Volume 67, Issue 4, 2023
Journal of Scientific Research
of
The Banaras Hindu University

Applications of Multi-Reverse Primes

Deepak kumar sharma[*1], Arvind bhatt[2], and Priti singh[3]

[1] Department of Mathematics, Uttarakhand open University, India, dksharma@uou.ac.in
[2] Department of Mathematics, Uttarakhand open University, India, arvindbhatt@uou.ac.in
[3] Department of Mathematics, Patna Science College, Patna, India, pritisingh.mnnit@gmail.com

Abstract: **In this manuscript we are introducing an application related to multi-reverse primes in cryptography. This is expansion of the work related to multi-reverse primes and their distributions.**

Index Terms: Cryptography, Multi -Reverse Primes, Prime Numbers.

I. INTRODUCTION

As we know that the number theory is a queen of Mathematics. Numbers are important in our life. Mathematicians have long been interested by the study of numbers, especially prime numbers. On the other side, people have always felt the need for security while sharing information. A strong link between the two disciplines has gradually grown over the past twenty years as a result of the development of new mathematical techniques and the astounding advancements in computing. The most secure information transmission techniques currently in use, which have recently benefited from the growth of e-commerce, are based on algorithms that take advantage of unique qualities of prime numbers.

In the previous paper **Sharma, D. K., (2021)** define Distributions of Multi – Reverse Primes within the Given interval & Their Application in Asymmetric Cryptographic Algorithm.

In the present paper we are defining an algorithm for improving the security level of data. The multi-reverse prime numbers are using for this algorithm. In this manuscript we are also defining a Java programme for making encryption and decryption key.

The followings definitions will be needed in the sequel.

Definition 1.1. Let p, q, and r be three different primes having x, y, and z digits respectively, then two numbers $P_{x,y,z}$ = p q r and $P_{z,y,x}$ = r q p such that $P_{x,y,z} \neq P_{z,y,x}$ are called Multi-reverse primes.

The following are the examples:

$P_{1,1,2}$ = 3217 and $P_{2,1,1}$ = 1723 are multi-Reverse primes as $3217 \neq 1723$.

Definition 1.2. The process of converting a plaintext into ciphertextic called ‘encryption’.

Definition 1.3. The process of converting a ciphertext into plaintext is called ‘decryption’.

II. ALGORITHM

In this section we are defining an Algorithm for multi-reverse prime which is more advanced Algorithm than Rivest, Shamir, Adleman (RSA) Algorithm [11].

Step-1: Choose a pair of big Multi-reverse primes $P_{x,y,z} \neq P_{z,y,x}$.

[*] Deepak kumar sharma

DOI: 10.37398/JSR.2023.670410

Step-2: Find modulus,

$n = P_{x,y,z} * P_{z,y,x}$ **…….(1)**

Step-3: Find the totient function;
$\phi(n) = (P_{x,y,z}-1)(P_{z,y,x} -1)$ **…….(2)**

Step-4: Pick an integer e, in this manner e is co-prime to $\phi(n)$ and $1 < e < \phi(n)$.
The couple of numbers (n, e) build up the public key.

Step-5: Compute d such that
$e.d = 1 \bmod \phi(n)$. **… (3)**
d can be compute using the extended Euclidean algorithm. The private key can be constituted as the pair (n, d).

Encryption: Designated a plaintext P, be elected by the text as numbers using ASCII values, then the ciphertext C is deliberate make use of public key (n,e) as, $C = P^e \bmod n$.

Decryption: Using the private key (n, d), the plaintext can be found using, $P = C^d \bmod n$. Constitute the numbers as alphabets using ASCII values to observe the original text.

III. JAVA PROGRAM

In this section we are defining a Java program for obtaining encryption and decryption keys. The Java program for encryption and decryption are as given below.

```
import java.util.*;
import java.math.*;
public class rsa {

    private static final Scanner sc = new
Scanner(System.in);
    private int p, q, n, z, d = 0, e, i;
    public rsa() {

        System.out.println("Enter 1st multireverse
prime number p");
        p = sc.nextInt();
        System.out.println("Enter 2nd multireverse
prime number q");
        q = sc.nextInt();
        n = p * q;
        z = (p - 1) * (q - 1);
        System.out.println("the value of z = " + z);
        for (e = 2; e < z; e++) {
            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);
            if (x % e == 0) //d is for private key
exponent
            {
                d = x / e;
                break;
            }
        }

        System.out.println("the value of d = " + d);
    }
    private static int gcd(int e, int z) {
        if (e == 0) {
            return z;
        } else {
            return gcd(z % e, e);
        }
    }
    double encrypt(int msg) {
        return (Math.pow(msg, e)) % n;
    }
    double[] encrypt(String msg) {
        int[] charactersAsNumbers = new
int[msg.length()];
        for(int i = 0; i<msg.length(); i++) {
            charactersAsNumbers[i] =
msg.codePointAt(i);
        }

        System.out.println("Plain text as sequence of
numbers: " +
Arrays.toString(charactersAsNumbers));
        double[] encryptedMsg = new
double[msg.length()];
        for(int i = 0; i<charactersAsNumbers.length;
i++) {
            encryptedMsg[i] =
encrypt(charactersAsNumbers[i]);
        }
        return encryptedMsg;
```

```
    }

    BigIntegerdecrypt(double encrypted) {
        BigInteger N = BigInteger.valueOf(n);
        BigInteger C =
BigDecimal.valueOf(encrypted).toBigInteger();
        return (C.pow(d)).mod(N);
    }

    String decrypt(double[] encrypted) {
        StringBuilder builder = new StringBuilder();
        for(double encryptedCharacter: encrypted) {
            BigIntegerdecryptedCharacter =
decrypt(encryptedCharacter);
            builder.append(Character.toChars(decrypted
Character.intValue()));
        }
        return builder.toString();
    }
    public static void main(String args[]) {
        System.out.println("Enter the text to be
encrypted and decrypted");
        String msg = sc.nextLine();
        rsa rsa1 = new rsa();
        double[] c = rsa1.encrypt(msg);
        System.out.println("Encrypted message is: " +
Arrays.toString(c));
        String msgBack = rsa1.decrypt(c);
        System.out.println("Decrypted message is: " +
msgBack);
    }
}
```

Remark 3.1: The following example is proving the above Java program which shows that encryption and decryption keys are generating which is a very interesting example.

Enter the plain text (P) to be encrypted and decrypted: **deepak**
Enter 1st multi-reverse prime number $P_{\alpha,\beta,\gamma}$ as p = **1723**
Enter 2nd multi-reverse prime number $P_{\gamma,\beta,\alpha}$ as q = **3217**
The value of z = **5537952**
The value of e = **5**
The value of d = **2215181**
Plain text as sequence of numbers: [**100, 101, 101, 112, 97, 107**]
Encrypted message is: [**624636.0, 779165.0, 779165.0, 2566343.0, 1402098.0, 2003077.0**]
Decrypted message is: **deepak**

IV. CONCLUSION

The proposed algorithm is safe and sound as the multi-reverse primes are used in the RSA algorithm to enable the high level of security and to generate such an algorithm which is more secure than the previously generated algorithms. New algorithms can also be obtained from the proposed encryption & decryption method, which will be helpful in providing the greater level of security. Also proposed java program for Making encryption and decryption key. In future, we will try to generate multi-stage cryptographic algorithm for the security of data using different primes.

Acknowledgement: The critical comments of the referee are gratefully acknowledged.

V. REFERENCES

1. Sharma, D.K., Agarwal, S., & Uniyal, A.S. (2021), Distribution of Multi-Reverse Primes within the given interval & Their application in Asymmetric Cryptographic Algorithm. International Journal of Applied Engineering & Technology, Vol.3(1), 29-33.
2. Agarwal, S., (2021), Data Management Security and Integrity using Cryptography. Sankalpa Journal of Education IT and Management Research, 2319-7455, Issue 10, pp. 49-55.
3. Agarwal, A., Agarwal, S. & Singh, B.K. (2020), Algorithm for Data Encryption & Decryption Using Fibonacci Primes. Journal of Mathematical Control Science and Applications (JMCSA), Vol. 6, No. 1, pp. 63-71.
4. Agarwal, S.,(2019), Encryption & Decryption Using Linear Algebra: Advancement in Public Key Cryptography. Indian Journal of Economics and Business (IJEB), ISSN: 0972-5784, Vol. 18, No.1, pp. 167-180.
5. Agarwal, S. and Uniyal, A.S.(2018), Algorithms for Number Theoretic Functions & Special Numbers. International Journal of Research in Engineering, Science and Management (IJRESM), ISSN: 2581-5792, Vol-1, Issue-12, pp. 112-116.
6. Agarwal, S. and Uniyal, A.S.(2017), Enhancing the Security of ATM Password using Multi-dimensional Tree. International Journal of

Mathematics Research (IJMR), ISSN: 0976-5840, Volume 9, No. 1, pp. 53-58.
7. Agarwal, S. and Uniyal, A.S.(2015), Prime Weighted Graph in Cryptographic System for Secure Communication. International Journal of Pure and Applied Mathematics (IJPAM), (ISSN: 1311-8080), Volume 105, No. 3, pp. 325-338.
8. Agarwal, S. and Uniyal, A.S.(2015), Multiprimes Distribution within a Given Norms. International Journal of Applied Mathematical Sciences (JAMS), (ISSN 0973-0176), Volume 8, Number 2, pp. 126-132.
9. Agarwal, S., Uniyal, A.S.(2015), Elliptic Curves: An Efficient and Secure Encryption Scheme in Modern Cryptography. International Journal of Advance Research in Science & Engineering (IJARSE), Volume 04, Issue 03, pp. 134-143.
10. Agarwal, S. and Uniyal, A.S.(2009), Application of Marvelous Ternary Codes in Classical Cryptosystem. Stochastic Modeling and Applications (SMA), (ISSN 0972-3641), Volume 13, Number 2, pp. 41-49.
11. Apostal, T.M. (1989), Introduction to analytic number theory, Springer Verlag, New York.
12. Rivest, R.L., Shamir, A and Adleman, L.M. (1978), "A Method for Obtaining Digital Signature and Public-key Cryptosystems," Communications of the ACM, Vol. 21, No. 2, pp. 120–126.
13. Diffie, W.,Hellman, M. (1976), New Direction in Cryptography, IEE Transaction in Information Theory IT.