Classical encryption algorithms

A shared numerical model

Use the mapping `A = 0, B = 1, ..., Z = 25`. Addition encrypts; subtraction decrypts. Modulo 26 wraps results into the alphabet.

```
encrypted = (plaintext + shift) MOD 26
plaintext = (encrypted - shift + 26) MOD 26
```

For a shift of 5, `X = 23` encrypts to `(23 + 5) MOD 26 = 2`, which is C.

> **State your convention:** Cipher questions must use one consistent letter-number mapping and direction. A different stated convention may produce different intermediate numbers.

Modulo practice

Determine the encrypted letters for A, M and Z using shift 4.

Check your answer

A → E, M → Q, Z → D.

Caesar cipher

A Caesar cipher applies one fixed shift to every plaintext letter.

Plaintext	C	O	D	E
Number	2	14	3	4
+3, mod 26	5	17	6	7
Ciphertext	F	R	G	H

The ciphertext is `FRGH`.

Decrypt

Determine the plaintext for `KHOOR` with Caesar shift 3.

Check your answer

Subtracting 3 gives `HELLO`.

Evaluate Caesar security

There are only 25 useful non-zero shifts. An attacker can try them all quickly. The same plaintext letter always becomes the same ciphertext letter, so language patterns and letter frequency remain visible.

> **Security judgment:** Caesar is useful for learning substitution and modulo arithmetic, not for protecting modern sensitive data.

Vigenère cipher

Vigenère repeats a keyword. Each key letter supplies a different shift using A = 0.

Plaintext	C	A	T	C	H
Repeated key	D	O	G	D	O
Shift	3	14	6	3	14

Ciphertext	F	O	Z	F	V

`CATCH` with key `DOG` becomes `FOZFV`.

Encrypt with Vigenère

Use key `KEY` to encrypt `DATA`. **Determine** the repeated key and ciphertext.

Check your answer

Repeated key: `KEYK`; shifts: 10, 4, 24, 10. D(3)+K(10)=N(13), A(0)+E(4)=E(4), T(19)+Y(24)=R(17), and A(0)+K(10)=K(10). The ciphertext is `NERK`.

Evaluate repeated-key security

Vigenère hides simple single-letter frequency better than Caesar because the same plaintext letter can encrypt differently at different key positions. However, a repeated keyword creates a period. With enough ciphertext, attackers can investigate repetition, infer key length and apply frequency analysis by position.

A longer keyword increases possibilities, but a predictable word and repeated use remain weaknesses.

Gronsfeld cipher

Gronsfeld uses a repeating sequence of decimal digits as shifts. It works like a Vigenère cipher whose shifts are limited to 0–9.

Plaintext	M	A	P	S
Key	2	0	4	2
Ciphertext	O	A	T	U

With repeating key `204`, `MAPS` becomes `OATU`.

Gronsfeld varies shifts but uses only ten possible shift values and repeats the digit key, so it remains vulnerable to period and frequency analysis.

One-time pad

A one-time pad uses a key that is:

- truly random
- at least as long as the plaintext
- kept secret
- used once only.

Encryption can still use modular addition, but the key does not repeat. Under the required conditions, the ciphertext does not reveal a unique plaintext because every same-length plaintext has a possible key.

> **The conditions matter:** a repeated, predictable or reused pad is not a secure one-time pad. Key generation, secure distribution and storage are major practical challenges.

Why reuse fails

Two messages use the same pad. **Explain** the resulting weakness.

Check your answer

Combining the two ciphertexts can cancel the shared key relationship and expose relationships between the plaintexts. Language patterns or a known part of one message can then help recover the other. Reuse removes the defining one-time condition.

Compare the algorithms

Algorithm	Key pattern	Main weakness

Caesar	One repeated shift	Tiny key space and preserved frequency
Vigenère	Repeated letter-key shifts 0–25	Repeated period can be analysed
Gronsfeld	Repeated digit shifts 0–9	Repeated period and restricted shifts
One-time pad	Random, message-length, never reused	Secure key creation, distribution and handling

Develop an encryption algorithm

```
BEGIN
INPUT plaintext
INPUT shifts
SET ciphertext = ""
SET keyIndex = 0
```

FOR EACH character IN plaintext DO IF character IS A LETTER THEN SET plainNumber = LetterToNumber(UPPERCASE(character)) SET shift = shifts[keyIndex MOD LENGTH(shifts)] SET cipherNumber = (plainNumber + shift) MOD 26 SET ciphertext = ciphertext + NumberToLetter(cipherNumber) SET keyIndex = keyIndex + 1 ELSE SET ciphertext = ciphertext + character ENDIF ENDFOR

OUTPUT ciphertext END

One design choice is whether spaces and punctuation advance the key. State and apply the choice consistently. The example advances only for letters.

Desk check the algorithm

Use plaintext `AZ!` and shifts `[2, 4]`.

Character	Key index	Shift	Output
A	0	2	C
Z	1	4	D

Character	Key index	Shift	Output
!	2	Not used	!

The output is `CD!`.

Independent practice: compare keys

Use plaintext `NETWORK`.

1. Encrypt it with Caesar shift 4.
2. Encrypt it with Vigenère key `LINK`.
3. Encrypt it with Gronsfeld key `314`.
4. **Analyse** the repeated patterns and **Evaluate** relative resistance to simple brute force and frequency analysis.

View a sample solution

1. Caesar: `RIXASVO`.
2. Repeated key `LINKLIN`; ciphertext `YMGGZZX`.
3. Repeated key `3143143`; ciphertext `QFXZPVN`.
4. Caesar has only 25 useful keys and one substitution alphabet, making exhaustive search and frequency analysis easy. Vigenère and Gronsfeld vary shifts, but their repeated periods can be detected. Vigenère permits 26 shift values per key position, while Gronsfeld uses only ten; actual security also depends on key length and predictability. None is appropriate for modern sensitive data.

Revision #5
Created 2026-10-06 00:40:35 UTC by Mr Napper
Updated 2026-10-06 01:32:30 UTC by Mr Napper