Network Security (CSE350) Programming Assignment-1

Abhinav Ujjawal (2021120)
Nikhil Suri (2021268)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Mono-alphabetic Substitution

- Mono-alphabetic Substitution of a pair of characters
- Plaintext is converted to Ciphertext using table-lookup
- Every two characters are mapped to some other two characters

Input Constraints

- Plaintext
  - Plaintext characters must belong to the set {A, B, C}
  - The plaintext must be a multiple of 4 + 16-bit Hash Value, however the plaintext must be of even length for encryption to work
  - The plaintext must satisfy the property p = (s, Hash(s))
- Key
  - Key is a Table, implemented as a Python Dictionary

Encryption

1) Encrypt the PlainText into CipherText, using the encryption key

- This is simply an O(n) procedure, where n = size of PlainText

Decryption

1) Decrypt the CipherText into PlainText, using the decryption key

- This is also an O(n) procedure, where n = size of CipherText

Hashing

We use this hash function to construct plaintexts that are recognizable, i.e, those that satisfy the property: p = (s, Hash(s)).

1. Initialize the initial Hash Value = “0000000000000000” (0-string of len=16)
2. Divide the input plaintext into blocks of N-character segments, where N is a constant specified in the implementation – in our implementation, N = 16
3. For each block of characters, do the following:
   a. Rotate the current hash value to the left by one bit
   b. XOR the block with the hash value, and store the result as the new Hash Value
4. Encode the bits in hash using this :
   a. 0 - A
   b. 1 - B
5. Return the Hash Value

Recognizability

The function “is_recognizable” checks whether a particular PlainText is recognizable or not.

The function works as follows:

1. Calculate the hash value of the candidate PlainText by calling the hash function
2. Compare the calculated hash value with the expected hash value, and check for equality
3. If the two values are equal, return "True", indicating that the candidate PlainText is recognizable. If the two values are different, return "False", indicating that the candidate PlainText is not recognizable

Brute-force Solution

- We get all the possible combinations of the key (9!=362,880). Iterate through these combinations and try to use them to decrypt the first ciphertext. If it is recognisable after decrypting, try this key over other ciphertexts.
- The asymptotic time complexity of discovering the key via brute force is O(9!) in our implementation. More generally, Brute-force is O((n^2)!), where n = number of symbols in the universe.