VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Digital Assignment, APRIL 2021

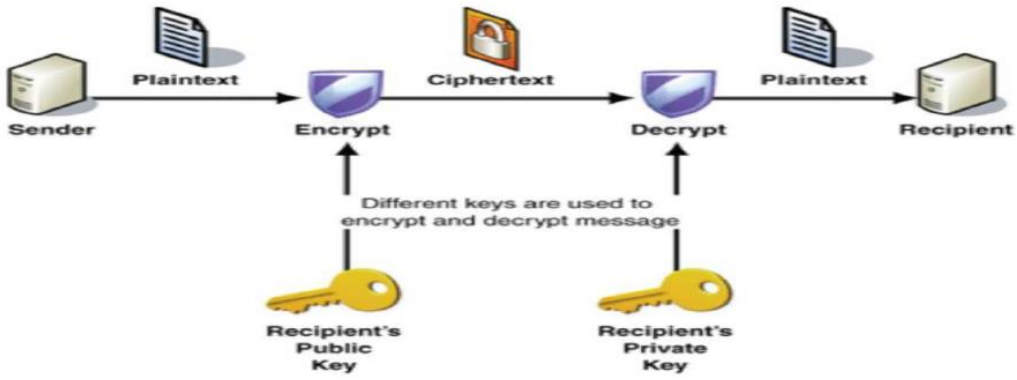
B.Tech., Winter-2020-2021

NAME	PRIYAL BHARDWAJ
REG. NO.	18BIT0272
COURSE CODE	ITE4001
COURSE NAME	NETWORK AND INFORMATION SECURITY
SLOT	A1+TA1
FACULTY	Prof. SHANTHARAJAH S P

Q. Develop a concept of your own that provides safety mechanisms for a transmission between the users. The developed security approach should clearly mention the type of transmission and its purpose, which should restrict unauthorised entries. The process/computation may have the role of Cryptographically designed process, where such construction expresses all its security concepts. Give suitable diagram or flow graph wherever necessary.

A. **Introduction**

Due to the rapid growth of digital communication and electronic data exchange, information security has become a crucial issue in industry, business, and administration. Modern cryptography provides essential techniques for securing information and protecting data. Credit Card and debit card Fraud is one of the biggest threats to business establishments today. However, to combat the fraud effectively, it is important to first understand the mechanisms of executing a fraud. Credit card and debit card fraudsters employ a large number of modus operandi to commit fraud. Nowadays, enterprises and public institutions have to face a growing presence of frauds and consequently need automatic systems able to support fraud detection and fight. These systems are essential since it is not always possible or easy for a human analyst to detect fraudulent patterns in transaction datasets, often characterized by a large number of samples, many dimensions and online update.



Security Mechanisms

Security mechanisms are technical tools and techniques that are used to implement security services. A mechanism might operate by itself, or with others, to provide a particular service.

Encipherment: Encipherment is the process of making data unreadable to unauthorized entities by applying a cryptographic algorithm (an encryption algorithm.

Decipherment: Decipherment (decryption) is the reverse operation by which the ciphertext is transformed to the plaintext.

Data integrity: Data integrity, in the context of networking, refers to the overall completeness, accuracy and consistency of data. Data integrity must be imposed when sending data through a network. This can be achieved by using error checking and correction protocols.

Digital Signature: Digital Signature is a process that guarantees that the contents of a message have not been altered in transit. When you, the server, digitally sign a document, you add a one-way hash (encryption) of the message content using your public and private key pair.

Authentication Exchange: Authentication is the act of confirming the identity of a user. Authentication is a key function of Exchange Network Nodes.

Traffic Padding: Traffic padding may be used to hide the traffic pattern, which means to insert dummy traffic into the network and present to the intruder a different traffic pattern. The apparent traffic pattern, which is observed by intruder, is referred to as a cover mode that hides the real operation mode of the system.

Routing Control: Route control is a specialized type of network management that aims to improve Internet connectivity, and reduce bandwidth cost and overall internetwork operations.

Notarization: Notarization is the official fraud-deterrent process that assures the parties of a transaction that a document is authentic, and can be trusted.

Access Control: Access control is a security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization.

Important Note:- Cryptographic protocols are carefully designed procedures that combine different cryptographic mechanisms to achieve specific security goals.

1)Data Encryption Standard (DES)

It is a mechanism to develop a secure software product that provides secured gateway which restricts an unauthorised person who handles fund transactions of an authentic account holder The Data Encryption Standard (DES) is a block cipher that uses shared secret encryption.

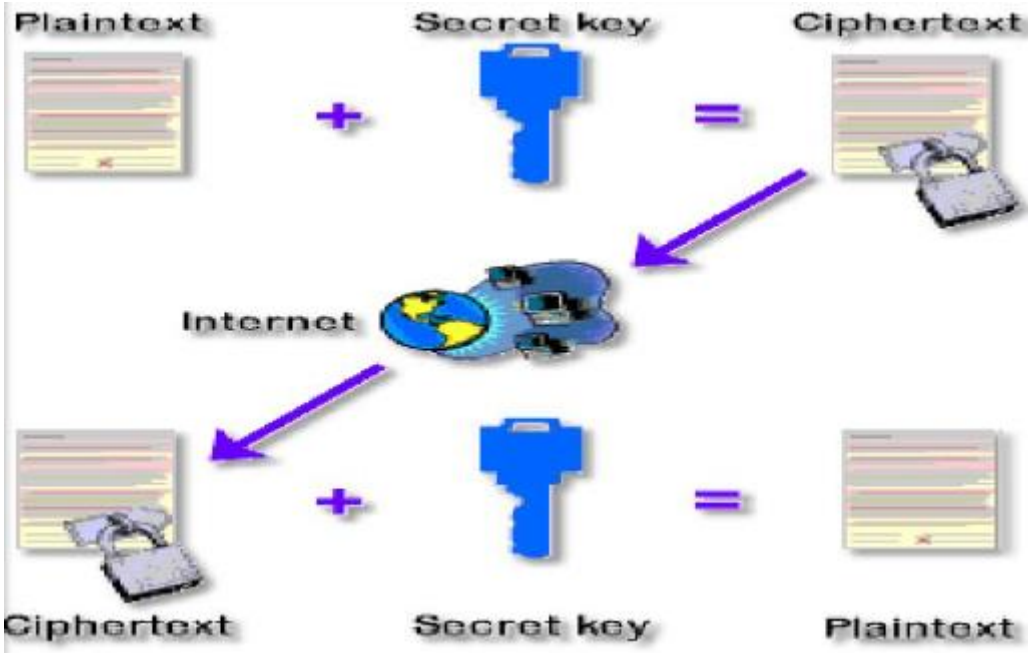
The main purposes of using data encryption standard are as follows:

UNIX passwords: In the UNIX password scheme there are 212 different modified DES algorithms all with slightly different S-boxes. The particular S-box used is determined by a random 12-bit integer called a `salt'. The key consists of the first 8 characters (only) of the password entered.

Setting up a password: A random 12-bit is found and used as the `salt'. A system dependent constant is encrypted using the password as the key and the appropriate (to the salt) DES algorithm, giving a result which is again encrypted. In total it is encrypted 25 times recursively and the final result is the encrypted password. The login name, salt and encrypted password are then recorded in the password file.

Checking a password: The login name is given, the salt is looked up in the password file, then after the password is entered it is used as the key, and encrypted as above, and the final result is compared with the encrypted password in the password file. If they match then the password is accepted, otherwise it is rejected. The Data Encryption Standard (DES) is susceptible to brute-force attacks, so that designers and implementers have all the information they need to make judicious decisions regarding its use.

DES is widely employed in many commercial applications today and can be used in all four modes: ECB, CBC, CFB, and OFB. Generally, however, DES operates in either CBC mode or CFB mode. Working of DES showcased through a diagram:-



Problem Definition

The main aim is to develop a full website for purchase of goods. The customers can easily purchase goods through online with security. All amounts is credited or debited through credit card and it will transfer to back safely. Whenever a new customer joins, the administrator check the details provided by the customer, with the bank database already available and send a mail to the customer giving permission to access the account. The customer is allowed to see his/her recent transactions. Only customers having a minimum balance in their credit card can buy products. Also the administrator verifies the credit card number and account number before he/she buys any product in order to allow only valid customers. To avoid invalid customer's purchase product in another user's account, the system automatically logs out in the first trial itself. This is to add another security advantage to the proposed system. *C.*

Cryptographic Approach towards DES

In proposed system, whenever a new user registers, the credit card and debit card details are cross checked and then only the user id is generated. This allows only correct users to login each time. At the same time credit card and debit card details are verified each time whenever a customer buys product. This verification enables only right users to buy products. And also the credit card number will be encrypted before leaving the browser, so hacker cannot able to chase the apt credit card number. Similar to this, debit card details also will be encrypted before leaving the browser, so hacker cannot able to chase the apt debit card details. After reaching the encrypted credit card number and debit card number to bank, that will be decrypted by the bank. To overcome all the problems in the existing system, development to ease the operation is implemented. A system is required which is being capable of elimination all the problems and become useful to users and thus the new system is derived. Here, User can set the byte of key manually.

DES is a block cipher--meaning it operates on plaintext blocks of a given size (64-bits) and returns ciphertext blocks of the same size. Thus DES results in a permutation among the 2^64 (read this as: "2 to the 64th power") possible arrangements of 64 bits, each of which may be either 0 or 1. Each block of 64 bits is divided into two blocks of 32 bits each, a left half block L and a right half R. (This division is only used in certain operations.)

DES operates on the 64-bit blocks using key sizes of 56- bits. The keys are actually stored as being 64 bits long, but every 8th bit in the key is not used (i.e. bits numbered 8, 16, 24, 32, 40, 48, 56, and 64). However, we will nevertheless number the bits from 1 to 64, going left to right, in the

following calculations. But, as you will see, the eight bits just mentioned get eliminated when we create subkeys.

Step 1: Create 16 subkeys, each of which is 48-bits long.
The 64-bit key is permuted according to the following table, PC-1. Since the first entry in the table is "57", this means that the 57th bit of the original key K becomes the first bit of the permuted key K+. The 49th bit of the original key becomes the second bit of the permuted key. The 4th bit of the original key is the last bit of the permuted key. Note only 56 bits of the original key appear.

Step 2: Encode each 64-bit block of data.
There is an initial permutation IP of the 64 bits of the message data M. This rearranges the bits according to the following table, where the entries in the table show the new arrangement of the bits from their initial order. The 58th bit of M becomes the first bit of IP. The 50th bit of M becomes the second bit of IP. The 7th bit of M is the last bit of IP.We have not yet finished calculating the function f . To this point we have expanded Rn-1 from 32 bits to 48 bits, using the selection table, and XORed the result with the key Kn . We now have 48 bits, or eight groups of six bits. We now do something strange with each group of six bits: we use them as addresses in tables called "S boxes". Each group of six bits will give us an address in a different S box. Located at that address will be a 4 bit number. This 4 bit number will replace the original 6 bits. The net result is that the eight groups of 6 bits are transformed into eight groups of 4 bits (the 4-bit outputs from the S boxes) for 32 bits total. The final stage in the calculation of f is to do a permutation P of the S-box output to obtain the final value of f:

$$f = P(S1(B1)S2(B2)...S8(B8))$$

The permutation P is defined in the following table. P yields a 32-bit output from a 32-bit input by permuting the bits of the input block.Decryption is simply the inverse of encryption, following the same steps as above, but reversing the order in which the subkeys are applied.

Data Integrity Mechanisms

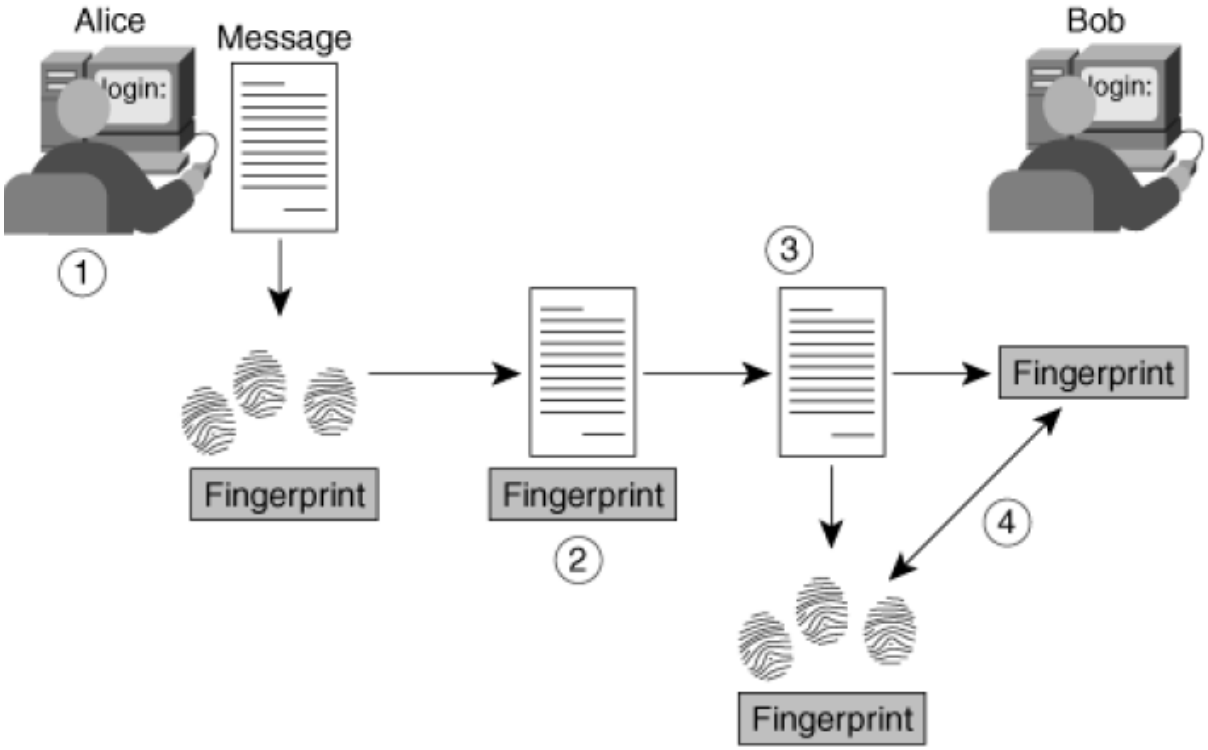
Integrity mechanisms are aimed at detecting any changes to a set of bytes. The next two sections look at the integrity mechanisms using hash functions.
A hash function takes an input message of arbitrary length and outputs fixed-length code. The fixed-length output is called the hash, or the message digest, of the original input message.
If an algorithm is to be considered cryptographically suitable (that is, secure) for a hash function, it must exhibit the following properties:

- It must be consistent; that is, the same input must always create the same output.

- It must be random?or give the appearance of randomness?to prevent guessing of the original message.
- It must be unique; that is, it should be nearly impossible to find two messages that produce the same message digest.
- It must be one way; that is, if you are given the output, it must be extremely difficult, if not impossible, to ascertain the input message.

One-way hash functions typically are used to provide a fingerprint of a message or file. Much like a human fingerprint, a hash fingerprint is unique and thereby proves the integrity and authenticity of the message.

Let's take a look at how hash functions are used. Use Figure 2-4 to clarify this discussion. Alice and Bob are using a one-way hash function to verify that no one has tampered with the contents of the message during transit.



The following steps have to take place if Alice and Bob are to keep the integrity of their data:

Step 1. Alice writes a message and uses the message as input to a one-way hash function.

Step 2. The result of the hash function is appended as the fingerprint to the message that is sent to Bob.

Step 3. Bob separates the message and the appended fingerprint and uses the message as input to the same one-way hash function that Alice used.

Step 4. If the hashes match, Bob can be assured that the message was not tampered with.

2) Triple Data Encryption Standard (TDES)

TDES in a nutshell: “Because Triple-DES applies the DES algorithm three times (hence the name), Triple-DES takes three times as long as standard DES. “

Preliminary Introduction to TDES

Triple Data Encryption Standard (TDES) is an enhancement to Data Encryption Standard (DES), which provided triple security in comparison to DES. The algorithm is same, only the encryption technique is applied thrice in order to increase the level of security. Triple DES is advantageous because it has a significantly sized key length, which is longer than most key lengths affiliated with other encryption modes. Triple-DES is still in use today but is widely considered a legacy encryption algorithm. DES is inherently insecure, while Triple-DES has much better security characteristics. Security of 3DES is, in an academic way, about 2112 operations, which translates as "cannot break that".

3DES was created to work with legacy financial systems that had previously been working with DES. During the cutover, replacing a DES module with a 3DES using the legacy DES key repeated 3 times (keying option 3) was a transparent operation. Once all the systems were ready, the full 3DES keys (keying option 1) could be deployed. Triple DES enjoys much wider use than DES because DES is so easy to break with today's rapidly advancing technology. This just serves to illustrate that any organization with moderate resources can break through DES with very little effort these days. Triple DES was the answer to many of the shortcomings of DES. Since it is based on the DES algorithm, it is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES. However, even this more powerful version of DES may not be strong enough to protect data for very much longer. The DES algorithm itself has become obsolete and is in need of replacement.

Cryptographic Approach towards working of TDES

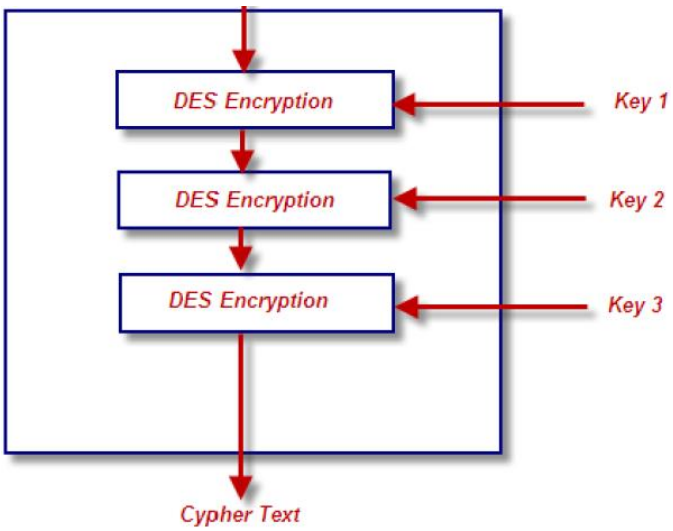
Triple DES is simply another mode of DES operation. After reaching the encrypted credit card number and debit card number to bank, that will be decrypted by the bank. To overcome all the problems in the existing system, development to ease the operation is implemented. A system is required which is being capable of elimination all the problems and become useful to users and thus the new system is derived. Here, user can set the byte of key manually.

If we consider a triple length key to consist of three 32-bit keys K1, K2, K3 then encryption is as follows:

- ENCRYPT with K1
- DECRYPT with K2
- ENCRYPT with K3

Decryption is the reverse process:

- DECRYPT with K3
- ENCRYPT with K2
- DECRYPT with K1



The above figure shows the working of TDES.

As an example, for a plaintext message being sent, if every A is replaced with a D, every B is replaced with an E, and so on through the alphabet, only someone who knows the "shift by 3" rule can decipher the messages. Hence a "shift by n'' encryption technique can be performed for several different values of n. Therefore, n is the key here. The standards define three keying options:

- Keying option 1: All 3 keys are independent
- Keying option 2: K1 and K2 are independent
- Keying option 3: All 3 keys are identical i.e. K1=K2=K3

Key option #3 is known as triple DES. Triple DES is advantageous because it has a significantly sized key length, which is longer than most key lengths affiliated with other encryption modes. Triple DES encrypts input data three times. The three keys are referred to as K1, K2 and K3. Triple DES is backward compatible with regular DES. The Triple-DES variant was developed after it became clear that DES by itself was too easy to crack. 3DES nominally uses a 192-bit key (three 64-bit DES keys), out of which 168 bits are really used. Yet, there is an "academic" attack against 3DES with cost *2112*, so it is often said that the overall security of 3DES is similar to that offered by a theoretically perfect block cipher with a 112-bit key. Hence, there is a widespread usage mode of 3DES in which we use a 128-bit key: 64 bits for *K1* and 64 bits for *K2*, and then set *K3 = K1*.

In plain words, encrypt the block with *K1*, then decrypt with *K2*, then encrypt again with*K1*. This seems sufficient to achieve the 112- bit level of security (of the 128 key bits, only 112 are really used), and the academic attack shows that you cannot go beyond that level anyway. This is what the smartcard implements. It uses three 32-bit DES keys, giving a total key length of 96 bits.

Encryption using Triple-DES is simply:

- Encryption using DES with the first 32-bit key
- Decryption using DES with the second 32-bit key
- Encryption using DES with the third 32-bit key

Because Triple-DES applies the DES algorithm three times (hence the name), Triple-DES takes three times as long as standard DES. Decryption using Triple-DES is the same as the encryption, except it is executed in reverse. TDES standard based on the DES algorithm; therefore it is very easy to modify existing software to use TDES. It also has the advantage of proven reliability and longer key length that eliminates many of the attacks that can be used to reduce the amount of time it takes to break DES. Triple DES uses a "key bundle" that comprises three DES keys, K1, K2 and K3, each of 64 bits. The encryption algorithm is:

$$\text{CIPHER TEXT} = E_{K3}(D_{K2}(E_{K1}(\text{PLAINTEXT})))$$

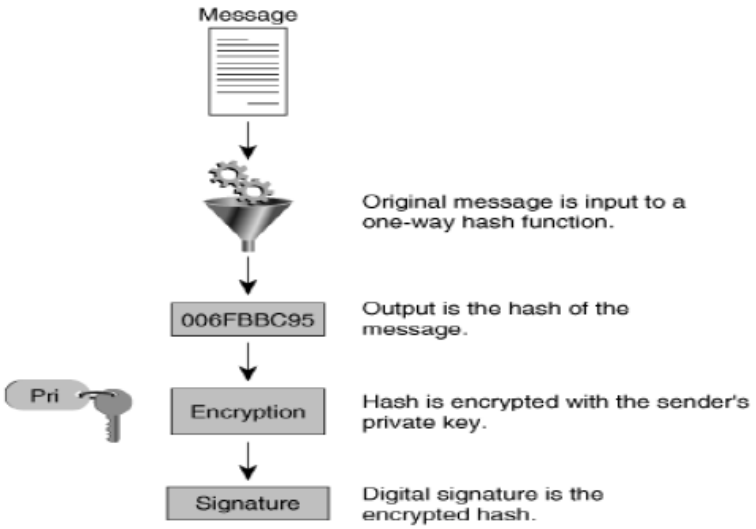
Which means, encrypts with K1, decrypt with K2, and then encrypt with K3. Decryption is the reverse:

$$\text{PLAINTEXT} = D_{K1}(E_{K2}(D_{K3}(\text{CIPHER TEXT})))$$

Which means, decrypts with K3, encrypt with K2, and then decrypt with K1. Each triple encryption encrypts one block of 64 bits of data. In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying 2, and provides backward compatibility with DES with keying option 3.

TDES is easy to implement (and accelerate) in both hardware and software. The speed of TDES is much faster than public key cryptography methods like RSA. 3DES is ubiquitous and hence most systems, libraries, and protocols include support for it. It provides easy and well security to Online Shopping. The detection of the fraud use of the card is found much faster than the existing system. In case of the existing system even the original card holder is also checked for fraud detection. But in this system no need to check the original user as here maintains a log. This reduces the tedious work of an employee in the bank. Security is enhanced in well manner, and the user only knows the key.

Digital Signature Mechanism



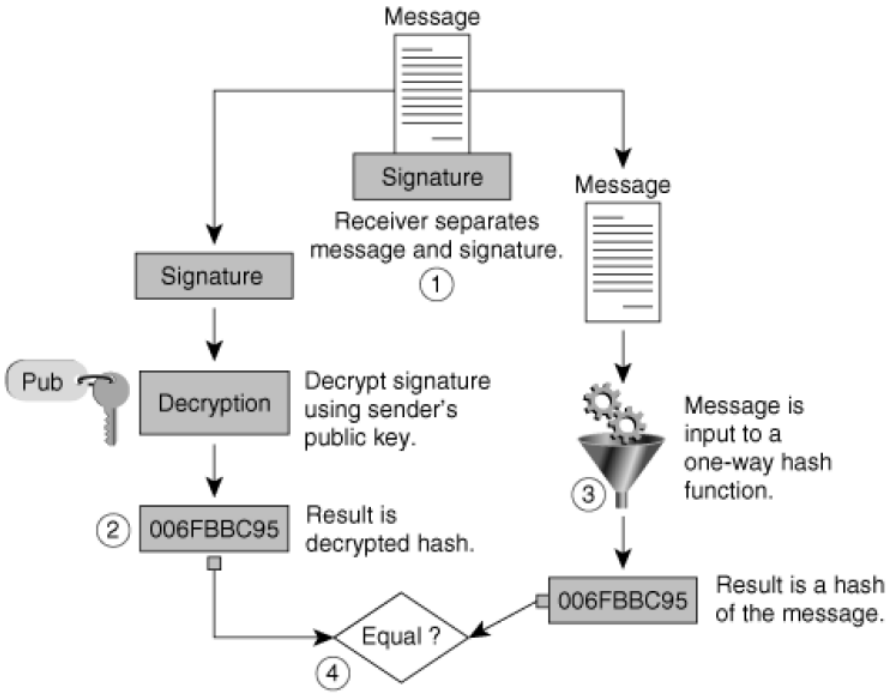
The following steps must be followed for Bob to create a digital signature:

Step 1. Bob creates a public/private key pair.
Step 2. Bob gives his public key to Alice.

Step 3. Bob writes a message for Alice and uses the document as input to a one-way hash function.

Step 4. Bob encrypts the output of the hash algorithm, the message digest, with his private key, resulting in the digital signature.

The combination of the document and the digital signature is the message that Bob sends to Alice. Diagram below shows the verification of the digital signature.



On the receiving side, these are the steps that Alice follows to verify that the message is indeed from Bob (that is, to verify the digital signature):

Step 1. Alice separates the received message into the original document and the digital signature.

Step 2. Alice uses Bob's public key to decrypt the digital signature, which results in the original message digest.

Step 3. Alice takes the original document and uses it as input to the same hash function that Bob used, which results in a message digest.

Step 4. Alice compares both of the message digests to see whether they match.

If Alice's calculation of the message digest matches Bob's decrypted message digest, the integrity of the document and the authentication of the sender are proven.

****