CS2107 Self-Exploration Activity 5

Notes:

In this Activity 5, you can perform the following tasks:

1. To use `openssl` and Linux commands to perform **hash** operations.
2. To observe how SHA-1 is **vulnerable** to collision attacks.
3. To use `openssl` to perform **MAC** operations.
4. To try out some Python scripts, online tools, and SageMath that can allow you to implement the **RSA** encryption scheme.

Task 1: Using `openssl` and Linux Commands to Perform Hash Operations

You can use the **`openssl` command** to perform **cryptographic hash** operations, such by using SHA-1 and hash functions in the SHA-2 family.

For example, you can run SHA-1 hash function on an input file as follows:

```
openssl dgst -sha1 <input_file>
```

Please try running various different hash functions supported by `openssl`.

Alternatively, you can also use **Linux commands** to perform hash operations, such as by using **`sha1sum`**, **`sha224sum`**, **`sha256sum`**, **`sha384sum`**, and **`sha512sum`** commands. You can check the man page of, for instance, `sha1sum` at: https://man7.org/linux/man-pages/man1/sha1sum.1.html. Also notice that, the "*BUGS*" section of `sha1sum`'s man page puts the following caution:

"Do *not* use the SHA-1 algorithm for security related purposes."

The next task will help convince you why the caution is important.

Task 2: Observing How SHA-1 is Vulnerable to Collision Attacks

As discussed in our lecture, **SHA-1 is vulnerable** to collision attacks. The **SHAttered** attack (https://shattered.io) has broken SHA-1 in practice.
You can confirm the attack **proof-of-concept** made available at the website.

Do download the following two PDF files:

- https://shattered.io/static/shattered-1.pdf
- https://shattered.io/static/shattered-2.pdf

First, do generate the SHA-256 digests of the two PDF files by using either `openssl` or `sha256sum`. Are their two digests the same? Consequently, are the two PDF files the same? If you are curious, you can open the two PDF files and visually compare them.

Next, generate the SHA-1 digests of the two files. Are their two digests now the same? What can you conclude based on the SHA-1 digests of the two files?

Task 3: Using `openssl` to Perform MAC Operations

You can use the **`openssl` command** to perform **MAC** operations. For example, you can perform **HMAC-SHA256** on an input file with your supplied (alphanumeric string) key as follows:

```
openssl dgst -sha256 -hmac <key> <input_file>
```

Please try running various different MACs supported by `openssl`.

Task 4: Trying out Python Scripts, Online Tools, and SageMath that Implement RSA Encryption Scheme

Let's now observe some **Python scripts** that implement various required modules of the RSA encryption scheme. We can refer to the scripts given in the following chapters of the "*Cracking Codes with Python: An Introduction to Building and Breaking Ciphers*" book (available freely online):

- Chapter 22 - **Finding and generating prime numbers**: https://inventwithpython.com/cracking/chapter22.html;
- Chapter 23 - **Generating keys for the public key cipher**: https://inventwithpython.com/cracking/chapter23.html;
- Chapter 24 - **Programming the public key cipher**: https://inventwithpython.com/cracking/chapter24.html.

Do inspect and run the given scripts to perform several steps that are required in performing an RSA encryption.

There are also **online tools** that allow you to perform RSA encryptions, such as:

- "RSA Encryption Decryption" tool: https://8gwifi.org/rsafunctions.jsp;
- "RSA Calculator" tool: https://www.cs.drexel.edu/~jpopyack/Courses/CSP/Fa17/notes/10.1_Cryptography/RSAWorksheetv4e.html.

Lastly, you can also experiment with **SageMath** to implement the RSA scheme. The SageMath system (https://www.sagemath.org/) is a free open-source mathematics software system licensed under the GPL. It aims to provide a free open source alternative to Magma, Maple, Mathematica and Matlab, by leveraging many existing open-source packages, including NumPy, SciPy, matplotlib, and R.

To perform **RSA encryptions using SageMath**, you can refer to the following tutorial: https://doc.sagemath.org/html/en/thematic_tutorials/numtheory_rsa.html.