SIL765: Networks and System Security

Semester II, 2023-2024

Assignment-1

January 15, 2024

Problem: Basic Cryptanalysis

Background

Advanced Encryption Standard (AES) is the most popular block cipher utilized for the encryption of electronic data. AES performs operations on 16 bytes of input data at a time using a 128-bit key. AES has been shown to be robust against attacks. However, if the secret key is itself weak (i.e., easily predictable), then AES cannot provide the same security guarantee. In this assignment, you will validate this statement.

Problem Details

We will consider three cases.

1. Case-1: The first secret key has 16 least significant bits (LSBs) that are random (i.e., either 0 or 1). The rest of the bits in this secret key are 0s.
2. Case-2: The second secret key has 32 LSBs that are random, and the rest are 0s.
3. Case-3: The third secret key has 48 LSBs that are random, and the rest are 0s.

You will be given access to an encryption oracle, i.e., you can give a plaintext to this oracle and get back the corresponding ciphertext. Your goal is to interact with the encryption oracle, obtain some pairs of plaintext and ciphertext for each case and find the secret key.

Hint: The simplest approach will be the brute force mechanism where you try all combinations of bits for the secret key. The expectation in this assignment is that you come up with an approach that finds the secret key more efficiently than the brute force mechanism.

Given Files

You are being provided with the following two files.

- **get_encrypt** - will be used to interact with the "Encryption Oracle" which is being shared as an assignment on Gradescope.
- **decrypt_text** - will be used to submit your solution for "Assignment 1" on Gradescope.

Interacting with Encryption Oracle

You may need to analyze a large number of pairs of plaintext and the corresponding ciphertext to find the secret key. Please follow the below steps to obtain those pairs.

- Store the `get_encrypt.py` file in a folder named `oracle`.
- In the file `get_encrypt.py`, enter your *IITD Entry Number* in the variable `entry_number`, e.g., "2022JCS2669".
- Enter the plaintext you want to encrypt as a value to the `plaintext` variable in the `enc` function. You can enter any number of plaintexts there.
- Save the `get_encrypt.py` file, zip the `oracle` folder, and submit the zipped folder in "Encryption Oracle" assignment on Gradescope.
- You can note the corresponding ciphertext(s) that are shown on the Gradescope's screen.

Expected Submission

Store the following two (or more) files in a folder named `solution`, zip the folder, and submit the zipped folder in "Assignment 1" assignment on Gradescope.

- **decipher_text.py**:
  - Note that this file is already given to you as part of the assignment. In this file, there are two sets of comments "Do not change this" and "Write your script here". Please follow them while adding your code to facilitate Gradescope to auto-grade your submission.
  - You need to complete the functions "decipher1", "decipher2", and "decipher3" for the three cases.
  - You need to print the ciphertext, deciphered plaintext, and deciphered key in the three cases.
  - Finally, you need to output the deciphered plaintext and deciphered key in the three cases.
- **readme.pdf**: In this file, you should provide a detailed description of your approach for finding the secret key. You should also include a discussion about your code. Finally, you should mention the deciphered secret keys.

Grading

- We will use pairs of randomly generated plaintexts and ciphertexts to check whether you have deciphered the correct secret keys.
- You can upload your submission on Gradescope to see if your code is working correctly. You can make any number of submissions until the deadline.