**Methodology**

**1. Character-to-Number Mapping**

* Each character (A–Z) is mapped to a number (1–26), and space is mapped to 0.
* Example: 'A' → 1, 'B' → 2, ..., 'Z' → 26, ' ' → 0.

**2. Input Message Processing**

* User inputs a message and a square matrix.
* The message is converted into a list of numbers using the character map.
* If the length of the number list is not divisible by the matrix size, it is padded with 0 (space).

**3. Matrix Multiplication**

**a. Encryption**

* Message numbers are grouped into blocks matching the matrix size.
* Each block is treated as a vector and multiplied by the **encryption matrix**.
* Results are taken modulo 27 to map back into the 0–26 range.
* The resulting numbers are converted back to characters to form the encrypted message.

**b. Decryption**

* The **inverse of the encryption matrix** is computed.
* Encrypted message numbers are grouped and multiplied by the inverse matrix.
* Results are rounded and taken modulo 27.
* Numbers are then converted back to text.

**4. Flask Web Interface**

* A web form collects matrix values, message, and operation type (encrypt/decrypt).
* Flask routes process inputs, call the encryption/decryption functions, and render the output.

**5. Error Handling**

* Checks for matrix invertibility before decryption.
* Displays error messages for invalid inputs or non-invertible matrices.

**🔹 What Python Does**

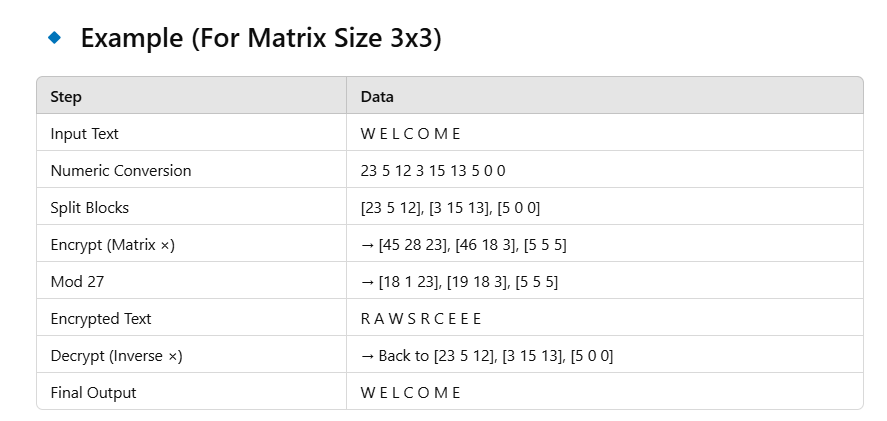
1. **Takes User Input**: Matrix (as key) and message via a web interface (Flask).
2. **Converts Characters to Numbers** using a mapping (A=1 to Z=26, space=0).
3. **Performs Matrix Multiplication** using NumPy for encryption/decryption.
4. **Handles Modulo Operations** to wrap results within valid character ranges.
5. **Displays Output**: Encrypted or decrypted message.

**🔹 Mathematics Applied**

1. **Linear Algebra**:
   * **Matrix Multiplication**: Message vectors × key matrix for encryption.
   * **Matrix Inversion**: Used in decryption to retrieve the original message.
2. **Modulo Arithmetic (mod 27)**:
   * Ensures all numerical results map within 0–26, which aligns with character mapping.
3. **Rounding of Floats** (Decryption):
   * Due to floating-point inversion, results are rounded to nearest integers.

**🔹 How It Helps (Benefits)**

1. **Cryptography**: Ensures **secure communication** by converting readable text into coded forms.
2. **Fast & Efficient**: Python with NumPy handles matrix operations quickly, even for larger messages.
3. **Automated Inversion**: Python calculates matrix inverse, saving manual effort and reducing errors.
4. **Educational Value**: Demonstrates real-world application of **matrices and modular arithmetic**.
5. **Customizable Security**: Users can define their own matrix keys, increasing encryption strength.



+---------------------------+

| Input Plain Text |

+---------------------------+

|

v

+----------------------------------------+

| Convert Each Character to Numbers |

| (A=1, B=2, ..., Z=26, Space=0) |

+----------------------------------------+

|

v

+--------------------------------------------------+

| Split Number Sequence into Blocks of Size 'n' |

| (e.g., for 3x3 matrix, blocks of 3 numbers) |

+--------------------------------------------------+

|

v

+-----------------------------+

| Multiply Each Block by Key |

| Matrix (Encryption Matrix) |

+-----------------------------+

|

v

+--------------------------------+

| Apply Modulo 27 to Each Entry |

+--------------------------------+

|

v

+--------------------------------+

| Convert Numbers Back to Text |

+--------------------------------+

|

v

+---------------------------+

| Encrypted Message |

+---------------------------+

### For Decryption:

+---------------------------+

| Input Encrypted Text |

+---------------------------+

|

v

+----------------------------------------+

| Convert Each Character to Numbers |

+----------------------------------------+

|

v

+--------------------------------------------------+

| Split Number Sequence into Blocks of Size 'n' |

+--------------------------------------------------+

|

v

+------------------------------------------+

| Multiply Each Block by Inverse Matrix |

| (Decryption Matrix) |

+------------------------------------------+

|

v

+-----------------------------+

| Apply Modulo 27, Round Off |

+-----------------------------+

|

v

+--------------------------------+

| Convert Numbers Back to Text |

+--------------------------------+

|

v

+---------------------------+

| Original Message |

+---------------------------+