Chapter 1

Introduction

1.1 Introduction

In this ages of universal electronics connectivity's, of the viruses and the hackers, of electronic eavesdropping and electronic frauds, there is a need to store the information securely. And this, in turn, led to a heightened awareness to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks.[1]cryptography, the science of encryption, plays a central role in mobile phone communications, pay-tv, e-commerce, sending private emails, transmitting financial information, security of ATM cards, computer passwords, electronic commerce digital signature and touches on many aspects of our daily lives . Cryptography is the art or science encompassing the principles and methods of transforming an intelligible message (plaintext) into one that is unintelligible (cipher text) and then retransforming that message back to its original form .In modern times, cryptography is considered to be a branch of both mathematics and computer science, and is affiliated closely with information theory, computer security, and engineering. Although in the past cryptography referred only to the encryption and decryption of message using secret keys. Nowadays, cryptography generally classified into two categories, the symmetric and asymmetric.

The data transferred from the one system's to another over the public networks would be protected by the methods of the encryption. And on encryption the data is then encrypted/scrambled by any encryptions algorithm using the keys. And only the users which are having the access to the same 'keys' would be able to decrypt/de-scramble the encrypted data. And this method is known as the private key or the symmetric key cryptography. Thus, there are several standard symmetric key algorithms which are defined. Some of the examples are the AES, the 3DES etc. And these standard symmetric algorithms defined are then proven to be highly secured and also time tested. But the problems with these algorithms are that the key exchange. And the communicating parties require the shared secret, 'key', which is to be exchanged between them to have the secured communications. And the security of the symmetric keys algorithms will depends on the secrecy of the keys. And these Keys are

hundreds of bits in the length, and it also depends on the algorithm used. As there can be numbers of intermediates points between the communicating's parties through which the data passes, and these keys cannot exchange online in the secured manner. In the large network, where there are about hundreds of system connected, the offline keys exchanges seems too difficult and even it is unrealistic.

This is where public key cryptography comes to help. Using public key algorithm a shared secret can be established online between communicating parties with out the need for exchanging any secret data.

In public key cryptography each user or the device taking part in the communication have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user/device knows the private key whereas the public key is distributed to all users/devices taking part in the communication. Since the knowledge of public key does not compromise the security of the algorithms, it can be easily exchanged online. In public key cryptography, keys and messages are expressed numerically and the operations are expressed mathematically. The private and public key of a device is related by the mathematical function called the one-way function. One-way functions are mathematical functions in which the forward operation can be done easily but the reverse operation is so difficult that it is practically impossible. In public key cryptography the public key is calculated using private key on the forward operation of the one-way function. Obtaining of private key from the public key is a reverse operation. If the reverse operation can be done easily, that is if the private key is obtained from the public key and other public data, then the public key algorithm for the particular key is cracked. The reverse operation gets difficult as the key size increases. The public key algorithms operate on sufficiently large numbers to make the reverse operation practically impossible and thus make the system secure. For e.g. RSA algorithm operates on large numbers of thousands of bits long.

1.2 Objectives:

There are so many algorithms present to encrypt and decrypt the data for security purpose in cryptography. RSA is the most common algorithm for encryption and decryption. But still the current versions are slow and less secure. The objective of the proposed algorithm is to propose a faster variant of RSA algorithm & also make it more secure against Weiners attack.

Chapter 2

Introduction to cryptography

2.1 What is Cryptography?

Definition: Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography is not the only means of providing information security, but rather one set of techniques.

2.1.1Security Services:

The security services include:

- Data Confidentiality
- Data Integrity
- Authentication
- Non repudiation
- Access Control

a. Data Confidentiality:

Confidentiality has been defined by the International Organization for Standardization (ISO) inISO-17799 as "ensuring that information is accessible only to those authorized to have access" and is one of the cornerstones of information security. Confidentiality is one of the design goals for many cryptosystems, made possible in practice by the techniques of modern cryptography.
It is designed to protect data from disclosure attack. The service as defined by X.800 is very broad and encompasses confidentiality of whole message or part of a message and also protectio Against trafic analysis. That is it is designed to prevent snooping and trafic analysis [2,3]

b. Data Integrity:

Data Integrity is designed for the protection of data from unauthorized modification, insertion, deletion and replaying by an advisory. It can protect the whole message or the part of message.

c. Authentication:

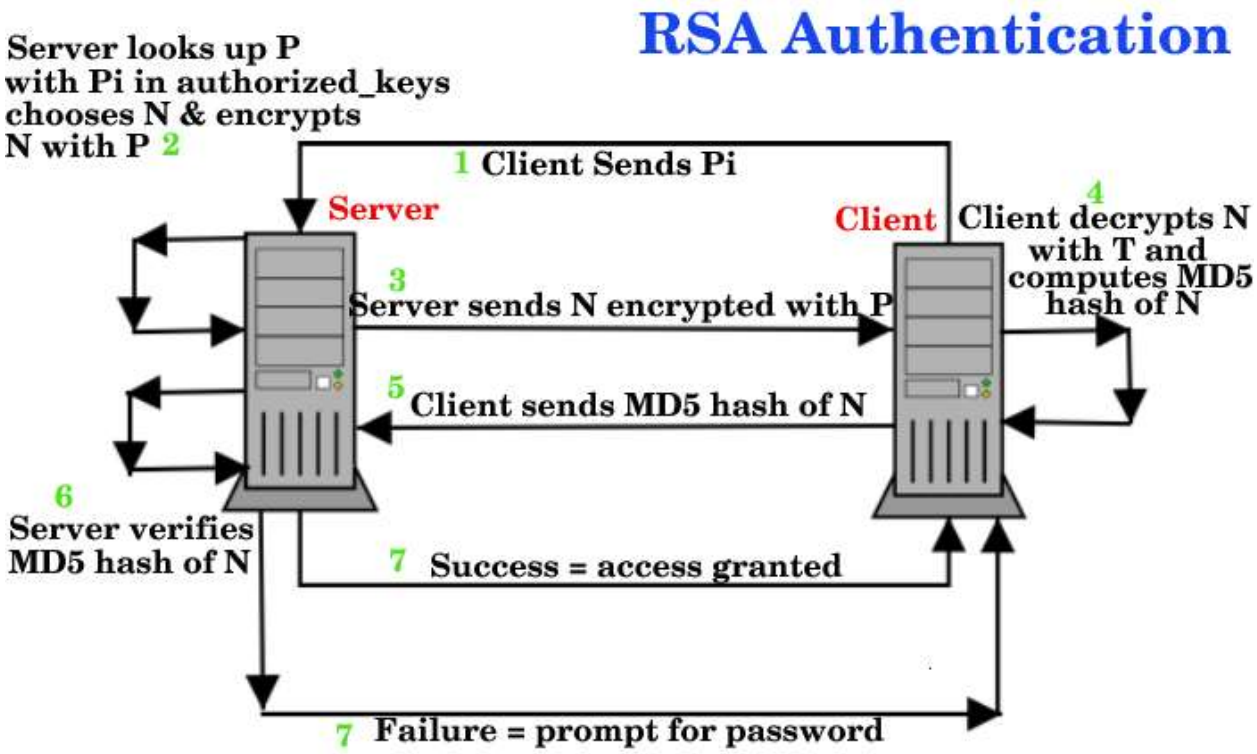

Figure 2.1RSA Authentication

This service provides the authentication of the party at the other end of the line. In the connection oriented communication, it provides the authentication of the sender or receiver during the connection establishment (peer entity authentication). In connectionless communication, it aauthenticates the source of data (also called data origin authentication).

d. Non-repudiation:

Non-repudiation service protects against repudiation by either the sender or the receiver of the data. In this with the proof of origin, the receiver of the data can later prove the identity of the sender. If denied. In non-repudiation with the real proof of delivery the sender of the data can later prove the data were delivered to the intended recipient [11, 12].

Non-repudiation is the concept of ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contract. Although this concept can be applied to any transmission, including television and radio, by far the most common application is in the verification and trust of signatures.

e. Access Control:

Access control is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system. An access control system, within thefield of physical security, is generally seen as the second layer in the security of a physical structure. It provides security against unauthorized access against data. The term access in this definition is very broad and can involve reading, writing, modifying, executing programs. [2,3]

2.1.2 Classification:

Cryptographic systems are generally classified along three independent dimensions:

1. Type of operations used for transforming plaintext to cipher text. All encryption algorithms are based on two general principles. Those are substitution, in which each element in the plain text is mapped into another element and transposition in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost. Most systems referred to as product systems, involved multiple stages of substitution and transposition.

2. The number of keys used: If sender and receiver use the same key, the system is referred to as symmetric, single key or secret key conventional encryption. If the sender and the receiver each uses a different key the system is referred to as asymmetric, two key, or public-key encryption.

3. The way in which the plaintext is processed: A block cipher processes the input on block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

2.2 Algorithms and Keys

A cryptographic algorithm, also called a cipher, is the mathematical function used for encryption and decryption. (Generally, there are two related functions: one for encryption and the other for decryption).

2.2.1 Terminology

If the security of an algorithm is based on keeping the way that algorithm works a secret, it is a restricted algorithm. Restricted algorithms have historical interest, but are woefully inadequate by today‟s standards. A large or changing group of users cannot use them, because every time a

user leaves the group everyone else must switch to a different algorithm. If someone accidentally reveals the secret, everyone must change their algorithm.

Even more damning, restricted algorithms allow no quality control or standardization. Every group of users must have their own unique algorithm. Such a group can‟t use off-the-shelf hardware or software products; an eavesdropper can buy the same product and learn the algorithm. They have to write their own algorithms and implementations. If no one in the group is a good cryptographer, then they won‟t know if they have a secure algorithm. Despite these major drawbacks, restricted algorithms are enormously popular for low-security applications. Users either don‟t realize or don‟t care about the security problems inherent in their system.

Modern cryptography solves this problem with a key, denoted by K. This key might be any one of a large number of values. The range of possible values of the key is called the keyspace. Both the encryption and decryption operations use this key (i.e., they are dependent on the key and this fact is denoted by the K subscript), so the functions now become:

$E_k(M) = C$

$D_k(C) = M$

Those functions have the property that :

$D_k(A_k(M)) = M$

Some algorithms use a different encryption key and decryption key . That is, the encryption key, K1, is different from the corresponding decryption key, K2. In this case:

$E_{k1}(M) = C$

$D_{k2}(C) = M$

$D_{k2}(E_{k1}(M)) = M$

All of the security in these algorithms is based in the key (or keys); none is based in the details of the algorithm. This means that the algorithm can be published and analyzed. Products using the algorithm can be mass-produced. It doesn‟t matter if an eavesdropper knows your algorithm; if she doesn‟t know your particular key, she can‟t read your messages.

2.3 Symmetric Algorithms

There are two general types of key-based algorithms: symmetric and public-key. Symmetric algorithms, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the

encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms, single-key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely. The security of a symmetric algorithm rests in the key; divulging the key means that anyone could encrypt and decrypt messages. As long as the communication needs to remain secret, the key must remain secret.

Encryption and decryption with a symmetric algorithm are denoted by:

Ek(M) = C

Dk (C) =M

Symmetric algorithms can be divided into two categories. Some operate on the plaintext a single bit (or sometimes byte) at a time; these are called stream algorithms or stream ciphers. Others operate on the plaintext in groups of bits. The groups of bits are called blocks, and the algorithms are called block algorithms or block ciphers. For modern computer algorithms, a typical block size is 64 bits large enough to preclude analysis and small enough to be workable. (Before computers, algorithms generally operated on plaintext one character at a time. You can think of this as a stream algorithm operating on a stream of characters.)

2.4 Public-Key Algorithms

Public-key algorithms (also called asymmetric algorithms) are designed so that the key used for encryption is different from the key used for decryption. Furthermore, the decryption key cannot (at least in any reasonable amount of time) be calculated from the encryption key. The algorithms are called “public-key” because the encryption key can be made public: A complete stranger can use the encryption key to encrypt a message, but only a specific person with the corresponding decryption key can decrypt the message. In these systems, the encryption key is often called the public key, and the decryption key is often called the private key. The private key is sometimes also called the secret key, but to avoid confusion with symmetric algorithms, that tag won‟t be used here.

Encryption using public key K is denoted by:

Ek(M) = C

Even though the public key and private key are different, decryption with the corresponding private key is denoted by:

Dk(C) = M

Sometimes, messages will be encrypted with the private key and decrypted with the public key; this is used in digital signatures. Despite the possible confusion, these operations are denoted by, respectively:

Ek(M) = C

Dk(C) = M

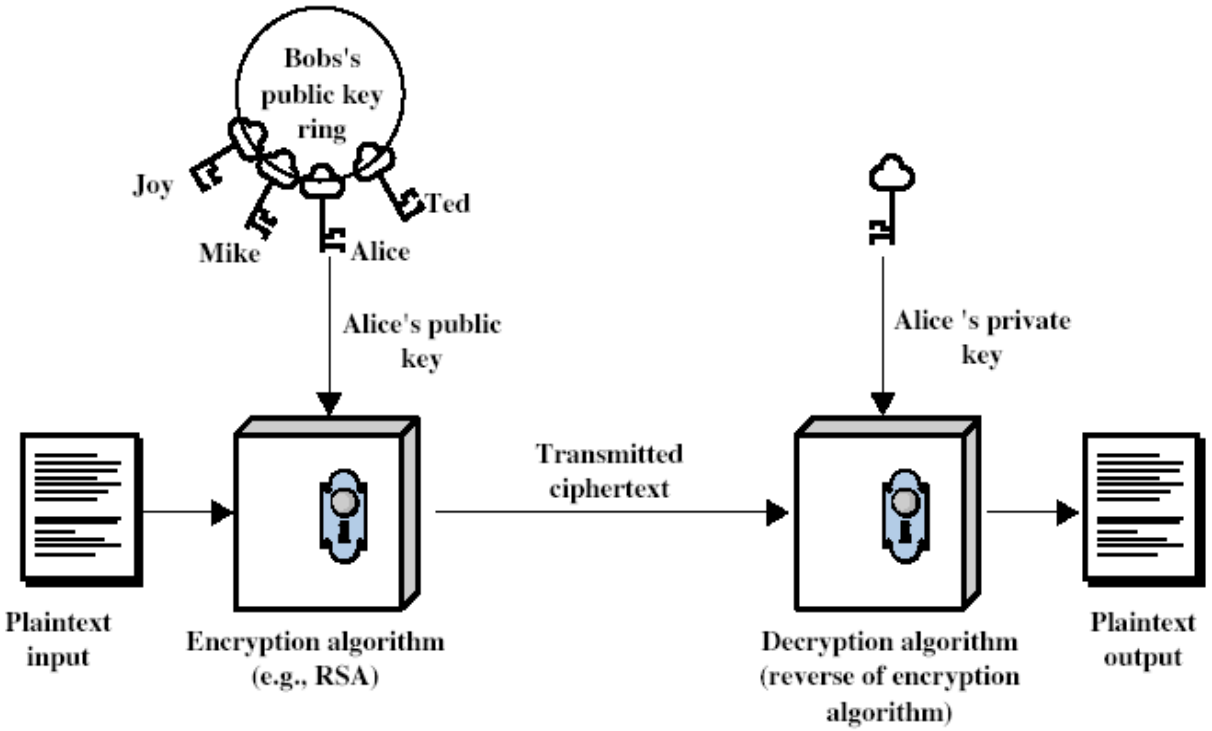

Figure 2.2 Public Key Algorithm

2.5 Symmetric-key vs. public-key cryptography

Symmetric-key and public-key encryption schemes have various advantages and disadvantages, some of which are common to both. This section highlights a number of these and summarizes features pointed out in previous sections.

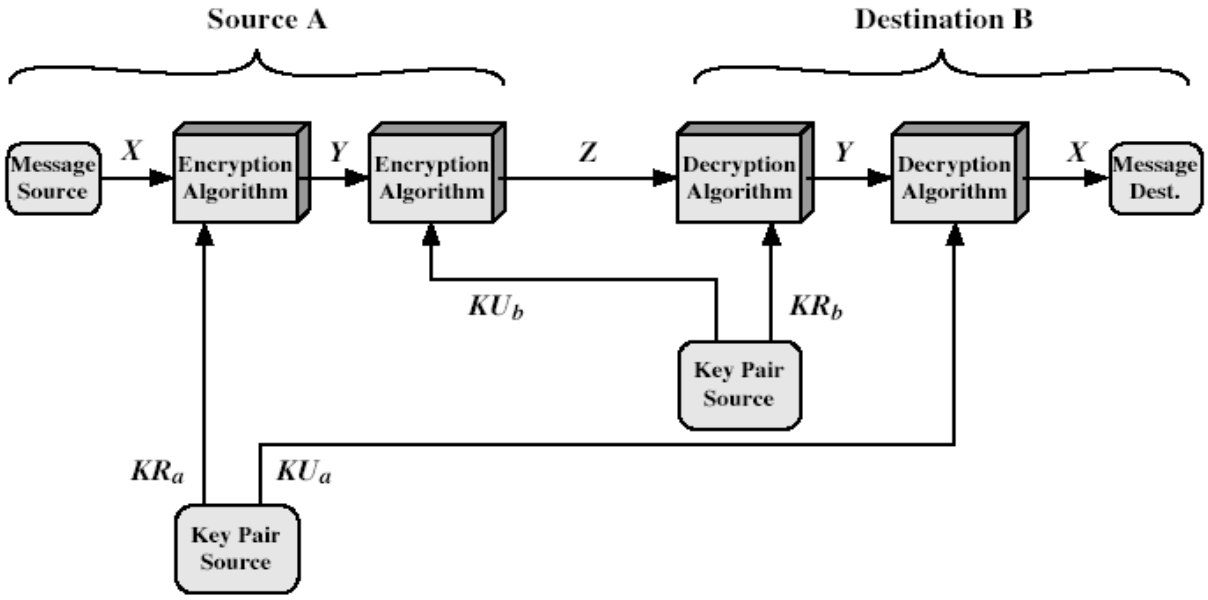

Figure 2.3 Symmetric Key

2.5.1 (i) Advantages of symmetric-key cryptography

1. Symmetric-key ciphers can be designed to have high rates of data throughput. Some hardware implementations achieve encrypt rates of hundreds of megabytes per second,while software implementations may attain throughput rates in the megabytes per second range.

2. Keys for symmetric-key ciphers are relatively short.

3. Symmetric-key ciphers can be employed as primitives to construct various cryptographic mechanisms including pseudorandom number generators, hash functions, and computationally efficient digital signature schemes, to name just a few.

4. Symmetric-key ciphers can be composed to produce stronger ciphers. Simple transformations which are easy to analyze, but on their own weak, can be used to construct strong product ciphers.

5. Symmetric-key encryption is perceived to have an extensive history, although it must be acknowledged that, notwithstanding the invention of rotor machines earlier, much of the knowledge in this area has been acquired subsequent to the invention of the digital computer, and, in particular, the design of the Data Encryption Standard in the early 1970s.

2.5.2 (ii) Disadvantages of symmetric-key cryptography

1. In a two-party communication, the key must remain secret at both ends.

2. In a large network, there are many key pairs to be managed. Consequently, effective key management requires the use of an unconditionally trusted TTP.

3. In a two-party communication between entities A and B, sound cryptographic practice dictates that the key be changed frequently and perhaps for each communication session.

4. Digital signature mechanisms arising from symmetric-key encryption typically require either large keys for the public verification function or the use of a TTP.

2.5.3 (iii) Advantages of public-key cryptography

1. Only the private key must be kept secret (authenticity of public keys must, however, be guaranteed).

2. The administration of keys on a network requires the presence of only a functionally trusted TTP as opposed to an unconditionally trusted TTP. Depending on the mode of usage, the TTP might only be required in an "off-line" manner, as opposed to in real time.

3. Depending on the mode of usage, a private key/public key pair may remain unchanged for considerable periods of time, e.g., many sessions (even several years).

4. Many public-key schemes yield relatively efficient digital signature mechanisms. The key used to describe the public verification function is typically much smaller than for the symmetric-key counterpart.

5. In a large network, the number of keys necessary may be considerably smaller than in the symmetric-key scenario.

2.5.4 (IV) Disadvantages of public-key encryption

1. Throughput rates for the most popular public-key encryption methods are several orders of magnitude slower than the best known symmetric-key schemes.

2. Key sizes are typically much larger than those required for symmetric-key encryption , and the size of public-key signatures is larger than that of tags providing data origin authentication from symmetric-key techniques.
3. No public-key scheme has been proven to be secure (the same can be said for block ciphers). The most effective public-key encryption schemes found to date have their security based on the presumed difficulty of a small set of number-theoretic problems.

4. Public-key cryptography does not have as extensive a history as symmetric-key encryption, being discovered only in the mid 1970.

2.6 Digital signatures

A cryptographic primitive which is fundamental in authentication, authorization, and no repudiation is the digital signature. The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information. The process of signing entails transforming the message and some secret information held by the entity into a tag called a signature. A generic description follows. Nomenclature and set-up

1. M is the set of messages which can be signed.

2. S is a set of elements called signatures, possibly binary strings of a fixed length.

3. SA is a transformation from the message set M to the signature set S, and is called a signing transformation for entity A. The transformation SA is kept secret by A, and will be used to create signatures for messages from M.

4. VA is a transformation from the set M x S to the set {true, false}. VA is called a verification transformation for A‟s signatures, is publicly known, and is used by other entities to verify signatures created by A.

2.6.1 Signing procedure

Entity A (the signer) creates a signature for a message m E M by doing the following: 1. Compute s = SA(m). 2. Transmit the pair (m, s). s is called the signature for message m.

2.6.2 Verification procedure

To verify that a signature s on a message m was created by A, an entity B (the verifier) performs the following steps:

1. Obtain the verification function VA of A.

2. Compute u = VA(m, s).

3. Accept the signature as having been created by A if u = true, and reject the signature if u = false.

2.6.3 Properties required for signing and verification functions

There are several properties which the signing and verification transformations must satisfy.

(a) s is a valid signature of A on message m if and only if VA(m, s) = true.
(b) (b) It is computationally infeasible for any entity other than A to find, for any m E M, an s E S such that VA(m, s) = true.

No one has yet formally proved that digital signature schemes satisfying (b) exist (although existence is widely believed to be true); however, there are some very good candidates.

2.7 RSA

Soon Soon after Merkle‟s knapsack algorithm came the first full-fledged public-key algorithm, one that works for encryption and digital signatures: RSA. Of all the public-key algorithms proposed over the years, RSA is by far the easiest to understand and implement. It is also the most popular. Named after the three inventors-Ron Rivest, Adi Shamir, and Leonard Adleman-it has since withstood years of extensive cryptanalysis. Although the cryptanalysis neither proved nor disproved RSA‟s security, it does suggest a confidence level in the algorithm.

RSA gets its security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large (100 to 200 digits or even larger) prime numbers. Recovering the plaintext from the public key and the ciphertext is conjectured to be equivalent to factoring the product of the two primes. To generate the two keys, choose two random large prime numbers, p and q. For maximum security, choose p and q of equal length. Compute the product:

n= pq

Then randomly choose the encryption key, e, such that e and (p - l)(q - 1) are relatively prime. Finally, use the extended Euclidean algorithm to compute the decryption key, d, such that

ed= 1 mod(p- l)(q- 1)

In other words,

d = e-1 mod ((p - l)(q - 1))

Note that d and n are also relatively prime. The numbers e and n are the public key; the number d is the private key. The two primes, p and q, are no longer needed. They should be discarded, but never revealed.

To encrypt a message m, first divide it into numerical blocks smaller than n (with binary data, choose the largest power of 2 less than n). That is, if both p and q are l00-digit primes, then n will have just under 200 digits and each message block, m, should be just under 200 digits long. (If you need to encrypt a fixed number of blocks, you can pad them with a few zeros on the left to ensure that they will always be less than n.) The encrypted message, c, will be made up of similarly sized message blocks, ci, of about the same length.

A short example will probably go a long way to making this clearer. If p = 47 and q= 71, then

Public Key:

n product of two primes, p and q (p and q must remain secret) e relatively prime to (p - l)(q - 1)

Private Key:

d e-1 mod ((p- l)(q- 1))

Encrypting:

c=me mod n

Decrypting

m=cd mod n

n= pq =3337

The encryption key, e, must have no factors in common with

(p- l)(q- 1)=46 l 70=3220

Choose e (at random) to be 79.

In that case d = 79-1 mod 3220 = 1019

This number was calculated using the extended Euclidean algorithm. Publish e and n, and keep d secret. Discard p and q.

To encrypt the message

m=6882326879666683

first break it into small blocks. Three-digit blocks work nicely in this case. The message is split into six blocks, mi, in which

m1 = 688

m2 = 232

m3 = 687

m4 = 966

m5 = 668

m6 = 003

The first block is encrypted as

68879 mod 3337 = 1570 = c1

Performing the same operation on the subsequent blocks generates an encrypted message:

c=1570 2756 2091 2276 2423 158

Decrypting the message requires performing the same exponentiation using the decryption key of 1019, so P 15701019 mod 3337 = 688 = m1

The rest of the message can be recovered in this manner.

Speed of RSA

In hardware, RSA is about 1000 times slower than DES. The fastest VLSI hardware implementation for RSA with a 51%bit modulus has a throughput of 64 kilobits per second

There are also chips that perform 1024bit RSA encryption. Currently chips are being planned that will approach 1 megabit per second using a 512-bit modulus; they will probably be available in 1995. Manufacturers have also implemented RSA in smart cards; these implementations are slower. In software, DES is about 100 times faster than RSA. These numbers may change slightly as technology changes, but RSA will never approach the speed of symmetric algorithms.

2.8 Data Encryption Standard:

Data encryption standard is the most widely used method of data encryption using a secret key. There are 72,000,000,000,000,000 (72 quadrillion) or more possible encryption keys that can be used. For each given message, the key is chosen at random from among this enormous number of keys. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key. It was developed in the 1970s by the National Bureau of Standards with the help of the National Security Agency. Its purpose is to provide a standard method for sensitive commercial and unclassified data. IBM created the first draft of the algorithm, calling it LUCIFER. DES officially became a standard in November of 1976. Data encryption algorithm has a 64-bit block size and uses a 56bit key during execution (8 parity bits are stripped of from the full 64-bit key). The DEA can also be used for single user encryption, such as to store files in a hard in encrypted form. NIST re-certifies DES every 5 years. DES has been in world wide use for over 20 years, and due to the fact that it is a defined standard that any system implementing DES can communicate with any other system using is it.

2.8.1 DES Encryption:

DES is a symmetric, block-cipher algorithm with a key length of 64 bits, and the algorithm operates on successive 64 bit blocks of plain text. Due to symmetric, the same key is used for encryption and decryption, and also uses the same algorithm for encryption and decryption.

Initially a transposition is carried out according to a set table (the initial permutation), the 64-bit plaintext block is then split into two 32-bit block, and 16 identical operations called rounds are carried out on each half. The two halves are joined back together , and the reverse of the initial permutation is carried out. The purpose of the first transposition is clear; it does not affect the security of the algorithm, but is through for the purpose of allowing plaintext and cipher text to be loaded into 8-bit chip in byte-sized pieces. In any round, only one half of the original 64-bit

block is operated on. The rounds alternate between the two halves. One round in DES consists of the following

2.8.1.1Key Transformation:

The keys reduced from 64-bit to 56-bit by removing every eighth bit, which are sometimes used for error checking. 16 different 48 bit sub-keys are then generated i.e. one for each round. This is achieved by splitting the 56-bit key into the two halves, and then circularly shifting them left by one or two bits, depending on the round .After this, 48 of the bits are selected. Because they are shifted, different groups of key bits are used in each sub key. This process is called compression permutation due to transposition of bits & reduction of the overall size.

2.8.1.2Expansion Permutation:

Whichever half of the block is being operated on undergoes a permutation after key transformation. In this operation, the expansion & the transposition are achieved simultaneously by allowing the first and fourth bits in each block for bit block to appear twice in the output that is the fourth input bit becomes the fifth and the seventh output bit. The expansion permutation achieves 3 things. Those are described below

1) It increases the size of the half block from 32 to 48 bit, the same number of bit as in compressed key subset, which is important as the next operation is to XOR the two together.

2) It produces a long string of data for the substitution operation that subsequently comprises it.

3) Because in the subsequent substitutions on the first & 4th bits appearing in 2 Sboxes, they affect two substitutions. The effect of this is the dependency of the output on the input bits is that the dependency of the output on the input bits increases rapidly.

2.8.1.3 XOR:

XOR operation performed with the appropriate subset key for that round & the resulting 48 block.

2.8.1.4Substitution*:*

After XOR operation the next operation is to perform substitution on the expanded block. There are 8 substitution boxes called S-boxes. The first S-Box operates on the first 6 bits of the 48 bit expanded block, the second S-box on the next 6 & so on. Each S-box operates from a table of 4 rows & 16 columns; each entry on a table is a 4 bit number. The 6 bit number the s-box takes as input is used to look up the appropriate entry in the table in the following way. The first and the 6 bits combine to form a two bit number corresponding to a row number, the second and fifth bit combine to form a 4 bit number corresponding to a particular column. The net result of the substitution phase is 8 4bit blocks that are then combined to form a 32 bit block. It is the non-linear relationship of the S-boxes that really provides DES with its security.

2.8.1.5 Permutation*:*

The 32 bit output of a substitution phase then undergoes a straight forward transposition using a table called P-Box. After all the round has been completed, the two half blocks of 32 bits are recombined to form a 64 bit output. The final permutation is performed on it, and the resulting 64 bit block is the desired DES encrypted cipher text of the input plain text block.

2.8.2 DES Decryption:

If one has the correct key decrypting DES is very easy. The decryption algo is identical to the encryption algo. The only change is to decrypt DES cipher text; the subsets of the keys use in each round are used in reverse, which is the 16th subset is used first.

2.8.3 Security of DES:

DES can no longer be considered a sufficiently secured algorithm if the DES secured message can be broken in minutes by a super computer. Then the rapidly increasing power of computer means, it‟llbe trivial to break DES in future. An extension of DES called DESX is considered virtually immune to key search.

2.9 International Data Encryption Algorithm:

The international data encryption algorithm is a symmetric block cipher developed by Xuejia Lai & James Massey in the Swiss federal institute of Technology in 1990 and was called the

proposed encryption standard PES. In 1991, Lai & Massey strengthened the algorithm against differential crypt analysis and called the result improved PES. The IPES name was changed to International Date Encryption Algorithm in 1992.

IDEA is one of a number of conventional encryption algorithms that have been proposed in recent years to replace DES. In terms of adoption, IDEA is one of the most successful of these proposals. IDEA is best known for its use in PGP (pretty good privacy) in network protocol.

2.9.1 The Algorithm

IDEA algorithm with the key length of 128 bits, a block size of the 64 bits and as with DES, the same algorithm provides encryption and decryption. IDEA consists of 8 rounds using 52 sub keys. Each round uses 6 sub keys with the remaining 4 being used for output transformation. The Sub-keys are created as follows:

1) The 128 bit key is divided into 8 16 bit keys to provide the first 8 sub keys.

2) The bits of the original key are then shifted 25 bits to the left, and then it is again split in 8 sub keys.

3) The shifting & splitting is repeated until all 52 sub keys (SK1-SK52) have been created.
The 64 bit plain text is first split into four blocks. A round then consists of the following steps:

Output block-1

(OB1) = b1*sk1 (multiply 1st sub-block with 1st sub key)

(OB2) = b2+sk2 (add 2nd sub-block with 2nd sub key)

(OB3) = b3+sk3 (add 3rd sub-block with 3rd sub key)

(OB4) = b4*sk4 (multiply 4th sub-block with 4th sub key)

(OB5) = OB1 XOR OB3 (XOR results of 1 & 3)

(OB6) = OB2 XOR OB4

(OB7) = OB5 * SK5

(OB8) = OB6+OB7

(OB9) = OB8*SK6

(OB10) = OB7+OB9

(OB11) = OB1 XOR OB9

(OB12) = OB3 XOR OB9

(OB13) = OB2 XOR OB10

(OB14) = OB4 XOR OB10

The input to the next round is the four sub blocks OB11, OB13, OB12, and OB14 in that order. After the eighth round, the four final output blocks (F1-F4) are used in a final transformation to produce 4 sub blocks of cipher text c1-c4 that are then rejoined to form final 64 bit block of cipher text

C1=F1*SK49

C2=F2+SK50

C3=F3+SK51

C4=F4+SK52

Cipher Text= C1 C2 C3 C4

2.9.2 Security provided by IDEA:

IDEA is approximately twice as fast as DES and is also considerably more secure. Using a brute force approach there are 2128 possible keys. If a billion chips that could each test 1 billion keys a second would try and crack an IDEA encrypted message, it would take them 1013 years. Being a fairly new algorithm, it is possible a better attack than brute force will be found, when coupled with more powerful machines in the future may be able to crack a message. However, a long way into future IDEA seems to be a very secure algorithm.

2.10 Blowfish:

Blowfish is a variable-length key block cipher developed by Bruce Schneier. It does not meet all the requirements for a new cryptographic standard discussed above: It is only suitable for

applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than DES when implemented in 32-bit microprocessors with large data caches, such as the Pentium and the Power PC.

2.10.1 Description of the Algorithm:

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts; a key-expansion part and a date-encryption part. Key expansion converts a key of at most 448 bits into several sub-key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key dependent permutation, and a key-and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

2.10.2 Sub-keys:

Blowfish uses a large number of sub-keys. These keys must be pre-computed before any data encryption or decryption.

1. The P-array consists of 18 32-bit sub-keys:
P1, P2…P18

2. There are four 32-bit S-boxes with 256 entries each:
S1, 0, S1, 1 ……., S1, 255;

S2, 0 S2, 1… S2, 255;

S3, 0, S3, 1… S3, 255;

S4, 0, S4, 1… S4, 255;

2.10.3 Encryption and Decryption:

Blowfish is a Feistel network consisting of 16 rounds. The input is a 64-bit data element, X Divide X into two 32-bit halves XL, XR

For I = 1 to 16

XL = XL XOR Pi

XR = F (XL) XOR XR

Swap XL and XR

Swap XL and XR (Undo the last swap)

XR = XR XOR P17

XL = XL XOR P18

Recombine XL and XR

Function F: Divide XL into four eight-bit quarters: a, b, c and d

F (XL) = ((S1,a + S2,b + mob 2 32) XOR S3,C) + S4,d MOB 232 (2.25) Decryption

Decryption is exactly the same as encryption, except that P1, P2 … P18 are used in the reverse order.

Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all sub keys are stored in cache.

Generating the Sub keys:

The sub keys are calculated using the Blowfish algorithm. The exact method is as

follows:

1. Initialize first the P-array and then the four S-boxes, in order with fixed string.
This string consists of the hexadecimal digits of fractional part of Pi

(Less the initial 3). For example:

P1 = 243f6a88

P2 = 85a308d3

P3 = 13198a2e

P4 = 03707344

2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is a least one equivalent longer key‟ for example, if A is a 64-bit key, then AA, AAA, etc equivalent keys.)

3. Encrypt the all-zero string with the Blowfish algorithm using the sub keys described in steps (1) and (2).

4. Replace P1 and P2 with the output of step (3).

5. Encrypt the output of step (30 using the Blowfish algorithm with the modified sub-keys.

6. Replace P3 and P4 with the output of step (5).

7. Continue the process, replacing all entries of the P-array, and then all four Sboxes in order, with the output of the continuously changing Blowfish algorithm. In total, 521 iterations are required to generate all required sub-keys. Applications can store the sub-keys rather than derivation process multiple times.

2.11 RC Cipher:

RC stands for *Ron's Code or Rivest Cipher*. These ciphers were designed by Ron Rivest for the RSA Data Security. Different RC Ciphers described briefly below.

2.11.1 RC2:

It was designed as a quick-fix replacement for DES that is more secure. It is a block cipher with a variable key size that has propriety algorithm RC2 is a variable-key length cipher. However, when using the Microsoft base cryptographic provider, the key-length is hard –coded to 40 bits. When using the Microsoft enhanced cryptographic provider, the key length is 128 bits by default and can be in the range of 40 to 128 bit in 8-bit increments.

2.11.2 RC4:

It was developed by Ron Rivest in 1987. It is a variable-key-size stream cipher. The details of the algorithm have not been officially published. The algorithm is extremely easy to describe and program just like RC2, 40 bit RC4 is supported by the Microsoft base Cryptography provider, and the enhanced provider allows keys in the range of 40 to 128 bits in 8-bit increments.

2.11.3 RC5:

RC5 is a block designed for speed. It allows a user defined key length, data block size, and number of encryption rounds. In particularly the key size can be as large as 2,048 bits. RSA Data security is working to have RC5 included in numerous internet standards including 1Psec.

2.12 Cryptanalysis:

Cryptanalysis is the method of obtaining the meaning of encrypted information without the information of the secret parameters that are normally required to obtain the meaning.This typically involves the knowing of the system, how it works and finding the secret key. In non-technical language, this is the practice of code breaking or cracking the code, although these phrases have a specialized technical meaning. "Cryptanalysis" is used also to refer to any attempt to circumvent the security of some other types of cryptographic algorithms and protocols in general, and not just encryption. However, cryptanalysis usually excludes methods of attack that do not primarily target weaknesses in the actual cryptography, although these types of attack are an important concern and are often more efiective than traditional cryptanalysis. [2, 3]
The International Telecommunication union-Telecommunication standardization Sector (ITU-T) provides some security services and some mechanism to implement those services.

2.13 Digital signatures

A cryptography discipline primitive that is prime in authentication, authorization, and no repudiation is that the digital signature. The aim of a digital signature is to produce a way for Associate in nursing entity to bind its identity to a chunk of knowledge. There are No one has yet formally proved that digital signature schemes satisfying (b) exist (although existence is widely believed to be true); however, there are some very good candidates.

Figure 2.6 shows the digital signature method. The sender uses language formula to sign the message. The message and also the signature area unit sent to the receiver. The receiver receives the message and also the signature and applies the collateral formula to the mix. If the result's true, the message is accepted; otherwise, it's rejected.
When a document is signed, anyone, together with shiny, will verify it as a result of everybody has access to Haney’s public key. Haney should not use her public key to sign documents as a result of then anyone may forge her signature.

2.14 Limitations of digital signature

a) Too slo10 to 40 times slower than RSA)

b) Too insecure (fixed 512bit key).

2.15 Diffie-Hellman algorithm

In 1976, Whitfield Diffie and Martin Hellman introduced a key exchange protocol using the discrete logarithm problem.

Diffie–Hellman establishes a shared secret which will be used for secret communications by exchanging information over a public network.

The purpose of the algorithm is used to enable users to securely exchange a key that can be used for subsequent encryption.

The algorithm chooses two public known numbers a prime number n and g that is a primitive root of n.

Suppose user A and B wish to exchange a key for their communication, then, user A select random integer(private key) $X_a < n$ and compute

Public key $y_a=g^{xa}$ mod n. Then user B select random integer (private key) $X_b< n$ and compute public key $y_b=g^{xb}$ mod n.

Secret key, computes by A is $K=y_b^{xa}$ mod n. Similarly

Secret key, computes by A is $K=y_a^{xb}$ mod n. (Thanuja R, Dilip Kumar S / International Journal of Engineering Research and Applications (IJERA))

2.16 Limitation of Diffie-Hellman Algorithm

The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack as follows, suppose there is a user C who is going

Intercept the secret key shared between user A and user B. User C generates two random private keys x_{d1} and x_{d2} and then computes corresponding public keys

$\mathbf{y_{d1}}$ and $\mathbf{y_{d2}}$. User A transmits public key y_a to user B. in the meanwhile user C intercepts y a and transmits his public key y_{d1} to user B. User C

Also calculate K2= $\mathbf{y_{d2}}^{\mathbf{xa}}$ mod n. B receives y_{d1} and calculate secret key K1=$\mathbf{y_{d1}}^{\mathbf{xb}}$ mod n. Now, user B transmits his private key xa to A. Now, C intercepts and transmits his own public

key y_{d2} to A. Now, A receives y_{d2} and calculates corresponding K2. Since the algorithm is easily cracked by discrete logarithm approach as above.

2.17 RSA

Soon after Merkle‟s knapsack algorithm came the first full-fledged public-key algorithm, one that works for encryption and digital signatures: RSA. Of all the public-key algorithms proposed over the years, RSA is by far the easiest to understand and implement. It is also the most popular.

Named after the three inventors-Ron Rivest, Adi Shamir, and Leonard Adleman-it has since withstood years of extensive cryptanalysis. Although the cryptanalysis neither proved nor disproved RSA security, but also it does suggest a confidence level in the algorithm.

RSA gets its security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large (100 to 200 digits or even larger) prime numbers.

To generate the two keys two large prime numbers P & Q are chosen at random. For maximum security the value of the choose p and q should

$n= p*q$

Then randomly choose the encryption key, e, such that e and $(p - l)*(q - 1)$ are relatively prime. Finally, use the extended Euclidean algorithm to compute the decryption key, d, such that

$ed= 1 \bmod (p- l)*(q- 1)$

In other words,

$d = e^{-1} \bmod ((p - l)(q - 1))$

A short example will probably go a long way to making this clearer. If p = 47 and q= 71, then

Public Key:

n product of two primes, p and q (p and q must remain secret) **e** relatively prime to (p - l) (q - 1)

Private Key:

$de =1 \bmod ((p- l)(q- 1))$

Encrypting:

$c=m^e \bmod n$

Decrypting

$m=c^d \bmod n$

2.18 factoring problem of pervious RSA algorithm:

Security of RSA public key cryptosystem relies on the idea that factorization of enormous variety. Number resolving is a very important drawback principally owing to its reference to RSA algorithmic rule of public key cryptosystem.[3]

If the method resolution(factorization) is finished, then the entire rule will become breakable. [1]

2.19 RSA attacks

Three attainable approaches to offensive the RSA algorithmic rule square measure as follows-

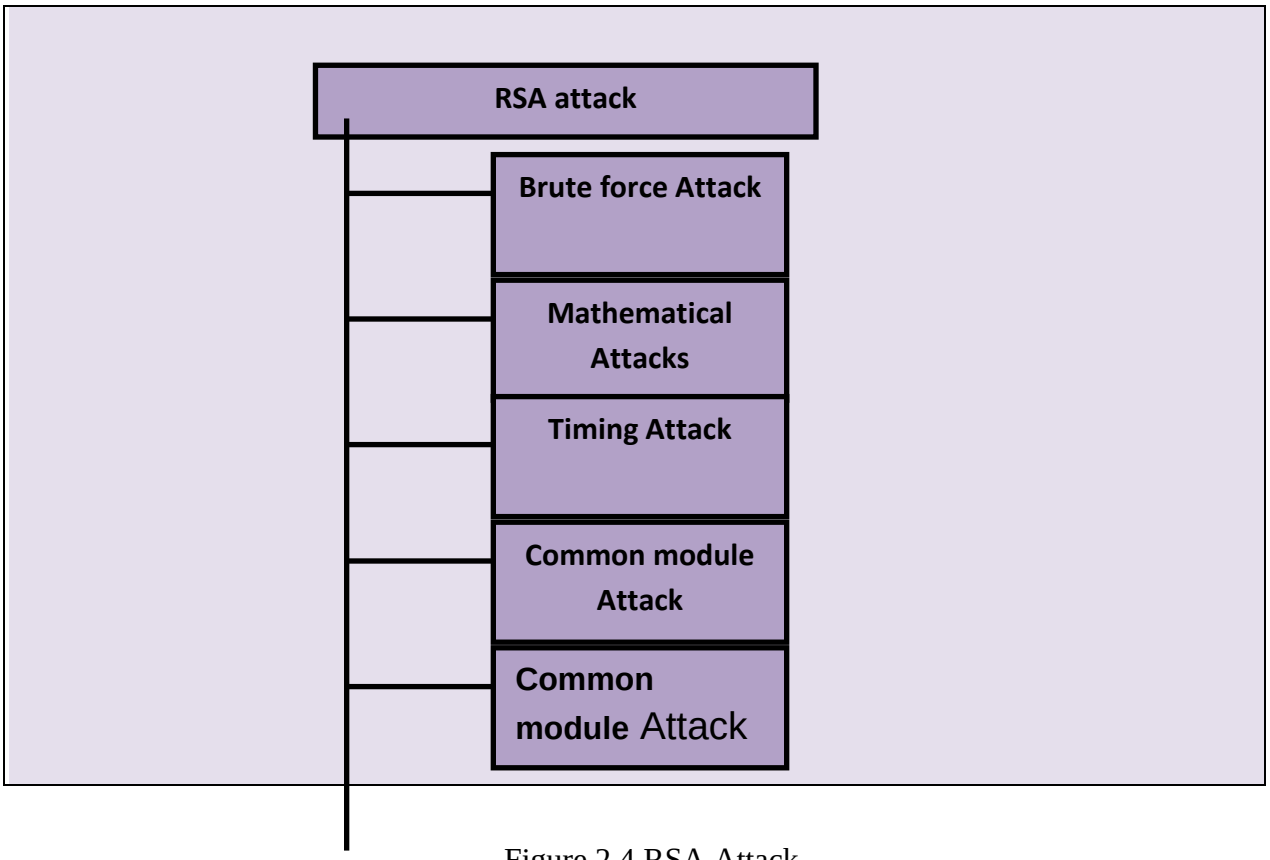

Figure 2.4 RSA Attack

- **Brute force attack:**

  This involves making an attempt all attainable private keys. A brute force attack against a cipher consists of breaking a cipher by making an attempt all attainable keys. Statistically, if the keys were originally chosen willy-nilly, the plaintext can become out there when regarding half the attainable keys are a unit tried.

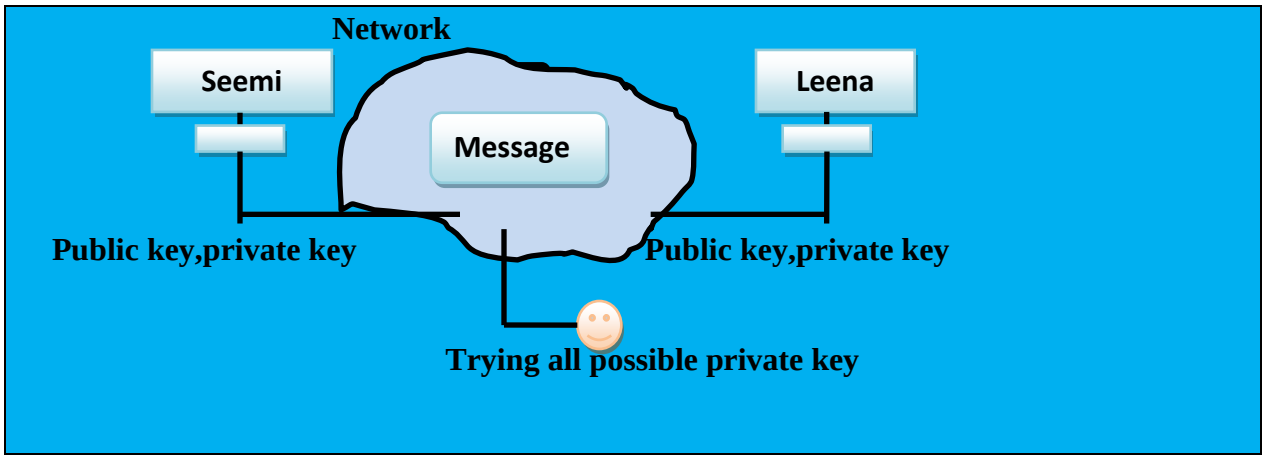

Figure 2.5 Brute force Attack

- **Mathematical Attacks**

  Mathematical attacks specialize in assistive the underlying structure of RSA perform. The primary intuitive attack is that they conceive to issue the modulus N. As a result of knowing the resolving of N, one could simply obtain φ (N), from that do will be determined by d = 1/e mod φ(N). However, at present, the quickest factoring algorithmic program runs in exponential time. Our objective is to survey RSA attacks that decrypt the message while not directly factorization N. There is a unit many approaches, all equivalent in impact to factorization the product of 2 prime numbers.

- **Timing Attack:**

These depend on the running time of the decryption scheme. Timing attack is applicable not just to RSA, but to other public key cryptography system. This attack is alarming for two reasons- it comes from a completely unexpected direction and it is a cipher text only attack.

Although the timing attack may be a serious threat, there are unit some countermeasures which will be used-

a) Constant exponentiation time- make positive that each one mathematical process takes identical quantity of your time before returning a result.this is an easy fix however will degrade performance.

b) Better performance may be achieved by adding a random delay to the mathematical operation rule to confuse the timing attack.

But Kocher shows that if defenders do not add enough noise, attackers might still succeed by collection further measurements to catch up on the random delays.

c) Blinding- multiply the ciphertext by a random variable before performing exponentiation. This process prevents the attacker from knowing what ciphertext bits are being processed inside the computer and therefore prevents the bit—by-bit analysis essential to the timing attack.

- **Common module attack:**

To avoid generating a special modulus =pq for every user one would like to repair N once and for all. A similar N is employed by all users. A trusty central may offer user I with a novel combine ei,di from that user I type a public key(N,ei) and a secret key (N,di).
At first look, this might appear to work: a ciphertext $c = m^{e}_{a} \bmod N$ meant for Alina can not be decrypted by Balina since Balina doesn't posses d_a. However, this can be incorrect and also the

Ensuing system is insecure. By truth one Balina will use his on exponents e_b, sound unit to issue the modulus N. Once N is factored Balina will recover Alina private key d_a from her public key e_a.

- **Wiener's attack**

The Wiener's attack, named once decipherer archangel J. Wiener. The wiener's attack uses the fraction technique to show the non-public key d once d is less.

A wiener's attack is based on two facts:

- If N=pq is a "good" RSA modulus (with p (approx) ≈ (approx) ≈√N), then N (approx) ≈φ(n).
- The wiener's set up is this: as a result of ed ≈ one mod m for variety of|some|many} modulus m>=1 and positive number e and d, then d looks as a divisor at intervals the convergence of e/m. (For identical reason that one can use the Euclidean algorithm to work reciprocal modulo m.)

2.17 Cryptanalysis attacks

Cryptanalysis is the method of obtaining the meaning of encrypted information without the informed about the secret parameters that are normally required to obtain the meaning.
This typically involves the knowing of the system, how it works and finding the secret key. In non-technical language, this is the practice of code breaking or cracking the code, although these phrases have a specialized technical meaning.

"Cryptanalysis" is used also to refer to any attempt to circumvent the security of some other types of cryptographic algorithms and protocols in general, and not just encryption. However, cryptanalysis usually excludes methods of attack that do not primarily target weaknesses in the actual cryptography, although these types of attack are an important concern and are often more effective than traditional cryptanalysis. [2, 3] The International Telecommunication Union-Telecommunication standardization Sector (ITU-T) provides some security services and some mechanism to implement those services.

Cipher Text Only Attack

In a Ciphertext only Attack, Donald must access to just some ciphertext.He tries to seek out the corresponding key and therefore the plaintext.

The assumption is that Donald is aware of the formula and may intercept the ciphertext.

- **Known Plain Text Attack**

  In A best-known Plain Text Attack, Donald Has Access To Some Plaintext/Ciphertext Pairs additionally To The Intercepted Ciphertext That He needs to interrupt.The plaintext/ciphertext pairs are collected earlier.

- **Chosen-Plaintext Attack**

  The chosen-plaintext attack is analogous to the known-plaintext attack, however the plaintext/ciphertext combines are chosen by offender herself.

Chapter 3

literature Survey

3.1 Introduction

The high growth in the networking technology leads a common culture for interchanging of the digital images very drastically. Hence it is more vulnerable of duplicating of digital image and re-distributed by hackers. Therefore the images has to be protected while transmitting it, Sensitive information like credit cards, banking transactions and social security numbers need to be protected. For this many encryption techniques are existing which are used to avoid the information theft. In recent days of Internet, the encryption of data plays a major role in securing the data in online transmission focuses mainly on its security across the internet. Different encryption techniques are used to protect the confidential data from unauthorized use.

Encryption is a very common technique for promoting the image security. Image encryption, video encryption, chaos based encryption have applications in many fields including the internet communication, multimedia systems, medical imaging, Tele-medicine and military Communication, etc. The evolution of encryption is moving towards a future of endless possibilities. Everyday new methods of encryption techniques are discovered

Literature Survey:

Evaluating the Effects of Cryptography Algorithms on power consumption for wireless devices has done by D. S. Abdul. Elminaamet.al., (2009) presents a performance evaluation of selected symmetric encryption algorithms on power consumption for wireless devices. Several points can be concluded from the Experimental results. First; in the case of changing packet size with and with out transmission of data using different architectures and different WLANs protocols, it was concluded that Blowfish has better performance than other common encryption algorithms used, followed by RC6.

DES and 3DES are known to have worm holes in their security mechanism, Blowfish and AES do not have any so far[6].

Evaluation Of Performance Characteristics Of Cryptosystem Using Text Files designed by challaNarasimham and JayaramPradhan(2008)- They performed the performance comparison for variable sized text files as input. An analysis on computational running times results in significant difference among the methods. He believe in that the performance of DES, especially in decryption method is very high than the alternatives. Despite the key distribution, DES is more suitable to the application, which has the decryption as the highest priority.He has proposed and performed the test cases on the two PKCS methods i.e., RSA and NTRU Though the encryption, decryption and complexity are high in NTRU, the RSA provides the highest security to the business application. He presented all these parameters with computational running times for all the methods, so as to select the appropriate method[7].

Abdel-Karim and his colleague Al Tamimi presented simulation results showed that Blowfish has a better performance than other common encryption algorithms used. Since Blowfish has not any known security weak points so far, which makes it an excellent candidate to be considered as a standard encryption algorithm. AES showed poor performance results compared to other algorithms since it requires more processing power.

P. Prasithsangaree and his collegeue P. Krishnamurthy have analyzed the Energy Consumption of RC4 and AES Algorithms in Wireless LANs in the year 2003.They have evaluated the performance of RC4 and AES encryption algorithms. The performance metrics were encryption throughput, CPU work load, energy cost and key size variation. Experiments show that the RC4 is fast and energy efficient for encrypting large packets. However, AES was more efficient than RC4 for a smaller packet size. From the results, it appears that we can save energy by using a combination of RC4 and AES to provide encryption for any packet size. The tradeoffs with security are not completely clear[9].

Comparative Analysis of AES and RC4 Algorithms for Better Utilization has designed by NidhiSinghal, J.P.S.Raina in the year (2011).The performance metrics were throughput, CPU process time, memory utilization, encryption and decryption time and key size variation. Experiments show that the RC4 is fast and energy efficient for encryption and decryption. Based on the analysis done as part of the research, RC4 is better than AES.we compare the encryption time of AES and RC4 algorithm over different packet size. RC4 takes less time to encrypt files

w.r.t. AES. In AES, CFB and CBC takes nearly similar time but ECB takes less time then both of these[10].

Efficiency and Security of Some Image Encryption Algorithms MarwaAbd El-Wahed et.al (2008) – worked in this paper, four image encryption algorithms have been studied by means of measuring the encryption quality, the memory requirement, and the execution time of the encryption. In addition, the security analysis of these schemes is investigated from cryptographic viewpoint; statistical and differential attacks. The results are compared, focusing on those portions where each scheme is performed differently

A Comparative Study Of Two Symmetric Encryption Algorithms Across Different Platforms *designed by* S.A.M Rizvi1 et.al., All algorithms run faster on Windows XP. The CAST runs slower than AES for text. Blowfish encrypts images most efficiently on all 3 platforms, even CAST runs faster on Windows XP for image data. But on Windows Vista and Windows7, AES and CAST perform at the similar speed .CAST performs better than BLOWFISH and AES on Windows XP for encrypting audio files, but on Windows Vista and Windows7, there is no significant difference in performance of CAST and AES, however BLOWFISH encrypts audio files at less speed for audio files[12].

ThroughPut Analysis of Various Encryption Algorithms presented by Gurjeevan Singh et al.,(2011)- For experiment a Laptop with 2.20 GHz C.P.U., 4GB RAM Core-2-Dou Processor and Windows 7 Home Premium (32-Bit) is used in which the performance data are collected. In this experiment software encrypts the text file size that ranges from 20 Kb to 99000 Kb. Their implementation is thoroughly tested and is optimized to give the maximum performance for the algorithm.The performance matrices are throughput. The throughput of encryption as well as decryption schemes is calculated but one by one. In the case of Encryption scheme throughput is calculated as the average of total plain text in k bytes divided by the average Encryption time and in the case of Decryption scheme throughput is calculated as the average of total cipher text is divided by the average Decryption time. This work presents the performance evaluation of selected symmetric algorithms. The selected algorithms are AES, 3DES, Blowfish and DES.

The presented simulation results show the numerous points. Firstly it was concluded that Blowfish has better performance than other algorithms followed by AES in terms of throughput. Secondly 3DES has least efficient of all the studied algorithms[15].

ShashiMehrotra Seth and her collegueRajanMishra(2011) jointly has done a Comparative Analysis Of Encryption Algorithms For Data Communication.The authors analyse the performance of encryption algorithm is evaluated considering the following parameters like Computation Time, Memory usage and Output Bytes, RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm[17].

DiaaSalamaAbdElminaametal.,(2010) [18]eavaluate the Performance of Symmetric Encryption Algorithms like AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. DiaaSalama et.al jointly done a research work in the title "Wireless Network Security Still Has no Clothes "[19]. The above research work evaluate the performanace of most common symmetrical encryption algorithms like AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6.

Ruangchaijatupon.P and his collogue Krishnamurthy.P(2001)[20] has done a research work on "Encryption and Power Consumption in Wireless LANs".

3.2 Fast implementation of RSA with Java:

In the setting of electronic commerce systems should run on a totally different package platforms like windows, unix, and Linux. If the RSA is developed with C language, it'll talk about a drug that the system cannot run on all platforms, though the system developed with c language is straightforward to transplant. As an equivalent data type has totally different|completely different} length on different platforms, this makes it tough to transplant. To make it simple to maintain the system, develop quickly and to unravel the matter of cross platforms. Author tries to develop this method with Java language. [25] Java is an associate degree object orientating language, and the system developed with Java language will run all platforms.

3.3 NEW COUNTERMEASURE SCHEME

Giraud [7] proposed a new countermeasure scheme based on the concept of Montgomery Ladder Exponentiation. The proposed algorithm performs two modular multiplications for each bit of exponent. Whereas the square and multiply algorithm which performs on average 1.5 modular multiplications per bit of the exponent. Therefore the proposed method is faster.

3.4 Possible Attacks on RSA signature:

- **Factorization**

The Problem of whole number factorization is one amongst the oldest in number theory. However, the safety of the many cryptanalytic techniques depends upon the intractableness of the number factorization drawback.[8]

If AN somebody is ready to issue the general public modulus n of some entity A, then the somebody will reckons φ(n) so, mistreatment the extended Euclidean algorithm, deduce the personal key d from φ(n) and public exponent e by finding ed≈1 (mod φ(n)),this constitutes a complete break of the system.

- **Existential forgery**

The basic plan behind RSA signature is to calculate $s=M^d \pmod n$ wherever M is (some operate of) the message. This suggests that associate degree resistor will select an associate degree discretionary s* could be a valid signature on m*.

This is one reason why an associate RSA signature is usually either of the forms

(a)S=(h(m))d(mod n),where he's a 1 manner collision resistant hash perform,giving a signature with appendix,or

(b) S=(R(m))d(mod n), wherever R could be a redundancyadding perform,giving a signature with message recovery for a message m of restricted length.

3.5 RSA Algorithm and its Security Issues

RSA key of length 1024 can be generated within two minutes on the platform of a common PC [21]. On the other hand, encryption/decryption operation on data less than 1024 bits can be done within two seconds. So we can say that the actual efficiency of the RSA based system is improved.

It gives the guarantees for the point of implementation high security, RSA algorithm using a long key length on the platform of any PC not a particular PC [26]. There may be various known attacks to break the security of the RSA algorithm, ―Brute force attack‖, which is a special kind of attack who does not care of any special parameters.

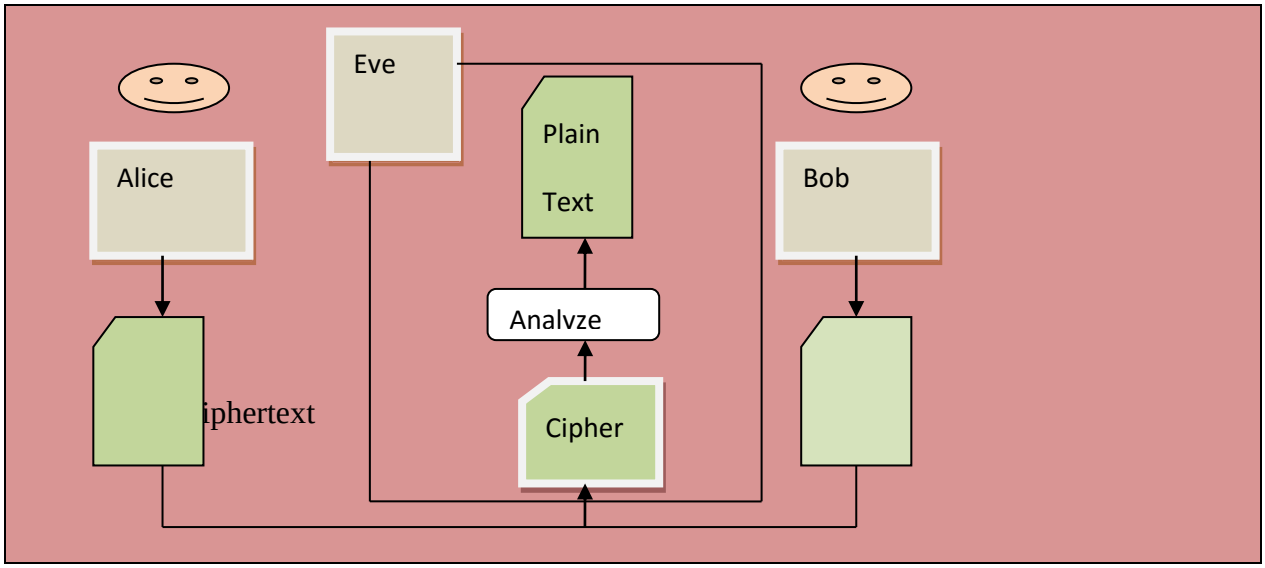

Fig 3.1. A broad level steps in IDEA

However, it is as also partitioned into two categories: Exhaustive attack & Factorization attack. The second type of attack is ―Subtle attack‖ who aims at the mathematic feature of some parameters [24].

We use the RSA algorithm for digital signature point of view. A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. If any digital signature is valid then it gives a recipient reason to give trust that the message was created by a known sender and during transformation, it was not altered by a third person.

Now consider two employees A & B. A have some public data taken from its own cloud. Suppose there are two employees A & B of different enterprises. A wants to send some message to B, then there are following steps:

A takes a document from the cloud, which B wants. Using Hash function this message is transferred into the message digest form. A's software then encrypts the message digest with his private key i.e. Digital signature. Using RSA, A will encrypt with B's public key & B will decrypt it with his private key and A's public key for verification of signature [21].

If we have to modify (or develop an algorithm) in an Encryption key (E) and Decryption key (D) such that all the functioning should be depends upon the Digital Signature as a software system. Then we have been obtaining results would be very optimal/optimum as well as the secrecy and authentic. To calculate encryption and decryption key using RSA algorithm is very complex, so if we discover some such type of algorithm so that these calculations become easy.

If this idea becomes successful, then the time complexity of the algorithm can be reduced as a result processing becomes faster and there is a quite difficult job to break key for hackers and crackers.

3.6 Different types of Mathematical Attacks on RSA Algorithm

Cryptanalysis is that the connected study of breaking decipher message. In associate cryptography theme, the main objective of the assailant is to recover the plaintext m from the connected cipher text. If he/she is flourishing, we are saying he has broken the RSA cryptosystem. In the case of digital signature, the goal of the offender is to forge signatures. At a lot of bold attack is to recover the non-public key d. If achieved, the offender will currently decipher all cipher texts and forge signatures at can. During this case the sole answer to revocation of the key.

3.7 Performance of RSA Algorithm with ECC

RSA has various security issues and general considerations based on mathematical calculations.[9]

RSA is the best algorithm for security purpose but it's key length is too large so to decrypt any message there is too much wastage of time and energy. So if some concept of ECC may be added then it will give better response for security as well as complexity point of view because ECC is strongest concept having higher security level than RSA and it is easy to use. Due to the recent development in field of factoring of large prime, the key length for secure RSA has increased. The increment in the length can increase the security of the RSA Cryptography, but it requires extra communicational, computational cost [22]. When we calculate multiplicative inverse of an element in GF (p) for small values of p, it is very easy. But when we calculate it for larger numbers then RSA becomes very complex so Euclid's algorithm can be extended for this purpose.

3.8 RSA Algorithm with Key Size

Large key size have two effects: regular increase in computing power and continuing refinement of factoring power [23]. Cyber crime's can be felt when we use internet and cloud computing offers a tempting target for many reasons. There are some providers such as Google and Amazon which having existing infrastructures to detect and survive a cyber attack. If a

cyber criminal can identify the provider whose vulnerabilities are the easiest to exploit, then it is a highly visible target [Uma Somani et.al].

3.9 RSA security depends on the Size of Prime number

The security of RSA algorithm depends on the size of prime number, for security purpose we select a very large prime number n, and we have some efficient methods to divide it. The calculation of private key e, similarly d can't be calculated from n and e. The attack is difficulty equivalence to the division of the product of two very large prime numbers say p, q, however the RSA having the higher security [3,21,4]. The private key e is used to encrypt when we are sending any plaintext message to others.

3.10 Blowfish has Better Performance than Other Common Encryption Algorithms

Evaluating the Effects of Cryptography Algorithms on power consumption for wireless devices has done by D. S. Abdul. Elminaam et.al. (2009) presents a performance evaluation of selected symmetric encryption algorithms on power consumption for wireless devices. Several points can be concluded from the Experimental results. First; in the case of changing packet size with and without transmission of data using different architectures and different WLANs protocols, it was concluded that Blowfish has better performance than other common encryption algorithms used, followed by RC6.

Abdel-Karim and his colleague Al Tamimi presented simulation results showed that Blowfish has a better performance than other common encryption algorithms used. Since Blowfish has not any known security weak point so far. AES showed poor performance results compared to other algorithms since it requires more processing power.

DES and 3DES are known to have worm holes in their security mechanism; Blowfish and AES do not have any so far [6].

3.11 An analysis on computational running times results in significant difference among the Security Algorithm

Evaluation of Performance Characteristics of Cryptosystem Using Text Files designed by challa Narasimham and Jayaram Pradhan (2008) - They performed the performance

comparison for variable sized text files as input. An analysis on computational running times results in significant difference among the methods. He believe in that the performance of DES, especially in decryption method is very high than the alternatives. Despite the key distribution, DES is more suitable to the application, which has the decryption as the highest priority. He has proposed and performed the test cases on the two PKCS methods i.e., RSA and NTRU Though the encryption, decryption and complexity are high in NTRU, the RSA provides the highest security to the business application. He presented all these parameters with computational running times for all the methods, so as to select the appropriate method [7].

3.12 Comparisons of RC4 and AES

P. Prasithsangaree and his collegeue P. Krishnamurthy have analyzed the Energy Consumption of RC4 and AES Algorithms in Wireless LANs in the year 2003.They have evaluated the performance of RC4 and AES encryption algorithms. The performance metrics were encryption throughput, CPU work load, energy cost and key size variation. Experiments show that the RC4 is fast and energy efficient for encrypting large packets. However, AES was more efficient than RC4 for a smaller packet size. From the results, it appears that we can save energy by using a combination of RC4 and AES to provide encryption for any packet size. The tradeoffs with security are not completely clear [9].

Comparative Analysis of AES and RC4 Algorithms for Better Utilization has designed by Nidhi Singhal, J.P.S.Raina in the year (2011).The performance metrics were throughput, CPU process time, memory utilization, encryption and decryption time and key size variation. Experiments show that the RC4 is fast and energy efficient for encryption and decryption. Based on the analysis done as part of the research, RC4 is better than AES. We compare the encryption time of AES and RC4 algorithm over different packet size. RC4 takes less time to encrypt files with respect to AES. In AES, CFB and CBC takes nearly similar time but ECB takes less time then both of these [10].

3.13 Image Encryption Algorithms

[29] four image encryption algorithms have been studied by means of measuring the encryption quality and the execution time of the encryption. In addition, the security analysis of these schemes is investigated from cryptographic viewpoint; statistical and differential attacks.

The results are compared, focusing on those portions where each scheme is performed differently.

3.14 A Comparative Study of Two Symmetric Encryption Algorithms across Different Platforms

A Comparative Study of Two Symmetric Encryption Algorithms across Different Platforms *designed by* S.A.M Rizvi1 et.al. All algorithms run faster on Windows XP. The CAST runs slower than AES for text. Blowfish encrypts images most efficiently on all 3 platforms, even CAST runs faster on Windows XP for image data. But on Windows Vista and Windows7, AES and CAST perform at the similar speed .CAST performs better than BLOWFISH and AES on

3.15 Throughput Analysis of Various Encryption Algorithms

Throughput Analysis of Various Encryption Algorithms presented by Gurjeevan Singh et al.,(2011)- For experiment a Laptop with 2.20 GHz C.P.U., 4GB RAM Core-2-Dou Processor and Windows 7 Home Premium (32-Bit) is used in which the performance data are collected. In this experiment software encrypts the text file size that ranges from 20 Kb to 99000 Kb. Their implementation is thoroughly tested and is optimized to give the maximum performance for the algorithm. The performance matrices are throughput. The throughput of encryption as well as decryption schemes is calculated but one by one. In the case of Encryption scheme throughput is calculated as the average of total plain text in k bytes divided by the average Encryption time and in the case of Decryption scheme throughput is calculated as the average of total cipher text is divided by the average Decryption time. This work presents the performance evaluation of selected symmetric algorithms.

3.16 Encryption Algorithms for Data Communication

Shashi Mehrotra Seth and her collegue Rajan Mishra (2011) jointly has done a Comparative Analysis Of Encryption Algorithms For Data Communication.The authors analyze the performance of encryption algorithm is evaluated considering the following parameters like Computation Time, Memory usage and Output Bytes, RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm [17].

3.17 Performance of Symmetric Encryption Algorithms

Diaa Salama Abd Elminaam etal.,(2010) [18]eavaluate the Performance of Symmetric Encryption Algorithms like AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. Diaa Salama et.al jointly had done a research work in the title “Wireless Network Security Still Has no Clothes” [19]. The above research work evaluates the performance of most common symmetrical encryption algorithms like AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6.

Ruangchaijatupon.P and his collogue Krishnamurthy.P (2001) [20] has done a research work on "Encryption and Power Consumption in Wireless LANs”.

Authors [6] proposed a new methodology to change the original modulus with the fake modulus. Therefore if the hacker factorizes this new modulus value then he will not be able to locate the original decryption key. But there are some Common Problems in Existing RSA Variants:

1) The main disadvantage of RSA encryption its slower speed.
2) Not secure against weiner’s attack.
3) Not secure against common modulus attack.
4) Not secure against known plaintext attack.
5) Not secure against low decryption exponent attack.

Chapter 4

Related Work & Problem Definition:

Related Work

RSA key of length 1024 can be generated within two minutes on platform of a common PC [21]. On the other hand, encryption/decryption operation on data less than 1024 bits can be done within two seconds. So we can say that the actual efficiency of RSA based system is improved. It gives the guarantees for the point of implementation high security RSA algorithm using long key length on the platform of any PC not a particular PC [25]. There may be various known attacks to break the security of RSA algorithm, ―Brute force attack‖, which is a special kind of attack who does not care of any special parameters. However it is as also partitioned into two categories: Exhaustive attack & Factorization attack. Second type of attack is ―Subtle attack‖ who aims at the mathematic feature of some parameters [24]. We use RSA algorithm for digital signature point of view. A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. If any digital signature is valid then it gives a recipient reason to give trust that the message was created by a known sender and during transformation, it was not altered by third person. Now consider two employees A & B. A have some public data taken from its own cloud. Suppose there are two employees A & B of different enterprises. A wants to send some message to B, then there are following steps:

A takes a document from cloud, which B wants.

Using Hash function this message is transferred into message digest form.

A's software then encrypts the message digest with his private key ie. Digital signature.

Using RSA, A will encrypt with B's public key & B will decrypt it with his private key and A's public key for verification of signature [21].

If we have to modify (or develop an algorithm) in an Encryption key (E) and Decryption key (D) such that all the functioning should be depends upon the Digital Signature as a software system. Then we have obtaining results would be very optimal/optimum as well as secrecy and authentic. To calculate encryption and decryption key using RSA algorithm is very complex so if we discover some such type of algorithm so that these calculations become easy. If this idea become successful then the time complexity of algorithm can be reduced as a result processing becomes

faster and there is quite difficult job to break key for hackers and crackers. RSA have various security issues and general considerations based on mathematical calculations.RSA is the best algorithm for security purpose but it's key length is too large so to decrypt any message there is too much wastage of time and energy. So if some concept of ECC may be added then it will give better response for security as well as complexity point of view because ECC is strongest concept having higher security level than RSA and it is easy to use. Due to the recent development in field of factoring of large prime, the key length for secure RSA has increased. The increment in the length can increase the security of the RSA Cryptography, but it requires extra communicational, computational cost [22]. When we calculate multiplicative inverse of an element in GF(p) for small values of p, it is very easy. But when we calculate it for larger numbers then RSA becomes very complex so Euclid's algorithm can be extended for this purpose. Large key size have two effects: regular increase in computing power and continuing refinement of factoring power [23]. Cyber crime's can be felt when we use internet and cloud computing offers a tempting target for many reasons. There are some providers such as Google and Amazon which having existing infrastructures to detect and survive a cyber attack. If a cyber criminal can identify the provider whose vulnerabilities are the easiest to exploit, then it is a highly visible target [Uma Somani et.al]. The security of RSA algorithm depends on the size of prime number, for security purpose we select a very large prime number n, and we have some efficient methods to divide it. The calculation of private key e, similarly d can't be calculated from n and e. The attack is difficulty equivalence to the division of the product of two very large prime numbers say p, q, however the RSA having the higher security [21]. The private key e is used to encrypt when we are sending any plaintext message to others.

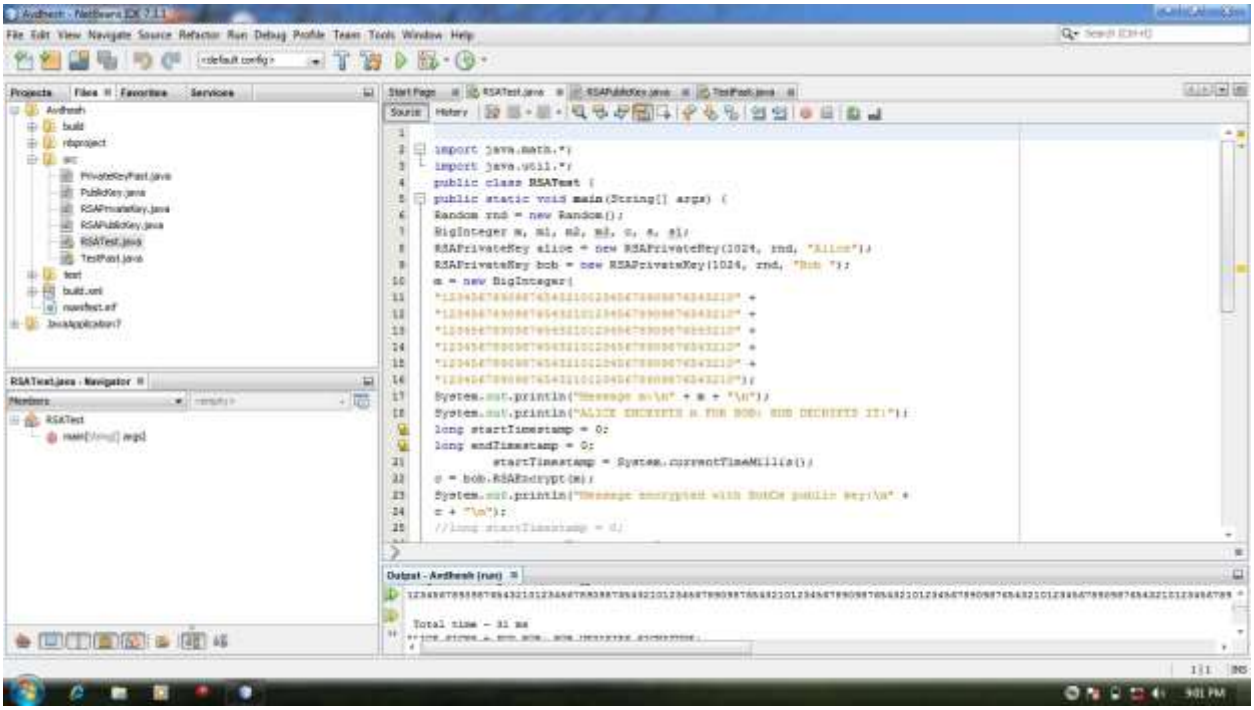

Figure 4.1 Related work to RSA

Problem Definition:

There are so many algorithms present to encrypt and decrypt the data for security purpose in cryptography. RSA is the most common algorithm for encryption and decryption. But still the current versions are slow and less secure. The objective of the proposed algorithm is to propose a faster variant of RSA algorithm & also make it more secure against Weiners attack.

Common Problems in Existing RSA Variants:

- The main disadvantage of RSA encryption its slower speed
- Not secure against wiener's attack
- Problem arise to common modulus attack
- known plaintext attack are possible
- Low decryption exponent attack if we know the decryption exponent.
- The decryption time slow.

Proposed Approach:-

The steps of the proposed work is as follows:

1. First choose random large prime integers p and q of roughly the same size but not too close to each other.
2. Calculate the product n = pq (ordinary integer multiplication)
3. Choose a random encryption exponent e It must not has any common factor with either p-1 or q-1.
4. Compute ed mod (p-1) * (q-1) = 1
5. Encryption Step:

   $c = m^e \bmod n$
6. Decryption Step:

In this step, we will use the larger value of d. Also we will split the n in to p and q. Then we will compute the plain text by applying the Fermat's theorem as follows:

- First compute

  $X1 = c^{dp} \bmod p$

  $X2 = c^{dq} \bmod q$

  Where dp = d mod p-1

  &

  dq = d mod q-1
- The compute

  W = (X2 – X1) * W1 mod q

  Where W1 = p modinverse q
- Then finally compute

  $M = c^d \bmod n = X1 + W * p$

Chapter 5

Proposed Solution

Proposed Cryptosystem:

The steps of the proposed work is as follows:

1. First choose random large prime integers p and q of roughly the same size but not too close to each other.
2. Calculate the product n = pq (ordinary integer multiplication)
3. Choose a random encryption exponent e It must not has any common factor with either p-1 or q-1.
4. Compute ed mod (p-1) * (q-1) = 1
5. Encryption Step:

   $c = m^e \bmod n$
6. Decryption Step:

In this step, we will use the larger value of d. Also we will split the n in to p and q. Then we will compute the plain text as follows:

- First compute

  $X1 = c^{dp} \bmod p$

  $X2 = c^{dq} \bmod q$

  Where dp = d mod p-1

  &

  dq = d mod q-1
- The compute

  W = (X2 – X1) * W1 mod q

  Where W1 = pmodinverse q
- Then finally compute

$M = c^d \bmod n = X1 + W * p$

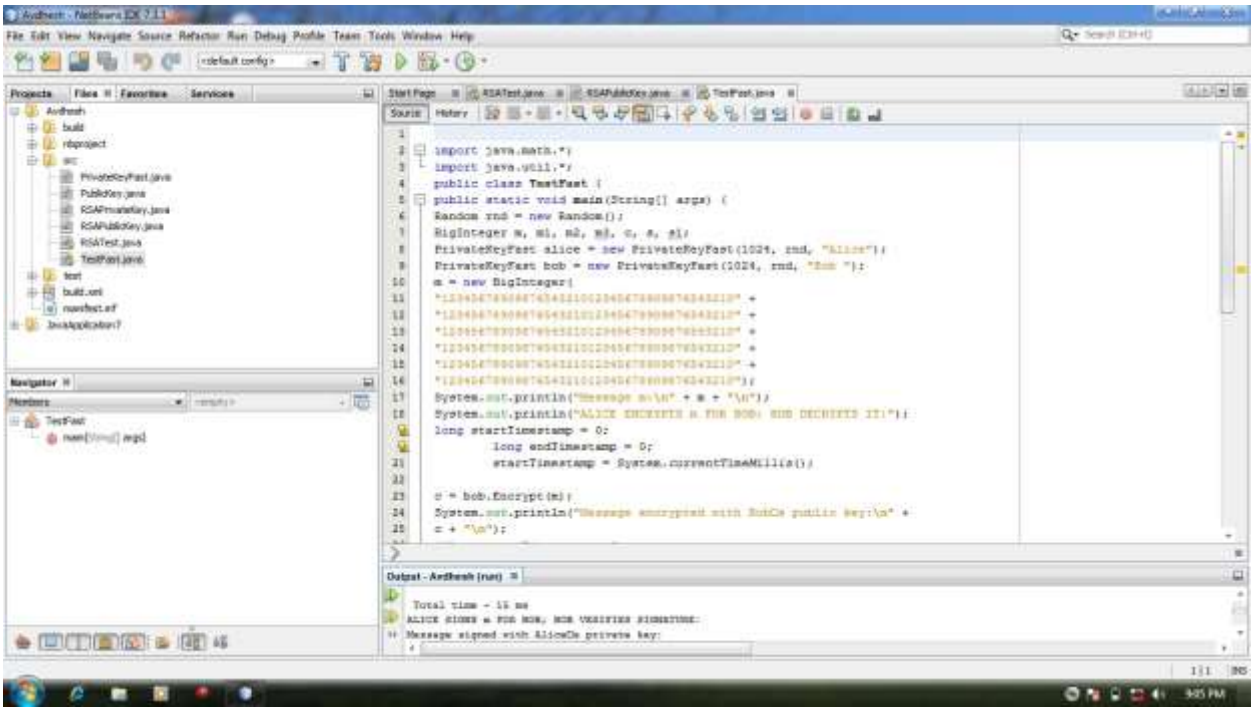

Figure 5.1 Proposed Cryptosystem

Advantages of Proposed Cryptosystem:

1. In our proposed cryptosystem, the encryption is faster in comparison to current variants of RSA cryptosystem because we are using e = 3. The binary representation of 3 is 011. It contains only two 1's. so it takes only 17 multiplications to exponentiate.
2. Also our proposed cryptosystem is more secure against low decryption exponentiation attack, because we are using a large value of d.

Chapter 6

Results

run:

Message m:

123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210

ALICE ENCRYPTS m FOR BOB; BOB DECRYPTS IT:

Message encrypted with Bob's public key:

711492378845477619191708088766200701351077166127171033382320896929170305746039193485326711153506005944653224992160537028033610267801429526858236542153463443832032195700906662475112683718235937580631283996205583343581584804986466495522628653104283783865353181249316704256737402155887619782261297658664602008 5

Original message back, decrypted:

123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210

Total time ~ 31 ms

Proposed

run:

Message m:

123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210

ALICE ENCRYPTS m FOR BOB; BOB DECRYPTS IT:

Message encrypted with Bob's public key:

262872053510487674852263341587993594247687779003232603826179291419410357822457674640586431536289445246649689946584184507788863110827510687433992853681777352085256146034342158633662689872676161239406272696760692230482528757598145409293093564983723376702620948099200490136948121093910115916808075275048251276 31

Original message back, decrypted:

123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210

Total time ~ 15 ms

Plain Text Used For Experimental Study:

12345678909876543210123456789098765432101234567890987654321012345678909876543210123456789098765432101234567890987654321012345678909876543210

123456789098765432101234567890987654321 0

123456789098765432101234567890987654321 0

Method	Decryption Time
RSA Existing Variant	31ms
Proposed RSA Variant	15ms

Table 6.1: Performance Comparison

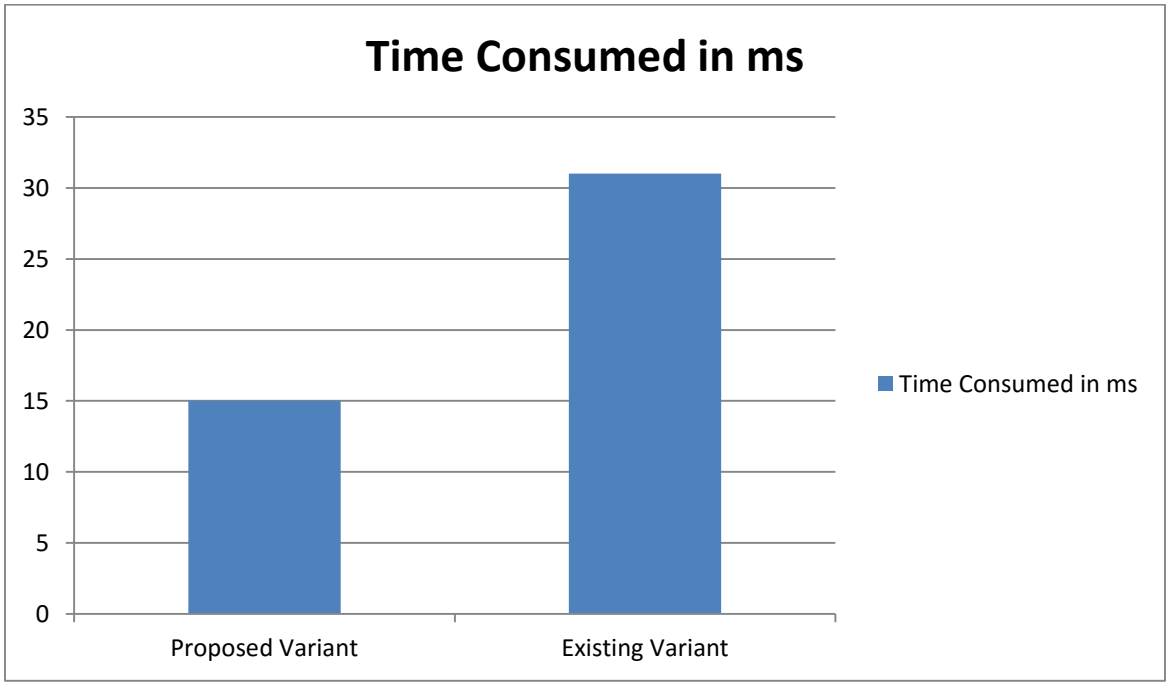

Figure 6.1: Proposed Variant v/s Existing Variant

Chapter 7

Conclusion

In this report the existing encryption techniques are studied and analyzed well to promote the performance of the encryption methods also to ensure the security proceedings. To sum up, all the techniques are useful for real-time encryption. Each technique is unique in its own way, which might be suitable for different applications. Everyday new encryption technique is evolving hence fast and secure conventional encryption techniques will always work out with high rate of security.

In this report, we discussed some existing variants of RSA cryptosystem. We also proposed a novel cryptosystem. The proposed cryptosystem is faster & it is also more secure.

References:

[1] William Stallings " Network Security Essentials (Applications and Standards)", Pearson Education, 2004.

[2] National Bureau of Standards, " Data Encryption Standard," FIPS Publication 46, 1977.

[3] Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard." Dr. Dobb's Journal, March 2001.

[4] Ramesh G, Umarani. R, " Data Security In Local Area Network Based On Fast Encryption Algorithm",International Journal of Computing Communication and Information System(JCCIS) Journal Page 85-90. 2010.

[5]DiaaSalama Abdul Minaam, Hatem M. Abdual-Kader, and Mohiy Mohamed Hadhoud "Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types" International Journal of Network Security, Vol.11, No.2, PP.78-87, Sept.

[6] SimarPreet Singh, and Raman Maini "COMPARISON OF DATA ENCRYPTION ALGORITHMS" International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp. 125-127

[7]ChallaNarasimham, JayaramPradhan," EVALUATION OF PERFORMANCE CHARACTERISTICS OF CRYPTOSYSTEM USING TEXT FILES" Journal of Theoretical and Applied Information Technology,pp55-59 2008.

[8] Abdel-Karim Al Tamimi," Performance Analysis of Data Encryption Algorithms "

[9] Prasithsangaree.P and Krishnamurthy.P(2003), "Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs," in the Proceedings of the
IEEE GLOBECOM 2003, pp. 1445-1449.

[10] Nidhi Singhal1, J.P.S.Raina2, Comparative Analysis of AES and RC4 Algorithms for Better Utilization", International Journal of Computer Trends and Technology- July to Aug Issue 2011 pp177-181.

[11] MarwaAbd El-Wahed, SalehMesbah, and Amin Shoukry," Efficiency and Security of Some Image Encryption Algorithms", Proceedings of the World Congress on Engineering 2008 Vol I WCE 2008, July 2 - 4, 2008, London, U.K.

[12] Dr. S.A.M Rizvi1 ,Dr. Syed Zeeshan Hussain2 and Neeta Wadhwa" A Comparative Study Of Two Symmetric Encryption Algorithms Across Different Platforms",

[13] Turki Al-Somani ,Khalid Al-Zamil "Performance Evaluation of Three Encryption/Decryption Algorithms on the SunOS and Linux Operating Systems", Theses

[14] 1Gurjeevan Singh, 2Ashwani Kumar Singla, 3K.S. Sandha," Through Put Analysis of Various Encryption Algorithms", IJCST Vol. 2, Issue 3, September 2011

[15] Gurjeevan Singh, Ashwani Kumar Singla,K.S. Sandha, "Through Put Analysis Of Various Encryption Algorithms", IJCST Vol. 2, Issue 3, September 2011.

[16] R.Chandramouli, "Battery power-aware encryption – ACM Transactions on Information and System Security (TISSEC),'' Vol. 9 Issue 2, May 2006.

[17] 1Shashi Mehrotra Seth, 2Rajan Mishra," Comparative Analysis Of Encryption Algorithms For Data Communication", IJCST Vol. 2, Issue 2, June 2011 pp.192-192.

[18] DiaaSalamaAbd Elminaam1, Hatem Mohamed Abdual Kader2, and Mohiy Mohamed Hadhoud2," Evaluating The Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol.10, No.3, PP.213{219, May 2010.

[19] Diaa Salama1, HatemAbdual Kader2, and Mohiy Hadhoud2" Wireless Network Security Still Has no Clothes", International Arab Journal of e-Technology, Vol. 2, No. 2, June 2011 pp.112-123.

[20].N.Ruangchaijatupon and P. Krishnamurthy, "Encryption and power consumption in wireless LANs-N,"The Third IEEE Workshop on Wireless LANs,

[21]Chong Fu, Zhi-liang Zhu, Sch. of Inf. Sci. & Eng., Northeastern Univ., Shenyang 110004,P.R.China. [22]William Stallings, ―Cryptography and Network Security: Principles and practices, Dorling Kindersley (india) pvt ltd., 4th edition(2009)

[23]AtulKahate ―Cryptography and Network Security‖ 3rd edition.

[24]Jiezhaopeng, Qi Wu, Jiangxi University of finance & Economics, Nanchang 330013,Jiangxi province,China ―Research and implementation of RSA Algorithm in Java‖.

[25]HongweiSi,YoulinCai, Zhimei Cheng, ―An improved RSA algorithm based on Complex numeric operation function‖.