Chapter 20: Cryptography

Technology Brief

As we have studied earlier, confidentiality, integrity, and availability are the three basic components around which we should build and maintain our security model. We must know different methods by which we can implement each one of these features. For example, by using encryption, we can make sure that only the sender and the 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. This section explains the concepts and different methods by which we can implement encryption and hashing in our network. Several terminologies need to be explained before moving to the main agenda of this section.

Cryptography Concepts

Cryptography

Cryptography is a technique of encrypting the clear text data into a scrambled code. This encrypted data is sent over public or private network toward destination to ensure the 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 all about confidentiality, is also concern 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.

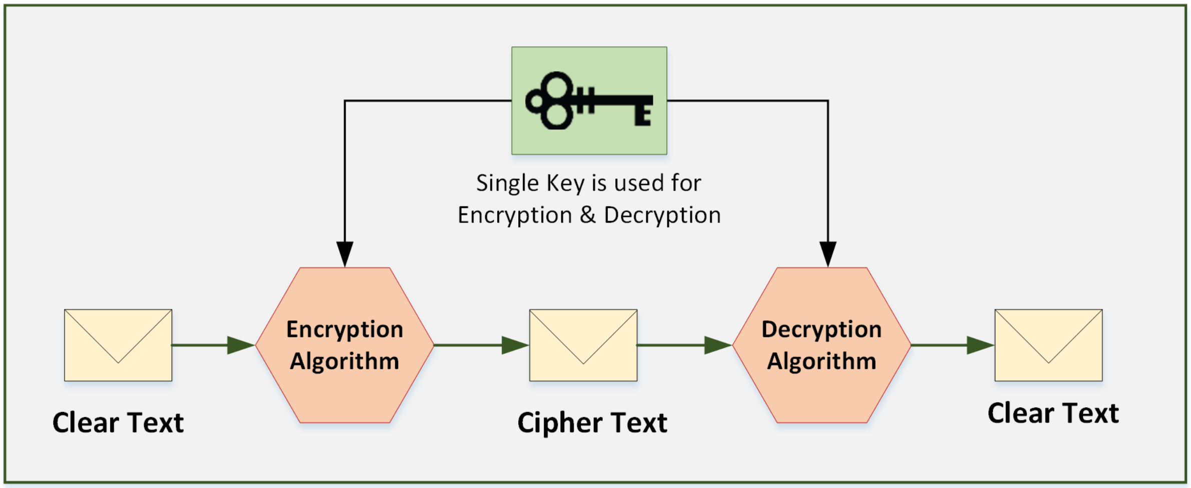


Figure 20-01. Symmetric Cryptography

Asymmetric Cryptography / Public Key Cryptography

Unlike Symmetric Ciphers, two keys are used. One key is publically known to everyone while one key is kept secret and is used to encrypt the data by 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 public key of the sender to decrypt the data. RSA, DSA and Diffie-Hellman Algorithm are popular examples of asymmetric ciphers. Asymmetric Key Cryptography delivers Confidentiality, integrity, authenticity and Non-repudiation by using Public and Private key concept. The private key is only known by the owner itself. Whereas, the Public key is issued by using Public Key Infrastructure (PKI) where a trusted Certification Authority (CA) certify the ownership of key pairs.

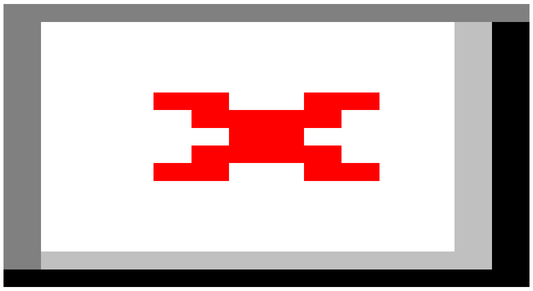


Figure 20-02. Asymmetric Cryptography

Government Access to Keys (GAK)

Government Access to keys (GAK) refers to the agreement between government and software companies. All or necessary keys are delivered to a governmental organization which keeps it securely and only uses them when the 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. Common methods by which ciphers replace original data with encrypted data are:

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 the character by shifting it three characters ahead of it. Therefore, "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 this example, the value of the key is three, and both nodes should know it 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," Key plays the main role in every cipher algorithm. Without knowing the key, data cannot be decrypted.

Stream Cipher

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

Block Cipher

A type of symmetric key cipher that encrypts the plain text on the fixed length of the group. 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 Algorithm (DES) is a Symmetric Key Algorithm that was used for encryption, but now, it is considered as insecure, however successors such as Triple DES, G-DES replaced DES encryption. DES uses 56-bit Key size that is too small to protect data consisting.

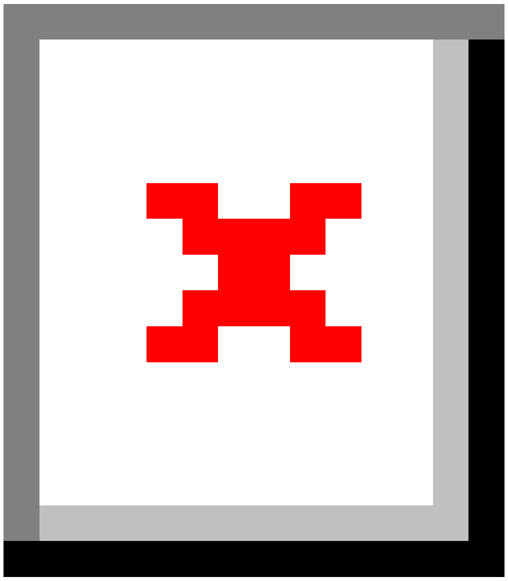


Figure 20-03. DES Algorithm

DES algorithm is consisting of 16 rounds processing the data with the 16 intermediary round keys of 48-bit generated from 56-bit cipher key by a Round Key Generator. Similarly, DES reverse cipher computes the data in clear text format from cipher text using same Cipher key.

The following are the major parameter of DES.

DES Algorithms Parameters	**Values**
Block size	64 bits
Key size	56 bits
Number of rounds	16
16 intermediary keys	48 bits

Table 20-01. DES Algorithm Parameters

Advanced Encryption Standard (AES)

When DES become insecure, and Performing DES encryption three times (3-DES or Triple-DES) took high computation and time, there was a need for another encryption algorithm that is more secure and effective than DES. "Rijndael" issues a new algorithm in 2000-2001 known as Advanced Encryption Algorithm (AES). AES is also a Private Key Symmetric Algorithm but stronger and faster than Triple-DES. AES can encrypt 128-bit data with 128/192/256 bit keys.

The following are the major parameter of AES.

AES Algorithms Parameters	AES-128	AES-192	AES-256
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 AES-128bit scenario. In 128-bit AES, there will be 10 rounds. Initial 9 rounds will be the performing the same step, i.e., Substitute bytes, shifting or rows, mixing of columns, and Adding round keys. The last round is slightly different with only Substitute bytes, shifting or rows and Adding round keys. The following figure shows the AES algorithm architecture.

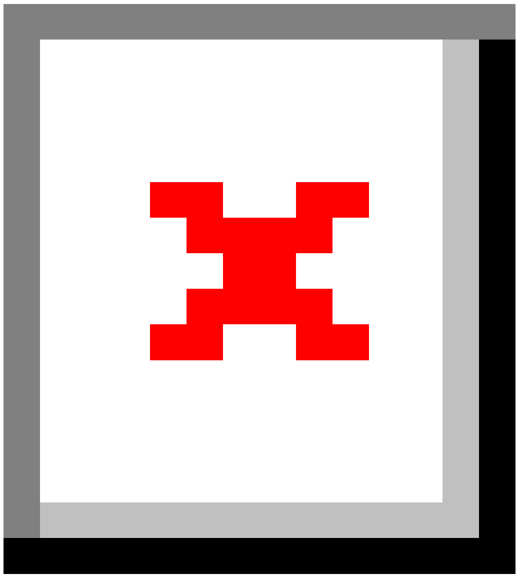


Figure 20-04. AES Algorithm

RC4, RC5, RC6 Algorithms

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

RC5 was a Symmetric Key Block Cipher introduce in 1994. RC5 has variable block size (32, 64 or 128 bit), Key size of 0 to 2040 bits and 0 to 255 rounds. RC5 is suggested with the 64-bit block size, 128-bit key and 12 rounds. 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 having a block size of 128-bits with 128, 192, 256 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

Using a basic concept of a signature, that we use in our daily life to prove the authenticity and actual origin of a document, in computer networking, Digital Signature Algorithm (DSA) is used. Digital Signature can provide three components of network security, i.e., Authenticity of a message, Integrity of a message, and Non-repudiation. The digital signature cannot provide the confidentiality of communication. However, it can be achieved by using encrypting the message and signature.

Digital Signature uses Public Key to sign and verify the packets. The signing of a document requires private key whereas Verification requires a Public key. The sender of the message signed with its private key and send it to the receiver. The receiver verifies the authenticity of the message by decrypting the packet with sender's public key. As sender's public key only decrypts the message, it verifies that sender of the 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. Integrity is signing and verifying the message obtained by using Hash Functions.

Digital Certificate contains various items and these items are listed below:

- **Subject**–Certificate holder's name.
- **Serial Number**-Unique number for certificate identification.
- **Public Key**–A copy of public of the certificate holder.
- **Issuer**–Certificate issuing authority's digital signature to verify that the certificate is real.
- **Signature Algorithm**–Algorithm used to digitally sign the certificate by the Certification Authority (CA).
- **Validity**–Validity of a certificate or we can say expiry date and time of the certificate.

The Digital certificate has X.509 version supported the format, and that is a standard format.

RSA (Rivest Shamir Adleman)

This algorithm is named after its creators, namely Ron Rivest, Adi Shamir, and Leonard Adleman. Also known as public key cryptography standard (PKCS) #1, the main purpose of its usage today is authentication. The key length varies from 512 to 2048 with 1024 being preferred one. 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. Calculates ϕ = (p -1) * (q - 1).
4. Chooses a random integer "e" i.e Encryption Key and calculates "d" Decryption Key so that d x e = 1 mod ϕ.
5. Announce "e" and "n" to the public; he keeps " ϕ" and "d" secret.

Lab 20-1: Example of RSA Algorithm

Case Study:

Alice creates a pair of keys for herself. She chooses p = 17 and q = 11. Calculate the value of following.

A- Calculate

$n = ?$

$\phi = ?$

B- She then chooses e =7

$d = ?$

C- Show how Bob can send a message “**88**” to Alice if he knows e and n.

A- Solution:

As we know:

$n = p * q$

$n= 17 * 11$

$\boxed{n= 187}$

Let’s find ϕ:

$\Phi= (p -1) * (q - 1)$

$\Phi= (17-1) * (11-1)$

$\Phi= (16) * (10)$

$\boxed{\Phi= 160}$

B- Solution:

If e = 7; let’s calculate the value of d

As we know that:

$d \text{ x } e = 1 \bmod \phi.$

$d= e^{-1} \bmod \phi$

$d= 7^{-1} \bmod 160$

$\boxed{d= 23}$

C- Solution:

Private Key of Alice will be (d,p,q) = (23,17,11)

Public Key of Alice will be (e,n) = (7,187)

Alice will share her Public key with Bob. Bob will encrypt the packet using Alice Public key and send it to her.

As we know:

$$\mathbf{C=M^e \bmod n}$$

Here:

"**C**" is Ciphertext

"**M**" is Message

$C= M^e \bmod n$

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

$\boxed{C= 11}$

Bob will send "**11**" to Alice. Alice will decrypt the Cipher using her private key to extract the original message.

As we know:

$$\mathbf{M= C^d \bmod n}$$

$M= (11)^{23} \bmod 187$

$\boxed{M= 88}$

Message Digest (One-way Hash) Functions

The message digest is a cryptographic hashing technique that is used to ensure the integrity of a message. Message and message digest can be sent together or separately through a communication channel. Receiver recalculates the Hash of the message and compares it with the Message digest to ensure if any changes have been made. One-Way-Hash of Message digest means the hashing function must be a one-way operation. The original message must not be able to recreate. 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 MD5 algorithm is one from the Message digest series. MD5 produces a 128-bit hash value that is used as a checksum to verify the integrity. Hashing is the technique to ensure the integrity. The hash value is calculated by computing specific algorithms to verify the integrity that the data was not modified. Hash values play an important role in proving the integrity not only of documents and images but also used in protocols to ensures the integrity of transporting payload.

Secure Hashing Algorithm (SHA)

As Message Digest 5 (MD5) is a cryptographic hashing algorithm, another most popular, more secure and widely used hashing algorithm is Secure Hashing Algorithm (SHA). SHA-1 is a secure hashing algorithm producing 160-bit hashing value as compared to MD5 producing 128-bit value. However, SHA-2 is even more secure, robust and safer hashing algorithm now.

Syntax: **The password is 12345**
SHA-1: **567c552b6b559eb6373ce55a43326ba3db92dcbf**

Secure Hash Algorithm 2 (SHA-2)

SHA2 has the option to vary digest between 224 bits to 512 bits. SHA-2 is a group of different hashes including SHA-256, SHA-384 and SHA 512. The stronger cryptographic algorithm will minimize the chances of compromise.

SHA-256

Syntax: **The password is 12345**
SHA-256: **5da923a6598f034d91f375f73143b2b2f58be8a1c9417886d5966968b7f79674**

SHA-384

Syntax: **The password is 12345**
SHA-384:
929f4c12885cb73d05b90dc825f70c2de64ea721e15587deb34309991f6d57114500465243ba08a554f8fe7c8dbbca04

SHA-512

Syntax: **The password is 12345**
SHA-512:
1d967a52ceb738316e85d94439dbb112dbcb8b7277885b76c849a80905ab370dc11d2b84dcc88d61393117de483a950ee253fba0d26b5b168744b94af2958145

Hashed Message Authentication Code (HMAC)

HMAC uses the mechanism of hashing, but it adds another feature of using the 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 hash. By using HMAC, if there is an attacker who is eavesdropping, it will not be able to inject or modify the data and recalculate the correct hash because he will not know the correct key used by HMAC.

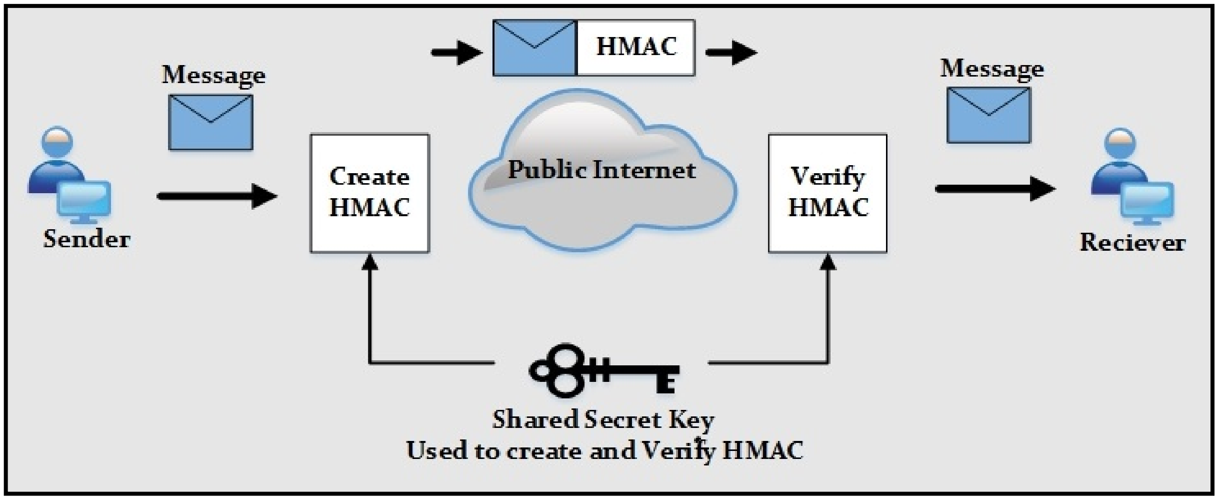


Figure 20-05. HMAC Working Conceptual Diagram

SSH (Secure Shell)

Secure Shell Protocol, commonly known in short as SSH protocol is a protocol that is 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 login but also with other protocols such as File Transfer Protocol (FTP), 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 SSH client connects to SSH server through a secure SSH channel over an insecure network.

Secure Shell (SSH) protocol is 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 be 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 client-side 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

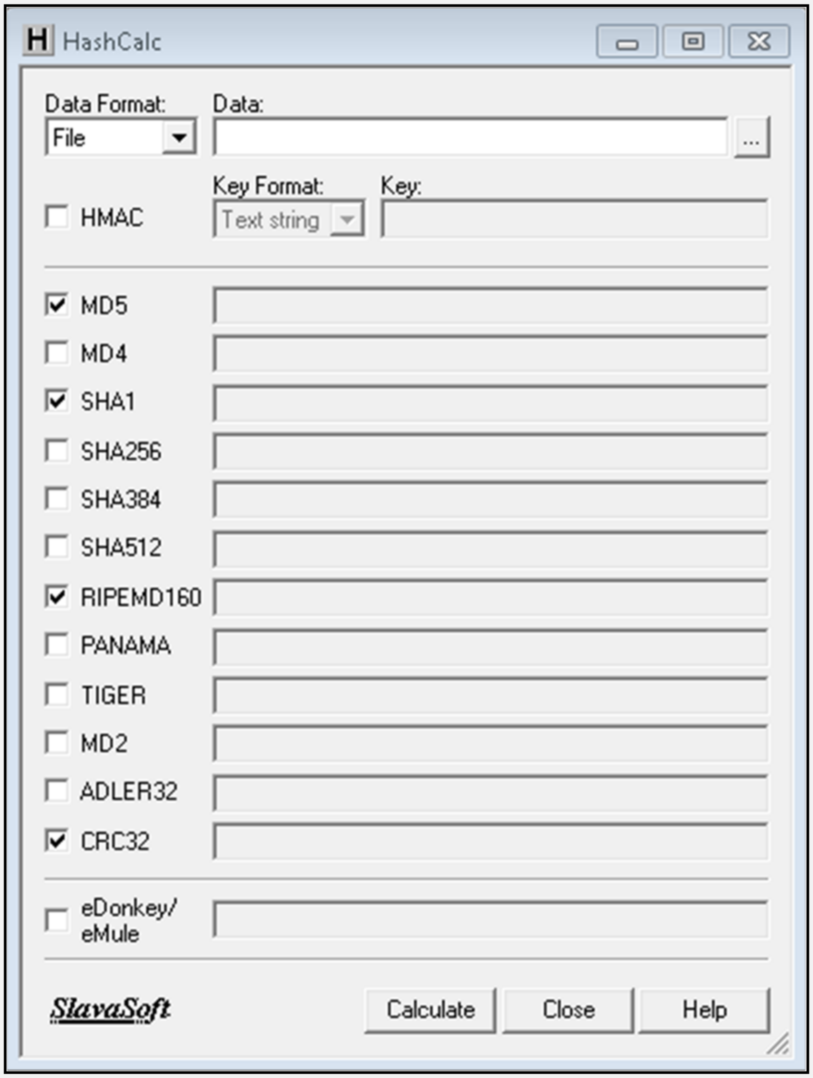
There are several MD5 Calculating tools available which can directly calculate the Hash value of text as well as offers to upload the desired file. Most popular tools are:

1. HashCalc
2. MD5 Calculator
3. HashMyFiles

Lab 20-2: Calculating MD5 using Tool

Calculating MD5 value using HashCalc

1. Open HashCalc tool.



HashCalc

Data Format: File

Data:

☐ HMAC

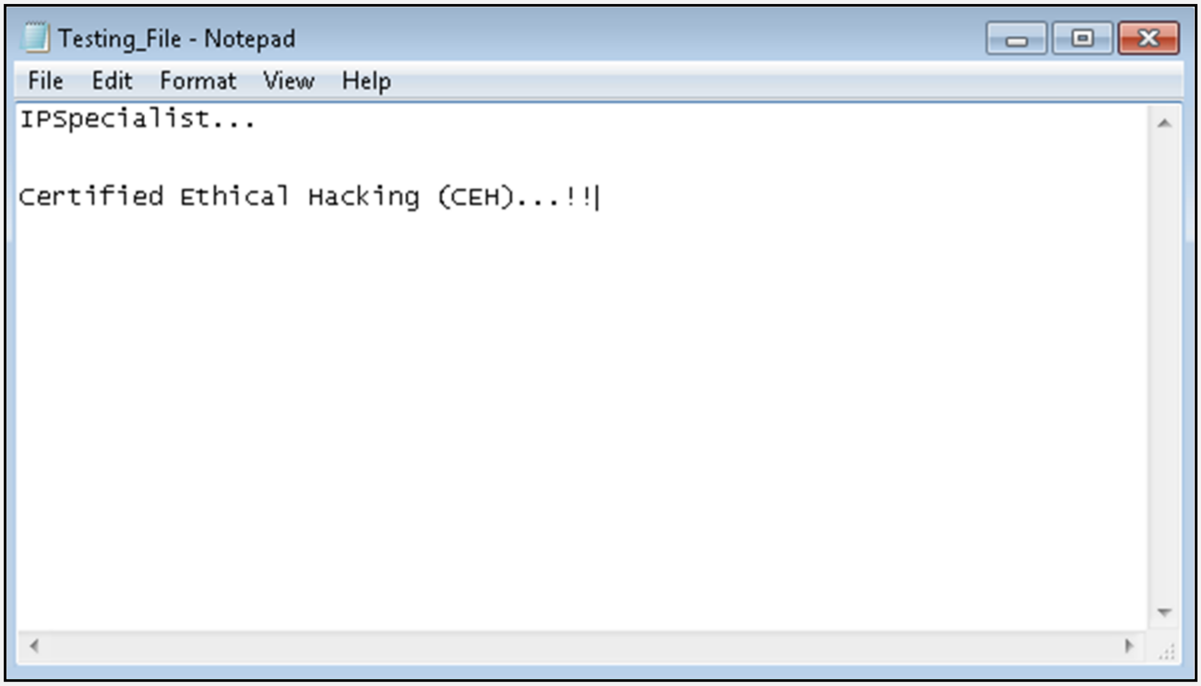
Key Format: Text string

Key:

☑ MD5
☐ MD4
☑ SHA1
☐ SHA256
☐ SHA384
☐ SHA512
☑ RIPEMD160
☐ PANAMA
☐ TIGER
☐ MD2
☐ ADLER32
☑ CRC32
☐ eDonkey/ eMule

Figure 20-06. HashCalc Tool

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

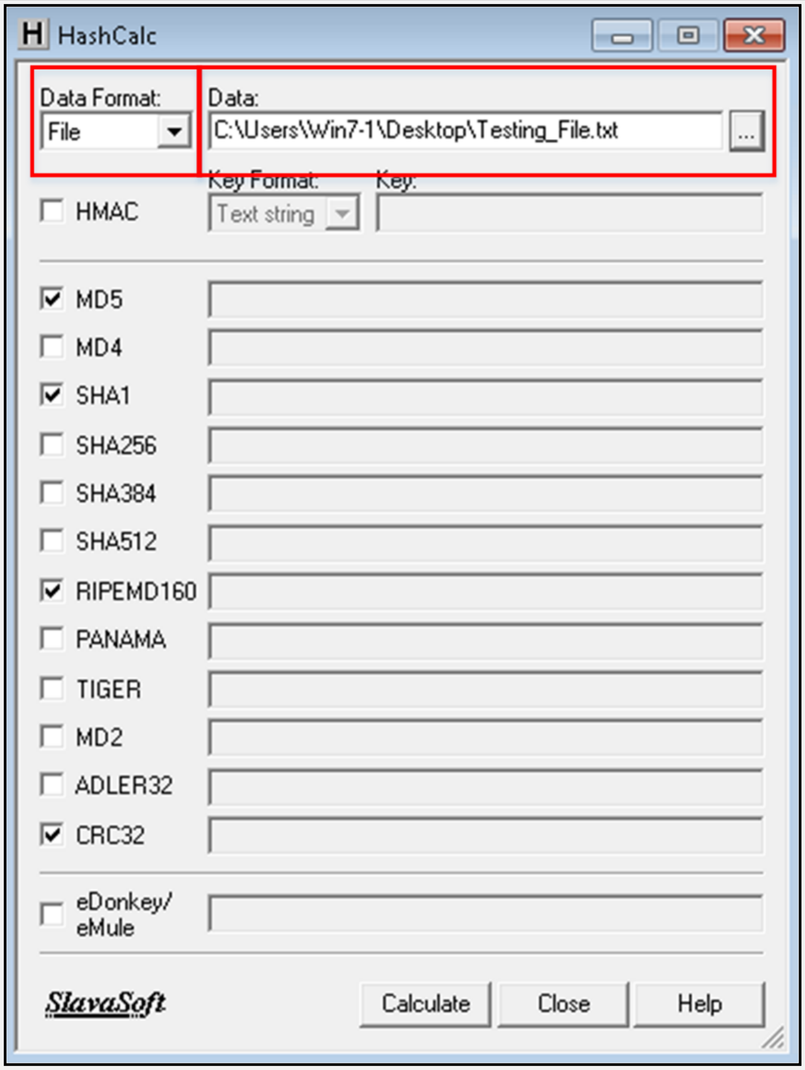


```
IPSpecialist...

Certified Ethical Hacking (CEH)...!!
```

Figure 20-07. Creating File for MD5 Calculation

3. Select Data Format as “File” and upload your file

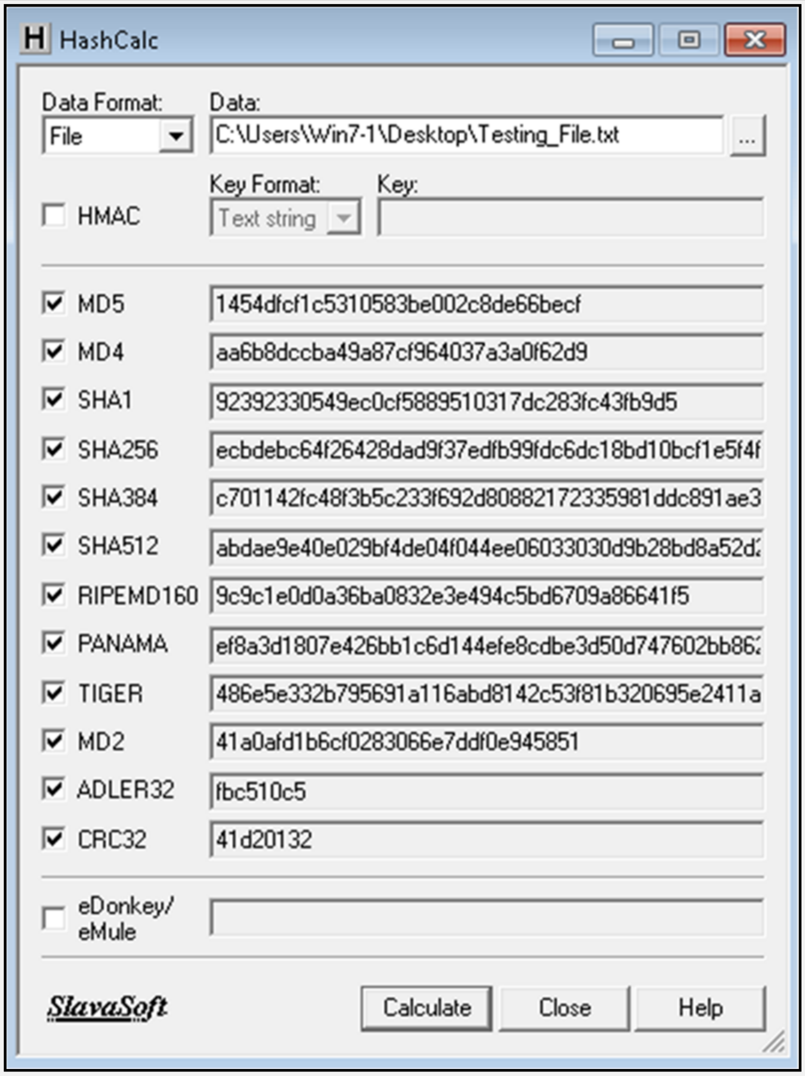


Data Format: File

Data: C:\Users\Win7-1\Desktop\Testing_File.txt

Figure 20-08. Uploading File to calculate Hash

4. Select Hashing Algorithm and Click Calculate



Data Format: File
Data: C:\Users\Win7-1\Desktop\Testing_File.txt

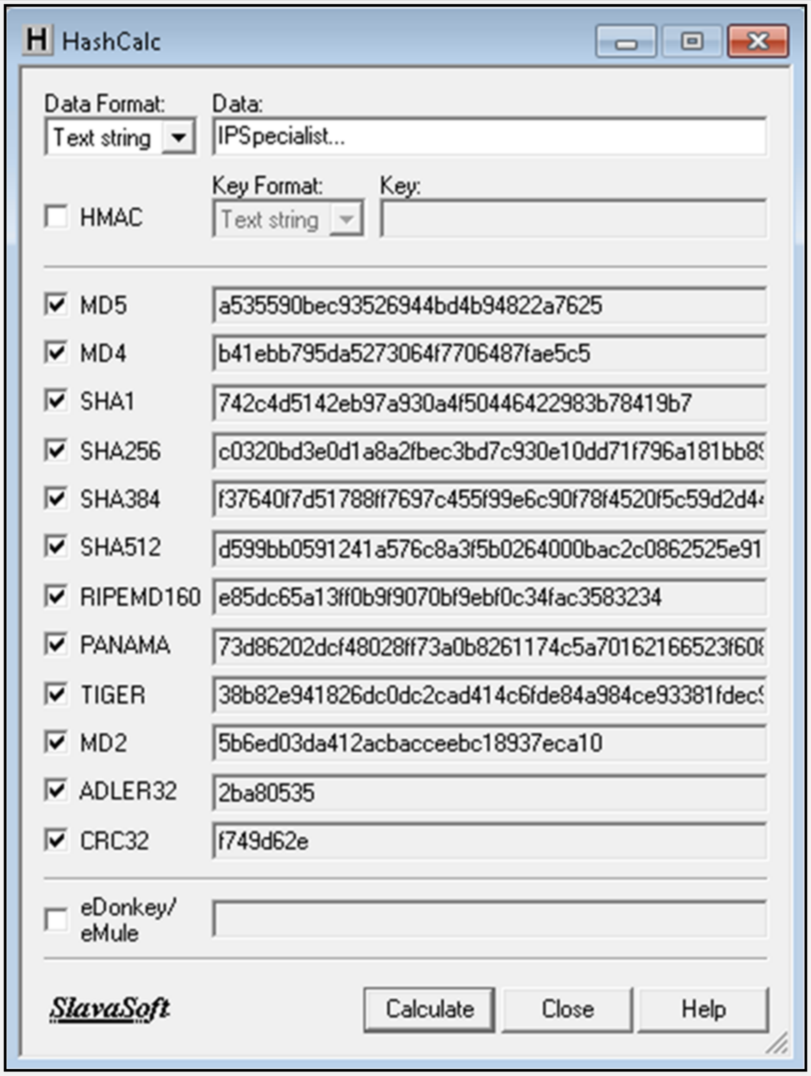
□ HMAC Key Format: Text string Key:

☑ 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

Figure 20-09. Calculating Hash

5. Now Select the Data Format to “**Text String**” and Type “**IPSpecialist…**” into Data filed and calculated MD5.

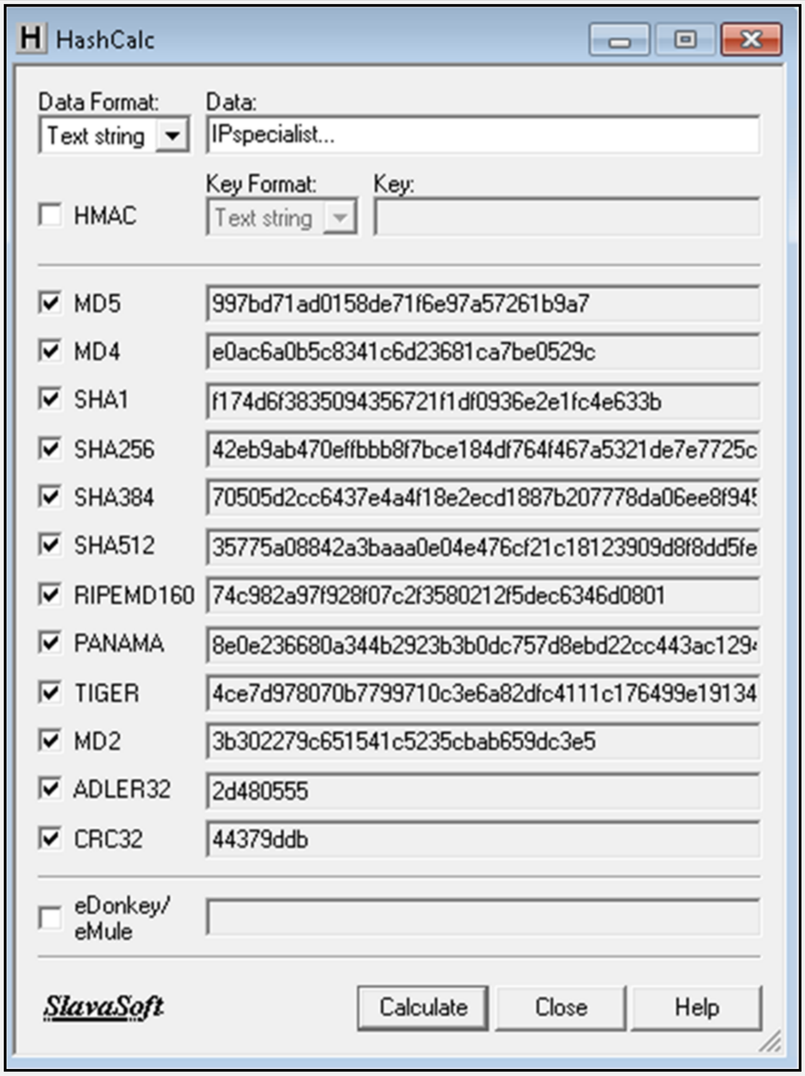


Data Format:	Text string
Data:	IPSpecialist...
□ HMAC	Key Format: Text string Key:
☑ MD5	a535590bec93526944bd4b94822a7625
☑ MD4	b41ebb795da5273064f7706487fae5c5
☑ SHA1	742c4d5142eb97a930a4f50446422983b78419b7
☑ SHA256	c0320bd3e0d1a8a2fbec3bd7c930e10dd71f796a181bb8
☑ SHA384	f37640f7d51788ff7697c455f99e6c90f78f4520f5c59d2d4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

Figure 20-10. Calculating Hash with Text String

MD5 Calculated for the text string “**IPSpecialist…**” is “**a535590bec93526944bd4b94822a7625**”

6. Now, let's see how MD5 value is changed from minor change.



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

Figure 20-11. Comparing Hash of different Text String

Just lowering the case of single alphabet changes entire hashing value. MD5 Calculated for the text string "**IPspecialist…**" is "**997bd71ad0158de71f6e97a57261b9a7**"

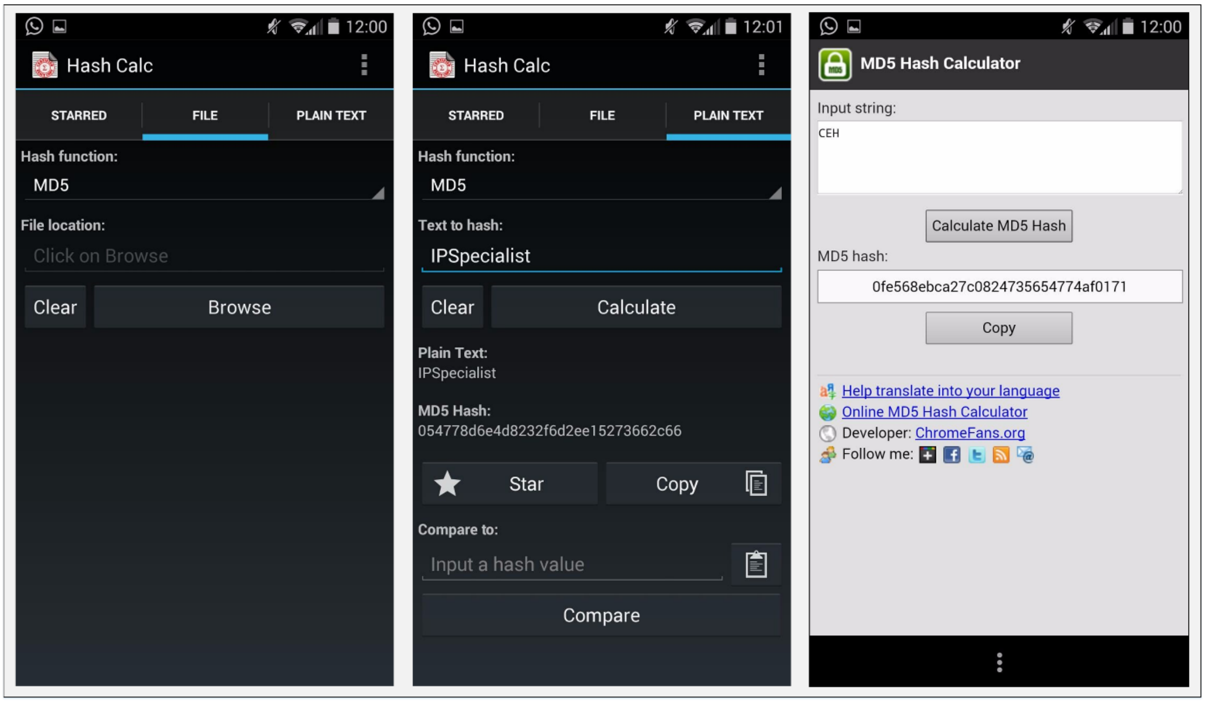
String	MD5
IPSpecialist…	a535590bec93526944bd4b94822a7625
IPspecialist…	997bd71ad0158de71f6e97a57261b9a7

Table 20-03. Comparing MD5 Values

Hash Calculators for Mobile:

Hash calculating tools for Mobile phones are:

- MD5 Hash Calculator
- Hash Droid
- Hash Calculator



Hash function: MD5

File location: Click on Browse

Hash function: MD5

Text to hash: IPSpecialist

Plain Text: IPSpecialist

MD5 Hash: 054778d6e4d8232f6d2ee15273662c66

Compare to: Input a hash value

Input string: CEH

MD5 hash: 0fe568ebca27c0824735654774af0171

Figure 20-12. Hashing tools for Mobile

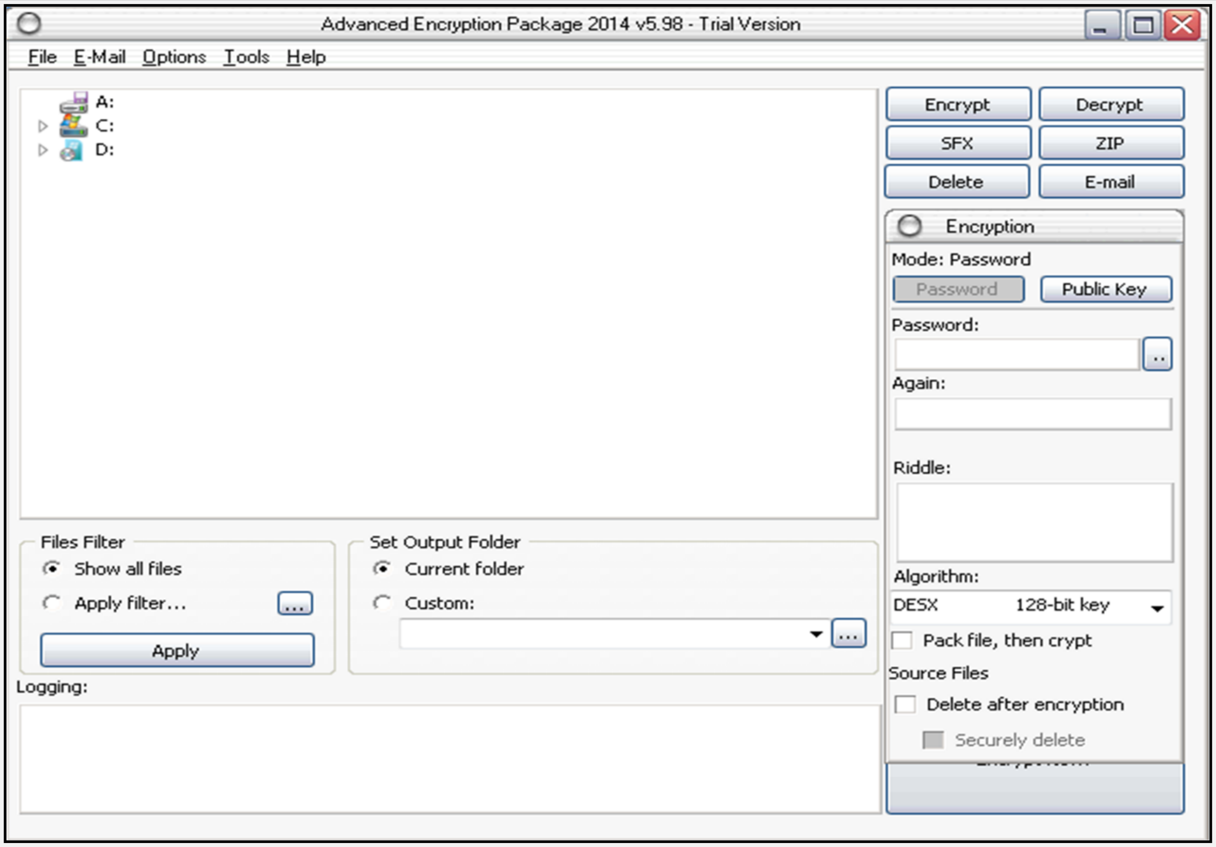
Cryptography Tool

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

Lab 20-3: Advanced Encryption Package 2014

Procedure:

1. Download and Install Advance Encryption Package Latest Version. In this Lab, we are using Advanced Encryption Package 2014 and 2017 to ensure compatibilities on Windows 7 and Windows 10.



Advanced Encryption Package 2014 v5.98 - Trial Version

Mode: Password

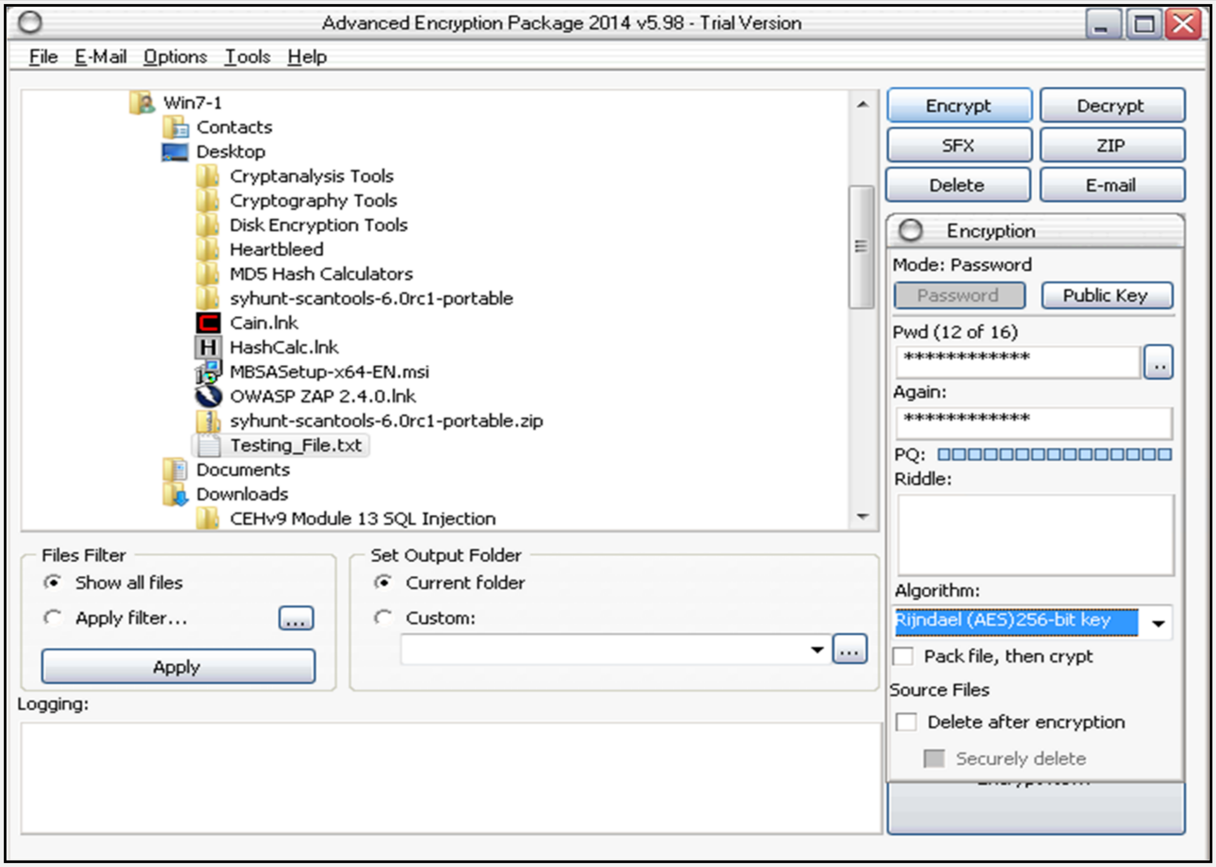
Algorithm: DESX 128-bit key

● Show all files

● Current folder

Figure 20-13. Advanced Encryption Package 2014

2. Select the File you want to Encrypt.
3. Set password
4. Select Algorithm



Mode: Password

Pwd (12 of 16) ************

Again: ************

Algorithm: Rijndael (AES)256-bit key

□ Pack file, then crypt

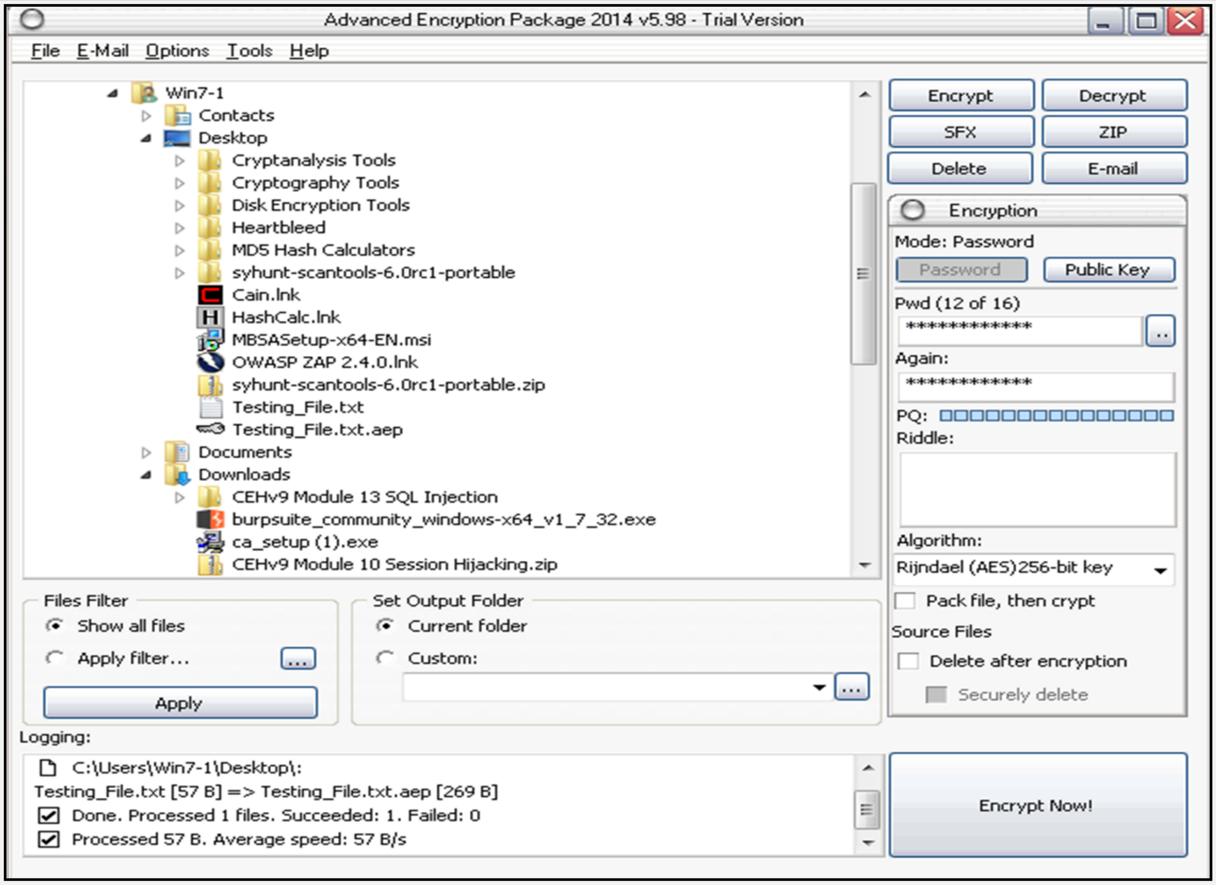
□ Delete after encryption

Files Filter: ● Show all files ○ Apply filter…

Set Output Folder: ● Current folder ○ Custom:

Figure 20-14. Uploading File to Encrypt

5. Click Encrypt



Mode: Password

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…

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
```

Figure 20-15. Encrypting with AES 256-bit

6. Compare both Files

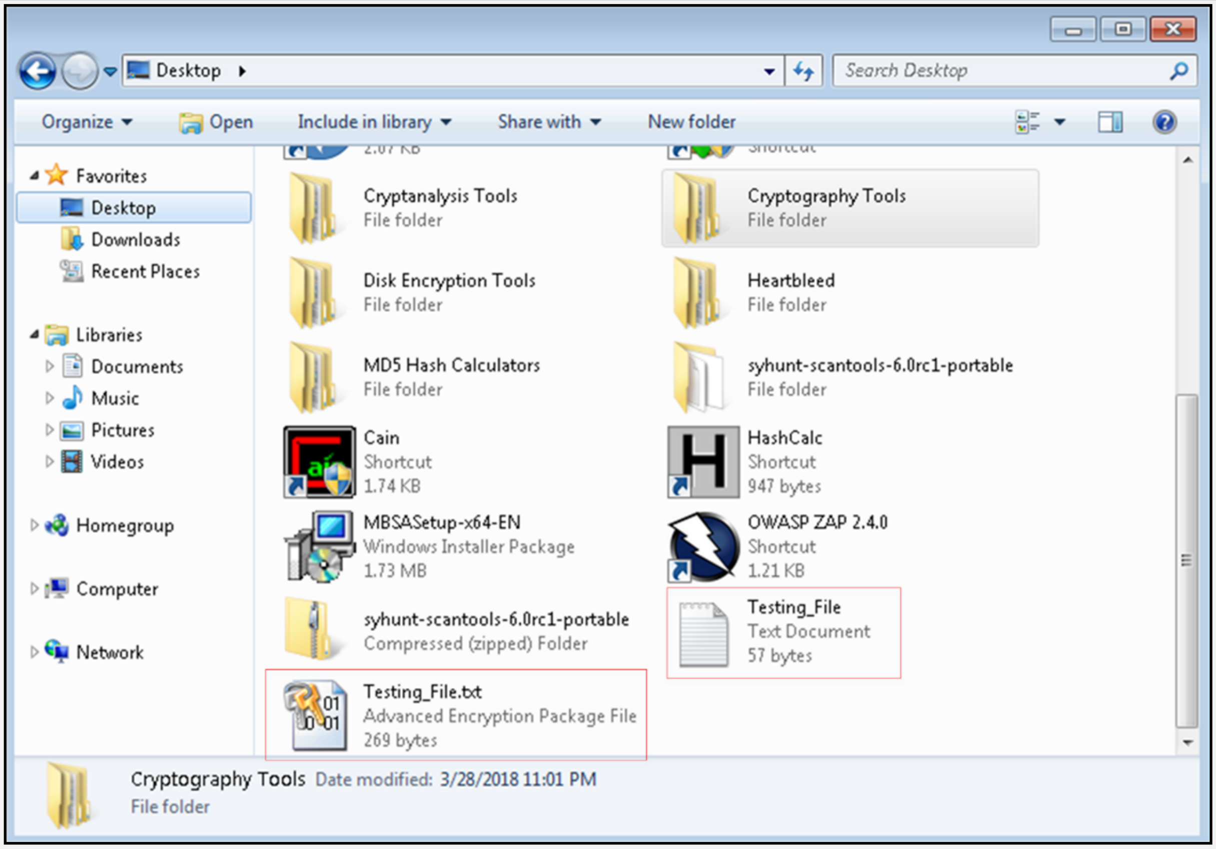
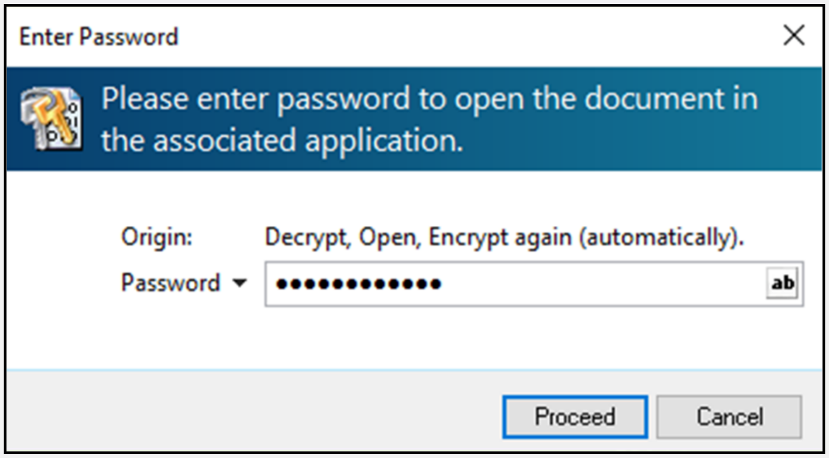


Figure 20-16. Comparing Encrypted and Original Files

7. Now, After forwarding it to another PC, in our case, in Windows 10 PC, decrypting it using Advanced Encryption package 2017.
8. Enter password


Enter Password

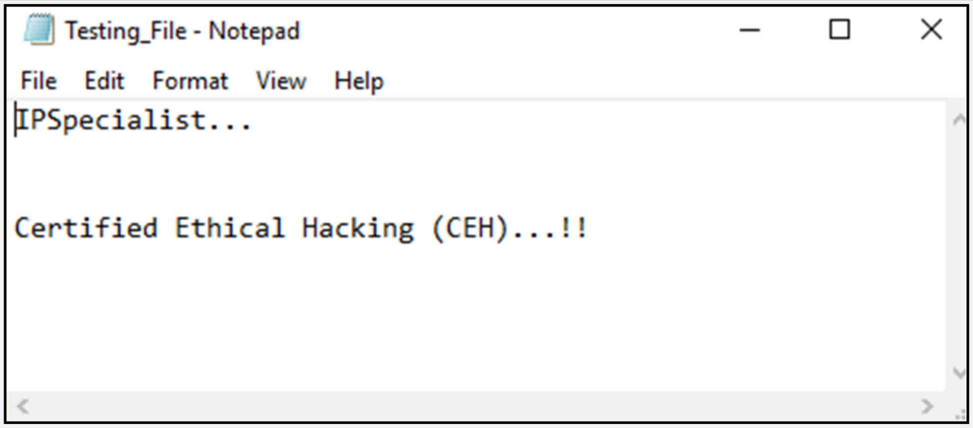
Please enter password to open the document in the associated application.

Origin: Decrypt, Open, Encrypt again (automatically).

Password: ●●●●●●●●●●●●

Figure 20-17. Decrypting File using Advanced Encryption Package 2017

9. File Successfully decrypted.



```
IPSpecialist...


Certified Ethical Hacking (CEH)...!!
```

Figure 20-18. 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.
Before moving to the original discussion, basic terminologies needs explanation.

Public and Private Key Pair

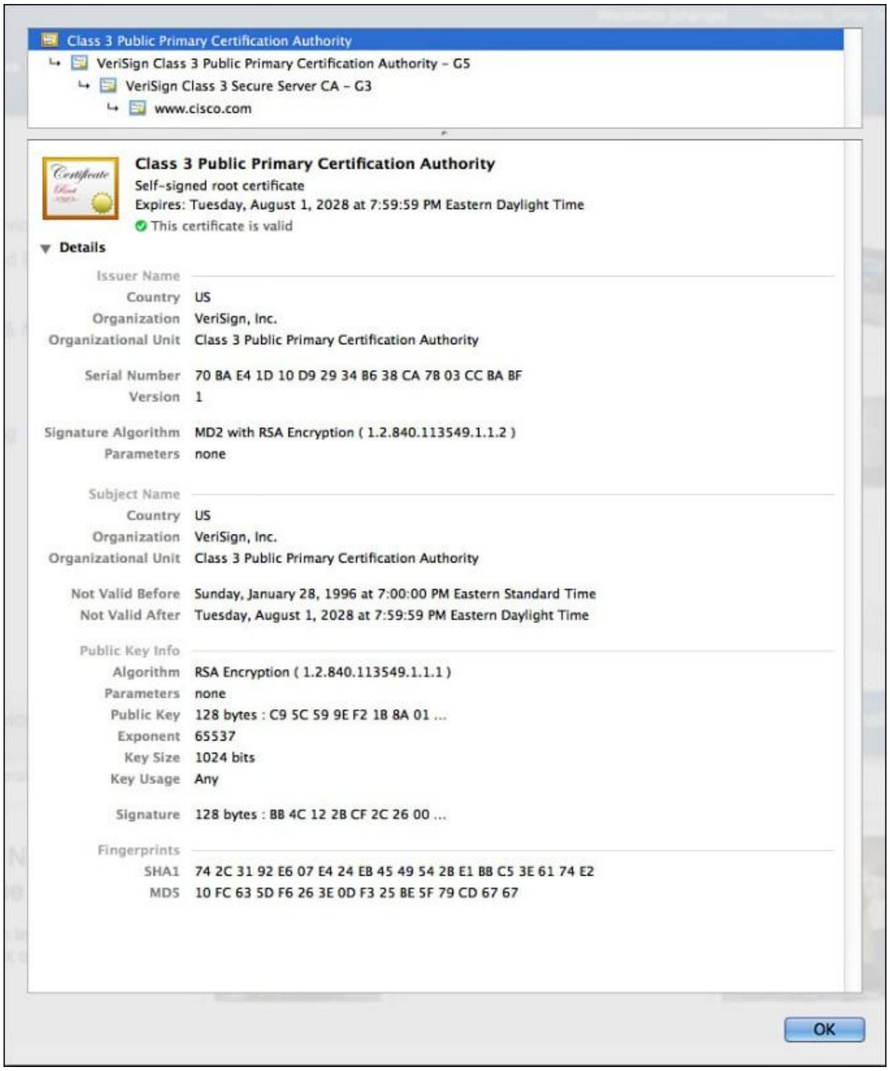
Public and Private Key pair work like a team in encryption/decryption process. The public key is provided to everyone, and the private key is secret. Every device makes sure that no one has its private key. We encrypt data sending to a particular node by using its public key. Similarly, the private key is used to decrypt the data. It is also true in the opposite case. If a node encrypts a data with its private key, the public key is for decryption.

Certification Authorities (CA)

A certificate authority (CA) is a computer or entity that creates and issues digital certificates. Number of things like 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

Root certificate provides the public key and other details of CA. An example of one is:



```
Class 3 Public Primary Certification Authority
 ↳ VeriSign Class 3 Public Primary Certification Authority – G5
    ↳ VeriSign Class 3 Secure Server CA – G3
       ↳ www.cisco.com
```

Class 3 Public Primary Certification Authority
Self-signed root certificate
Expires: Tuesday, August 1, 2028 at 7:59:59 PM Eastern Daylight Time
✓ This certificate is valid

▼ **Details**

```
        Issuer Name
            Country  US
       Organization  VeriSign, Inc.
Organizational Unit  Class 3 Public Primary Certification Authority

      Serial Number  70 BA E4 1D 10 D9 29 34 B6 38 CA 7B 03 CC BA BF
            Version  1

Signature Algorithm  MD2 with RSA Encryption ( 1.2.840.113549.1.1.2 )
         Parameters  none

       Subject Name
            Country  US
       Organization  VeriSign, Inc.
Organizational Unit  Class 3 Public Primary Certification Authority

   Not Valid Before  Sunday, January 28, 1996 at 7:00:00 PM Eastern Standard Time
    Not Valid After  Tuesday, August 1, 2028 at 7:59:59 PM Eastern Daylight Time

    Public Key Info
          Algorithm  RSA Encryption ( 1.2.840.113549.1.1.1 )
         Parameters  none
         Public Key  128 bytes : C9 5C 59 9E F2 1B 8A 01 ...
           Exponent  65537
           Key Size  1024 bits
          Key Usage  Any

          Signature  128 bytes : BB 4C 12 2B CF 2C 26 00 ...

       Fingerprints
               SHA1  74 2C 31 92 E6 07 E4 24 EB 45 49 54 2B E1 BB C5 3E 61 74 E2
                MD5  10 FC 63 5D F6 26 3E 0D F3 25 BE 5F 79 CD 67 67
```

Figure 20-19. Example Root Certificate

There are multiple informative sections in the figure above including a serial number, issuer, country and organization names, validity dates, and public key itself. Every OS has its placement procedure regarding the certificates. 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 identity certificate is similar to root certificate except that it provides the public key and identity of client computer or device. For example, a client router or web server who wishes to make SSL connections with other peers.

Signed Certificate Vs. Self Signed Certificate

Self-Singed Certificates and Signed Certificates from a Certificate Authority (CA) provide security in the same way. Communication using these types of certificates are protected encrypted by high-level security. Presence of Certificate Authority implies that a trusted source certifies communication. The signed security certificate is to be purchase whereas 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.

Email Encryption

Digital Signature

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

There are two categories of Digital Signatures:

1. Direct Digital Signature
2. Arbitrated Digital Signature

Direct Digital Signature

Direct Digital Signatures involves only sender and receiver of the message assuming that receiver has sender's Public Key. The sender may sign entire message or hash with the private key and send it towards 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 date and then send it to the recipient. It requires a suitable level of trust and can implement 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 remote VPN. In public, there is no IPsec software running. Normal users also need to do encryption in different cases like online banking, electronic shopping. In such situations, SSL comes in to play. The good thing about Secure Socket Layer (SSL) is that almost every single web browser in use today supports SSL. By using SSL, the web browser makes an HTTPS-based session with the server instead of HTTP. Whenever a browser tries to make HTTPS based session with a server, a certificate request is sent to the server in the background. The server in return, reply 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 certificate is valid, now 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) are often used interchangeably, and provide encryption and authentication of data in motion. These protocols are intended for a scenario where users want secure communication over an unsecured network like 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 may be needed. SSL only spills the data after a process known as 'handshake.' If a hacker bypasses the encryption process than everything from bank account information to a secret conversation is visible which malicious users may use for personal gain.

SSL was developed by Netscape in 1994 with an intention to protect web transactions. The last version for 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.
- Protection of normal TCP/UDP connection

Working

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 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 algorithms.
- **Sub-phase 2.** This phase includes one-way or two-way authentication between client and server end. A master key from is sent by client side by using server's public key to start protecting the session.
- **Sub-phase 3.** The last phase calculates a session key, and cipher suite is finally activated. HMAC provides data integrity features by using either SHA-1 or MD5. Similarly, using DES-40, DES-CBC, 3DEC-EDE, 3DES-CBC, RC4-40, or RC4-128 provides confidentiality features.
    - ❖ **Session Keys Creation.** Methods for generating session keys are as follows:
        - ▪ *RSA Based.* Using public key of peer encrypts shared secret string.

- *A fixed DH Key Exchange.* Fixed Diffie-Hellman based key exchanged in a certificate creates a session key.
- *An ephemeral DH Key Exchange.* It is considered to be the best protection option as actual DH value is signed with the private key of the sender, 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 in the middle attacks.

Phase 2 (Secure Data Transfer)

In this phase, secure data transfer takes place between encapsulating endpoints. Each SSL session has unique session ID which exchanges during the authentication process. The session ID is used to differentiate between old and new session. The client can request the server to resume the session based on this ID (In case, sever has a session ID in its cache).
TLS 1.0 is considered to be a bit more secure than the last version of SSL (SSL v3.0). Even US Government has also declared not to use SSL v3.0 for highly sensitive communications due to latest vulnerability named as POODLE. After POODLE vulnerability, most web browsers have disabled SSL v3.0 for most of the communication and services. Current browsers (Google Chrome, Firefox, and others) support TLS 1.0 by default and latest versions of TLS (TLS 1.1 and TLS 1.2) optionally. TLS 1.0 is considered to be equivalent to SSL3.0. However, newer versions of TLS are considered to be far more secure than SSL. Keep in mind that SSL v3.0 and TLS 1.0 is not compatible with each other as TLS uses Diffie-Hellman and Data Security Standard (DSS) while SSL uses RSA.

Apart from secure web browsing by using HTTPS, SSL/TLS can also use for securing other protocols like FTP, SMTP, and SNTP, and other protocols.

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 the PGP software, created by Phil Zimmermann. The main purpose of OpenPGP is to ensure an 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 disk to secure files and directories by converting into an encrypted format. Disk encryption encrypts every bit on disk to prevent unauthorized access to data storage. There are several disk encryption tools available to secure disk volume such as:

- Symantec Drive Encryption
- GiliSoft Full Disk Encryption

Cryptography Attacks

Cryptography attacks are intended to recover encryption key. Once an attacker has the encryption key, he can decrypt all messages. Weak encryption algorithms are not resistant enough to 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

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.

Cipher-text 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 plain text messages as much as possible to guess the key. Once the attacker has the encryption key, it 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 chosen plaintext attack, the attacker has information about encryption algorithm or may have access to the workstation encrypting the messages. The attacker sends chosen plaintexts through encryption algorithm to extract ciphertexts and then encryption key. Chosen plaintext attack is vulnerable in the scenario where public key cryptography is being in use, and the public key is used to encrypt the message. In the worst case, an attacker can expose sensitive information.

Chosen Cipher-text 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 Cipher-text Attack

Adaptively chosen ciphertext attack is an interactive type of chosen plaintext attack where an attacker sends some ciphertexts to be decrypted and observe the results of decryption. Adaptively chosen ciphertext attacks gradually reveal the information about 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 the information from the previous encryptions.

Rubber Hose Attack

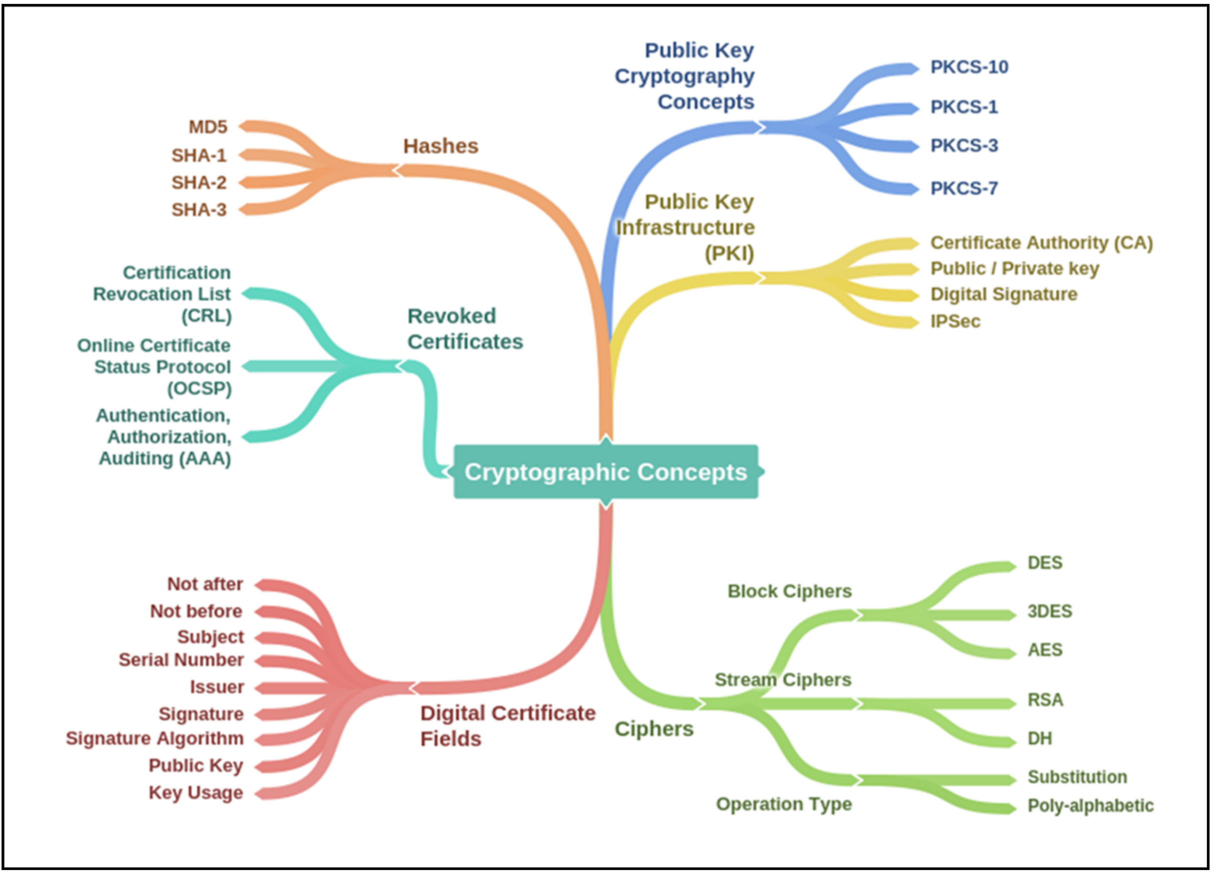
Rubber hose attack is a technique of gaining information about cryptographic secret such as passwords, keys, encrypted files, by torturing a person.

Code Breaking Methodologies

Code Breaking Methodology includes several tricks and techniques such as through social engineering techniques which are helpful to break encryption and expose the information in it like cryptographic keys and message. 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
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
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
Ciphers
Block Ciphers
DES
3DES
AES
Stream Ciphers
RSA
DH
Operation Type
Substitution
Poly-alphabetic

g