Highly Secure System – 2014 Semester 2

Lecture: Andrew Ensor

Assignment one: Symmetric Cipher

Chosen topic: Enigma machine

**1. History and Usage**

The Enigma machine is a rotor cipher machine which was widely used during the period between 1930 and 1950 for encryption. The core mechanism of Enigma machine was first developed in 1918 by Arthur Scherbious, a German engineer in the commercial companies in secure communication. Over the decades after this invention, German military has produced its own version with more complex plugs and electronic circuits to avoid interceptor to decrypt the message.

The first break though of cryptanalysis against commercial Enigma was first done by three Polish mathematicians in 1929. They have later created a mechanical device called the “Bomba” that simulates 26\*26\*26 positions of the 3 Enigma rotors. By the year 1939, Bomba could simulate up to 6 rotors. When these mathematicians fled abroad during WW2, these knowledge and materials were passed to Polish allies and the most notably the Britain's Government Code and Cipher School where Alan Turing was working. After 3 month, the famous “Turing Bomb” was created which had 36 rotors to simultaneously test possible cases of twelve 3-rotor Enigma machine. The machine used known-plain-text attack on words such as “General” or “To the group”.

**2. Description of the Algorithm**

Our implementation of the algorithm contains mechanism to adjust number of rotors and initial letters of the rotor (key), with an option to encrypt using fixed reflector.

2.1 Encryption

Each rotors are given a scrambled sequence of alphabet along with an integer denoting the position with in the rotor. All of communication between the rotors are done through int value of chars for efficiency and clarity. The encryption starts by initializing each rotors to a given position according to the key (e.g. key for 3 rotors can be “ABC”). Than the algorithm takes the first letter of the plain text and finds the index within the first rotor. This index gets converted to an alphabet than used as the input for the next rotor to be substituted again and so on and so forth until it goes through all rotors or reaches the reflector. The reflector is a fixed rotor, meaning there are only 13 possible permutation that works both ways (e.g K↔M). After the permutation using the reflector, the output travel backwards till the first rotor. After a letter is encoded, the first rotor gets shifted by 1, and if full rotation was reached than the following rotor also gets rotated and so forth. Repeat until every letter is encrypted.

2.2 Decryption

With out a reflector: Reverse the direction of rotor rotation with the same key (initial rotor position).

With a reflector: Use the same algorithm with the same key.

2.3 Strength

When using the reflector, same algorithm can be used for encryption as well as decryption.

Has high number of possible key for a poly-alphabetic substitution cipher.

2.4 Weakness

Reflector was once considered as a great improvement as it will double the number of rotors, and same Enigma machine could be used to decipher with out switching between modes. On the other hand, a vulnerability has been exploited as it turns out that using reflector causes any letter to never encrypt itself (e.g. G will never become G).

It is vulnerable against cribs: known-plain-text attack.

The keys (initial rotor setting) were often not really random (e.g. AAA, initials of girlfriend, abbreviations)

**3. Cryptanalysis**

Our chosen cryptanalysis attempts to find the key of the given cipher text with a combination of brute-force and frequency analysis against bi-gram chart. This algorithm assumes that the Enigma machined used to encrypt the plain text is available for observation. In another word, this algorithm relies on the fact that we know the number of rotors used and the letter sequences on each rotors.

Start by preparing every possible key for number of rotors used to encrypt the plain-text (E.g. if number of rotor is 3 than generate key from “AAA” to “ZZZ”). For each key generated, decrypt the text then validate the decrypted text before passing on the frequency analysis. The validation process first ensures there is no white spaces in the decrypted text and checks whether, by chance, the decrypted text matches the original plain text. After the validation process, the decrypted text goes through frequency analysis against bi-gram chart to calculate a score for the key that was used to decrypt the cipher text.

**4. Evaluation**

Table 1: Result of cryptanalysis algorithm : Number of rotor VS time

From the result of our cryptanalysis algorithm as shown in Table1, it is clear that every rotor adds exponential amount of time against brute-force and frequent analysis. This algorithm has time complexity of O(n^x).

**5. Instructions**