Bangladesh University of Business & Technology (BUBT)

A project report on

"Rsa encryption using openssl"

Submitted by

Pallab Majumdar (ID: 18192103050)

Al Ahad Sufian (ID: 18192103056)

Joy Adhikary (ID: 18192103062)

Mafuja Akter Mitu (ID: 18192103068)

Habibullah (ID: 18192103080)

Submitted to

Dr.Abdullah- Al-Musa

Lecturer

Department of Computer Science & Engineering

Acknowledgment

We would like to pay our gratitude to the Almighty Allah who created us with all the abilities to understand analysis and develop the process with patience. We are thankful to our supervisor Dr.Abdullah- Al-Musa, Lecturer, Computer Science and Engineering Department, Bangladesh University of Business and Technology for his professional guidance and motivation during the work of this project which is a major part of it. Without his valuable support and guidance, this project could not reach this level of development from our point of view.

We would like to thank all the Faculty members, Department of CSE, Bangladesh University of Business and Technology for their valuable time spend in requirements analysis and evaluation of the project work. We would like to express our sincere and warm gratitude to all those who have encouraged us directly, provided mental encouragement and criticized our work in several phases during the development of this project and for preparing this project indirectly.

Abstract

Security requirements will change time to time. It is always necessary to provide appropriate security services to any communication. Later 1970's many such mechanism has come. One among that is public key cryptography. There are few end-users today who make use of real security applications. These applications tend to be too complicated, exposing too much detail of the cryptographic process. Users need simple inherent security that doesn't require more of them simply clicking the secure checkbox. Cryptography is a first abstraction to separate specific algorithms from generic cryptographic processes in order to eliminate compatibility and upgradeability problems. The core idea is enhance the security of RSA algorithm. The invention of RSA algorithm brings one of the significant improvements in the field of cryptography. Until that period, only symmetric key cryptography was in act which used only one key for both encryption and decryption. This makes that single key as the most sensitive one. Because of that key will be disclosed then entire communication will be compromised irrespective of the complexity of conversion applied over it. That is the reason why when this idea was proposed; it opened another window in the security providing arena. Hence the researchers have shown lot of interest in developing similar algorithms which gain some momentum over a period of time. At the same time this new approach has also started facing various hurdles in the name of various attacks. In that way this approach had a long history of over 40 years. In this dissertation public key algorithm RSA and enhanced RSA are compared analysis is made on time based on execution time.

On the other side some other approaches like elliptic curve cryptography also proposed based on the idea of public key cryptography. The purpose of this project is to analyze the present scenario of RSA algorithm along with its past history in the modern cryptography era. Also this project presents various related work done over a period of time related to this algorithm

along with the summary of all the works. In this dissertation public key algorithm RSA and enhanced RSA using openssl are compared analysis is made on time based on execution time.

Declaration

We hereby declare that the project on Rsa encryption using openssl submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT) is our own work and that it contains no material which has been accepted for the award to the candidate(s) of any other degree or diploma, except where due reference is made in the text of the project. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the project.

Al Ahad Sufian
ID: 18192103056

Pallab Majumdar
ID: 18192103050

Habibullah
ID:18192103080

Joy Adhikary
ID: 18192103062

Mafuja Akter Mitu
ID: 18192103068

Dedication

Dedicated to our parents, teachers, friends and who loved us for all their love and inspiration.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In today's networked world, many applications need security, and cryptography is one of the primary tools for providing that security. The primary goals of cryptography, data confidentiality, data integrity, authentication, and non-repudiation (accountability) can be used to thwart numerous types of network-based attacks, including eavesdropping, IP spoofing, connection hijacking, and tampering. OpenSSL is a cryptographic library; it provides implementations of the industry's best-regarded algorithms, including encryption algorithms such as 3DES ("Triple DES"), AES and RSA, as well as message digest algorithms and message authentication codes.

Using cryptographic algorithms in a secure and reliable manner is much more difficult than most people believe. Algorithms are just building blocks in cryptographic protocols, and cryptographic protocols are notoriously difficult to get right. Cryptographers have a difficult time devising protocols that resist all known attacks, and the average developer tends to do a lot worse. For example, developers often try to secure network connections simply by encrypting data before sending it, then decrypting it on receipt. That strategy often fails to ensure the integrity of data. In many situations, attackers can tamper with data, and sometimes even recover it. Even when protocols are well designed, implementation errors are common. Most cryptographic protocols have limited applicability, such as secure online voting. However, protocols for securely communicating over an insecure medium have ubiquitous

applicability. That's the basic purpose of the SSL protocol and its successor, TLS (when we generically refer to SSL, we are referring to both SSL and TLS): to provide the most common security services to arbitrary (TCP-based) network connections in such a way that the need for cryptographic expertise is minimized.

Ultimately, it would be nice if developers and administrators didn't need to know anything about cryptography or even security to protect their applications. It would be nice if security was as simple as linking in a different socket library when building a program. The OpenSSL library strives toward that ideal as much as possible, but in reality, even the SSL protocol requires a good understanding of security principles to apply securely. Indeed, most applications using SSL are susceptible to attack. Nonetheless, SSL certainly makes securing network connections much simpler. Using SSL doesn't require any understanding of how cryptographic algorithms work. Instead, you only need to understand the basic properties important algorithms have. Similarly, developers do not need to worry about cryptographic protocols; SSL doesn't require any understanding of its internal workings in order to be used. You only need to understand how to apply the algorithm properly.

The goal of this project is to document the OpenSSL library and how to use it properly. This is a book for practitioners, not for security experts. We'll explain what you need to know about cryptography in order to use it effectively, but we don't attempt to write a comprehensive introduction on the subject for those who are interested in why cryptography works.

1.2 Problem Statement

The main problem is that while conversing between multiple host or end point there is possibility is that the the communication can affected by the man-in-the-middle(MITM) attack.

- The existing banking system has less security.
- Messaging app is less secured.
- The encryption system different organization use is not that secure compare to RSA.
- The existing system is not that user friendly.

1.3 Objectives

- The main objective behind creating this project is making the communication between multiple host is more secured by using well known algorithm named RSA algorithm.
- Our project is very easy to use and it's a full automatic process created using bash script.
- The directories and file created by our system is very well organized and identifiable.

1.4 Purpose

The primary goal of cryptography is to secure important data as it passes through a medium that may not be secure itself. Usually, that medium is a computer network. There are many different cryptographic algorithms, each of which can provide one or more of the following services to applications:
Confidentiality (secrecy).
Data is kept secret from those without the proper credentials, even if that data travels through an insecure medium. In practice, this means potential attackers might be able to see garbled data that is essentially "locked," but they should not be able to unlock that data without the proper information. In classic cryptography, the encryption(scrambling) algorithm was the secret. In modern cryptography, that isn't feasible. The algorithms are public, and cryptographic keys are used in the encryption and decryption processes. The only thing that needs to be secret is the key. In addition, as we will demonstrate a bit later, there are common cases in which not all keys need to be kept secret.

1.5 Organization of This Report

The rest of the book is organized in the following way. In Chapter 1, we will show the background and related research studies. After that,

- **In Chapter 2,** describes objectives, the Technology we used.
- **In Chapter 3,** consists of Literature review.It describes the Existing System, the Technology we are used and a review of the existing system

- **In Chapter 4,** consists of our Methodology, system architecture working methodology
- **In Chapter 5,** consist of Socio-economic Impact.
- **In Chapter 6,** concludes the Report of Our Project. In this chapter, we will discuss about limitations and future works. In the limitation part, we will discuss about the limitations of our system. In future works, we will discuss about the modules which we will develop in the future.

Chapter 2

Objectives

2.1 Overview

RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total – or factoring – is considered infeasible due to the time it would take using even today's supercomputers.

The public and private key generation algorithm is the most complex part of RSA cryptography. Two large prime numbers, p and q, are generated using the Rabin-Miller primality test algorithm. A modulus, n, is calculated by multiplying p and q. This number is used by both the public and private keys and provides the link between them. Its length, usually expressed in bits, is called the key length. The public key consists of the modulus n and a public exponent, e, which is normally set at 65537, as it's a prime number that is not too large. The e-figure doesn't have to be a secretly selected prime number, as the public key is shared with everyone.

The private key consists of the modulus n and the private exponent d, which is calculated using the Extended Euclidean algorithm to find the multiplicative inverse with respect to the totient of n. The idea of making one of your own encryption algorithms public on the internet seems very strange at first. However, this is actually one of the most important steps in RSA encryption.

If Person C intercepts your message to Person B, they already know the encryption key

(exponent e, modulus n). However, what he/she doesn't have is the decryption exponent d. Since you encrypted your message with Person B's encryption key, only Person B has the decryption key (exponent d, modulus n) to decrypt it. Person C is only missing one piece of information, exponent d, which turns out to be the hardest piece of information to find.

Person C also knows that de 1 (mod (n)), or de 1 (mod (p - 1)(q - 1)). Since he/she knows that n = pq, the simplest way to find n would be to somehow factor n into the exact primes used by Person B in the algorithm. From there, he/she could simply calculate the congruence to find d.

With larger (which are more secure) primes, this turns out to be nearly impossible to do.

If p = 7717 and q = 7919, n would be 61110923.

If we let e = 5, then all Person C knows is e = 5, n = 61110923.

Clearly, it would take very long to factor n, but imagine what would happen if

p = 982451653, q = 961748941. Then n would be 944871836856449473.

Now factoring n is basically impossible to do by hand. However, even this value of n is smaller than most values of n used in RSA Encryption. It took 290 computers over the internet and a supercomputer 4 months to find that

n=109417386415705274218097073220403576120037329454492059909138421314763499842889347847179972

Had prime factors

p=10263959282974110577205419657399167590071656780803806680334193352179071130777 9,

q=10660348838016845482092722036001287867920795857598929152227060823719306280864 3.

In this case, n had only 155 digits. Many values of n have over 200 digits, making the RSA algorithm nearly unbreakable.

2.2 How is RSA secure?

RSA security relies on the computational difficulty of factoring large integers. As computing power increases and more efficient factoring algorithms are discovered, the ability to factor larger and larger numbers also increases.

Encryption strength is directly tied to key size. Doubling key length can deliver an exponential increase in strength, although it does impair performance. RSA keys are typically 1024- or 2048-bits long, but experts believe that 1024-bit keys are no longer fully secure against all

attacks. This is why the government and some industries are moving to a minimum key length of 2048-bits.

Barring an unforeseen breakthrough in quantum computing, it will be many years before longer keys are required, but elliptic curve cryptography (ECC) is gaining favor with many security experts as an alternative to RSA to implement public key cryptography. It can create faster, smaller and more efficient cryptographic keys.

Modern hardware and software are ECC-ready, and its popularity is likely to grow. It can deliver equivalent security with lower computing power and battery resource usage, making it more suitable for mobile apps than RSA.

A team of researchers, which included Adi Shamir, a co-inventor of RSA, successfully created a 4096-bit RSA key using acoustic cryptanalysis. However, note that any encryption algorithm is vulnerable to attack.

Chapter 3

Literature Review

3.1 Cryptographic Algorithms

The SSL protocol covers many cryptographic needs. Sometimes, though, it isn't good enough. For example, you may wish to encrypt HTTP cookies that will be placed on an end user's browser. SSL won't help protect the cookies while they're being stored on that disk. For situations like this, OpenSSL exports the underlying cryptographic algorithms used in its implementation of the SSL protocol.

Generally, you should avoid using cryptographic algorithms directly if possible. You're not likely to get a totally secure system simply by picking an algorithm and applying it. Usually, cryptographic algorithms are incorporated into cryptographic protocols. Plenty of nonobvious things can be wrong with a protocol based on cryptographic algorithms. That is why it's better to try to find a well-known cryptographic protocol to do what you want to do, instead of inventing something yourself. In fact, even the protocols invented by cryptographers often have subtle holes.

If not for public review, most protocols in use would be insecure. Consider the original WEP protocol for IEEE 802.11 wireless networking. WEP (Wired Equivalent Privacy) is the protocol that is supposed to provide the same level of security for data that physical lines provide. It is a challenge, because data is transmitted through the air, instead of across a wire. WEP was designed by veteran programmers, yet without soliciting the opinions of any professional cryptographers or security protocol developers. Although to a seasoned developer with mod-

erate security knowledge the protocol looked fine, in reality, it was totally lacking in security. Nonetheless, sometimes you might find a protocol that does what you need, but can't find an implementation that suits your needs. Alternatively, you might find that you do need to come up with your own protocol. For those cases, we do document the SSL cryptographic API. Five types of cryptographic algorithms are discussed in this book: symmetric key encryption, public key encryption, cryptographic hash functions, message authentication codes, and digital signatures.

3.2 Overview of RSA algorithm

The Rivest-Shamir-Adleman (RSA) encryption algorithm is an asymmetric encryption algorithm that is widely used in many products and services. Asymmetric encryption uses a key pair that is mathematically linked to encrypt and decrypt data. A private and public key are created, with the public key being accessible to anyone and the private key being a secret known only by the key pair creator. With RSA, either the private or public key can encrypt the data, while the other key decrypts it. This is one of the reasons RSA is the most used asymmetric encryption algorithm.

The option to encrypt with either the private or public key provides a multitude of services to RSA users. If the public key is used for encryption, the private key must be used to decrypt the data. This is perfect for sending sensitive information across a network or Internet connection, where the recipient of the data sends the data sender their public key. The sender of the data then encrypts the sensitive information with the public key and sends it to the recipient. Since the public key encrypted the data, only the owner of the private key can decrypt the sensitive data. Thus, only the intended recipient of the data can decrypt it, even if the data were taken in transit.

The other method of asymmetric encryption with RSA is encrypting a message with a private key. In this example, the sender of the data encrypts the data with their private key and sends encrypted data and their public key along to the recipient of the data. The recipient of the data can then decrypt the data with the sender's public key, thus verifying the sender is who they say they are. With this method, the data could be stolen and read in transit, but the true purpose of this type of encryption is to prove the identity of the sender. If the data were

stolen and modified in transit, the public key would not be able to decrypt the new message, and so the recipient would know the data had been modified in transit.
The technical details of RSA work on the idea that it is easy to generate a number by multiplying two sufficiently large numbers together, but factorizing that number back into the original prime numbers is extremely difficult. The public and private key are created with two numbers, one of which is a product of two large prime numbers. Both use the same two prime numbers to compute their value. RSA keys tend to be 1024 or 2048 bits in length, making them extremely difficult to factorize, though 1024 bit keys are believed to breakable soon.

3.3 Overview of Openssl

SSL is currently the most widely deployed security protocol. It is the security protocol behind secure HTTP (HTTPS), and thus is responsible for the little lock in the corner of your web browser. SSL is capable of securing any protocol that works over TCP.
An SSL transaction starts with the client sending a handshake to the server. In the server's response, it sends its certificate. As previously mentioned, a certificate is a piece of data that includes a public key associated with the server and other interesting information, such as the owner of the certificate, its expiration date, and the fully qualified domain name associated with the server.

During the connection process, the server will prove its identity by using its private key to successfully decrypt a challenge that the client encrypts with the server's public key. The client needs to receive the correct unencrypted data to proceed. Therefore, the server's certificate can remain public—an attacker would need a copy of the certificate as well as the associated private key in order to masquerade as a known server.
However, an attacker could always intercept server messages and present the attacker's certificate. The data fields of the forged certificate can look legitimate (such as the domain name associated with the server and the name of the entity associated with the certificate). In such a case, the attacker might establish a proxy connection to the intended server, and then just eavesdrop on all data. Such an attack is called a "man-in-the-middle" attack and is shown

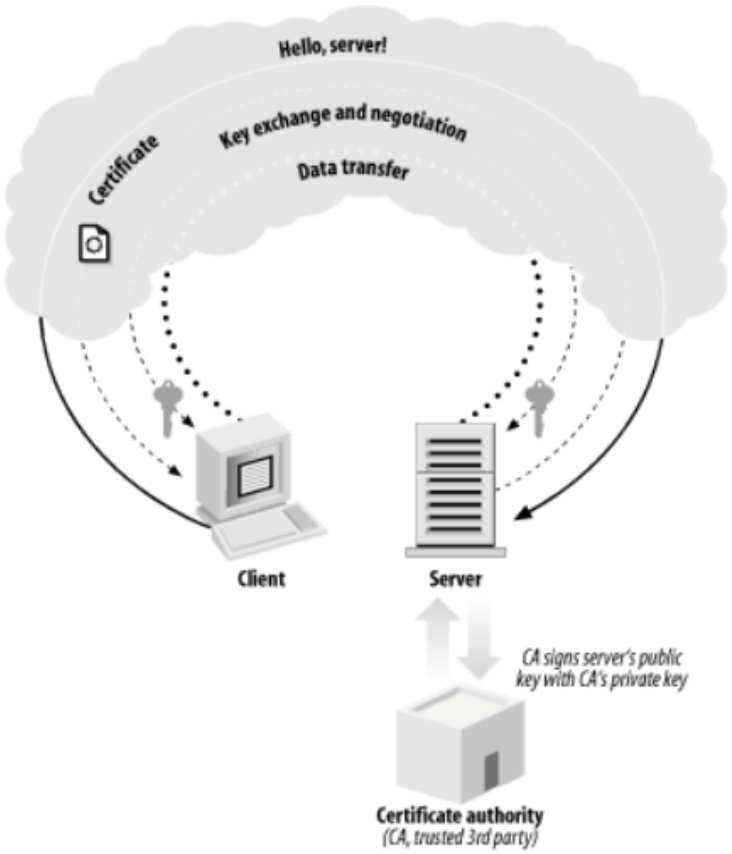

Figure 3.1: Direct communication in openssl

in Figure 1-4. To thwart a man-in-the-middle attack completely, the client must not only perform thorough validation of the server certificate, but also have some way of determining whether the certificate itself is trustworthy. One way to determine trustworthiness is to hardcode a list of valid certificates into the client. The problem with this solution is that it is not scalable. Imagine needing the certificate for every secure HTTP server you might wish to use on the net stored in your web browser before you even begin surfing.

The practical solution to this problem is to involve a trusted third party that is responsible for keeping a database of valid certificates. A trusted third party, called a Certification Authority , signs valid server certificates using its private key. The signature indicates that the Certification Authority has done a background check on the entity that owns the certificate being presented, thus ensuring to some degree that the data presented in the certificate is

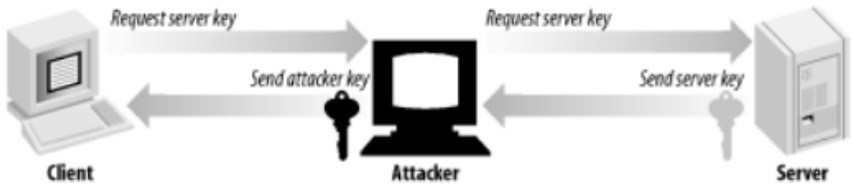

Figure 3.2: Man-In-The-Middle attack

accurate. That signature is included in the certificate, and is presented at connection time. The client can validate the authority's signature, assuming that it has the public key of the Certification Authority locally. If that check succeeds, the client can be reasonably confident the certificate is owned by an entity known to the trusted third party, and can then check the validity of other information stored in the certificate, such as whether the certificate has expired.

Although rare, the server can also request a certificate from the client. Before certificate validation is done, client and server agree on which cryptographic algorithms to use. After the certificate validation, client and server agree upon a symmetric key using a secure key agreement protocol (data is transferred using a symmetric key encryption algorithm). Once all of the negotiations are complete, the client and server can exchange data at will.

The details of the SSL protocol get slightly more complex. Message Authentication Codes are used extensively to ensure data integrity. Additionally, during certificate validation, a party can go to the Certification Authority for Certificate Revocation Lists (CRLs) to ensure that certificates that appear valid haven't actually been stolen. We won't get into the details of the SSL protocol (or its successor, TLS). For our purposes, we can treat everything else as a black box. Again, if you are interested in the details, we recommend Eric Rescorla's book SSL and TLS.

3.4 Symmetric key encryption

Symmetric key algorithms encrypt and decrypt data using a single key. As shown in Figure 1-1, the key and the plaintext message are passed to the encryption algorithm, producing

ciphertext. The result can be sent across an insecure medium, allowing only a recipient who has the original key to decrypt the message, which is done by passing the ciphertext and the key to a decryption algorithm. Obviously, the key must remain secret for this scheme to be effective.

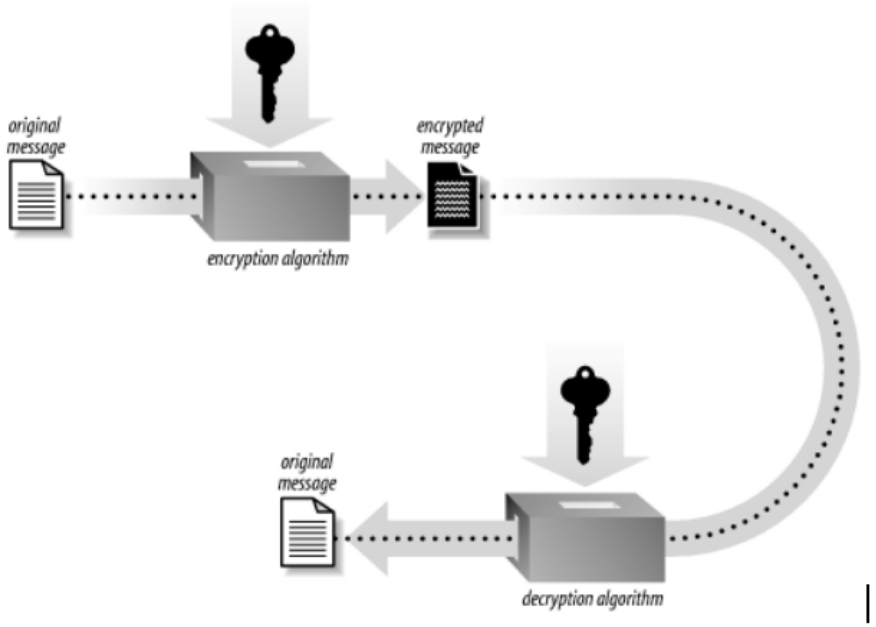

Figure 3.3: Symmetric key cryptography

The primary disadvantage of symmetric key algorithms is that the key must remain secret at all times. In particular, exchanging secret keys can be difficult, since you'll usually want to exchange keys on the same medium that you're trying to use encryption to protect. Sending the key in the clear before you use it leaves open the possibility of an attacker recording the key before you even begin to send data.

One solution to the key distribution problem is to use a cryptographic key exchange protocol. OpenSSL provides the Diffie-Hellman protocol for this purpose, which allows for key agreement without actually divulging the key on the network. However, Diffie-Hellman does not guarantee the identity of the party with whom you are exchanging keys. Some sort of authentication mechanism is necessary to ensure that you don't accidentally exchange keys with an attacker.

Right now, Triple DES (usually written 3DES, or sometimes DES3) is the most conservative symmetric cipher available. It is in wide use, but AES, the new Advanced Encryption Standard, will eventually replace it as the most widely used cipher. AES is certainly faster than 3DES, but 3DES has been around a lot longer, and thus is a more conservative choice for the ultra-paranoid. It is worth mentioning that RC4 is widely supported by existing clients and servers. It is faster than 3DES, but is difficult to set up properly (don't worry, SSL uses RC4 properly). For purposes of compatibility with existing software in which neither AES nor 3DES are supported, RC4 is of particular interest.
Security is related to the length of the key. Longer key lengths are, of course, better. To ensure security, you should only use key lengths of 80 bits or higher. While 64-bit keys may be secure, they likely will not be for long, whereas 80-bit keys should be secure for at least a few years to come. AES supports only 128-bit keys and higher, while 3DES has a fixed 112 bits of effective security.[1] Both of these should be secure for all cryptographic needs for the foreseeable future. Larger keys are probably unnecessary. Key lengths of 56 bits (regular DES) or less (40-bit keys are common) are too weak; they have proven to be breakable with a modest amount of time and effort.

3.5 Public key encryption

Public key cryptography suggests a solution to the key distribution problem that plagues symmetric cryptography. In the most popular form of public key cryptography, each party has two keys, one that must remain secret (the private key) and one that can be freely distributed (the public key). The two keys have a special mathematical relationship. For Alice to send a message to Bob using public key encryption (see Figure 1-2), Alice must first have Bob's public key. She then encrypts her message using Bob's public key, and delivers it. Once encrypted, only someone who has Bob's private key can successfully decrypt the message (hopefully, that's only Bob).

Public key encryption solves the problem of key distribution, assuming there is some way to find Bob's public key and ensure that the key really does belong to Bob. In practice,

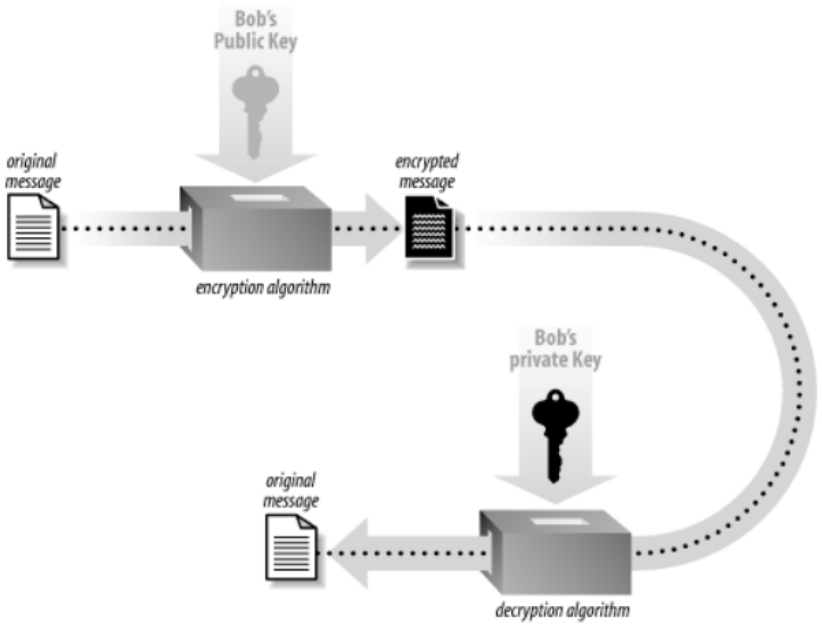

Figure 3.4: Public key cryptography

public keys are passed around with a bunch of supporting information called a certificate , and those certificates are validated by trusted third parties. Often, a trusted third party is an organization that does research (such as credit checks) on people who wish to have their certificates validated. SSL uses trusted third parties to help address the key distribution problem.

Public key cryptography has a significant drawback, though: it is intolerably slow for large messages. Symmetric key cryptography can usually be done quickly enough to encrypt and decrypt all the network traffic a machine can manage. Public key cryptography is generally limited by the speed of the cryptography, not the bandwidth going into the computer, particularly on server machines that need to handle multiple connections simultaneously.

As a result, most systems that use public key cryptography, SSL included, use it as little as possible. Generally, public key encryption is used to agree on an encryption key for a symmetric algorithm, and then all further encryption is done using the symmetric algorithm. Therefore, public key encryption algorithms are primarily used in key exchange protocols and when non-repudiation is required.

RSA is the most popular public key encryption algorithm. The Diffie-Hellman key exchange

protocol is based on public key technology and can be used to achieve the same ends by exchanging a symmetric key, which is used to perform actual data encryption and decryption. For public key schemes to be effective, there usually needs to be an authentication mechanism involving a trusted third party that is separate from the encryption itself. Most often, digital signature schemes, which we discuss below, provide the necessary authentication.

Keys in public key algorithms are essentially large numbers with particular properties. Therefore, bit length of keys in public key ciphers aren't directly comparable to symmetric algorithms. With public key encryption algorithms, you should use keys of 1,024 bits or more to ensure reasonable security. 512-bit keys are probably too weak. Anything larger than 2,048 bits may be too slow, and chances are it will not buy security that is much more practical. Recently, there's been some concern that 1,024-bit keys are too weak, but as of this writing, there hasn't been conclusive proof. Certainly, 1,024 bits is a bare minimum for practical security from short-term attacks. If your keys potentially need to stay protected for years, then you might want to go ahead and use 2,048-bit keys.

When selecting key lengths for public key algorithms, you'll usually need to select symmetric key lengths as well. Recommendations vary, but we recommend using 1,024-bit keys when you are willing to work with symmetric keys that are less than 100 bits in length. If you're using 3DES or 128-bit keys, we recommend 2,048-bit public keys. If you are paranoid enough to be using 192-bit keys or higher, we recommend using 4,096-bit public keys.

Requirements for key lengths change if you're using elliptic curve cryptography (ECC), which is a modification of public key cryptography that can provide the same amount of security using faster operations and smaller keys. OpenSSL currently doesn't support ECC, and there may be some lingering patent issues for those who wish to use it. For developers interested in this topic, we recommend the book Implementing Elliptic Curve Cryptography, by Michael Rosing (Manning).

3.6 anti-tampering

The basic idea behind data integrity is that there should be a way for the recipient of a piece of data to determine whether any modifications are made over a period of time. For

example, integrity checks can be used to make sure that data sent over a wire isn't modified in transit. Plenty of well-known checksums exist that can detect and even correct simple errors. However, such checksums are poor at detecting skilled intentional modifications of the data. Several cryptographic checksums do not have these drawbacks if used properly. Note that encryption does not ensure data integrity. Entire classes of encryption algorithms are subject to "bit-flipping" attacks. That is, an attacker can change the actual value of a bit of data by changing the corresponding encrypted bit of data.

3.7 Non-repudiation

Cryptography can enable Bob to prove that a message he received from Alice actually came from Alice. Alice can essentially be held accountable when she sends Bob such a message, as she cannot deny (repudiate) that she sent it. In the real world, you have to assume that an attacker does not compromise particular cryptographic keys. The SSL protocol does not support non-repudiation, but it is easily added by using digital signatures.
These simple services can be used to stop a wide variety of network attacks, including:

3.8 Related Works

In 2005, C. Dods, N. P. Savvy and M. Stam [1] talked about different issues related with mark plans dependent on upon hash capacities. Such plans are at present alluring in some constrained applications, however their significance may increment if at any time a down to earth quantum PC was constructed. They additionally examined issues identified with both their execution and their security and give the primary complete treatment of commonsense usage of hash based mark conspires in the writing .

In 2006, D. R. Stinson [2] contemplated issues identified with the thought of a "secure" hash capacities. A few vital conditions are considered, just as a famous adequate condition (the purported irregular prophet demonstrates). Specifically, he considered the essential inquiry "does impact obstruction suggest preimage opposition?" and gave incomplete responses

to this inquiry – both positive and negative! – in view of consistency properties of the hash work under thought .

Bunch RSA [3] in 1989; the work was done to achieve numerous unscrambling forms at the expense of roughly one. More than one occupation are consolidated to make a group and unscrambling of the total cluster is performed in a solitary procedure, along these lines lessening the expense of numerous decoding forms .

In 2006, Carlos Cid [4] underscored on cryptographic hash capacities. His paper gives an outline of cryptographic hash capacities and a portion of the ongoing advancements influencing their security, specifically the disclosure of proficient techniques for developing impacts for calculations, for example, MD5 and SHA-1. We additionally talk about the numerous ramifications of these ongoing assaults, and the conceivable bearings for the improvement of the hypothesis of hash capacities .

In 2007, Zanin, Di Pietro, and Mancini [5] in their examination introduced another appropriated mark convention dependent on the RSA cryptographic calculation, which is reasonable for expansive scale impromptu systems. This mark convention is appeared to be disseminated, versatile, and powerful while staying subject to tight security and engineering requirements. The investigation uncovers that the power of this convention plan can be upgraded by including just a small amount of the hubs on the system .

Zanin et al. shown that their convention conspire is right, since it permits a picked number of hubs to create a substantial cryptographic mark; it is secure, in light of the fact that an assailant who bargains less than the given number of hubs can't disturb the administration or produce a sham mark; and it is effective, in view of the low overhead in contrast with the quantity of highlights gave .

The creators in [6] proposed another calculation dependent on RSA. The proposed calculation was having new parameters to build the unpredictability of encryption procedure and

decoding process. The proposed technique is secure in contrast with past strategies. Be that as it may, it is computationally over the top expensive. Utilization of numerous parameters in encryption and unscrambling process, makes it very time wasteful .

Work done in [7] introduced another modulus rather than modulus n. in past techniques, n was result of 2 prime numbers. Rather than n , another variable in transmitted to beneficiary. It is increasingly secure yet estimation of new factor is taking a ton of time relatively.

Another refreshed rendition of RSA was proposed by creators in [**?**], it utilizes the idea of four prime numbers rather than two. Four prime numbers were duplicated to discover augmentation modulus. They additionally proposed a period effective key age process. Age of open key and private key are reliant on new factor. They were not reliant on augmentation modulus n .

Bunch RSA [5] in 1989; the work was done to achieve numerous unscrambling forms at the expense of roughly one. More than one occupations are consolidated to make a group and unscrambling of the total cluster is performed in a solitary procedure, along these lines lessening the expense of numerous decoding forms .

This variation works for little and distinctive open examples for a similar modulus N. Decoding of the two figure messages in Batch RSA should be possible at the expense of roughly one RSA unscrambling. Pertinence of this variation is confined to figure writings with truth be told, exceptionally little open examples and where decodings must be taken care of in mass, for example in banks .

As this variation does not contribute a lot to the present work just the essential thought is given here. Idea of the calculation can be comprehended by a model.
Key age and Encryption techniques are same as in standard RSA. Two mes-sages (M1 and M2) are encoded with little open examples bringing about two figure writings C1 and C2.

Open keys for C1 and C2 are thought to be e1 = 3 and e2 = 5 separately .

MultiPrime RSA [8] was intended to upgrade the unscrambling velocity of RSA cryptosystem by taking multiple primes for the modulus. It comprises of k primes p1, p2 ...pk as opposed to utilizing just two as in standard RSA. This variation is increasingly reasonable for use in asset obliged gadgets as it is progressively proficient regarding computational speed when contrasted with RSA CRT .

Elisa Bertino explained the challenges, concepts and approaches of database security. Many concepts regarding database security were provided and most significant techniques were discussed which were based on accessing control systems. He defines the key access control models which were mandatory access control models and the role-based access control (RBAC) model. He also described security for advanced data management systems. The major drawback was that a new device was to be issued, when an individual user required to change the subscription .

Elisa Bertino explained the challenges,[9] concepts and approaches of database security. Many concepts regarding database security were provided and most significant techniques were discussed which were based on accessing control systems. He defines the key access control models which were mandatory access control models and the role-based access control (RBAC) model. He also described security for advanced data management systems. The major drawback was that a new device was to be issued, when an individual user required to change the subscription .

Hua Li et al. explained new compact dual-core architecture [10] used in AES. The practice of using a new compact architecture started that consisted of two independent cores that practice encryption and decryption simultaneously. In order to provide round keys for encryption and decryption, a proposed key generation unit with a 32-bit data path was explored. The concept used to implement shift rows was the important design which helps to increase the encryption time. The major .

Limitation was that in comparison to the other designs, this design also requires fewer more hardware resources. H. C. Williams modified the RSA public-key encryption algorithm. He suggested that if the encryption procedure was broken into a certain number of operations then the remainder used as modulus could be factored after a few more operations. This technique was in similar appearance to RSA so as to produce digital signatures. The main limitation of this scheme was that very large prime numbers were used and generated mathematical errors were observed. Adam .

J. Elbirt et al. explained the AES block cipher algorithm [11] using an FPGA based kit. They proposed that for hardware implementations of encryption algorithms, reprogrammable devices were the best choice. The disadvantage was that when the implementation size was increased then the number of rounds unrolled was also enhanced and this increase was partially offset by the packing of the round keys within the round structure. Hung-Yu Chien highlighted an efficient time bound hierarchical key assignment scheme. They proposed a tamper resistant device that has a new time bound key assignment scheme. It significantly improves the computational performance and reduces the implementation cost as well. Taher Elgamal proposed a signature scheme based on discrete logarithms and implemented the Diffie-Hellman key distribution scheme that achieves a public key cryptosystem. The security of both systems depends on the difficulty of computing discrete logarithms over finite fields. Martin E. Hellman extended the Shannon theory approach to cryptography. He discussed Shannon's random cipher model which was conservative than in such a case when a randomly chosen cipher was considered, the security falls significantly. The concept of matching a cipher to a language and the trade-off between local and global uncertainty were also developed. The limitation of this approach is that it is not directly applicable to designing practical cryptographic systems .

Jason H. Li et al. worked on scalable key management and clustering schemes for secure group communication in Adhoc and WSN [12]. They describe scalable key management and clustering to achieve a more secured system. The scalability problem was solved by

partitioning communicating devices into subgroups with a leader in each subgroup .

Efficient Clustering Approach (DECA) provided robust clustering to form subgroups. Analytical and simulation results pinpoint the fact that DECA was energy efficient and resilient against node mobility. This scheme was not suitable for large cluster sizes .

Hung-Min Sun et al. proposed a dual RSA algorithm and also did the acute analysis of the security of the algorithm [12]. Dual RSA is a variant of RSA which is helpful in some specific situations that require two instances of RSA with the advantage of reducing the storage requirements for the keys. The main drawback of using dual RSA was that the computational complexity of the key generation algorithms was also optimized .

Mao-Yin Wang et al. configured single and multi-core AES architectures [13] for flexible security. According to them, the major building blocks for the architecture of AES was a group of AES processors. Each AES processor provides a block cipher scheme with a novel key expansion design approach for the original AES algorithm. In this multi-core architecture the memory controller of each AES processor was designed for the maximum overlapping between data transfer and encryption and thus reducing interrupt handling load of the host processor.

Tomasz Rams et al. surveyed a group key distribution scheme with self-healing propert[14]te. They analyzed and compare the most significant key distribution schemes by looking at the selective key distribution algorithms, at the redistributed secret data management, and the self-healing mechanisms. Limitation of the self-healing techniques adds some redundant information to the broadcast message so as to allow user nodes to recover previous session keys which were lost due to communication errors .

Zhiguo Wan et al. worked on a hierarchical attribute [15] based solution for flexible and scalable access control in cloud computing. They proposed hierarchical attribute-set-based encryption by extending ciphertext policy attribute-set-based encryption with a hierarchical structure of users. The proposed scheme not only achieves scalability due to its

hierarchical structure but also inherits flexibility. It employed multiple value assignments for access expiration time to deal with user revocation more efficiently than existing schemes .

Yang Li et al. worked on a New Fault-Based Side-Channel Attack [16] called fault sensitivity analysis attack Using Fault Sensitivity. They explained the successful FSA attacks against three Advanced Encryption Standard hardware implementations, where two of them were resistant to the differential fault analysis. They also discussed the countermeasures against the proposed FSA attacks .

Chong Hee Kim et al. improved differential fault analysis on AES key schedule. Proposed advanced encryption standard for which the main target is known as DFA. Implementation of AES is known to be vulnerable to DFA which could be split into two categories depending on the fault location that has the DFA on the state and the DFA on the key schedule. The major limitation is that if the key schedule is not redone for recomputation then it cannot prevent DFA on the AES Key Schedule. The major problem was that if the key schedule was not done again for recomputation then it cannot prevent DFA on the AES Key Schedule .

Shengrong Bu et al. worked on Distributed Combined Authentication and Intrusion Detection [17] with Data Fusion in High-Security Mobile Ad Hoc Networks. Multimodal biometrics was deployed to work with intrusion detection systems to alleviate the shortcomings of unimodal biometric systems. Each device in the network had measurement and estimation limitations, Observations of each device were fused and more than one device could be chosen using Dempster- Shafer theory for data fusion. Combining continuous authentication and intrusion detection could be an effective approach to improve the security performance in high-security MANETs .

L.J. Garcia Villalba et al. securely extended optimized link state routing protocol [9]. Their study presented an extension of OLSR called COD-OLSR that provides security for OLSR in case of incorrect message generation attacks which can occur in two forms. This was one of its main features and was taken into account for the current topology of nodes sending

the message. The behavior of COD-OLSR against different attackers in a variety of situations is evaluated .

Ho Won Kim et al. designed and implemented a private and public key crypto processor and its application to a Security System [21]. A special-purpose microprocessor was optimized for the execution of cryptography algorithms. This crypto processor could be used for various security applications such as storage devices, embedded systems, network routers, security gateways using IPSec and SSL protocol, etc. They presented the design and implementation of a crypto processor composed of a 32-bit RISC processor and coprocessor blocks dedicated to the AES, KASUMI, SEED, triple- DES, ECC and RSA crypto algorithms .

From systematic literature review performed in this section, It is clear that the Common Problems in Existing RSA Variants: The main disadvantage of RSA decryption is its:

- slower speed.
- Large key generation time.
- Not secure against wiener's attack.
- Problem arise to common modulus attack.
- known plaintext attack are possible

This report has elaborated the basic concept of cryptography and the key management schemes. A review of modern methods is also done in brief. The most of the modern data security techniques have been reviewed. Each of the method has been analyzed with the advantages and the disadvantages. Then a list of common problems in the current version has been identified. On basis of the project gap identified, the problem was formulated.

Chapter 4

Methodology

- Create two different virtual machine in a same network.
  Virtual machine name 1: vm20@serverA
  IP : 192.168.122.20
  Virtual machine name 2: vm20@serverB
  IP : 192.168.122.21

- Sharing a common file in both virtual machine using Network File Share (NFS).
  Virtual machine 1: vm20@serverA (nfs server)
  Shared Directory name : commonA
  Shared Directory location: /commonA
  Virtual machine 1: vm20@serverB (nfs clinet)
  Shared Directory name : commonB
  Shared Directory location: /commonB

Our project is a fully automated created using bash script. At first, we created two virtual machines and then we made a sharable directory. Any of these servers can access this directory. This is created by using NFS(Network File System).

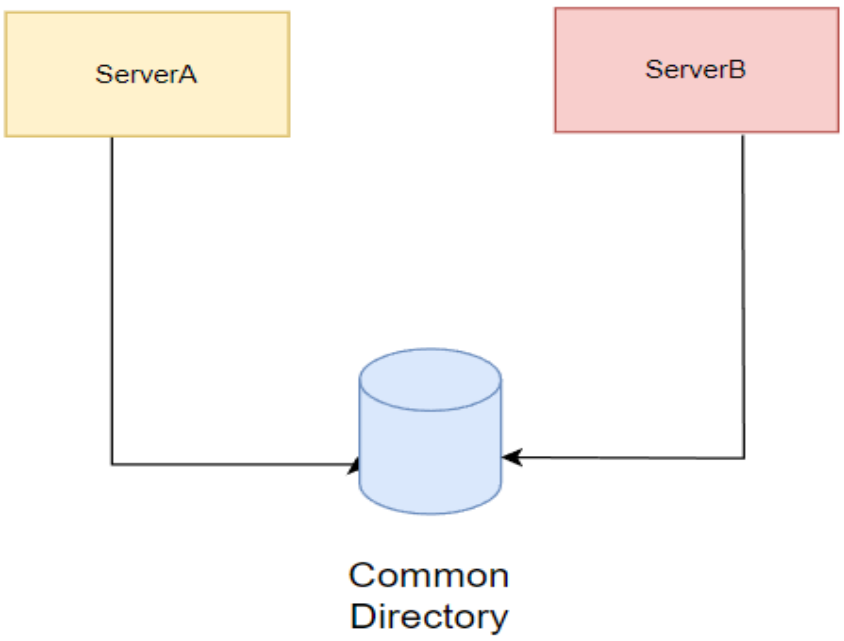

Figure 4.1: Method of file sharing

While sending an encrypted file from ServerA to ServerB at first we create a file in the serverA then encrypt it using publicA.pem key.When the ServerB want to decrypt the file he will need keypairB.pem key to decrypt the file.

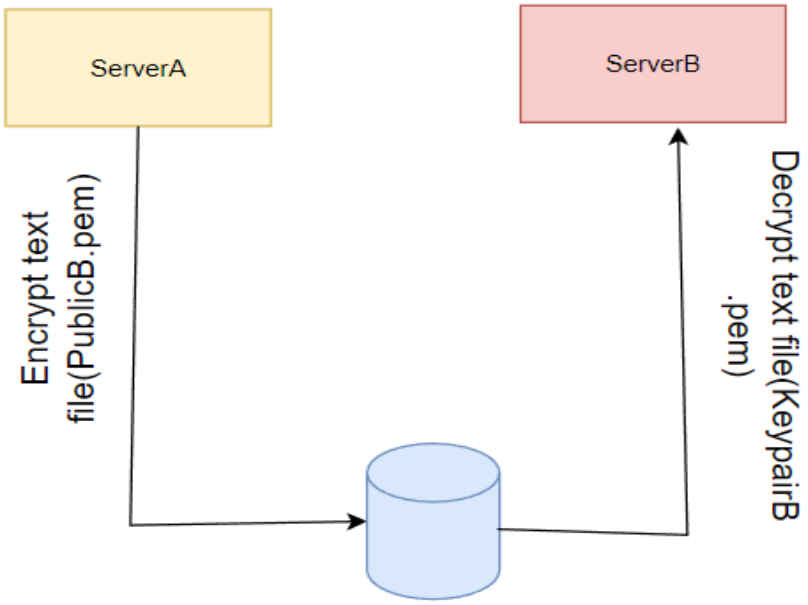

Figure 4.2: Method of encryption and decryption between ServerA to ServerB

While sending an encrypted file from ServerB to ServerA at first we create a file in the serverB then encrypt it using publicB.pem key.When the ServerA want to decrypt the file he will need keypairA.pem key to decrypt the file.

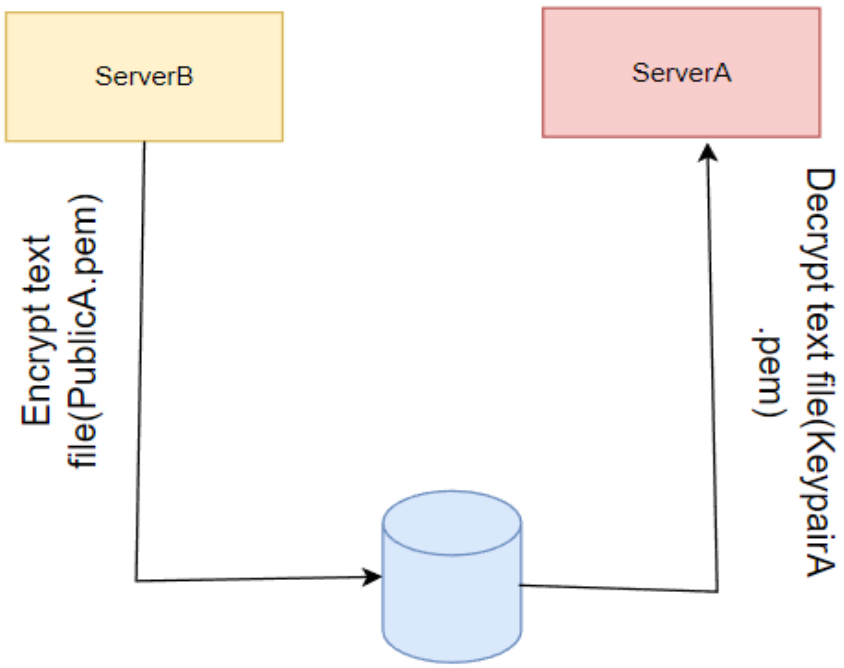

Figure 4.3: Method of encryption and decryption between ServerA to ServerB

I. encrypta.sh

```
#!/bin/bash
echo
echo
echo "Encrypting System is Running (openssl-rsa2048)"
echo
echo "Please enter 2 information for encrypting your realfile"
echo
echo "You are commonA (Server A)"
echo
echo "Enter real text file name:"
read realfile
echo "Enter the converted encrypt file name (what you want):"
read encryptfile
echo
echo "Using publicB.pem file"


openssl rsautl -encrypt -in /commonA/text_file/$realfile -out
/commonA/encrypt_file/A$encryptfile -inkey /commonA/pem_file/publicB.pem -
pubin
chmod 777 /commonA/encrypt_file/A$encryptfile

echo
echo "Your encrypted file is save in: "/commonA/encrypt_file/A$encryptfile
echo
```

Figure 4.4: Encrypt file in serverA

II. decrypta.sh

```
#!/bin/bash
echo
echo "Decrypting System Running (openssl-rsa2048)"
echo
echo "Please enter 4 information for decrypt you encrypted file"
echo "You are commonA (server A):"
echo "Enter encrypt file name:"
read encrypt
echo "Enter the converted decrypt file name (what you want):"
read decryptfile
echo "Entering keypairA.pem:"



openssl rsautl -decrypt -in /commonA/encrypt_file/$encrypt -out
/commonA/decrypt_file/A$decryptfile -inkey /commonA/key_file/keypairA.pem

echo
echo
echo "This is output and this file is also save in "
/commonA/decrypt_file/A$decryptfile
echo
cat /commonA/decrypt_file/A$decryptfile
echo
echo
```

Figure 4.5: Decrypt file in serverA

I. encryptb.sh

```
#!/bin/bash
echo
echo
echo "Encrypting System is Running (openssl-rsa2048)"
echo
echo "Please enter 2 information for encrypting your realfile"
echo
echo "You are commonB (Server B)"
echo
echo "Enter real text file name:"
read realfile
echo "Enter the converted encrypt file name (what you want):"
read encryptfile
echo
echo "Using publicA.pem file"


openssl rsautl -encrypt -in /commonB/text_file/$realfile -out
/commonB/encrypt_file/B$encryptfile -inkey /commonB/pem_file/publicA.pem -
pubin
chmod 777 /commonB/encrypt_file/B$encryptfile

echo
echo "Your encrypted file is save in: "/commonB/encrypt_file/B$encryptfile
echo
echo
```

Figure 4.6: Encrypt file in serverB

II. decryptb.sh

```
#!/bin/bash
echo
echo "Decrypting System Running (openssl-rsa2048)"
echo
echo "Please enter 2 information for decrypt you encrypted file"
echo "You are commonB (server B):"
echo "Enter encrypt file name:"
read encrypt
echo "Enter the converted decrypt file name (what you want):"
read decryptfile
echo "Entering keypairB.pem:"



openssl rsautl -decrypt -in /commonB/encrypt_file/$encrypt -out
/commonB/decrypt_file/B$decryptfile -inkey /commonB/key_file/keypairB.pem

echo
echo
echo "This is output and this file is also save in "
/commonB/decrypt_file/B$decryptfile
echo
cat /commonB/decrypt_file/B$decryptfile
echo
echo
```

Figure 4.7: Decrypt file in serverB

4. Generate private key (keypair) for serverA and serverB

For serverA:
at first we have to enter into key_file directory, by using this command

```
" cd /commonA/key_file"
```

then we will enter this command in our terminal to generate private key

```
" openssl genrsa -out keypairA.pem 2048"
```

For serverB:
at first we have to enter into key_file directory, by using this command

```
" cd /commonB/key_file"
```

then we will enter this command in our terminal to generate private key

```
" openssl genrsa -out keypairB.pem 2048"
```

Figure 4.8: Generate private key for ServerA and ServerB

5. Generate public key in pem_file for public key.

For serverA:
at first we have to enter into pem_file directory, by using this command

“ cd /commonA/pem_file”

then we will enter this command in our terminal to generate public key

“ openssl rsa -in /commonA/keypairA.pem -pubout -out publicA.pem”

For serverB:
at first we have to enter into pem_file directory, by using this command

“ cd /commonB/pem_file”

then we will enter this command in our terminal to generate public key

“ openssl rsa -in /commonB/keypairB.pem -pubout -out publicB.pem”

Figure 4.9: Generate public key for serverA and ServerB

Chapter 5

Socio-economic Impact

5.1 Overview

Data is becoming largely existent in today's world than they were anticipated some three decades ago. Individuals are keeping a lot more information than organizations kept in the past. Significant amounts of such information are valued and consequently preferred to be known to them alone. Such valued information includes their financial details, medical records, locations, as well as professional and network information. Businesses and organizations possess larger amounts of information than individuals.

A good amount of such information is critical to their sustained existence and growth. Their intellectual properties and trade secrets are kept away from potential exploits, thus, considered very private. Governments and agencies keep sensitive information that may affect the stability of their jurisdictions, politically or economically, if divulged.The necessity to keep such information within the required confines describes a component purpose of Information Security, which involves the totality of activities to ensure the protection of information assets that use, store, or transmit information from risk through the application of policies, education, training, awareness, and technology.

Data security involves the consideration of potential confidentiality, integrity, and availability threats to data services, using functions such as identification, authentication,authorization and audit.An important and popular methodology for enforcing information security is encryption, which is itself an element of cryptography. Cryptography provides a secret com-

munication mechanism between two or more parties.Symmetric and Public Key Cryptography employ various algorithms to ensure the security of data items 'at rest', 'in use',and 'in motion'.

Data encryption may not be an explicit solution to information security problems, as organizations remain increasingly vulnerable to data breach incidents, but it is still the most efficient fix when deployed adequately [4]. This has led to the growing availability of full disk encryption tools.Disk manufacturers are embedding full encryption tools into their products, making encryption more available for use [5].The study conducted by showed the increased usage of full disk, virtual volume, native disk, and flash drive encryption over two years. However, for reasons other than the cost of deployment and managing an encryption solution,some organizations have shunned or are still undecided about adopting encryption solutions. They insisted that "availability is more important than confidentiality" .

The time the encryption and decryption processes take before data is made accessible to potential users may cause delay in organizations operations, depending on how complex the base algorithm is.Such delays may escalate to a sort of denial-of-service situation, which may be adverse to organizations' businesses.On the electronic discovery front, unavailability problem prevents anticipated investigation of cyber-incidents.

5.2 It's Not New to Use Secure Encryption

First off, there have always been ways for people to communicate that are incomprehensible to the authorities. If the message is not in the words as they appear, you cannot read it, even if you are able to intercept a letter and read the words written on the sheet of paper.

Since at least the Roman Empire, ciphers have been utilized that could not be deciphered by modern eavesdroppers. A fresh set of these ciphers is merely made available by new technology.

For instance, using a one-time pad correctly can offer entirely safe encryption. If not earlier, this method has been in use since 1882.

Other modern technologies "merely" makes it extremely challenging to decipher, taking hundreds of thousands or even millions of years to do it using a brute-force approach. These

intervals are sufficient to make these ciphers effectively impossible to decipher.
Governments are therefore unable to take any actual action to stop communication that is unreadable by security services. Making it difficult for the average person to use such communication is the only thing that can be done.

5.3 Terrorists are criminals

Terrorists are criminals by definition; not only are the terrorist activities themselves unlawful, but it's frequently also against the law to possess the weapons used in such acts.
Therefore, even though secure communication is outlawed, terrorists won't be deterred from utilizing it.
Criminal organizations in particular will not hesitate to utilize whatever methods at their disposal to protect the privacy of their communications: if something gives them an advantage over the police or anybody else who would want to stop them, they will use it.

5.4 Society relies on secure encryption

Secure encrypted communication makes it impossible for criminals to read the communications of ordinary persons as well as the communications of government agencies.
This website employs HTTPS for all traffic, much like a growing number of other websites. When configured properly, this ensures that anyone eavesdropping on internet traffic cannot determine which pages you visited or extract any of the data delivered to or from the website by you as a visitor.
This is essential for services like online banking because it stops hackers from intercepting your conversations with your bank in order to gain your passwords and account information. Online banking would not be possible if such communications could not be trusted to be secure since the risk of fraud from password theft would be too high.
Similar to safe Virtual Private Networks (VPNs), which rely on secure encryption to send data between machines only connected via the internet, many businesses use VPNs.

As a result, they are able to securely transfer data between locations or between distant workers without having to worry about snoopers intercepting their communications.
Without reliable encryption, many large international corporations would suffer greatly since they couldn't rely on sending data securely over the internet and would instead have to send it physically by courier.

5.5 A "back door" or "government secret key" destroys the security of encryption.

Some of the proposals from politicians have been to require that companies that provide encryption services must also provide a means whereby government security services can also decrypt the communications if required.
This requires that either (a) the company in question keeps a database of all the encryption/decryption keys used for all communications, or (b) the encryption algorithm used allows for decryption via a "back door" or "secret key" in addition to the standard decryption key, so that the government security services can gain access if required, without needing to know the customer's decryption key.
Keeping a database of the decryption keys just provides a direct target for attack by computer criminals. Once such a database is breached, none of the communications provided by that company can be considered secure. This is clearly not a good state of affairs, given the number of times that password databases get compromised.
None of the communications offered by that company can be regarded as secure after such a database has been compromised. Given how often password databases are breached, this situation is obviously undesirable.
That remains choice (b): offering a "back door" or "secret key" or other method by which the security services can decrypt a communication that would otherwise be encrypted.But this seriously undermines such encryption.
Criminal computer crackers will endeavor to ensure that they too may access the communication once they are aware that a back door exists; they won't wait for a warrant from the Home Secretary or another government agency in charge of issuing such warrants! Any such

group that succeeds in gaining access would probably not divulge it to the public; instead, they would merely use it to make sure they could access communications that were important to them, whether that was because they had a direct need for the information or because it could be sold to other criminal organizations.

The computation required to identify the key decreases dramatically if there is a single key that can decrypt all communication using a given system. This is especially true if you have access to the raw, unencrypted message. The more messages that are transmitted with a given key, the easier it is to identify the key. The huge volume of electronic communications in use today would mean that the secret back door key would be much more readily compromised than any individual encryption key.

5.6 Privacy is a human right

Nobody shall be the target of willful intrusion into their personal space, those of their loved ones, their homes, or their correspondence, or of attacks on their character or honor.

Everyone has a right to legal protection from these types of intrusions or assaults.Our correspondence is shielded from interference, even that from the government, thanks to secure encryption.

The right to freedom of expression, which is protected by article 19 of the Universal Declaration of Human Rights, is also violated by restrictions on the use of encryption.

Everyone has the right to freedom of expression, which includes the ability to hold opinions without interference and the freedom to seek out, receive, and share ideas across all boundaries and media.

The limitation on our freedom of expression is clear: if I truly had that freedom, I could convey any string of characters or numbers to anyone without facing any resistance. It makes no difference if the string of characters or numbers is an encrypted message.

Therefore, any attempt to restrict the use of specific encryption techniques restricts my capacity to send any message I choose, as specific combinations of letters and numbers are forbidden solely due to their meaning.

Amnesty International and other human rights groups interact with their employees via safe

encrypted communications. Their capacity to carry out their humanitarian activities would suffer if such communications could not be protected from interference. and could endanger their workers.

5.7 Encryption is mathematics

Computer encryption is just a mathematical algorithm applied to a series of numbers. It is ridiculous to consider that performing mathematical operations on a sequence of numbers could be outlawed merely because that sequence of numbers has meaning to someone.
Note: We strongly oppose any efforts to limit the usage of encryption technology. With little to no upside and significant disadvantages, it is technologically and morally flawed.
Politicians should reject any attempts to limit the use of encryption technology, and those running in the UK elections this week should let their prospective supporters know that they will do so.

5.8 ECONOMIC IMPORTANCE OF ENCRYPTION

According to a 1999 study conducted by the Computer Security Institute, financial losses due to computer security breaches grew to over $1 billion in 1998, and 62 percent of respondents reported computer security breaches within the last twelve months.5 This is probably a conservative estimate. A recent assessment of computer security-related economic losses concluded that accurate estimates of economic losses associated with security breaches are difficult to establish because organizations are reluctant to report losses in fear of dampening customer confidence in the safety of their systems.6 The costs of such losses can be characterized in three areas:

- **Direct cost:** These include expenditures for such products as firewalls or antivirus software, the incremental costs of products offering superior safety features or assurances, and training.

- **Indirect cost:** These include such costs as higher computer system prices as a result of more powerful CPUs needed to implement security algorithms or from deferred sales due to consumer concerns about security trustworthiness.

- **System failure costs:** These include the costs or losses resulting from fraud, sabotage or similar direct attacks on the security of a system. The potential and real losses in these and other areas have driven the market for security products. This is particularly true for encryption once DES was approved in 1977. A recent survey by the International Data Corp. (IDC) of 300 commercial U.S. companies with revenue over $100 million concluded that expenditures for security products have grown with increased use of Internet communications by firms. The company found that the worldwide Internet security software market grew 67 percent, from 1996 revenues of $1.2 billion to 1997 sales of $2.0 billion, and that revenues continued to grow to an estimated $3.1 billion in 1998. Moreover, this market was expected to reach $4.2 billion in 1999 and $7.4 billion by 2002, according to IDC estimates.7 The survey also concluded that encryption was second only to user authentication in potential growth. Finally, it concluded that three industries—financial services, telecommunications and transportation—were expected to exceed a 40 percent adoption rate for user authentication by the year 2000.

Finally, we think we should be encouraging the use of strong encryption rather than discouraging it, to protect us from those who would intercept our digital communication and use that for their gain and to our detriment.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

The effectiveness of data encryption as a mechanism for enforcing information privacy is massive. This is evident by the reported widespread use of various data encryption solutions at the organizational and individual levels. However,its huge success for data access restriction has been a threat for digital forensics processes over the years. Cyber-criminals have been exploiting the information confidentiality ability of data encryption solutions, to restrict digital forensics investigators' access to potential evidence. The ubiquitous availability, inexpensive cost and easy implementation of encryption solutions enhance the threats posed to digital forensics processes.

Investigators sometimes get around the encryption challenge through careful and thoughtful planning of search and seizure, thorough search for exposed encryption keys, and advanced in-memory data retrieval techniques. Yet,a minimum of 60% of computer incidents involving data encryption end up unprosecutable.

The TrueCrypt software went even further by providing users with plausible deniability and non-repudiation abilities. This makes digital forensics investigations of encrypted disk drives harder and less feasible. Consequently, this undesired situation constitutes an indirect reason for the rise in the occurrence of computer incidents. As much as data encryption helps offenders get away from being caught, the necessity for data privacy and security cannot be sacrificed for digital forensics.

Unfortunately, the only digital forensics solution to threatening information security solution will have to be unanimously considered by disk drive manufacturers. There Should be a technology that will provide a backdoor for digital forensics investigators to gain access to the most secure encrypted disk drives. However, there will have to be a restriction to the distribution of such technology when it comes to existence. This is to avoid its abuse by non-law enforcement practitioners (and potential computer criminals)to illegally access target data.

6.2 Future Scope

The strength of the method is discovered to rely on the size of the key after researching several encryption algorithms. The security of the method increases along with the key length, but performance suffers and vice versa. We must optimize the key length to prevent this. A new method has been developed to address the shortcomings in RSA that were discovered after a critical analysis of it. The suggested approach speeds up computing while also enhancing system security. In the future, an effort can be done to make the algorithm less difficult.

Bibliography

[1] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *International conference on the theory and applications of cryptographic techniques*, pages 308–318. Springer, 1998.

[2] Scott A Vanstone and Robert J Zuccherato. Short rsa keys and their generation. *Journal of Cryptology*, 8(2):101–114, 1995.

[3] Gilles Brassard. *Advances in Cryptology-CRYPTO'89: Proceedings*, volume 435. Springer, 1995.

[4] Scott A Vanstone and Robert J Zuccherato. Using four-prime rsa in which some of the bits are specified. *Electronics Letters*, 30(25):2118–2119, 1994.

[5] Hung-Min Sun and Mu-En Wu. Design of rebalanced rsa-crt for fast encryption. In *Proceedings of Information Security Conference*, pages 16–27, 2005.

[6] Ravi Shankar Dhakar, Amit Kumar Gupta, and Prashant Sharma. Modified rsa encryption algorithm (mrea). In *2012 second international conference on advanced computing & communication technologies*, pages 426–429. IEEE, 2012.

[7] Rohit Minni, Kaushal Sultania, Saurabh Mishra, and Durai Raj Vincent. An algorithm to enhance security in rsa. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–4. IEEE, 2013.

[8] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[9] Prerna Mahajan and Abhishek Sachdeva. A study of encryption algorithms aes, des and rsa for security. *Global Journal of Computer Science and Technology*, 2013.

[10] Charles Fisher. Flat file encryption with openssl and gpg. *Linux Journal*, 2016(270):2, 2016.

[11] Kalpana Singh and Shefalika Ghosh Samaddar. Enhancing koyama scheme using selective encryption technique in rsa-based singular cubic curve with avk. *Int. J. Netw. Secur.*, 14(3):164–172, 2012.

[12] Uma Somani, Kanika Lakhani, and Manish Mundra. Implementing digital signature with rsa encryption algorithm to enhance the data security of cloud in cloud computing. In *2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010)*, pages 211–216. IEEE, 2010.

[13] Fausto Meneses, Walter Fuertes, José Sancho, Santiago Salvador, Daniela Flores, Hernán Aules, Fidel Castro, Jenny Torres, Alba Miranda, and Danilo Nuela. Rsa encryption algorithm optimization to improve performance and security level of network messages. *IJCSNS*, 16(8):55, 2016.

[14] Wuling Ren and Zhiqian Miao. A hybrid encryption algorithm based on des and rsa in bluetooth communication. In *2010 Second International Conference on Modeling, Simulation and Visualization Methods*, pages 221–225. IEEE, 2010.

[15] Farah Jihan Aufa, Achmad Affandi, et al. Security system analysis in combination method: Rsa encryption and digital signature algorithm. In *2018 4th International Conference on Science and Technology (ICST)*, pages 1–5. IEEE, 2018.

[16] Hoyoung Yu and Youngmin Kim. New rsa encryption mechanism using one-time encryption keys and unpredictable bio-signal for wireless communication devices. *Electronics*, 9(2):246, 2020.

[17] Aljaafari Hamza and Basant Kumar. A review paper on des, aes, rsa encryption standards. In *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, pages 333–338. IEEE, 2020.