*Cryptography*

*SAHA ANUP (丁子杰) Id: 178801032 Mail :ar.up@outlook.com Software engineering*

## Index: Page

About Cryptography………………………….. 2

Modern Cryptography concerns ……………… 3

Different type of Cryptography …………………….. 3

Symmetric Encryption …………………………….. 5

Brute Force Attack ……………………………... 5

The Data Encryption Standard (DES) ……………… 8

Advanced Encryption Standard ……………………. 10

Public key cryptography ……………………… 11

Rivest-Shamir-Adleman (RSA) Algorithm …………. 14

Message Authentication ……………………….. 15

Authentication Using Symmetric Encryption ……….. 16

Message Authentication Code …………………….. 17

Hash Function …………………… 19

One-way Hash Function ……………………… 22

Secure Hash Function ………………………. 25

## 

## Cryptography

The essential technology underlying virtually all automated network and computer security applications are cryptography. To simply say cryptography is associated with the process of converting ordinary plain text into unintelligible text and vice-versa .It is a method of string and transmitting data in a particular form so that only those for whom it is intended can read and process it .Cryptography not only protects data from theft or alteration ,but can also be used for user authentication.

*K.1 More about Cryptography:*

Earlier cryptography was effectively synonymous with encryption but nowadays cryptography is mainly based on mathematical theory and computer science practice. And it is practice and study about hiding information, it is sometimes called code but that’s not it’s real name .We can call it science cause it is used to try to keep information secret and safe .Cryptography is used in bank, computer password and electrical engineering and shopping on the internet .When a message is sent using cryptography ,it is changed or encrypted before it is sent .This method called a code or more preciously Cipher .The text which is changed is called cipher text .And after change the text it makes the reader read the message hard to read and if someone want to read the massage it must change it back to previous form or decrypt the text. How to change it back is a secret it’s only known by the person who send the message and the one who get the message should know the secret way to change or decrypt the message for that other people unable to decrypt the message and read the message .And discover this secret is called cracking or cryptanalysis and also called code breaking .Different type of cryptography can be easier or harder to use and can hide the secret message better or worse. Ciphers use a Key which is a secret that hides the secret message .The cryptographic method needn’t be secret ,various people can use the same method but different keys so they cannot read the others messages .Ciphers that allow billions of keys are cracked by more complex ways .During the 20th century computers becomes the principle tool of cryptography.

*K.2: Modern cryptography concerns with:*

1.Confidentiality – Information cannot be understood by anyone cause

2.Integrity – Information cannot be changed.

3.Non-rejection – Sender cannot deny his/her intentions in the transmission of the information at the last stage.

4.Authentication – Sender and receiver can confirm each-others.

*K.3: Different kind of cryptography:*

There are three types of cryptography techniques used in general.

1. Symmetric-key cryptography
2. Public-key cryptography
3. Hash functions

*K.3.1 Symmetric Cryptography*

Symmetric encryption was the only type of encryption in use prior to the introduction of public-key encryption in the late 1970s. Symmetric encryption has been used for secret communication by countless individuals and groups, from Julius Caesar to the German U-boat force to present-day diplomatic, military, and commercial users. It remains by far the more widely used of the two types of encryption. A symmetric encryption scheme has five ingredients.

*3.1.1 Plaintext:* This is the original message or data that is fed into the algorithm as input. Plain text is the most portable format because it is supported by nearly every application on every machine. It’s quite limited, however because it can’t contain any formatting commands. Plaintext refers to any message that is not encrypted and we also call it clear text.

*3.1.2 Encryption algorithm:* The encryption algorithm performs various replacements and transformations on the plaintext.

*3.1.3 Secret key:* The secret key is also input to the encryption algorithm. The exact replacement and transformations performed by the algorithm depend on the key. And secret key use to encrypt and decrypt the message. But in secret key there is major problem that is the logistical issue of how to get the key from one party to another party without allowing access to an attacker.

*3.1.4 Ciphertext:* This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands is incomprehensible.

*3.1.5 Decryption algorithm:* It is necessary the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produce the original plaintext.

*k.3.1.2 Requirement Secure use of Symmetric encryption*

There are two requirements for secure use of symmetric encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.

2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

*K.3.1.3 Different kinds of Symmetric attack*

There are two general approaches to attacking a symmetric attacks scheme.

*Cryptanalysis:*

The first attack is known as cryptanalysis. Cryptanalysis is the study of analyzing information systems in order to study the hidden aspects of the systems. Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown.

In addition to mathematical analysis of cryptographic algorithms, cryptanalysis includes the study of side channel attacks that do not target weaknesses in the cryptographic algorithms themselves, but instead exploit weaknesses in their implementation. Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. If the attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised on.

*Brute-force attack:*

The second method, known as the brute-force attack. The brute force attack is a trial and error method used by application programs to decode encrypted data such as passwords, data encryption standard keys or personal identification numbers such like pin. As a matter of fact, this attack can be used against any type of encryption. And brute force attack or Brute Force login attack is one of the most common attacks conducted against Web application .The brute force attacks are not executed by individuals ,but bots which can test millions of login combinations in a short amount of time .This method is likely to start at one digit passwords ,then moving on to two digit passwords and so on ,until the password is cracked .The brute force attack is most commonly used by hacker to crack encrypted data, but it is slow method .It is also used by security analysis to test company network security .

There are three groups of options which hackers can modify in order to crack the account with the brute force attack:

**Brute force char-set**: This group of options allows hackers to select the define characters set which was selected before it contains Lower case and Uppercase, digits, Special symbols. All printable ASCII or they can define custom character set and the brute force attack will guess a password by trying all probable variations by given character set.

**Password length and position:** This group of options allows hackers to define the maximum and minimum length of the password that will be checked. The number of possible combinations grows as the length of the password increase and so does the length of time of the attack.

**Distributed attack:** If an attack is distributed, that means that more than one computer is participating in the attack. This group of options allows hackers to define how many computers are participating in the attack and select the same settings for all computers so the attack could be more effective. There are three types of different brute force attack.

**Dictionary Attack:** This attack will use words from a dictionary and literature, or list of common passwords to guess your password. This method can be very effective because a lot of people use weak and common passwords like admin or password.

**Search Attack:** This kind of the attack can be very slow because it will try to cover all possible combinations of a given character set and a password length range. The complexity of this type of attack is very high due to huge key space available for the proposed system.

**Rule-based search Attack:** By creating a certain rule for password generation, rule -based search attack enables an increase of combination space coverage without slowing down the process too much.

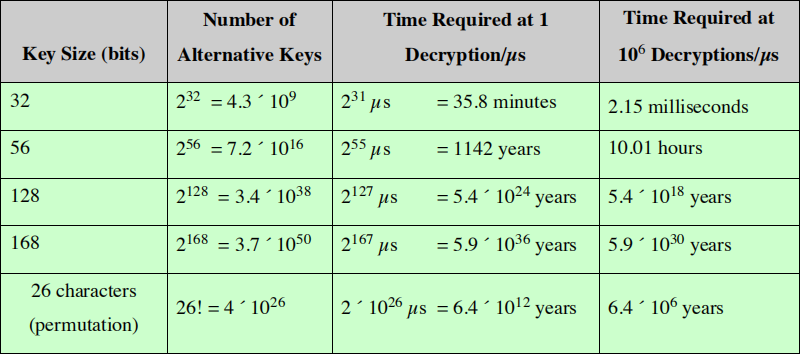
For example

If we have an alphanumeric 8-characters, there are 26 alphabets in English. Double them for both UPPER and lower cases and counts settles on 26+26 =52. Then we add the numeric digits 52+10 = 62

So, we have 62 characters in total. For 8-character-password, it will be 628 which will make 2.1834011×1014 possible combinations.

If we attempt 218 trillion combinations at one try per second, it would take 218 trillion seconds or 3.6 trillion minutes. More preciously it would take around 7 million year to crack the password with the final combination. Surely it can take less, but 7 million years is the maximum time limit to crack an alphanumeric 8-character password.

Below the table is “Average Time Required for Exhaustive Key Search”



***Preventing Brute force attack:*** Brute-force attacks may not be complicated to understand, but it can be difficult to protect against and sometimes this attack can be very messy. These attack can be used against any type of encryption and access the personal data and can be whole server .These speed at which someone might brute-force encryption depends on hardware ,it increase as technology becomes faster and capable of doing more calculations per seconds .Actually ,there is no way to protect completely ,but there are few steps that can defends against this attacks:

**Install Security:** Plugininstall a highly trusted and rated security plugin to help protect against brute force attacks. A good security plugin will make it easy to implement the following recommendations.

**Use a strong password:** Try to guess hard type of passwords. It should identify avoid dictionary words or common series of numbers in passwords. Word-press password Strength Detector can come in handy when making the password strong .Also changing password periodically can help the website stay safe

**Limit the number of log in attempts**: Limit the number of failed login attempts the site will allow before locking out the user. If user enter the wrong password more than specific numbers of times, they will be locked out. Also the admin should extend the period between the two possible log in after a wrong password entered. The more often they enter wrong password, the longer they have to wait to try again. This way a high performance computer can be solved down in spite of the large number of calculation it could possibly do.

**Ban specified IP address**: With a good plugin installed and from the word press panel you will be able to see how many time your website comes under attack .If you notice that the same IP address is trying to access multiple times to your site ,you should definitely ban or block that IP address.

**Update and Upgrade website**: It is crucial to always update your site to the latest word press version when you see the notification in your admin section. This also applies all installed plugins and themes. A few quick steps to update your version will save you from potentially costly problems down the road.

*The Data Encryption Standard (DES)*

**Data Encryption Standard: Data encryption is an early data encryption standard endorsed by the US National Bureau of Standards (now National institute of standards and Technology). It was phased out at the start of the 21st century by a more secure encryption standard, known as the Advanced Encryption standard (AES), which was better suited for the securing commercial transactions over the internet. In 1973 the NBS issued a public request for proposals for a crypt-o-algorithm to be considered for a new cryptography standard .No viable submissions were received .A second request was issued in 1974 ,and IBM submitted the patented Lucifer algorithm that had been devised by one of the companies research , a few year years earlier .The Lucifer algorithm was evaluated in secret consultations between the NBS and the US National Security Agency (NSA).After some modifications to the internet functions and a shortening of the code key size from 112 bits to 56 bits the full details of the algorithm that was to become data encryption standard were publish in the federal register in 1975 .Following almost two years of the public evaluation and comment ,the standard itself was adopted at the end of 1976 and published at the beginning of 1977 ..As a consequence of certification of the standard by the NBS and its commitment to evaluate and certify implementations ,it was mandated that the DES be used in unclassified US government applications for the protection of binary coded data during transmission and storage in computer systems and networks and on a case by case basis for the protection of classified information. The use of DES algorithm was mandatory for all financial transaction of the U.S government involving electronic fund transfer, including those conducted by members bank of the Federal Reserve System. Subsequent adoption of the DES by standard organizations worldwide caused the DES to become a international standard for business and commercial data security as well. The DES is a product block cipher in which 16 iterations or rounds of substitution and transposition process are cascaded. The block size is 64 bits. The key, which controls the transformation also consists of 64 bits. The key which controls the transformations also consists of 64 bits ,however only 56 of these cab be chosen by the user and are actually key bits .The remaining 8 are parity check bits and hence totally redundant .The figure is a functional schematic of the sequence of events that occurs in one round of the DES encryption transformation .At each intermediate stage of the transformation process the cipher output .The security of the DES is no greater than its work factor the brute force effort required to search 256 keys .That is a search for a needle in a haystack of 72 quadrillion straws .In 1977 that was considered an impossible task .In 1999 a special purpose DES search engine combined with 100000 personal computer on the internet to find a DES challenge key in 22 hours .An earlier challenge key was found by distributed computing over the internet in 39 days and by the special purpose search engine alone in 3 days .For some time its apparent that the DES ,though never broken in the usual cryptography and its sense , was no longer secure .A way was devised that effectively gave the DES a 112 bit key ironically the key size of the Lucifer algorithm originally proposed by IBM in1974 .This known as ‘Triple DES .Thus equipment that only implemented the older single DES .Banking standard adopted this scheme for security.**

Cryptography has traditionally been a secretive science so much so that it was only at the end of the 20th century that the principles on which the cryptography and its analysis of the Japanese and German cipher machines of World War 2 were based were declassified and released. What was different about the DES was that it was a totally public cryptography algorithm. Every detail of its operations—enough to permit anyone who wished to program it on a microcomputer was widely available in published form and on the Internet. The paradoxical result was that what was generally conceded to have been one of the best cryptography systems in the history of the cryptography was also the least secret.

Advanced Encryption Standard

The Advanced Encryption Standard or AES is a symmetric block cipher chosen by U.S government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data .The NIST started development of AES in 1997 when it announced the need for a successor algorithm for DES which was starting to become vulnerable to brute force attack .The new advanced encryption algorithm would be unclassified and had to be capable of protecting sensitive government well .It was intended to be easy to implement in hardware and software as well as restricted environment and offer good defense against various attack techniques.

Encryption Standard (AES), which should have a security strength equal to

or better than 3DES and significantly improved efficiency.

AES Feature

The selection process for this new symmetric key algorithm was fully open to public security and comment this ensure a through, transparent analysis of the designs submitted.

NIST specified the new advanced encryption standard algorithm must be a block cipher capable of handling 128bits blocks, using key size at 128 ,192 and 256 bits; others criteria for being chosen as the next advanced encryption standard algorithm included.

***Security***: Competing algorithm were to be judged on their ability to resist attack, as compared to other submitted ciphers, though security strength was to be considered the most important factor in the competition.

***Cost***: Intended to be released under a global nonexclusive and royalty free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.

***Implementation***: Algorithm and implementation characteristics to be evaluated included the flexibility of the algorithm suitability of the algorithm to be implemented in hardware or software overall relative simplicity of implementation.

*K.3.2 Public key cryptography*

Public-key encryption, first publicly proposed by DIFFIE and HELLMAN in 1976, is the first truly revolutionary advance in encryption in literally thousands of years. The most commonly used implementations of public key cryptography are based on algorithms represented by RIVEST SHAMIR ADELMAN Data security. For one thing, public-key algorithms are based on mathematical functions rather than on simple operations on bit patterns. Public Key cryptography involves a pair of keys known as a public key and a private key which is associated with an entity that needs to authenticate its identity electronically or to sign or encrypt with the public key can decryption only with the corresponding private key. More important, public-key cryptography is asymmetric, involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key. The use of two keys has profound consequences in the areas of confidentiality, key distribution, and authentication. Before proceeding, we should first mention several common misconceptions concerning public-key encryption. One is that public-key encryption is more secure from cryptanalysis than symmetric encryption. In fact, the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher. There is nothing in principle about either symmetric or public-key encryption that makes one superior to another from the point of view of resisting cryptanalysis. A second misconception is that public-key encryption is a general-purpose technique that has made symmetric encryption obsolete. On the contrary, because of the computational overhead of current public-key encryption schemes, there seems no foreseeable likelihood that symmetric encryption will be abandoned. Finally, there is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for symmetric encryption. In fact, some form of protocol is needed, often involving a central agent, and the procedures involved are no simpler or any more efficient than those required for symmetric encryption.

RSA public key can be any size. Typical sizes today are 1024 to 2048 bits. Public key cryptography enables the following:

Encryption and decryption, which allow two communicating parties to disguise data that they send to each other The sender encrypts, or scrambles the data before sending it. The receiver decryption the data after receiving it, the encrypted data is not understood by an intruder.

Non-repudiation which prevents the sender of the data from clamming, at the later date that the data was never sent and also it prevents from data being altered.

Here is a simple example how the encryption works

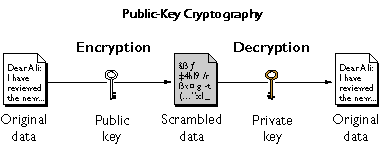


Figure: Public-key Cryptography

This figure shows us how we can freely distribute the public key so that only the selected user can read the data that was encrypted with the public key .In general to send encrypted data to someone ,we must encrypted the data with that person’s public key ,and the person receiving the data decrypt it with the corresponding private key .If we compare symmetric key encryption with public key encryption we will find that public key encryption require more calculation than symmetric key encryption .Therefore public key encryption is not always appropriate for large amount data .However ,it is possible to use public key encryption to send s symmetric key which we can use to encrypt additional data .The reverse of what is shown in the figure also works that the advantage. Data encrypted with private key can also decrypt only with public key .However this is not a desirable way to encrypt private data because it means that anyone with your public key ,which is by definition published could decrypt the sensitive data .Despite this private key encryption is useful because it enables to use private key to sign data with digital signature ,anyone with same public key can be assured that only you sent data .This is an important requirement for electronic commerce and other commercial application of cryptography.

More preciously public-key encryption scheme has these ingredients

• ***Plain-text***: This is the readable message or data that is fed into the algorithm as input. We generally text the message that delivered to others person and readable.

• ***Encryption algorithm***: The encryption algorithm performs various

transformations on the plain-text. It can transfer from character to binary and others so that others can’t stole the text or message.

• ***Public and private key***: This is a pair of keys that have been selected

so that if one is used for encryption, the other is used for decryption.

The exact transformations performed by the encryption algorithm

depend on the public or private key that is provided as input.

• ***Cipher-text***: This is the scrambled message produced as output. It

depends on the plain-text and the key. For a given message, two

different keys will produce two different cipher-texts.

• ***Decryption algorithm***: This algorithm accepts the cipher-text and the With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user protects his or her private key, incoming communication is secure. At any time, a user can change the private key and publish the companion public key to replace the old public key. The key used in symmetric encryption is typically referred to as a secret key. The two keys used for public-key encryption are referred to as the public key and the private key. Invariably, the private key is kept secret, but it is referred to as a private key rather than a secret key to avoid confusion with symmetric encryption.

*K.3.2.1 Rivest-Shamir-Adleman (RSA) Algorithm*

RSA is one of the first public key cryptography system that widely used for secure data transmission. It was first developed in 1977 by RON RIVEST, ADI SHAMIR and LEN ADLEMAN at MIT. The RSA scheme has since that time regained supreme as the only widely accepted and implemented approach to public key encryption. In such a system the encryption key is public and it is different from the decryption key which is kept secret. RSA involves the use of three individual functions. The first functions of RSA is to generate both a public key as well as private key .The public key is used solely for encryption and can be handed out to anyone .The private key ,however used for decryption and it is specially generated to work only with the public key that it was made for .The user kept the private key to himself/herself and uses it much like a password to access the contents of an encrypted file .The second function of RSA is perform the encryption that secures the file and web pages .RSA does by choosing two random prime numbers and multiplying them in a way that produces a specific result .RSA is a cipher in which the plain-text and cipher-text are integers between 0 and n – 1 for some n. Encryption involves modular arithmetic. The strength of the algorithm is based on the difficulty of factoring numbers into their prime factors. The third function of RSA is to perform decryption for encrypted file by using an equation that results in the exact opposite pf the original equation, therefore reveling the two original prime numbers and allowing the user to access the encrypted information.

***K.3.2.2 Advantages***

RSA is used for two important reasons: its ability to allow users to electronically sign their domains and other digital property and its capability of providing encryption for websites, servers, and files in general. RSA can be used in email, online commerce, and many other systems that require privacy for their users. Additionally, RSA is rather complex and is almost impossible to decrypt without a private key.

*K.4.1 Message Authentication*

In cryptography, a message authentication code (MAC) sometimes known as a tag, is a short piece of information used to authenticate a message -in the others words, to confirm that the message came from the stated sander and has not been changed. The MAC value protects both a message’s data integrity as well as its authenticity by allowing verifies to detect any changes to the message content .Encryption protects against passive attack (eavesdropping). A different requirement is to protect against active attack (falsification of data and transactions). Protection against such attacks is known as message authentication. A message, file, document, or other collection of data is said to be authentic when it is genuine and came from its alleged source. Message authentication is a procedure that allows communicating parties to verify that received messages are authentic. The two important aspects are to verify that the contents of the message have not been altered and that the source is authentic. We may also wish to verify a message's timeliness (it has not been artificially delayed and replayed) and sequence relative to other messages flowing between two parties. The abbreviation MAC can also be used for describing algorithms that can create an authentication code and verify its correctness.

Informally message authentication code consists of three algorithms.

* + - 1. A key generation algorithm selects a key from the key space at random.
      2. A signing algorithm efficiently returns a tag given the key and the message
      3. A verify algorithm efficiently verifies the authenticity of the message given the key and the tag. That is, return accepted when the message and tag are not tamped with or forged, and otherwise return rejected.

For a secure unforgeable message authentication code, it should be computationally infeasible to compute a valid tag of the given message without knowledge of the key, even if for the worst case, we assume the adversary can forge the tag of any message except the given one.

*K.4.2 Authentication Using Symmetric Encryption*

Encryption with a secret symmetric key does not prove authenticity unless we use an authenticated encryption mode of operation such as GCM. Authenticated encryption algorithms generate a message authentication code in addition to encrypting the message, and if the shared key is properly secured this can be used to prove the authenticity and integrity of the message but non-repudiation. Regular symmetric key encryption with a shared key does not prove the integrity nor authenticity of the message because nothing prevents an attacker from generating a random message which the receiver will then decrypt and accept .Though the attacker doesn’t know what the decrypt message will look like there are many situations in which having the receiver accept a randomly generated message may be advantageous to an attacker .Only someone with access to the symmetric key can generate a MAC of message , so using authenticated encryption with a properly secured symmetric key is enough to authenticate the message. MACs don’t provide non-repudiation because in symmetric cryptography the receiver also has the same secret key, so there is no way the receiver to prove that the shared key to be receiver signed the message. An emphasized previously using a shared key for authentication requires the shared key to be properly secured. Specifically, if the receiver generates the symmetric key and sends it to the sender asymmetrically encrypted with the receiver’s public key then this symmetric key can’t be used to prove the authenticity of messages sent by the sender since nothing in the process proves the identity of the sender. Anyone with access to the receiver’s public key can generate a random symmetric key and encrypt it with the receiver’s public key.

A symmetric key can be used to authenticate a message only if the scheme for generating the shared symmetric key includes two-way authentication. Schemes which include two-way authentication to generate shared key for singing and encrypting message are commonly known as a Secure Authenticated Channel or SAC.

*K.4.3 Message Authentication without Message Encryption*

In this section, we examine several approaches to message authentication that do not rely on message encryption. In all of these approaches, an authentication tag is generated and appended to each message for transmission. The message itself is not encrypted and can be read at the destination independent of the authentication function at the destination. Because the approaches discussed in this section do not encrypt the message, message confidentiality is not provided. Because symmetric encryption will provide authentication, and because it is widely used with readily available products, why not simply use such an approach, which provides both confidentiality and authentication? Here are three situations in which message authentication without confidentiality is preferable:

1. There are a number of applications in which the same message is broadcast to a number of destinations. An example is the notification to users that the network is now unavailable or an alarm signal in a control center. It is cheaper and more reliable to have only one destination responsible for monitoring authenticity. Thus, the message must be broadcast in plaintext with an associated message authentication tag. The responsible system performs authentication. If a violation occurs, the other destination systems are alerted by a general alarm.

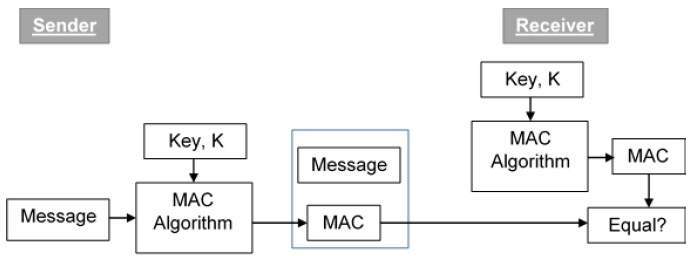
2. Another possible scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages. Authentication is carried out on a selective basis, with messages chosen at random for checking.

3. Authentication of a computer program in plaintext is an attractive service. The computer program can be executed without having to decrypt it every time, which would be wasteful of processor resources. However, if a message authentication tag were attached to the program, it could be checked whenever assurance is required of the integrity of the program. Thus, there is a place for both authentication and encryption in meeting security requirements.

*K.4.4 Message Authentication Code*

Mac algorithm is a symmetric key cryptography technique to provide message authentication. For establishing MAC process, the sender and receiver share a symmetric key k (for example).

Essentially, a mac is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication. The process of using mac for authentication is depicted in the following illustration



Let us try to understand the entire process in details ---

* The sender uses some publicly known MAC algorithm, inputs the message and the secret key k and produces a mac value.
* Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and mac uses secret during the compression.
* The sender forwards the message along with the MAC. Here, we assume that the message is sent in the clear, as we are concerned of providing message origin authentication, not confidentiality. If confidentiality is required then the message needs encryption.
* On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
* The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.
* If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified. As a bottom-line, a receiver safely assumes that the message is not the genuine.

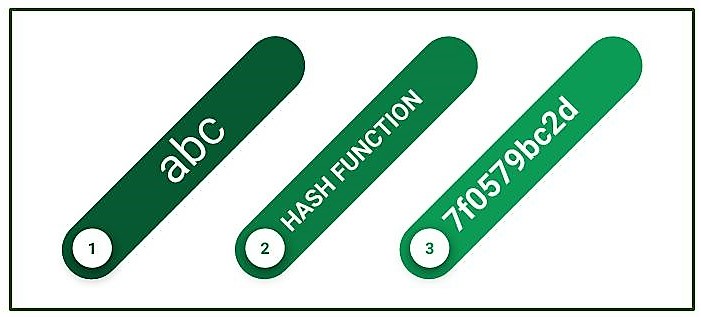
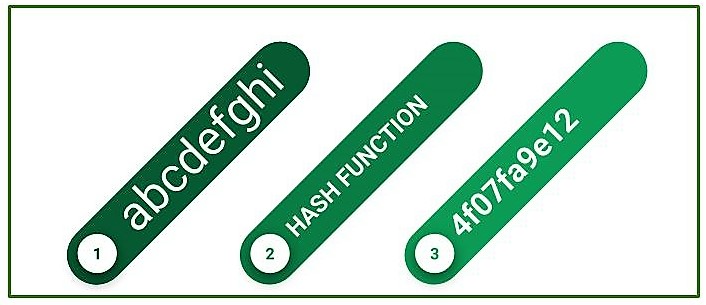
## *K.4.5 Limitations of MAC*

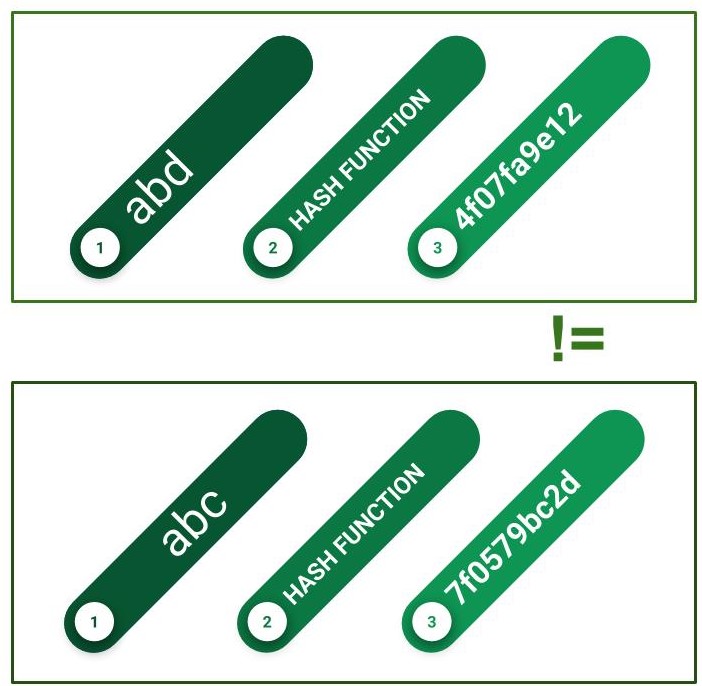
There are two major limitations of MAC, both due to its symmetric nature of operation −

* **Establishment of Shared Secret.**
  + It can provide message authentication among pre-decided legitimate users who have shared key.
  + This requires establishment of shared secret prior to use of MAC.
* **Inability to Provide Non-Repudiation**
  + Non-repudiation is the assurance that a message originator cannot deny any previously sent messages and commitments or actions.
  + MAC technique does not provide a non-repudiation service. If the sender and receiver get involved in a dispute over message origination, MACs cannot provide a proof that a message was indeed sent by the sender.
  + Though no third party can compute the MAC, still sender could deny having sent the message and claim that the receiver forged it, as it is impossible to determine which of the two parties computed the MAC.

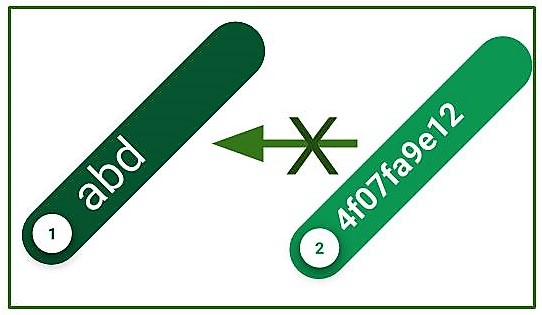
Both these limitations can be overcome by using the public key based digital signatures discussed in following section.

*K.4.6 Hash Function*

Hash Function is a function which has a huge role in making a System Secure as it converts normal data given to it as an irregular value of fixed length. We can imagine it to be a Shaker in our homes.  
When we put data into this function it outputs an irregular value. The Irregular value it outputs is known as **“Hash Value”**. Hash Values are simply number but are often written in Hexadecimal. Computers manage values as Binary. Hash value is also a data and are often managed in Binary.  
  
Hash function is basically performing some calculations in the computer. Data values that are its output is of fixed length. Length always varies according to the hash function. Value doesn’t vary even if there is a large or small value.  


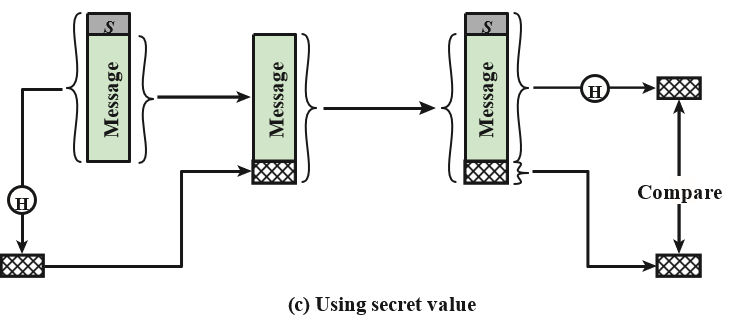
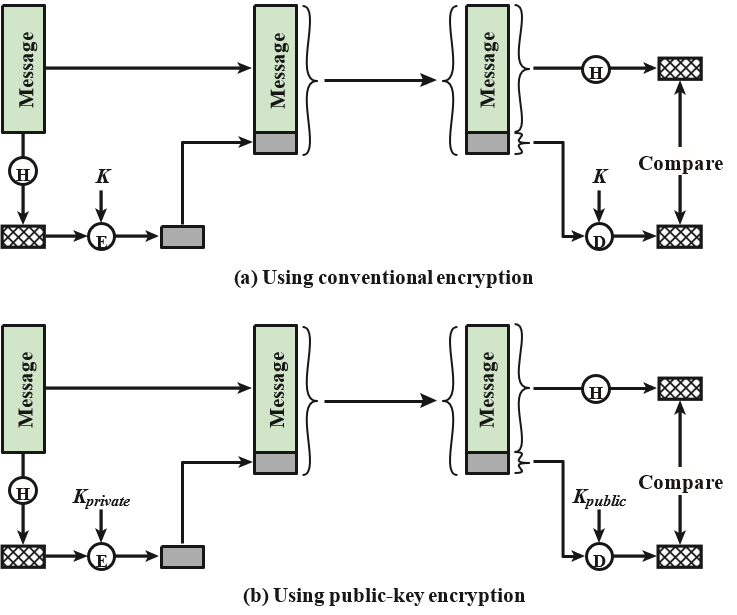
If given same input, two hash functions will invariable produce the same output. Even if input data entered differs by a single bit, huge change in their output values. Even if input data entered differs huge, there is a very minimal chance that the hash values produced will be identical. If they are equal it is known as **“Hash Collision”.**  


Converting Hash Codes to its original value is impossible task to perform. This is Main difference between Encryption as Hash Function.



*K.4.7 One-Way Hash Function*

A variation on the message authentication code that has received much attention is the one-way hash function. As with the message authentication code, a hash function accepts a variable-size message M as input and produces a fixed-size message digest H(M) as output. Unlike the MAC, a hash function does not also take a secret key as input. To authenticate a message, the message digest is sent with the message in such a way that the message digest is authentic. Figure illustrates three ways in which the message can be authenticated. The message digest can be encrypted using symmetric encryption (part a); if it is assumed that only the sender and receiver share the encryption key, then authenticity is assured. The message digest can also be encrypted using public-key encryption (part b).



The public-key approach has two advantages: it provides a digital signature as well as message authentication, and it does not require the distribution of keys to communicating parties. These two approaches have an advantage over approaches that encrypt the entire message in that less computation is required. Nevertheless, there has been interest in developing a technique that avoids encryption altogether. Several reasons for this interest:

• Encryption software is somewhat slow. Even though the amount of data to be encrypted per message is small, there may be a steady stream of messages into and out of a system.

• Encryption hardware costs are nonnegligible. Low-cost chip implementations of DES are available, but the cost adds up if all nodes in a network must have this capability.

• Encryption hardware is optimized toward large data sizes. For small blocks of data, a high proportion of the time is spent in initialization/invocation overhead.

• Encryption algorithms may be covered by patents and must be licensed, adding a cost.

• Encryption algorithms may be subject to export control

Figure shows a technique that uses a hash function but no encryption for message authentication. This technique assumes that two communicating parties, say A and B, share a common secret value SAB. When A has a message to send to B, it calculates the hash function over the concatenation of the secret value and the message: MDM = H(SAB||M). It then sends [M||MDM] to B. Because B possesses SAB, it can recompute H(SAB||M) and verify MDM. Because the secret value itself is not sent, it is not possible for an attacker to modify an intercepted message. As long as the secret value remains secret, it is also not possible for an attacker to generate a false message. This third technique, using a shared secret value, is the one adopted for IP security; it has also been specified for SNMPv3.

## *K.5.1 SECURE HASH FUNCTIONS*

An essential element of many security services and applications is a secure hash function. A hash function accepts a variable-size message *M* as input and produces a fixed-size tag H(*M*), sometimes called a message digest, as output. For a digital signature, a hash code is generated for a message, encrypted with the sender's private key, and sent with the message. The receiver computes a new hash code for the incoming message, decrypts the hash code with the sender's public key and compares. If the message has been altered in transit, there will be a mismatch.

To be useful for security applications, a hash function H must have the following properties:

1. H can be applied to a block of data of any size.
2. H produces a fixed-length output.
3. H(*x*) is relatively easy to compute for any given *x*, making both hardware and software implementations practical.
4. For any given value *h*, it is computationally infeasible to find *x* such that H(*x*) = *h*. This is sometimes referred to in the literature as the **one-way** **property**.
5. For any given block *x*, it is computationally infeasible to find *y* " *x* such that H(*y*) = H(*x*). This is sometimes referred to as **weak collision resistance**.
6. It is computationally infeasible to find any pair (*x*, *y*) such that H(*x*) = H(*y*). This is sometimes referred to as **strong collision resistance**.

In recent years, the most widely used hash function has been the Secure Hash Algorithm (SHA). SHA was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993. When weaknesses were discovered in SHA, a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1. SHA-1 produces a hash value of 160 bits. In 2002, NIST produced a revised version of the standard, FIPS 180-2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512. These new versions have the same underlying structure and use the same types of modular arithmetic and logical binary operations as SHA-1. In 2005, NIST announced the intention to phase out approval of SHA-1 and move to a reliance on the other SHA versions by 2010. Researchers have demonstrated that SHA-1 is far weaker than its 160-bit hash length suggests, necessitating the move to the newer versions of SHA.