Milan Haruyama

CS 428

Dr. Laxima Niure Kandel

28 April 2022

SNEED Encryption Scheme

SNEED is an encryption scheme developed by Milan Haruyama that incorporates elements stream and block ciphers. SNEED shares a similar structure to one-time pad.

SNEED first reads a user-created text file called “plain.txt” to determine the length of the plaintext. Afterwards, the plaintext is broken into *n-byte* long blocks, where *n* is the largest prime integer factor of the plaintext length. If the plaintext length is odd, nine bytes will be appended that correspond to the string “oddoddodd”.

Encryption occurs when each block is XOR’d with the corresponding key byte. For example, each byte of Block 1 is XOR’d with the first byte of the key, each byte of Block 2 is XOR’d with the second byte of the key, and so on. After this, each byte of this operation is XOR’d with the key in descending byte order. For example, the first byte of the previous operation is XOR’d with the last byte of the key, the second byte of this previous operation is XOR’d with the penultimate byte of the key, and so on. Since the above function is deterministic, decryption occurs when the steps are performed in reverse.

Some of the issues that had to be overcome when developing SNEED was creating a data type in C to store both the bytes and the length of each component. This was accomplished by creatin a “component” structure. The next obstacle was creating a function to find prime factors of a length. The developer ultimately used a publicly available code to solve this issue. The next obstacle was to properly scan text files and append nine bytes to any odd plaintext lengths. The final obstacle, after the encryption scheme was created, was creating the decryption scheme by reversing the encryption scheme. This includes removing the appended bytes from the ciphertext. All of these issues involved working with pointers in C. If coded incorrectly, pointers can cause the program to crash. Hence, ensuring each pointer was implemented correctly was essential for the developer. Along with this, proper documentation of the program was also vital.

According to Dr. Laxima Niure Kandel, the security of SNEED is similar to that of one-time pad since both share a similar structure. This means that SNEED is only secure with the use of one-time keys and becomes vulnerable when a key is used more than once.