Parallel Project Report

CUDA Parallel Implementation of Brute Force Cracking Password via SHA-1

BY

Miss Kanrawee	**Chiamsakul**	**6188049**
Mr. Tharit	**Chantanalertvilai**	**6188068**

Present

Assoc. Prof. Dr. Sudsanguan Ngamsuriyaroj

ITCS443 Parallel and Distributed Systems

Faculty of Information and Communication Technology

Mahidol University

2020

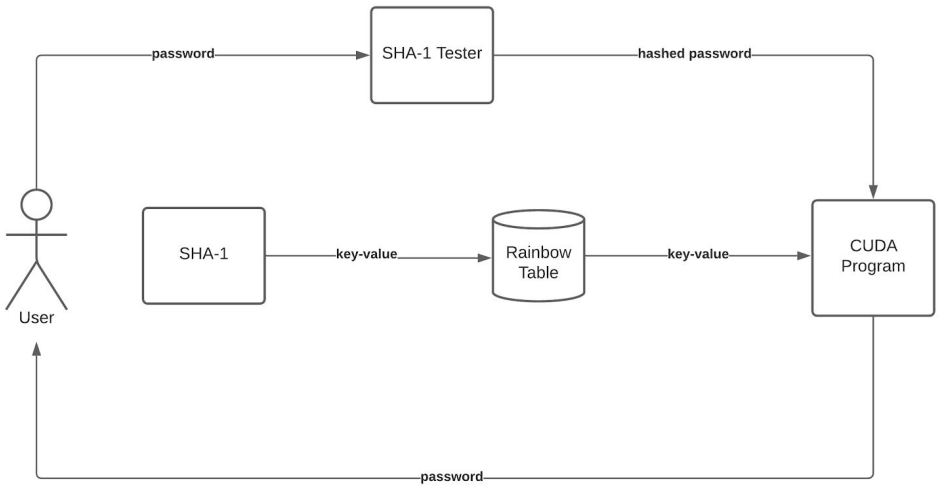
Objective

The objective of this project is to create a program that applies the SHA-1 algorithm to hash the password number from “00000000” to “99999999” into hexadecimal value and input the hash value into the CUDA program to find the input value of this hash value. Since the SHA-1 is a one-way cryptographic hash function, the output of the function can’t be decode, so we need to create a rainbow table that records the key-value of all the given numbers and their hash value in advance.

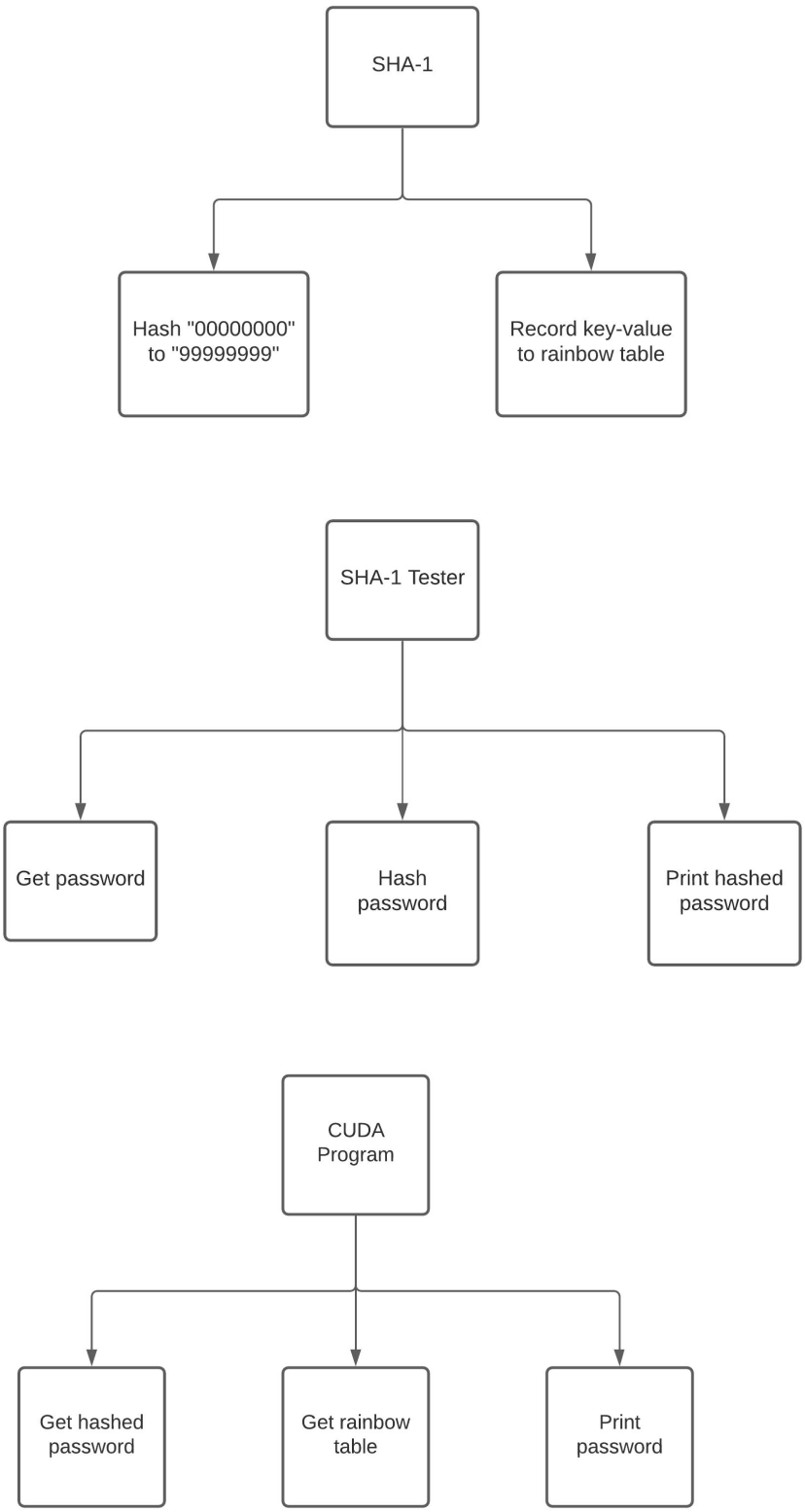
The System Overview

SHA-1, or Secure Hash Algorithm 1, is a cryptographic hash function designed by the United States National Security Agency. The function takes an input and returns a 20-byte hash value known as a message digest.

In our project, SHA-1 is a c program that hashes the password from “00000000” to “99999999” and records it to the database as a rainbow table, which is a text file that keeps the key-value of passwords and hash values, separated by a comma. This code is referenced from https://www.ipa.go.jp/security/rfc/RFC3174EN.html#71. SHA-1 Tester is another c program that receives the user input password and prints the hash value to the user. Then, the hash value (hashed password) will be the input of our CUDA program, which will look up the hash value in the rainbow table and prints the corresponding password back to the user. The system overview diagram is shown below.



The Structure Chart Showing All Functions



The Sample Inputs and Outputs of the Project

Correct input:

```
Password (00000000 - 99999999): 00000000
70352f4161eda4ff3c32294af068ba70c3b38b
```

```
Password (00000000 - 99999999): 12345678
7c222fb2927d828af22f592134e8932480637cd
```

```
Password (00000000 - 99999999): 99999999
d528fca3b163c0573e88b528544bec28ecf185
```

Incorrect input (the program will ask the user to input again):

```
Password (00000000 - 99999999): 1234567890123
Password (00000000 - 99999999): 02587
Password (00000000 - 99999999): 000000
Password (00000000 - 99999999): 55555555
19dd466e43cdbd3833abc0609eba6d8786f9b342
```

Rainbow Table: