Week 7: Cryptography: Block Ciphers

Random Number Generators

Question 1: What are good examples of random number generators? Where does randomness come from?

A. Radioactive Decay
B. RF emissions from the big bang
C. Thermal Emissions
D. Mouse and Keyboard movements

Question 2: Below is the source code for C's random number generator. Under what circumstances is this random number generator "truly random"?

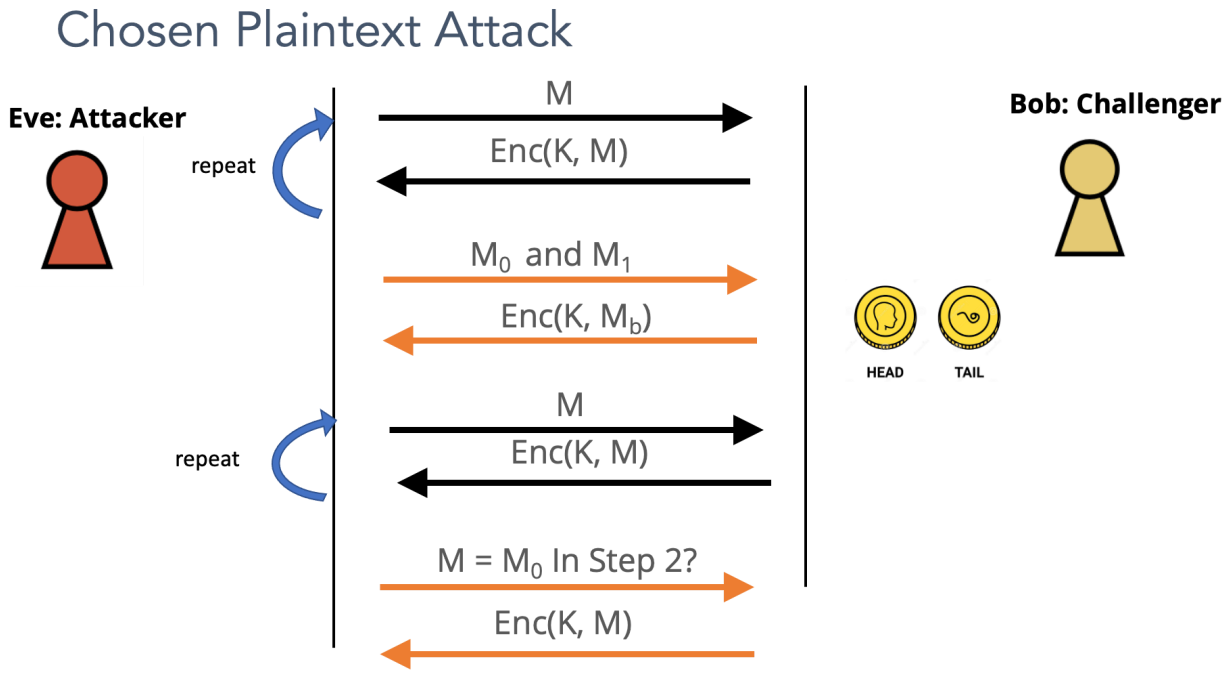
```
unsigned long int next = 1;
/* rand: return pseudo-random integer on 0…32767 */
int rand(void){
   next = next * 11-3515245 + 12345;
   return (unsigned int) (next/65536) % 32768;
}
/* srand: set seed for rand() */
void srand(unsigned int seed){
    next = seed;
}
```

Question 3: We've seen that random number generators are generated on most machines from physical inputs such as mouse clicks etc .How are random numbers generated for Virtual Machines and Servers?

Defining a rigorous threat model: Chosen Plaintext Attack

Below is a depiction of an Adversary that is capable of launching a Chosen Plaintext Attack. Here we have an adversary Eve and a challenger Bob. We design a CPA game as follows:

1. **Step 1:** Eve is allowed to ask Bob for encryptions of messages of Eve's choosing. Eve can send a plaintext message to Bob and Bob will always send back the encryption message, encrypted with a secret key. Eve is allowed to repeat this as many times as she wants.
2. **Step 2:** Eve then chooses two different messages M0 and M1 and sends both messages to Bob. Bob flips a fair coin, and if it is heads, then Bob encrypts M0, otherwise Bob encrypts M1.
3. **Step 3:** Eve can then repeats step 1 as many times as she likes, until she's ready to guess the message that Bob encrypted in Step 2. (Eve may guess M0 or M1).
4. **Step 4:** If Eve can guess the message with probability greater than ½ then Eve has won the game.



Question 1: To make this a practical security definition we will enforce that M0 and M1 have to be of the same length in plaintext and allow a crypto scheme to leak the length of the plaintext. Why would this caveat be necessary?

A. It's not really necessary, it's just an artifact
B. It's necessary because otherwise our game would incorrectly mark some secure schemes as insecure.
C. It's not practical for a crypto scheme to hide the length of the plaintext.

Question 2: Our second caveat in our threat model is going to be that Eve is limited to a practical number of encryption requests. Why would this caveat be necessary?

A. It's not really necessary, it's just an artifact
B. To only account for computationally feasible attacks (what does this mean?)
C. To distinguish between attacks that are theoretically possible but not practical.

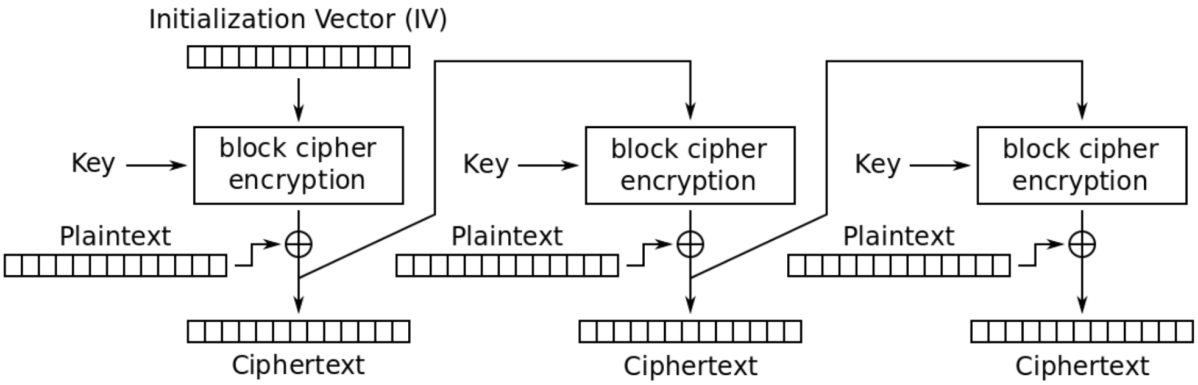
Question 3: We saw that block ciphers can be fully described by an Encryption Function E that takes as input a randomly generated k-bit key and an n-bit plaintext message to produce an n-bit ciphertext c. We said that the encryption function scrambles each of the 2^n possible input plaintexts to a different pseudorandom ciphertext output. And the particular "scramble" or mapping from input to output is maintained by our key. Based on our above definition of CPA security, are block ciphers CPA secure?

A. Yes, because the key is random
B. Yes because the encryption function is a random mapping from input to output.
C. No, because for the same key and the same message we get the same ciphertext.

Question 4: Assume that an adversary chooses an encryption scheme and runs the CPA game a large number of times, winning with probability 0.6. Is the encryption scheme CPA secure? Why or why not?

A. Yes, explain why:
B. No, explain why:

Block Ciphers: Encryption Models



Cipher Feedback (CFB) mode encryption

Question 1: Above we have a diagram of the CFB mode or the Cipher Feedback Mode whose encryption is given as follows:

$$C_i = \begin{cases} \text{IV}, i = 0 \\ E_k(C_{i-1}) \oplus P_i, \quad \text{Otherwise} \end{cases}$$

What is the decryption formula for this CFB mode?

Question 2: Select the true statements about CFB mode:

A. Encryption can be parallelized
B. Decryption can be paralellized
C. The scheme is CPA secure

Question 3: What happens if two messages are encrypted with the same key and IV? What can the attacker learn about the two messages just by looking at their ciphertexts? HINT: Think about how each block cipher transforms the corresponding plaintext block.

A. The attacker can determine if two messages have identical prefix up to the first block containing the difference.
B. The attacker can determine the entire plaintext by using the decryption formula on the available ciphertexts.
C. The attacker cannot determine any of the inputs, since CFB is CPA secure.

Question 4: If an attacker recovers the IV used for the CFB mode, but not the key, will they be able to decrypt a ciphertext encrypted with the recovered IV and a secret key?