Homework 2

Encryption

Botond Lajos Perényi

Task 1: Confusion and diffusion

Part i: Encrypt THEWANDCHOOSESTHEWIZARD using Vigenère cipher with MAGIC as key: FHKECZDIPQASKAVTECQBMRJ. I used my code from the previous assignment to achieve this. (In my code I used lowercase, meaning I had to convert the message and key in my code.)

Part ii: Change one letter in the plaintext: Encrypt THEWANDCHOOSESTHELIZARD using Vigenère cipher with MAGIC as key: FHKECZDIPQASKAVTERQBMRJ. I changed one letter compared to the original message, and only one letter changed in the resulting ciphertext.
The diffusion property is when changing one bit (in this case letter) in the plaintext changes around half of the resulting ciphertext. This does not happen with Vigenère cipher, as changing one letter in the plaintext changes only one letter in the ciphertext. This means that the diffusion property is not achieved in the Vigenère cipher. (The diffusion property should not depend on which letter is changed in the plaintext.)

Part iii: Change one letter in the key: Encrypt THEWANDCHOOSESTHEWIZARD using Vigenère cipher with MANIC as key: FHRECZDPPQASRAVTEJQBMRQ. I changed one letter in the key compared to the original key, and five letters changed in the resulting ciphertext (compared with the original ciphertext).
The confusion property is when changing one bit (a letter in this case) in the key changes around half of the resulting ciphertext for a given plaintext. With this message (which is 23 letters long) the expected change should be around 11, but this does not happen with changing one letter in the key. Thus the confusion property is not properly achieved in the Vigenère cipher.

Task 2: Pseudorandom function

A pseudorandom function is one where the adversary cannot meaningfully distinguish between a random function or an encryption (with a given key K) is applied to the message based on the output.

- $F'(k, m) = F(k, m) \| 0^n$
  This function does not satisfy the pseudorandom function's definition: an adversary can say if the last character is '1', then it was the generated function. This produces a greater hit result than it would be purely by chance.
- $F'(k, m \| m') = F(k, m) \| F(k, m' \oplus 0^n)$
  One can generate a message so that $m = m' \oplus 0^n$ (e.g.: a message with all 1's). In this case the result from F' will consist of the same two blocks if the encryption was used. This will help discover if the result is the product of the random function or the F'.
- $F'(k, m \| m') = F(k, 0 \| m) \oplus F(k, m' \| 1)$
  Generate a message with the following properties: $0 \| m = m' \| 1$. This would mean that the message looks like this: $0 \| u \| 1 \| 0 \| u \| 1$ where u is a binary string. In this case the F' will return all 1's, which will help disseminate from the random function's output.

Task 3: Output feedback mode

Part i: Encrypting DOG (which is $00011, 01110, 00110$) with OFB generated the following result: $10000, 10100, 11111$. I used my own implementation, which can be found in Source code.

Part ii: After flipping the fifth bit of the ciphertext we get $10001, 10100, 11111$. Decrypting this with the same OFB (with the same initialization vector), we got $00010, 01110, 00110$. Comparing to the previous encryption we can see that only one bit changed.

Part iii: Changing the initialization vector (only one bit) and decrypting the original ciphertext $(10000, 10100, 11111)$ we get $01011, 01111, 00100$. Comparing it to the original message $(00011, 01110, 00110)$, we can see that 3 bits changed in the resulting plaintext.

Task 4: CFB

Cipher Feedback mode encryption means that the cipher c_i is calculated as such:

$$c_i = E(k, c_{i-1}) \oplus m_i$$

A collision with $c_i = c_j$ can be written as such:

$$E(k, c_{i-1}) \oplus m_i = E(k, c_{j-1}) \oplus m_j$$

From this we can deduce:

$$m_i \oplus m_j = E(k, c_{i-1}) \oplus E(k, c_{j-1})$$

This means, we can separate the messages and the cipher blocks' output in a way, that would not be possible without this collision.

Task 5: IND-OT-CPA

Pigpen cipher under IND-OT-CPA is not secure because it retains the patterns in the plaintext. This means we can send two messages with two very distinct patterns, and the resulting ciphertext will display the same pattern.
Example: $m_1 = ABABABAB \dots AB$ and $m_2 = ABCABC \dots ABC$. In this case the ciphertext will have 2 repeating symbols for m_1 and 3 repeating symbols for m_2, making it possible to identify them with just one query.