Applied Cryptography

Unit-2 Symmetric key Cryptography

Lecture Notes 6:

Feistel Structure

Recommended Reading:

Katz-Lindell: Chapter 6:6.2.5

1. **Introduction:**
   - As discussed previously, ideal block cipher, the use of ideal block cipher is limited in practice because the key length is too large.
   - Let's discuss about block cipher designs that enables secured communications with smaller key lengths.
   - One researcher who worked on designing practical block cipher is Horst Feistel. Horst Feistel was a German born researcher who worked in IBM.
   - He is famous for leading the IBM team whose design became the Data Encryption Standard or DES. But let's first review Feistel cipher.
2. **Feistel Structure:**
   - Feistel cipher structure framework for symmetric block ciphers, and it is used for many block ciphers including DES.
   - Feistel was motivated to design a practical block cipher because he knew that ideal block cipher would be limited in practice.
   - Feistel wanted an approximation of ideal block cipher built out of components that are easily realizable.
   - Feistel proposed that the ideal block cipher can be approximated by utilizing the concept of a product cipher, which executes two or more simple ciphers in sequence. So that the final product is cryptographically stronger than any of the component ciphers.
   - In particular, Feistel proposed the use of a cipher that alternates substitutions and permutations, this structure is called Feistel cipher or Feistel network.

- Feistel cipher uses a finite number of bits for the key length of k bits, which is significantly smaller than the key length of an ideal block cipher, which is n x 2 to the nth power bits given the block length of n.
- Also, because the key is k bits, their 2 to the kth power possible keys, which is smaller than the 2 to the nth power, factorial possible transformations or keys for the ideal block cipher.
- The figure illustrates the Feistel Cipher where the encryption process is shown on the left and the decryption on the right.
- The cipher progresses downward. Feistel cipher partitions input block into two halves, the left half and the right half, which are processed through multiple rounds. Each round performs a substitution with the left half of the data, which is based on the function F of the right half of the data and the subkey.

Then permutation follows, swapping the two halves.

Let's look at the diagram. From the top of the diagram, the plain text block is divided into two halves, L0 and R0, which are the inputs to round 0. The two halves of the data iterates through n rounds to generate the cipher text block. For each round i, the inputs of round i are Li and Ri, derived from the previous round and a subkey Ki, which is derived from the key K. The key K is not directly used in the rounds, and the subkeys Ki are different from K and from each other. The substitution and the permutation transformations come from the processing within the rounds. The function F is called a round function and is designed for substitution. The output of the round function F is with the left half of the data in each round. At the end of each round, Feistel cipher swaps the left half and the right half for permutation. For each round i, we can mathematically express the operation as following, and this is iterative operation. The left half of the round output is nearly the right half of the previous round output, so Li is equal to Ri- 1. For the right half of the round output, Ri, it is actually a result of the between the left half of the previous round and the F function output. The F function inputs are the right half of the previous round and the subkey Ki. Therefore, Ri = Li- 1, with F of Ri-1 in Ki Let's compare that decryption process of the

Feistel cipher to the encryption process. We see that the structure is identical and that the processes are the same, except for the parameter

values. Essentially, the same hardware or software is used for both encryption and decryption, with just a slight change in how the keys are used. For decryption, the ciphertext is the input and the subkeys Ki are used in the reverse order. That is use Kn the first round, Kn- 1 in the second round, and so on until K1 is used in the last round for decryption. This is a nice feature because it means that there's no need to implement two different algorithms, one for encryption and one for decryption. That is, one implementation can be used for both encryption and decryption but with changes in the subkey inputs and the data inputs. Feistel cipher is a structure that many symmetric block ciphers use. There's some design parameters for Feistel cipher that can vary according to the block cipher design. These design parameters are the block size, how many bits the block can process, which accounts for both the L sub i and the R sub i bits. Key size is another parameter, and key size is the length of the key. The number of rounds is another parameter. The subkey generation algorithm, as well as the round function design. In the rest of the module, we will see the most prominent block cipher based on Feistel cipher, data encryption standard or DES, which has been developed by IBM, including Horst Feistel himself.