Lab – Compare Data with a Hash .

- Objectives

Use a hashing program to verify the integrity of data.

- Background / Scenario

It is important to understand when data has been corrupted or it has been tampered with. A hashing program can be used to verify if data has changed, or if it has remained the same. A hashing program performs a hash function on data or a file, which returns a (usually much shorter) value. There are many different hash functions, some very simple and some very complex. When the same hash is performed on the same data, the value that is returned is always the same. If any change is performed on the data, the hash value returned will be different.

- Required Resources
- PC with Internet access
    - Create a Text file
        1. Search your computer for the Notepad program and open it.
        2. Type some text in the program.

File 1.txt:

Cybersecurity is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access. In a computing context, security includes both cybersecurity and physical security.

File 2.txt:

Cyberwarfare is computer- or network-based conflict involving politically motivated attacks by a nation-state on another nation-state. In these types of attacks, nation-state actors attempt to disrupt the activities of organizations or nation-states, especially for strategic or military purposes and cyberespionage.

File 3.txt:

Information Assurance (IA) refers to the steps involved in protecting information systems, like computer systems and networks. There are commonly five terms associated with the definition of information assurance:

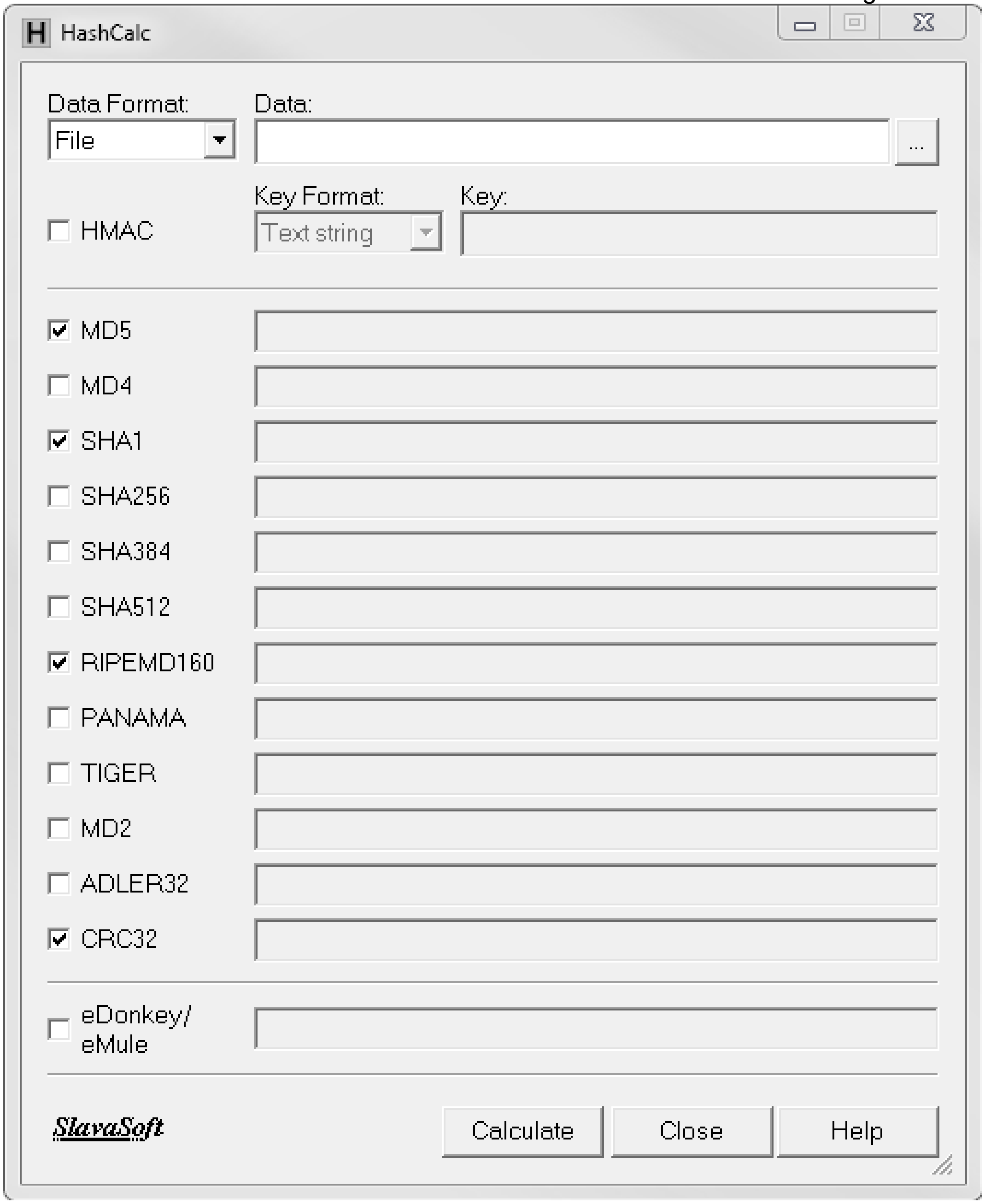
- Integrity
- Availability
- Authentication
- Confidentiality
- Nonrepudiation

IA is a field in and of itself. It can be thought of as a specialty of Information Technology (IT), because an IA specialist must have a thorough understanding of IT and how information systems work and are interconnected. With all of the threats that are now common in the IT world, such as viruses, worms, phishing attacks, social engineering, identity theft and more, a focus on protection against these threats is required. IA is that focus.

1. Choose **File > Save**. Save the File into different filename given.
2. Navigate to **Desktop**.
3. Type **Hash** in the **File name:** field, and click **Save**.

- Install HashCalc
    1. Open a web browser and navigate to http://www.slavasoft.com/download.htmLinks to an external site..
    2. Click **Download** in the **HashCalc 2.02**
    3. Open the **zip** file and run the **setup.exe** file inside.
    4. Follow the installation wizard to install HashCalc. Ask your instructor to help if you have any questions about the installation.
    5. Click **Finish** on the last screen, and close the **README** file if it opened. You may read the file if you wish.

6. HashCalc is now installed and running.



HashCalc

Data Format: File

Data:

□ HMAC Key Format: Text string Key:

☑ MD5

□ MD4

☑ SHA1

□ SHA256

□ SHA384

□ SHA512

☑ RIPEMD160

□ PANAMA

□ TIGER

□ MD2

□ ADLER32

☑ CRC32

□ eDonkey/ eMule

- Calculate a hash of the text file you have created
  1. Set the following items in HashCalc:
     - Data Format: **File**.
     - Data: Click the **…** button next to the Data field, navigate to the **Desktop** and choose the **txt** file.
     - Uncheck **HMAC**.
     - Uncheck all hash types except **MD5**.
  2. Click the **Calculate**

What is the value next to **MD5**?

d743632325841e3985aa18279d152804

- Make a change to the file 1.txt file
    1. Navigate to the **Desktop** and open the **txt** file.
    2. Make a minor change to the text, such as deleting a letter, or adding a space or period.
    3. Click **File > Save**, and close **Notepad**.
- Calculate a new hash of the Hash.txt file
    1. Click the **Calculate** button in HashCalc again.

What is the value next to **MD5**?

761236b51538245db5067123da27ff59

Is the value different from the value recorded in Step 3?

Yes

1. Place a check mark next to all of the hash types.
2. Click **Calculate**.
3. Notice that many of the hash types create a hash of a different length. Why?

The hashes use different bits to create a hash value.

Reflection:

1. What is Hashing and how does it work?

It is an algorithm that takes input or in this case, an arbitrary amount of data, and creates a fixed-size output called hash value. This value is an enciphered text that can only be unlocked/deciphered if you know the hash function. Hashing secures data by encoding it mathematically through a certain algorithm that only the implementer may know

2. After calculating the 3 files, what insights have you come up after using hashing?

The hash value of a certain file will always vary meaning it can be used in data protection. Hashing can be used to secure personal information such as passwords and instead of obtaining the password itself, the hash key can be used to verify the user. Although hashing is very beneficial, certain cybercriminals with fast hardware can easily crack hashed credentials. Some hash functions have collisions and MD5 is proven to contain known collisions.

3. In what way hashing can help in protecting files?

Hashing can help validate the authenticity and integrity of certain types of input. I have seen hashing being used in plaintext passwords in databases like MySQL but can also be used to validate files if they have not been tampered with.

4. Define and give the respective uses of MD5, SHA1, RIPEMD and CRC 32.

MD5 - was originally designed for use as a secure cryptographic hash algorithm for authenticating digital signatures. But MD5 has been deprecated for uses other than as a noncryptographic checksum to verify data integrity and detect unintentional data corruption.

SHA1 - It's most often used to verify a file has been unaltered. This is done by producing a checksum before the file has been transmitted, and then again once it reaches its destination. Despite it being cryptographically broken, it is still being widely used.

RIPEMD - a cryptographic hash function based upon the Merkle–Damgård construction. It is used in the Bitcoin standard. It moderates the creation and management of addresses, and is also used for transaction verification.

CRC 32 - is an error-detecting code often used for detection of accidental changes to data. Encoding the same data string using CRC32 will always result in the same hash output, thus CRC32 is sometimes used as a hash algorithm for file integrity checks. It is commonly used in digital networks and storage devices to detect accidental changes to digital data.