

RESEARCH ON VARIOUS CRYPTOGRAPHY TECHNIQUE

A COURSE PROJECT REPORT

By

**MADUNURI HARSHINI (RA2111028010112)
NEHA BHARDWAJ(RA2111028010104)
YELURI BADRINATH (RA2111028010111)
CHITRAREKHA RAJPUT(RA2111028010126)
MAHANTHI(RA2111028010115)
SIMRAN(RA2111028010190)**

Under the guidance of

Dr. Mary Subaja Christo

In partial fulfilment for the Course

of

18CSE381T - CRYPTOGRAPHY

in Department of Networking and Communications



FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chengalpattu District

NOVEMBER 2023

TABLE OF CONTENTS

Chapters	Contents
1	Abstract
2	Problem statement
3	Introduction
4	Literature survey
5	Architecture
6	Algorithm Presentation
7	Code implementation
8	Justification
9	Advantages of RSA method
10	Conclusion
11	References

ABSTRACT

Security is an important in protecting data against intruders. One of the most important methods for ensuring data secrecy is cryptography. Cryptography is secret writing for data security protection. Well-hidden data cannot easily be read, modified or fabricated [1]. Cryptography protects crucial data via changing it into unclear data that can only be accessed via authorized receivers, who then converts the uncertain data into the original textual content. The process of changing original text into unclear text (ciphertext) with a certain key referred to as encryption, and the opposite of encryption process is referred to as decryption process. Privacy and security management present challenges to exam. An e-exam database requires security and reliability. Thus, an e-exam user's identity must be established. Computerized exams are prone to significant problems such as leaks, attackers and so on. One solution is to encrypt the questions inside the database. Encryption is the conversion of plaintext to text that is not clear. The two fundamental techniques for encrypting data are "symmetric cryptography," which entails the usage of the same key to encrypt/ decode information; and "asymmetric cryptography," which makes use of public and private keys to encrypt/ decode information. Examples of symmetric algorithms are Data Encryption Standard (DES), Triple-DES (3DES), Blowfish, and Advanced Encryption Standard (AES). The most wellknown asymmetric algorithms are RSA and ELGAMAL Schema.

PROBLEM STATEMENT

In today's interconnected and data-driven world, the security and privacy of digital information are of paramount importance. With the increasing sophistication of cyber threats and the need to protect sensitive data, the field of cryptography plays a crucial role in safeguarding information. However, the selection and implementation of the appropriate cryptography technique remain a complex challenge. The problem statement for research on various cryptography techniques can be framed as follows:

Problem Statement:

In the realm of information security, the selection and deployment of cryptography techniques are critical for ensuring the confidentiality, integrity, and authenticity of digital data. The problem lies in the diverse range of cryptography methods available, each with its unique features, strengths, and limitations. As organizations and individuals strive to secure their data and communications, they face the following challenges:

Complexity in Choosing the Right Technique

Performance Trade-offs

Vulnerabilities and Evolving Threats

Interoperability and Standards

Quantum Computing Impact

INTRODUCTION

In our increasingly digital and interconnected world, the secure transmission and protection of sensitive information are paramount. The foundation of this security lies in the field of cryptography, which provides the means to encrypt data and communications, making them unintelligible to unauthorized parties. Cryptography techniques are central to ensuring data confidentiality, integrity, and authenticity, serving as the bedrock of modern information security.

This research endeavor embarks on an exploration of various cryptography techniques, their principles, applications, strengths, and limitations. The goal is to shed light on the diverse array of cryptographic methods available, offering insights into their suitability for specific use cases, performance considerations, and their resilience in the face of evolving cyber threats. The significance of this research can hardly be overstated, as it provides a comprehensive understanding of the tools that underpin the security of digital information. The digital age has brought with it an unprecedented volume of data and an ever-expanding digital landscape. As more aspects of our lives and businesses migrate online, the need for robust security measures becomes increasingly evident. Cryptography, with its myriad techniques and algorithms, plays a pivotal role in safeguarding data, whether it's during transmission over networks, storage in databases, or authentication for online services. Without cryptography, our digital world would be vulnerable to eavesdropping, tampering, and impersonation.

LITERATURE SURVEY

S. N O	TITLE OF PAPER	AUTHORS	METHODOLOGY	JOURNAL NAME	LIMITATIONS OR FURTHER ENHANCEMENT
1	A survey on various most common encryption techniques	E Thambiraja, G Ramesh, Dr R umarani	Cryptography-symmetric and asymmetric	International journal of advance research in computer science and software engineering 2(7),2012	Advanced encryption will be central to any modernized digital infrastructure
2	A Survey paper on cryptography technique	A. Joseph Amalraj1 , Dr. J. John Raybin Jose	International Journal of Computer Science and Mobile Computing, Vol.5 Issue.8, August2016	In the modern era evaluation of networking and wireless networks has come in information and communication technology, there are so many things that gives facility to deal with these technology using internet	large-scale quantum computers capable of breaking current algorithms could be available by the end of the decade

3	Study on digital signature security	Rabeya Sultana, Tashrifa Shahid	RSA digital signature algorithm public key infrastructure	International Journal of Research Publication and Reviews	Not discussed
4	study on transaction using cryptography	Siddhanth karanth,2 Shishir S Hegde	Blockchain Technology, Transaction System, Crypto, MultiChains, Meshnets.	Nitte Meenakshi Institute of Technology, Bangalore, India	The researchers believe that Blockchain has immense potential in both academia and industry. In this section, we have briefly discussed different future scopes for the Blockchain technology
5	triple des	Patil and Biradar 2018	Triple Des (TDES) Due to advances in key searching, the Triple DES (3 DES) algorithm was introduced as a replacement for DES. TDES uses DES encryption in three rounds and has a key length of 168 bits (56*3). The Encrypt-Decrypt-Encrypt (EDE) series uses either two or three 56 bit keys.	International Journal of Scientific Research in Computer Science Engineering and Information Technology	financial transactions

6.	A SECURE DATA COMMUNICATION SYSTEM USING CRYPTOGRAPHY AND STEGANOGRAPHY	Saleh Saraireh Department of Communications and Electronic Engineering, Philadelphia University,	The main objective of this paper is to introduce a secure communication system that employs both cryptography and steganography to encrypt and embed the secret message to be transmitted over a non secure channel	Encryption and Decryption Process and Embedding and Extraction Process	the proposed system provides high level of security
----	---	---	---	--	---

ARCHITECTURE

Symmetric Cryptography: In symmetric key cryptography, a single secret key is used for both encryption and decryption. Examples include:

- **Data Encryption Standard (DES):** A widely used symmetric encryption algorithm.
- **Advanced Encryption Standard (AES):** A more secure and modern replacement for DES.

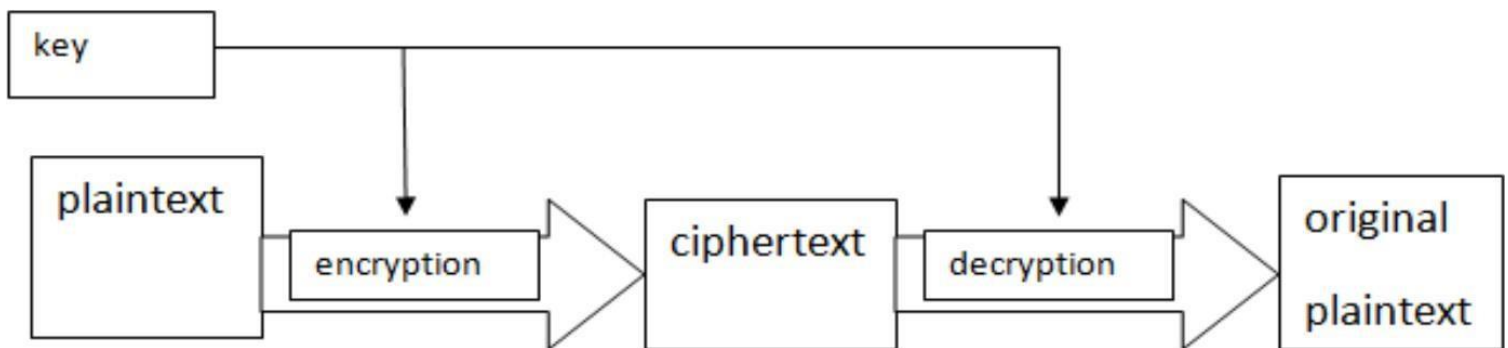


Fig. 1. Symmetric cryptosystem

Asymmetric cryptography: In asymmetric key cryptography, there are two keys, a public key and a private key. Data encrypted with one key can only be decrypted with the other. Examples include:

- **RSA (Rivest-Shamir-Adleman):** A widely used asymmetric encryption algorithm.
- **Elliptic Curve Cryptography (ECC):** Known for its efficiency and security, especially in constrained environments.

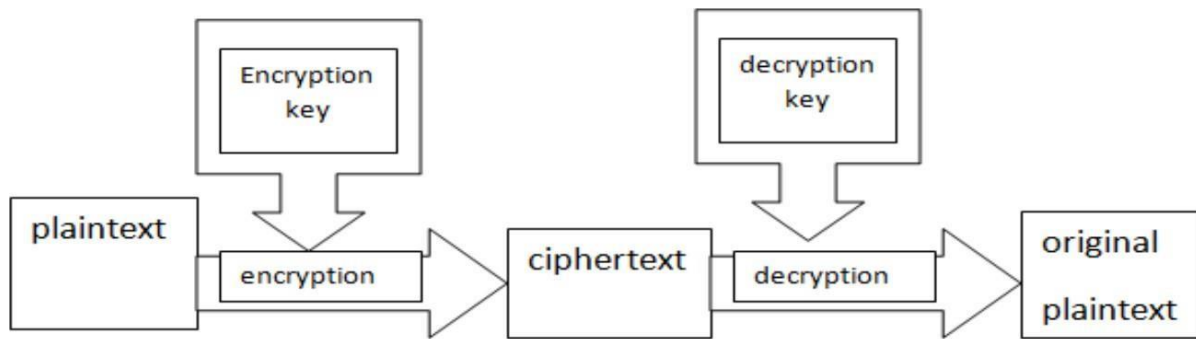


Fig. 2: Asymmetric cryptosystem

S-DES Key Generation: S-DES(Simplified Data Encryption Standard) relies upon on the using of a shared key that consist of 10- bit and share it amongbothsenderand receiver.Pairof8-bitsubkeys.

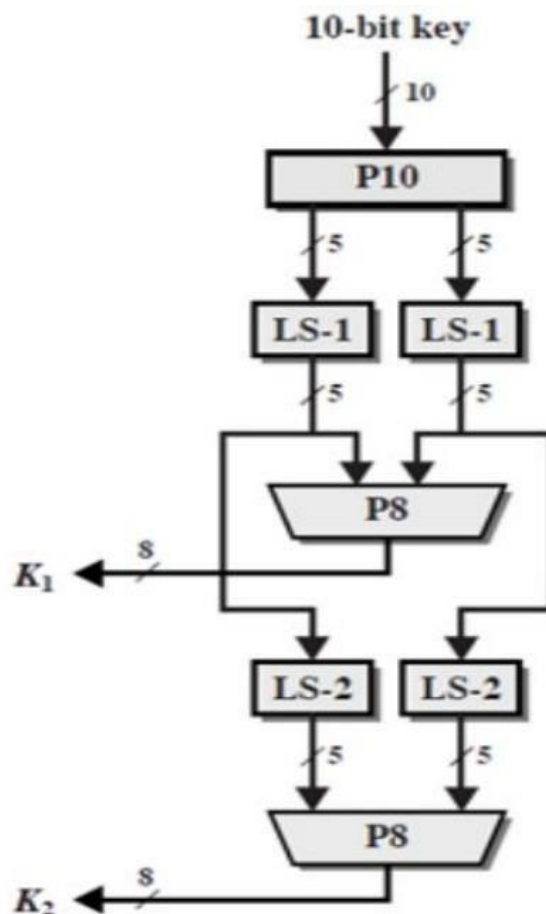


Fig. 3: Key generation for S-DES [3]

Blowfish is fast algorithm, license free, and unpatented. It uses a key length in the range of 32– 448 and a sixty-four-bit block. The Blowfish algorithm makes use 16 roundfortheenciphermentprocedure.

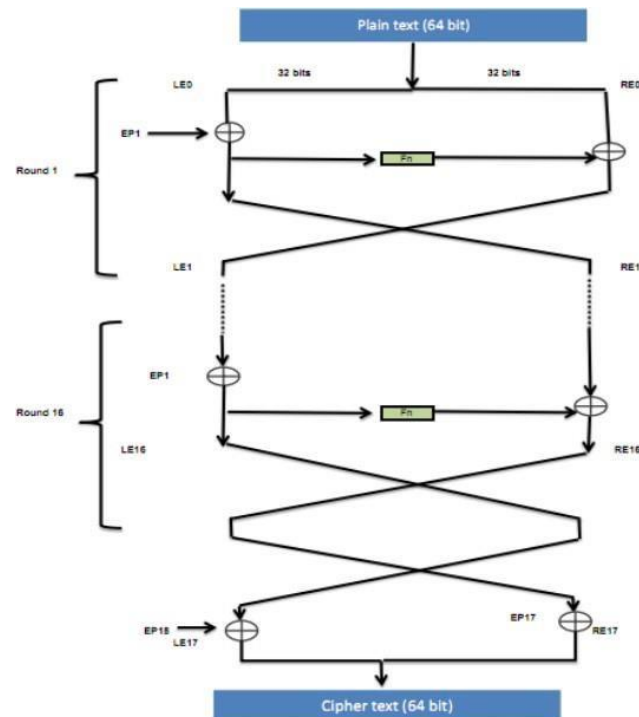


Fig. 6: Blowfish encryption algorithm [2]

AES(Advanced Encryption Standard) is a block cipher with a block size of 128bits. The key length canbe 128, 192 or 256bits.

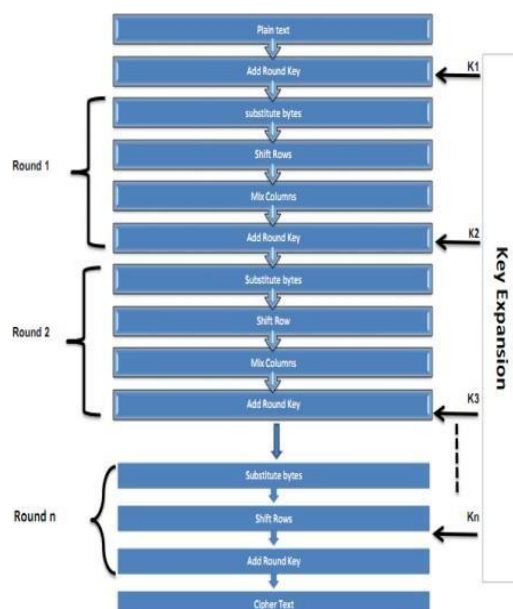


Fig. 7: AES algorithm [2]

ALGORITHM PRESENTATION

The RSA (Rivest-Shamir-Adleman) algorithm is a widely used public key cryptography system for secure data transmission and digital signatures. Here's a simplified representation of the RSA key generation and encryption/decryption process:

Key Generation:

- Choose two distinct prime numbers, p and q .
- Compute the modulus, $n = p * q$.
- Calculate the totient of n , $\phi(n) = (p-1) * (q-1)$.
- Choose an integer e such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$.
- Compute the modular multiplicative inverse of e (d) modulo $\phi(n)$, such that
- $(d * e) \% \phi(n) = 1$
- The public key is (n, e) , and the private key is (n, d)

Encryption:

- Convert the plaintext message (M) into an integer, where $M < n$.
- Compute the ciphertext (C) using the public key (n, e) with the formula: $C = M^e \% n$.

Decryption:

- Compute the plaintext message (M) from the ciphertext (C) using the private key (n, d) with the formula: $M = C^d \% n$.
- It's important to note that in practice, RSA key generation involves additional steps, including random prime number generation, padding schemes, and secure key management. The security of RSA depends on the difficulty of factoring the modulus n , which is why it's important to choose large prime numbers to enhance security.

CODE IMPLEMENTATION

```
// Java Program to Implement the RSA Algorithm
import java.math.*;
import java.util.*;

class RSA {
    public static void main(String args[])
    {
        int p, q, n, z, d = 0, e, i;

        // The number to be encrypted and decrypted
        int msg = 12;
        double c;
        BigInteger msgback;

        // 1st prime number p
        p = 3;

        // 2nd prime number q
        q = 11;
        n = p * q;
        z = (p - 1) * (q - 1);
        System.out.println("the value of z = " + z);

        for (e = 2; e < z; e++) {

            // e is for public key exponent
            if (gcd(e, z) == 1) {
                break;
            }
        }
        System.out.println("the value of e = " + e);
        for (i = 0; i <= 9; i++) {
            int x = 1 + (i * z);

            // d is for private key exponent
            if (x % e == 0) {
                d = x / e;
                break;
            }
        }
        System.out.println("the value of d = " + d);
    }
}
```

```

        c = (Math.pow(msg, e)) % n;
        System.out.println("Encrypted message is : " + c);

        // converting int value of n to BigInteger
        BigInteger N = BigInteger.valueOf(n);

        // converting float value of c to BigInteger
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("Decrypted message is : "
                           + msgback);
    }

    static int gcd(int e, int z)
    {
        if (e == 0)
            return z;
        else
            return gcd(z % e, e);
    }
}

```

Input:

Z=20,e=3,d=7

Output:

```

Output
java -cp /tmp/qyfhKwnG9K RSA
the value of z = 20
the value of e = 3
the value of d = 7
Encrypted message is : 12.0
Decrypted message is : 12

```

CODE FOR HASH TECHNIQUES

```
// Java program to demonstrate
// how to hash a password

package java_cryptography;

import java.util.Scanner;
import org.springframework
    .security
    .crypto
    .bcrypt
    .BCrypt;

public class Hashing {

    // Creating a private instance
    // of Scanner class
    private static Scanner sc;

    // BCrypt is a password Hashing
    // Function based on Blowfish
    // Algorithm.
    public static String Password_Hash(
        String password)
    {
        return BCrypt.hashpw(
            password, BCrypt.gensalt());
    }

    // Verifying password with the
    // hashed password.
    public static boolean Verify_Password(
        String password,
        String hashed_password)
    {
        return BCrypt.checkpw(
            password, hashed_password);
    }

    public static void main(
        String args[]) throws Exception
```

```

{

    // Scanner class instance connected
    // to the Input Stream(System.in)
    sc = new Scanner(System.in);

    System.out.println(
        "Enter the password: ");

    // Scanner class instance
    // reading the user input
    String p = sc.nextLine();

    // Generate hashed password
    String passwordHash
        = Password_Hash(p);

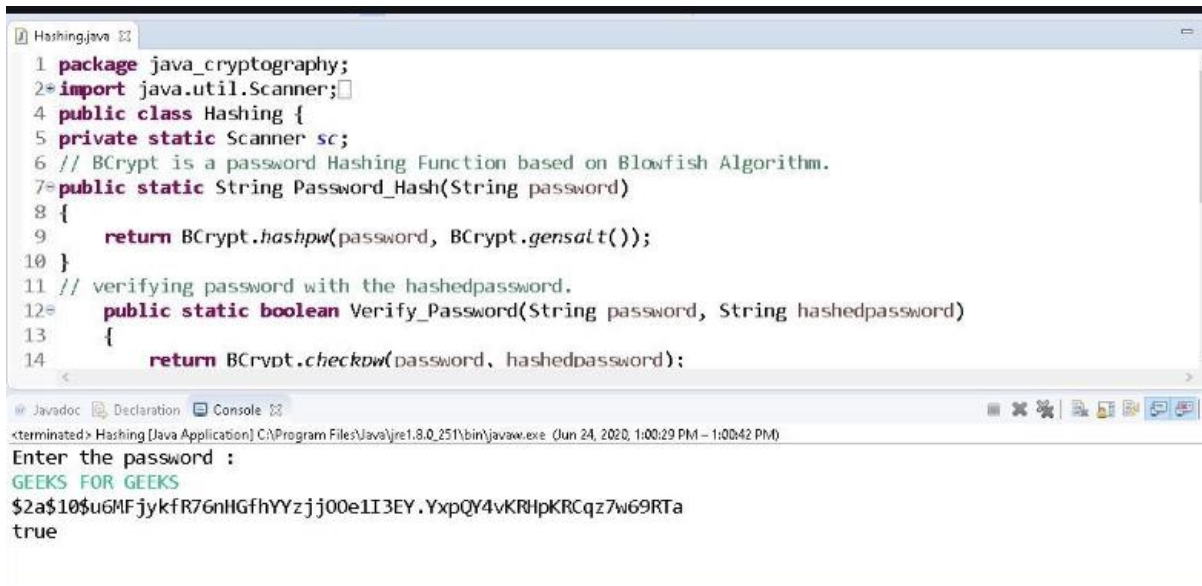
    // Print Hashed Password
    System.out.println(
        "Hashed-password: "
        + passwordHash);

    // Printing the result of verification
    // of hashed password
    // with original password
    System.out.println(
        "Verification: "
        + Verify_Password(
            p, passwordHash));

}
}

```


OUTPUT



The screenshot shows a Java IDE with a file named `Hashing.java` open. The code defines a `Hashing` class with two static methods: `Password_Hash` and `Verify_Password`. The `Password_Hash` method uses `BCrypt.hashpw` to hash a password. The `Verify_Password` method uses `BCrypt.checkpw` to verify a password against its hash. The console output shows the program running, prompting for a password, and displaying the resulting hash and the verification result.

```
1 package java_cryptography;
2 import java.util.Scanner;
3
4 public class Hashing {
5     private static Scanner sc;
6     // BCrypt is a password Hashing Function based on Blowfish Algorithm.
7     public static String Password_Hash(String password)
8     {
9         return BCrypt.hashpw(password, BCrypt.gensalt());
10    }
11    // verifying password with the hashedpassword.
12    public static boolean Verify_Password(String password, String hashedpassword)
13    {
14        return BCrypt.checkpw(password, hashedpassword);
15    }
16 }
```

Enter the password :
GEEKS FOR GEEKS
\$2a\$10\$u6MFjykfR7GnHGfhYYzjj00e1I3EY.YxpQY4vKRHpKRCqz7w69RTa
true

JUSTIFICATION

- The purpose of the project is to provide a practical survey of both the principles and practice of cryptography. Cryptography encompasses a large number of algorithms which are used in building secure applications.
- The fourth step is to implement your encryption solutions using the appropriate tools and technologies. You can use software or hardware solutions, or a combination of both, depending on your data assets and requirements. For example, you can use encryption software to encrypt files, folders, disks, databases, emails, or cloud storage. You can also use encryption hardware to encrypt devices, such as laptops, USB drives, or smartphones. You should also ensure that your encryption solutions are compatible with your data formats, applications, systems, and network

ADVANTAGES OF RSA TECHNIQUE

The RSA (Rivest-Shamir-Adleman) encryption technique offers several advantages, making it one of the most widely used public-key cryptosystems for secure data transmission and digital signatures. Here are some key advantages of the RSA technique:

***Security*:** RSA is based on the mathematical difficulty of factoring large composite numbers. The security of RSA relies on the practical impossibility of factoring the product of two large prime numbers. This makes it highly secure and resistant to attacks by conventional computers.

***Public and Private Keys*:** RSA uses a pair of keys – a public key for encryption and a private key for decryption. The separation of keys ensures that the encryption key can be openly shared, while the decryption key is kept secret, providing security without requiring a shared secret between communicating parties.

***Digital Signatures*:** RSA is widely used for digital signatures. By encrypting a message with a private key, anyone can verify the authenticity and integrity of the message using the corresponding public key. This feature is crucial for secure authentication and data integrity.

***Asymmetric Encryption*:** RSA is an asymmetric encryption technique. Asymmetric encryption is slower than symmetric encryption, but it provides a crucial advantage in secure key exchange. It allows two parties to establish a secure communication channel without sharing a common secret in advance.

***Key Management*:** RSA offers key management benefits. The ability to change keys without requiring a change in the encryption process simplifies key rotation and enhances security.

***Secure Data Transmission*:** RSA is often used to secure data transmission over insecure networks like the internet. It ensures that data exchanged between parties remains confidential and cannot be intercepted by unauthorized entities.

***Open Standards*:** RSA is widely adopted and supported by various cryptographic libraries, making it an open and interoperable standard for encryption and digital signatures.

***Quantum-Resistant*:** While RSA is vulnerable to quantum computing attacks, post-quantum cryptography methods are being developed to counter this threat. RSA can be upgraded to quantum-resistant cryptography to maintain security in the post-quantum era.

***Versatile Use Cases*:** RSA is versatile and can be used for various applications, including secure email, SSL/TLS encryption for web communication, secure data storage, digital certificates, and secure authentication.

***Longevity*:** RSA has proven its longevity and security over several decades, making it a trusted and well-established encryption technique.

***High Trust*:** The RSA algorithm is widely used and well-known, which enhances trust in the security of systems that implement it.

CONCLUSION

The cryptographic algorithms vary in terms of parameters which includes encipherment and decipherment time, memory, throughput and CPU utilization. This research analyzes the want to improve a combine encipherment algorithm that mixes various encipherment algorithms on the basis of all appropriate factors that are used to increase the overall safety and security of encipherment methods.

REFERENCES

- P. Pfleeger, S. L. Pfleeger, and J. Margulies, Security in Computing. New Jersey: Prentice Hall, 2015.
- M. Mushtaq Faheem, S. Jamel, A. Hassan Disina, Z. A. Pindar, N. Shafinaz Ahmad Shakir, and M. Mat Deris, "A survey on the cryptographic encryption algorithms," Int. J. Adv. Comput. Sci. Appl., 8(11), 2017, pp. 333-344.
- G. Suman, C. Krishna, and M. T. Se, "Improved cryptosystem using SDES algorithm with substitution ciphers," International Journal of Advanced Research in Computer Science and Software Engineering, 3(7), 2013, pp. 131-136.
- S. N. Habib, R. Awan, and W. Haider, "A modified simplified data encryption standard algorithm," International Journal of Computer Science and Software Engineering, 6(7), pp. 152-154, 2017