects mappings that are not one-to-one.
4. For a reverse check, keep a copy of your ciphertext, paste the recovered plaintext into the input and click Reverse substitution check. Compare the complete result with the original. Use Start / resume variant to return to your saved attempt, not to erase it.

Evidence to record

- A frequency calculation for one symbol and a short manual coincidence-index calculation.
- Recovered plaintext and mappings for all symbols that actually occur. Explain why absent symbols cannot determine a unique complete key.

Common mistake

Do not automatically map the most frequent ciphertext symbol to E. In these tasks a space is also a symbol and may be more frequent than any letter.

Polyalphabetic substitution

Objective. Find a Vigenere period and recover one shift for each periodic group.

The method

Use A=0, B=1, ..., Z=25, space=26. Encryption is $c=(m+k) \bmod 27$; decryption is $m=(c-k) \bmod 27$. A key repeats over every character, including spaces. A candidate period p partitions the ciphertext into p groups. Each group uses a single shift.

Worked example

Demo plaintext

MEET ME AT THE OLD BRIDGE

Demo key

LEMON

Demo ciphertext

XIQGMXILQFKXTSMZPPNOBMPUR

First symbol

M=12, L=11, so $(12+11) \bmod 27=23$, which is X. Reverse check: $(23-11) \bmod 27=12$, which is M.

A wraparound example

At position 4, T=19 and O=14. Their sum is 33; $33 \bmod 27=6$, so the ciphertext symbol is G.

The space matters

At position 5, space=26 and N=13. The result is 12=M. Skipping the space would shift every later key position.

Your variant - procedure

1. Use Frequencies and coincidence index, then Compare periods. Start with 1-20. Keep at least three plausible hypotheses; multiples of the true period may also have high scores.
2. Enter a candidate period and select each Periodic group in turn. Click Group frequencies. Use common letters and spaces to propose a shift $k=(c-m) \bmod 27$.
3. Convert the recovered shifts back into letters and enter the Vigenere key. Click Decrypt. Inspect the entire text, not just the first word.
4. Copy the plaintext into the input with the same key and click Encrypt your plaintext. The complete ciphertext must match. Explain why the minimal repeating key, rather than one of its multiples, is the best description.

Evidence to record

- A table of candidate periods and the reason for your selected period.
- One manual modular calculation, recovered key, plaintext and exact reverse check.

Common mistake

Do not use modulo 26. This classroom alphabet contains 27 symbols because space participates in encryption.

Multiple-key cipher

Objective. Recover a composite periodic key and investigate the primary repeating keys.

The method

Two periodic keys add at each position: $c=(m+k_1+k_2) \bmod 27$. Their composite period divides $\text{LCM}(u,v)$, where u and v are the primary lengths. It can be smaller. The primary keys are not unique: add a constant shift to one and subtract it from the other.

Worked example

Demo primary keys

GO (length 2) and SUN (length 3). $\text{LCM}(2,3)=6$.

Key streams

```
G O G O G O
S U N S U N
```

Composite key

YHTF A

First composite symbol

$G=6$ and $S=18$, so $6+18=24=Y$. At position 5, $G=6$ and $U=20$ give 26, which is a real space.

Demo ciphertext

JLXYZMBGTYZTELSQHBOHJC

Recovered plaintext

MEET ME AT THE LIBRARY

Your variant - procedure

1. Run Kasiski examination. Choose repeated trigrams and calculate distances between their positions. Use common divisors as period hypotheses; accidental repeats can give unrelated distances.
2. Compare candidate periods and inspect Group frequencies. Recover the composite key using the same method as Lab 2. A composite key can contain spaces and need not be a word.
3. For the English assessed variants, the primary key lengths are 5 and 6. First try the period analysis without using this hint. Enter the recovered composite key and proposed lengths, then click Decompose into two keys.
4. For coprime lengths, click Enumerate primary-key shifts and inspect the 27 equivalent pairs. Justify choosing meaningful primary words. Enter the words separated by a semicolon, build the composite key, decrypt and reverse-check.

Evidence to record

- At least three repeat-distance calculations and a supported period hypothesis.
- Composite key, primary-key lengths, a valid pair of primary keys and an explanation of ambiguity.

Common mistake

A program-generated decomposition is one valid numerical solution. Its first pair need not form meaningful words, and the source data alone may not identify a unique word pair.

Probable-word analysis

Objective. Use a guessed plaintext fragment to reveal a key fragment, then validate the hypothesis globally.

The method

A probable word, or crib, gives $k=(c-m) \bmod 27$ at its proposed location. A short periodic key must be consistent wherever the same key position recurs. A running key has no periodicity requirement and must cover the entire message.

Worked example

Short-key demo ciphertext

EHXGWEQGUSSOUDWLZ AULT LLBGVW

Crib and position

Assume THE KEY at position 1, with period 4.

Derived key fragment

MATHMAT

Consistent short key

The fragment MATHMAT repeats consistently with period 4. Enter MATH and decrypt.

Recovered plaintext

THE KEY IS HIDDEN IN THE BOOK

Running-key contrast

Ciphertext: JLECLTWDHDEFZVOEOUSN

Crib THE MAP at position 1 reveals key fragment READ TH. With period 0, this is evidence about a running text, not a repeating seven-letter key.

Your variant - procedure

1. Select word_v for the short-key exercise. Propose at least three words appropriate to the message topic, and justify them before testing.
2. Enter a probable word, a 1-based position and a candidate period. Click Test word at position or Test all positions. Calculate at least the first three recovered key symbols manually.
3. Use consistency and meaningful key fragments to refine your hypothesis. Enter a full key and click Decrypt. Use Encrypt to verify for the reverse check. A consistent short crib can still be wrong.
4. For word_rk choose period 0. Extend both plaintext and running-key fragments. Your key must cover every character, including spaces; distinguish supported letters from assumptions.

Evidence to record

- Your crib choices, tested positions, derived fragments and two rejected hypotheses.
- Complete result for the short-key file and a supported reconstruction for the running-key file, with reverse checks.

Common mistake

The crib is assumed, not known. A readable fragment at the guessed location alone is not proof that the rest of the message is correct.

Transposition ciphers

Objective. Recover symbol order in a block, along a graph path, and in a table.

The method

A transposition changes order, not symbol counts. Block and Hamiltonian modes use $C_i = M_{(\text{key}_i)}$. Table mode uses the inverse convention $C_{(\text{key}_i)} = M_i$ inside each column. Keys contain each position from 1 to N exactly once.

Worked example

Block demo

Plaintext: MEETMEATNOONXXX

Key: 3 1 5 4 2

Ciphertext: EMMTETEONAXOXXN

One block by hand

MEETM -> EMMTE. Read plaintext positions 3,1,5,4,2. To invert, place ciphertext characters back into those positions.

Graph demo

Graph 1, path 1 2 3 6 4 5. Every consecutive pair is an edge, and all six vertices occur once. SECRET becomes SECTRE.

Table demo

Plaintext MEETMEATNOON is written down columns with 4 rows. Natural rows are MMN / EEO / EAO / TTN. Key 3 1 4 2 sends original row 1 to destination row 3, row 2 to row 1, row 3 to row 4 and row 4 to row 2.

Table ciphertext

ETMEETMAONNO

Your variant - procedure

1. For trans_simple, use Text-length divisors and justify at least two block-size hypotheses. Search candidate keys for sizes 2-8; a probable word can filter the candidates.
2. For trans_gam, select graph $g = 1 + ((\text{variant} - 1) \bmod 5)$. Open Hamiltonian graphs, record its adjacency matrix and test only Hamiltonian paths. Compare the number of graph paths with N!.
3. For trans_tab, select Table transposition and use 8 rows for the assessed 8 by 8 message. Display table rows and inspect Row adjacency scores before deciding the order.
4. For every mode, enter the key as numbers separated by spaces and click Decrypt. Reverse-encrypt with the same mode. Preserve any explicit X padding when comparing strings.

Evidence to record

- All three exercise files: recovered text, key, reasoning and exact reverse check.
- A manually decrypted block, graph adjacency matrix, valid path and explanation of the two key conventions.

Common mistake

A high bigram score is a ranking heuristic, not a proof. The search returns only its best 30 candidates. Do not silently remove padding before reverse verification.

Additive stream ciphers

Objective. Recover a seven-stage linear recurrence from a probable bigram and decrypt a 16-byte message.

The method

XOR operates bit by bit: $0 \text{ XOR } 0=0$, $0 \text{ XOR } 1=1$, $1 \text{ XOR } 0=1$ and $1 \text{ XOR } 1=0$. Encryption is $C=P \text{ XOR } G$, so $G=C \text{ XOR } P$. English characters use ASCII bytes, most significant bit first. A 16-byte message produces 128 bits.

Worked example

Demo plaintext

SEND THE MESSAGE

First two ciphertext bytes

1100010000110110

Guess SE at position 1

S is 01010011 and E is 01000101. Their concatenation is 0101001101000101.

Known keystream fragment

10010111101110011

Recovered recurrence

$L=7$; $C=[1,0,0,0,0,0,1,1]$. In this convention, $s_n=s_{(n-6)} \text{ XOR } s_{(n-7)}$. Initial output bits: 1001011.

What must be checked

Extend the recurrence to all 128 bits. XOR with the ciphertext and recover SEND THE MESSAGE. The full message, not only SE, must be meaningful and reverse-check exactly.

Your variant - procedure

1. Validate VAR structure. Its four decimal 32-bit words are another representation of the same ciphertext, not the secret key.
2. Begin with $L=7$. Try probable bigrams such as TH, HE, IN, ER, AN, RE, ON, AT, EN or ND at character positions. Other pairs may be necessary. Manually XOR 16 bits for one hypothesis.
3. Work through Berlekamp-Massey by hand, then compare with Keystream and Berlekamp-Massey. Record each discrepancy, L and coefficient vector. At least $2L$ bits are normally required.
4. Form linear equations over $GF(2)$ and compare. Solve linear equations over $GF(2)$. Rewind if your crib was not at position 1. Preserve the final spaces used to fill the 16-byte assessed messages and verify all 128 bits.

Evidence to record

- Manual XOR, Berlekamp-Massey trace, $GF(2)$ system and recovered recurrence.
- Initial sequence, full keystream, recovered plaintext and a bit-for-bit reverse check.

Common mistake

The coefficient vector defines a recurrence, not an unspecified drawing of register taps. State the convention explicitly. UTF-8 must not be used for byte calculations on original Russian material.

Manual Berlekamp-Massey record

Demonstration keystream: 1001011101110011

The step number n below starts at 0. C is listed as $[C_0, C_1, \dots, C_L]$, with $C_0=1$. A nonzero discrepancy triggers a connection-polynomial update. The final recurrence is $s_n = s_{n-6} \text{ XOR } s_{n-7}$.

n	bit	d	L	C
0	1	1	1	[1, 1]
1	0	1	1	[1, 0]
2	0	0	1	[1, 0]
3	1	1	3	[1, 0, 0, 1]
4	0	0	3	[1, 0, 0, 1]
5	1	1	3	[1, 0, 1, 1]
6	1	0	3	[1, 0, 1, 1]
7	1	0	3	[1, 0, 1, 1]
8	0	0	3	[1, 0, 1, 1]
9	1	1	7	[1, 0, 1, 1, 0, 0, 1, 0]
10	1	0	7	[1, 0, 1, 1, 0, 0, 1, 0]
11	1	1	7	[1, 0, 0, 1, 1, 1, 1, 0]
12	0	1	7	[1, 0, 0, 0, 1, 0, 0, 0]
13	0	1	7	[1, 0, 0, 0, 0, 0, 1, 1]
14	1	0	7	[1, 0, 0, 0, 0, 0, 1, 1]
15	1	0	7	[1, 0, 0, 0, 0, 0, 1, 1]

For your own variant, recreate this table by hand before comparing with the program. Compare the final vector with the GF(2) solution and verify the entire message.

Report and troubleshooting

Your report

Enter your student label or group if required, state the lab and variant, and write your recovered key and plaintext in the conclusion. Explain at least two rejected hypotheses and one accepted hypothesis. Include the manual calculations requested in the chapter.

How to verify

Whenever possible, apply the inverse operation to your recovered result and reproduce every symbol or bit of the original ciphertext. Keep spaces and padding. A good-looking beginning is weaker evidence than a complete exact reverse check.

Sharing work

The current version does not submit work to an instructor automatically. Follow the submission method your instructor announces. If requested, the optional Download report and protocol buttons provide a portable copy; Show protocol text is a manual backup alternative.

If something goes wrong

If you see a storage warning, keep the tab open and copy the protocol text. If a mapping or key is rejected, read the error and check the alphabet, key uniqueness, length, graph constraints and spaces. Reset this attempt clears only the selected attempt after confirmation.

Suggested marking rubric

Criterion	Points out of 20
Method and manual calculation	4
Reasoned hypothesis testing	6
Correct result and reverse check	6
Conclusion and explanation	4

Demonstration messages are synthetic teaching examples. They are not assessed variant answers.