Week 6 Practical: *Hashing*

Dr. Qublai K. Ali Mirza

University of Gloucestershire

qalimirza@glos.ac.uk

UNIVERSITY OF GLOUCESTERSHIRE

Overview

Recap

- Last week we looked at *Advanced Encryption System (AES)* and *TwoFish*
- We also looked at how each of them works and how they can be implemented in CryptTool 2
- This week we will be looking at *MD5* and *SHA* hashing algorithms

Overview

- Designed by Ron Rivest in 1991
- Based on a non-linear function F which involves
  - Modular addition
  - Left rotation
- A flaw was identified in 1996
- Considered no longer collision resistant by 2004

How it works

- Step 1: The input message is *padded* so that its length is congruent 448 *mod* 512
- Step 2: A 64-bit representation of the message length is appended, making the resulting message length an exact multiple of 512
- Step 3: Initialise MD buffer, which consists of 4 32-bit registers
- Step 4: Process the message in 16 32-bit words, for a total of 64 rounds

MD5 in CryptTool 2

- From the *CryptTool 2* startup menu, click *Hash Functions*
- Then click *MD5*
- In the*Input* text box, type in: `This is a test string for MD5`
- Finally click on the *Play* button to execute

MD5 in CryptTool 2

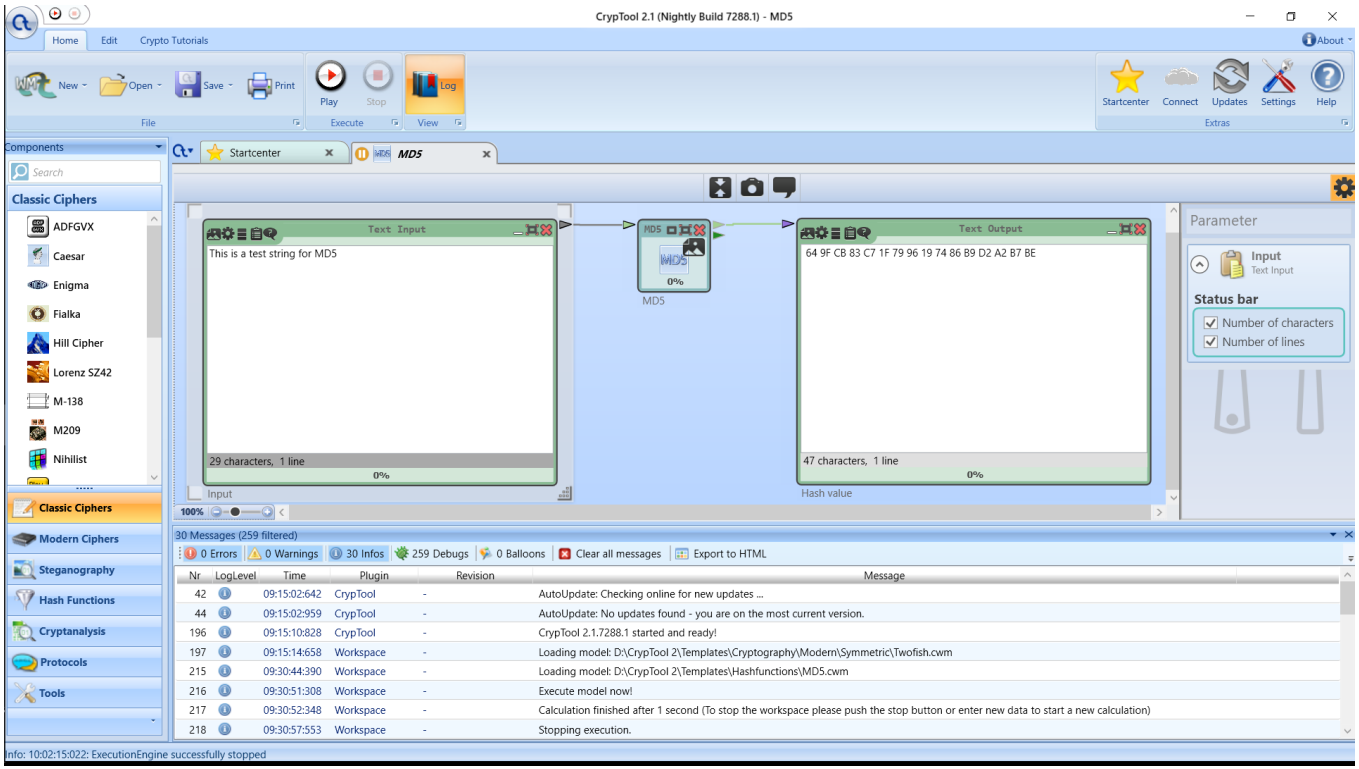

Figure: MD5 hashing in CryptTool 2

MD5 collision

- From the *CryptTool 2* startup menu, click *Hash Functions*
- Then click *MD5 Collision Finder*
- In the *Prefix* text box, type in: `MD5 is used to create a digital signature that ensures integrity`
- Then in the *Suffix* text box, type in: `Hello World`
- Finally click on the *Play* button to execute

MD5 collision in CryptTool 2

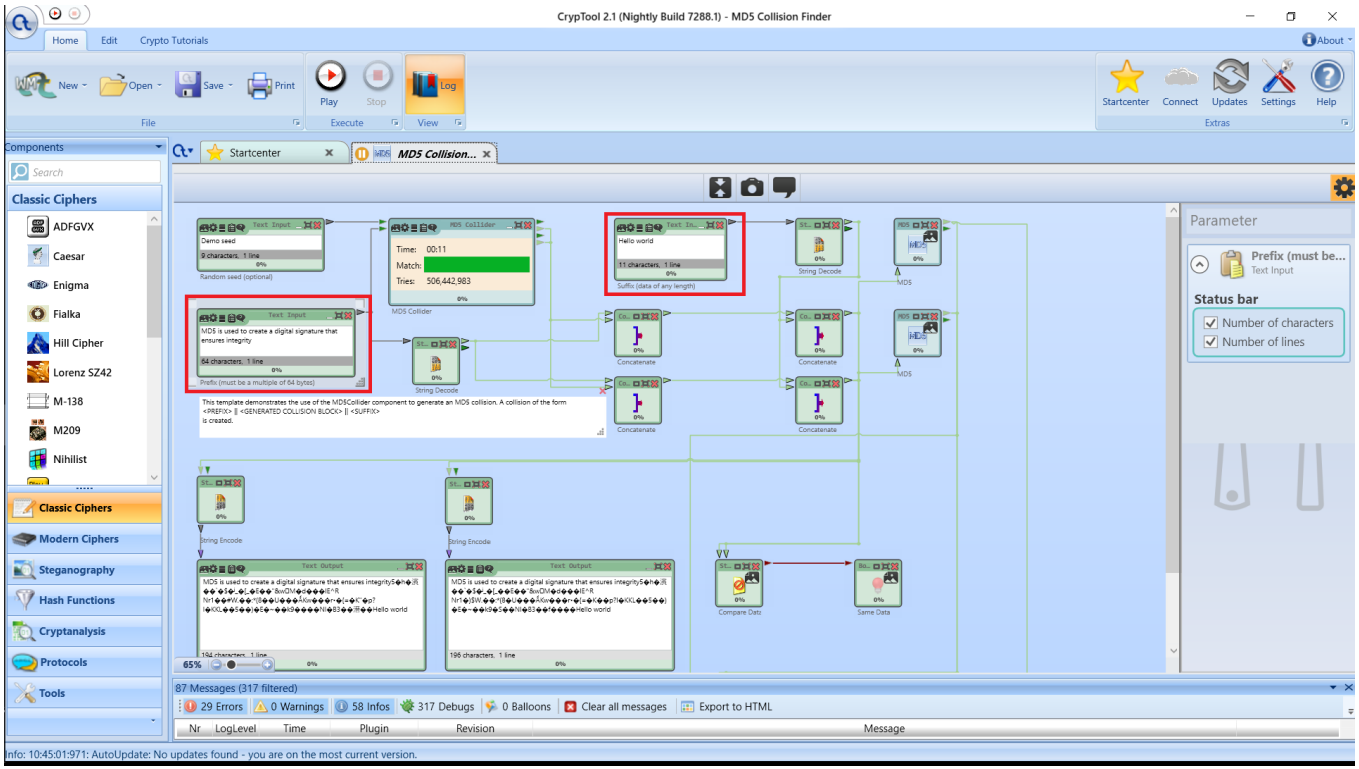

Figure: MD5 Collision in CryptTool 2

UNIVERSITY OF GLOUCESTERSHIRE

Overview

- Published in 1995
- Operates in blocks of 512 bits
- Total number of rounds: 80
- Produces a 160-bit hash value
- Max. message length: 2^{64-1} bits

SHA-1 on CryptTool 2

- From the *CryptTool 2* startup menu, click *Hash Functions*
- Then click *SHA-1*
- In the *Prefix* text box, type in: `Secure Hash Authentication 1 (SHA-1) is a cryptographic hash function that takes an input and produces a 160-bit hash value called a message digest. It is usually represented as a 40-digit long hexadecimal number.`
- Finally click on the *Play* button to execute

SHA-1 on CryptTool 2

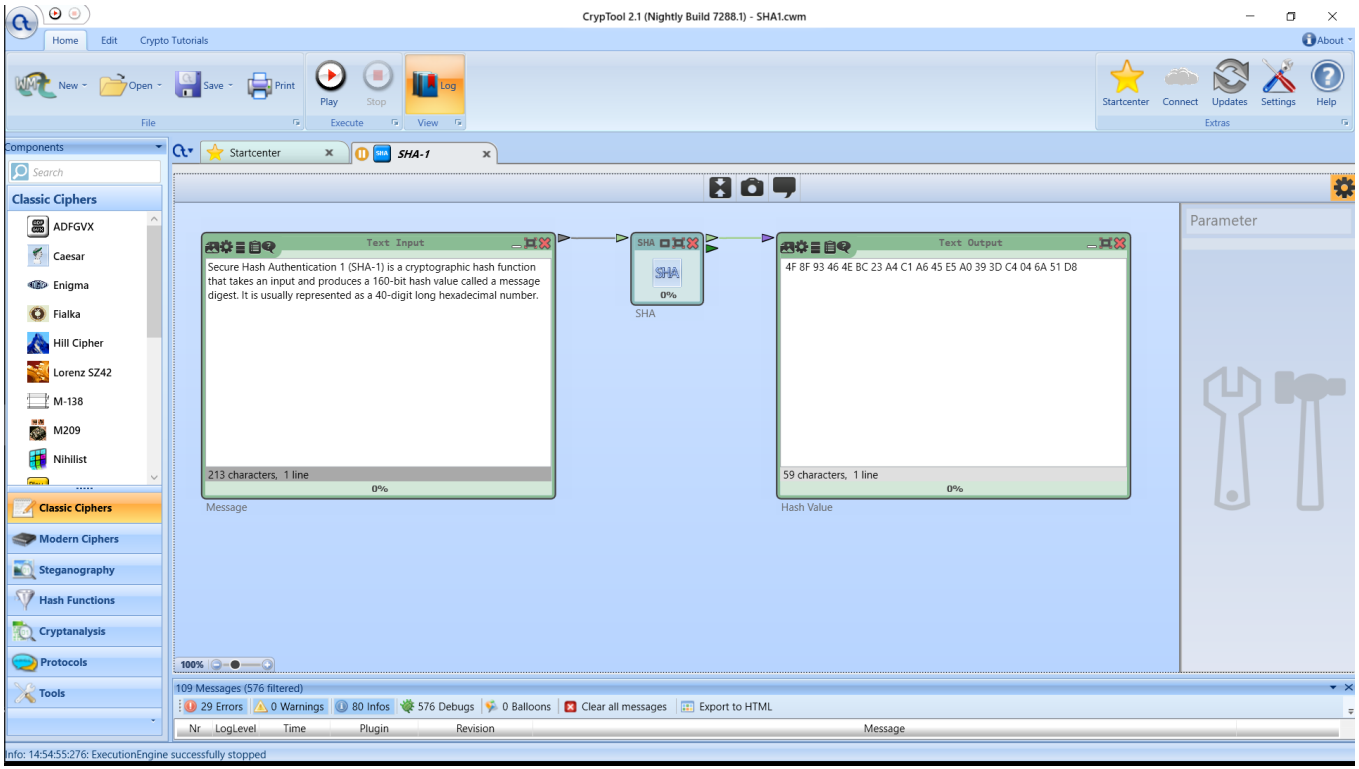

Figure: SHA-1 on CryptTool 2

Overview

- In real-world environments, passwords are usually stored as hash strings
- They can be hashed using either *MD5* or *SHA-** approaches
- However it does not mean that we can't crack them using *hash collision*
- For this session we will be using *dictionary attack* to break a password

Getting a hash string

- For this section, we will be using CryptTool 2 and *QuickHash*
- You can download QuickHash here
- Then open *QuickHash* and perform the following:
  - Make sure `SHA-256` is selected in *Algorithms*
  - Type in: `apple` in the *Text Hashing* text box
- Then copy in the resulting hash string generated at the bottom

QuickHash configuration

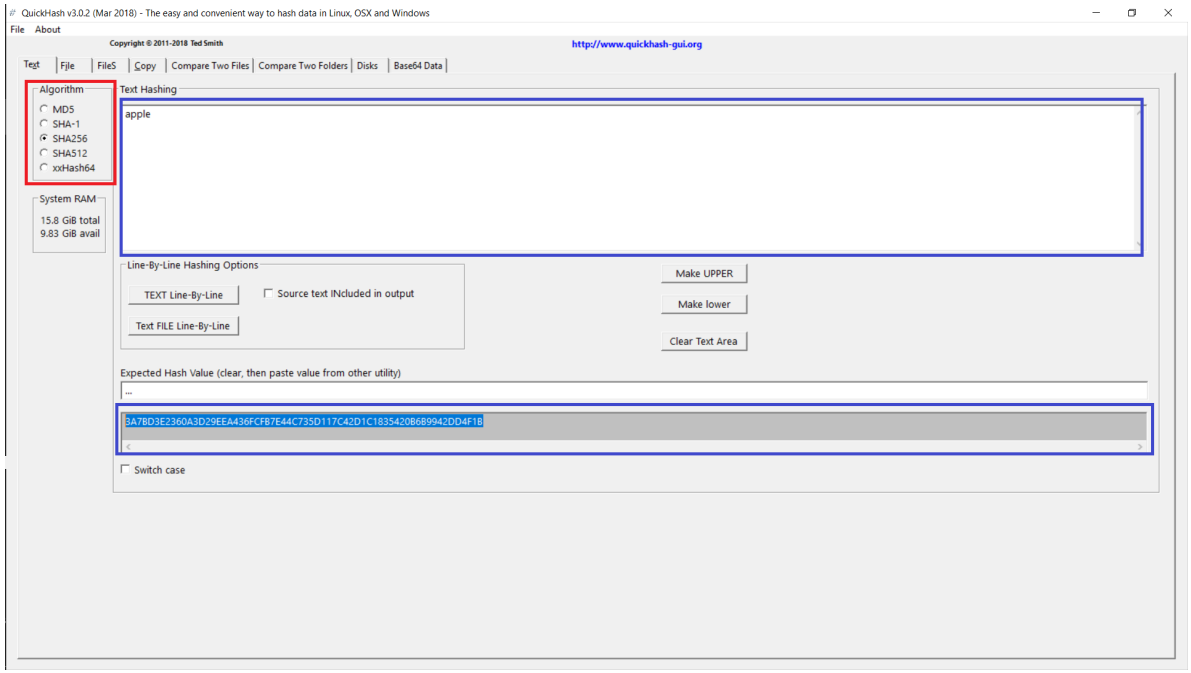

Figure: Hash generation using *QuickHash*

Password cracking in CryptTool 2

- From the *CryptTool 2* startup menu, click *Hash Functions*
- Then click *Dictionary Attack*
- In the *Test password* text box, paste in the hash value obtained from *Quick Hash*
- Finally click on the *Play* button to execute
- It will take a while, but you will eventually get your password

Password cracking in CryptTool 2

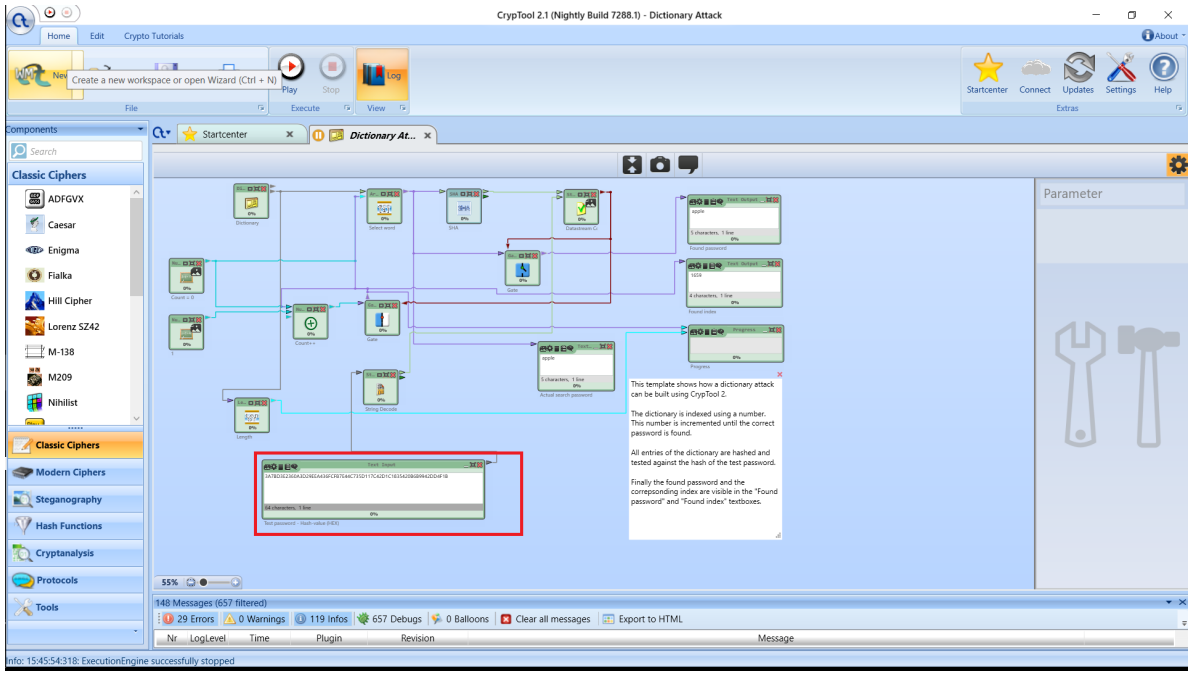

Figure: Password cracking in CryptTool 2

Bringing it all together

- We looked at *MD5* and *SHA*
- We also looked at how to crack hashed passwords in CryptTool 2
- Next week: *Asymmetric Encryption*

Post-sessional work

- Create a CryptTool project which accepts a plaintext, and
  - Hashes it using a hashing algorithm of your choice
  - Encrypts it using either DES or AES
  - Recover the original plaintext back
  - **NB:** For simplicity, you might want to use a dictionary word as a password

Q & A