Transposition
1. Permutation
2. Column Transposition
3. Double Transposition

Permutation

Key

\[
\begin{array}{l}
1 \quad 2 \quad 3 \quad 4 \\
\end{array}
\]

\[
\begin{array}{l}
\text{LETUS} \quad \text{GO TO THE STORE} \\
\text{EXXXX} \\
\end{array}
\]

ESLTU OTGTOEOHSTEXRX

2 3 4 1

HOUSE

Sweep columns
reorder columns
so the top makes
words
Double Transposition
2 diff keywords

def doublet(char* plain, char* key1, char* key2):
    char* mid = trans(plain, key1);
    return &trans(mid, key2);

We can set an equivalent single trans w/ a key word of length len(key1) + len(key2)

Can get more obfuscation from combining multiple simple operations + this is easier than trying to come up with a single more complicated equivalent operation.