Project on the topic SHA-1

Description :-

This project is all about SHA-1 (Secure Hash Algorithm). I have implemented this on python and created a program which hashes your password (Basically it will convert your password in a unique string of texts). And also, I have made SHA-1 brute-force program which will convert the hashes into password.

So,I have made two programs, One will convert your password into hashes and other one will convert the hashes into password.

Implementation :-

Program 1 :-

Description :-

As the first program is implemented using python, python comes with some libraries like *hashlib*, *bcrypt* and many more but in this program we haven't used any libraries, I have maked this program with pure python code without any libraries with SHA-1 logic.

Implementation Process :-

Steps :-

1. First, I processed a chunk of data.. the message length and chunk size.
2. Second, I took initial hash values (Digest Variables).
3. Third, The pre-processing will start and we will break the chunks into sixteen four bytes big-endian words and then extend the sixteen four bytes words into eighty four bytes words.
4. Fourth, At last, after all the process we will print the hashed value.

I have explained the implementation as comments in the code as well.

Example Output :-

So, Below you can see the output of the program.
The program is running fine and also generating hash values.

```
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1.py"
Enter your password : yuvi
Hash is :  f313de9b1ee393f2d5dc695747c79cc984a125ad
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1.py"
Enter your password : anmol
Hash is :  63973ab5058977b0ef4f258e0abf98a3807413fa
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1.py"
Enter your password : anisha
Hash is :  08a831b82f10f8aa4dff199424a0fd2245adb540
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1.py"
Enter your password : hiranmoy
Hash is :  a813e6c50fd56039eea769a01dd7a8b0be856e78
PS C:\Users\KIIT\Desktop\Project>
```

Program 2 :-

Description :-

So, The second program is all about SHA-1 hash cracking via brute-force attack. To implement this, I have used some libraries like *urllib.request* and *hashlib.*

urllib.request - It is a python module for fetching URLs. It offers a very simple interface in the form of the *urlopen* function. So, in this program I will fetch the URL from the urlopen function.

hashlib - It is a module which implements a common interface to many different secure hash and message digest algorithms. Included are the FIPS secure hash algorithms SHA1, SHA224, SHA256, SHA384, and SHA512 (defined in FIPS 180-2) as well as RSA's MD5 algorithm.

Implementation Process :-

Steps :-

1. First, I will import the hashlib and urllib.request library.
2. Second, I will take input from the user.
3. Third, I will open a file full of password guesses. (I have maked my own URL of passwords)
4. Fourth, I will take a guess from the list of passwords I opened, and split it by line.

5. Fifth, I will hash the guess we took from the password list so we can compare it to the hash the user gave us.
6. Sixth, I will compare the hash the user gave us to the hashed version of the password guess and determine if they are equal.
7. Seventh, I will tell the program what to do if the password guess matches, which is to print the current guess and quit the program. And I will also tell the program what to do if the password guess don't match, which is to return to step 3 to get a new password from the list.
8. Eighth, At last I will tell the program what to do if we get all the way through the password list without finding a match.
I have explained the implementation as comments in the code as well.

Example Output :-

When I gave some hash to test the program, I found the password that was used to create a hash, allowing us to reverse the *one way* SHA-1 hash as shown below.

```
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1bruteforcer.py"
Enter the hash to crack : f313de9b1ee393f2d5dc695747c79cc984a125ad
Password guessing..  anmol  doesn't match, trying next..
Password guessing..  anisha  doesn't match, trying next..
Password guessing..  hiranmoy  doesn't match, trying next..
The password is :  yuvi
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1bruteforcer.py"
Enter the hash to crack : 63973ab5058977b0ef4f258e0abf98a3807413fa
The password is :  anmol
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1bruteforcer.py"
Enter the hash to crack : 08a831b82f10f8aa4dff199424a0fd2245adb540
Password guessing..  anmol  doesn't match, trying next..
The password is :  anisha
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1bruteforcer.py"
Enter the hash to crack : a813e6c50fd56039eea769a01dd7a8b0be856e78
Password guessing..  anmol  doesn't match, trying next..
Password guessing..  anisha  doesn't match, trying next..
The password is :  hiranmoy
PS C:\Users\KIIT\Desktop\Project>
```

And, When I gave the hash which was not there, I did not find a matching password in the list as you can see below.

```
PS C:\Users\KIIT\Desktop\Project> python -u "c:\Users\KIIT\Desktop\Project\sha1bruteforcer.py"
Enter the hash to crack : 41249e26fc1ec5fc0fcf2d554006bfe1c4ae22d0
Password guessing..  anmol  doesn't match, trying next..
Password guessing..  anisha  doesn't match, trying next..
Password guessing..  hiranmoy  doesn't match, trying next..
Password guessing..  yuvi  doesn't match, trying next..
Password guessing..    doesn't match, trying next..
Password not in the file.. try again
PS C:\Users\KIIT\Desktop\Project>
```

Conclusion :-

So, Both of our programs are running well and generating output with no errors. I learned a lot while doing this project and it was fun to do. Also, learned so many things.

Contributor :-

Yuvraj Kaushal