**Short Report: Web-Based Text Encryption Tool**

**1. Introduction**

This project is a **web-based text encryption tool** designed to demonstrate classical cryptographic techniques using JavaScript.  
Users can input plaintext, select an encryption algorithm, provide a key (if required), and view the resulting ciphertext.  
The goal is to illustrate how different encryption methods transform data to protect it from unauthorized access.

**2. Implemented Encryption Techniques**

The system includes **three classical algorithms**:  
**Transposition Cipher**, **Polyalphabetic Cipher**, and **Vigenère Cipher**.  
Each represents a different approach to encryption — rearrangement or substitution.

**Transposition Cipher**

**Type:** Rearrangement Cipher  
**Concept:**  
In a transposition cipher, the characters of the plaintext are **rearranged** according to a specific pattern or key, without changing the actual letters.

**Working Principle:**

1. The plaintext is written into a grid with columns equal to the key length.
2. The columns are then read in alphabetical order of the key letters.
3. The resulting sequence of letters forms the ciphertext.

**Example:**  
Plaintext: HELLO WORLD  
Key: EOODHLWLL R

Grid (rows):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Z | E | B | R | A |
| H | E | L | L | O |
| W | O | R | L | D |

Alphabetical order of key: **A, B, E, R, Z**  
Ciphertext (read column by column): OLLERDHELOW

**Decryption:**  
The reverse process reconstructs the grid column-wise and reads it row by row.

**Polyalphabetic Cipher**

**Type:** Substitution Cipher  
**Concept:**  
The Caesar cipher is one of the simplest and oldest encryption techniques.  
It works by shifting each letter of the plaintext by a fixed number of positions in the alphabet.  
For example, a shift of **3** means **A → D**, **B → E**, **C → F**, and so on.

**Working Principle:**

1. Choose a fixed shift value (key).
2. Replace each letter in the plaintext with the letter that comes shift positions after it in the alphabet.
3. If the shift goes past ‘Z’, it wraps around to the start of the alphabet.
4. Decryption reverses this by shifting letters back by the same amount.

**Example:**  
Plaintext: HELLO  
Key: 3  
Shifts: H->K, E->H, L->O, L->O, O->R  
Ciphertext: KHOOR

**Decryption:**  
Subtract the shift (−3) from each ciphertext letter to get back the original text.

**Vigenère Cipher**

**Type:** Substitution Cipher (Special Case of Polyalphabetic)  
**Concept:**  
The Vigenère cipher is a refined version of the polyalphabetic cipher.  
It uses the same principle of shifting each letter by the amount indicated by the key but skips non-alphabetic characters so the key aligns only with letters.

**Working Principle:**

1. Repeat the key to match the length of the plaintext.
2. Shift each plaintext letter by the alphabetical index of the corresponding key letter.

**Example:**  
Plaintext: ATTACK AT DAWN  
Key: LEMON  
Ciphertext: LXFOPV EF RNHR

**Decryption:**  
Reverse the shifts using the same key.

**3. Validation and User Interaction**

* Input validation ensures users provide both text and a valid key.
* A simple interface allows encrypting, decrypting, and copying results.
* JavaScript handles all encryption logic directly in the browser (no server-side code required).

**4. Conclusion**

This project demonstrates fundamental encryption methods that laid the foundation for modern cryptography.  
While classical ciphers like **Transposition**, **Ceaser**, and **Vigenère** are no longer secure for practical use, they effectively illustrate core cryptographic concepts such as **permutation**, **substitution**, and **key-based transformation**.  
The web tool provides an educational and interactive way to explore how encryption works at a conceptual level.