CSCI 530 - Security Systems

Rishit Saiya - 4940905271, Assignment - 1

September 15, 2022

1

Why do we use encryption modes of operation to convert block ciphers into stream ciphers?

In the case of Block Ciphers, when a same block of plain text is used, the cipher text also turns out to be the same. This crucial pattern is easily identified by the adversary can a clever guess can be made to figure out the pattern of encryption. In order to overcome this problem, Stream Ciphers were introduced wherein the encryption transformation is done a bit, byte or block at a time fashion which makes it incredibly hard to reverse engineer the possible combinations of pattern matching to decrypt the text. A typical example of Block Cipher is ECB (Electronic Code Book) which essentially is a DES Mode of Operation. In ECB, cipher is made using a block at a time and the above mentioned problem exists here. Whereas, Stream Ciphers such as OFB, CFB, CBC use the encryption in a bit-by-bit or in a stream fashion which is hard to decrypt by some patter matching or guess work.

What is important about the initialization vector (IV) in a stream cipher, and what happens if the IV is known by the adversary? Explain your answer. What happens if we always use an IV of 0000 or 1111?

Initialization Vector (IV) is essentially a random generally string used along with a secret key in general to encrypt data in the stream ciphers. This IV is important as this helps as an additional operation which when operated on same plaintext doesn't yield same cipher. This solves the above mentioned problem. If the IV, is known by any adversary, it doesn't affect the sanctity of the encryption/cipher text here as the Encryption key is not known. Encryption Key in essence will be private and IV is generally used only once during the encryption process. In the cases where the IV was 0000 or 1111, the resulting cipher text will be same for the plain text.

Can a protocol be designed so it is just as safe to use the same IV to encrypt a message stream, as it is to use a different IV for the stream each time? Explain

your answer.

In the scenario where same IV is used repetitively to encrypt the plaintext, it becomes predictive for the adversary. The starting blocks of plaintext would generate essentially the same cipher text. With a large sample space of such cipher text, it will be easy for any adversary to exploit this vulnerability. In such case, an elegant idea would be to randomize every entering first blocks of plaintext data as that will cascade further encryption in the process and the yielded ciphertext won't be common as mentioned above.

2

According to RSA algorithm, n can be calculated as:

$$n = p \times q \implies n = 11 \times 23 = 253$$

For given e = 3, d can be calculated as follows:

$$ed \equiv 1 \text{mod } (p-1)(q-1) \implies 3d \equiv 1 \text{mod } 220$$

A clever way of interpretation of this congruence relation would be that when 3d is divided by 220, the remainder is 1. So in simple terms, a series of 3d can be as follows:

$$3d = 221, 441, 661, 881, 1101, 1321, ...$$

441 seems to a feasible solution as d can be a integer in that case. Hence, value of d is:

$$3d = 441 \implies d = 147$$

In the Encryption process of RSA, given that m = 5, c can be calculated as:

$$c \equiv m^e \text{mod } n \implies c \equiv 5^3 \text{mod } 253 \implies c = 125$$

For the Decryption process of RSA, m can be calculated as:

$$m \equiv c^d \text{mod } n \implies m \equiv 125^{147} \text{mod } 253 \implies m \equiv 5^{441} \text{mod } 253$$

In order to calculate, 5^{441}, Modular Exponentiation Technique was used as follows:

$$441 = (110111001)_2 \implies 441 = (1 + 8 + 16 + 32 + 128 + 256)$$

Therefore, 5^{441} can be split as follows:

$$5^{441} = 5^{1+8+16+32+128+256}$$

Multiplication of Congruence will be used for future calculations. The property states that:

$$\begin{aligned} a &\equiv b \quad \text{mod } m \\ c &\equiv d \quad \text{mod } m \\ \implies ab &\equiv cd \quad \text{mod } m \end{aligned} \tag{1}$$

Using this, it becomes easier to calculate the larger values and it is done is below fashion:

$$\begin{aligned}5 &\equiv 5 \mod 253\\ 5^8 &\equiv 246 \mod 253\\ 5^{16} &\equiv 49 \mod 253\\ 5^{32} &\equiv 124 \mod 253\\ 5^{128} &\equiv 213 \mod 253\\ 5^{256} &\equiv 82 \mod 253\end{aligned}$$

Using Equation 1, all above congruence relations can be multiplied resulting in:

$$\begin{aligned}5^{441} &\equiv (5 \times 246 \times 49 \times 124 \times 213 \times 82)\text{mod } 253\\ \implies 5^{441} \equiv (60270 \times 2165784)\text{mod } 253 &\implies 5^{441} \equiv 5\text{mod } 253\end{aligned}$$

Hence, m = 5, which indeed proves that deciphering c with d yields m again.

3

When using XOR with a random key as a method of encryption why is it important that they key be used only once?

The simple property of XOR which is mentioned in the below lines can be used to explain the claim. In the following equations, $\oplus$ is the XOR operator, M_1, M_2 are the plain text messages and C_1, C_2 are their respective ciphers.

$$M_1 \oplus K = C_1 \tag{2}$$

$$M_2 \oplus K = C_2 \tag{3}$$

From (2) and (3), we can say that:

$$C_1 \oplus C_2 = (M_1 \oplus M_2) \oplus (K \oplus K) \equiv C_1 \oplus C_2 = M_1 \oplus M_2$$

So essentially, we can see that the key totally gets eliminated for the very fact that the same key was used. Additionally, with the help of the cipher's XOR operation, now the attacker has the knowledge that even their respective messages' XOR operation would lead to the same XOR result. Using that XOR data, this information can be used to decode M_1 and M_2 by bruteforce or such methods.

What is the method of encryption called? State the purposes for which it is secure and the purposes for which it is weak and explain the reasons for your assessment.

The mentioned method of encryption where key only used once is called One Time Pad Encryption. Such encryption method is secure as it achieves the goal of Confidentiality provided that the key communication prior to sending message between the sender and receiver was done in a secure manner. Such encryption can be proven weak as well because of the following reasons:

- The key exchange prior to sending messages was obfuscated, then the key's integrity is in danger and same goes for the remaining whole process of encryption.

- If following the above idea, if a bit is altered in the key, then as discussed in the class, the attacker can check some existing pattern and cleverly deduce the plaintext.

Adding to the above reasons, the key management in terms of exchange of data size is extremely tedious. Since the key size is dependent upon the data stream, the key size increases with the increasing data stream size. It is conclusive that as we want to achieve secure communication means, we need to generate the key in a random fashion in order to overcome the above mentioned weakness.