

0010 0110 0000 0011 0100 0000 0100
0001 0100 0100 0011 0110 0000 0011
0011 0011 0000 0000 0101 0011 0000
000 0101 0101 0011 0001 0000 0011
001 0000 0101 0111 0011 0100 000
00 0111 0000 0011 0100 0001 000
11 0011 0000 0000 0011 0000 01
10 0101 0001 0011 0001 0000 00

CRYPTOGRAPHY

0110 0000 0011 0100 000
100 0100 0011 0110 00
011 0000 0000 0101 00
01 0101 0011 0001 0
00 0101 0111 0011 0
11 0000 0011 0100
4 0000 0000 0011

-JANARDAN
2020111005

CONTENTS

1)History

1.1)Cryptography from 1800 to WW1

1.2)World War II cryptography

1.2.a)Germany

1.2.b)Japan

1.2.c)Allies

1.2.c.i)Typex

1.2.c.ii)Sigaba

1.2.c.iii)M-209

1.2.c.iv)VIC Cipher

1.2.c.v)Enigma

2)Classical Cryptography

2.1)Symmetric-key cryptography

2.2)Public-key cryptography

3)Quantum Cryptography:

3.1)Introduction

3.2)History

3.3)Mechanism

3.4)Application

3.5)Advantages

3.6)Limitations

3.7)Future

4)Cryptography against Phishing

4.1)Introduction

4.2)Automated Challenge Response Method

4.3)Blacklist

4.4)Phish-Secure

4.5)LARX

4.6)Visual Cryptography

4.6.a)Registration Phase

4.2.b)Login Phase

5)GENETIC ALGORITHM AND PSEUDO RANDOM SEQUENCE GENERATING FUNCTIONS

5.1)Operators of GAs

5.1.a)Crossover

5.1.b)Mutation

5.1.c)Pseudorandom number generator:

5.1.d)The Encryption process

5.2)Proposed method

6)Symmetric Key Cryptography

6.1)Data Encryption Standard

6.2)Triple DES

6.3)Advanced Encryption Standard

6.4)RC2

6.5)RC4

6.6)RC5

6.7)RC6

6.8)Blowfish

6.9)Twofish

6.10)Comparison

7)Cryptography during Data Sharing and Accessing over Cloud

7.1)Abstract

7.2)Introduction

7.3)Challenges

7.4)Types of Attacks

7.4.a)Flooding Attacks

7.4.b)Law Enforcement Requests

7.4.c)Data Stealing Attacks

7.4.d)Denial of Service Attacks

7.4.e)XML Signature Wrapping Attacks

7.4.f)Cross site scripting attacks

7.5)Data Sharing and Accessing in the Cloud

7.5.a)Requirements

7.6)Need for Key Management in Cloud

7.6.a)Secure key stores

- 7.6.b)Access to key stores
 - 7.6.c)Key backup and recoverability
- 7.7)Identity and Access Management
 - 7.7.a)Role Based Access Control (RBAC)
 - 7.7.b)User Based Access Control (UBAC)
 - 7.7.c)Attribute Based Access Control (ABAC)
- 7.8)Proposed scheme
 - 7.8.a)Assumptions
 - 7.8.b)Mathematical Background
 - 7.8.c)Formats of Access policies
- 7.9)System Module for Encrypted Data Sharing
 - 7.9.a)Setup Phase
 - 7.9.b)Key Generation Phase
 - 7.9.c)Encryption Phase
 - 7.9.d)Extract Phase
 - 7.9.e)Decryption phase
- 7.10)A New Architecture
 - 7.10.i)Elliptic Curve Cryptography
 - 7.10.ii)Diffie Hellmann Key Exchange
 - 7.10.a)Establishment of connection
 - 7.10.b)Account creation
 - 7.10.c)Authentication
 - 7.10.d)Data Exchange
 - 7.10.d.i)The client side
 - 7.10.d.ii)The server side
 - 7.10.e)Computation of key for cryptography
 - 7.10.e.i)ECC
 - 7.10.e.ii)Diffie Hellman Key Exchange
- 7.11)Conclusion and future work

8)OAEP (OPTIMAL ASYMMETRIC ENCRYPTION PADDING)

- 8.1)Encoding
- 8.2)Decoding

9)Review on Network Security and Cryptography

- 9.1)Abstract
- 9.2)Introduction

9.3)Types of Security Attacks

9.3.a)Passive Attacks

9.3.b)Active Attacks

9.4)Security Services

9.4.a)Data Integrity

9.4.b)Data Confidentiality

9.4.c)Authenticity

9.4.d)Nonrepudiation

9.5.e)Access Control

9.5)Network Security Model

9.6)Need for Key Management in Cloud

9.7)Cryptography Mechanism

9.7.a)Secret Key Cryptography

9.7.a.i)Camellia

9.7.a.ii)KASUMI

9.7.b)Public-Key Cryptography

9.7.b.i)RSA

9.7.b.ii)Digital Signature Standard

9.7.c)Hash functions

9.7.c.i)Message Authentication Code

9.7.c.ii)HMAC

9.7.c.iii)CMAC

9.8)Network and Internet Security

9.8.a)Wireless Network Security

9.8.b)IP Security

9.8.c)Electronic Mail Security

9.8.d)Transport Level Security

9.9)Firewalls

9.9.a)Characteristics

9.9.b)Types

9.9.b.i)Packet Filter

9.9.b.ii)Stateful Packet Inspection

9.9.b.iii)Application-Level Gateway

HISTORY

Before the modern era, cryptography focused on message confidentiality (i.e., encryption)—conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by those for whom it was not intended without the secret, important knowledge namely the key, without which the message cannot be decrypted. Encryption aimed to ensure secrecy in communications.

The main classical cipher types are transposition ciphers, which rearrange the order of letters in a message, and substitution ciphers, which systematically replace letters or groups of letters with other letters or groups of letters

E.g: 'fly at once' => 'gmz bu podf'

Replacing each alphabet with the next alphabet

Some examples of early ciphers:

1) Caesar cipher:

Each letter in the plaintext was replaced by a letter with a fixed number of positions further down the alphabet. Suetonius reports that Julius Caesar used it with a shift of three to communicate with his generals.

Eg: If the key is to be known as right shift of 2, then

A => C, B => D and so on.

The encryption step performed by a Caesar cipher is often incorporated as part of more complex schemes, such as the Vigenere cipher, and still has modern application in the ROT13 system.

2) Atbash:

Atbash is a monoalphabetic substitution cipher originally used to encrypt the Hebrew alphabet. It is formed by taking the alphabet and mapping it to its reverse, so that the first letter becomes the last letter, the second letter becomes the second to last letter, and so on.

A => Z, B => Y,, Y => B, Z => A

3)In India, the 2000-year-old Kamasutra of Vatsyayana speaks of two different kinds of ciphers called **Kautiliyam** and **Mulavediya**.

Kautiliyam : The cipher letter substitutions are based on phonetic relations,such as vowels becoming consonants.

Mulavediya : the cipher alphabet consists of pairing letters and using the reciprocal ones.

Cryptography from 1800 to WW1:

It was in the 19th century that cryptography developed anything more than ad hoc approaches to either encryption or cryptanalysis .Cryptanalysis is the decryption and analysis of codes, ciphers or encrypted text. Cryptanalysis uses mathematical formulas to search for algorithm vulnerabilities and break into cryptography or information security systems.

Example of cryptanalysis is Charles Babbage's Crimean War era work on mathematical cryptanalysis of polyalphabetic ciphers.

In World War I the Admiralty's Room 40 (Room 40 was the cryptanalysis section of the British Admiralty during the First World War) broke German naval codes and played an important role in several naval engagements during the war, notably in detecting major German sorties into the North Sea that led to the battles of Dogger Bank and Jutland as the British fleet was sent out to intercept them. However its most important contribution was probably in decrypting the Zimmermann Telegram, a cable from the German Foreign Office sent via Washington to its ambassador Heinrich von Eckardt in Mexico which played a major part in bringing the United States into the war.

In 1917, Gilbert Vernam proposed a teleprinter cipher in which a previously prepared key, kept on paper tape, is combined character by character with the plaintext message to produce the ciphertext. This led to the development of electromechanical devices as cipher machines, and to the only unbreakable cipher, the one time pad. During the 1920s, Polish naval-officers assisted the Japanese military with code and cipher development.

World War II cryptography:

By World War II, mechanical and electromechanical cipher machines were in wide use, although—where such machines were impractical—code books and manual systems continued in use. Great advances were made in both cipher design and cryptanalysis, all in secrecy. Information about this period has begun to be declassified as the official British 50-year secrecy period has come to an end, as US archives have slowly opened, and as assorted memoirs and articles have appeared.

Germany:

The Germans made heavy use, in several variants, of an electromechanical rotor machine known as Enigma. Mathematician Marian Rejewski, at Poland's Cipher Bureau, in December 1932 deduced the detailed structure of the German Army Enigma, using mathematics and limited documentation supplied by Captain Gustave Bertrand of French military intelligence acquired from a German clerk. This was the greatest breakthrough in cryptanalysis in a thousand years and more, according to historian David Kahn. Rejewski and his mathematical Cipher Bureau colleagues, Jerzy Rozycki and Henryk Zygalski, continued reading Enigma and keeping pace with the evolution of the German Army machine's components and encipherment procedures for some time. As the Poles' resources became strained by the changes being introduced by the Germans, and as war loomed, the Cipher Bureau, on the Polish General Staff's instructions, on 25 July 1939, at Warsaw, initiated French and British intelligence representatives into the secrets of Enigma decryption.

Soon after the invasion of Poland by Germany on 1 September 1939, key Cipher Bureau personnel were evacuated southeastward; on 17 September, as the Soviet Union attacked Poland from the East, they crossed into Romania. From there they reached Paris, France; at PC Bruno, near Paris, they continued working toward breaking Enigma, collaborating with British cryptologists at Bletchley Park as the British got up to speed on their work breaking Enigma. In due course, the British cryptographers – whose ranks included many chess masters and mathematics dons such as Gordon Welchman, Max Newman, and Alan Turing (the conceptual founder of modern computing) – made substantial breakthroughs in the scale and technology of Enigma decryption.

German code breaking in World War II also had some success, most importantly by breaking the Naval Cipher No. 3. This enabled them to track and sink Atlantic convoys. It was only Ultra intelligence that finally persuaded the admiralty to change their codes in June 1943. This is surprising given the success of the British Room 40 code breakers in the previous world war.

At the end of the War, on 19 April 1945, Britain's highest level civilian and military officials were told that they could never reveal that the German Enigma cipher had been broken because it would give the defeated enemy the chance to say they "were not well and fairly beaten".

The German military also deployed several teleprinter stream ciphers. Bletchley Park called them the Fish ciphers; Max Newman and colleagues designed and deployed the Heath Robinson, and then the world's first programmable digital electronic computer, the Colossus, to help with their cryptanalysis. The German Foreign Office began to use the one-time pad in 1919; some of this traffic was read in World War II partly as the result of recovery of some key material in South America that was discarded without sufficient care by a German courier.

The *Schlüsselgerät 41* was developed late in the war as a more secure replacement for Enigma, but only saw limited use.

Japan:

A US Army group, the SIS, managed to break the highest security Japanese diplomatic cipher system (an electromechanical stepping switch machine called Purple by the Americans) in 1940, before the attack on Pearl Harbour. The locally developed Purple machine replaced the earlier "Red" machine used by the Japanese Foreign Ministry, and a related machine, the M-1, used by Naval attachés which was broken by the U.S. Navy's Agnes Driscoll. All the Japanese machine ciphers were broken, to one degree or another, by the Allies.

The Japanese Navy and Army largely used code book systems, later with a separate numerical additive. US Navy cryptographers (with cooperation from British and Dutch cryptographers after 1940) broke into several Japanese Navy crypto systems. The break into one of them, JN-25, famously led to the US victory in the Battle of Midway; and to the publication of that fact in the Chicago Tribune shortly after the battle, though the Japanese seem not to have noticed for they kept using the JN-25 system.

Allies:

The Americans referred to the intelligence resulting from cryptanalysis, perhaps especially that from the Purple machine, as 'Magic'. The British eventually settled on 'Ultra' for intelligence resulting from cryptanalysis, particularly that from message traffic protected by the various Enigmas. An earlier British term for Ultra had been 'Boniface' in an attempt to suggest, if betrayed, that it might have an individual agent as a source.

Allied cipher machines used in World War II included the British TypeX and the American SIGABA; both were electromechanical rotor designs similar in spirit to the Enigma, albeit with major improvements. Neither is known to have been broken by anyone during the War. The Poles used the Lacida machine, but its security was found to be less than intended by Polish Army cryptographers in the UK, and its use was discontinued. US troops in the field used the M-209 and the still less secure M-94 family machines. British SOE agents initially used 'poem ciphers' (memorized poems were the encryption/decryption keys), but later in the War, they began to switch to one-time pads.

The VIC cipher was a very complex hand cipher, and is claimed to be the most complicated known to have been used by the Soviets.

Typex:

Typex machines were British cipher machines used from 1937. It was an adaptation of the commercial German Enigma with a number of enhancements that greatly increased its security. The cipher machine was used until the mid-1950s.

Like Enigma, Typex was a rotor machine. Typex contained five rotors, as opposed to three or four in the Enigma. Like the Enigma, the signal was sent through the rotors twice, using a "reflector" at the end of the rotor stack. On a Typex rotor, each

electrical contact was doubled to improve reliability. Of the five rotors, typically the first two were stationary. These provided additional enciphering without adding complexity to the rotor turning mechanisms. Their purpose was similar to the plugboard in the Enigmas, offering additional randomization that could be easily changed. Unlike Enigma's plugboard, however, the wiring of those two rotors could not be easily changed day-to-day.

The major improvement the Typex had over the standard Enigma was that the rotors in the machine contained multiple notches that would turn the neighbouring rotor. This eliminated an entire class of attacks on the system, whereas Enigma's fixed notches resulted in certain patterns appearing in the ciphertext that could be seen under certain circumstances. On some models, operators could achieve a speed of 20 words a minute, and the output ciphertext or plaintext was printed on paper tape. For some portable versions, such as the Mark III, a message was typed with the left hand while the right hand turned a handle.



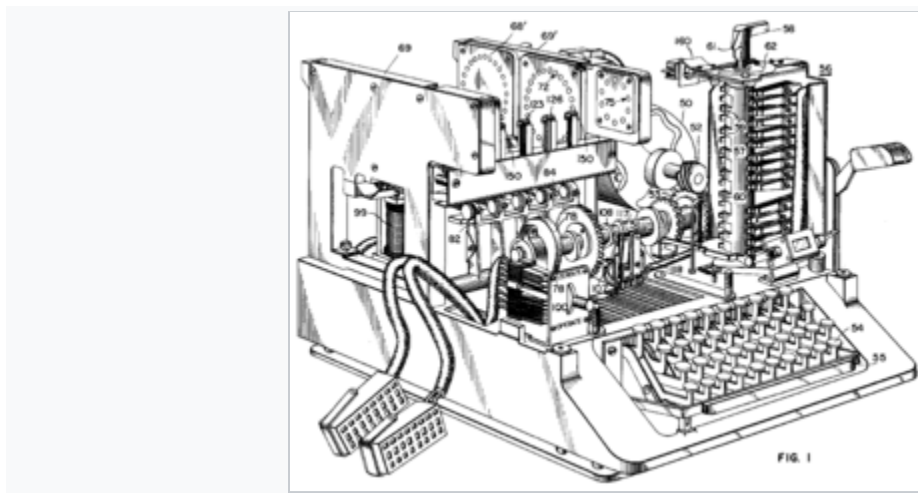
SIGABA:

The ECM Mark II was a cipher machine used by the United States for message encryption from World War II until the 1950s. The machine was also known as the SIGABA or Converter M-134 by the Army, or CSP-888/889 by the Navy, and a modified Navy version was termed the CSP-2900. Like many machines of the era it used an electromechanical system of rotors to encipher messages, but with a number of security improvements over previous designs. No successful cryptanalysis of the machine during its service lifetime is publicly known.

Because SIGABA did not have a reflector, a 26+ pole switch was needed to change the signal paths through the alphabet maze between the encryption and decryption modes. The long “controller” switch was mounted vertically, with its knob on the top of the housing. See image. It had five positions, O, P, R, E and D. Besides encrypt (E) and decrypt (D), it had a plain text position (P) that printed whatever was

typed on the output tape, and a reset position (R) that was used to set the rotors and to zeroize the machine. The O position turned the machine off. The P setting was used to print the indicators and date/time groups on the output tape. It was the only mode that printed numbers. No printing took place in the R setting, but digit keys were active to increment rotors.

Although the SIGABA was extremely secure, the US continued to upgrade its capability throughout the war, for fear of the Axis cryptanalytic ability to break SIGABA's code. When the German's ENIGMA messages and Japan's Type B Cipher Machine were broken, the messages were closely scrutinized for signs that Axis forces were able to read the US cryptography codes. Axis prisoners of war (POWs) were also interrogated with the goal of finding evidence that US cryptography had been broken. However, both the Germans and Japanese were not making any progress in breaking the SIGABA code.



M-209:

The M-209, designated CSP-1500 by the United States Navy is a portable, mechanical cipher machine used by the US military primarily in World War II, though it remained in active use through the Korean War. The M-209 was designed by Swedish cryptographer Boris Hagelin in response to a request for such a portable cipher machine, and was an improvement of an earlier machine, the C-36.

Six adjustable *key wheels* on top of the box each display a letter of the alphabet. These six wheels comprise the external key for the machine, providing an initial state, similar to an initialization vector, for the enciphering process.

To encipher a message, the operator sets the key wheels to a random sequence of letters. An enciphering-deciphering knob on the left side of the machine is set to "encipher". A dial known as the indicator disk, also on the left side, is turned to the first letter in the message. This letter is encoded by turning a hand crank or *power*

handle on the right side of the machine; at the end of the cycle, the ciphertext letter is printed onto a paper tape, the key wheels each advance one letter, and the machine is ready for entry of the next character in the message. To indicate spaces between words in the message, the letter "Z" is enciphered. Repeating the process for the remainder of the message gives a complete ciphertext, which can then be transmitted using Morse code or another method. Since the initial key wheel setting is random, it is also necessary to send those settings to the receiving party.

VIC Cipher:

The VIC cipher was a pencil and paper cipher. It was arguably the most complex hand-operated cipher ever seen, when it was first discovered. The initial analysis done by the American National Security Agency (NSA) in 1953 did not absolutely conclude that it was a hand cipher, but its placement in a hollowed out 5¢ coin implied it could be decoded using pencil and paper. The VIC cipher remained unbroken until more information about its structure was available. Although certainly not as complex or secure as modern computer operated stream ciphers or block ciphers, in practice messages protected by it resisted all attempts at cryptanalysis by the NSA from its discovery in 1953.

The VIC cipher can be regarded as the evolutionary pinnacle of the Nihilist cipher family. The Nihilist cipher is a manually operated symmetric encryption cipher, originally used by Russian Nihilists in the 1880s to organize terrorism against the tsarist regime. The term is sometimes extended to several improved algorithms used much later for communication by the First Chief Directorate with its spies. The VIC cipher has several important integrated components, including mod 10 chain addition, a lagged Fibonacci generator, a recursive formula used to generate a sequence of pseudorandom digits, a straddling checkerboard, and a disrupted double transposition.

Enigma:

The Enigma machine is a cipher device developed and used in the early- to mid-20th century to protect commercial, diplomatic, and military communication. It was employed extensively by Nazi Germany during World War II, in all branches of the German military. The Germans believed, erroneously, that use of the Enigma machine enabled them to communicate securely and thus enjoy a huge advantage in World War II. The Enigma machine was considered so secure that it was used to encipher even the most top-secret messages.

The Enigma has an electromechanical rotor mechanism that scrambles the 26 letters of the alphabet. In typical use, one person enters text on the Enigma's keyboard and another person writes down which of the 26 lights above the keyboard is illuminated at each key press. If plain text is entered, the illuminated letters are the encoded ciphertext. Entering ciphertext transforms it back into readable plaintext. The rotor

mechanism changes the electrical connections between the keys and the lights with each keypress.

The security of the system depends on a set of machine settings that were generally changed daily during the war, based on secret key lists distributed in advance, and on other settings that were changed for each message. The receiving station has to know and use the exact settings employed by the transmitting station to successfully decrypt a message.

While Nazi Germany introduced a series of improvements to Enigma over the years, and these hampered decryption efforts, they did not prevent Poland from cracking the machine prior to the war, enabling the Allies to exploit Enigma-enciphered messages as a major source of intelligence. Many commentators say the flow of Ultra communications intelligence from the decryption of Enigma, Lorenz, and other ciphers, shortened the war substantially, and might even have altered its outcome.

Cryptanalysis of the Enigma ciphering system enabled the western Allies in World War II to read substantial amounts of Morse-coded radio communications of the Axis powers that had been enciphered using Enigma machines. This yielded military intelligence which, along with that from other decrypted Axis radio and teleprinter transmissions, was given the codename *Ultra*. This was considered by western Supreme Allied Commander Dwight D. Eisenhower to have been "decisive" to Allied victory.

The Enigma machines were a family of portable cipher machines with rotor scramblers. Good operating procedures, properly enforced, would have made the Enigma machine unbreakable. However, most of the German military forces, secret services, and civilian agencies that used Enigma employed poor operating procedures, and it was these poor procedures that allowed the Enigma machines to be reverse-engineered and the ciphers to be read.

The German plugboard-equipped Enigma became Nazi Germany's principal crypto-system. It was broken by the Polish General Staff's Cipher Bureau in December 1932, with the aid of French-supplied intelligence material obtained from a German spy. A month before the outbreak of World War II, at a conference held near Warsaw, the Polish Cipher Bureau shared its Enigma-breaking techniques and technology with the French and British.

In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.

CLASSICAL CRYPTOGRAPHY

Cryptography, as derived from Greek *kryptos*, "secret" and *graph*, "writing", is the practice and study of hiding information. It means to keep the messages secret during transmission i.e., hiding of information from untrusted and unreliable elements. Cryptography is also defined as converting the original or plain text into encrypted or cipher text. The cipher text is produced with various cryptographic algorithms and keys. Keys used are secret and variable as without these the message could be decrypted by just knowing the algorithm used therefore useless. The cipher text is then sent across the transmission media and is deciphered or decrypted in its original form on the other side by the receiver. Mainly there are two main forms of cryptography: Symmetric and Public-key cryptography.

A. Symmetric-key cryptography :

Symmetric key cryptography uses the same cryptographic algorithm and the same key to encipher and decipher messages. The key is chosen randomly from all possible keys. Examples of commonly used symmetric encryption algorithms are Data Encryption Standard (DES), 3 DES, Rivest Cipher (RC-4) and International Data Encryption Algorithm (IDEA).

B. Public-key (Asymmetric) cryptography:

Asymmetric key cryptography uses two different but mathematically related keys for encryption and decryption process, one to encrypt and the other corresponding key for decryption. The key which is known publicly is called public key and the one which is kept private is called the private key. Examples of commonly used asymmetric encryption algorithms are RSA, Diffie-Hellman key exchange, ElGamal, Elliptic curve cryptography.

QUANTUM CRYPTOGRAPHY

Abstract:

Quantum cryptography is a technology that ensures ultimate security. Compared to current cryptography that could be defeated by the development of an ultra high-speed computer, quantum cryptography ensures secure communication because it is based on the fundamental physical laws. It is an emerging technology in which two parties may simultaneously generate shared, secret cryptographic key material using the transmission of quantum states of light. Quantum cryptography is a new method for secret communications offering the ultimate security assurance of the inviolability of a Law of Nature. Quantum cryptography relies on two important elements of quantum mechanics-the Heisenberg Uncertainty principle and the principle of photon polarization.

Introduction:

In our modern age of telecommunications and the Internet, information has become a precious commodity. Sometimes it must therefore be kept safe from stealing - in this case, loss of private information to an eavesdropper. There are many features to security and many applications, ranging from secure commerce and payments to private communications and protecting passwords. One essential feature for secure communications is that of cryptography, which not only protects data from stealing or modification, but can also be used for user authentication. The main aim of cryptography is to protect data transferred in the likely presence of an enemy. A cryptographic transformation of data is a procedure by which plaintext data is encrypted, resulting in an modified text, called ciphertext, that does not expose the original input. The cipher text can be reverse-altered by a designated recipient so that the original plaintext can be recaptured.

The techniques of cryptography are usually categorised as traditional or modern. Traditional techniques use operations of coding i.e. use of alternative words or phrases, transposition i.e. reordering of plaintext, and substitution i.e.alteration of plaintext characters). Whereas, modern techniques use computers, and depend upon extremely long keys, convoluted algorithms, and intractable problems to achieve assurances of security.

There are two main fields of modern cryptographic techniques: Public key encryption and Secret key encryption .

1) Public-key encryption - Messages are encrypted with a recipient's public key. The message cannot be decrypted by anyone who does not

possess the matching private key, who is thus presumed to be the owner of that key and the person associated with the public key.

2)Secret key - It is an encryption key known only to the party or parties that exchange secret messages. The risk in this system is that if either party loses the key or it is stolen, the system is broken.

The development of quantum cryptography was encouraged by the short-comings of classical cryptographic methods, which can be divided as either “public-key” or “secret-key” methods. Quantum cryptography is an approach to cryptography based on the laws of quantum physics.

History:

Quantum cryptography was first recommended by Stephen Wiesner in the early 1970s. The plan was issued in 1983 in *Sigact News*, and at the same time two scientists Bennet and Brassard, familiar with the idea of Wiesner, were ready to issue their own ideas. Then in 1984, they delivered the first quantum cryptography protocol called the "BB84." The protocol is provably secure, depending on the quantum property that information gain is only possible at the expense of disturbing the signal if the two states we are trying to distinguish are not orthogonal. The first experimental prototype based on this was made in 1991. It functioned over a distance of 32 centimeters. Over time, the technology has been improved and the distance extended to kilometers. Later on in June 2004, The first computer network in which communication is secured with quantum cryptography is up and running in Cambridge, Massachusetts. The leader of the quantum engineering team at BBN Technologies in Cambridge, Chip Elliott, transmitted the first packets of data across the Quantum Net. After that, a team at the University of Vienna transferred entangled photons across the river Danube, through free space in June 2003. In April 2004, the first money transfer encrypted by quantum keys occurred between two Austrian banks. The two buildings were 500 meters away from each other, yet fibre optics were fed through 1.5 kilometers of sewage system to link them together.

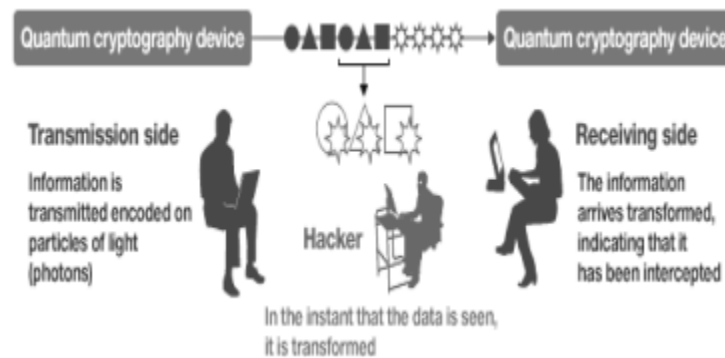
Mechanism:

Quantum cryptography depends on two important components of quantum mechanics

- 1)The Heisenberg Uncertainty principle and
- 2)The principle of photon polarization .

The Heisenberg Uncertainty principle states that, it is impossible to determine the quantum state of any system without disturbing that system. The theory of photon polarization states that an eavesdropper cannot copy unknown qubits i.e. unknown quantum states, due to the no-cloning theorem which was first introduced by Wootters

and Zurek in 1982. Depending on the theory of physics, quantum cryptography does not make it possible to eavesdrop on transmitted information. It is attracting considerable attention as a replacement for other contemporary cryptographic methods, which are based on computational security. Quantum cryptographic transmission encrypts the 0s and 1s of a digital signal on individual particles of light called photons. By contrast, modern optical transmission expresses the 0s and 1s of the digital signal as the strength and weakness of light respectively. Because the strong and weak light are made up of tens of thousands of photons which each convey the same information, if several photons are stolen (i.e., the signal is eavesdropped on) during transmission, it is not detected. On the other hand, in the case of quantum cryptography, if a third party detects (eavesdrops on) the signal, the information on the photons is suddenly transformed, meaning both that it is immediately noticeable that eavesdropping has appeared and that the third party is not able to decrypt the information.



Quantum cryptography can be used for the distribution of the secret key but for distribution of the secret key we need to secure the key and for the different basis of photon polarization are used. A pair of polarization states used to describe photon polarization such as horizontal/vertical is referred to as basis.

1. “|” denotes a photon in vertically polarized state.
2. “.” denotes a photon in horizontally polarized state.
3. “/” denotes a photon in a 45 degree polarized state.
4. “\” denotes a photon in a 135 degree polarized state.
5. “+” denotes the pair of states $\{|,.\}$, also called as the +- basis.
6. “X” denotes the pair of states $\{/, \backslash\}$, also called as the x-basis.

Application:

The most infamous and developed application of quantum cryptography is quantum key distribution (QKD). Quantum key distribution is a method used in the framework of quantum cryptography in order to produce a perfectly random key which is shared by a sender and a receiver while making sure that nobody else has a chance to learn about the key. The best known and popular scheme of quantum key distribution is based on the Bennet–Brassard protocol(i.e. BB84), which was invented in 1984 . It

depends on the no-cloning theorem [for non-orthogonal quantum states. The Bennet–Brassard protocol works as follows:

1. X creates a random bit (0 or 1) and then randomly selects one of her two bases to transmit.
2. X then prepares a photon polarization state depending both on the bit value and basis.
3. X then transmits a single photon in the state specified to Y, using the quantum channel.
4. This process is then repeated.
5. Y does not know the basis the photons were encoded in, so select a basis at random to measure.
6. After receiving all photons Y communicates with X on a public channel.
7. X broadcasts the basis each photon was sent in, and Y the basis each was measured in.
8. To check for the presence of eavesdropping X and Y now compare a certain subset of their remaining bit strings.
9. If a third party (Z) has gained any information about the photons' polarization, this will have introduced errors in Y's measurements.

X's random bit	0	1	1	0	1	0	0	1
X's random sending basis	+	+	x	+	x	x	x	+
Photon polarization X sends	↑	→	↘	↑	↘	↗	↗	→
Y's random measuring basis	+	x	x	x	+	x	+	+
Photon polarization Y measures	↑	↗	↘	↗	→	↗	→	→
Public Discussion Of Basis								
Shared secret key	0		1			0		1

-Sharing of the secret key with BB84 protocol

ADVANTAGES OF QUANTUM CRYPTOGRAPHY:

The purpose of quantum cryptography is to propose a radically different foundation for cryptography, viz. the uncertainty principle of quantum physics. Quantum cryptography can achieve most of the benefits of public-key cryptography, with the additional advantage of being provably secure, even against an opponent with superior technology and unlimited computing power, barring fundamental violation of accepted physical laws. In conventional information theory and cryptography, digital communications can always be tracked and copied, even by someone who is unaware of their meaning. Such copies can be stored and can be used in future, such as decryption of the message encrypted with the same secret key. However, when elementary quantum systems, such as polarized photons, are used to transmit digital information, the uncertainty principle gives rise to novel cryptographic phenomenon, unachievable with traditional transmission media. This principle can be used to propose a communication channel whose transmissions cannot be read or copied by an eavesdropper ignorant of certain key information used in forming the transmission. The eavesdropper cannot even gain partial information which is likely to be detected by the channel's legitimate users.

LIMITATIONS OF QUANTUM CRYPTOGRAPHY:

There are some inherent problems with using quantum states to transmit information, some to do with quantum theory itself, and some practical ones regarding equipment efficiency which affects the security of protocols.

A. Point to Point links and Denial of Service:

The quantum channel is a specialized piece of equipment, which by its very nature is a point-to-point connection: X and Y have to be at each end of it, with their photon sources and detectors. The point-to-point nature of QKD restricts potential growth, and gives rise to the possibility of a denial-of-service attack: if Z can't obtain key information, then cutting the physical link will mean X and Y can't either, which might serve Z's purposes just as well.

B. High Bit Errors Rate:

The bit error rate of a quantum key distribution is several percentages higher than an optical communication system, which can be devastating in terms of

practicality. There is an error control protocol called CASCADE that can correct the bit errors, but it also opens the system up to new attacks. The problem with CASCADE is there is a chance that a number of bits of the private key may be leaked to an attacker. One way to nullify these leaked bits is to use a process called privacy amplification. Privacy amplification takes the bit error corrected key and performs a compression function on it. This will guarantee that the bits leaked to an eavesdropper will become useless and that both parties will have the same key. This solves one problem but at the same time it weakens the security of the actual key because it is compressing the number of bits.

C. Losses in the Quantum Channel:

Free space quantum channels also have atmospheric and equipment dependent geometric losses. Since quantum signals cannot be amplified, eventually the losses on the channel will be so high that readings obtained at detectors will be indistinguishable from dark count rates. Unfortunately, it is impossible to avoid lossy channels: they introduce security weaknesses .

D. Key Distribution Rate:

The length of the quantum channel also has an effect on the achievable rate of key distribution. The rate at which key material can be sent decreases exponentially with respect to distance, and is regarded as another limiting factor in the usability of QKD systems.

E. Photon Sources and Detectors:

The quality of photon sources and detectors can have a significant impact on the security of a protocol. An ideal photon detector should have the following properties,

- High efficiency over a large spectral range.
- Low probability of generating noise (i.e. low dark count).
- The time between the detection of a photon and the corresponding electrical signal should be as constant as possible.
- The dead time after a detection event should be as small as possible to allow for higher data transfer rates.

F. Sending one photon of light at a time:

The only way the Heisenberg's uncertainty principle will combat eavesdropping is if only one copy of the photon is sent. In practice this is one challenge that has faced researchers.

G. Classical Authentication:

Quantum cryptography does not provide digital signatures and related features, such as certified mail or the ability to settle a dispute before the judge.

H. Distance Limitation:

One of the challenges for the researchers is distance limitation. Currently, quantum key distribution distances are limited to tens of kilometers because optical amplification destroys the qubit state.

Future:

For now, non-quantum cryptography is very secure , because it depends on algorithms that can't be broken in less than the lifetime of the universe by all the currently existing computers. So in theory, there is not much demand for quantum cryptography yet; thus, we don't know when this technology will take a step forward and quantum cryptography techniques will become essential to protect our information. When quantum computers will come into play, the computational speeds will increase considerably, so the mathematical complexity of algorithms will become less of a challenge. It is still arguable whether or not it will be possible to simply increase the numbers used in the algorithms and thus increase the complexity enough to outrun even quantum computers. Yet there is no question about the fact that quantum cryptography is a true invention in the field. It is still being refined and developed further. However, already it has been clear that even with its current defectiveness, it is many steps above everything that was settled before it. All we need is some years, or maybe decades or even centuries, to renew this method and make it feasible in the real world.

CRYPTOGRAPHY AGAINST PHISHING

Introduction:

Online transactions are nowadays becoming very common and there are various attacks present behind this. In these types of various attacks, phishing is identified as a major security threat and new innovative ideas are arising with this in each second so preventive mechanisms should also be so effective. Phishing is a form of online identity theft that aims to steal sensitive information such as online banking passwords and credit card information from users. Phishers attempt to fraudulently acquire sensitive information, such as passwords and credit card details, by masquerading as a trustworthy person or business in an electronic communication. The conduct of identity theft with this acquired sensitive information has also become easier with the use of technology and identity theft can be described as “a crime in which the impostor obtains key pieces of information such as Social Security and driver's license numbers and uses them for his or her own gain”.

A new method is introduced here which can be used as a safe way against phishing. As the name describes, in this approach, the website cross verifies its own

identity and proves that it is a genuine website (to use bank transaction, E-commerce and online booking system etc.) before the end users and make both the sides of the system secure as well as an authenticated one. The concept of image processing and an improved visual cryptography is used. Image processing is a technique of processing an input image and to get the output as either an improved form of the same image and/or characteristics of the input image. In Visual Cryptography an image is decomposed into shares and in order to reveal the original image, an appropriate number of shares should be combined.

Phishing web pages are forged web pages that are created by malicious people to mimic Web pages of real web sites. Most of these kinds of web pages have high visual similarities to scam their victims. Some of these kinds of web pages look exactly like the real ones. Victims of phishing web pages may expose their bank account, password, credit card number, or other important information to help webpage owners. It includes techniques such as tricking customers through email and spam messages, man in the middle attacks, installation of keyloggers and screen captures. Emails are one of the most common techniques for phishing, due to its simplicity, ease of use and wide reach. Phishers can deliver specially crafted emails to millions of legitimate email addresses very quickly and can fool the recipients utilising well known flaws in the SMTP. Some of the most common techniques used by phishers include official looking and sounding emails, copying legitimate corporate emails with minor URL changes, obfuscation of target URL information etc. Methods like virus/worm attachments to emails, crafting of personalised or unique email messages are also common.

Automated Challenge Response Method:

Researchers propose user-based mechanisms to authenticate the server. Automated Challenge Response Method is one such authentication mechanisms, includes challenge generation module from server which in turn interacts with Challenge Response interface in client and request for response from user Challenge Response module in turn will call the get response application which is installed in the client machine Once the challenge-response is validated user credentials are demanded from client and it is validated by server to proceed the transaction. Automated Challenge-Response Method ensures two way authentication and simplicity. The proposed method also prevents man-in-the-middle attacks since the response is obtained from the executable which is called by the browser and third man interruption is impossible. Here instead of getting a response from the get-response executable it is better to update the get-response executable automatically from the bank server when the responses are about to nullify.

Blacklist:

Blacklist is a DNS based anti-phishing approach technique now most commonly used by the browser. Anti Phishing Work Group, Google and other organizations have provided an open blacklist query interface. Internet Explorer7, Netscape Browser8.1, Google Safe Browsing (a feature of the Google Toolbar for Firefox) are important

browsers which use blacklists to protect users when they are navigating through phishing sites. Because every URL in the blacklist has been verified by the administrator, the false alarm probability is very low. However there are a lot of technical disadvantages. Firstly, the phishing websites found are a very small proportion so the failed alarm probability is very high. Secondly, generally speaking, the life cycle of a phishing website is only a few days. A website might be shut down before we find and verify that it is a phishing website. Heuristic-based anti-phishing technique is to estimate whether a page has some phishing heuristics characteristics.

Phish-Secure:

A three factor authentication scheme named Phish-Secure focuses on counter attack phishing. Here as a first factor of authentication, an image similarity detection is done which helps in finding out which page the user tends to visit, then it is checked for Phishing. For this purpose a system captures the image of a webpage in a particular resolution in the required format. This image is termed as Visual image. If the attacker is going to create a Phishing site he is going to use the replica of the original webpage in order to fool the users. Now Phish-Secure gets the Visual image of the visited page and collects the mean RGB value of the image. This is termed as V_RGB. The database with Phish-Secure uses consists of details about the page which has to be authenticated. The actual mean RGB of various WebPages is stored in the database which is denoted as A_RGB. Phish-Secure will utilize this information and make a comparison to find out the similarity between the visited page and the page in the database. The similarity is obtained in means of percentage, if the percentage of similarity (PS) is greater than 99% then Phish-Secure concludes which website the user is tending to visit. This is carried out by taking the corresponding URL in the database and checking is done in order to find whether the site is Phishing or not.

As a second factor of authentication Phish-Secure grabs the destination IP in Layer 3 which gives information about to which IP address the user is getting connected, this is referred to as VIP. If an attacker's web server IP address has already been found guilty the particular IP is blacklisted. Phish-Secure checks this Blacklist with the VIP and will warn the user. On the other hand if the VIP is not found in the Blacklist, further verification is done in the following step.

Here in this step Phish-Secure grabs the actual list of IP addresses of the provider which he tends to connect. This is because any provider may have multiple servers for the purpose of load balancing and the user may be connected to his location accordingly. In order to avoid any confusion Phish-Secure gets the list of IP addresses which is referred to as actual IP and is checked with the V_IP (ie) the IP address to which the user is getting connected. If these two IP addresses are same Phish-Secure identifies the particular site as genuine and returns a message as authenticated. On the other hand if there is a mismatch in the above verification Phish-Secure identifies the site as Phishing and warns the user. In addition to this the VIP is added to the black list.

so that in future if the attacker uses the same web server and tries to attack Phish Secure detects the site as Phishing in the second step.

These Popular Technologies Have Several Drawbacks:

- Blacklist-based technique with low false alarm probability, but it cannot detect the websites that are not in the blacklist database. Because the life cycle of phishing websites is too short and the establishment of a blacklist has a long lag time, the accuracy of the blacklist is not too high.
- Heuristic based anti-phishing technique, with a high probability of false and failed alarm and it is easy for the attacker to use technical means to avoid the heuristic characteristics detection.
- Similarity assessment based technique is time consuming. It takes too long a time to calculate a pair of pages, so using the method to detect phishing websites on the client terminal is not suitable. And there is a low accuracy rate for this method depending on many factors, such as the text, images and similarity measurement technique. However, this technique (in particular, image similarity identification technique) is not perfect enough yet.

LARX:















An offline phishing detection system named LARX, acronym for Large-scale Anti-phishing by Retrospective data-eXploration to counter phishing attacks has been proposed. First, it uses traffic archiving in a vantage point to collect network trace data. Secondly, LARX leverages cloud computing technology to analyze the experimental data in a way similar to the "divide and conquer" scheme. It used two existing cloud platforms, Amazon Web Services and Eucalyptus. A physical server is also used for comparison. All of LARX's phishing filtering operations are based on a cloud computing platform and work in parallel. Finally, as an offline solution, LARX can be effectively scaled up to analyze a large volume of network trace data for phishing attack detection.

There are various mutual authentication methods using cell phones such as browsing using phones, password generation etc. There are various problems regarding these methods such as hijacking account setup. theft of the trusted device and attacks on the network

Thus there are various methods present in online manipulations for making the systems safe from these types of attacks But we can see that they have their own problems which make it again unsafe. So a system based on visual cryptography which can perform as a new method can overcome these problems effectively.

Visual Cryptography:

Each pixel in the original image is encrypted into two sub pixels called shares.

Pixel	Probability	Shares #1 #2	Superposition of the two shares
	$p = 0.5$	 	
	$p = 0.5$	 	
	$p = 0.5$	 	
	$p = 0.5$	 	

White Pixels

Black Pixels

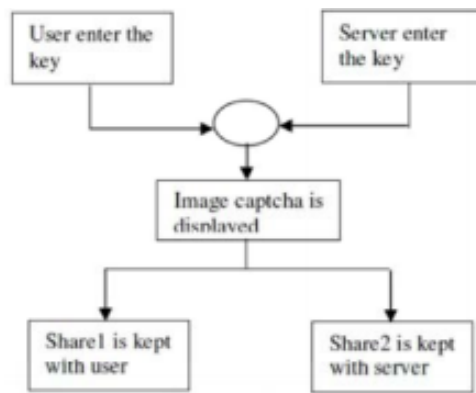
This image denotes the shares of a white pixel and a black pixel. The choice of shares for a white and black pixel is randomly determined (there are two choices available for each pixel). Neither share provides any clue about the original pixel since different pixels in the secret image will be encrypted using independent random choices. When the two shares are superimposed, the value of the original pixel P can be determined. If P is a black pixel, we get two black sub pixels, if it is a white pixel, we get one black sub pixel and one white sub pixel.

For phishing detection and prevention, we are proposing a new methodology to detect the phishing website. Our methodology is based on the Anti-Phishing Image Captcha validation scheme using visual cryptography. It prevents password and other confidential information from the phishing websites.

The proposed approach can be divided into two phases:

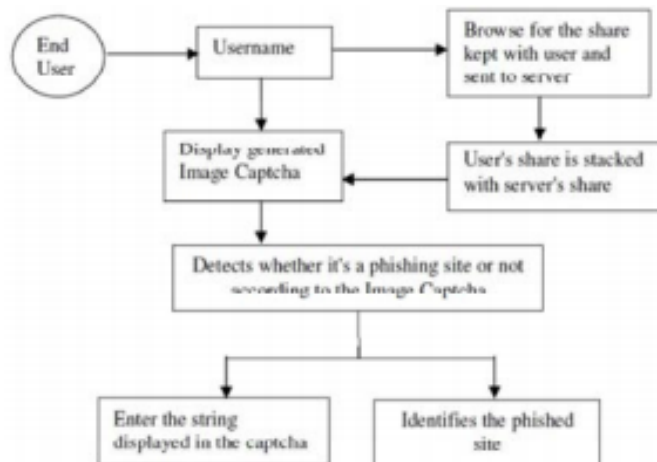
Registration Phase:

In the registration phase, a keystring(password) is asked from the user at the time of registration for the secure website. The key string can be a combination of alphabets and numbers to provide a more secure environment. This string is concatenated with a randomly generated string in the server and an image captcha is generated. The image captcha is divided into two shares such that one of the shares is kept with the user and the other share is kept in the server. The user's share and the original image captcha is sent to the user for later verification during the login phase. The image captcha is also stored in the actual database of any confidential website as confidential data. After the registration, the user can change the key string when it is needed.



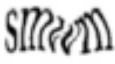



Login Phase:

In the Login phase first the user is prompted for the username (user id). Then the user is asked to enter his share which is kept with him. This share is sent to the server where the user's share and share which is stored in the database of the website, for each user, is stacked together to produce the image captcha. The image captcha is displayed to the user. Here the end user can check whether the displayed image captcha matches with the captcha created at the time of registration. The end user is required to enter the text displayed in the image captcha and this can serve the purpose of password and using this, the user can log in into the website. Using the username and image captcha generated by stacking two shares one can verify whether the website is a genuine secure website or a phishing website and can also verify whether the user is a human user or not.





When a user attempts to log in into the site, in order to increase the security the image captcha is encrypted using many algorithms. This encryption Phase contains many algorithms like Blowfish, Splitting and Rotating algorithm and Visual Cryptography Scheme. First the "Blowfish Algorithm" is applied to the original image captcha then the image captcha is divided into many blocks and rearranged. After the image captcha blocks are rearranged, the "Splitting and Rotating Algorithm" is applied to the image

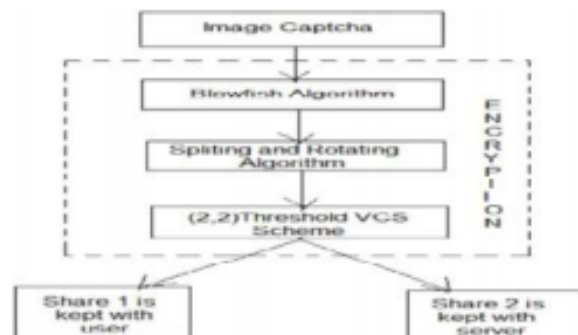
captcha, then the rearranged blocks are rotated. Then the rearranged and rotated blocks are combined. Then the VCS scheme is applied to the combined blocks. This scheme is used to divide the encrypted image captcha into two shares based on white and black pixels. When the two subpixels are identical blocks it is considered as a white pixel. Likewise when the two subpixels are different the original pixel is considered as black pixel. This VCS scheme adds more complication to the image captcha. At last one of the shares is kept with the user and another part of the share is kept with the server. When two shares are stacked together and the reverse process of encryption takes place the original image captcha is revealed. From this the user can check whether the website is original or fake. At the same time the server can verify whether the user is a human being or a robot.

Original Captcha	Share 1	Share 2	Reconstructed Captcha
			

Case1:

Original Captcha	Share 1	Share 2	Reconstructed Captcha
			

Case2:



GENETIC ALGORITHM AND PSEUDO RANDOM SEQUENCE GENERATING FUNCTIONS

Genetic algorithms (GAs) are a class of optimization algorithms belonging to symmetric cryptography . Many problems can be solved using genetic algorithms through modeling a simplified version of genetic processes. The concept of genetic algorithms can be used with pseudorandom functions to encrypt and decrypt data streams. The encryption process is applied over a binary file so that the algorithm can be applied over any type of text as well as multimedia data.

Operators of GAs:

The genetic algorithm is a search algorithm based on the mechanics of natural selection and natural genetics. The genetic algorithm belongs to the family of evolutionary algorithms, along with genetic programming, evolution strategies, and evolutionary programming. The set of operators usually consists of mutation, crossover and selection.

Crossover:

Crossover operator has the significance as that of crossover in natural genetic process. In this operation two chromosomes are taken and a new one is generated by taking some attributes of the first chromosome and the rest from the second chromosome. In GAs a crossover can be of following types:

1. Single Point Crossover:

In this crossover, a random number is selected from 1 to n as the crossover point, where n being the number of chromosomes. Any two chromosomes are taken and an operator is applied.

2. Two Point Crossover:

In this type of crossover, two crossover points are selected and the crossover operator is applied.

3. Uniform Crossover:

In this type, bits are copied from both chromosomes uniformly

Mutation:

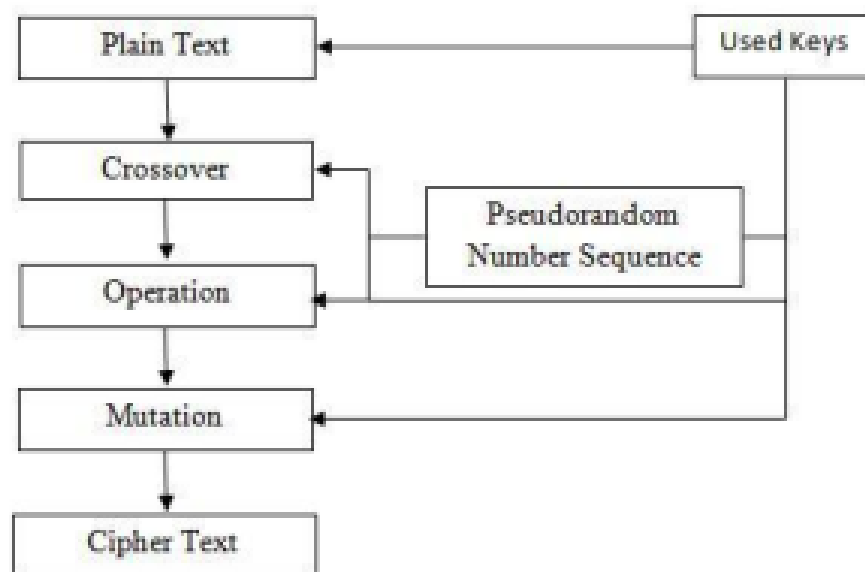
Mutation is a genetic operator used to maintain genetic diversity from one generation of population to the next. It is similar to biological mutation. Mutation allows the algorithm to avoid local minima by preventing the population chromosomes from

becoming too similar to each other. GAs involve string based modifications to the elements of a candidate solution. These include bit-reversal in bit-string GAs.

Pseudorandom number generator:

A pseudorandom number generator is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state. A PRNG suitable for Cryptographic applications is called a cryptographically secure PRNG (CSPRNG).

The Encryption process:



Proposed method for encrypting binary files:

Pseudorandom Sequence:

PRNG used to generate a Pseudorandom Sequence of numbers for our encryption and decryption method. In this method we are using two sequence and the values needed to generate those sequence are stored as Secret key

Crossover:

In this method we are using three types of crossover operation (Single point, Two Point, Uniformed). Modulating the Pseudorandom Sequence by 3 the crossover operations will executed according to the sequence

Single point crossover:



Two point crossover:



Uniformed crossover:



Mutation:

For Mutation operation we are using flip bit mutation technique (i.e. if the genome bit is 1, it is changed to 0 and vice versa). Mutation operation will be applied on one of the binary data blocks. The index number of that data block is stored as Secret key.

Key structure:

In the algorithm we have basically used five keys: one is for dividing the plane text into blocks. Second is for generating pseudorandom sequences for crossover operation. Third is for generating another pseudorandom sequence, fourth key is for modulating the sequence (Range 16 - 255) and last one is for mutation operation.

Encryption:

1. Transform the file into binary sequences and divide the sequence containing 'n' bits in each block. The value of 'n' is a Secret Key.

2. Generate the first pseudorandom sequence of numbers and modulate the sequence by 3 to select any of the three crossover operations, from that sequence we can select:
 - a) Single point
 - b) Uniformed
 - c) Two point

The crossover operation will be applied on each block of binary digits.

3. Generate another pseudorandom sequence of numbers and modulate the sequence with a secret key (range from 16-255) We denote the sequence as:

$$Z_1, Z_2, Z_3, \dots, Z_n$$

Find out the decimal value of each crossover blocks i.e.:

$$C_1, C_2, C_3, \dots, C_n$$

Now do the following operation,

$$X_i = Z_i \oplus Z_i \ll (n/2)$$

$$E_i = C_i \oplus X_i, \quad i = 1, 2, 3, 4, \dots, n$$

4. Repeat step 3 until end of the data.

5. Now between $E_1, E_2, E_3, \dots, E_n$ block select a block for mutation. Perform mutation operation on the mentioned block, the number will be then saved into the key file,

$$E_i = (255 - E_i)$$

i.e. the encrypted block looks like:

$$E_1, E_2, (255 - E_3), \dots, E_n.$$

6. By printing the $E_1, E_2, E_3, \dots, E_n$ values into a file (any file format) we can get our cipher text.

Decryption:

The steps for decryption are just reversal of the encryption process.

1) Read the key file and read the values from the reverse direction. Read the encrypted text file to get the encrypted text and divide it into 'n' bit per sequence mentioned in the key.

- 2) Do Mutation of the mentioned block number in the key:

$$E_i = 255 - (255 - E_i)$$

3) Generate a pseudorandom sequence and modulate the sequence with the secret key,

We denote the sequence as:

$Z_1, Z_2, Z_3 \dots Z_n$

Find out the decimal value of each crossover blocks i.e.:

$E_1, E_2, E_3 \dots E_n$

Now do the following operation,

$$X_i = Z_i \oplus Z_{i \ll (n/2)}$$

$$C_i = E_i \oplus X_i, i = i+1 (i = 1, 2, 3, 4 \dots n)$$

4) (Crossover Operation) Generate the pseudorandom sequence by reading the keys from the key file and modulate the sequence by 3 to select the three crossover operations:

- a. Single point
- b. Uniformed
- c. Two point

5) After the reverse crossover process, convert the plain binary bit blocks into the desired output file format as the decrypted file.

SYMMETRIC KEY CRYPTOGRAPHY

Symmetric encryption transforms plaintext into cipher-text using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the cipher-text. A Symmetric encryption scheme has five ingredients - Plaintext, Encryption Algorithm, Secret Key, Cipher-text and Decryption Algorithm. The Secret Key is shared by both, the sender and the receiver which they must have obtained in a secure fashion & should keep the key hidden, lest anyone who finds the key would be able to extract the hidden message

The Symmetric encryption, also referred to as the conventional system, or single-key system, was the only type of encryption in use prior to the development of Public-Key encryption in the 1970s. It remains by far the most widely used of the two types of encryption. It was in use way before the computer era, and can be traced back to ancient Rome & Egypt. The ciphers which were in use before the advent of the computers are termed as the classical encryption algorithms. They were all very intriguing & worked on texts, but now is the time of bits & bytes. Therefore, many new ciphers were created to help maintain the secrecy of the digital information. Some of the ciphers are:

A. Data Encryption Standard (DES):

DES is the most widely used symmetric cipher. It was designed by IBM based on their Lucifer Cipher. DES is a 64 bit block cipher which means that it encrypts data 64 bits at a time. DES is based on a cipher known as the Feistel block cipher. This was a block cipher developed by the IBM cryptography researcher Horst Feistel in the early 70's. As with most encryption schemes, DES expects two inputs - the plaintext to be encrypted and the secret key. It consists of a number of rounds where each round contains bit-shuffling, non-linear substitutions (S-boxes) and exclusive OR operations. Initially, 56 bits of the key are selected from the initial 64 by permuted choice. The remaining eight bits are either discarded or used as parity check bits. The 56 bits are then divided into two 28-bit halves; each half is thereafter treated separately. In successive rounds, both halves are rotated left by one or two bits and then 48 sub key bits are selected by permuted choice, 24 bits from the left half and 24 from the right. The key schedule for decryption is similar, the sub keys are in reverse order compared to encryption.

B. Triple DES:

Triple Data Encryption Algorithm block cipher applies the DES cipher three times to each block of data. The original DES cipher's key size of 56 bits was initially sufficient, but the increase in computational power with time made brute-force attacks feasible. Triple DES provides an easy method of increasing the security of DES to protect against such attacks. It takes three 64-bit keys, making an overall key length of 192 bits. Applying DES three times consecutively using the three different keys makes the encryption triple strong. Triple DES runs three times slower than DES, but it is much more secure. The method for decrypting is the same as the method for encryption, except it is executed backwards.

C. Advanced Encryption Standard (AES):

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES is based on the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. AES is a block cipher with a block length of 128 bits. AES allows for three different key lengths: 128, 192, or 256 bits. Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical. Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption.

D. RC2:

In cryptography RC2 (also known as ARC2) is a symmetric block-key cipher designed by Ron Rivest in 1987. "RC" stands for "Ron's Code" or "Rivest Cipher". The development of RC2 was sponsored by Lotus under the guidance of NSA. Lotus were seeking a custom cipher that could be exported as part of their Lotus Notes software. Initially, the details of the algorithm were kept secret but source code for RC2 was anonymously posted to the Internet. RC2 is a 64-bit block cipher with a variable size key. Its 18 rounds are arranged as a source-heavy Feistel network, with 16 rounds of one type (MIXING) punctuated by two rounds of another type (MASHING)

E. RC4:

Another cipher designed by Ron Rivest, RC4 was initially a trade secret, but in the year (1994) that it was created, a description of it was leaked on the Internet and was confirmed to be genuine as it was yielding the same result as the licensed RC4. The name RC4 is trademarked, so RC4 is frequently alluded to as ARCFOUR or ARC4 (meaning alleged RC4) to stay away from trademark issues. RSA Security has never formally released the algorithm; Rivest has, in any case, linked to the English Wikipedia article on RC4 in his own course notes. RC4 has gotten to be a piece of

some commonly utilized encryption conventions and guidelines, including WEP and WPA for wireless cards and TLS.

F. RC5:

In cryptography, RC5 is a symmetric-key block cipher notable for its simplicity. Designed by Ronald Rivest in 1994, RC stands for "Rivest Cipher", or alternatively, "Ron's Code". The Advanced Encryption Standard (AES) candidate RC6 was based on RC5. Unlike many schemes, RC5 has a variable block size (32, 64 or 128 bits), key size (0 to 2040 bits) and number of rounds (0 to 255). The original suggested choices of parameters were a block size of 64 bits, a 128-bit key and 12 rounds. A key feature of RC5 is the use of data-dependent rotations; one of the goals of RC5 was to prompt the study and evaluation of such operations as a cryptographic primitive. RC5 also consists of a number of modular additions and exclusive-OR (XOR)s. The general structure of the algorithm is a Feistel-like network .

G. RC6:

Rivest Cipher 6 (RC6) is a symmetric key block cipher derived from RC5. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) competition. The algorithm was one of the five finalists, and also was submitted to the NESSIE and CRYPTREC projects. It is a proprietary algorithm, patented by RSA Security. RC6 is very similar to RC5 in structure, using data-dependent rotations, modular addition, and XOR operations; in fact, RC6 could be viewed as interweaving two parallel RC5 encryption processes, however, RC6 does use an extra multiplication operation not present in RC5 in order to make the rotation dependent on every bit in a word, and not just the least significant few bits .

H. Blowfish:

Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Blowfish algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches. The algorithm consists of two parts: a key expansion part and a data-encryption part. Key expansion converts a key of at most 448 bits into several sub keys arrays totalling 4168 bytes.

I. Twofish:

Twofish is a symmetric key block cipher with a block size of 128 bits and key sizes up to 256 bits. It was one of the five finalists of the AES contest, but was

not selected for standardization. Twofish has been developed on the earlier block cipher Blowfish. It was designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson. The Twofish cipher has not been patented and the reference implementation has been placed in the public domain. As a result, the Twofish algorithm is free for anyone to use without any restrictions whatsoever. It is one of a few ciphers included in the OpenPGP standard (RFC 4880). However, Twofish has seen less widespread usage than Blowfish, which has been available longer. Twofish's distinctive features are the use of pre-computed key-dependent Sboxes, and a relatively complex key schedule. One half of an n-bit key is used as the actual encryption key and the other half of the n-bit key is used to modify the encryption algorithm (key-dependent S-boxes) .

Comparison:

Algorithm	Creator	Block size	Key length	Rounds	Algorithm	Effectiveness	Attacks
DES	IBM (1975)	64 bits	56 bits	16	Fiestel Network	Slow	Brute Force
3DES	IBM (1978)	64 bits	64*3 = 192 bits	48	Fiestel Network	Slow specially in Software	Theoretically Possible
AES	J. Daemen and V. Rijmen (1998)	128, 192, 256 bits	128, 192, 256	9, 11, 13	Substitution Permutation Network	Effective in both Hardware & Software	Side Channel Attacks
RC2	Ron Rivest (1994)	64 bits	8 – 1024 bits	16	Source Heavy Fiestel Network	Efficient in Software	Related Key Network
RC4	Ron Rivest (RSA Security) (1994)	2064 bits, 1684 effective	40 – 2048 bits	256	Fiestel Network	Effective in both Hardware & Software	Fluhrere Mantin & Shamir Attack
RC5	Ron Rivest (1994)	32, 64, 128 bits	0 – 2040 bits (128 suggested)	1 – 255 (12 suggested originally)	Fiestel Network	Slow	Differential Attack
RC6	Ron Rivest et al. (1998)	128 bits	128, 192, 256 bits	20	Fiestel Network	Slow	Brute Force
BLOWFISH	Bruce Schneier (1993)	64 bits	32 – 488 bits (128 default)	16	Fiestel Network	Efficient in Software	Differential Attack, Pseudorandom Permutation
TWOFISH	Bruce Schneier et al. (1998)	128 bits	128, 192, 256 bits	16	Fiestel Network	Efficient in Software	Truncated Differential Cryptanalysis (Partially Broken)

Cryptography during Data Sharing and Accessing Over Cloud

Abstract:

Data sharing is an important functionality in cloud environment. With the advent of the World Wide Web and the emergence of e-commerce applications and social networks, organizations across the world generate a large amount of data daily. This data would be more useful to cooperating organizations if they were able to share their data. In this article, an efficient methodology is provided to securely, efficiently, and flexibly share data with others in cloud computing. In this technique, the secret key holder can release a constant-size aggregate key for flexible choices of cipher-text set in cloud storage, but the other encrypted files outside the set remain confidential. Secure cryptographic architecture and working methodology are proposed in this paper for optimal services over the cloud.

Introduction:

Cloud computing represents one of the magnificent shifts in information technology which can enhance collaboration, agility, scaling and availability, and provide the potential for cost reduction through optimized and efficient computing. Different from the existing technologies and computing approaches, cloud is defined with five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), SPI service models (Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)), and deployment models (Public, Private, Hybrid, Community).

Cloud Sharing is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Access control is generally a policy or procedure that allows, denies or restricts access to a system. It may, as well, monitor and record all attempts made to access a system. Access Control may also identify users attempting to access a system unauthorized. It is a mechanism which is very much important for protection in computer security. Various access control models are in use, including the most common Mandatory Access Control (MAC), Discretionary Access Control (DAC) and

Role Based Access Control (RBAC). All these models are known as identity based access control models. In all these access control models, users (subjects) and resources (objects) are identified by unique names. Identification may be done directly or through roles assigned to the subjects. These access control methods are effective in an unchangeable distributed system, where there are only a set of Users with a known set of services.

Challenges:

There are two important challenges in secure outsourcing.

- 1) The stored data must be protected against unauthorized access.
- 2) Both the data and the access to data need to be protected from cloud storage service providers (e.g., cloud system administrators).

In these scenarios, relying on password and other access control mechanisms is insufficient. Cryptographic encryption mechanisms are typically employed. However, simply having encryption and decryption implemented in the cloud database systems is insufficient. In order to support both challenges, data should be encrypted first by users before it is outsourced to a remote cloud storage service and both data security and data access privacy should be protected such that cloud storage service providers have no abilities to decrypt the data, and when the user wants to search some parts of the whole data, the cloud storage system will provide the accessibility without knowing what the portion of the encrypted data returned to the user is about.

The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial. In a shared-tenancy cloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM co-resident with the target one.

Types of Attacks on the Cloud

There are a number of types of privacy and security attacks in the Cloud. The following contains a summary of the common types of attacks that may occur in the Cloud.

1. Flooding Attacks:

A malicious user can send requests to the Cloud; he/she can then easily overload the server by creating bogus data requests to the Cloud. The attempt is to increase the workload of the Cloud servers by consuming lots of resources needlessly.

2. Law Enforcement Requests:

When the FBI or government demand a Cloud Service Provider access to its data, the Cloud Service Provider is least likely to deny them. Hence, there may be an inherent threat to user privacy and confidentiality of data.

3. Data Stealing Attacks :

A term used to describe the stealing of a user account and password by any means such as through brute-force attacks or over-the-shoulder techniques. The privacy and confidentiality of user's data will be severely breached. A common mechanism to prevent such attacks is to include an extra value when authenticating. This value can be distributed to the right user by SMS and hence mitigate the likelihood of data confidentiality issues.

4. Denial-of-Service Attacks :

Malicious code is injected into the browser to open many windows and as a result deny legitimate users access to services.

5. XML Signature Wrapping Attacks:

Using different kinds of XML signature wrapping attacks, one can completely take over the administrative rights of the Cloud user and create, delete, modify images as well as create instances.

6. Cross site scripting attacks:

Attackers can inject a piece of code into web applications to bypass access control mechanisms. Researchers found this possible with Amazon Web Services in November 2011. They were able to gain free access to all customer data, authentication data, and tokens as well as plaintext passwords.

Data Sharing and Accessing in the Cloud:

With the advancements in Cloud computing, there is now a growing focus on implementing data sharing capabilities in the Cloud. With the ability to share data via the Cloud, the number of benefits increases multifold. As businesses and organizations are now outsourcing data and operations to the Cloud, they benefit further with the ability to share data between other businesses and organizations. Employees also benefit as they can share work and collaborate with other employees and can also continue working at home or any other place such as the library. They don't need to worry about losing work as it is always in the Cloud. With social users, the ability to share files, including documents, photos and videos with other users provides great benefit to them. When considering data sharing and collaboration, simple encryption techniques do not suffice, especially when considering key management. To enable secure and confidential data sharing and collaboration in the Cloud, there needs to first be proper key management in the Cloud. However, the main problem with data sharing in the

Cloud is the privacy and security issues. The Cloud is open to many privacy and security attacks, which make many users wary of adopting Cloud technology for data sharing purposes.

Requirements of Data Sharing in the Cloud:

To enable data sharing in the Cloud, it is imperative that only authorised users are able to get access to data stored in the Cloud. The ideal requirements of data sharing in the Cloud are:

- The data owner should be able to specify a group of users that are allowed to view his/her data
- Any member of the group should gain access to the data anytime without the data owner's intervention.
- No other user, other than the data owner and the members of the group, should gain access to the data, including the Cloud Service Provider.
- The data owner should be able to revoke access to data for any member of the group.
- The data owner should be able to add members to the group.
- No member of the group should be allowed to revoke rights of other members of the group or join new users to the group.
- The data owner should be able to specify who has read/write permissions on the data owner's files

Achieving privacy and security requirements in the Cloud architecture can go a long way to attracting large numbers of users to adopting and embracing Cloud technology.

• Data Confidentiality:

Unauthorized users (including the Cloud), should not be able to access data at any given time. Data should remain confidential in transit, at rest and on backup media. Only authorized users should be able to gain access to data.

• User revocation:

When a user is revoked access rights to data, that user should not be able to gain access to the data at any given time. Ideally, user revocation should not affect other authorized users in the group for efficiency purposes.

• Scalable and Efficient:

Since the number of Cloud users tends to be extremely large and at times unpredictable as users join and leave, it is imperative that the system maintain efficiency as well as be scalable.

• Collusion between entities:

When considering data sharing methodologies in the Cloud, it is vital that even when certain entities collude, they should still not be able to access any of the

data without the data owner's permission. Earlier works of literature on data sharing did not consider this problem, however collusion between entities can never be written off as an unlikely event.

Need for Key Management in Cloud :

Encryption provides data protection while key management enables access to protected data. It is strongly recommended to encrypt data in transit over networks, at rest, and on backup media. In particular, data encryption at rest (e.g., for long-term archival storage) can avoid the risk of malicious cloud service providers or malicious multi-tenants abuse. At the same time, secure key stores (including key backup and recoverability) and access to key stores must be securely implemented since improper (or access to) key storage could lead to the compromise of all encrypted data. Key management is anything you do with a key except encryption and decryption and covers the creation/deletion of keys, activation/deactivation of keys, transportation of keys, storage of keys and so on. Most Cloud service providers provide basic key encryption schemes for protecting data or may leave it to the user to encrypt their own data. Both encryption and key management are very important to help secure applications and data stored in the Cloud. Requirements of effective key management are

- **Secure key stores:**

The key stores themselves must be protected from malicious users. If a malicious user gains access to the keys, they will then be able to access any encrypted data the key is corresponded to. Hence the key stores themselves must be protected in storage, in transit and on backup media.

- **Access to key stores:**

Access to the key stores should be limited to the users that have the rights to access data. Separation of roles should be used to help control access. The entity that uses a given key should not be the entity that stores the key.

- **Key backup and recoverability:**

Keys need secure backup and recovery solutions. Loss of keys, although effective for destroying access to data, can be highly devastating to a business and Cloud providers need to ensure that keys aren't lost through backup and recovery mechanisms

Identity and Access Management :

Secure management of identity and access control is a critical factor to prevent account and service hijacking. It is strongly recommended to prohibit sharing of account credentials, to leverage strong (multi-factor) authentication if possible, and to consider delegated authentication and managing trust across all types of cloud services. Access

control is a security feature that controls how users and systems communicate and interact with one another. Access means flow of information between subject and object. Subject is an active entity that requests access to an object or the data in an object whereas an object is a passive entity that contains information. There are broadly three types of access control:

- **Role Based Access Control (RBAC):**

In RBAC, users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For example, only faculty members and senior secretaries might have access to data but not the junior secretaries.

- **User Based Access Control (UBAC):**

In UBAC, the access control list (ACL) contains the list of users who are authorized to access data. This is not feasible in clouds where there are many users.

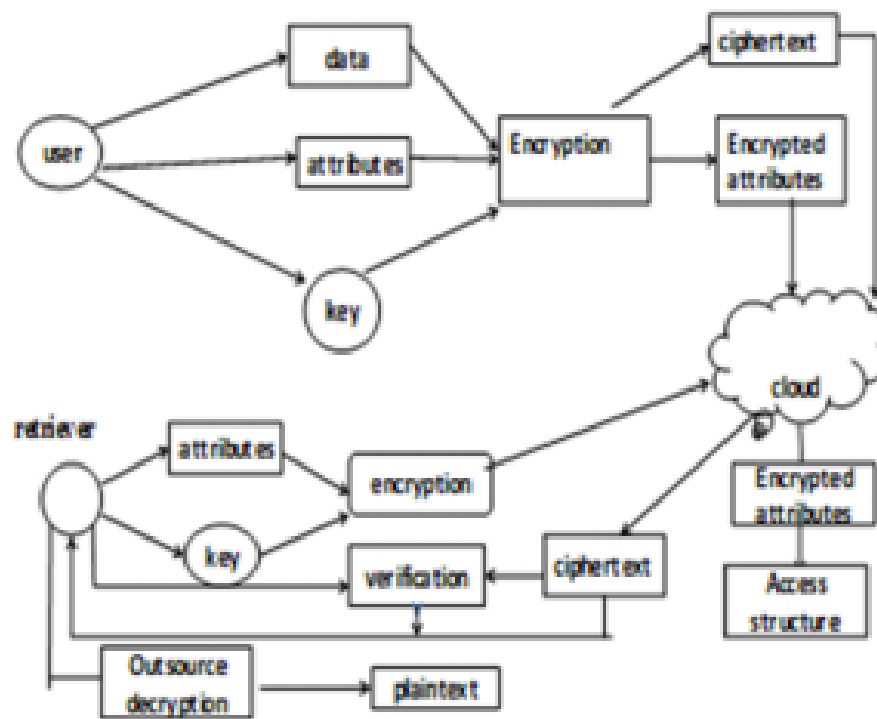
- **Attribute Based Access Control (ABAC):**

ABAC is more extended in scope, in which users are given attributes, and the data has an attached access policy. Only users with a valid set of attributes, satisfying the access policy, can access the data.

It is important to control the access of data so that only authorized users can access the data. Using ABE, the records are encrypted under some access policy and stored in the cloud. Users are given sets of attributes and corresponding keys. Access control is also gaining importance in online social networking where users (members) store their personal information, pictures, and videos and share them with selected groups of users or communities they belong to.

Proposed scheme:

In this section a cryptographic architecture model is proposed in the Cloud environment. In this technique a user or sender interacts with three elements: Data, Attribute and Key. These elements are used to encrypt the message. Encrypted messages are also known as cipher –text. Now cipher-text is sent to the receiver via cloud or network. There may be a process of verification of messages if needed. To get the message in plain-text, cipher-text is decrypted by using key and attributes values in public-key encryption fashion.



Assumptions:

In this paper, assumptions are made as follows:

- 1) The cloud is honest-but-curious, which means that the cloud administrators can be interested in viewing the user's content, but cannot modify it. Honest-but-curious model of adversaries do not tamper with data so that they can keep the system functioning normally and remain undetected.
- 2) Users can have either read or write or both access to a file stored in the cloud.
- 3) All communications between users/clouds are secured by Secure Shell Protocol, SSH.

Mathematical Background:

Bilinear pairings on elliptic curves are used. Let G be a cyclic group of prime order q generated by g . Let G_T be a group of order q . We can define the map $e : G \times G \rightarrow G_T$. The map satisfies the following properties:

- 1) $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G$ and $a, b \in \mathbb{Z}_q$, $\mathbb{Z}_q = \{0, 1, 2, \dots, q-1\}$
- 2) Non-degenerate : $e(g, g) = 1$.

Bilinear pairing on elliptic curves groups is used. The choice of curve is an important consideration because it determines the complexity of pairing operations.

Formats of Access Policies:

Access policies can be in any of the following formats:

- 1) Boolean functions of attributes,
- 2) Linear Secret Sharing Scheme (LSSS) matrix, or
- 3) Monotone span programs. Any access structure can be converted into a Boolean function.

Let $Y: \{0,1\}^n \rightarrow \{0,1\}$ be a monotone Boolean function. A monotone span program for Y over a field F is an $l \times t$ matrix M with entries in F , along with a labeling function $a: [l] \rightarrow [n]$ that associates each row of M with an input variable of Y , such that, for every

$(x_1, x_2, x_3, \dots, x_n) \in \{0,1\}^n$, the following condition is satisfied:
 $Y(x_1, x_2, x_3, \dots, x_n) = 1 \Leftrightarrow \exists v \in F^{[l]} : vM = [1, 0, 0, 0, \dots, 0]$ and $(\forall i : x_{a(i)} = 0 \Rightarrow v_i = 0)$

In other words, $Y(x_1, x_2, x_3, \dots, x_n) = 1$ if and only if the rows of M indexed by $\{i \mid x_{a(i)} = 1\}$ span the vector $[1, 0, 0, 0, \dots, 0]$.

For handling the fault tolerance, there should be several Key Distribution Centre (KDC) located at multiple servers over the cloud. It helps in parallel encryption and distributed processing of messages. Attributes should also be distributed at multiple servers. The use of public-key encryption gives more flexibility for the cloud applications. For example, in enterprise settings, every employee can upload encrypted data on the cloud storage server without the knowledge of the company's master secret key. In Cloud Computing, outsourced data might not only be accessed but also updated frequently by users for various application purposes. Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how our main scheme can be adapted to build upon the existing work to support data dynamics, including block level operations of modification, deletion and insertion. Data dynamics support is achieved by replacing the index information i with the m_i in the computation of block signatures and using the classic data structure-Merkle hash tree (MHT) for the underlying block sequence enforcement.

There are three users, a creator, a reader and a writer. Creator receives a token γ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id (like health/social insurance number), the trustee gives her a token γ . There are multiple KDCs, which can be scattered. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud

System Module for Encrypted Data Sharing :

Proposed encryption scheme consists of five polynomial-time algorithms as follows:

- Setup Phase
- Key Generation Phase
- Encryption Phase
- Extract Phase
- Decryption Phase

The data owner establishes the public system parameter via Setup and generates a public/master-secret key pair via Key Generation. Messages can be encrypted via Encrypt by anyone who also decides what cipher-text class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of cipher-text classes via Extract. The generated keys can be passed to delegates securely (via secure emails or secure devices). Finally, any user with an aggregate key can decrypt any ciphertext provided that the cipher-text's class is contained in the aggregate key via Decrypt.

Suppose the sender wants to share his data m_1, m_2, \dots, m_i on the server. First Setup $(1^\lambda, n)$ is performed to get param and execute KeyGeneration phase to get the public/master-secret key pair (PK; MSK). The system parameter param and public-key PK can be made public and the master-secret key MSK should be kept secret by the sender. Anyone (including sender) can then encrypt each m_i by $CT_i = \text{Encrypt}(PK, M, A)$. The encrypted data is uploaded to the server. With param and PK, people who cooperate with sender can update sender's data on the server. Once the sender is willing to share a set S of his data with the receiver, he can compute the aggregate key KS for receiver by performing Extract(MSK, S). Since KS is just a constant size key, it is easy to be sent to the receiver via a secure email.

Setup Phase:

The setup algorithm $(1^\lambda, n)$ takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK. executed by the data owner to set up an account on an untrusted server. On input a security level parameter 1^λ and the number of cipher-text classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity. Randomly pick a bilinear group G of prime order p where $2^\lambda \leq p \leq 2^{\lambda+1}$, a generator $g \in G$ and $\alpha \in \mathbb{Z}_p$. Compute $g_i = g^{\alpha i} \in G$ for $i = 1, \dots, n, n+2, \dots, 2n$. Output the system parameter as $\text{param} = \langle g, g_1, \dots, g_n, g_{n+2}, g_{2n} \rangle$

Key Generation Phase:

KeyGeneration (MSK,S). The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. Pick $y \in_R Z_p$. It outputs a private key SK. Executed by the data owner, to randomly generate a public/master-secret key pair ($PK = g^y$, $MSK = y$). The sizes of ciphertext, publickey, master-secret key and aggregate key in the proposed scheme are all of constant size. The public system parameter has size linear in the number of ciphertext classes, but only a small part of it is needed each time and it can be fetched on demand from large (but non confidential) cloud storage.

Encryption Phase :

Encrypt (PK, M, A). The encryption algorithm takes as input the public parameters PK, a message $M \in GT$, and an access structure A over the universe of attributes. The algorithm will encrypt M and produce a ciphertext CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains A. It is executed by anyone who wants to encrypt data. Users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further categorized into different classes.

Extract Phase:

Extract ($MSK = y$, S). It is executed by the data owner for delegating the decrypting power for a certain set of cipher-text classes to a delegate. On input the master secret key MSK and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by K_S where $K_S = \prod_{j \in S} g^{y_{n+1-j}}$. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of cipher-text classes.

Decryption Phase:

The decryption algorithm takes as input the public parameters PK, a cipher-text CT, which contains an access policy A, and an aggregate-key K_S , which is generated by Extract. If the set S of attributes satisfies the access structure A then the algorithm will decrypt the cipher-text and return a message M.

A New Architecture:

We aim at removing the security threats for cloud architecture by using two encrypting techniques: Diffie Hellmann Key Exchange and Elliptic Curve Cryptography.

ELLIPTIC CURVE CRYPTOGRAPHY:

Overview:

Elliptic Curve Cryptography (ECC) was proposed by Koblitz and Miller in the 1980s. ECC is a public key cryptographic scheme. It uses properties of Elliptic Curves to develop cryptographic algorithms. Security of ECC is based on the intractability of ECDLP i.e. Elliptic Curve Discrete Logarithm Problem. Elliptic Curve Cryptography is defined with the following parameters as:

$$P = (q, FR, a, b, c, G, n, h)$$

q: the prime number or 2^m that defines the curve's form.

FR: field representation.

a, b: the curve coefficients.

G: the base point (G_x, G_y).

n: the order of G. It must be a big prime number.

h: cofactor co-efficient.

Elliptic Curves (EC) over finite fields are used to implement public-key protocols. The Elliptic curve is defined on either prime field $GF(p)$ or binary field $GF(2^n)$. Since arithmetic in the latter field is much faster, we work in $GF(2^n)$. An elliptic curve E is defined by the simplified projective coordinates as follow:

$$Y^2Z + XYZ = X^3 + aX^2Z + bZ^3$$

This public key cryptography scheme is defined over two fields:

1) Prime Galois Field, $GF(p)$:

The equation of Elliptic Curve is: $Y^2 \bmod p = x^3 + ax + b \bmod p$

Where $4a^3 + 27b^2 \bmod p \neq 0$ with elements of $GF(p)$ as integers between 0 and $p-1$.

2) Binary extension Galois Field, $GF(2^m)$:

The equation of Elliptic Curve is: $y^2 + xy = x^2 + ax^2 + b$, where $b \neq 0$.

Discrete Logarithm Problem (DLP):

Elliptic curve system is based on DLP. A group structure given by elliptic curves over finite fields is used to implement these schemes. Group elements are some rational points lying curve. They have a special point called point at infinity. The group operation is addition of points. It is carried out by arithmetic operations in finite fields. Major building block of ECC is scalar point multiplication. We take a point P and add it

to itself. This operation is performed some n no of times to get the resulting point Q. Number of times P is added is called k. To obtain k from Q and P is called the Elliptic Curve Discrete Logarithm Problem (ECDLP).

Advantages:

Till date, there is no sub exponential-time algorithm to solve ECDLP in selected elliptic curve groups . Hence, cryptosystems that rely on ECDLP provide high strength-perbit. This makes ECC work on smaller key sizes. It requires less memory than other DLP-based systems. The general key size for ECC is around 163 bits, providing the same security level as 1024 key bits of RSA. This makes ECC's very attractive for implementations in areas where we have memory limitations and computational overhead is a concern

Diffie Hellmann Key Exchange:

Diffie-Hellman key exchange protocol is the first public key cryptography scheme. It was proposed by Witfield Diffie and Martin Hellman in 1976 . It uses two keys -- one secret and other private key. If Sender wants to communicate with the receiver, he encrypts the message with his private key and senders' public key. On the receiving end, the receiver decrypts the sent message using his private key and sender's public key. This scheme is based on the difficulty of computing logarithmic functions for prime exponents. This is known as the Discrete Logarithm Problem (DLP).Algorithm is as follow:

1) Select two Global Public Elements:

A prime number p and an integer α that is a primitive root of p .

2) Sender Key Generation:

Sender selects a random integer $X_A < p$ which is private and computes $Y_A = \alpha^{X_A} \bmod p$, which is public.

3) Receiver Key Generation:

Receiver selects a random integer $X_B < p$ which is private and computes $Y_B = \alpha^{X_B} \bmod p$, which is public.

4) Sender calculates secret key:

$$K = (Y_B)^{X_A} \bmod p$$

5) Receiver calculates secret key which is identical to sender secret key. $K = (Y_A)^{X_B} \bmod p$.

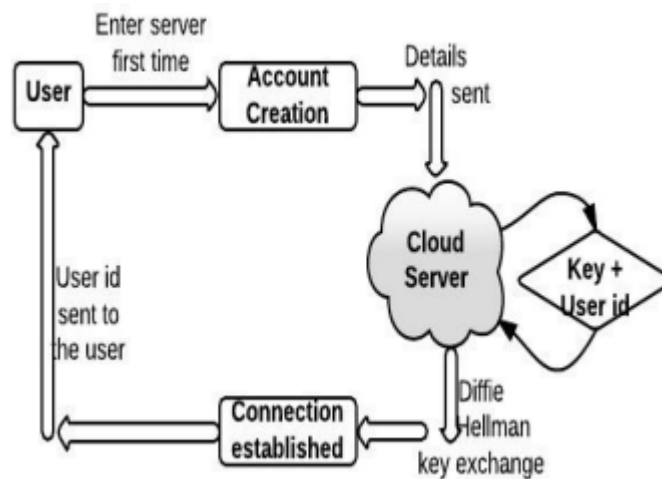
To deploy these two methods, we have proposed a new architecture which can be used to design a cloud system for better security and reliability on the cloud servers at the same time maintaining the data integrity from user point of view. Our system involves following steps:

1. Establishment of connection:

As soon as the user logs in our system for the first time, he is asked to make an account in the system. The initial connection is established with the help of HTTPS and SSL protocols.

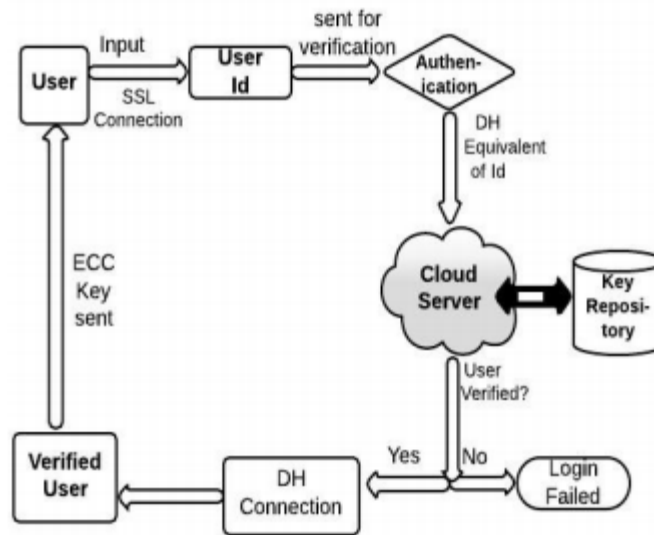
2. Account Creation:

For the first time when a secured connection is formed, the user is asked to fill in the account details required for account creation in our cloud system. These details are sent over the internet to our server. The account is created in the system. Further, the connection is then established by Diffie Hellmann Key Exchange protocol. The server also generates the user id which acts as a unique user identifier, its Diffie Hellman equivalent stream, required private and public key for ECC encryption. The user id is sent to the user over the secured channel. User is asked to keep this id as a secret because it is used as a tool to authenticate him every time he logs on to the system.



3. Authentication:

As soon as the user opens the home page of the cloud server, SSL connection is established. As the account is created, the user is asked to authenticate himself giving all the necessary details and the secret user id sent to him earlier. The cloud server checks the validity of the user by first finding out the Diffie Hellman equivalent of the user id from the server repository. If the key matches, then the connection is established by this protocol again and the user is logged in to the server. At the back end of the user, its private key and the ECC algorithm is sent for encryption.



4. Data Exchange:

The data exchange here includes 2 steps:

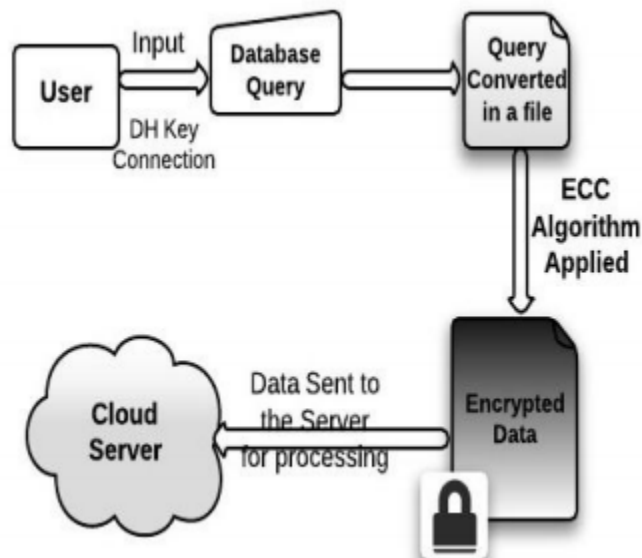
A. The client side:

The client wants to fetch data from a server repository; his query is converted in a form of file and encrypted using his public key. This encrypted data is then sent to the client for processing.

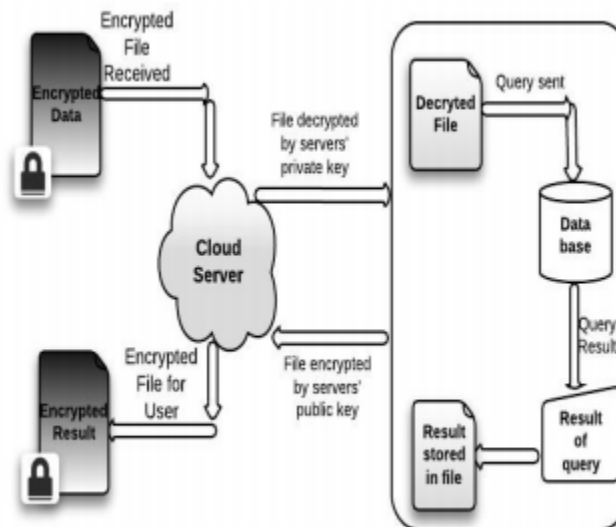
B. The server side:

The server receives the encrypted data. It decrypts it using the private key and processes user queries. The result obtained is encrypted again and sent to the client side.

Data processing view of client:



Data processing view of Server:



Computation of key for cryptography:

The key generation in this architecture takes place at two levels: one for ECC and other for Diffie Hellman.

1. For ECC:

The public key is a point on the curve. Private key is a random number. The public key is generated by multiplying the private key with generator point G . This point generation and other factors are discussed below.

A. Computation of Point on the Curve :

ECC algorithm has the ability to compute a new point on the curve given the product points. We encrypt this point as information to be exchanged between the end users.

B. Choice of Field:

To analyse algorithms with smaller computations, we use polynomial time algorithms and for complex computations can be evaluated with exponential time algorithms. The equation of an elliptic curve is given as,

$$y^2 = x^3 + ax + b$$

C. Integer Factorization:

Given an integer n which is the product of two large primes' p and q , we have:

$$y^2 = x^3 + ax + b$$

It is easy to calculate n for given p and q . It is computationally infeasible to determine p and q for large values of n . Its security depends on the difficulty of factoring the large prime numbers. The method used to solve Integer Factorization problem is the Number Field Sieve Which is a sub exponential algorithm.

D. Key Generation:

Key generation is an important part. An algorithm should generate both public and private keys. The sender will encrypt the message data with the receiver's public key and the receiver will decrypt with its private key. Select a number, d in the range of n . We generate the public key using following equation,

$$Q = d * P$$

d = the random number in range of $(1 \text{ to } n-1)$.

P = a point on curve.

Q = public key.

d = private key

E. Encryption:

Let ' m ' be a message to be sent. Consider ' m ' has point ' M ' on the curve ' E '. Randomly select a value ' k ' from $[1 - (n-1)]$. Two cipher texts will be generated, let it be $B1$ and $B2$.

$$B1 = k * P$$

$$B2 = M + (k * P)$$

F. Decryption:

Use the following equation to obtain the original message that was sent i.e ' m '.

$$M = B2 - d * B$$

M is the original data that was sent.

2. Diffie Hellman Key Exchange:

This protocol is one of the pioneers in the birth of public key cryptography. It follows the following steps.

Input:

G is an abelian group;

$g \in G$, m is prime multiplicative order.

Output:

A secret $s \in G$ which will be shared by both sides.

Steps:

Sender generates random $d_A \in \{2, \dots, m-1\}$ and compute $e_A = g^{d_A}$

Sender sends e_A to the receiver.

Receiver generates a random $d_B \in \{2, \dots, m-1\}$ and computes $e_B = g^{d_B}$

Receiver sends e_B to receiver.

Sender calculates $s = (e_B)_A^d = g_{A \ B}^{d \ d}$

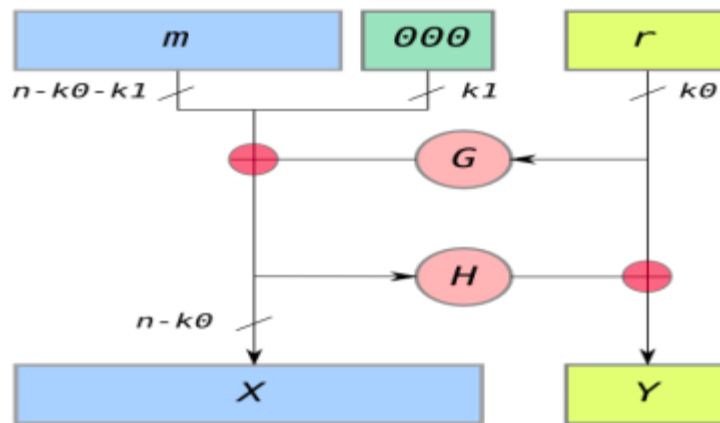
Receiver calculates $s = (e_A)_B^d = g_{A \ B}^{d \ d}$.

Conclusion and future work:

Efficient search on encrypted data is also an important concern in clouds. However, security concerns have become the biggest obstacle to cloud adoption of cloud because all information and data (including reallocation of data, and security management level) are completely under the control of cloud service providers. This paper presented an encrypted data sharing methodology for cloud environments which is decentralized and prevents many attacks. Key Distribution is done in a decentralized way. Public-key based on encryption is used. In cloud storage, the number of cipher-texts usually grows rapidly. So we have to reserve enough cipher-text classes for the future extension. Proposed technique does not authenticate users, who want to remain anonymous while accessing the cloud. In the future, work can be done on distributed and scalable Big Data sharing methodology with anonymous authentication on clouds.

OAEP (OPTIMAL ASYMMETRIC ENCRYPTION PADDING)

OAEP was introduced by Bellare and Rogaway. Optimal Asymmetric Encryption Padding (OAEP) is a padding scheme often used together with RSA encryption. The OAEP algorithm is a form of Feistel network which uses a pair of random oracles G and H to process the plaintext prior to asymmetric encryption. When combined with any secure trapdoor one-way permutation, this processing is proved in the random oracle model to result in a combined scheme which is semantically secure under chosen plaintext attack. When implemented with certain trapdoor permutations (e.g., RSA), OAEP is also proved secure against chosen ciphertext attack. OAEP can be used to build an all-or-nothing transform.



In the diagram,

- n is the number of bits in the RSA modulus.
- k_0 and k_1 are integers fixed by the protocol.
- m is the plaintext message, an $(n - k_0 - k_1)$ -bit string
- G and H are cryptographic hash functions fixed by the protocol

Encoding:

1. messages are padded with k_1 zeros to be $n - k_0$ bits in length.
2. r is a random k_0 -bit string
3. G expands the k_0 bits of r to $n - k_0$ bits.
4. $X = m000...0 \oplus G(r)$
5. H reduces the $n - k_0$ bits of X to k_0 bits
6. $Y = r \oplus H(X)$
7. The output is $X || Y$ where X is shown in the diagram as the leftmost block and Y as the rightmost block

Decoding:

1. recover the random string as $r = Y \oplus H(X)$
2. recover the message as $m00...0 = X \oplus G(r)$

Two main goals of OAEP:

1. Add an element of randomness which can be used to convert a deterministic encryption scheme (e.g., traditional RSA) into a probabilistic scheme.
2. Prevent partial decryption of cipher texts (or other information leakage) by ensuring that an adversary cannot recover any portion of the plaintext without being able to invert the trapdoor one-way permutation.

Review on Network Security and Cryptography

Abstract:

With the advent of the World Wide Web and the emergence of e-commerce applications and social networks, organizations across the world generate a large amount of data daily. Data security is the utmost critical issue in ensuring safe transmission of information through the internet. Also network security issues are now becoming important as society is moving towards the digital information age. As more and more users connect to the internet it attracts a lot of cyber-criminals. It comprises authorization of access to information in a network, controlled by the network administrator. The task of network security not only requires ensuring the security of end systems but of the entire network.

Introduction:

Internet has become more and more widespread. If an unauthorized person is able to get access to this network, he can not only spy on us but he can easily mess up our lives. Network Security & Cryptography is a concept to protect network and data transmission over wireless networks. A network security system typically relies on layers of protection and consists of multiple components including networking monitoring and security software in addition to hardware and appliances. All components work together to increase the overall security of the computer network. Security of data can be done by a technique called cryptography. So one can say that cryptography is an emerging technology, which is important for network security.

Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

Cryptography is the science of writing in secret code. More generally, it is about constructing and analyzing protocols that block adversaries; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography. Modern cryptography exists at the

intersection of the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. The development of the World Wide Web resulted in broad use of cryptography for ecommerce and business applications. Cryptography is closely related to the disciplines of cryptology and cryptanalysis. Techniques used for decrypting a message without any knowledge of the encryption details fall into the area of cryptanalysis. Cryptanalysis is what the layperson calls “breaking the code.” The areas of cryptography and cryptanalysis together are called cryptology. Encryption is the process of converting ordinary information (called plaintext) into unintelligible text (called ciphertext). Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. Cryptosystem is the ordered list of elements of finite possible plaintexts, finite possible ciphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key.

The challenging problem is how to effectively share encrypted data. Encrypt messages with a strongly secure key which is known only by sender and recipient end is a significant aspect to acquire robust security in a sensor network. The secure exchange of key between sender and receiver is too difficult a task in resource constrained sensor networks. Data should be encrypted first by users before it is outsourced to a remote cloud storage service and both data security and data access privacy should be protected such that cloud storage service providers have no abilities to decrypt the data, and when the user wants to search some parts of the whole data, the cloud storage system will provide the accessibility without knowing what the portion of the encrypted data returned to the user is about.

Types of Security Attacks:

1. Passive Attacks:

This type of attack includes observation or monitoring of communication. A passive attack attempts to learn or make use of information from the system but does not affect system resources. The goal of the opponent is to obtain information that is being transmitted. Types of passive attacks:

- **Traffic Analysis:**

The message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.

- **Release of Message Contents:**

Read contents of message from sender to receiver.

2. Active Attacks:

An active attack attempts to alter system resources or affect their operation. It involves some modification of the data stream or the creation of a false stream. Types of active attacks:

- **Modification of Messages:**

Some portion of a legitimate message is altered, or that messages are delayed or reordered.

- **Denial of Service:**

An entity may suppress all messages directed to a particular destination

- **Replay:**

It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

- **Masquerade:** It takes place when one entity pretends to be a different entity.

Security Services:

It is a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. It enhances the security of data processing and transferring.

1. Data Integrity:

It can apply to a stream of messages, a single message, or selected fields within a message. A loss of integrity is the unauthorized modification or destruction of information.

2. Data Confidentiality:

Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.

3. Authenticity:

Provide authentication to all the nodes and base stations for utilizing the available limited resources. It also ensures that only the authorized node can participate in the communication.

4. Nonrepudiation:

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

5. Access Control:

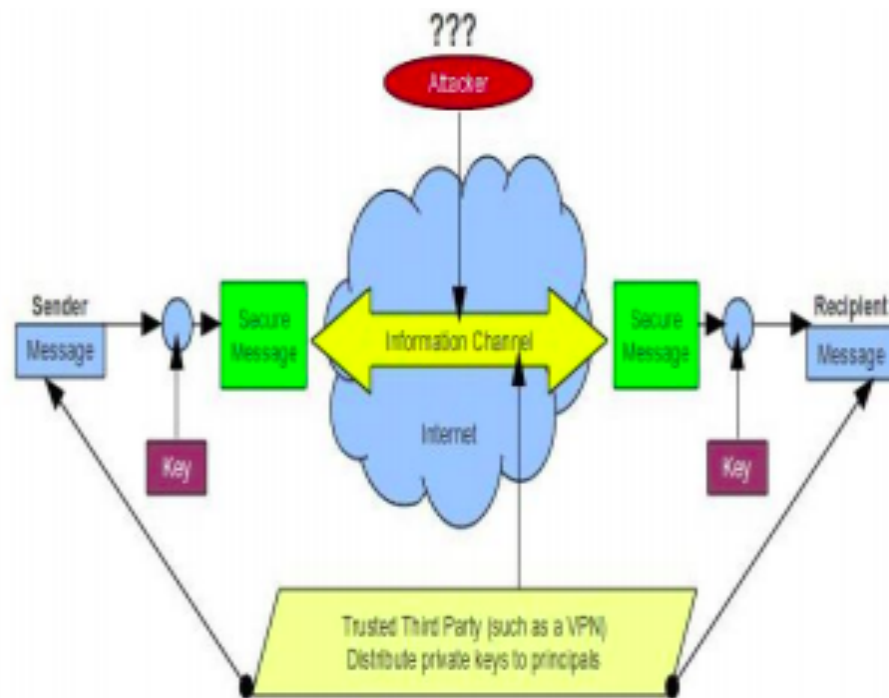
Access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to

gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

Network Security Model:

A message is to be transferred from one party to another across some sort of Internet service. A third party may be responsible for distributing the secret information to the sender and receiver while keeping it from any opponent. Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

- A security-related transformation on the information to be sent. Messages should be encrypted by key so that it is unreadable by the opponent.
- An encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.



Need for Key Management in Cloud:

Encryption provides data protection while key management enables access to protected data. It is strongly recommended to encrypt data in transit over networks, at rest, and on backup media. In particular, data to encrypt their own data. Both encryption and key management are very important to help secure applications and data stored in the Cloud. Requirements of effective key management are :

- **Secure key stores:** The key stores themselves must be protected from malicious users. If a malicious user gains access to the keys, they will then be able to access any encrypted data the key is corresponded to. Hence the key stores themselves must be protected in storage, in transit and on backup media.

- **Access to key stores:** Access to the key stores should be limited to the users that have the rights to access data. Separation of roles should be used to help control access. The entity that uses a given key should not be the entity that stores the key.

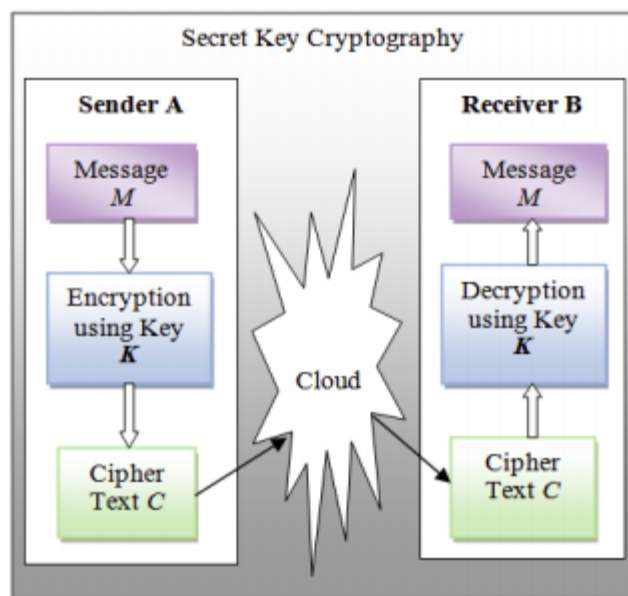
- **Key backup and recoverability:** Keys need secure backup and recovery solutions. Loss of keys, although effective for destroying access to data, can be highly devastating to a business and Cloud providers need to ensure that keys aren't lost through backup and recovery mechanisms.

Cryptography Mechanism:

Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. The term is most often associated with scrambling plaintext messages (ordinary text, sometimes referred to as cleartext) into ciphertext (a process called encryption), then back again (known as decryption). There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions.

1. Secret Key Cryptography:

With secret key cryptography, a single key is used for both encryption and decryption. The sender A uses the key K (or some set of rules) to encrypt the plaintext message M and sends the ciphertext C to the receiver. The receiver applies the same key K (or ruleset) to decrypt the cipher text C and recover the plaintext message M. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption.



With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key.

Secret key cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing. A block cipher is so-called because the scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher. Block ciphers can operate in one of several modes; the following four are the most important:

- **Electronic Codebook (ECB):**

This mode is the simplest, most obvious application: the secret key is used to encrypt the plaintext block to form a ciphertext block. Two identical plaintext blocks, then, will always generate the same ciphertext block. Although this is the most common mode of block ciphers, it is susceptible to a variety of brute-force attacks.

- **Cipher Block Chaining (CBC):**

This mode adds a feedback mechanism to the encryption scheme. In CBC, the plaintext is exclusively-ORed (XORed) with the previous ciphertext block prior to encryption. In this mode, two identical blocks of plaintext never encrypt to the same ciphertext.

- **Cipher Feedback (CFB):**

This mode is a block cipher implementation as a self-synchronizing stream cipher. CFB mode allows data to be encrypted in units

smaller than the block size, which might be useful in some applications such as encrypting interactive terminal input. If we were using 1-byte CFB mode, for example, each incoming character is placed into a shift register the same size as the block, encrypted, and the block transmitted. At the receiving side, the ciphertext is decrypted and the extra bits in the block (i.e., everything above and beyond the one byte) are discarded.

- **Output Feedback (OFB):**

This mode is a block cipher implementation conceptually similar to a synchronous stream cipher. OFB prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that is independent of both the plaintext and ciphertext bitstreams.

Stream ciphers come in several flavors but two are worth mentioning here. Self-synchronizing stream ciphers calculate each bit in the keystream as a function of the previous n bits in the keystream. It is termed "self synchronizing" because the decryption process can stay synchronized with the encryption process merely by knowing how far into the n -bit keystream it is. One problem is error propagation; a garbled bit in transmission will result in n garbled bits at the receiving side. Synchronous stream ciphers generate the keystream in a fashion independent of the message stream but by using the same keystream generation function at sender and receiver. While stream ciphers do not propagate transmission errors, they are, by their nature, periodic so that the keystream will eventually repeat.

Secret key cryptography algorithms that are in use today include:

- **Camellia:**

A secret-key, block-cipher crypto algorithm developed jointly by Nippon Telegraph and Telephone (NTT) Corp. and Mitsubishi Electric Corporation (MEC) in 2000. C has some characteristics in common with AES: a 128-bit block size, support for 128-, 192-, and 256-bit key lengths, and suitability for both software and hardware implementations on common 32-bit processors as well as 8-bit processors (e.g., smart cards, cryptographic hardware, and embedded systems). Camellia is a Feistel cipher with either 18 rounds (when using 128-bit keys) or 24 rounds (when using 192 or 256-bit keys). Every six rounds, a logical transformation layer is applied: the so-called "FLfunction" or its inverse. Camellia uses four 8 x 8-bit S-boxes with input and output affine transformations and logical operations. The cipher also uses input and output key whitening. The diffusion layer uses a linear transformation based on a matrix with a branch number of 5.

- **KASUMI:**

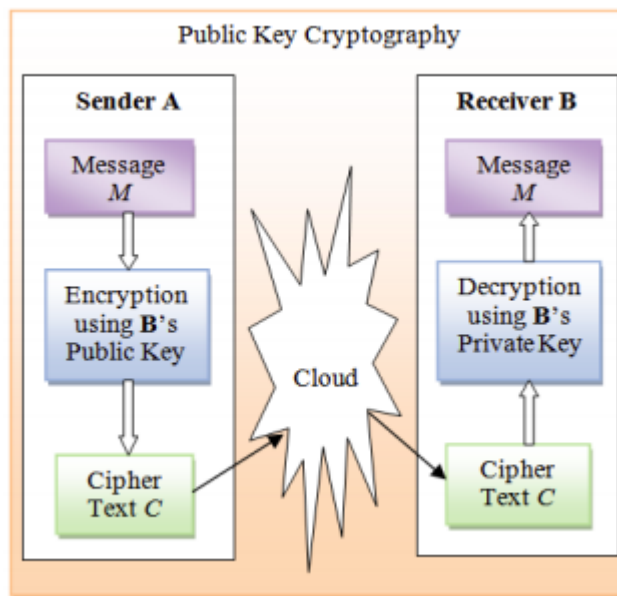
A block cipher using a 128-bit key and block size 64-bit, is part of the Third-Generation Partnership Project (3gpp), formerly known as the Universal Mobile Telecommunications System (UMTS). KASUMI is the intended

confidentiality and integrity algorithm for both message content and signaling data for emerging mobile communications systems. KASUMI is used in the A5/3 key stream generator and in GPRS in the GEA3 key stream generator. In 2010, Dunkelman, Keller and Shamir published a new attack that allows an adversary to recover a full A5/3 key by related-key attack . The core of KASUMI is an eight-round Feistel network. The round functions in the main Feistel network are irreversible Feistel-like network transformations. In each round the round function uses a round key which consists of eight 16-bit sub keys derived from the original 128-bit key using a fixed key schedule.

DES , AES , Twofish and Blowfish discussed beforehand also belong to secret key cryptography.

2.Public-Key Cryptography:

Public-key cryptography is a form of cryptosystem in which encryption and decryption are performed using the different keys—one a public key and one a private key. These keys are mathematically related although knowledge of one key does not allow someone to easily determine the other key. The sender A uses the public key of receiver B (or some set of rules) to encrypt the plaintext message M and sends the ciphertext C to the receiver. The receiver applies its own private key (or ruleset) to decrypt the cipher text C and recover the plaintext message M . Because a pair of keys is required, this approach is also called asymmetric cryptography. Asymmetric encryption can be used for confidentiality, authentication, or both.



Public-key cryptography algorithms that are in use today for key exchange or digital signatures include:

1. RSA :

The first, and still most common, public key cryptography implementation, named for the three MIT mathematicians who developed it — Ronald Rivest, Adi Shamir, and Leonard Adleman . RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n , that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors. RSA has three phases:

- **Key Generation:**

Receiver generates a public/private key pair. Algorithm is as follow:

- 1) Select p, q such that p and q both are prime, $p \neq q$
- 2) Calculate $n = p * q$
- 3) Calculate $f(n) = (p - 1)(q - 1)$
- 4) Select integer e such that $\gcd(f(n), e) = 1$; $1 < e < f(n)$
- 5) Calculate d such that $d \equiv e^{-1} \pmod{f(n)}$
- 6) Public key PUK= (e, n)
- 7) Private key PRK= (d, n)

- **Encryption:**

Encryption is done by the sender with the receiver's Public Key.

Algorithm is as follow:

- 1) Plain Text M is known, $M < n$
- 2) Cipher Text C is calculated as $C = M^e \pmod{n}$

- **Decryption:**

Decryption is done by the receiver using his Private Key. Algorithm is as follow:

- 1) Cipher Text C is known
- 2) Plain Text M is calculated as $M = C^d \pmod{n}$

2.Digital Signature Standard:

The digital signature standard (DSS) is an NIST standard that uses the secure hash algorithm (SHA) .A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. Typically the signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

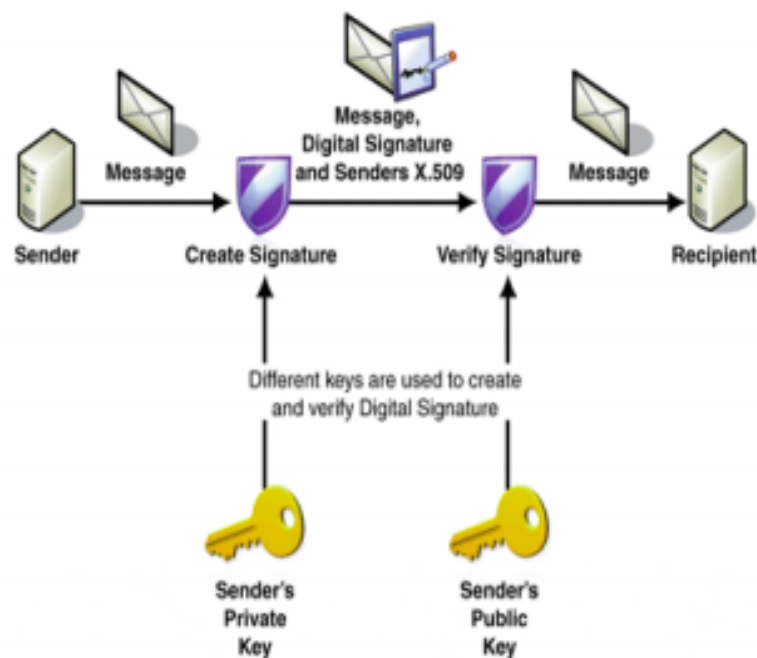
Sender can sign a message using a digital signature generation algorithm. The inputs to the algorithm are the message and sender's private key. Any

other user, say receiver, can verify the signature using a verification algorithm, whose inputs are the message, the signature, and the sender's public key.

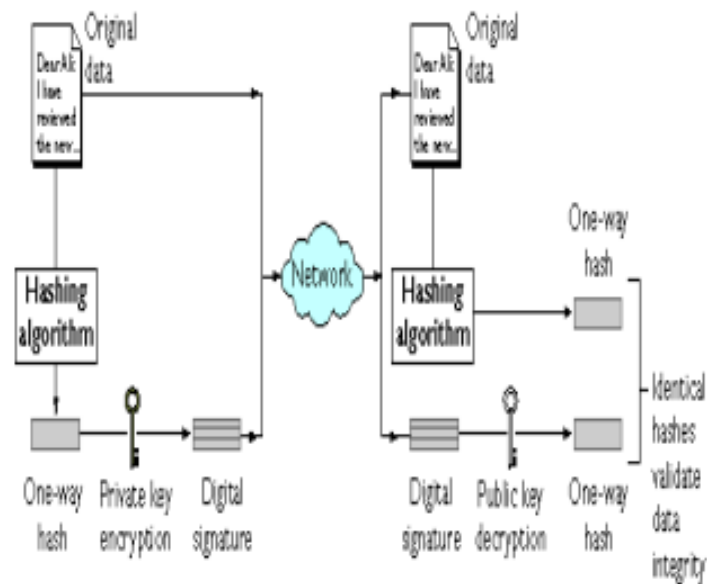
The DSS uses an algorithm that is designed to provide only the digital signature function. It cannot be used for encryption or key exchange. Nevertheless, it is a publickey technique. In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted. The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid. Because only the sender knows the private key, only the sender could have produced a valid signature.

The DSS approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number generated for this particular signature. The signature function also depends on the sender's private key PR_a and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key PU_g . The result is a signature consisting of two components, labeled s and r .

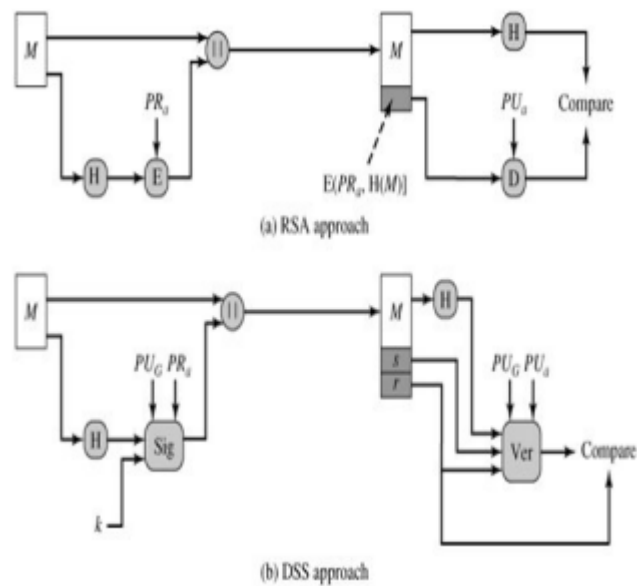
Digital Signature Without Hash Function:



Digital Signature With Hash Function:



Digital Signature Approaches:



Elliptic curves and Diffie-Hellman Key exchange discussed above are also part of Public-Key Cryptosystems

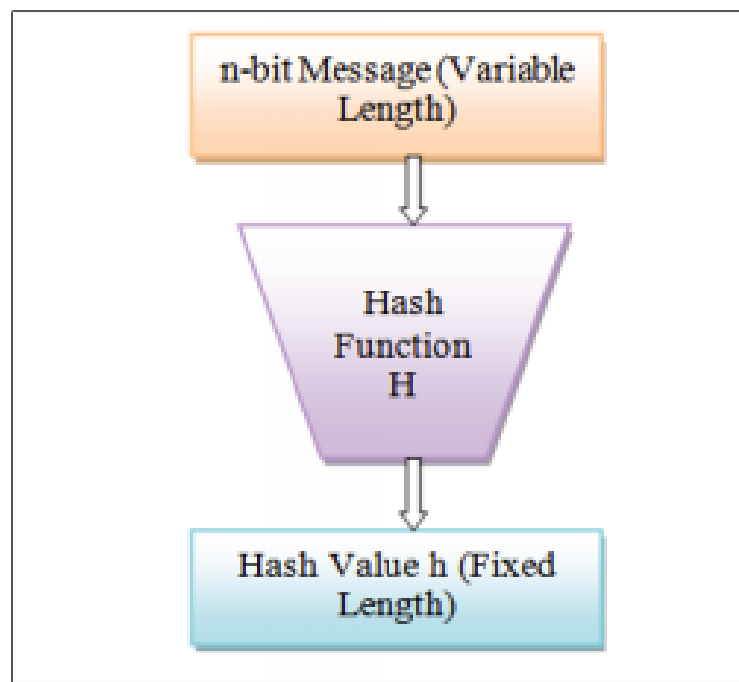
Applications for Public-Key Cryptosystems:

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
Elliptic Curve	Yes	Yes	Yes
DSS	No	Yes	No

Hash Functions:

Hash functions, also called message digests and one way encryption, are algorithms that, in some sense, use no key. A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value $h = H(M)$. In general terms, the principal object of a hash function is data integrity. A change to any bit or bits in results, with high probability, in a change to the hash code. Virtually all cryptographic hash functions involve the iterative use of a compression function. The compression function used in secure hash algorithms falls into one of two categories: a function specifically designed for the hash function or an algorithm based on a symmetric block cipher. SHA and Whirlpool [19] are examples of these two approaches, respectively.

The hash algorithm involves repeated use of a compression function, f , that takes two inputs (a n -bit input from the previous step, called the chaining variable, and a b -bit block) and produces a n -bit output. At the start of hashing, the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value. It is seen that $b > n$. A cryptographic hash function can be used to construct a pseudorandom function (PRF) or a pseudorandom number generator (PRNG).



Secure Hash Algorithm (SHA) is a family of cryptographic hash functions.

Algorithm	Message Digest Size	Message Size	Block Size	Word Size	No of Step
SHA-1	160	$< 2^{64}$	512	32	80
SHA-224	224	$< 2^{64}$	512	32	64
SHA-256	256	$< 2^{64}$	512	32	64
SHA-384	384	$< 2^{128}$	1024	64	80
SHA-512	512	$< 2^{128}$	1024	64	80

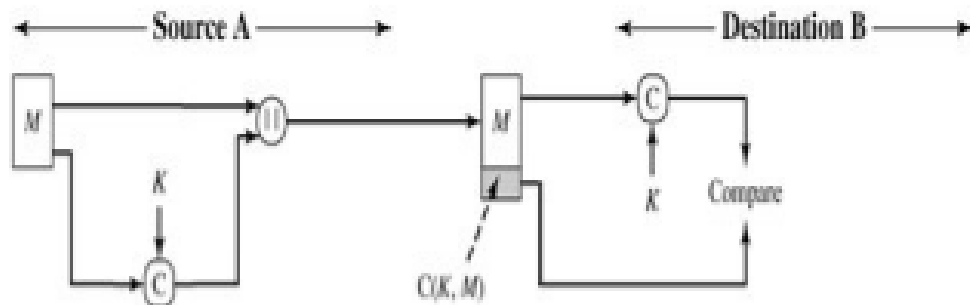
All sizes are measured in bits.

Message Authentication Code:

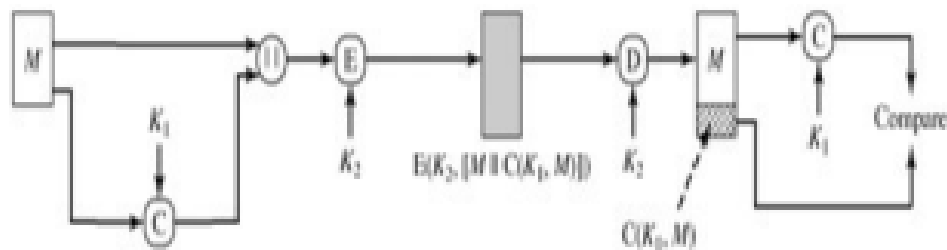
Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that data received are exactly as sent (i.e., contain no modification, insertion, deletion, or replay). In many cases, there is a requirement that the authentication mechanism assures that the purported identity of the sender is valid. When a hash function is used to provide message authentication, the hash function value is often referred to as a message digest. More commonly, message authentication is achieved using a message authentication code (MAC), also known as a keyed hash function or cryptographic checksum. Typically, MACs are used

between two parties, say sender and receiver, that share a secret key K to authenticate information exchanged between those parties. A MAC function C takes as input a secret key K and a variable-length data block or message M and produces a fixed-length hash value MAC, referred to as the message authentication Code. This can then be transmitted with or stored with the protected message. If the integrity of the message needs to be checked, the MAC function can be applied to the message and the result compared with the stored MAC value.

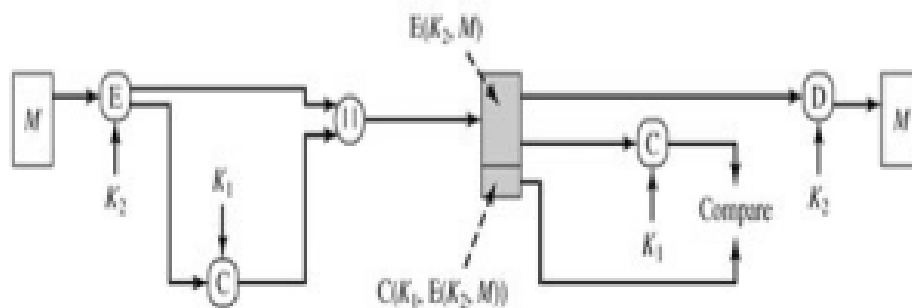
$$\text{MAC} = C(K, M)$$



(a) Message authentication



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

HMAC:

Hash-based message authentication code (HMAC) is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authentication of a message. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and on the size and quality of the key.

Hash-based message authentication code (HMAC) provides the server and the client each with a public and private key. The public key is known, but the private key is known only to that specific server and that specific client. The client creates a unique HMAC, or hash, per request to the server by combining the request data and hashing that data, along with a private key and sending it as part of a request. The server receives the request and regenerates its own unique HMAC. The server compares the two HMACs, and, if they're equal, the client is trusted and the request is executed. This process is often called a secret handshake.

HMAC can be expressed as:

$$\text{HMAC}(K,M) = H[K^+ \oplus \text{opad}] \parallel H[K^+ \oplus \text{ipad}] \parallel M$$

Where,

K = secret key; recommended length is $\geq n$; if key length is greater than b -bit block, the key is input to the hash function to produce an n -bit key

M = message input to HMAC,

H = cryptographic hash function,

K^+ = K padded with zeros on the left so that the result is b bits in length

\parallel = concatenation

opad = 01011100 (5C in hexadecimal) repeated $b/8$ times,

ipad = 00110110 (36 in hexadecimal) repeated $b/8$ times.

CMAC:

Cipher-based message authentication codes (CMACs) are a tool for calculating message authentication codes using a block cipher coupled with a secret key. CMAC can be used to verify both the integrity and authenticity of a message. This mode of operation fixes security deficiencies of CBC-MAC (CBC-MAC is secure only for fixed-length messages). To generate an l -bit CMAC tag (t) of a message (m) using a b -bit block cipher (E) and a secret key (k), one first generates two b -bit subkeys (k_1 and k_2).

Sub-keys Algorithm:

1) Calculate a temporary value $k_0 = E_k(0)$.

- 2) If $\text{msb}(k_0) = 0$, then $k_1 = k_0 \ll 1$, else $k_1 = (k_0 \ll 1) \oplus C$; where C is a certain constant that depends only on b . (Specifically, C is the non-leading coefficient of the lexicographically first irreducible degree- b binary polynomial with the minimal number of ones.)
- 3) If $\text{msb}(k_1) = 0$, then $k_2 = k_1 \ll 1$, else $k_2 = (k_1 \ll 1) \oplus C$.
- 4) Return keys (k_1, k_2) for the MAC generation process.

CMAC Tag Generation Algorithm:

- 1) Divide message into b -bit blocks $m = m_1 \parallel \dots \parallel m_{n-1} \parallel m_n'$ where m_1, \dots, m_{n-1} are complete blocks. (The empty message is treated as 1 incomplete block.)
- 2) If m_n' is a complete block then $m_n = k_1 \oplus m_n'$ else $m_n = k_2 \oplus (m_n' \parallel 10\dots 0_2)$.
- 3) Let $c_0 = 00\dots 0_2$.
- 4) For $i = 1, \dots, n$, calculate $c_i = E_k(c_{i-1} \oplus m_i)$.
- 5) Output $t = \text{msbl}(c_n)$.

Network and Internet Security:

Internet security is a tree branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing. Different methods have been used to protect the transfer of data, including encryption.

Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority.

Types of Network Security:

1. Wireless Network Security:

Wireless security is the prevention of unauthorized access or damage to computers using wireless networks. The most common types of wireless security are Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). WEP is a notoriously weak security standard. The password it uses can often be cracked in a few minutes with a basic laptop computer and widely available software tools. WAP security is primarily provided by the Wireless Transport Layer Security (WTLS), which provides security services between the mobile device (client) and the WAP gateway to the Internet. There are several approaches to WAP end-to-end security. One notable approach assumes that the mobile device implements TLS over TCP/IP and the wireless network supports transfer of IP packets. The WAP architecture is designed to

cope with the two principal limitations of wireless Web access: the limitations of the mobile node (small screen size, limited input capability) and the low data rates of wireless digital networks. Two important WTLS concepts are the secure session and the secure connection, which are defined in the specification as:

a) Secure connection:

A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session. Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice.

b) Secure session:

An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection. There are a number of states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states.

2. IP Security:

Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to network), or between a security gateway and a host (network-to-host). IPsec is said to be especially useful for implementing virtual private networks and for remote user access through dial-up connection to private networks. A big advantage of IPsec is that security arrangements can be handled without requiring changes to individual user computers.

IPsec provides two choices of security service: Authentication Header (AH), which essentially allows authentication of the sender of data, and Encapsulating Security Payload (ESP), which supports both authentication of the sender and encryption of data as well. The specific information associated with each of these services is inserted into the packet in a header that follows the IP packet header. Separate key protocols can be selected, such as the ISAKMP/Oakley protocol. IPsec uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.

IPsec protects any application traffic over an IP network. Applications can be automatically secured by IPsec at the IP layer.

Modes of Operation:

IPsec can be implemented in a host-to-host transport mode, as well as in a network tunneling mode.

Transport mode:

Only the payload of the IP packet is usually encrypted and/or authenticated. The routing is intact, since the IP header is neither modified nor encrypted; however, when the authentication header is used, the IP addresses cannot be translated, as this will invalidate the hash value. The transport and application layers are always secured by hash, so they cannot be modified in any way (for example by translating the port numbers).

Tunnel mode:

The entire IP packet is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new IP header. It is used to create virtual private networks for network-to-network communications (e.g. between routers to link sites), host-to-network communications (e.g. remote user access) and host-to-host communications (e.g. private chat).

3. Electronic Mail Security:

Email is vulnerable to both passive and active attacks. The protection of email from unauthorized access and inspection is known as electronic privacy. In countries with a constitutional guarantee of the secrecy of correspondence, email is equated with letters and thus legally protected from all forms of eavesdropping. With the explosively growing reliance on e-mail, there grows a demand for authentication and confidentiality services. Two schemes stand out as approaches that enjoy widespread use: Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extension S/MIME.

PGP:

It is an open-source, freely available software package for e-mail security. It provides authentication through the use of digital signature, confidentiality through the use of symmetric block encryption, compression using the ZIP algorithm, and e-mail compatibility using the radix-64 encoding scheme. PGP incorporates tools for developing a public-key trust model and public-key certificate management.

S/MIME:

It is an Internet standard approach to e-mail security that incorporates the same functionality as PGP. It is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.

4. Transport-Level Security:

Transport-Level Security (TLS) is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL. Secure Socket Layer (SSL) provides security services between TCP and applications that use TCP. The Internet standard version is called Transport Layer Service (TLS). The TLS Record Format is the same as that of the SSL Record Format. SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code. SSL/TLS includes protocol mechanisms to enable two TCP users to determine the security mechanisms and services they will use. HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. Secure Shell (SSH) provides secure remote logon and other secure client/server facilities. The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use. All types of communication using SSH, such as a terminal session, are supported using separate channels.

Firewalls:

A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction. Firewalls impose restrictions on incoming and outgoing Network packets to and from private networks. Incoming or outgoing traffic must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as choke points (borrowed from the identical military term of a combat-limiting geographical feature). Firewalls can create choke points based on IP source and TCP port number. They can also serve as the platform for IPsec. Using tunnel mode capability, a firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet. A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer.

Characteristics of Firewalls:

Firewalls can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet. It includes following characteristics:

- All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall.

- Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies.
- The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications.

Types of Firewalls:

A firewall may act as a packet filter. It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria. Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets.

a) Packet Filter:

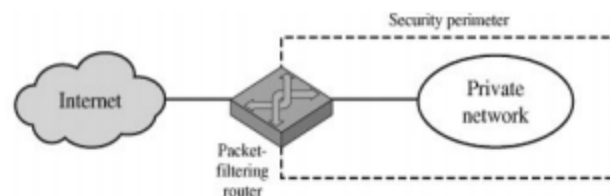
A packet filter is a first generation firewall that processes network traffic on a packet-by-packet basis. Its main job is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet. The router is known as a screening router, which screens packets leaving and entering the network. Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter firewall allows a given application, all functions available within that application will be permitted. Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as network layer address spoofing. Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform.

b) Stateful Packet Inspection:

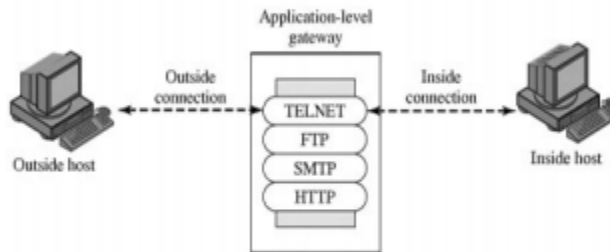
In a stateful firewall the circuit-level gateway is a proxy server that operates at the network level of an Open Systems Interconnection (OSI) model and statically defines what traffic will be allowed. Circuit proxies will forward Network packets (formatted unit of data) containing a given port number, if the port is permitted by the algorithm. The main advantage of a proxy server is its ability to provide Network Address Translation (NAT), which can hide the user's IP address from the Internet, effectively protecting all internal information from the Internet. A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections. Some stateful firewalls also keep track of TCP sequence numbers to prevent attacks that depend on the sequence number, such as session hijacking. Some even inspect limited amounts of application data for some well-known protocols like FTP, IM and SIP commands, in order to identify and track related connections.

c)Application-Level Gateway:

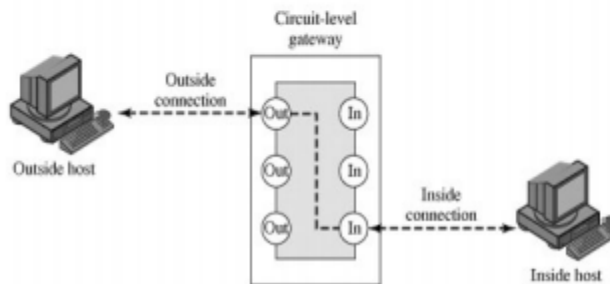
An application-level firewall is a third generation firewall where a proxy server operates at the very top of the OSI model, the IP suite application level. It is also known as application proxy. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. A network packet is forwarded only if a connection is established using a known protocol. Application-level gateways are notable for analyzing entire messages rather than individual packets of data when the data are being sent or received.



(a) Packet-filtering router



(b) Application-level gateway



(c) Circuit-level gateway