Practice Questions

1. IaaS Cloud Computing Service offers:
A. Remote Data Center Deployment
B. Platform-as-a-Service
C. Software Hosting
D. Migration of OSes to Hybrid Model

2. Which of the following is an example of SaaS?
A. Cisco WebEx
B. Cisco Metapod
C. Amazon EC2
D. Microsoft Azure

3. Cloud deployment model accessed by multiple parties having shared resources is a:
A. Private Cloud
B. Public Cloud
C. Hybrid Cloud
D. Community Cloud

4. A person or organization that maintains a business relationship with, and uses service from, Cloud Providers is known as:
A. Cloud Auditor
B. Cloud Broker
C. Cloud Carrier
D. Cloud Consumer

5. A person who negotiates the relationship between Cloud Provider & Consumer is called:
A. Cloud Auditor
B. Cloud Broker
C. Cloud Carrier
D. Cloud Supplier

Chapter 20: Cryptography

Technology Brief

As we studied earlier, confidentiality, integrity, and availability are the three basic components around which we should build and maintain our security model. We must know the different methods by which we can implement each one of these features. For example, using encryption, we can make sure that only the sender and receiver can read clear text data. Anybody between the two nodes needs to know the key to decrypt the data. Similarly, hashing is used to ensure the integrity of data. The following section explains the concepts and various methods by which we can implement encryption and hashing in our network. Several terminologies need to be explained before moving to the main topic of this section.

Cryptography Concepts

Cryptography

Cryptography is a technique of encrypting clear text data into scrambled code. The encrypted data is then sent over public or private network toward its destination to ensure confidentiality. This encrypted data, known as "Ciphertext", is decrypted at the destination for processing. Strong encryption keys are used to avoid key cracking. The objective of cryptography is not purely about confidentiality; it also concerns integrity, authentication, and non-repudiation.

Types of Cryptography

Symmetric Cryptography

Symmetric Key Cryptography is the oldest and most widely used cryptography technique in the domain of cryptography. Symmetric ciphers use the same secret key for the encryption and decryption of data. Most widely used symmetric ciphers are AES and DES.

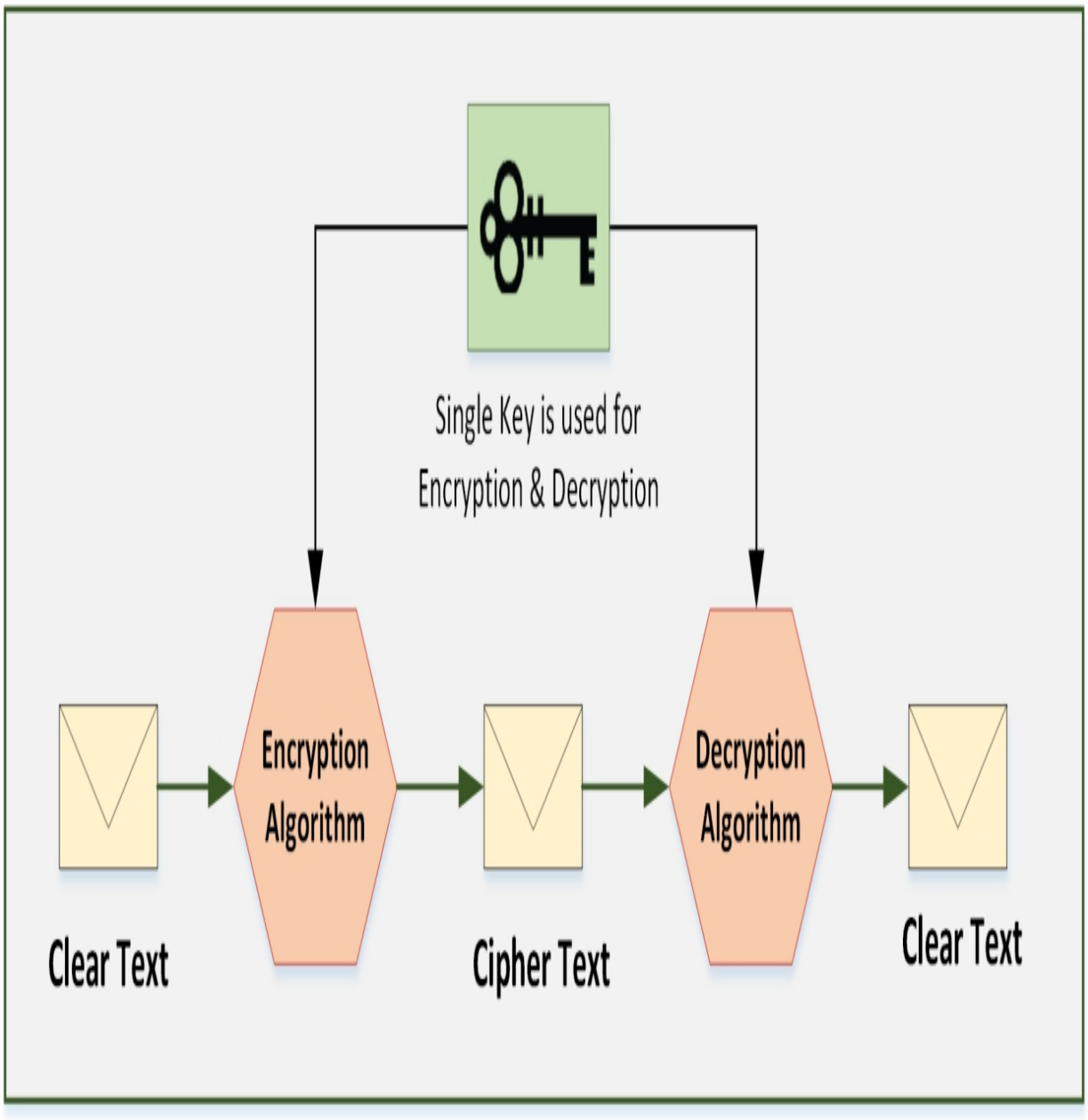


Figure 20-01 Symmetric Cryptography

Asymmetric Cryptography/Public Key Cryptography

Unlike Symmetric Ciphers, in Asymetric Cryptography two keys are used. Everyone publically knows one key, while the other key is kept a secret and is used to encrypt data by the sender; hence, it is also called Public Key Cryptography. Each sender uses its secret key (also known as a Private Key) for encrypting its data before sending. The receiver uses the respective sender's public key to decrypt the data.

RSA, DSA, and the Diffie-Hellman Algorithm are popular examples of asymmetric ciphers. Asymmetric key cryptography delivers confidentiality, integrity, authenticity, and non-repudiation by using the public and private key concept. The private key is only known by the owner itself whereas the public key is issued by Public Key Infrastructure (PKI), where a trusted Certificate Authority (CA) certifies the ownership of key pairs.

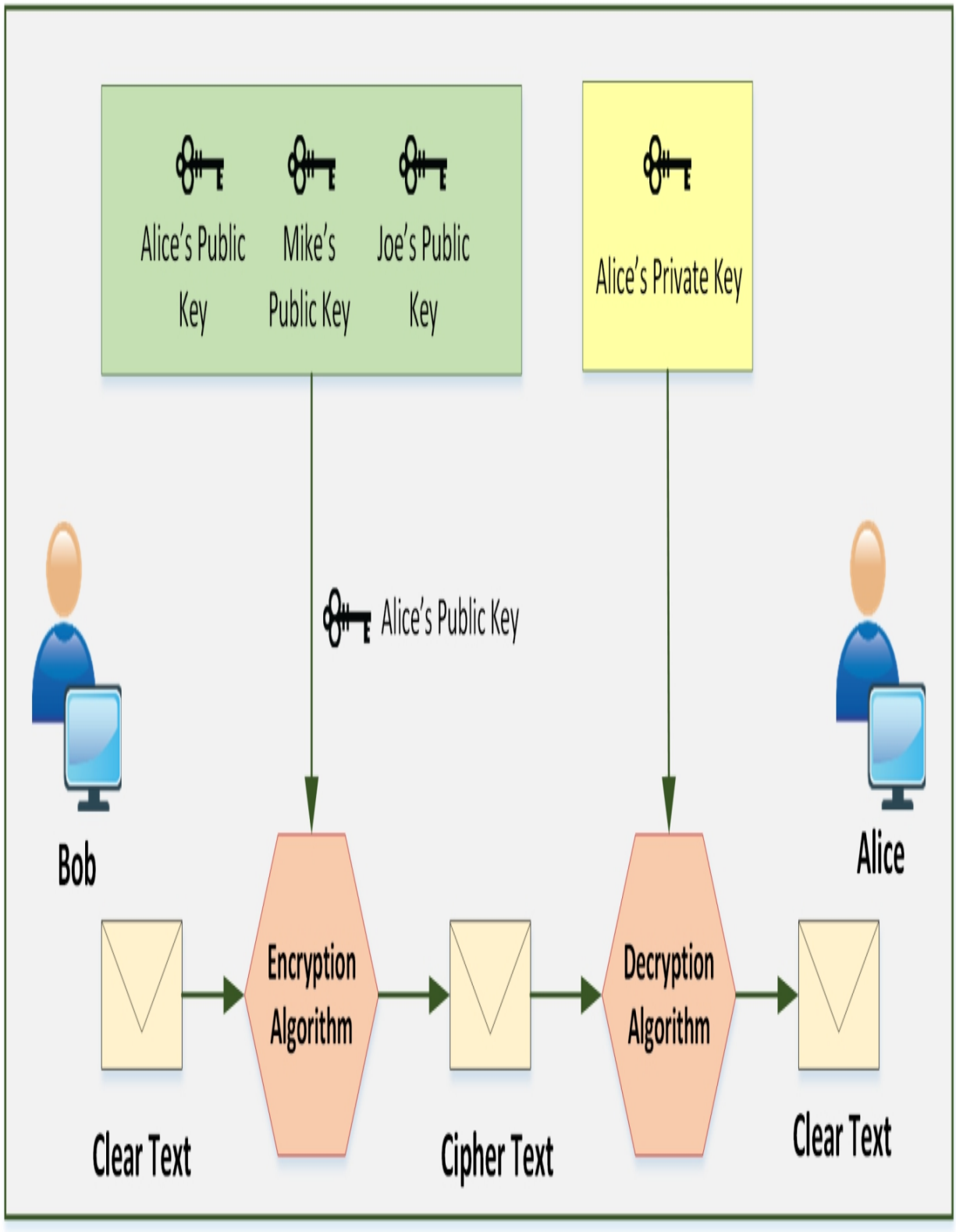


Figure 20-02: Asymmetric Cryptography

Government Access to Keys (GAK)

Government Access to Keys (GAK) refers to agreements between government and software companies. All or necessary keys are delivered to a governmental organization, which keeps them securely and only uses them when a court issues a warrant to do so.

Encryption Algorithms

Ciphers

A cipher is a set of rules by which we implement encryption. Thousands of cipher algorithms are available on the internet. Some of them are proprietary while others are open source. The following are the common methods by which ciphers replace original data with encrypted data.

Substitution

In this method, every single character of data is substituted with another character. A very simple example in this regard would be to replace a character by shifting it three characters along. Here, "D" would replace "A" and so on. To make it more complex, we can select certain letters to be replaced in the whole text. In our example, the value of the key is three, and both nodes should know that value, otherwise they would not be able to decrypt the data.

Polyalphabetic

This method makes substitution even more difficult to break by using multiple character substitution.

Keys

In the above example of substitution, we used a key of "three". Keys play the main role in every cipher algorithm. Without knowing the key, data cannot be decrypted. *Stream Cipher*

A Stream Cipher is a type of symmetric key cipher that encrypts plain text one by one. There are various types of stream ciphers, for example, synchronous, asynchronous, etc. RC4 is the most common type of stream cipher design. The transformation of encrypted output varies during the encryption cycle.

Block Cipher

This is a type of symmetric key cipher that encrypts plain text by processing the fixed length blocks. The transformation of encrypted data does not vary in a block cipher. It encrypts the block of data using the same key on each block. DES and AES are common types of block cipher design.

Data Encryption Standard (DES)

Data Encryption Standard (DES) algorithm is a symmetric key algorithm used for encryption that is now considered insecure. However, successors such as Triple DES and G-DES have replaced DES encryption. DES uses 56-bit key size that is too small to protect data.

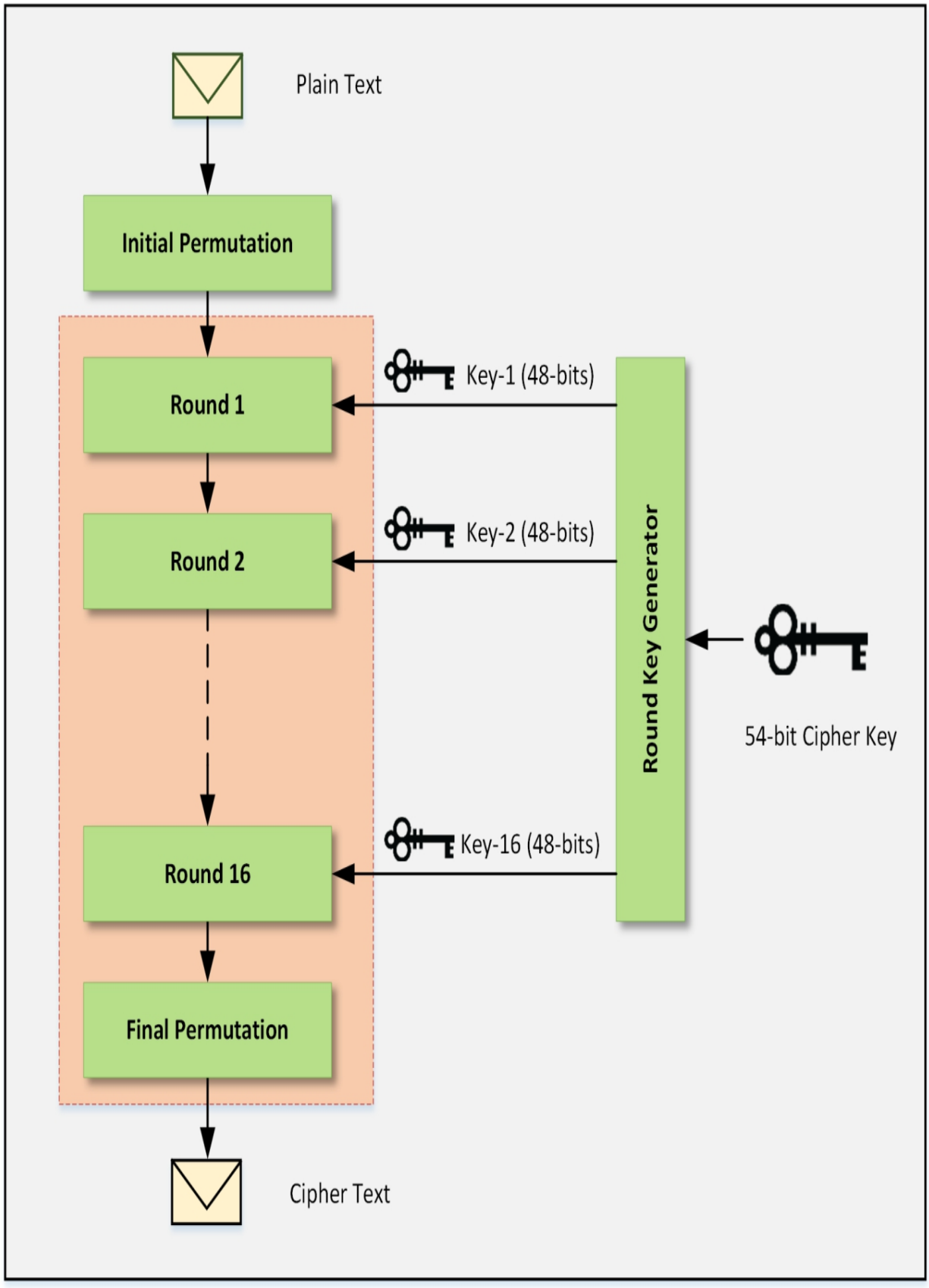


Figure 20_03: DES Algorithm

Figure 20-03: DES Algorithm

The DES algorithm consists of 16 rounds, which process the data with 16 intermediary round keys of 48-bits. These intermediary keys are generated from 56-bit cipher keys by a Round Key Generator. Similarly, a DES reverse cipher computes the data in clear text format from cipher text using the same cipher key.

The following are the major parameters of DES:

DES Algorithm Parameters Values Block Size 64 bits Key Size 56 bits 16Number of Rounds

16 Intermediary Keys 48 bits *Table 20-01: DES Algorithm Parameters*

Advanced Encryption Standard (AES)

When DES becomes insecure and performing DES encryption three times (3-DES or Triple-DES) takes high computation and time, another encryption algorithm is needed that is more secure and effective. Rijndael issued a new algorithm in 2000-200 1 known as Advanced Encryption Algorithm (AES). AES is also a private key symmetric algorithm but it is stronger and faster than Triple-DES. AES can encrypt 128-bit data with 128/ 192/256 bit keys.
The following are the major parameters of AES.

AES Algorithms Parameters AES 128 AES 192 AES256 Block Size 4 / 16 / 128 bits 6 / 24 / 192 bits 8 / 32 / 256 Key Size 4 / 16 / 128 bits 4 / 16 / 128 bits 4 / 16 / 128 bits Number of Rounds 10 12 14 Round Key Size 4 / 16 / 128 bits 4 / 16 / 128 bits 4 / 16 / 128 bits Expanded Key Size 44 / 176 bits 52 / 208 60 / 240

Table 20-02: AES Algorithm Parameters

To understand the AES algorithm, consider an AES 128-bit scenario. In 128-bit AES, there will be 10 rounds. The initial9 rounds perform the same step, i.e., substitute bytes, shift rows, mix columns, and add round keys. The last round is slightly different with only substitute bytes,

shifting rows and adding round keys. The following figure shows the AES algorithm architecture.

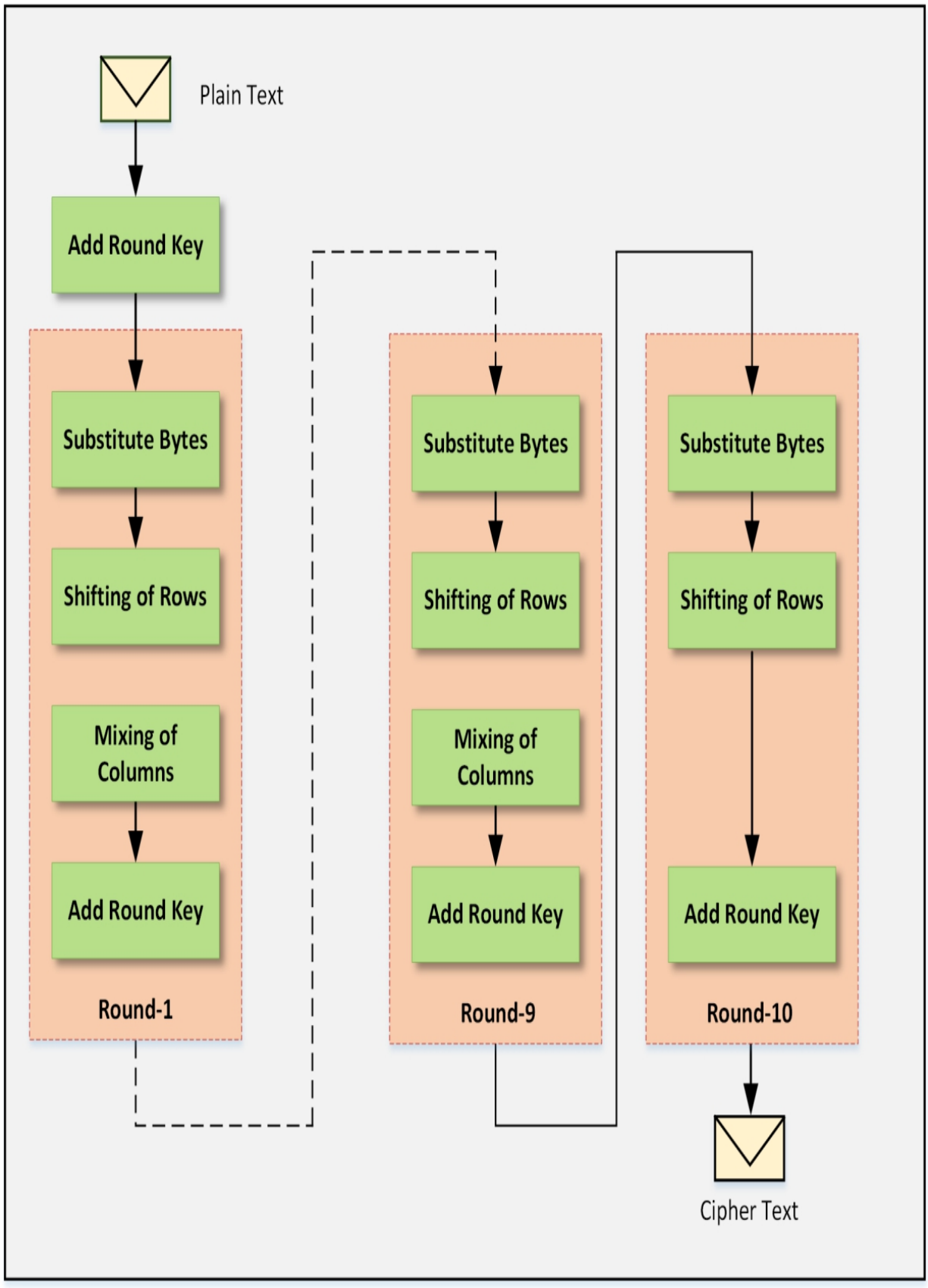


Figure 20. 04: AES Algorithm

Figure 20-04: AES Algorithm

RC4, RC5, RC6 Algorithms

RC 4 is an older encryption technique designed in 1987 by Ron Rivest based on stream cipher. RC4 is used in SSL and WEP protocols. RC4 generates a pseudorandom stream used for encrypting plain text by bit-wise exclusive-Or (similar to the Vernam cipher except for the generated pseudorandom bits). Similarly, the process of decryption is performed as it is a symmetric operation. In the RC4 algorithm, a 24-bit Initialization Vector (IV) generates a 40- or 128-bit key.

RC 5 is a symmetric key block cipher introduced in 1994. RC5 has variable block sizes (32, 64, or 128 bits) with a key size of 0 to 2040 bits and 0 to 255 rounds. It is suggested that RC5 is used with the 64-bit block size, 128-bit key, and 12rounds. RC5 also consists of some modular additions and exclusive OR (XOR)s.
RC6 is also a symmetric key block cipher that is derived from RC5 with a block size of 128 bits with 128-, 192-, 256-, and up to 2040-bit key support. RC6 is very similar to RC5 in structure, using data-dependent rotations, modular addition, and XOR operations. RC6 does use an extra multiplication operation not present in RC5 to make the rotation dependent.

The DSA and Related Signature Schemes

A signature, just as it used in daily life, proves authenticity and proves the actual origin of a document. In computer networking, the Digital Signature Algorithm (DSA) is used to sign a digital document. A Digital Signature can provide three components of network security, i.e. authenticity of a message, integrity of a message, and nonrepudiation. A digital signature cannot provide confidentiality of communication. However, this can be achieved by using encrypted messages and signatures.

A digital signature uses a public key to sign and verify packets. The signing of a document requires a private key whereas verification requires a public key. The sender of a message signs it with his/her private key and sends it to the receiver. The receiver verifies the authenticity of the message by decrypting the packet with the sender's

public key, as the sender's public key only decrypts the message and verifies the sender of that message.

The integrity of a message is preserved by signing the entire message. If any content of the message is changed, it will not get the same signature. In a nutshell, integrity is the process of signing and verifying a message obtained by using Hash Functions.

A Digital Certificate contains various items, listed below:

• Subject: The certificate holder's name
• Serial Number: A unique number for certificate identification • Public Key: A copy of the certificate holder's public key
• Issuer: A certificate issuing authority's digital signature to verify that the

certificate is real
• Signature Algorithm: An algorithm used by the Certificate Authority (CA) to
sign a certificate digitally
• Validity: Validity of a certificate, or expiry date and time, of the certificate

A Digital Certificate has X.509 version supported format, which is the standard format.

Note: Certificate validation determines whether the certificate and public key it contains are trustworthy. The verification process is completed by a Certificate Authority.

RSA (Rivest Shamir Adleman)

This algorithm is named after its creators, Ron Rivest, Adi Shamir, and Leonard Adleman. Also known as Public Key Cryptography Standard (PKCS) # 1, the main purpose of its use today is authentication. The key length varies from 5 12 to 2048 with 1024 being preferred. RSA is one of the de-facto encryption standards.

The RSA Signature Scheme

1. Two very large prime numbers "p" and "q" are required.
2. Multiply the above two primes to find n, the modulus for encryption and decryption. In other words, n = p * q.
3. Calculate ϕ = (p 1) * (q 1).
4. Choose a random integer "e", i.e., Encryption Key. Calculate "d" (Decryption Key) so that d x e = 1mod ϕ.
5. Announce "e" and "n" to the public, while keeping “ϕ" and "d" secret.

Lab 20- 1: Example of an RSA Algorithm

Case Study:

Alice creates a pair of keys for herself. She chooses p = 17 and q = 1
1. Calculate the value of following.

Calculate:
n = ?
ϕ = ?
She then chooses e =7
d = ?
Show how Bob can send the message “88 ” to Alice if he knows e and n.

Solution:

As we know:
n = p * q n= 17 * 1 1 n= 187
Let’s find ϕ:
Φ= (p 1) * (q 1) Φ= (17- 1) * (1 1- 1) Φ= (16) * (10)
Φ= 160

Solution:

Let’s calculate the value of d if e = 7. As we know:
d x e = 1mod ϕ.
$d= e^{-1} \bmod \phi$
$d= 7^{-1} \bmod 160$
d= 23

Solution:

Alice’s Private Key will be (d,p,q) = (23, 17, 1 1) Alice’s Public Key will

be (e,n) = (7, 187)

Alice will share her public key with Bob. Bob will then encrypt the packet using Alice's public key and send a message to her.

As we know:

$C=M^e \bmod n$

Here:

"C" is Ciphertext "M" is Message

$C= M^e \bmod n$

$C= (88)^7 \bmod 187$ C= 1 1

Bob will send " 1 1 " to Alice. Alice will decrypt the cipher using her private key to extract the original message.

As we know:

$M= C^d \bmod n$

$M= (1\ 1)^{23} \bmod 187$ M= 88

Message Digest (One-Way Hash) Functions

The Message Digest is a cryptographic hashing technique used to ensure the integrity of a message. Message and message digest can be sent together or separately through a communication channel. A receiver recalculates the hash of the message and compares it with the message digest to ensure no changes have been made. One-Way-Hashing of a message digest means the hashing function must be a one-way operation. The original message must not be able to be recreated. The message digest is a unique fixed-size bit string that is calculated in a way that if a single bit is modified, it changes 50% of the message digest value.

Message Digest Function: MD5

The MD 5 algorithm is from the message digest series. MD5 produces a 128-bit hash value used as a checksum to verify integrity. Hashing is the technique for ensuring integrity. The hash value is calculated by

computing specific algorithms to verify the integrity of data to ensure it was not modified. Hash values play an important role in proving integrity not only of documents and images but also in protocols to ensure the integrity of a transporting payload.

Secure Hashing Algorithm (SHA)

As Message Digest 5 (MD5) is a cryptographic hashing algorithm. Another more popular, secure, and widely used hashing algorithm is the Secure Hashing Algorithm (SHA). SHA 1 is a secure hashing algorithm producing a 160-bit hashing value compared to MD5, which produces a 128-bit value. However, SHA2 is now an even more secure, robust, and safer hashing algorithm.

Syntax: The password is 12345
SHA 1: 567c552b6b559eb6373ce55a43326ba3db92dcbf

Secure Hash Algorithm 2 (SHA2)

SHA 2 has the option of varying a digest between 224 bits to 5 12 bits. SHA2 is a group of different hashes including SHA256, SHA384, and SHA 5 12. The stronger cryptographic algorithm will minimize the chances of compromise.

SHA256

Syntax: The password is 12345
SHA256: 5da923a6598f034d9 1f375f73 143b2b2f58be8a 1c94 17886d5966968b7f79674

SHA384
Syntax: The password is 12345

SHA384: 929f4c 12885cb73d05b90dc825f70c2de64ea72 1e 15587deb3430999 1f6d57 1 14500465243ba08a554f8fe7c8dbbca04

SHA5 12 Syntax: The password is 12345
SHA5 12:

1d967a52ceb7383 16e85d94439dbb 1 12dbcb8b7277885b76c849a80905ab370dc 1 1d2b84dcc88d6 1393 1

17de483a950ee253fba0d26b5b 168744b94af2958 145

Hashed Message Authentication Code (HMAC)

HMAC uses the mechanism of hashing, but adds the further feature of using a secret key in its operation. Both peers only know this secret key. Therefore, in this case, only parties with secret keys can calculate and verify the hash. By using HMAC, if there is an attacker eavesdropping, he/she will not be able to inject or modify the data and recalculate the correct hash because he/she will not know the correct key used by HMAC.

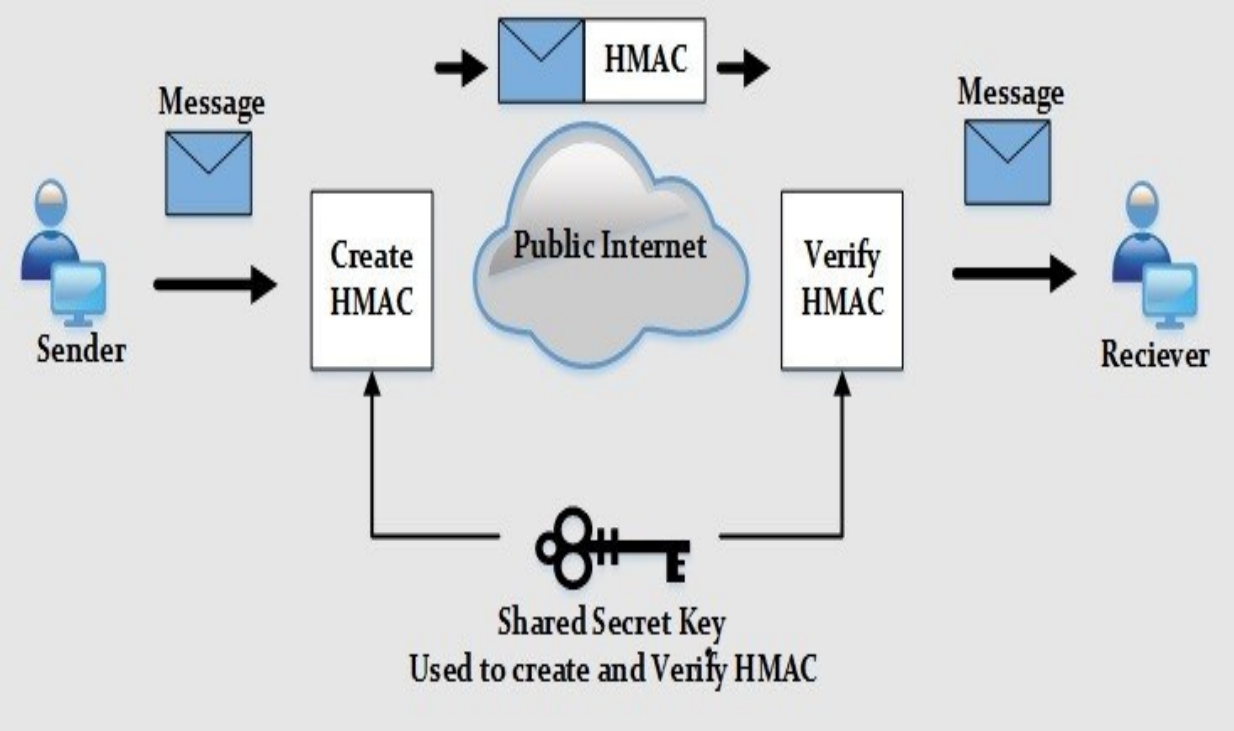


Figure 20-05: HMAC Working Conceptual Diagram

SSH (Secure Shell)

Secure Shell Protocol, commonly known in short as the SSH protocol, is a protocol used for secure remote connections. It is a secure alternative to insecure protocols such as Telnet, rlogin, and FTP. SSH is not only used for remote log in but also with other protocols such as File Transfer Protocol (FTP) and Secure Copy Protocol (SCP). SFTP (SSH File Transfer Protocol) is popularly used for secure file transfer as it runs over SSH. SSH protocol functions over client-server architecture

where the SSH client connects to the SSH server through a secure SSH channel over an insecure network.

Secure Shell (SSH) protocol consists of three major components:

• The Transport Layer Protocol [SSH-TRANS] provides server authentication, confidentiality, and integrity. It may optionally also provide compression. The transport layer will typically run over a TCP/IP connection, but might also be used on top of any other reliable data stream

• The User Authentication Protocol [SSH-USERAUTH] authenticates the clientside user to the server. It runs over the transport layer protocol
• The Connection Protocol [SSH-CONNECT] multiplexes the encrypted tunnel into several logical channels. It runs over the user authentication protocol. Cryptography Tools

MD5 Hash Calculators

Several MD5 calculating tools are available that can directly calculate the hash value of text as well as offers to upload the desired file. Most popular tools are:

1. HashCalc
2. MD5 Calculator
3. HashMyFiles

Lab 20-2: Calculating MD5 using HashCalc Tool

1. Open HashCalc tool.

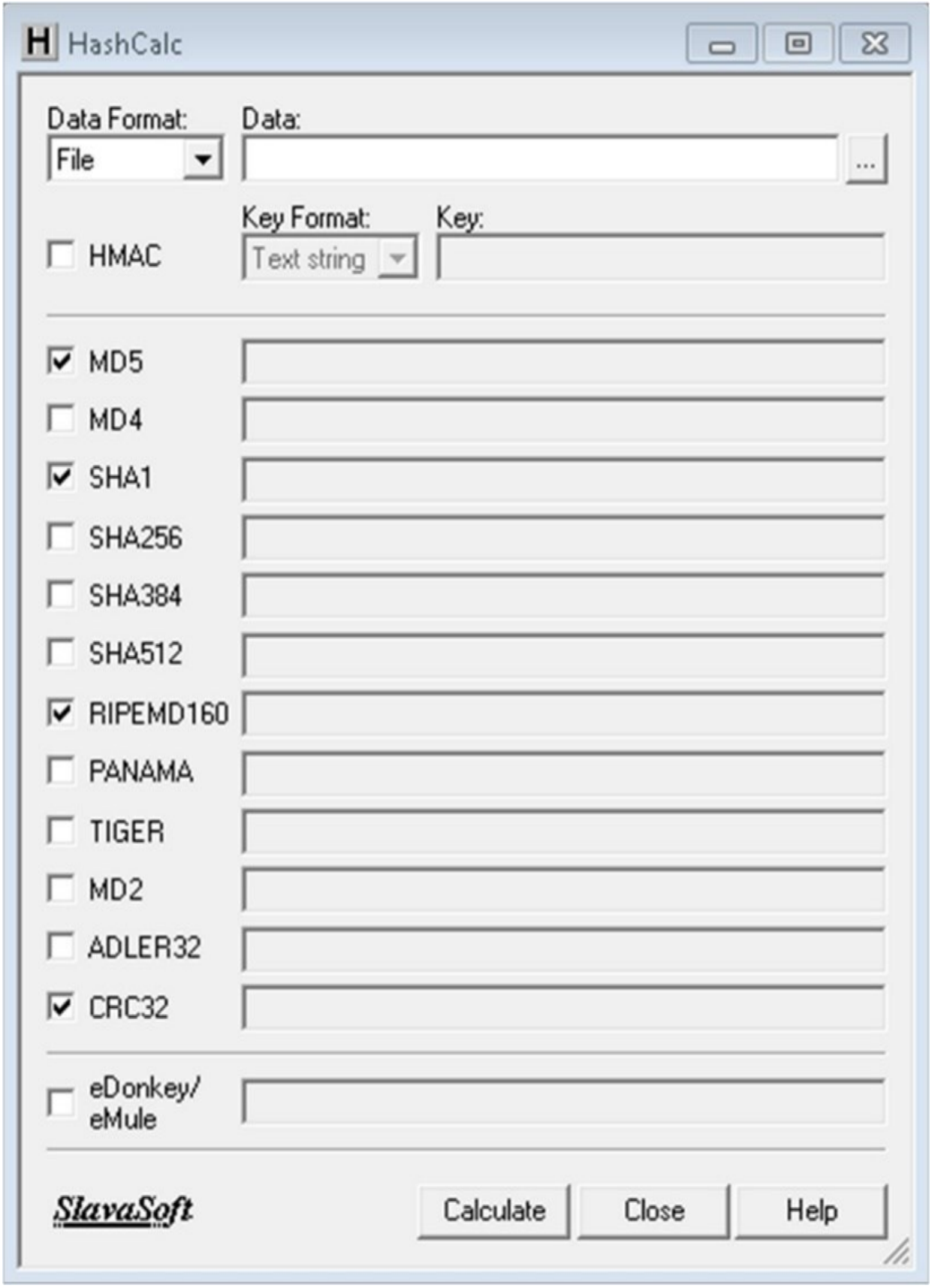

HashCalc
Data Format:
File
Data:
...
HMAC
Key Format:
Text string
Key:
MD5
MD4
SHA1
SHA256
SHA384
SHA512
RIPEMD160
PANAMA
TIGER
MD2
ADLER32
CRC32
eDonkey/
eMule
SlavaSoft
Calculate
Close
Help

Figure 20-06: HashCalc Tool

2. Create a new file with some content in it, as shown below.

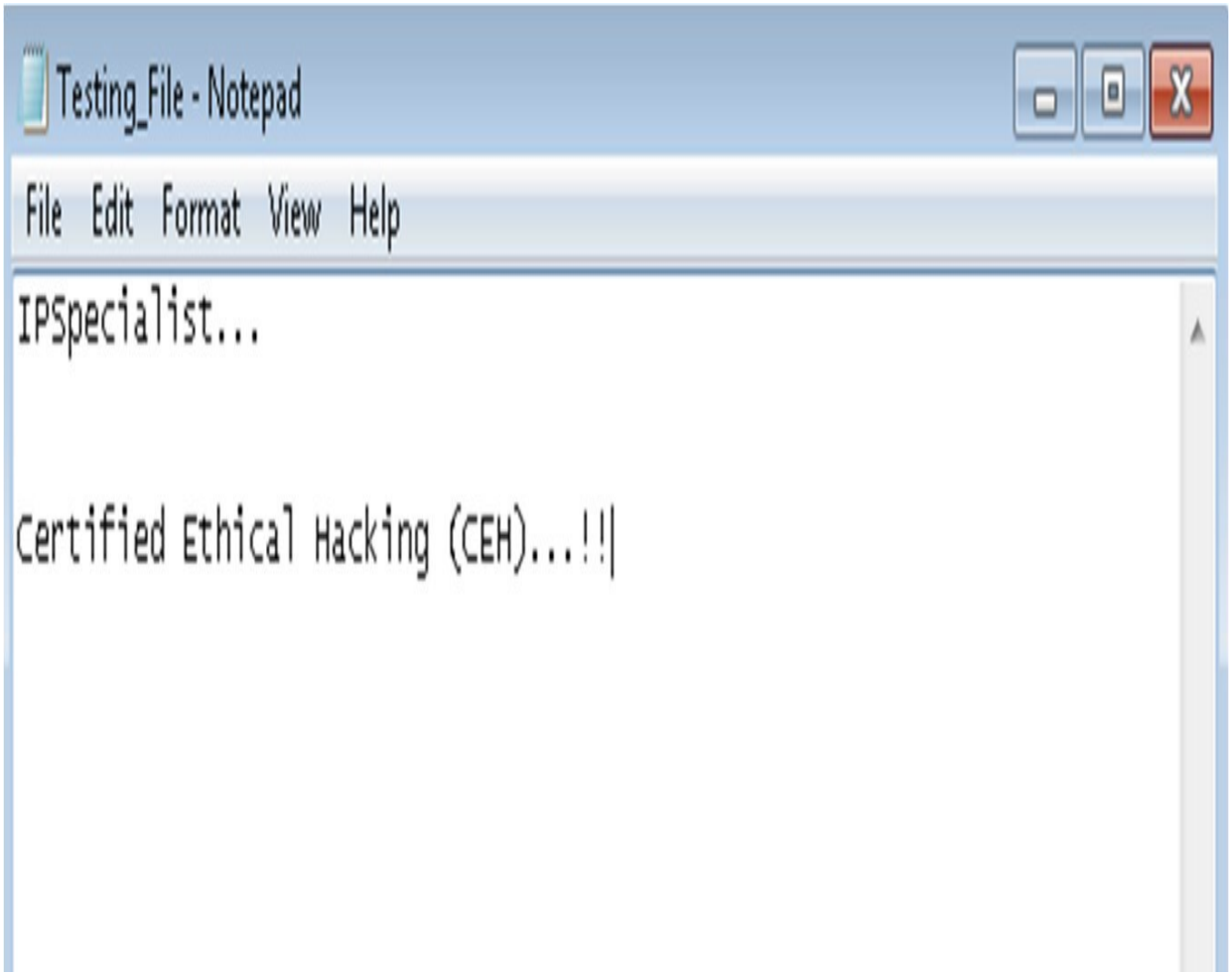


Figure 20-07: Creating a File for MD5 Calculation

3. Select Data Format as "File" and upload your file.

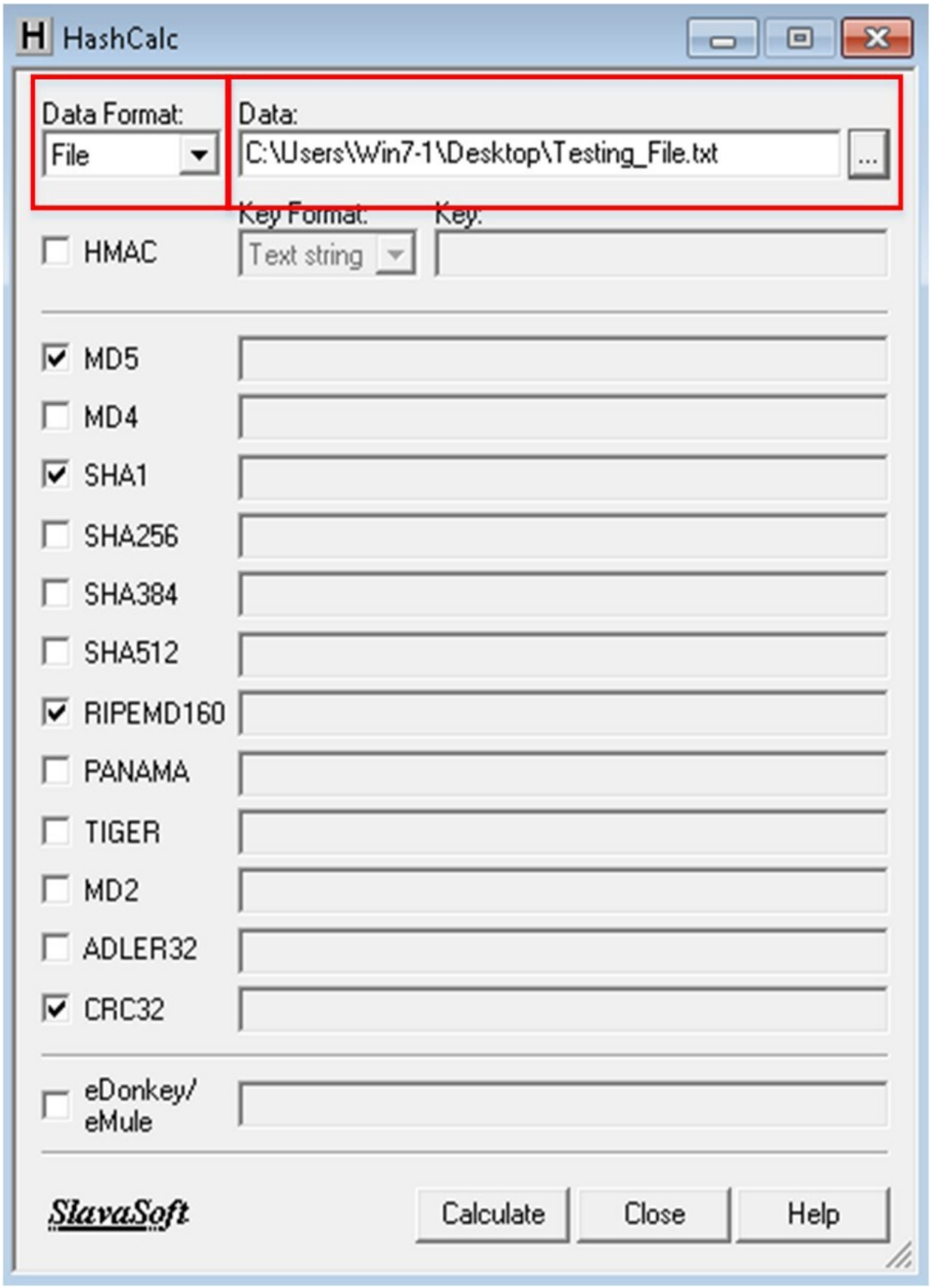

HashCalc
Data Format:
File
Data:
C:\Users\Win7-1\Desktop\Testing_File.txt
...
HMAC
Key Format:
Text string
Key:
MD5
MD4
SHA1
SHA256
SHA384
SHA512
RIPEMD160
PANAMA
TIGER
MD2
ADLER32
CRC32
eDonkey/
eMule
SlavaSoft
Calculate
Close
Help

Figure 20-08: Uploading a File to Calculate Hash

4. Select Hashing Algorithm and click “Calculate”.

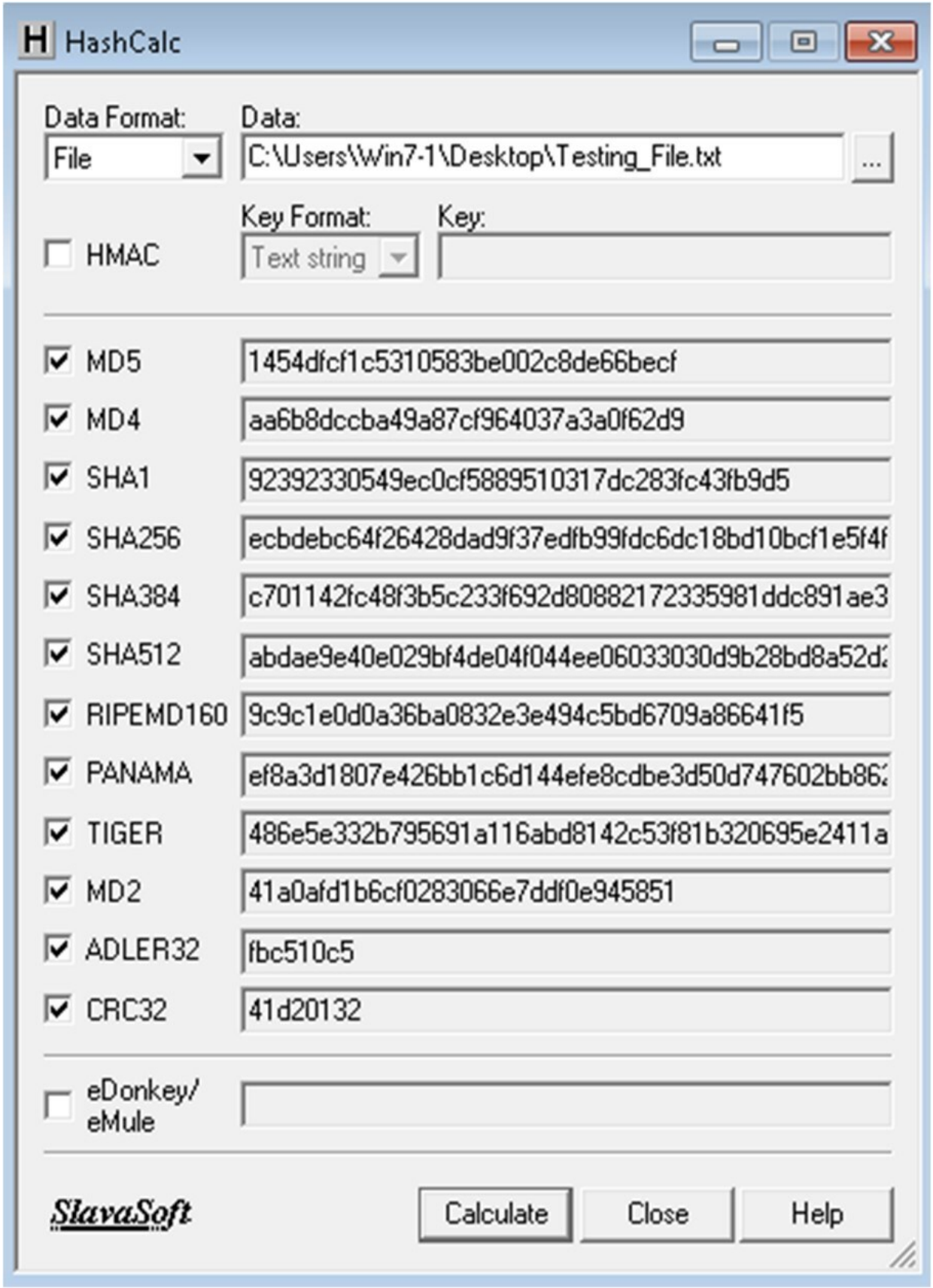

HashCalc
Data Format:
File
Data:
C:\Users\Win7-1\Desktop\Testing_File.txt
...
HMAC
Key Format:
Text string
Key:
MD5
1454dfcf1c5310583be002c8de66becf
MD4
aa6b8dccba49a87cf964037a3a0f62d9
SHA1
92392330549ec0cf5889510317dc283fc43fb9d5
SHA256
ecbdebc64f26428dad9f37edfb99fdc6dc18bd10bcf1e5f4f
SHA384
c701142fc48f3b5c233f692d80882172335981ddc891ae3
SHA512
abdae9e40e029bf4de04f044ee06033030d9b28bd8a52d
RIPEMD160
9c9c1e0d0a36ba0832e3e494c5bd6709a86641f5
PANAMA
ef8a3d1807e426bb1c6d144efe8cdbe3d50d747602bb86
TIGER
486e5e332b795691a116abd8142c53f81b320695e2411a
MD2
41a0afd1b6cf0283066e7ddf0e945851
ADLER32
fbc510c5
CRC32
41d20132
eDonkey/
eMule
SlavaSoft
Calculate
Close
Help

Figure 20-09: Calculating Hash

5. Now, change the data format to “Text String ,” and Type “IPSpecialist… ” into the filed and calculated MD5.

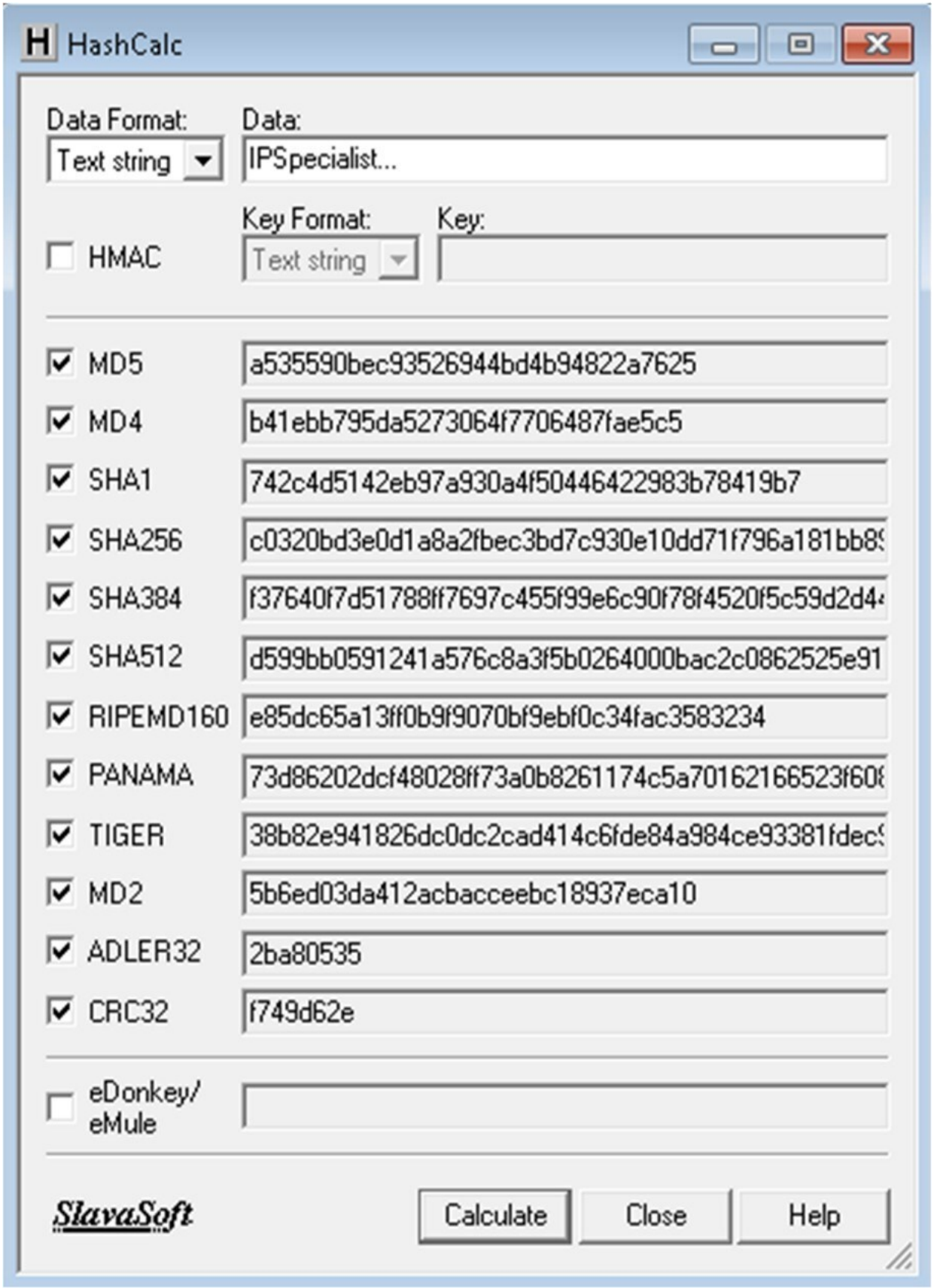

HashCalc
Data Format:
Text string
Data:
IPSpecialist...
HMAC
Key Format:
Text string
Key:
MD5
a535590bec93526944bd4b94822a7625
MD4
b41ebb795da5273064f7706487fae5c5
SHA1
742c4d5142eb97a930a4f50446422983b78419b7
SHA256
c0320bd3e0d1a8a2fbec3bd7c930e10dd71f796a181bb8
SHA384
f37640f7d51788ff7697c455f99e6c90f78f4520f5c59d2d4
SHA512
d599bb0591241a576c8a3f5b0264000bac2c0862525e91
RIPEMD160
e85dc65a13ff0b9f9070bf9ebf0c34fac3583234
PANAMA
73d86202dcf48028ff73a0b8261174c5a70162166523f60
TIGER
38b82e941826dc0dc2cad414c6fde84a984ce93381fdec
MD2
5b6ed03da412acbacceebc18937eca10
ADLER32
2ba80535
CRC32
f749d62e
eDonkey/
eMule
SlavaSoft
Calculate
Close
Help

Figure 20-10: Calculating Hash with Text String

MD5 calculated for the text string "IPSpecialist… " is "a535590bec93526944bd4b94822a7625 ".

6. Now, let's see how the MD5 value has changed from this minor change.

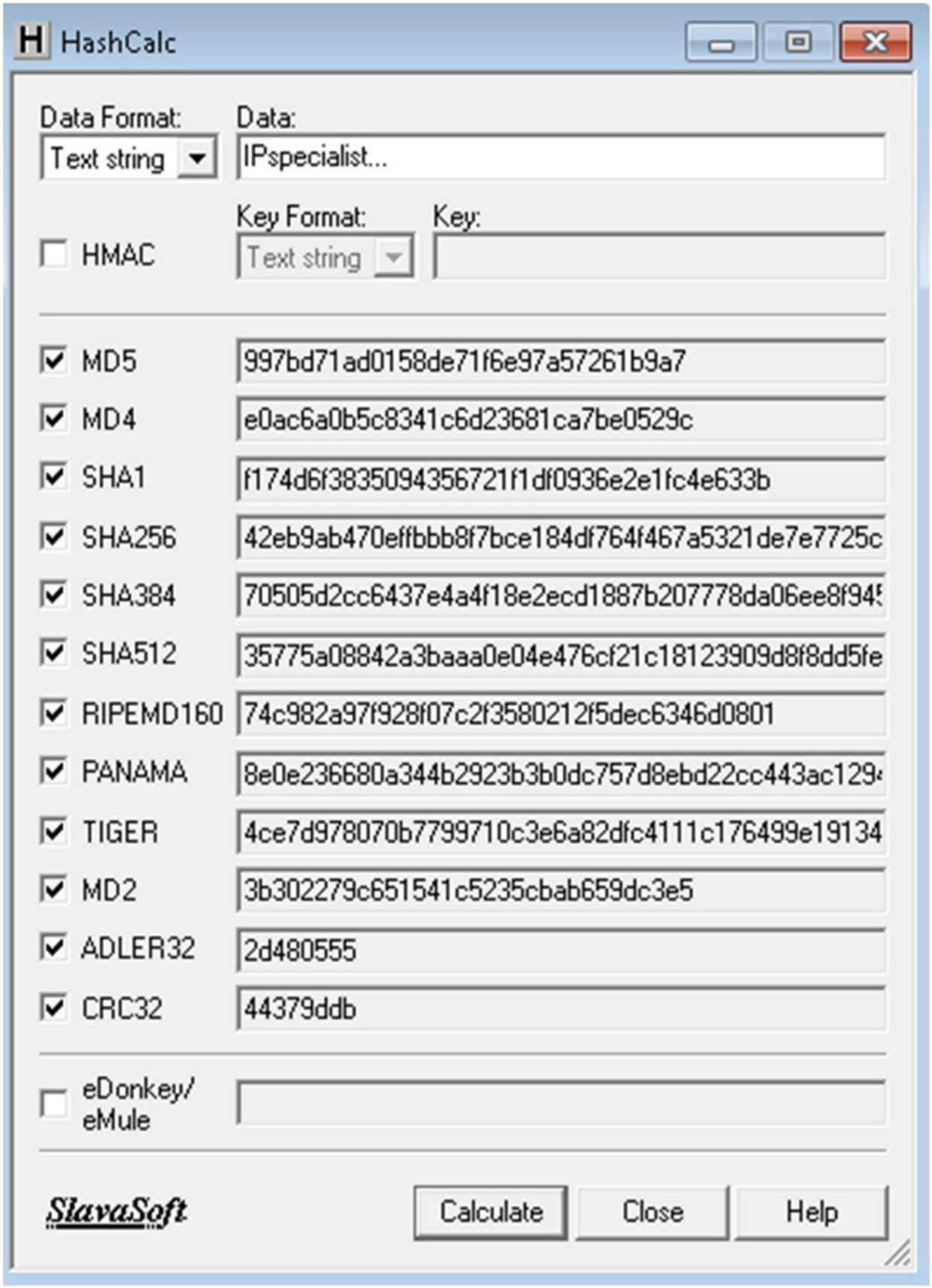

HashCalc
Data Format:
Text string
Data:
IPspecialist...
HMAC
Key Format:
Text string
Key:
MD5
997bd71ad0158de71f6e97a57261b9a7
MD4
e0ac6a0b5c8341c6d23681ca7be0529c
SHA1
f174d6f3835094356721f1df0936e2e1fc4e633b
SHA256
42eb9ab470effbbb8f7bce184df764f467a5321de7e7725c
SHA384
70505d2cc6437e4a4f18e2ecd1887b207778da06ee8f94
SHA512
35775a08842a3baaa0e04e476cf21c18123909d8f8dd5fe
RIPEMD160
74c982a97f928f07c2f3580212f5dec6346d0801
PANAMA
8e0e236680a344b2923b3b0dc757d8ebd22cc443ac129
TIGER
4ce7d978070b7799710c3e6a82dfc4111c176499e19134
MD2
3b302279c651541c5235cbab659dc3e5
ADLER32
2d480555
CRC32
44379ddb
eDonkey/
eMule
SlavaSoft
Calculate
Close
Help

Figure 20-11: Comparing the Hash of Different Text String

Just lowering the case of single alphabet changes entire hashing value. MD5 calculated for the text string "lPspecialist… " is "997bd7 1ad0 158de7 1f6e97a5726 1b9a7".

String MD5

lPSpecialist… a535590bec93526944bd4b94822a7625 lPspecialist… 997bd7 1ad0 158de7 1f6e97a5726 1b9a7

Table 20-03: Comparing MD5 Values

Hash Calculators for Mobile:

Hash calculating tools for mobile phones are:

- MD5 Hash Calculator
- Hash Droid
- Hash Calculator

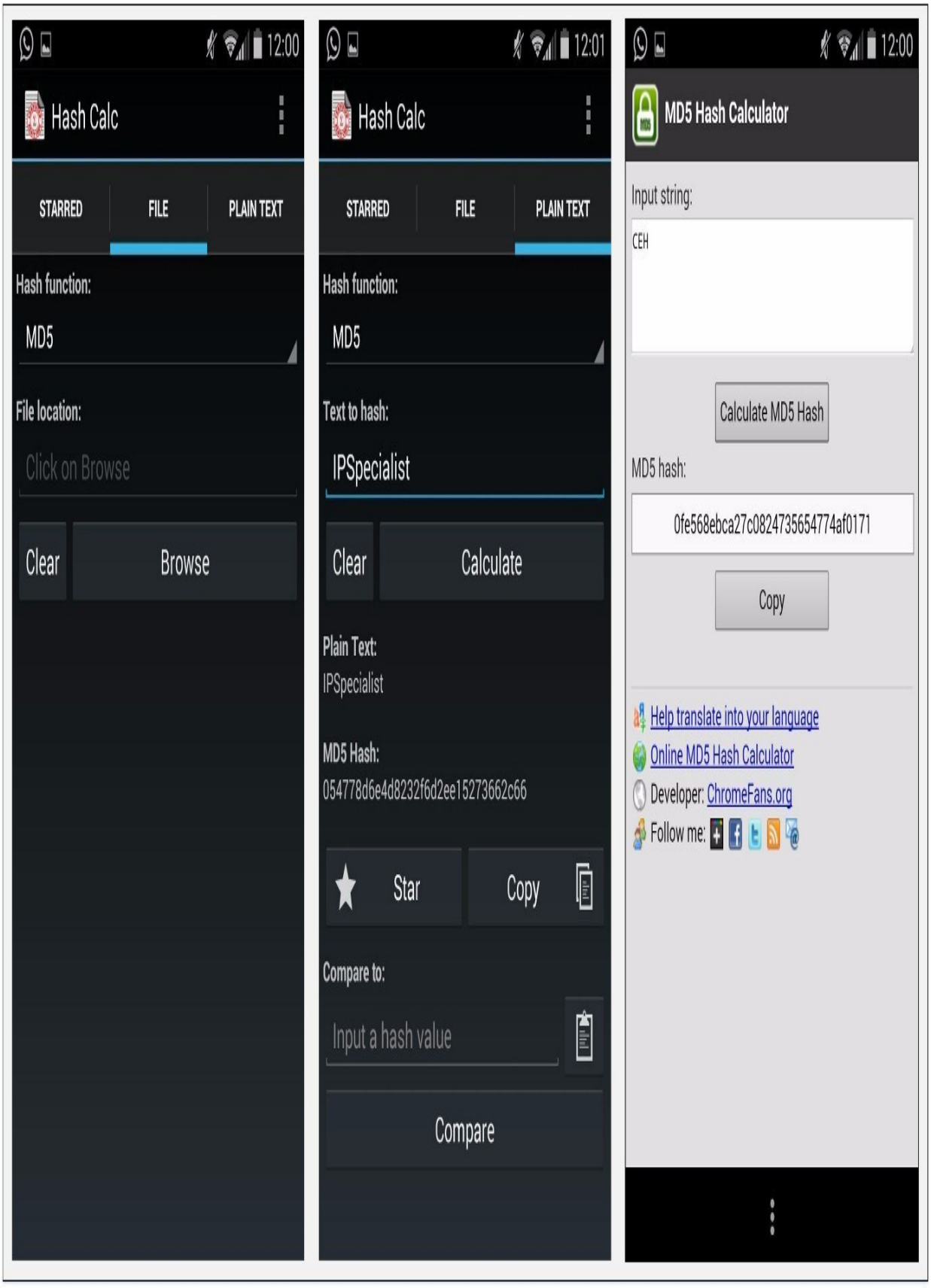


Figure 20 - 12: Hashing Tools for Mobile

Figure 20-12: Hashing Tools for Mobile

Cryptography Tools

There are several tools available for encrypting files such as Advanced Encryption Package and BCTextEncoder. Similarly, some mobile cryptography applications are Secret Space Encryptor, CryptoSymm, and Cipher Sender.

Lab 20-3: Advanced Encryption Package 20 14

Procedure:

1. Download and install Advance Encryption Packages' latest version. In this Lab, we are using Advanced Encryption Package 20 14 and 20 17 to ensure compatibilities on Windows 7 and Windows 10.

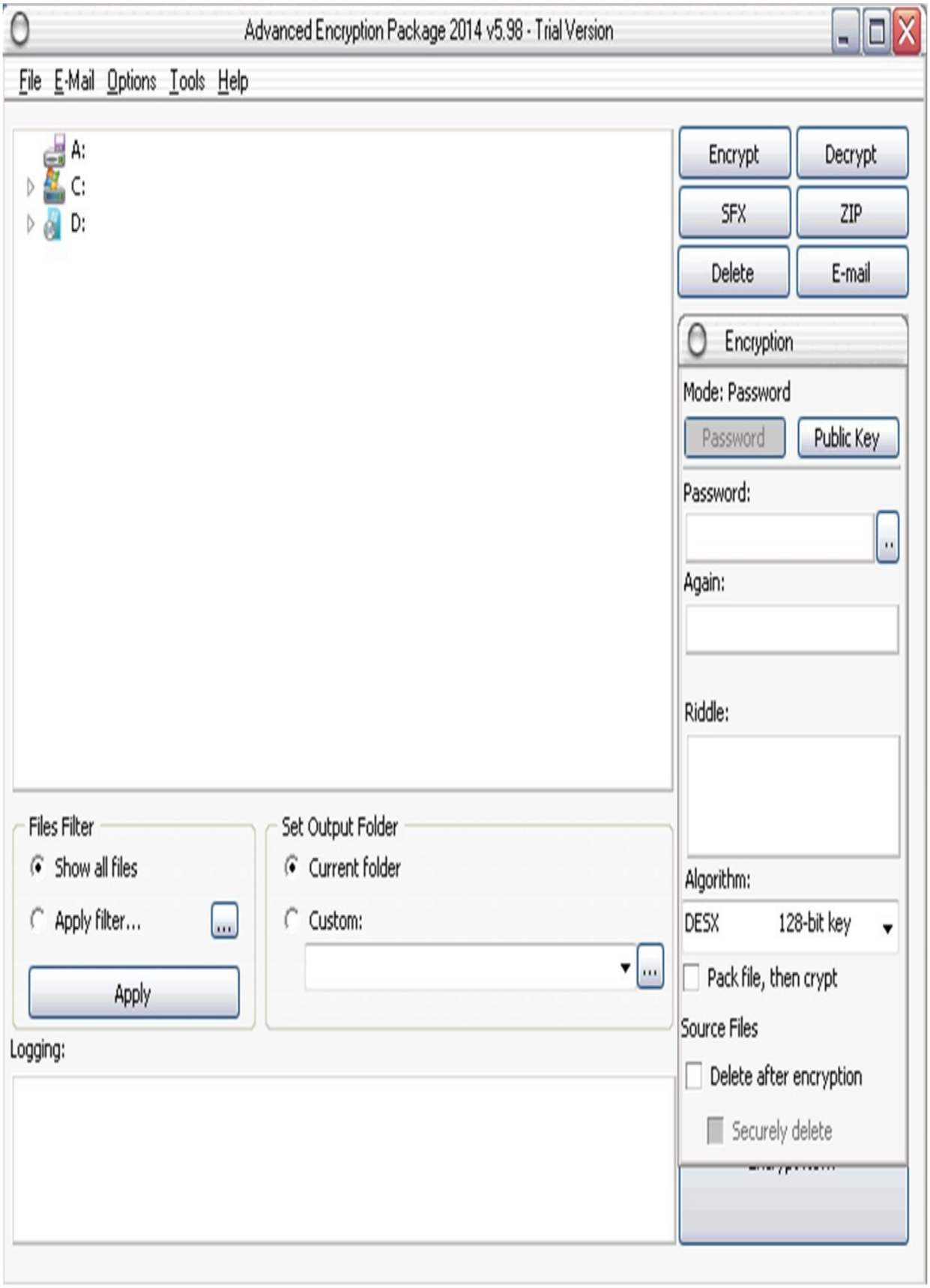


Figure 20-13: Advanced Encryption Package 20.14

Figure 20-13: Advanced Encryption Package 2014

2. Select the file you want to encrypt, set password and select "Algorithm".

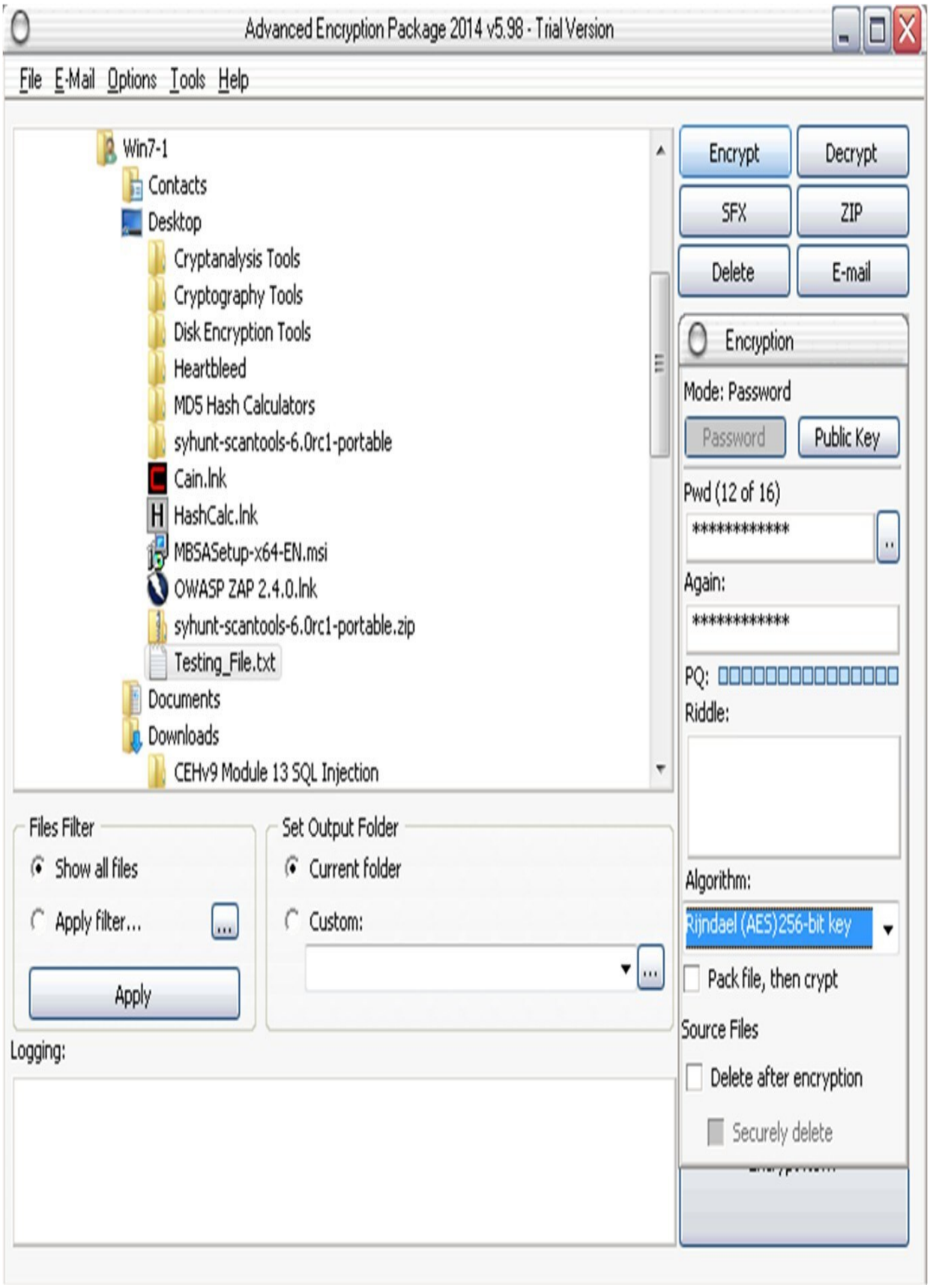

Advanced Encryption Package 2014 v5.98 - Trial Version
File E-Mail Options Tools Help
Win7-1
Contacts
Desktop
Cryptanalysis Tools
Cryptography Tools
Disk Encryption Tools
Heartbleed
MD5 Hash Calculators
syhunt-scantools-6.0rc1-portable
Cain.lnk
HashCalc.lnk
MBSASetup-x64-EN.msi
OWASP ZAP 2.4.0.lnk
syhunt-scantools-6.0rc1-portable.zip
Testing_File.txt
Documents
Downloads
CEHv9 Module 13 SQL Injection
Encrypt
Decrypt
SFX
ZIP
Delete
E-mail
Encryption
Mode: Password
Password
Public Key
Pwd (12 of 16)
Again:
PQ:
Riddle:
Files Filter
Show all files
Apply filter...
Apply
Set Output Folder
Current folder
Custom:
Algorithm:
Rijndael (AES) 256-bit key
Pack file, then crypt
Source Files
Delete after encryption
Securely delete
Logging:

Figure 20-14: Uploading a File to Encrypt

3. Click “Encrypt”.

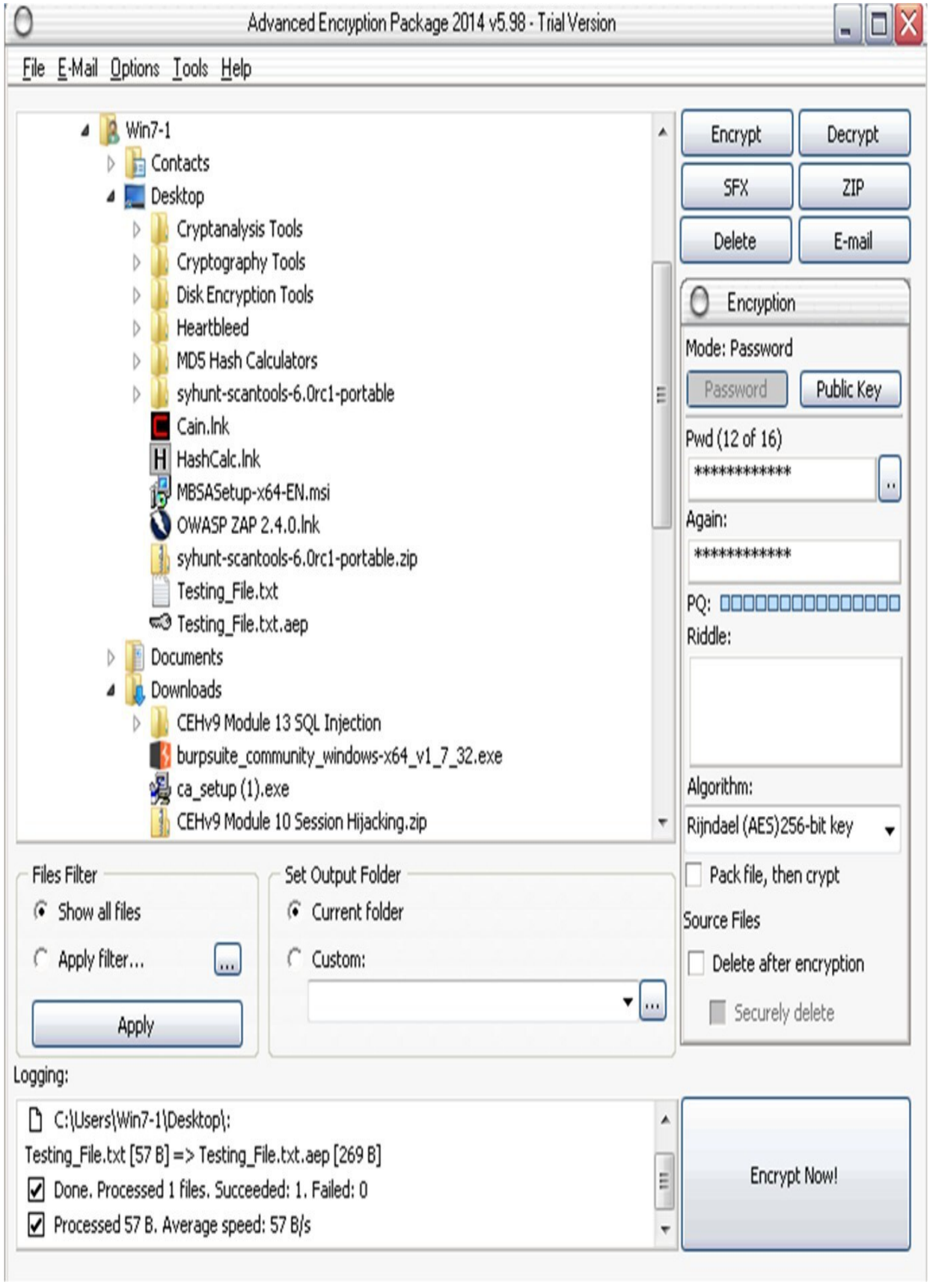

Advanced Encryption Package 2014 v5.98 - Trial Version
File E-Mail Options Tools Help
Win7-1
Contacts
Desktop
Cryptanalysis Tools
Cryptography Tools
Disk Encryption Tools
Heartbleed
MD5 Hash Calculators
syhunt-scantools-6.0rc1-portable
Cain.lnk
HashCalc.lnk
MBSASetup-x64-EN.msi
OWASP ZAP 2.4.0.lnk
syhunt-scantools-6.0rc1-portable.zip
Testing_File.txt
Testing_File.txt.aep
Documents
Downloads
CEHv9 Module 13 SQL Injection
burpsuite_community_windows-x64_v1_7_32.exe
ca_setup (1).exe
CEHv9 Module 10 Session Hijacking.zip
Encrypt
Decrypt
SFX
ZIP
Delete
E-mail
Encryption
Mode: Password
Password
Public Key
Pwd (12 of 16)
****
Again:
****
PQ:
Riddle:
Algorithm:
Rijndael (AES) 256-bit key
Pack file, then crypt
Source Files
Delete after encryption
Securely delete
Files Filter
Show all files
Apply filter...
Apply
Set Output Folder
Current folder
Custom:
Logging:
C:\Users\Win7-1\Desktop\:
Testing_File.txt [57 B] => Testing_File.txt.aep [269 B]
Done. Processed 1 files. Succeeded: 1. Failed: 0
Processed 57 B. Average speed: 57 B/s
Encrypt Now!

Figure 20-15: Encrypting with AES 256-bit

4. Compare both files.

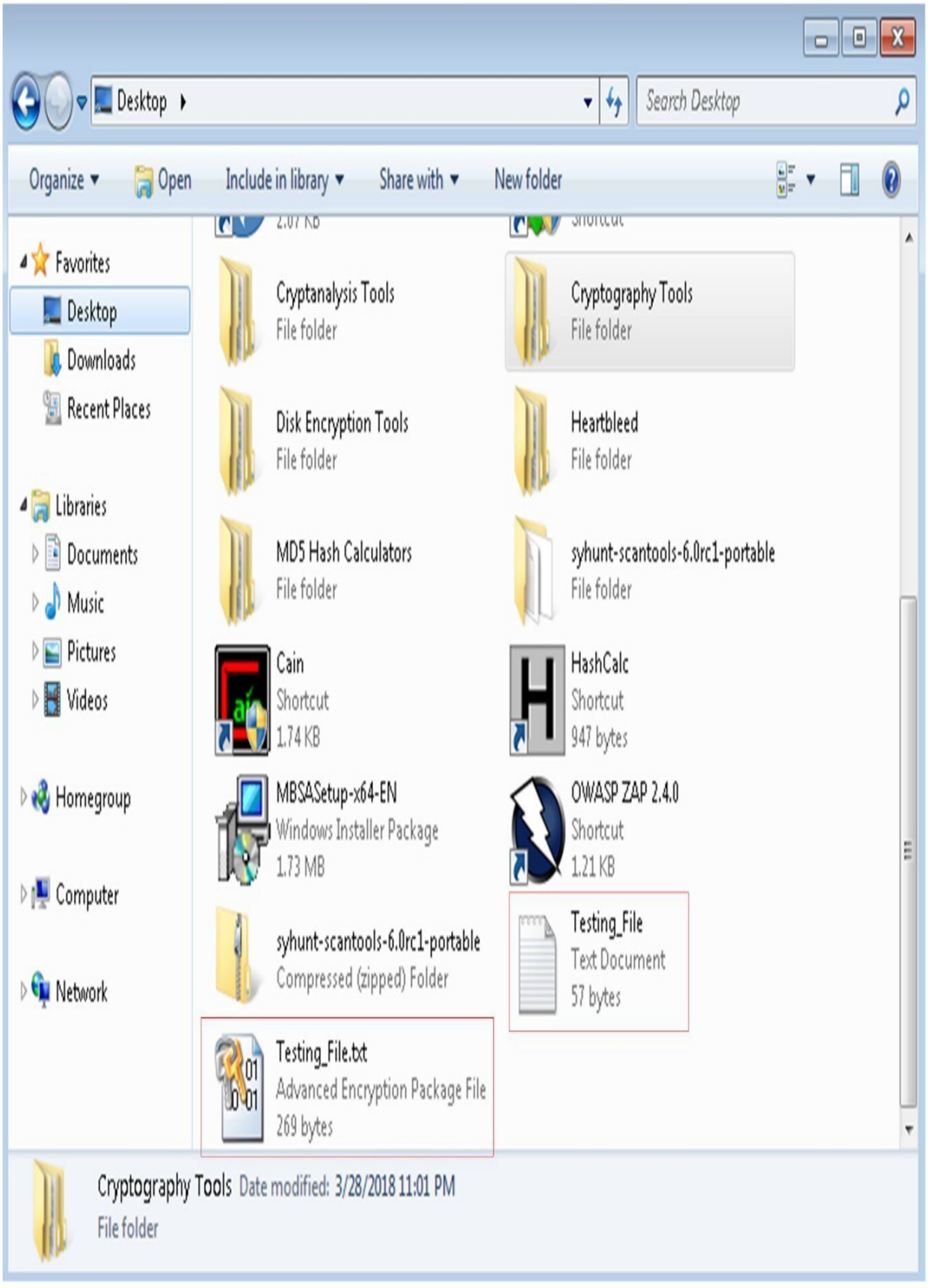

Desktop
Search Desktop
Organize
Open
Include in library
Share with
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Homegroup
Computer
Network
Cryptanalysis Tools
File folder
Cryptography Tools
File folder
Disk Encryption Tools
File folder
Heartbleed
File folder
MD5 Hash Calculators
File folder
syhunt-scantools-6.0rc1-portable
File folder
Cain
Shortcut
1.74 KB
HashCalc
Shortcut
947 bytes
MBSASetup-x64-EN
Windows Installer Package
1.73 MB
OWASP ZAP 2.4.0
Shortcut
1.21 KB
syhunt-scantools-6.0rc1-portable
Compressed (zipped) Folder
Testing_File
Text Document
57 bytes
Testing_File.txt
Advanced Encryption Package File
269 bytes
Cryptography Tools Date modified: 3/28/2018 11:01 PM
File folder

Figure 20-16: Comparing Encrypted and Original Files

5. Now, after forwarding it to another PC, in our case a Windows 10 PC, decrypt it using Advanced Encryption package 20 17.
6. Enter the password.

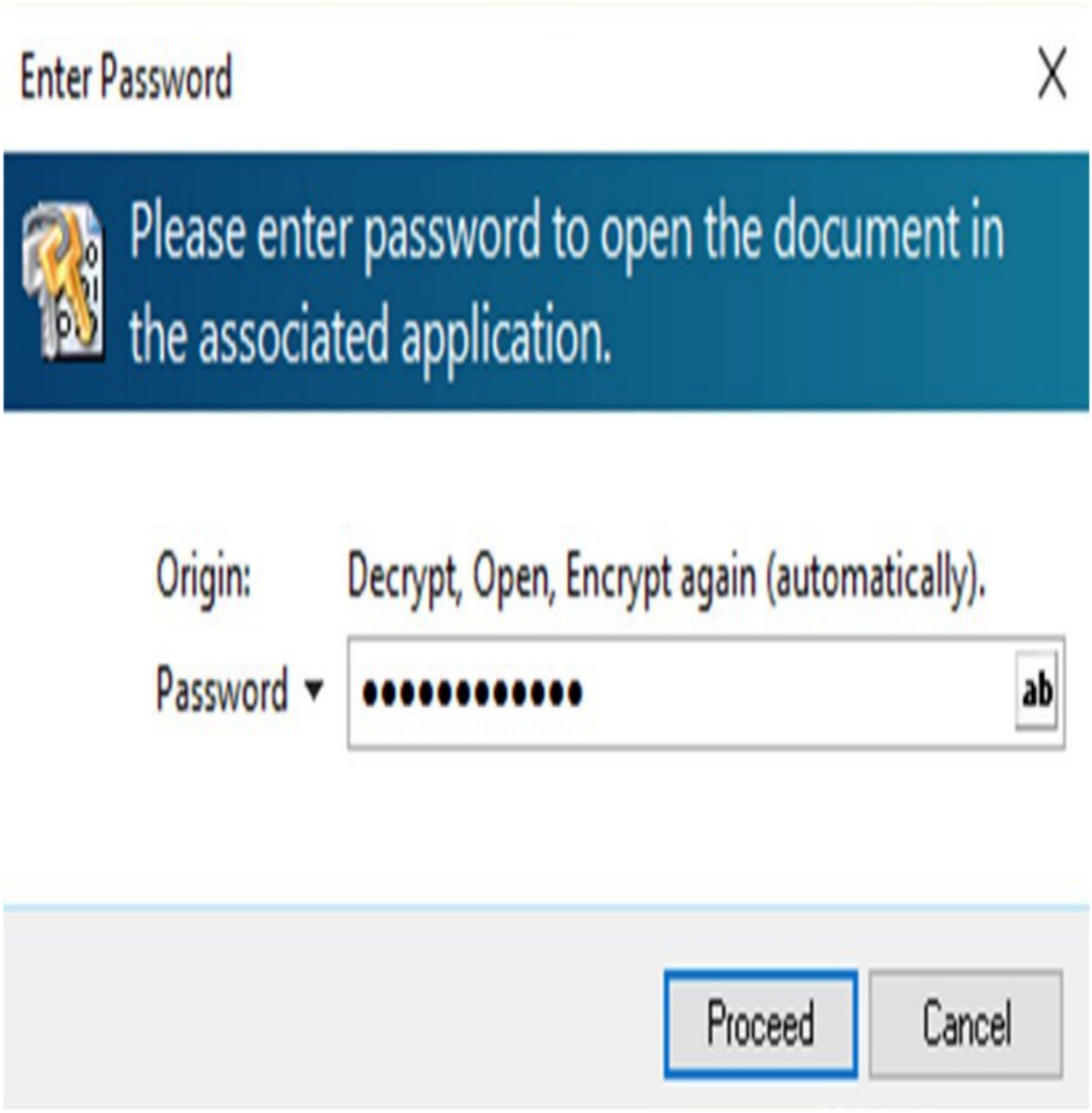


Figure 20-17: Decrypting a File Using Advanced Encryption Package 20 17

7. The file is successfully decrypted.

Figure 20-18: The Decrypted File

Public Key Infrastructure (PKI)
Public Key Infrastructure

PKI is the combination of policies, procedures; hardware, software, and people that are required to create, manage, and revoke digital certificates. A Public Key Infrastructure (PKI) allows users of the internet and other public networks to engage in secure communication, data exchange, and money exchange. This is done through public and private cryptographic key pairs provided by a certificate authority.

Before moving to the original discussion, basic terminologies need to be explained.

Public and Private Key Pair

The Public and Private Key Pair work like a team in the encryption/decryption process. The public key is provided to everyone and the private key is secret. No one has a device's private key. We encrypt data sent to a particular node by using its public key. Similarly, the private key is used to decrypt the data. This is also true in the opposite case. If a node encrypts data with its private key, the public key is used for decryption.

Certificate Authorities (CA)

A Certificate Authority (CA) is a computer or entity that creates and issues digital certificates. A number of things such as IP address, fully qualified domain name, and the public key of a particular device is present in the digital certificate. CA also assigns a serial number to the digital certificate and signs the certificate with its digital signature.

Root Certificate

A Root Certificate provides the public key and other details of CA. An example of a Root certicate is:

Figure 20-19: Example Root Certificate

There are multiple informative sections in the figure above, including serial number, issuer, country and organization names, validity date, and the public key itself. Every OS has its placement procedure regarding

certificates. A certificate container for specific OS can be searched on the internet to get to the certificates stored on the local computer.

Identity Certificate

The purpose of an Identity Certificate is similar to a root certificate except that it provides the public key and identity of a client computer or device. A good example for this is a client router or web server that wishes to make SSL connections with other peers.

Signed Certificate vs. Self-signed Certificate

Self-signed Certificates and Signed Certificates from a Certificate Authority (CA) provide security in the same way. Communication using these types of certificates are protected and encrypted by high-level security. The presence of a Certificate Authority implies that a trusted source has Certificates are purchased whereas certified the communication. Signed Security

Self-signed Certificates can be configured to optimize cost. A third-party Certificate Authority (CA) requires verification of domain ownership and other verification to issue a certificate.

Note: Cross certification enables entities in one Public Key Infrastructure (PKI) to trust entities in another PKI. This mutual trust relationship is typically supported by a crosscertification agreement between Certificate Authorities (CAs) in each PKI.

Email Encryption
Digital Signature

A Digital Signature is a technique to evaluate the authenticity of digital documents as the signature authenticates the authenticity of a document. A digital signature confirms the author of the document, date and time of signing, and authenticates the content of the message.

There are two categories of digital signature:
1. Direct Digital Signature

2. Arbitrated Digital Signature

Direct Digital Signature

Direct Digital Signatures involves only the sender and receiver of a message, assuming that the receiver has the sender's public key. The sender may sign the entire message or hash with the private key and send it toward the destination. The receiver decrypts it using the public key.

Arbitrated Digital Signature

Arbitrated Digital Signatures involves a third party called "Trusted Arbiter". The role of this arbiter is to validate the signed messages, insert the date, and then send it to the recipient. It requires a suitable level of trust and can be implemented with either public or private key.

SSL (Secure Sockets Layer)

In a corporate environment, we can implement the security of corporate traffic over the public cloud by using site-to-site or a remote VPN. In the public cloud, there is no IPsec software running. Normal users also need to do encryption in some cases such as online banking and electronic shopping. In such situations, SSL comes into play. The good thing about Secure Socket Layer (SSL) is that almost every single web browser in use today supports SSL. By using SSL, a web browser makes an HTTPS-based session with the server instead of HTTP. Whenever a browser tries to make a HTTPS-based session with a server, a certificate request is sent to the server in the background. The server, in return, replies with its digital certificate containing its public key. The web browser checks the authenticity of this certificate with a Certificate Authority (CA). Let's assume that the certificate is valid. Now the server and the web browser have a secure session between them.

SSL and TLS for Secure Communication

The terms SSL (Secure Socket Layer) and TLS (Transport Layer Security), often used interchangeably, provide encryption and

authentication of data in motion. These protocols are intended for a scenario where users want secure communication over an unsecured network, for example, the public internet. Most common applications of such protocols are web browsing, Voice over IP (VOIP), and electronic mail.

Consider a scenario where a user wants to send an email to someone or wants to purchase something from an online store where credit card credentials are required. SSL only spills the data after a process known as a 'handshake'. If a hacker bypasses the encryption process, everything from the bank account information to any secret conversation is visible, and malicious users can get hold of it to use for personal gain.

SSL was developed by Netscape in 1994 with the intention of protecting web transactions. The last version of SSL was version 3.0. In 1999, IETF created Transport Layer Security, which is also known as SSL 3. 1 as TLS is, in fact, an adapted version of SSL.

The following are some of the important functionalities SSL/TLS has been designed to do:

- Server authentication to client and vice versa
- Select common cryptographic algorithm
- Generate shared secrets between peers
- Protect normal TCP/UDP connections

Working

The working of SSL and TSL is divided into two phases:

Phase 1 (Session Establishment)

In this phase, common cryptographic protocol and peer authentication take place. There are three sub-phases within the overall phase 1 of SSL/TLS as explained below: • Sub-phase 1: In this phase, hello messages are exchanged to negotiate common parameters of SSL/TLS such as authentication and encryption of algorithms • Sub-phase 2: This phase includes one-way or two-way authentication between client and server end.

• Sub-phase 3: The last phase calculates a session key, and a cipher suite is finally activated. HMAC provides data integrity features by using either SHA 1 or MD5. Similarly, using DES40, DES-CBC, 3DEC-EDE, 3DES-CBC, RC4-40, or RC4- 128 provides confidentiality features

❖ Session Key Creation: Methods for generating session keys are as follows: ■ *RSA Based:* Using the public key of a peer encrypts a shared secret string

■ *A fixed DH Key Exchange:* Fixed Diffie-Hellman-based key exchanged in a certificate creating a session key

■ *An ephemeral DH Key Exchange:* This is considered the best protection option as an actual DH value is signed with the sender's private key, and hence each session has a different set of keys

■ *An anonymous DH Key Exchange without any Certificate or Signature:* Avoiding this option is advised, as it cannot prevent man-inthe-middle attacks.

Phase 2 (Secure Data Transfer)

In this phase, secure data transfer takes place between encapsulating endpoints. Each SSL session has a unique session ID, which is exchanged during the authentication process. The session ID is used to differentiate between an old and new session. The client can request the server resume the session based on this ID (in this event, the sever has a session ID in its cache).
TLS 1.0 is considered a bit more secure than the last version of SSL (SSL v3.0). Even the U.S. government has declared it will not use SSL v3.0 for highly sensitive communications due to the latest vulnerability named POODLE. After the POODLE vulnerability, most web browsers disabled SSL v3.0 for most communication and services. Current browsers (Google Chrome, Firefox, and others) support TLS 1.0 by default and the latest versions of TLS (TLS 1. 1 and TLS 1.2) optionally. TLS 1.0 is considered equivalent to SSL3.0. However, newer versions of TLS are considered far more secure than SSL. Keep

in mind that SSL v3.0 and TLS 1.0 are not compatible with each other as TLS uses Diffie-Hellman and Data Security Standard (DSS) while SSL uses RSA.

Apart from secure web browsing, HTTPS and SSL/TLS can also be used for securing other protocols such as FTP, SMTP, and SNTP.

Note: OPPORTUNISTICTLS STARTTLS is a protocol command issued by an email client. It indicates that a client wants to upgrade an existing insecure connection to a secure one using the SSL/TLS protocol.

Pretty Good Privacy (PGP)

OpenPGP is the most widely used email encryption standard. It is defined by the OpenPGP Working Group of the Internet Engineering Task Force (IETF) as a Proposed Standard in RFC 4880. OpenPGP is derived from PGP software created by Phil Zimmermann. The main purpose of OpenPGP is to ensure end-to-end encryption over email communication; it also provides message encryption and decryption and password manager, data compression, and digital signing.

Disk Encryption

Disk Encryption refers to the encryption of a disk to secure files and directories by converting the data into an encrypted format. Disk encryption encrypts every bit on the disk to prevent unauthorized access to data storage. There are several disk encryption tools available to secure disk volume, for example:

- Symantec Drive Encryption
- GiliSoft Full Disk Encryption

Cryptography Attacks

Cryptography Attacks are intended to recover an encryption key. Once an attacker has the encryption key, he/she can decrypt all messages. Weak encryption algorithms are not resistant enough for cryptographic attacks. The process of finding vulnerabilities in a code, encryption algorithm, or key management scheme is called Cryptanalysis. It may be

used to strengthen a cryptographic algorithm or to decrypt the encryption.

Known Plaintext Attack

A Known Plaintext Attack is a cryptographic attack type where a cryptanalyst has access to plaintext and the corresponding ciphertext and seeks to discover a correlation between them.

Ciphertext-only Attack

A Ciphertext-only Attack is a cryptographic attack type where a cryptanalyst has access to a ciphertext but does not have access to the corresponding plaintext. The attacker attempts to extract the plain text or key by recovering as many plain text messages as possible to guess the key. Once the attacker has the encryption key, he/she can decrypt all messages.

Chosen Plaintext Attack

A Chosen Plaintext Attack is a cryptographic attack type where a cryptanalyst can encrypt a plaintext of his choosing and observe the resulting ciphertext. It is the most common attack against asymmetric cryptography. To attempt a chosen plaintext attack, the attacker has information about the encryption algorithm or may have access to the workstation encrypting the messages. The attacker sends chosen plaintexts through the encryption algorithm to extract ciphertexts and then uses the encryption key. A chosen plaintext attack is vulnerable in a scenario where public key cryptography is in use and the public key is used to encrypt the message. In the worst cases, an attacker can expose sensitive information.

Chosen Ciphertext Attack

A Chosen Ciphertext Attack is a cryptographic attack type where a cryptanalyst chooses a ciphertext and attempts to find the corresponding plaintext.

Adaptive Chosen Ciphertext Attack

A Adaptive Chosen Ciphertext Attack is an interactive type of chosen plaintext attack where an attacker sends some ciphertexts to be decrypted and observes the results of decryption. An adaptive chosen ciphertext attack gradually reveals the information about the encryption.

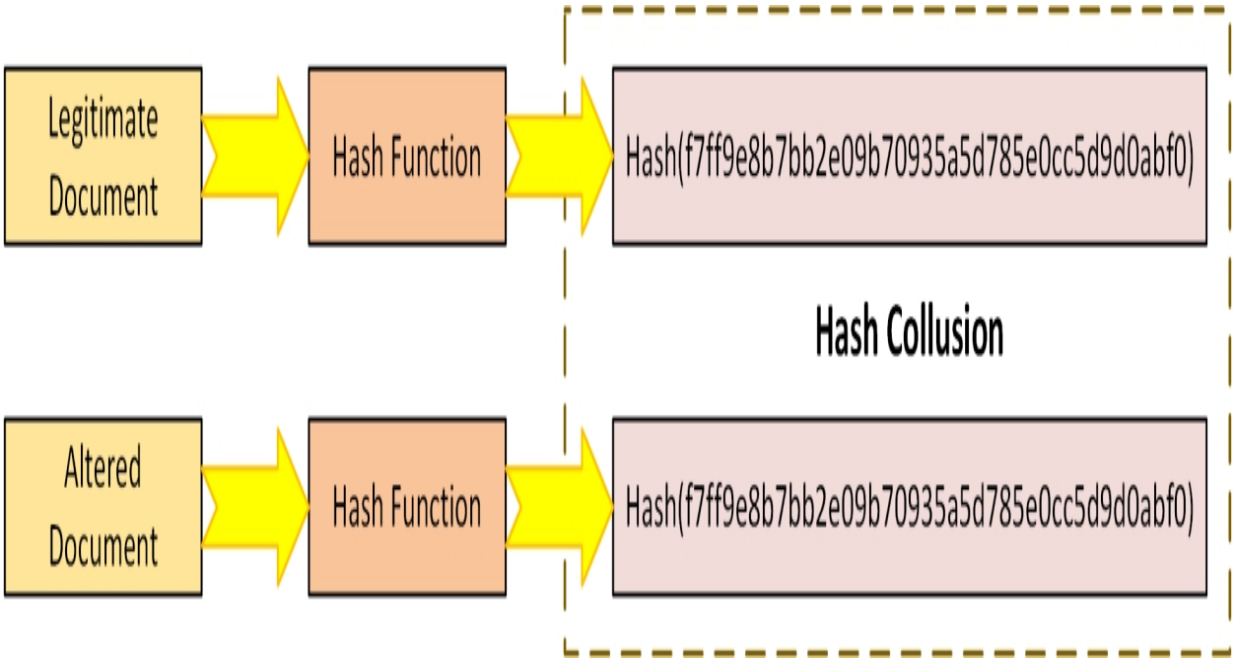
Adaptive Chosen Plaintext Attack

An Adaptive Chosen Plaintext Attack is a form of chosen plaintext cryptographic attack where the cryptanalyst issues a series of interactive queries, choosing subsequent plaintexts based on information from previous encryptions.

Rubber Hose Attack

A Rubber Hose Attack is the technique of obtaining information about cryptographic secrets such as passwords, keys, or encrypted files by torturing a person. *Collision*

Collision refers to a hash collision, which means two different plaintexts have the same hash value. This is a rare condition that is not supposed to exist in a hash algorithm. The hashing process accepts an infinite input length and produces a finite output. Consider a scenario where an attacker finds a hash collision among legitimate and altered documents. Now, being undetected the attacker can easily fool the target.

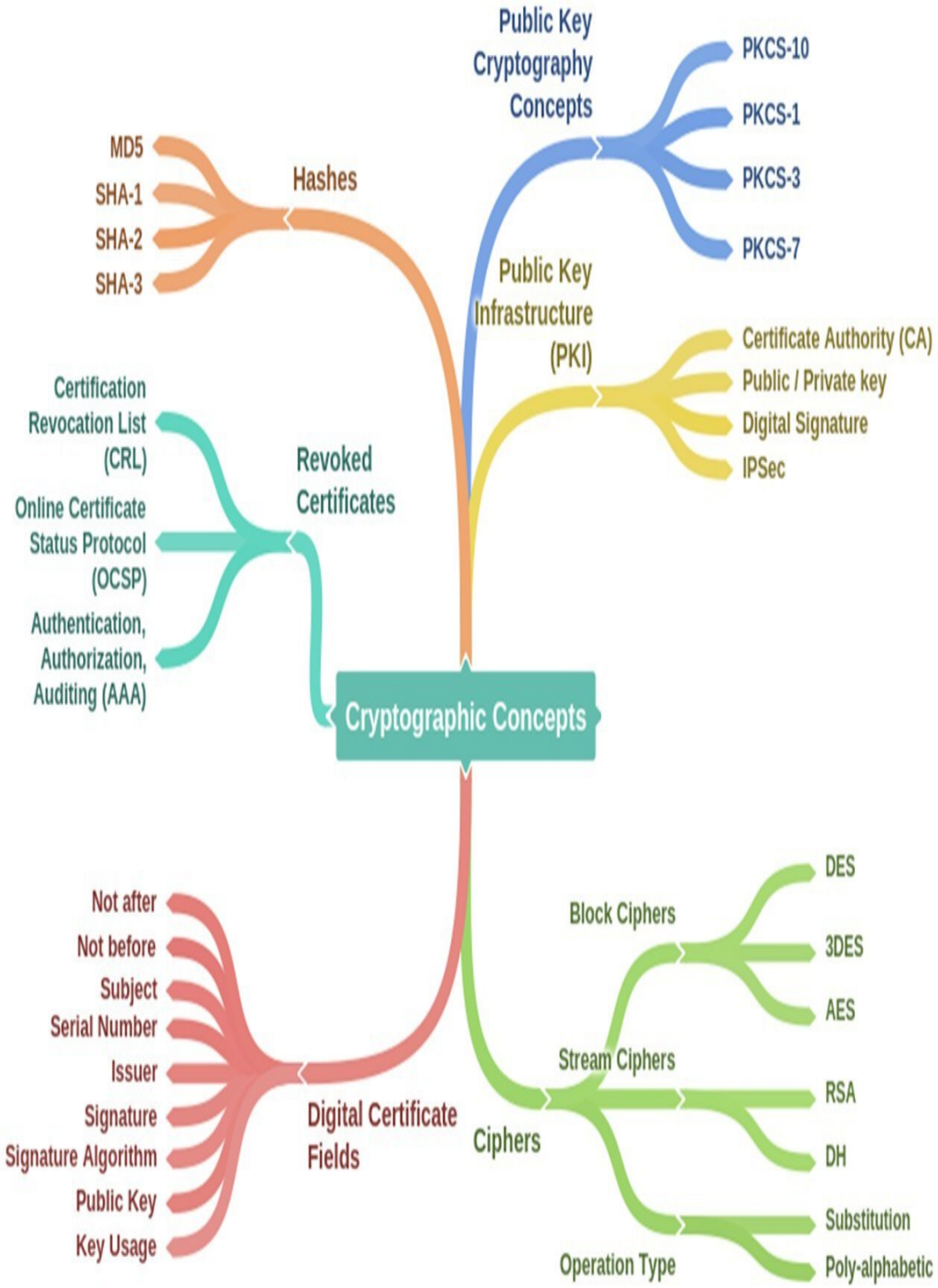


Code Breaking Methodologies

Code Breaking Methodology includes several tricks and techniques, for example, using social engineering, that are helpful to break encryption and expose the information in it such as cryptographic keys and messages. The following are some effective techniques and methodologies:

- Brute Force
- One-Time Pad
- Frequency Analysis

Mind Map


Cryptographic Concepts
Hashes
MD5
SHA-1
SHA-2
SHA-3
Public Key Cryptography Concepts
PKCS-10
PKCS-1
PKCS-3
PKCS-7
Public Key Infrastructure (PKI)
Certificate Authority (CA)
Public / Private key
Digital Signature
IPSec
Revoked Certificates
Certification Revocation List (CRL)
Online Certificate Status Protocol (OCSP)
Authentication, Authorization, Auditing (AAA)
Digital Certificate Fields
Not after
Not before
Subject
Serial Number
Issuer
Signature
Signature Algorithm
Public Key
Key Usage
Ciphers
Block Ciphers
DES
3DES
AES
Stream Ciphers
RSA
DH
Operation Type
Substitution
Poly-alphabetic

Practice Questions

1. Symmetric Key Cryptography requires:
A. Same Key for Encryption & Decryption
B. Different Keys for Encryption & Decryption
C. Public Key Cryptography
D. Digital Signatures

2. AES & DES are the examples of:
A. Symmetric Key Cryptography
B. Asymmetric Key Cryptography
C. Public Key Cryptography D. Stream Ciphers

3. The cipher that encrypts the plain text one by one is known as: A. Block Cipher
B. Stream Cipher
C. Mono-alphabetic Ciphers
D. Polyalphabetic Ciphers

4. 64-bit Block Size, 56-bit Key size, & 16 number of rounds are the parameters of:
A. DES
B. AES
C. RSA
D. RC6

5. Digital Certificate's "Subject" field shows: A. Certificate Holder's Name
B. Unique Number for Certificate Identification C. The Public Key of the Certificate Holder D. Signature Algorithm

6. RSA key length varies from: A. 5 12- 1024
B. 1024-2048
C. 5 12-2048
D. 1024-4096

7. The message digest is used to ensure: A. Confidentiality
B. Integrity