PRACTICAL: 8

AIM:

An organization has transferred sensitive financial documents between departments over a network and wants to ensure that the files remain unchanged during transit. The IT team is tasked with using HashCalc and MD5 Calculator to confirm the integrity of these files by comparing hash values generated before and after the transfer. Verify the integrity of critical data files by generating and comparing hash values using HashCalc and MD5 Calculator, ensuring no tampering or corruption has occurred during transmission or storage.

THEORY:

The MD5 (message-digest algorithm) hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

The MD5 hash function 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.

Although originally designed as a cryptographic message authentication code algorithm for use on the internet, MD5 hashing is no longer considered reliable for use as a cryptographic checksum because security experts have demonstrated techniques capable of easily producing MD5 collisions on commercial off-the-shelf computers. An encryption collision means two files have the same hash. Hash functions are used for message security, password security, computer forensics and cryptocurrency.

Ronald Rivest, founder of RSA Data Security LLC and professor at Massachusetts Institute of Technology, designed MD5 in 1991 as an improvement to a prior message-digest algorithm, MD4. Describing it in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321, "The MD5 Message-Digest Algorithm," he wrote:

The algorithm takes as input a message of arbitrary length and produces as output a 128-bit 'fingerprint' or 'message digest' of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be 'compressed' in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

IETF suggests MD5 hashing can still be used for integrity protection, noting: "Where the MD5 checksum is used in line with the protocol solely to protect against errors, an MD5 checksum is still an acceptable use." However, it added that "any application and protocol that employs MD5 for any purpose needs to clearly state the expected security services from their use of MD5."

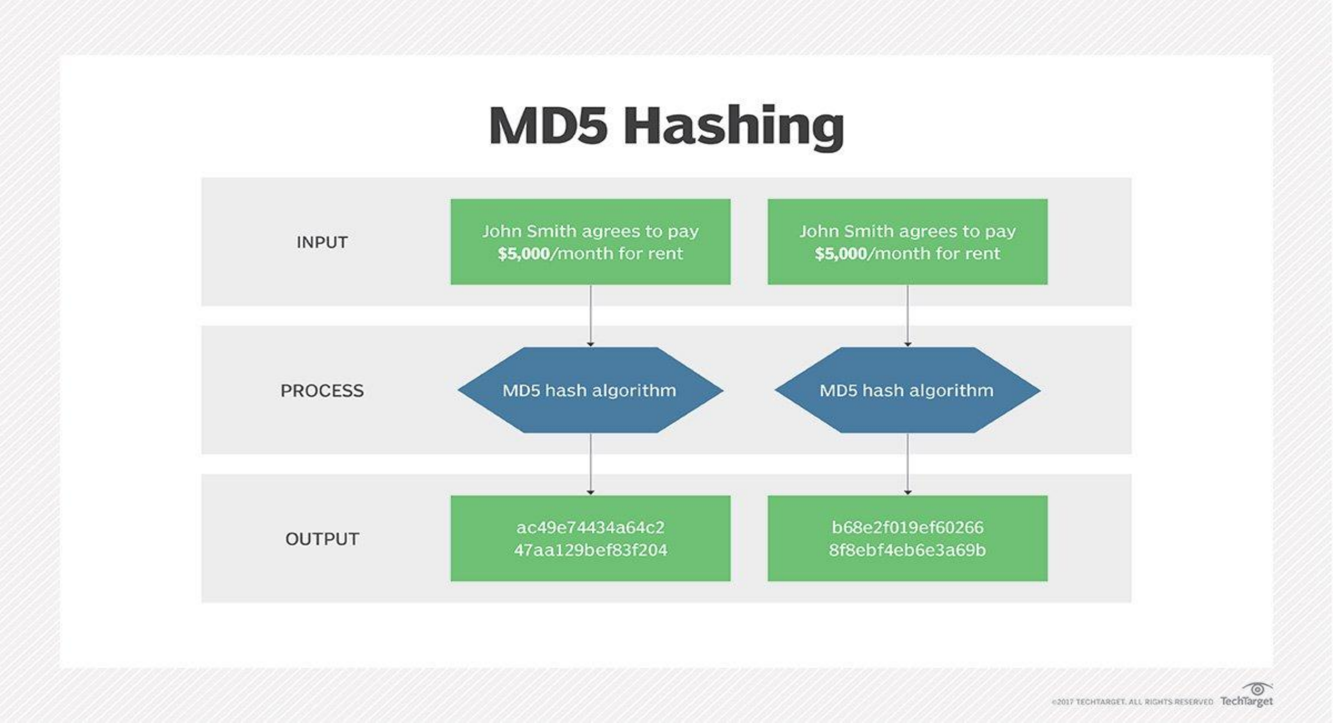

MD5 Hashing
INPUT
John Smith agrees to pay $5,000/month for rent
John Smith agrees to pay $5,000/month for rent
PROCESS
MD5 hash algorithm
MD5 hash algorithm
OUTPUT
ac49e74434a64c2 47aa129bef83f204
b68e2f019ef60266 8f8ebf4eb6e3a69b
©2017 TECHTARGET. ALL RIGHTS RESERVED TechTarget

Figure 1: MD5

Message digests, also known as *hash functions*, are one-way functions; they accept a message of any size as input and produce as output a fixed-length message digest.

MD5 is the third message-digest algorithm Rivest created. MD2, MD4 and MD5 have similar structures, but MD2 was optimized for 8-bit machines, in comparison with the two later algorithms, which are designed for 32-bit machines. The MD5 algorithm is an extension of MD4, which the critical review found to be fast but potentially insecure. In comparison, MD5 is not quite as fast as the MD4 algorithm, but offered much more assurance of data security.

The MD5 message-digest hashing algorithm processes data in 512-bit strings, broken down into 16 words composed of 32 bits each. The output from MD5 is a 128-bit message-digest value.

Computation of the MD5 digest value is performed in separate stages that process each 512-bit block of data along with the value computed in the preceding stage. The first stage begins with the message-digest values initialized using consecutive hexadecimal numerical values. Each stage includes four message-digest passes, which manipulate values in the current data block and values processed from the previous block. The final value computed from the last block becomes the MD5 digest for that block.

The goal of any message-digest function is to produce digests that appear to be random. To be considered cryptographically secure, the hash function should meet two requirements:

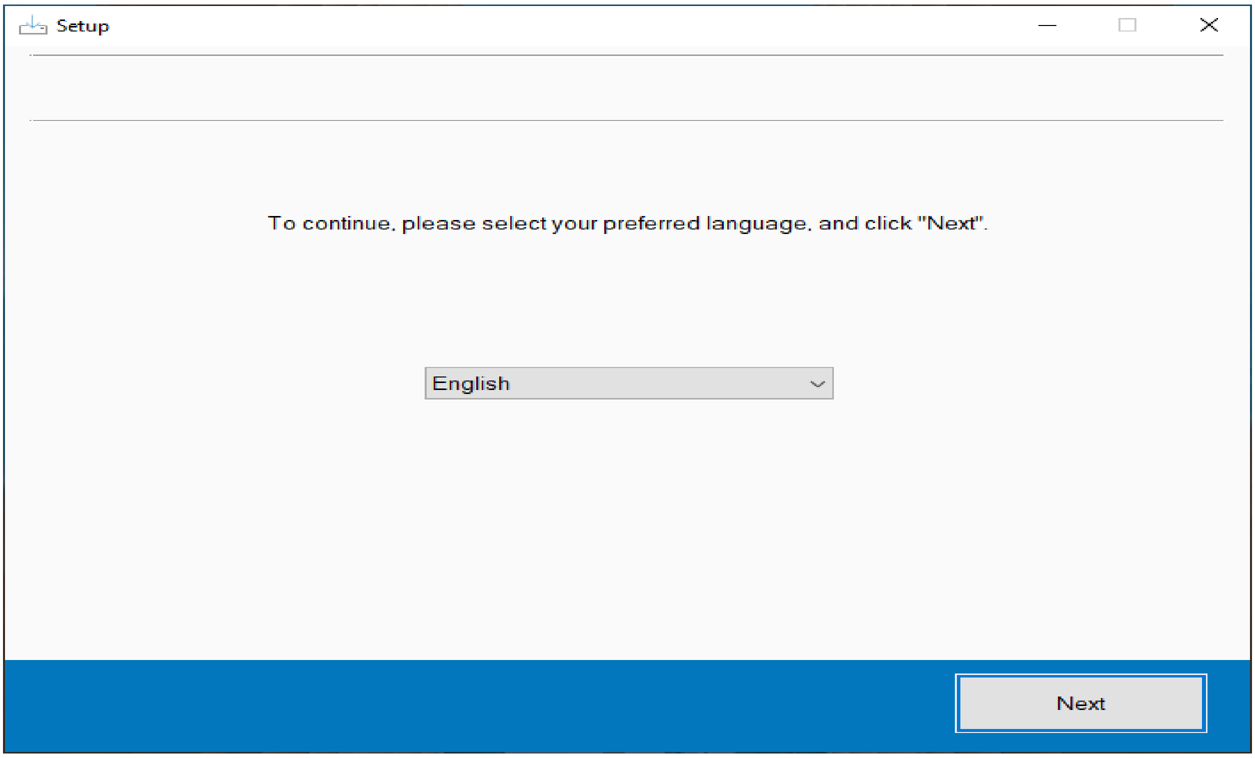
1. It is impossible for an attacker to generate a message matching a specific hash value.
2. It is impossible for an attacker to create two messages that produce the same hash value.

MD5 hashes are no longer considered cryptographically secure methods and should not be used for cryptographic authentication, according to IETF.

CODE:

N/A

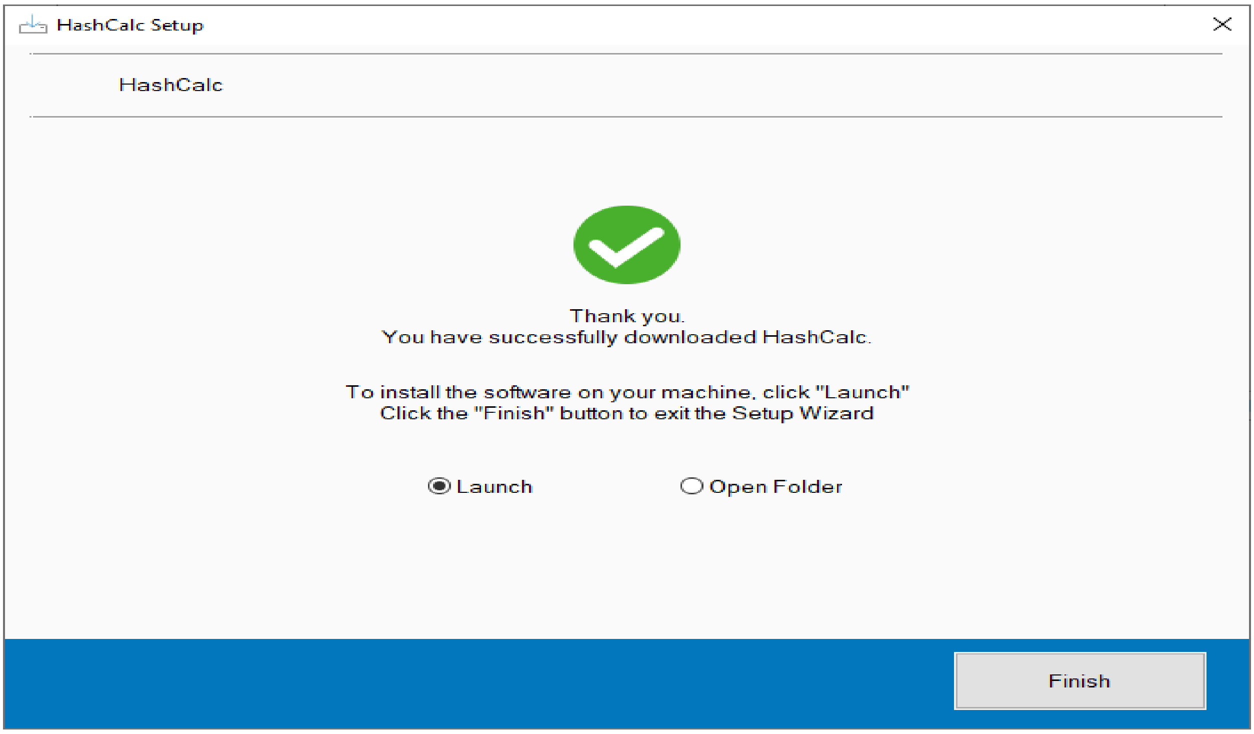
OUTPUT:



Setup

To continue, please select your preferred language, and click "Next".

Figure 2:Start setting up HashCalc



HashCalc Setup

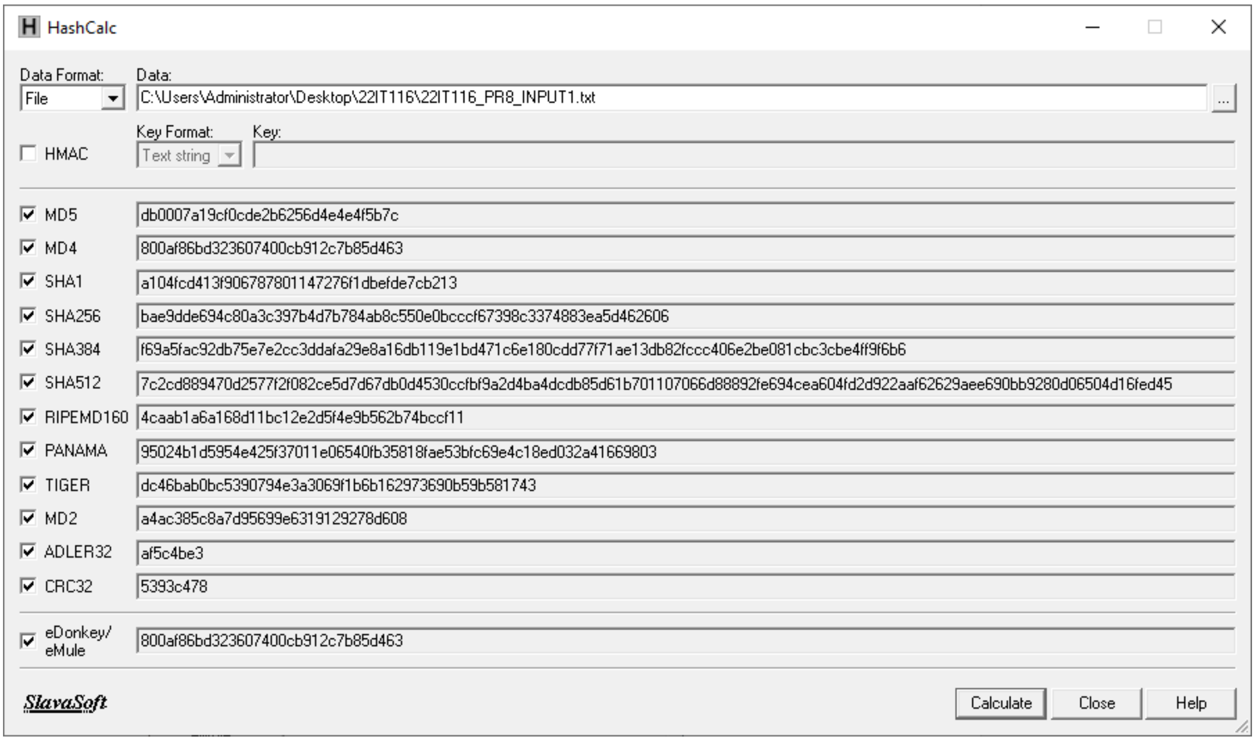
HashCalc

Thank you.
You have successfully downloaded HashCalc.

To install the software on your machine, click "Launch"
Click the "Finish" button to exit the Setup Wizard

● Launch ○ Open Folder

Figure 3:Finish setup of HashCalc and launch it



Data Format: File

Data: C:\Users\Administrator\Desktop\22IT116\22IT116_PR8_INPUT1.txt

□ HMAC

Key Format: Text string

Key:

☑ MD5 db0007a19cf0cde2b6256d4e4e4f5b7c

☑ MD4 800af86bd323607400cb912c7b85d463

☑ SHA1 a104fcd413f906787801147276f1dbefde7cb213

☑ SHA256 bae9dde694c80a3c397b4d7b784ab8c550e0bcccf67398c3374883ea5d462606

☑ SHA384 f69a5fac92db75e7e2cc3ddafa29e8a16db119e1bd471c6e180cdd77f71ae13db82fccc406e2be081cbc3cbe4ff9f6b6

☑ SHA512 7c2cd889470d2577f2f082ce5d7d67db0d4530ccfbf9a2d4ba4dcdb85d61b701107066d88892fe694cea604fd2d922aaf62629aee690bb9280d06504d16fed45

☑ RIPEMD160 4caab1a6a168d11bc12e2d5f4e9b562b74bccf11

☑ PANAMA 95024b1d5954e425f37011e06540fb35818fae53bfc69e4c18ed032a41669803

☑ TIGER dc46bab0bc5390794e3a3069f1b6b162973690b59b581743

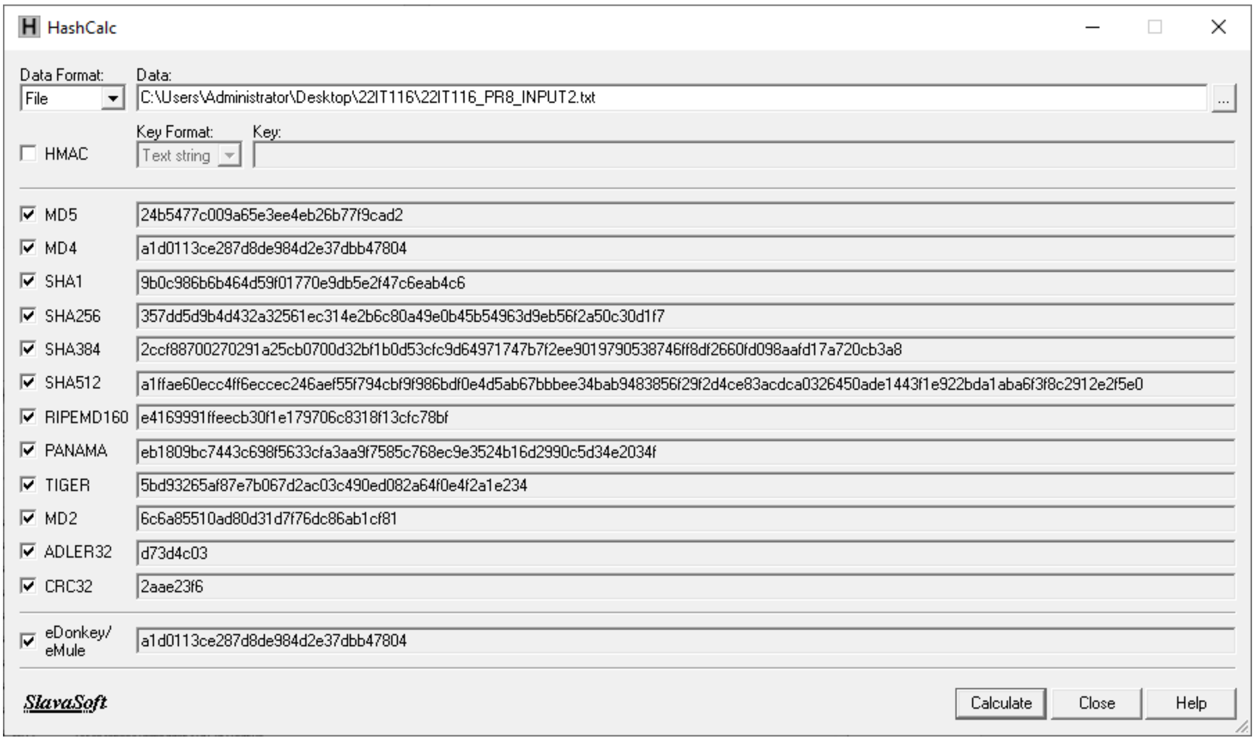
☑ MD2 a4ac385c8a7d95699e6319129278d608

☑ ADLER32 af5c4be3

☑ CRC32 5393c478

☑ eDonkey/ eMule 800af86bd323607400cb912c7b85d463

Figure 4:Hash values for Input1.txt file



Data Format: File

Data: C:\Users\Administrator\Desktop\22IT116\22IT116_PR8_INPUT2.txt

□ HMAC

Key Format: Text string

Key:

☑ MD5 24b5477c009a65e3ee4eb26b77f9cad2

☑ MD4 a1d0113ce287d8de984d2e37dbb47804

☑ SHA1 9b0c986b6b464d59f01770e9db5e2f47c6eab4c6

☑ SHA256 357dd5d9b4d432a32561ec314e2b6c80a49e0b45b54963d9eb56f2a50c30d1f7

☑ SHA384 2ccf88700270291a25cb0700d32bf1b0d53cfc9d64971747b7f2ee9019790538746ff8df2660fd098aafd17a720cb3a8

☑ SHA512 a1ffae60ecc4ff6eccec246aef55f794cbf9f986bdf0e4d5ab67bbbee34bab9483856f29f2d4ce83acdca0326450ade1443f1e922bda1aba6f3f8c2912e2f5e0

☑ RIPEMD160 e4169991ffeecb30f1e179706c8318f13cfc78bf

☑ PANAMA eb1809bc7443c698f5633cfa3aa9f7585c768ec9e3524b16d2990c5d34e2034f

☑ TIGER 5bd93265af87e7b067d2ac03c490ed082a64f0e4f2a1e234

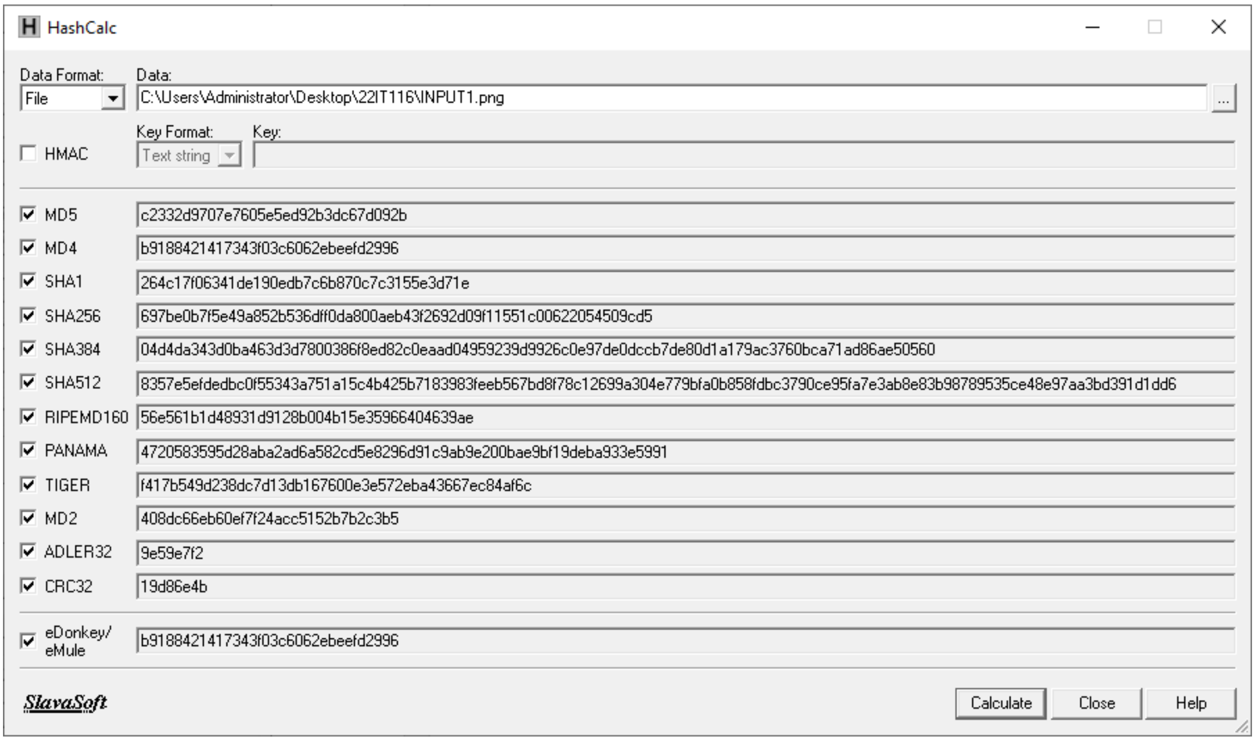
☑ MD2 6c6a85510ad80d31d7f76dc86ab1cf81

☑ ADLER32 d73d4c03

☑ CRC32 2aae23f6

☑ eDonkey/ eMule a1d0113ce287d8de984d2e37dbb47804

Figure 5:Hash values for Input2.txt file



Data Format: File

Data: C:\Users\Administrator\Desktop\22IT116\INPUT1.png

□ HMAC Key Format: Text string Key:

☑ MD5 c2332d9707e7605e5ed92b3dc67d092b

☑ MD4 b9188421417343f03c6062ebeefd2996

☑ SHA1 264c17f06341de190edb7c6b870c7c3155e3d71e

☑ SHA256 697be0b7f5e49a852b536dff0da800aeb43f2692d09f11551c00622054509cd5

☑ SHA384 04d4da343d0ba463d3d7800386f8ed82c0eaad04959239d9926c0e97de0dccb7de80d1a179ac3760bca71ad86ae50560

☑ SHA512 8357e5efdedbc0f55343a751a15c4b425b7183983feeb567bd8f78c12699a304e779bfa0b858fdbc3790ce95fa7e3ab8e83b98789535ce48e97aa3bd391d1dd6

☑ RIPEMD160 56e561b1d48931d9128b004b15e35966404639ae

☑ PANAMA 4720583595d28aba2ad6a582cd5e8296d91c9ab9e200bae9bf19deba933e5991

☑ TIGER f417b549d238dc7d13db167600e3e572eba43667ec84af6c

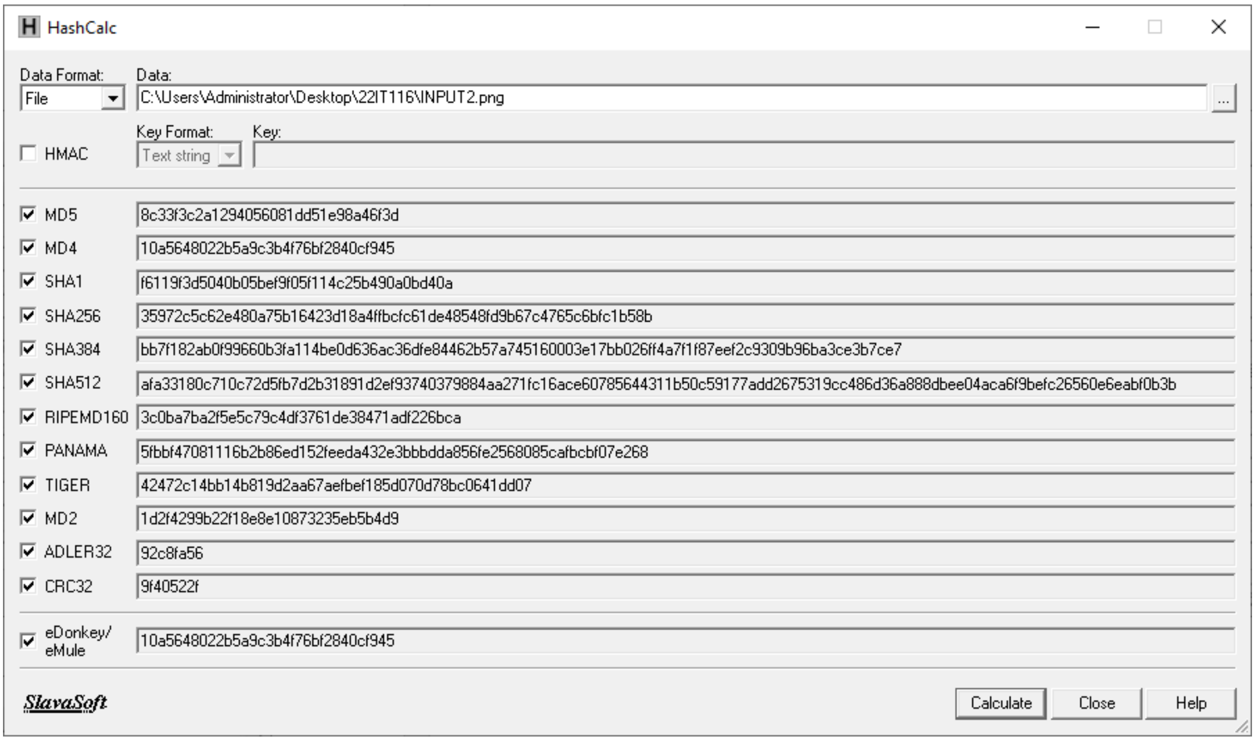
☑ MD2 408dc66eb60ef7f24acc5152b7b2c3b5

☑ ADLER32 9e59e7f2

☑ CRC32 19d86e4b

☑ eDonkey/ eMule b9188421417343f03c6062ebeefd2996

Figure 6:Hash values for INPUT1.png photo



Data Format: File

Data: C:\Users\Administrator\Desktop\22IT116\INPUT2.png

□ HMAC Key Format: Text string Key:

☑ MD5 8c33f3c2a1294056081dd51e98a46f3d

☑ MD4 10a5648022b5a9c3b4f76bf2840cf945

☑ SHA1 f6119f3d5040b05bef9f05f114c25b490a0bd40a

☑ SHA256 35972c5c62e480a75b16423d18a4ffbcfc61de48548fd9b67c4765c6bfc1b58b

☑ SHA384 bb7f182ab0f99660b3fa114be0d636ac36dfe84462b57a745160003e17bb026ff4a7f1f87eef2c9309b96ba3ce3b7ce7

☑ SHA512 afa33180c710c72d5fb7d2b31891d2ef93740379884aa271fc16ace60785644311b50c59177add2675319cc486d36a888dbee04aca6f9befc26560e6eabf0b3b

☑ RIPEMD160 3c0ba7ba2f5e5c79c4df3761de38471adf226bca

☑ PANAMA 5fbbf47081116b2b86ed152feeda432e3bbbdda856fe2568085cafbcbf07e268

☑ TIGER 42472c14bb14b819d2aa67aefbef185d070d78bc0641dd07

☑ MD2 1d2f4299b22f18e8e10873235eb5b4d9

☑ ADLER32 92c8fa56

☑ CRC32 9f40522f

☑ eDonkey/ eMule 10a5648022b5a9c3b4f76bf2840cf945

Figure 7:Hash values for Input2.png photo

LATEST APPLICATIONS:

- Hashi Corp Vault
- Acronis True Image
- VeraCrypt
- GnuPG (GPG)
- OpenSSL
- AWS S3

LEARNING OUTCOME:

In this practical, I will learn how to verify the integrity of sensitive financial documents transferred between departments using HashCalc and MD5 Calculator. I will generate hash values before and after the transfer, comparing them to ensure the files remain unchanged. Through this process, I will understand how hashing algorithms help detect data tampering or corruption, practice secure file transfer methods, and troubleshoot integrity issues, ensuring critical files are securely transmitted and stored.

REFERENCES:

1. HashCalc: https://download.cnet.com/hashcalc/3000-2250_4-10130770.html
2. MD5: https://www.techtarget.com/searchsecurity/definition/MD5
3. ChatGPT: https://chatgpt.com/