Indian J Pure Appl Math (2024) 55:1477–1484
https://doi.org/10.1007/s13226-023-00452-9

ORIGINAL RESEARCH

Encryption and decryption of signed graph matrices through RSA algorithm

Obaidullah Wardak · Deepa Sinha · Anshu Sethi

Received: 21 November 2022 / Accepted: 19 June 2023 / Published online: 4 July 2023

Abstract Information security and confidentiality are burning issues in today's digitally connected world. Encryption is considered a fast-moving trend in this reference. Although we see rigorous enhancements in the development of information security tools, storage and retrieval, enormous challenges still occur due to unsecured transmission channels, hacking tools and the ubiquity of the Internet. This paper proposes a new cryptography mechanism for secure data transmission and retrieval using a signed graph, its adjacency matrix and the RSA algorithm.

Keywords Signed graph · Encryption · Decryption · Algorithm · RSA algorithm

1 Introduction

Secret communications have been needed by military officers and diplomats since ancient times. In today's advanced age, where the internet, mobile phones and computer technology are widely used in almost every sphere of life, the need to keep important information secure and confidential is also increasing day by day. Overtime, as data security continues to evolve, new ways to break the confidential communication are being discovered.

Cryptography is the study of changing restricted information into coded data so that can securely arrive at its end without spillage. It was fundamentally used for wartime plans. Old-style cryptography revisits 2000 years. Modern cryptography was established by Shannon in 1949 [5]. New forms of cryptography have emerged as a result of the advancement of digital communications. It deals with issues such as confidentiality, privacy, authentication, passwords, digital signatures, identification, digital money and network security. It is now an essential component of modern society.

The process to transform the original message into a code format is called the encryption and the reverse process is known as decryption [4]. Encryption protects the original contents from eavesdropping. Plaintext is

Communicated by Shariefuddin Pirzada.

Deepa Sinha and Anshu Sethi have authors contributed equally to this work.

O. Wardak · D. Sinha (✉)
Mathematics, South Asian University, New Delhi, Delhi 110068, India
E-mail: deepasinha@sau.ac.in

O. Wardak
E-mail: obaid.wardak22@live.com

A. Sethi
Mathematics, The NorthCap University, Gurugram, Haryana 122017, India
E-mail: anshumalhotra@ncuindia.edu

a non-encrypted message. With the help of a specified key, plaintext information is encoded or encrypted to generate cipher text in any encryption technique. This cipher text is then decrypted and converted into a readable message.
A key is a piece of information that is used to encode the original data and then decrypt it to obtain the true text. The authorized recipient can easily open the hidden message using the provided key, but an interceptor cannot. Mainly, three types of schemes are used in modern cryptography, i.e., symmetric key cryptography, asymmetric or public key cryptography and hash functions [9]. Symmetric key cryptography implements a single key for both encryption and decryption, whereas public key cryptography implements one key for encryption and another for decryption. To encrypt information irreversibly, hash functions implement a transformation.

The study of signed graphs as networks is of great importance, where signed graphs can be used as matrices and these conversion can be implemented in various encryption and decryption techniques [6], where securities are represented as nodes of a signed graph and edges represent the correlation between the securities. The idea takes it into network security [3].

A *signed graph* [1] is defined as an ordered pair $S = (S^u, \sigma)$, where S^u is a graph $G = (V, E)$, called the underlying graph of S and $\sigma : E \rightarrow \{+1,-1\}$ is a mapping from the edge set E of S^u into the set $\{+1,-1\}$, called the *signature* of S.

The number of positive (negative) edges incident at a vertex v, denoted by $d^+(v)$, $(d^-(v))$, is called *positive (negative)* degree of the vertex v in S. The *total degree d(v)* of the vertex v in S is the sum $d(v) = d^+(v) + d^-(v)$. A signed graph can be represented as a network using matrices, therefore, encryption and decryption mechanism can be applied to convert a network (or signed graph) to another network and vice-versa. Further, to add more security, Ron Rivest, Adi Shamir and Leonard Adleman (RSA) algorithm [2], with high end of security keys is implemented.

1.1 RSA algorithm

In 1977 an algorithm was described by Ron Rivest, Adi Shamir and Leonard Adleman (RSA) and was the first publically known example of a secure, practical algorithm that could be used to encrypt secret messages without somehow establishing a secret key beforehand. Before the 1970's, nobody even had any reason to think that such a remarkable thing should even be possible [2].

As a public-key cryptography system, RSA algorithm calls on the intended recipient of an encrypted message to create a public and private key. He publishes the public key, which anyone can then use to encrypt messages to him and keeps private key secret. In the case of the perfect code system, the ability to create a public/private keypair depended on the ease with which one can create graphs with known perfect codes, in spite of the difficulty associated with finding them.

In the RSA algorithm public/private key-pair generation is based on the difficulty of factoring large integers. In real-life computer implementations of RSA algorithm, the numbers which would have to be factored to break the system are hundreds of digits long and would take even the worlds fastest computers thousands or millions of years to factor [2].

1.2 Mechanism for generating public and private key in RSA algorithm

For generating public key:

1. Select two large prime numbers p and q (say) and calculate $N = p * q$.
2. Choose a number e (say) such that e must be:
   2.1. An integer.
   2.2. Relatively prime to N, i.e., $\gcd(e, N) = 1$.
   2.3. $1 < e < \phi(N)$, where $\phi(N)$ is Euler's totient function [2].

Then, public key is (e, N).
For generating private key:

1. Calculate the value of $\phi(N)$, such that $\phi(N) = (p - 1)(q - 1)$.
2. Calculate the value of d (say) such that $(d \times e) \pmod{\phi(N)} = 1$.

Then, private key is (d, N).
The public and private keys are used for encryption and decryption respectively.

Based on the concept of *signed graphs* [6] and to add more security to a network, an effective mechanism is designed for encryption and decryption of matrices through signed graphs by RSA algorithm. Further, this concept is implemented on a network to explain the numerical interpretation of the given mechanism.

2 Proposed mechanism for encryption and decryption

Let the adjacency matrix of the signed graph S with n vertices $\{v_1, v_2, \ldots v_n\}$ and no parallel edges be represented as, $A = [a_{ij}]$, where $a_{ij} \in \{-1, 0, 1\}$ for $i, j = 1, 2, ..., n$ such that

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \text{ is a positive edge.} \\ -1 & \text{if } v_i v_j \text{ is a negative edge.} \\ 0 & \text{if } v_i v_j \text{ is not an edge.} \end{cases}$$

The adjacency matrix related to a signed graph S is symmetric [10]. Due to this presentation of the matrices, the upper triangular matrix U (say) is same as L^T and L^T denotes the transpose of lower triangular matrix L (say). In general, the adjacency matrix is representing as follows:

$$A = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \ddots & \ddots & \ddots & \ddots \\ a_{1n} & a_{2n} & \cdots & 0 \end{pmatrix}$$

Where $a_{ij} = a_{ji}$.
Upper triangle or lower triangle matrix is enough to recognize the adjacency matrix. Let us consider only the upper triangular matrix for the encryption and decryption process.

2.1 To calculate the numerical value for the signed graph S by applying the proposed mechanism

Write all the elements of an upper triangular matrix as a number starting from the last row and for each row the direction of writing should be right to left. Number of digits in this number is $r = \frac{n^2-n}{2}$, where n is the number of vertices in the signed graph S.

Now, every -1 is considered itself as a decimal number and remaining which is string of only 0 and 1 separated by -1 are considered as binary numbers.

In next step, we add 1 in the beginning and end of every binary number and convert all the binary numbers into decimal numbers, where -1 itself is considered as a decimal number. Thus, we will have a string of decimal numbers $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$.

For **encryption**, the values $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$, are encrypted by applying RSA algorithm to get the encrypted numbers ($ciphertext$) $\alpha_1, \alpha_2, \ldots, \alpha_k$ (say), such that $\alpha_i = (d_i)^e \pmod N$, where $1 < k \leq r$, and this $ciphertext$ is sent to receiver who can derive the original matrix by applying some reverse process.

For **decryption**, after receiving the encrypted values (cipher text) $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $1 < k \leq r$. t is decrypted to get the values of $d_1, d_2, \cdots, d_k$, such that $d_i = (\alpha_i)^d \pmod N$, where $1 < k \leq r$. Further, every $d_i, i = 1, \ldots, k$ except -1 is converted to binary numbers to construct the upper triangular matrix U.

The order of constructing the matrix is row-wise. Here the first right most digits represent the last entry of the top row to last row. By continuing this process the upper triangular matrix is derived. The lower triangular matrix is constructed by getting the transpose of the upper triangular matrix and the adjacency matrix of given signed graph is produced by putting them together.

2.2 Algorithm to encrypt and decrypt the signed graph S

Encryption of the Adjacency Matrix:
Input: Adjacency matrix A
Output: Numerical values $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$
Step 1: $n :=$ Number of rows in A.
Step 2: $i := 1$

2.1 $j := 1$

2.1.1 $a_{ij} := 0$

2.1.2 $j := j + 1$

2.1.3 if $j < i$, then go to (2.1.1), else go to (2.2)

2.2 $i := i + 1$

2.3 if $i < n$, then go to **step 1**, else return U as an upper triangular matrix.

Step 3: $n := (n-1)^{th}$ triangular number.
Step 4: $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$.
Step 5: $i := 1$

5.1 $j := i + 1$

5.1.1 $d = a_{ij}$ if $a_{ij} = -1$, else go to (5.1.2)

5.1.2 $d := d + a_{ij} \times 2^n$

5.1.3 $j := j + 1$

5.1.4 if $j < n$, go to (5.1.1), else go to (5.2)

5.2 $i := i + 1$

5.3 if $i < n - 1$, go to (5.1), else return the decimal value.

Step 6: Return the decimal values, i.e., $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$.
Step 7: Apply the RSA algorithm to receive the values of $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $1 < k \leq r$.
Decryption to receive the Adjacency Matrix:
Input: Numerical values, i.e., $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $1 < k \leq r$.
Output: Adjacency matrix A
Step 1: Apply the RSA algorithm on the values $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $1 < k \leq r$, to receive $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$.
Step 2: Without -1 convert all the decimal numbers $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$, into binary numbers. Construct the required upper triangular matrix.
Step 3: $N :=$ length of the string (number of digits) d (say), which includes all $d_1, d_2, \ldots, d_k$, where $1 < k \leq r$. Determine k-th triangular number, N.
Step 4: $k := 0$.
Step 5: $s := 0$.
Step 6: Check whether $s \neq N$, if so go to (6.1), else go to **step** 7

6.1 $k := k + 1$

6.2 $s := s + 1$

6.3 go to **step** 6

Step 7: $i := 1$

7.1 $j := 1$

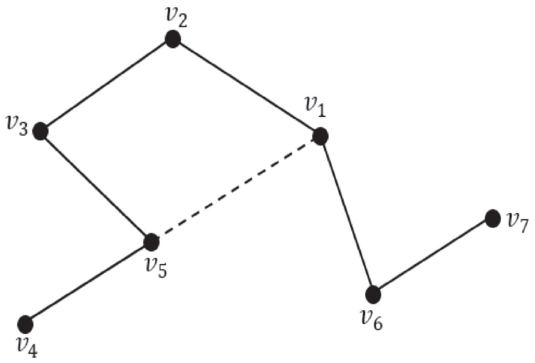

Fig. 1 Signed graph S

7.2 $U(i, j) := d(r)$

7.3 $j := j + 1$

7.4 if $j < k + 1$, then go to (7.1), else go to (7.2)

7.5 $i := i + 1$

7.6 if $i < k$, then go to (7.1), otherwise go to step 8.

Step 8: Add a row with all zero elements to matrix U.
Construct the required lower triangular matrix.
Step 9: Transpose the defined matrix U, i.e. $U^T = L$ and find the given adjacency matrix of signed graph
Step 10: $A := U + L$.

2.3 Numerical interpretation

Consider the signed graph S as shown in **Figure** 1.
The adjacency matrix of S is:

$$S = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Number of elements in the given upper triangular matrix is $r = (n^2 - n)/2$ which is $(7^2 - 7)/2 = 21$. Applying encryption process, all the elements of the given upper triangular matrix will be represented as: 10000100100000101-1001.
To convert two binary numbers to decimal numbers and one -1, first write the decimal number 1 at the beginning and end of every binary number. Hence,
$d_1 = 1100001001000001011$, $d_2 = -1$ and $d_3 = 10011$.
Firstly, convert the binary values of d_1 and d_3 into decimal numbers.
$d_1 = 1 \times 2^{18} + 1 \times 2^{17} + \cdots + 1 \times 2 + 1 \times 2^0 = 397835$,
$d_3 = 1 \times 2^4 + 0 \times 2^3 + \cdots + 1 \times 2 + 1 \times 2^0 = 19$.
Therefore, for implementing encryption and decryption using RSA algorithm, we have 3 values, i.e., $d_1 = 397835$, $d_2 = -1$ and $d_3 = 19$.

Encryption
Input values of d_1, d_2 and d_3.

1. Choose two primes $p = 397$ and $q = 401$.
2. Compute $N = p \times q = 397 \times 401 = 159197$.
3. Compute $\phi(N) = (p-1) \times (q-1) = 396 \times 400 = 158400$.
4. Choose e such that $1 < e < \phi(N)$ and $gcd(e, N) = 1$; $e = 23$.
5. Compute d such that $(d \times e) \pmod{\phi(N)} = 1$; $(d \times 23) \pmod{158400} = 1$ then $d = 6887$.
6. Public key is $(e, N) = (23, 159197)$ and private key is $(d, N) = (6887, 159197)$.
7. Encryption with public key:

$\alpha_1 = (d_1)^e \pmod{N} = 397835^{23} \pmod{159197} = 134495$,
$\alpha_2 = (d_2)^e \pmod{N} = -1^{23} \pmod{159197} = 159196$,
$\alpha_3 = (d_3)^e \pmod{N} = 19^{23} \pmod{159197} = 80901$.
Thus, $\alpha_1 = 134495$, $\alpha_2 = 159196$ and $\alpha_3 = 80901$ are the encrypted values.
For **Decryption** use the public key $(d, N) = (6887, 159197)$.
Input the values of α_1, α_2 and α_3.
$d_1 = \alpha_1^d \pmod{N} = 134495^{6887} \pmod{159197} = 397835$,
$d_2 = \alpha_2^d \pmod{N} = 159196^{6887} \pmod{159197} = -1$,
$d_3 = \alpha_3^d \pmod{N} = 80901^{6887} \pmod{159197} = 19$.
After converting all the positive integers into binary numbers and by the reverse process, the upper triangular matrix U is received and after transpose of U, the lower triangular matrix L and $U + L = A$ is constructed.

2.4 Complexity of algorithm

The complexity for different steps in encryption process is as follows:
In Step 1, input is number of rows of the adjacency matrix, so the complexity is $O(1)$.
In Step 2, adjacency matrix of A is entered and complexity involved in this step is $O(n^2)$.
In Step 5, decimal numbers are generated from binary numbers for n number of vertices and the complexity involved $= O(n^2)$.
Total complexity involved $= O(1) + O(n^2) + O(n^2) = O(n^2)$.
The complexity for different steps in decryption process is as follows:
In Step 1, RSA algorithm is applied to generate decimal numbers and the complexity involved $= O(log2)$.
In Step 2, decimal numbers are converted to binary number and upper triangular matrix is constructed and the complexity involved $= O(n^2)$.
In Step 3, new string generated for decryption and complexity $= O(n)$.
In Step 6 and Step 7, upper triangular matrix converted to lower triangular matrix and the complexity involved $= O(n^2)$.
In Step 9, transpose of $n \times n$ matrix is required and the complexity involved $= O(n^2)$.
In Step 10, addition of two matrix is done, complexity $= O(n^2)$.
Total complexity involved $= O(log2) + O(n^2) + O(n^2) + O(n^2) + O(n^2) = O(n^2)$.
Thus, encryption and decryption mechanism can be applied to signed graph matrices and it will be more secured through the implementation of RSA algorithm (*see* **Figure** 2).

3 Application

The technique defined above constitutes the input data in the form of adjacency matrix with entries as 0, 1 and -1. Security mechanism is implemented for a network through matrices and RSA algorithm. The technique

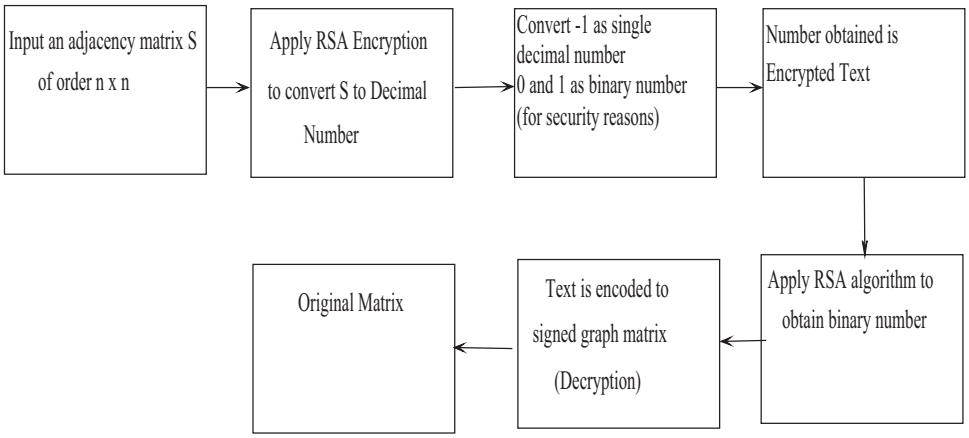

Fig. 2 Encryption and Decryption Process

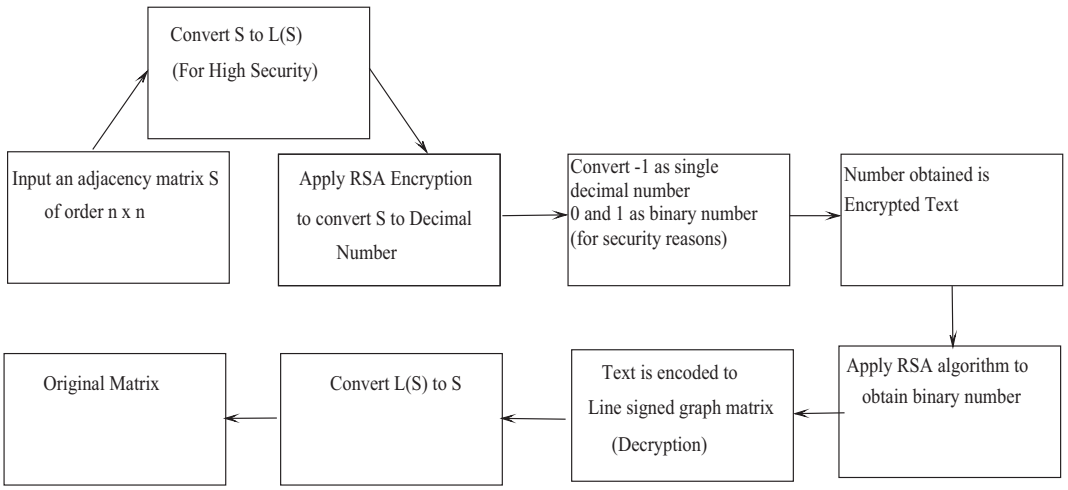

Fig. 3 Application

can further be extended to more secured networks by conversion of signed graph into line signed graph and common-edge signed graph (*see* **Figure** 3).

4 Conclusion and scope

In this paper, a special algorithm and RSA algorithm is implemented for signed graph through matrices. The technique proposed allows either of the two keys to encrypt a network [6] and the opposite key to decrypt it [1] through encryption decryption process, thus promising confidentiality, integrity, authenticity and non-reputability of data and electronic communications. Further, the approach can be extended to different types of signed graphs, [7,8] and RSA algorithm can be extended to image security through signed graphs.

Data Availability No data were used to support the findings of the study.

Declarations

Conflicts of Interest All the authors declare that they have no conflicts of interest regarding the publication of this paper.

References

1. Acharya, M., & Sinha, D. (2006). Common-edge sigraphs. AKCE International Journal of Graphs Combinatorics, 3(2), 115-130.
2. Nisha, S., & Farik, M. (2017). RSA public key cryptography algorithm - A review. International Journal of Scientific and Technology Research, 6(7), 187-191.
3. Ni, B., Qazi, R., Rehman, S., & Farid, G. (2021). Some graph-based encryption schemes. Journal of Mathematics, 2021(8), Article ID 6614172.
4. Rosen, K. H. (2005). Elementary number theory and its applications. Pearson Addison-Wesley, 5th edition.
5. Shannon, C.E. (1949). Communication theory of secrecy systems. Bell System Technical Journal, 28(4), 656-715.
6. Sinha, D., & Sethi, A. (2016). Encryption using network and matrices through signed graphs. International Journal of Computer Applications, 138(4), 6-13.
7. Sinha, D., & Sethi, A. (2014). An optimal algorithm to detect balancing in common-edge sigraph. International Journal of Computer Applications, 93(10), 19-25.
8. Sinha, D., Upadhyaya, S., & Katatria, P. (2013). Characterization of common edge signed graphs. Journal Applied Discrete Mathematics, 161, 1275-1285.
9. Stinson, D. R. (2018). Cryptography: Theory and Practice. Chapman and Hall/CRC, Boca Raton, FL, USA, 4th edition.
10. Zaslavsky, T. (2022). Glossary of signed and gain graphs and allied areas. Electronic Journal of Combinatorics, DS9, Available online: https://www.combinatorics.org/files/Surveys/ds9/ds9v1-1998.pdf.