SENG2250/6250 System and Network Security

School of Electrical Engineering and Computing

Semester 2, 2020

Lab 2: Topic 2 - Cryptographic Techniques

Objectives

1) Review the knowledge of Topic 2 cryptographic techniques (symmetric-key cryptography).
2) Apply cryptographic techniques for problem-solving.
3) Implement cryptographic operations in programming.

Part 1 Review Questions

1. Explain in which cases would be suitable to use symmetric-key encryption, which cases would better to use asymmetric-key encryption. Give an example for each.
2. What is the perfect secrecy?
3. Describe the encryption and decryption processes of the CBC operation mode.
4. What is a cryptographic hash function?
5. How would stream cipher relate to block cipher?

Part 2 Exercises

6. **Cryptanalysis**: Apply cryptanalysis to reveal the (meaningful) plaintext of the following ciphertext:

   KRHPH FL BX BAAWH ZX KRH KPHH

   (The ciphertext was generated by using the monoalphabetic substitution cipher.)

7. **Triple-DES**
   a. Find out the meet-in-the-middle attacks. (e.g., https://en.wikipedia.org/wiki/Meet-in-the-middle_attack.)
   b. Why does double-DES have (approx.) 2^{57} security level.
   c. Why does Triple-DES (3 independent keys) have (approx.) 2^{112} security level.
   d. Why is the middle portion of the triple-DES is decryption rather than encryption?

8. **Hash Functions**

   a. Is the following function H a hash function? Why?

$$H(x) = \ x \ mod\ 65537, x \in \mathbb{N}$$

   b. Is the above function H a cryptographic hash function? Why?

9. **Programming**

Fast modular exponentiation ($b^e\ mod\ n$) is an essential operation for many modern security algorithms. In Lab 1, you have implemented the modular exponentication function. But it is slow if the base and exponant are very large. We introduce you a fast modular exponentiation as below. Write a (C/C++/Java/Python) program to implement the fast modular exponentiation operation based on the following pseudocode.

```
function powmod2(base b, exponent e, modulus n) {
    if n  = 1
         return 0
    rs = 1
    while (e > 0) {
        if (e & 1) == 1
            rs = (rs * b) mod n
        e = e >> 1
        b = (b*b) mod n
    }
    return rs
}
```

Use the above implementation to find the solutions

$$3^3\ mod\ 7 =?$$
$$10^8\ mod\ 133 =?$$
$$3785^{8395}\ mod\ 65537 =?$$
$$17^{45} \times 17^{61}\ mod\ 1023 =?$$
$$17^{45+61}\ mod\ 1023 =?$$

Try to understand the algorithm.

Part 3 Discovery (external readings)

10. Self-study: Vigenère cipher (e.g., https://en.wikipedia.org/wiki/Vigenère_cipher)

    a. Encrypt the plaintext:

    CRYPTOGRAPHY IS A KEY OF CYBERSECURITY

    using the tabula recta of the Vigenère cipher.

    b. Is the Vigenère cipher secure against cryptanalysis?

11. Self-study: Birthday Paradox (e.g., https://en.wikipedia.org/wiki/Birthday_problem)

    a. What is the birthday paradox?

    b. How does it relate to the security of hash functions?