A modified RSA encryption algorithm

1st Prof. Dr. Reena Kharat
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Savitribai Phule Pune University, Pune
reena.kharat@pccoepune.org

2nd Bhavesh Gulabani
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Savitribai Phule Pune University, Pune
bhaveshgulabani1@gmail.com

3rd Chinmay Joshi
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Savitribai Phule Pune University, Pune
chinmayjoshi9911@gmail.com

4th Saurabh Damle
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Savitribai Phule Pune University, Pune
saurabhd.1108@gmail.com

5th Shubhankar Gengaje
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Savitribai Phule Pune University, Pune
shubhu311@gmail.com

***Abstract*—Asymmetric encryption, also known as public key encryption, is a form of cryptosystem in which encryption and decryption are performed using two different keys-a public key and a private key. RSA being the most widely used public key cryptosystem provides robust confidentiality. However, the security of RSA depends on the prime factorization of the integer value n which is globally visible in the system. In this paper, a modified version of the RSA algorithm is proposed which further strengthens the security of RSA by extending the algorithm with additional transformations. These transformations include modifying the final RSA cipher by applying an equation which is not known publicly and then shifting the cipher by a calculated number of bits. This enhances the security of RSA making it difficult for an external entity to access the confidential information.**

Keywords — RSA algorithm, Public key cryptography, asymmetric key cryptography

I. Introduction

From the earliest beginnings to modern times, virtually all cryptographic systems have been based on the elementary tools of substitution and permutation. The development of public key cryptosystems is one of the greatest revolutions in the entire history of cryptography. These systems are based on mathematical functions rather than substitution and permutation. Additionally, public key cryptosystems are asymmetric involving the use of two separate keys, as compared to symmetric encryption which uses only one key. The elements of public key cryptosystems are plaintext, an encryption algorithm, public and private keys, ciphertext, and a decryption algorithm. A number of algorithms have been proposed for public-key cryptography, however quite a few of these have turned out to be breakable. The RSA algorithm developed by Ron Rivest, Adi Shamir, and Len Adleman at MIT has proven to be a robust and widely accepted public key encryption algorithm. Its security is based on the difficulty of factorization of large prime numbers. However, there may exist cases where the numbers are not sufficiently large which can lead to a potential vulnerability in the system. This paper proposes an extension to the existing RSA scheme by applying additional transformations to further enhance the security. The contents of this paper are further divided as follows: Section II contains the relevant literature review for the algorithm, Section III briefly explains the existing RSA technique and Section IV explores the modified version of RSA. Section V concludes the paper providing an insight into the future research possible.

II. Literature review

R.L. Rivest, A. Shamir, and L. Adleman [1] proposed a method for implementing a public-key cryptosystem whose security relies on the difficulty of factoring large prime numbers. Their research provided an insight into the core mathematical equations of the algorithm. The major advantage of this technique was that courier or other secure means were not needed to transmit the message since the message could be enciphered using an encryption key publicly revealed by the intended recipient. Also, the message could be "signed" using a privately held decryption key. Thus, verification could be done by anyone using the corresponding public revealed encryption key. The authors suggested that the security could be further examined especially, the factoring of large prime numbers. Gabriel Vasile Iana1, Petre Anghelescu1 and Gheorghe Serban [2] in their research provided an implementation of the complete RSA algorithm on FPGA and showcased their results. In their paper titled, "A Modified RSA Encryption Technique Based on Multiple public keys", Amare Anagaw Ayele1 and Dr. Vuda Sreenivasarao[3] proposed a modified version of the original RSA algorithm which employed multiple public keys. The authors have set the number of keys to four in their research. This enhanced the security of the technique and provided better confidentiality while transmitting sensitive data. However due to an increased number of keys, the computation time increased drastically. Further research could be conducted to reduce the computational complexity. William Stallings [6] in his book, "Cryptography and Network Security Principles and Practice", explained the various algorithms used in cryptography and provided a thorough description of the underlying math. The author gave an insight about the RSA algorithm, along with its advantages and disadvantages in Chapter 9 of the book. The steps proved to be an inspiration for the enhancement proposed in this paper.

III. RSA ALGORITHM

The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and $n - 1$ for some n. The RSA technique can be briefly stated as follows:

1. Select two prime numbers p and q
2. Calculate $n = p \times q$
3. Calculate $\emptyset(n) = (p-1) \times (q-1)$
4. Select an integer e such that e is relatively prime to $\emptyset$(n) and less than $\emptyset$(n) i.e., $\gcd(\emptyset(n), e) = 1; 1 < e < \emptyset(n)$
5. Calculate d such that $d \equiv e^{-1}(mod\ \emptyset(n))$. The value of *d* can be calculated using Extended Euclidean Algorithm
6. Set Public Key, $PU = \{e, n\}$
7. Set Private Key, $PR = \{d, n\}$
8. Sender side – Encryption
    a. Select plain text, *M* such that $M < n$
    b. Calculate cipher text, $C = M^e \ mod\ n$
    c. Transmit cipher text
9. Receiver side – Decryption
    a. Receive cipher text, *C*
    b. Calculate plain text, $M = C^d \ mod\ n$

For a string of alphabets, this scheme can be applied to each letter converted into its corresponding numerical format.

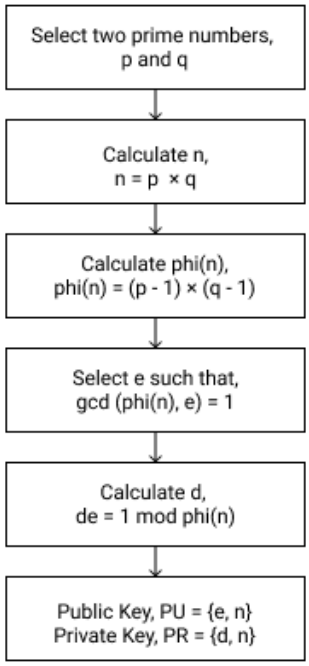
IV. PROPOSED RSA ALGORITHM

This paper proposes a modified RSA algorithm based on transformations applied to the cipher text, *C* at the sender side before transmitting to the receiver. The receiver then regains the original cipher and then performs the decryption as per the original RSA technique. The modified version can be stated as:

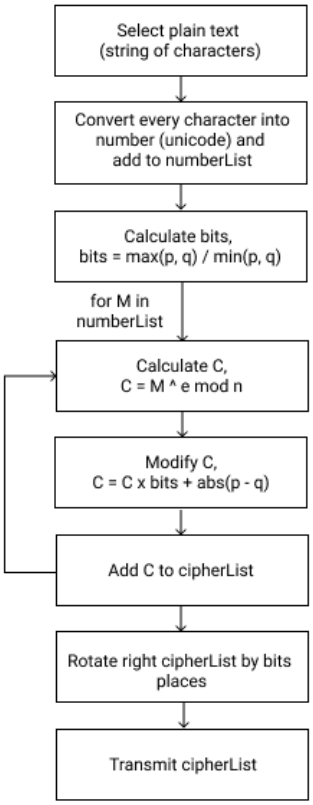
1. Select two prime numbers p and q
2. Calculate $n = p \times q$
3. Calculate $\emptyset(n) = (p-1) \times (q-1)$
4. Select an integer e such that e is relatively prime to $\emptyset$(n) and less than $\emptyset$(n) i.e., $\gcd(\emptyset(n), e) = 1; 1 < e < \emptyset(n)$
5. Calculate d such that $d \equiv e^{-1}(mod\ \emptyset(n))$. The value of *d* can be calculated using Extended Euclidean Algorithm
6. Set Public Key, $PU = \{e, n\}$
7. Set Private Key, $PR = \{d, n\}$
8. Sender side – Encryption
    a. Select plain text, *S* which is a string of characters
    b. Convert each character into an integer based on its corresponding Unicode value and append it to a list called *numberList*.
    c. Calculate $bits = \left\lceil \frac{\max(p,\ q)}{\min(p,\ q)} \right\rceil$, initialize *cipherList* as an empty list.
    d. For each number *M* in *numberList*:
        i. Calculate $C = M^e \ mod\ n$
        ii. Modify $C = C \times bits + |p - q|$
        iii. Append *C* to *cipherList*
    e. Rotate right the *cipherList* by *bits* places
    f. Transmit *cipherList*
9. Receiver side – Decryption
    a. Receive *cipherList*
    b. Calculate $bits = \left\lceil \frac{\max(p,\ q)}{\min(p,\ q)} \right\rceil$, initialize *plainTextList* as an empty list.
    c. Rotate the *cipherList* left by bits places.
    d. For every number *C* in *cipherList:*
        i. Regain $C = \frac{(C - |p-q|)}{bits}$
        ii. Calculate $M = C^d \ mod\ n$
        iii. Append M to *plainTextList*
    e. Convert each integer into a character based on its corresponding Unicode value and append it to the *plainTextString*

The flowcharts for the proposed technique are:

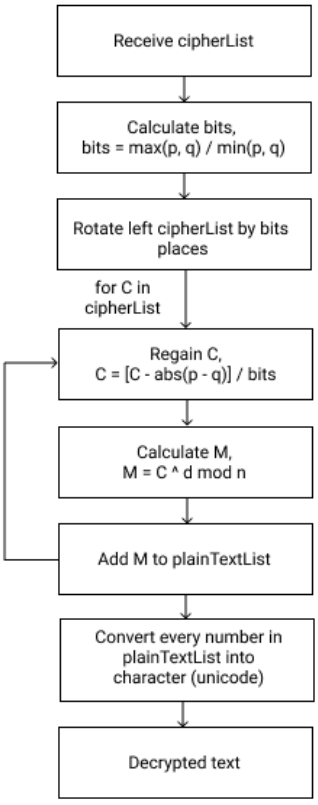
1. Initialization of variables:



2. Sender side – Encryption



3. Receiver side – Decryption



V. CONCLUSION

The proposed encryption algorithm is an extended version of the previously robust RSA algorithm. The transformations applied are simple mathematical equations which can be customized as per the algorithm design and its use case. Further research can be conducted to identify a mathematically strong transformation which can serve as a universal extension to the RSA approach.

REFERENCES

[1] Rivest, R, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM 21 (2): 120–126, doi: 10.1145/359340.359342, 1977.

[2] Gabriel Vasile Iana1, Petre Anghelescu1, Gheorghe Serban, "RSA encryption algorithm implemented on FPGA", University of Pitesti, Department of Electronics and Computers, Romania, Arges, Pitesti, Str. Targul din Vale, No. 1, Code: 110040.

[3] Amare Anagaw Ayele1 Dr. Vuda Sreenivasarao, "A Modified RSA Encryption Technique Based on Multiple public keys", IJIRCCE 2013.

[4] Sonal Sharma,Prashant Sharma,Jitendra Yadav, "Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm" IJARCSSE2012.

[5] Andrzej Chmielowiec, "*Fixed points of the RSA encryption algorithm",* Elsevier 2009.

[6] William Stallings, "Cryptography and Network Security Principles and Practice", Fifth edition, Prentice Hall.