Problem: The Enigmatic Communiqué

Blockchain Web3
Cyber Security Club

August 7, 2025

Objective

To understand and implement the **Playfair cipher**, a classic polygraphic substitution cipher. This mission requires not just implementing the decryption algorithm, but also performing a small act of cryptanalysis to discover the secret key before decryption can begin.

The Intelligence Briefing

Cipher System: The Playfair Cipher

Unlike simple substitution ciphers that replace one letter at a time, the Playfair cipher is a **polygraphic cipher**, encrypting letters in pairs (digraphs). This was a significant advancement in the 1850s because it disguised single-letter frequencies, making it much harder to break with standard frequency analysis.

The cipher relies on a **5x5 key square**, which is a grid containing the letters of the alphabet.

1. **The Key Square:** The grid is built using a secret **keyword**. First, the unique letters of the keyword are placed in the grid, and then the remaining letters of the alphabet are filled in.

2. **The Twist:** To fit the 26-letter alphabet into a 25-cell grid, one letter must be omitted. While 'J' is often combined with 'I', our target agent uses a different protocol: their grid **completely omits the letter 'Q'**.

3. **Encryption Rules:** Letters are encrypted based on their position in the grid:

   - **Same Row:** If two letters are in the same row, they are replaced by the letters immediately to their right (wrapping around if necessary).
   - **Same Column:** If in the same column, they are replaced by the letters immediately below them (wrapping around).
   - **Rectangle:** If they form a rectangle, they are replaced by the letters on the same row but at the other corners of the rectangle.

The Scenario: An Agent's Final Message

An undercover agent has gone dark. Their last transmission is a single, encrypted message. Standard decryption has failed, likely due to a custom protocol. Your team has one piece of vital intel: the agent always encoded their keyword based on a personal "parting phrase" they used in debriefings.

Your Mission

Your mission is to decrypt the agent's final message. This is a two-part operation: first, deduce the keyword, then implement the Playfair decryption algorithm according to the agent's unique protocol.

The Intel & Rules of Engagement

1. **The Keyword Clue:** The agent's parting phrase was always: *"Remember the old saying about what curiosity did to the cat."* Your keyword is the primary nine-letter subject of that saying.

2. **The Custom Protocol:** As mentioned, the 5x5 key square **omits 'Q'**. The alphabet for your grid is A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, R, S, T, U, V, W, X, Y, Z.

3. **The Ciphertext:**

```
"KBVE MPBI YXAL SKAY RPMI YSGA TLHE LONP DE"
```

4. **Decryption Process:**

   - Solve the clue to find the keyword.
   - Construct the 5x5 key square using the keyword and the 'Q'-less alphabet.
   - Group the ciphertext into pairs of letters.
   - Apply the inverse Playfair rules to each pair to find the original plaintext digraphs.
   - Combine the decrypted digraphs to reveal the message. Note that the original message may have used 'X' to separate double letters or to pad a final odd letter. You may need to interpret the final message logically.

Implementation Guide

You should structure your solution around a primary decryption function. It's also highly recommended to create helper functions to generate the key square and find character coordinates within it.

```python
def generate_key_square(keyword: str) -> list[list[str]]:
    """
    Generates the 5x5 Playfair key square based on the keyword and the ’Q’-less alphabet.
    """
    # Your logic here to build the grid
    pass

def decrypt_playfair(ciphertext: str, keyword: str) -> str:
    """
    Decrypts a Playfair-encrypted message.
    """
    # 1. Generate the key square using your helper function.
    # 2. Prepare the ciphertext (remove spaces, process in pairs).
    # 3. Loop through pairs and apply inverse Playfair rules.
    # 4. Join the decrypted pairs to form the final message.
    pass

# --- How to use your code ---
# First, figure out the keyword from the clue!
keyword = "???" # Replace with your deduced keyword

ciphertext = "KBVE MPBI YXAL SKAY RPMI YSGA TLHE LONP DE"

# Decrypt the message
secret_message = decrypt_playfair(ciphertext, keyword)
print(f"Decrypted Message: {secret_message}")
```