CS553 Cryptography

BitBees

Question 6

1 Known Plaintext Attack on Hill Cipher

We have mentioned two methods to launch a Known Plaintext Attack on Hill Cipher. First method involves using the known plaintext and ciphertext pairs, also known as 'crib' to perform a series of matrix multiplication in order to calculate the key matrix. The other method involves trying all possible key matrices to perform encryption on known plaintext and comparing it with known ciphertext.

1.1 Crib Attack on Hill Cipher

Let us assume that we have a key K which is a 2×2 matrix.

$$K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

Assuming we have two of our known plaintext as $m_1 = (w_1, w_2), m_2 = (w_3, w_4)$. The equation for hill cipher looks like

$$c = wK$$

where,

w is the message(plaintext),
c is the ciphertext, and
K is the key

Using the above equation we have,

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} c_m \\ c_{m+1} \end{bmatrix}$$

and

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} c_{m+2} \\ c_{m+3} \end{bmatrix}$$

The above equations can be rewritten as

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} = \begin{bmatrix} c_m \\ c_{m+2} \end{bmatrix}$$

and

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix} = \begin{bmatrix} c_{m+1} \\ c_{m+3} \end{bmatrix}$$

Hence we can obtain the key K by computing the inverse of w like this

$$\begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} = \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix}^{-1} \begin{bmatrix} c_m \\ c_{m+2} \end{bmatrix}$$

and

$$\begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix} = \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix}^{-1} \begin{bmatrix} c_{m+1} \\ c_{m+3} \end{bmatrix}$$

This is the core concept behind the code we have written to break Hill Cipher with known crib. Now given a variable sized crib, we select all possible 4 letter combinations under a certain condition.

Please note that not all w are invertible, therefore it is necessary to try multiple combinations within the crib in order to find a w which is invertible.

Let i_1, i_2, i_3, i_4 be the indices of the chosen letters in a n-sized crib, where $n \geq 4$

$$i_2 = i_1 + 1 \tag{1}$$
$$i_4 = i_3 + 1 \tag{2}$$
$$i_2 \mod 2 = 0 \tag{3}$$
$$i_3 \mod 2 = 0 \tag{4}$$

With all these condition satisfied, we write the code for this attack in Python using SageMath library. Let's test our code to decrypt a message where we know encryption of a few letters in the message.

As per the requirements, we choose our plaintext to be the names of the group members. Concatenation of the names of the group members looks like this.

$$M = \texttt{SATVIKVEMUGANTIANANYAHOODACHAITANYABISHT}$$

The key(not known to the adversary) is

$$K = \begin{bmatrix} 14 & 3 \\ 1 & 20 \end{bmatrix}$$

Let us assume that the adversary knows that the encryption of the substring `VEMUGANTI` is `MNGUGSTDI`.

He tries to decrypt the ciphertext $C =$ `SCBJSQMNGUGSTDIYANYZHKCKQJJQIEGFYZBUAUNL`

Applying the KPA(ciphertext, crib, encrib) returns us the message M, which means that the attacker successfully recovered the entire plaintext by knowing the encryption of a few letters of plaintext.

Within the KPA function, we have a subroutine called crib_dragging. This subroutine basically takes the known plaintext and its corresponding ciphertext and generates all valid combinations according to Eq (1), (2), (3), (4).

This method relies on the fact that the matrix w generated is invertible. It is possible that none of the combination yields a w which is invertible for certain cribs.

For example if the adversary knew that the decryption of `BISHT` is `UAUNL` under the same key K, he would not be able to deduce the key using the above mentioned method as none of the combinations of crib yield a w which is invertible.