Hindawi
Journal of Applied Mathematics
Volume 2024, Article ID 2084342, 12 pages
https://doi.org/10.1155/2024/2084342

Research Article

Graph Crypto-Stego System for Securing Graph Data Using Association Schemes

Anuradha Sabharwal,[1] **Pooja Yadav**,[2] **and Kamal Kumar**[3]

[1]*Department of Mathematics, University of Delhi, Delhi 110007, India*
[2]*Department of Mathematics, Kamala Nehru College, University of Delhi, New Delhi 110049, India*
[3]*Amity School of Applied Sciences, Amity University Haryana, Manesar, Gurugram 122051, India*

Correspondence should be addressed to Anuradha Sabharwal; anuradha.sabharwal@gmail.com and Kamal Kumar; kkumar10@ggn.amity.edu

Received 1 April 2023; Revised 29 June 2023; Accepted 12 February 2024; Published 2 March 2024

Academic Editor: Theodore E. Simos

Cryptography has recently become a critical area to research and advance in order to transmit information safely and securely among various entities, especially when the transmitted data is classified as crucial or important. This is due to the increase in the use of the Internet and other novel communication technology. Many businesses now outsource sensitive data to a third party because of the rise of cloud computing and storage. Currently, the key problem is to encrypt the data such that it may be stored on an unreliable server without sacrificing the ability to use it effectively. In this paper, we propose a graph encryption scheme by using cryptography and steganography. Data is encrypted using association schemes over finite abelian groups and then hiding the encrypted data behind randomly chosen cover image. We implemented and evaluated the efficiency of our constructions experimentally. We provide experimental results, statistical analysis, error analysis, and key analysis that demonstrates the appropriateness and efficiency of the proposed technique.

1. Introduction

Massive-scale database applications that may be represented as graph problems have recently attracted more attention to graph databases, which store, manage, and query large graphs. Large web graphs, online social networks, biological networks, RDF databases, and communication networks are a few examples of applications. As a result, numerous systems have been developed in both academia (such as [1–6]) and industry to maintain, query, and analyse huge graphs (e.g., Neo4j, Titan, DEX, and GraphBase). In paper [1], Han et al. proposed a general, disk-based graph engine called TurboGraph to process billion-scale graphs very efficiently by using modern hardware on a single PC. Kyrola and Guestrin [2] developed a new data structure, parallel adjacency lists (PAL), for efficiently managing graphs with billions of edges on disk. In paper [7], a two-layer image encryption scheme is proposed which involves bit-level encryption in the time-frequency domain. In paper [8], logistics, tent maps, and adaptive key generation modules are used to store encrypted image in cloud storage. Additionally, with the introduction of cloud computing, businesses and startups have a natural urge to outsource the storage and management of their databases to a cloud provider. Many database owners, however, are less enthusiastic about storing their databases in the cloud due to growing worries about data security and privacy.

In this paper, we propose a graph encryption method by using cryptography and steganography. We first encrypt the graph database using association schemes and then hide the encrypted data behind a cover image so that there is no suspicion to the data. With the help of cryptography, we can encrypt the original data before communicating it, rendering it unintelligible to outsiders until it is decrypted at the

receiving end. Meng et al. [9] developed graph encryption techniques that effectively allow approximative shortest-distance queries on huge encrypted graphs. By creating a number of indexes to store the data required to respond to queries, Liu et al. [10] investigated a graph encryption approach for a crucial graph query type known as top-k Nearest Keyword (kNK) searches. For graph data that supports all-path searches, Xu et al. [11] presented a searchable symmetric encryption system. Zhang et al. [12] presented the Privacy-preserving Graph encryption for Accurate constrained Shortest distance queries (PGAS) that is capable of delivering accurate constrained shortest distance (CSD) queries and ensuring the privacy of the graph data. Wang et al. [13] developed a Secure Graph DataBase (SecGDB) encryption system to encrypt graph structures and enforce private graph queries across the encrypted graph database and to support the shortest distance queries with the least amount of time and storage complexity, and their design specifically makes strategic use of effective additively homomorphic encryption and jumbled circuits.

In cryptography, no matter how impossible they are to crack, plainly visible encrypted messages draw interest. In order to prevent an intruder from discovering the secret data, we can use steganography to hide it in a multimedia file. Different file types can be deployed, but because digital photographs are so common on the Internet, they are preferable. Text, video, image, and even audio data can all be concealed with steganography. In past, cryptography and steganography have been applied to many two-dimensional data files such as images, videos, audio, text, and pdf [14–17]. Till now, only cryptographic methods are applied to graph data. In this paper, we use steganography to hide the encrypted graph behind a cover image. Enhanced imperceptibility is achieved by concealing the original information in the cover image's margins [18, 19].

2. Preliminaries and Notations

2.1. Association Scheme. The study of algebraic combinatorics relies on association schemes (AS), which Bose and Shimamoto established in 1952 [20]. It has significance in design theory, group theory, coding theory, and graph theory. These days, information security is a top priority. Therefore, we study the implications of association schemes in cryptography. For a finite set A, let the partition of $A \times A$ be denoted by $\mathfrak{P}$ and let $\mathbb{Z}_p^r = \underbrace{\mathbb{Z}_p \times \mathbb{Z}_p \times \cdots \times \mathbb{Z}_p}_{r \text{ times}}$. Below is some basic fundamental literature to association scheme.

Definition 1. Let $\mathfrak{P}$ be a partition of $A \times A$, where A is a finite set, and let $T_0, T_1, \cdots, T_n$ be binary relations on $\mathfrak{P}$. Then, $\mathcal{S} = (A, \mathfrak{P})$ forms n-class association scheme if the subsequent conditions hold:

(1) Identity relation: $T_0 = \{(x, y): x, y \in A\} \in \mathfrak{P}$

(2) $T^* = \{(x, y) | (y, x) \in T\} \in \mathfrak{P}$ for any relation $T \in \mathfrak{P}$

(3) If $(x, y) \in T_j$, then $|z \in A: (x, z) \in T_h$ and $(z, y) \in T_i| = p_{hi}^j$, where p_{hi}^j is a constant independent of x and y for all integers $0 \le h, i, j \le n$

$\mathcal{S}$ is called symmetric association scheme, if $T = T^*$ for every relation T in $\mathfrak{P}$ and commutative association scheme, if $p_{hi}^j = p_{ih}^j \forall 0 \le h, i, j \le n$. Here, $\{p_{hi}^j\}_{0 \le h,i,j \le n}$ are called *intersection numbers* or *parameters* of $\mathcal{S}$. Let the set $xT = \{y \in A | (x, y) \in T\}$ for $x \in A$ and $T \in \mathfrak{P}$. The elements x and y in A are called *j th associates* if $(x, y) \in T_j$ with $x \ne y$. It can be easily proved that every symmetric association scheme is commutative. We cite [21] for results of more elementary association schemes.

Sabharwal et al. [22, 23] have constructed nonsymmetric commutative AS for symmetric groups, dihedral groups, and several abelian groups. In [22], nonsymmetric AS has also been constructed for the group $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_n}$ in particular.

Theorem 1. *Let $A = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_n}$, where $m_1 \le m_2 \le \cdots \le m_n$ and $\gcd(m_1, m_2, \cdots, m_n) \ne 1$. For each $d_j \in \mathbb{Z}_{m_j} \forall 1 \le j \le n$, the relations T_j in $\mathfrak{P}$ defined by*

$$\begin{aligned} T_j = \{(x, y) | \, & y_n \equiv (j + x_n) \bmod m_n, y_r \\ & \equiv (j + d_r + x_r) \bmod m_r \forall 1 \le r \\ & \le n - 1 | x = (x_1, x_2, \cdots, x_n), y = (y_1, y_2, \cdots, y_n) \in A\} \end{aligned} \tag{1}$$

form a nonsymmetric commutative AS, and its intersection numbers are given by

$$p_{hi}^j = \begin{cases} 1 & \text{if } j \equiv (h + i) \bmod m_n, \text{and}, \\ & j + d_r^{(j)} \equiv \left(h + i + d_r^{(h)} + d_r^{(i)}\right) \bmod m_r \forall 1 \le r \le n - 1, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where

$$h = \sum_{k=1}^{n} \left(\prod_{l=k+1}^{n} m_l \right) d_k^{(h)}; i = \sum_{k=1}^{n} \left(\prod_{l=k+1}^{n} m_l \right) d_k^{(i)}; j = \sum_{k=1}^{n} \left(\prod_{l=k+1}^{n} m_l \right) d_k^{(j)}. \tag{3}$$

Further, $T_j^* = T_J$ can be determined by solving the following equations:

$$\begin{aligned} J + j &\equiv 0 \bmod m_n, \\ J + j + D_r + d_r &\equiv 0 \bmod m_{r,} \quad \forall 1 \le r \le n - 1, \end{aligned} \tag{4}$$

where $j = \sum_{k=1}^{n} (\prod_{l=k+1}^{n} m_l) d_k^{(j)}$ and $J = \sum_{k=1}^{n} (\prod_{l=k+1}^{n} m_l) D_k^{(J)}$.

Proof. Refer [22]. □

3. Proposed Approach

In the proposed method, we first use association schemes over $\mathbb{Z}_m^n$ to encrypt graph data and then use LSB method to hide the graph data behind a cover image. In Theorem 1, each relation T_j is considered as a function from $\mathbb{Z}_m^n \longrightarrow \mathbb{Z}_m^n$. Thus, the association scheme on $\mathbb{Z}_m^n$ (as defined in the previous theorem) represents Vigenère cipher (VC).

3.1. Graph Encryption/Decryption. In our graph encryption technique, a graph is represented by the notation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denote the set of vertices and $\mathcal{E}$ denote the set of edges. Here, we consider both directed and undirected graphs. In an undirected graph, each edge is formed by two vertices, represented by $E = (U, V)$, where $E \in \mathcal{E}$ and $U, V \in \mathcal{V}$, and in a directed graph, the edges have orientations and each edge is represented as an ordered pair of vertices.

The graph data with r nodes is first converted in adjacency matrix $A(\mathcal{G})$ of order $r \times r$. Let k be the encryption key and $n = r^2$. Then, we apply AS over $\mathbb{Z}_r^n$ on $A(\mathcal{G})$ (as defined in Theorem 1). That is, kth relation, T_k is applied on $A(\mathcal{G})$ to obtain encrypted graph data B. Further, this encrypted data is written in binary form and is merged with a cover image by storing them at the LSB of the cover image, which results in a stego image as shown in Figure 1. This stego image is then ready for secure transmission without any suspicion of the data. The steps for producing a graph stego image using this method are covered in Algorithm 1.

Similar strategies have been adopted to recover the data from the graph stego image that is provided in Algorithm 2. For decryption, we first determine decryption key K from k (by solving sets of equations in Theorem 1). Then, we apply AS over $\mathbb{Z}_r^n$ on the encrypted graph data to obtain matrix D which is the same as adjacency matrix of original graph data as shown in Figure 2.

Encryption of data alone does not provide appropriate security. Applying steganography will hide the secret information in a multimedia file, allowing unnoticed secure transmission of the secret data in an insecure network.

4. Experimental Results

By statistically analysing the suggested approach, this part investigates the numerical simulations and analytical results for secure graph data transmission. A laptop with the features 12th Gen Intel(R) Core (TM) i7-1255U CPU at 1.70 GHz is used to execute this technique. MATLAB R2021b is used for this purpose. We have selected two graphs: a directed graph with 5 nodes (Figure 3(a)) and a complete graph K5 (Figure 4(a)) and two cover images (Figures 3(b) and 4(b)). Using an association scheme, the secret graph data is first encrypted (Figures 3(c) and 4(c)). After concealing this encrypted data in the LSB of the cover image, we are left with the graph stego image as displayed in Figures 3(d) and 4(d). Figures 3(e) and 4(e) show the extracted image. We note that there was no distortion in the retrieved data (see Figures 3(f) and 4(f)). It can be seen

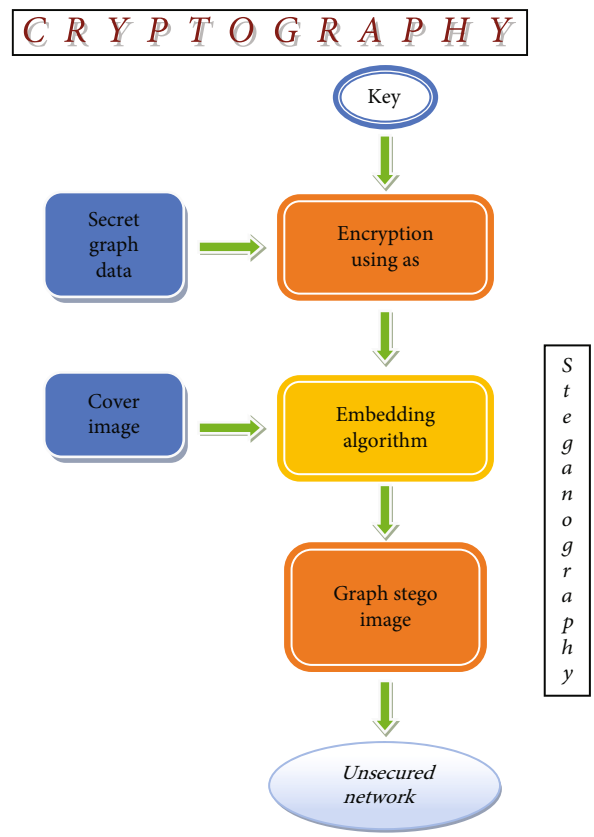


Figure 1: Security model for constructing graph stego image.

that there is an accurate recovery of the original graph data without any loss or error of information. In order to assess the resilience of the cryptosystem, the examination of cover image and graph stego image has been described in terms of their change rate of number of pixels (NPCR), unified average changing intensity (UACI), mean square error (MSE), peak noise-to-signal ratio (PNSR), correlation coefficient (CC), histogram analysis, and pixel intensity correlation graphs.

4.1. Histogram Analysis. The histogram is a graphical representation of the frequency and pixel intensity value of an image. The cryptosystem's robustness depends on the histogram. In a pixel intensity correlation graph of an image, the x-axis represents the pixel values (ranging from 0 to 255) and the y-axis represents the corresponding number of pixels in the image. This graph gives an overall idea about the intensity distribution of an image. The histograms and pixel intensity correlation graphs for the RGB components of the cover image and the graph stego image are given in Figures 5–8, respectively, from which we can deduce that the histogram of the image which is used to hide the data is similar to the histogram of the image in which data is hidden. This indicates that no one can predict that any information is encoded in the graph stego image.

4.2. Error Analysis. The analysis of the cover image and the graph stego image, along with its NPCR, UACI, MSE, PNSR,

1: **Input:** Secret Graph data(G), Cover Image(CI), AS key(k)
2: **Output:** Graph Stego Image
3: **procedure** Encryption
1. Write the adjacency matrix $A(\mathscr{G})$ of the given graph data which is of order r.
2. Find $t_1, t_2, \cdots, t_r$ such that $k = \sum_{i=1}^{r}(\prod_{j=i+1}^{r} r)t_i$
3. Apply relation T_k of AS over $\mathbb{Z}_r^{r^2}$ (as defined in Theorem 1) on $A(\mathscr{G})$ to obtain the cipher graph data B.
4. Store the number of nodes at the first 8 bits of CI.
5. Store the bits of the encrypted data B at the remaining LSB of the RGB components of the CI.
4: **end procedure**

Algorithm 1: Algorithm for encryption procedure to construct graph stego image.

1: **Input:** Graph Stego Image(GSI), AS key(k)
2: **Output:** Secret Data retrieved
3: **procedure** Decryption
1. Obtain the hidden data(D) and number of nodes(n) from RGB components of the GSI.
2. Find decryption key K as discussed in Theorem 1 from key k.
3. Find $d_1, d_2, \cdots, d_r$ such that $K = \sum_{i=1}^{r}(\prod_{j=i+1}^{r} r)d_i$.
4. Apply relation T_K of AS over $\mathbb{Z}_r^{r^2}$ (as defined in Theorem 1) on D to obtain decrypted graph data.
4: end procedure

Algorithm 2: Algorithm for extracting secret data from graph stego image.

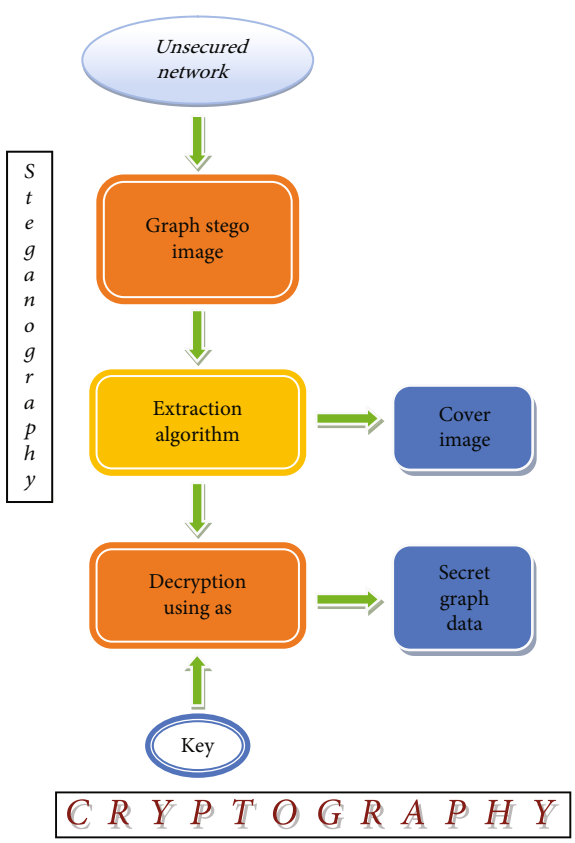


Figure 2: Security model for reconstructing secret graph data.

and CC values, is covered in this part. These values are listed in Tables 1 and 2.

4.2.1. Mean Square Error. According to the following formula, the MSE, a measure of peak error for the RGB components of two images, is calculated as follows:

$$\mathrm{MSE} = \frac{1}{h \times w}\sum_{i=1}^{h}\sum_{j=1}^{w}\left[\left|f(i\Delta x, j\Delta y) - f_0(i\Delta x, j\Delta y)\right|^2\right], \quad (5)$$

where h and w are the pixels and Δx and Δy are the pixel sizes of the image.

The values of MSE depict that the graph stego image with embedded encrypted graph data resembles with the cover image.

4.2.2. Peak Noise-to-Signal Ratio. According to the following formula, the PNSR, which reflects a measure of peak error in both the images, is calculated as follows:

$$\mathrm{PNSR} = 10\log_{10}\left(\frac{\mathrm{MPV}_I^2}{\mathrm{MSE}}\right), \quad (6)$$

where MPV_I is the maximum possible value of pixels of the image.

The high PNSR values of graph stego image and cover image as shown represent the similarity of these images.

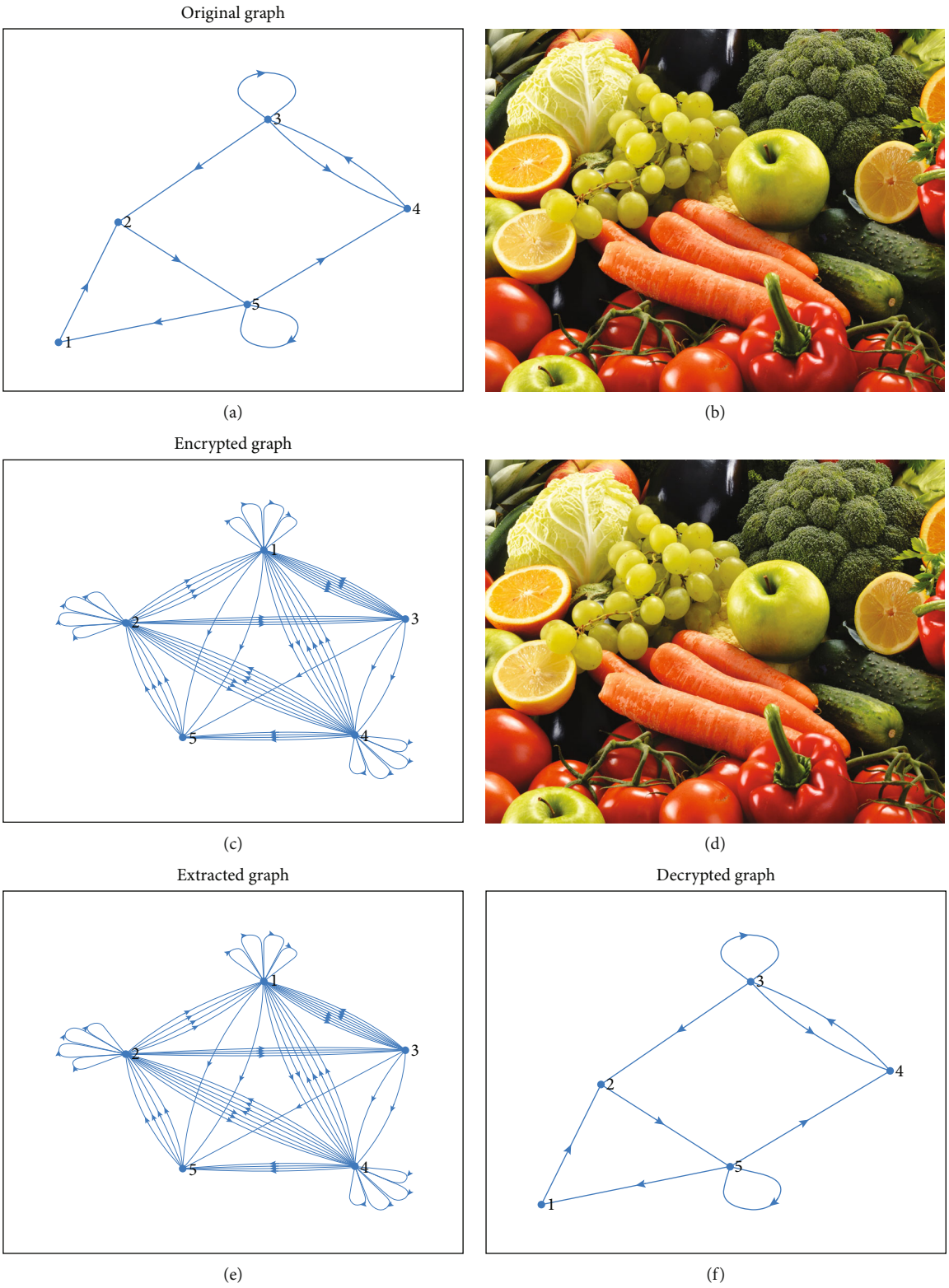


Figure 3: (a) Original directed graph with 5 nodes. (b) Cover image. (c) Encrypted graph. (d) Graph stego image. (e) Extracted image. (f) Correctly decrypted image.

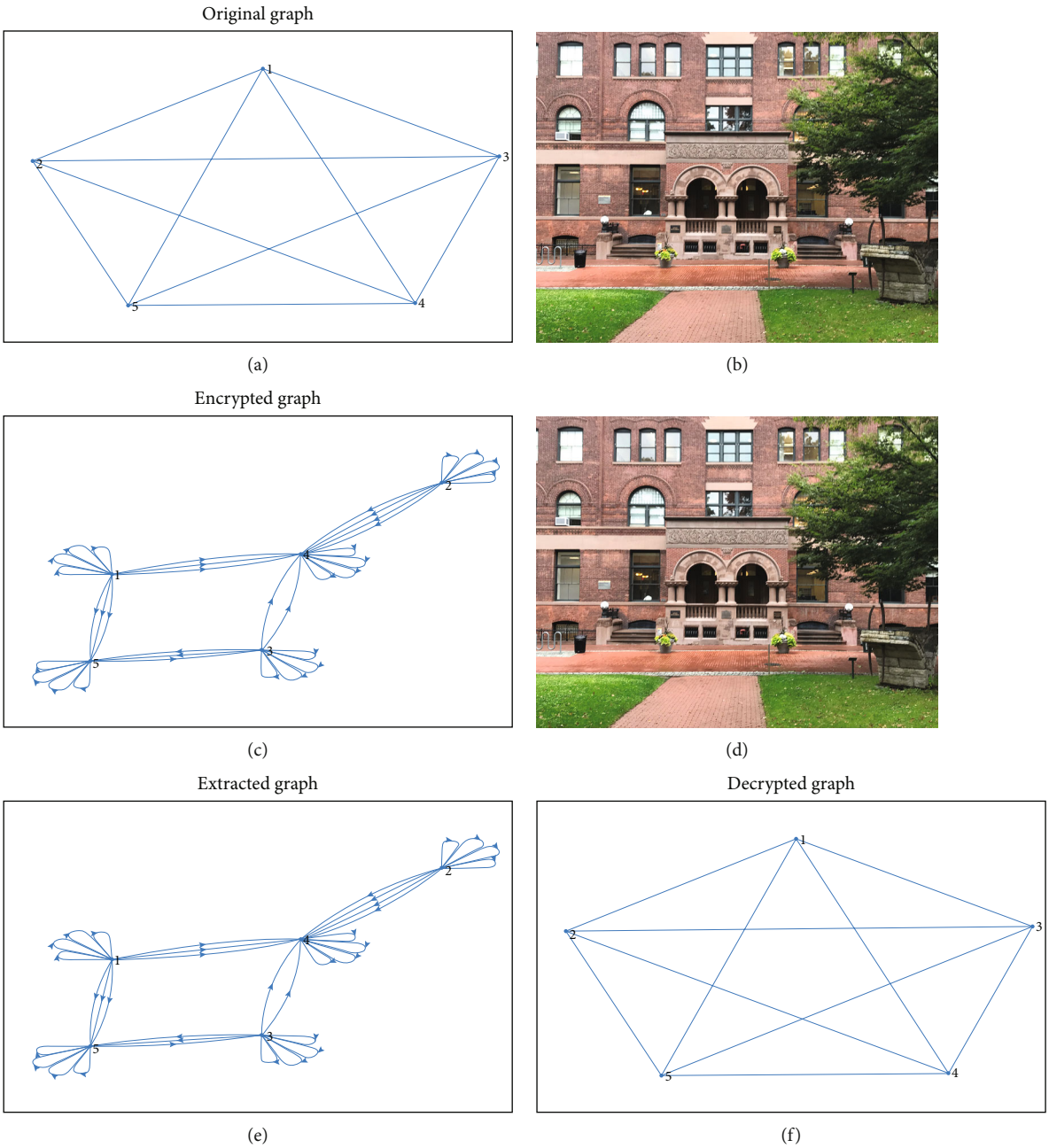


FIGURE 4: (a) Original complete graph K5. (b) Cover image. (c) Encrypted graph. (d) Graph stego image. (e) Extracted image. (f) Correctly decrypted image.

4.2.3. Correlation Coefficient. The following formula is used to determine the CC between cover image and graph stego image.

$$\mathrm{CC} = \frac{\sum_m \sum_n (x_{mn} - x)(y_{mn} - y)}{\sqrt{[\sum_m \sum_n (x_{mn} - x)]^2 [\sum_m \sum_n (y_{mn} - y)]^2}}, \quad (7)$$

where x and y are mean of the two images.

CC can range from −1 to +1 between two images. When the value of CC is +1, there is a strong positive linear correlation between these images, and when it is −1, there is a negative correlation. If the value is near to 0, it means there is no connection between these images.

4.2.4. Number of Pixel Change Rate. The NPCR counts the variations in pixel counts between two images. If the cover image and the graph stego image are denoted by $I_o(i, j)$ and $I_r(i, j)$, respectively, then the following formula can be used to determine NPCR between the two images:

$$\mathrm{NPCR} = \frac{\sum_i \sum_j d(i, j)}{h \times w} \times 100\%, \quad (8)$$

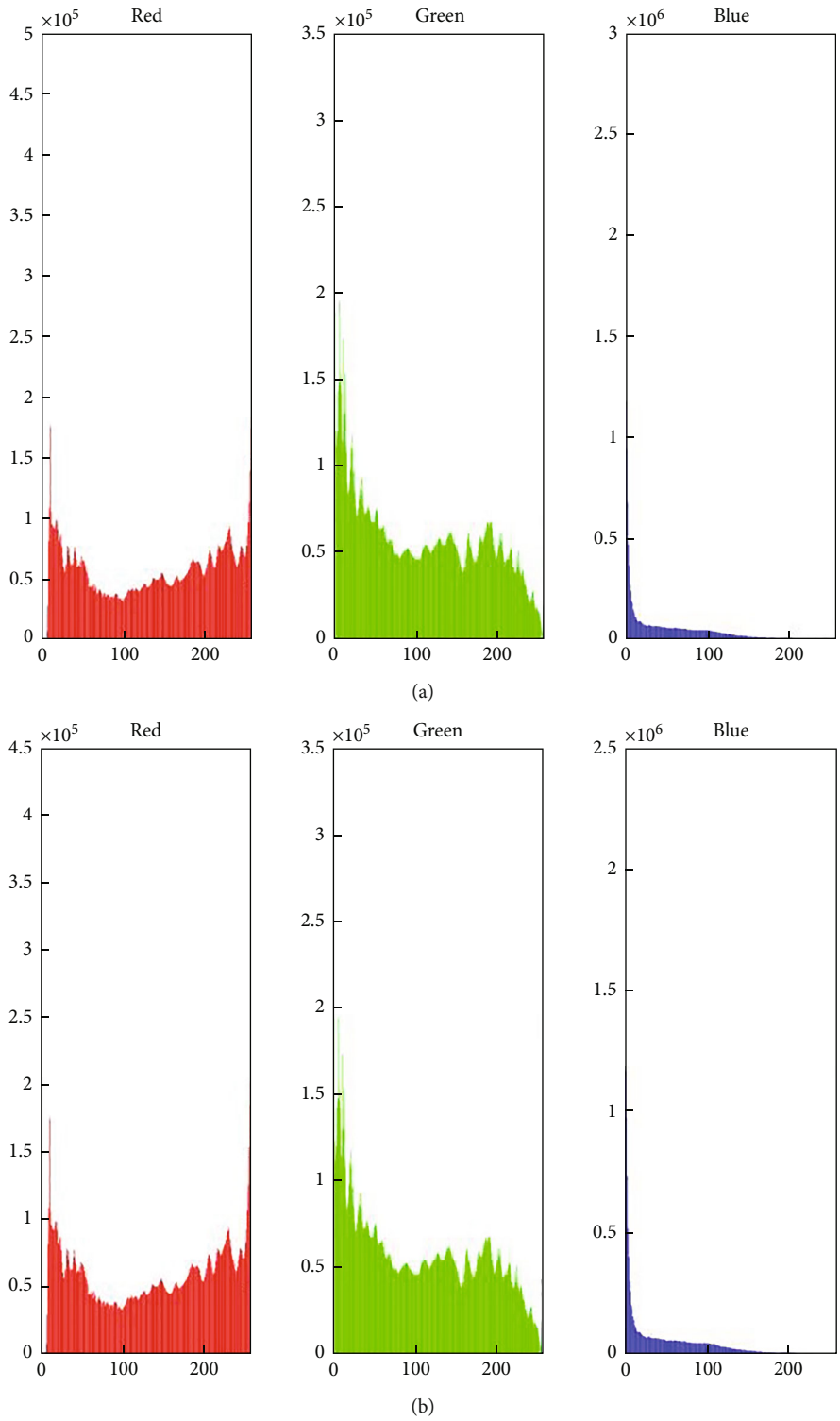


Figure 5: Histogram for Figure 3: (a) cover image; (b) graph stego image.

where

$$d(i,j) = \begin{cases} 1 & \text{if } I_o(i,j) \neq I_r(i,j), \\ 0 & \text{if } I_o(i,j) = I_r(i,j). \end{cases} \quad (9)$$

4.2.5. Unified Average Changing Intensity. The UACI value measures the average of variations in pixel intensity values between two images. The following is the formula for computing UACI between graph stego image and cover image:

$$\text{UACI} = \frac{1}{h \times w} \sum_i \sum_j \frac{|I_o(i,j) - I_r(i,j)|}{255} \times 100\%. \quad (10)$$

4.3. Key Space Analysis. Key space is a term used in cryptography to describe the largest key that can be used; therefore, key

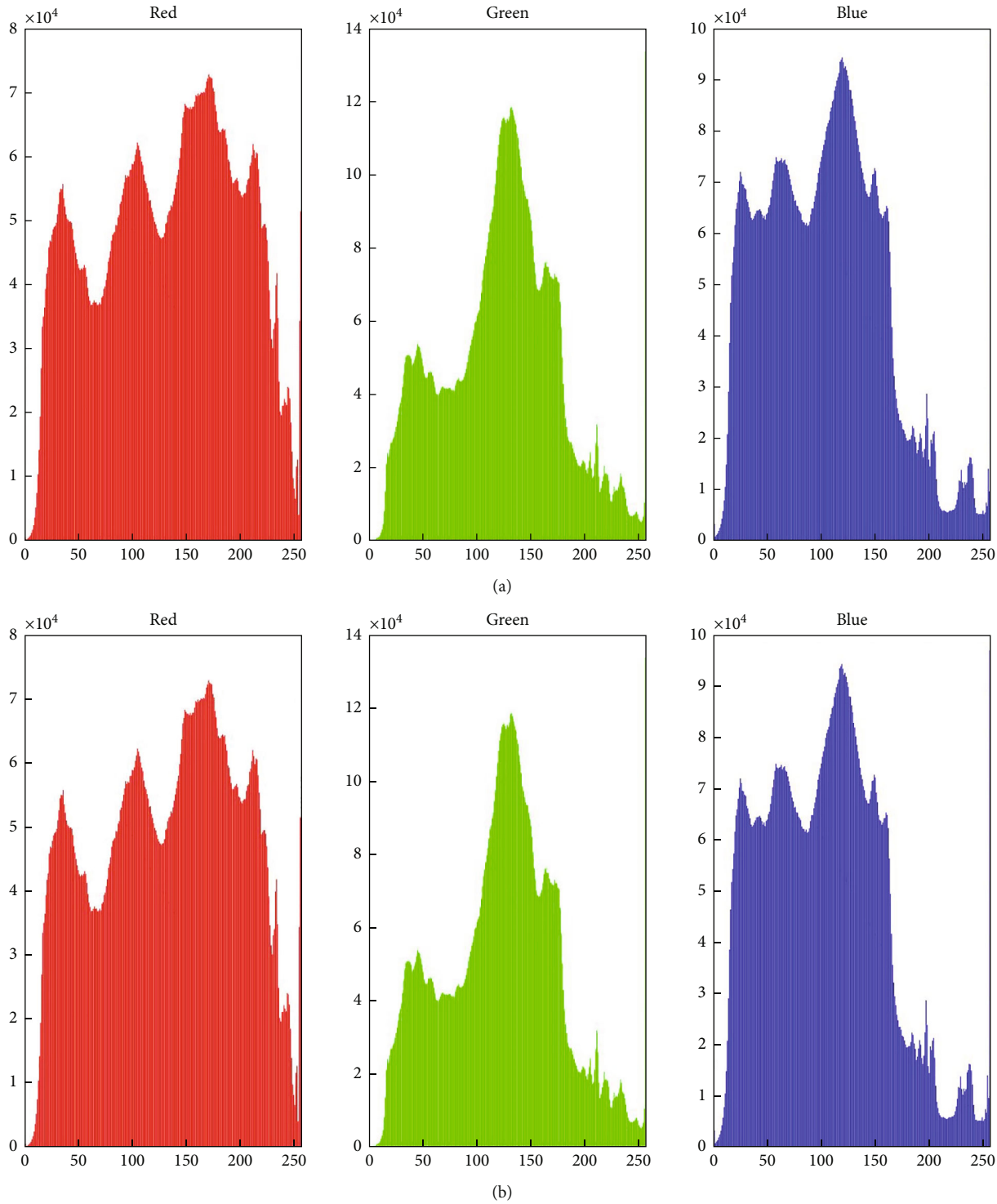


Figure 6: Histogram for Figure 4: (a) cover image; (b) graph stego image.

space is a function of key length. Individual analysis was conducted in this work to confirm the robustness of the cryptosystem in the event of an attack. Data is encrypted using an encryption key in cryptography over AS. The encryption key cannot be used directly on the original data, even if the attacker knows it. The proposed approach is used to calculate the subkey from the encryption key.

4.3.1. Key Sensitivity Analysis. The key sensitivity is one of the most important elements that determines a security system's robustness. Greater cryptosystem security is ensured by high key sensitivity, which prevents sensitive data from being taken without the precise key. In the suggested method, if any mismatch is found during the decryption stage, the original data cannot be restored. Even if the

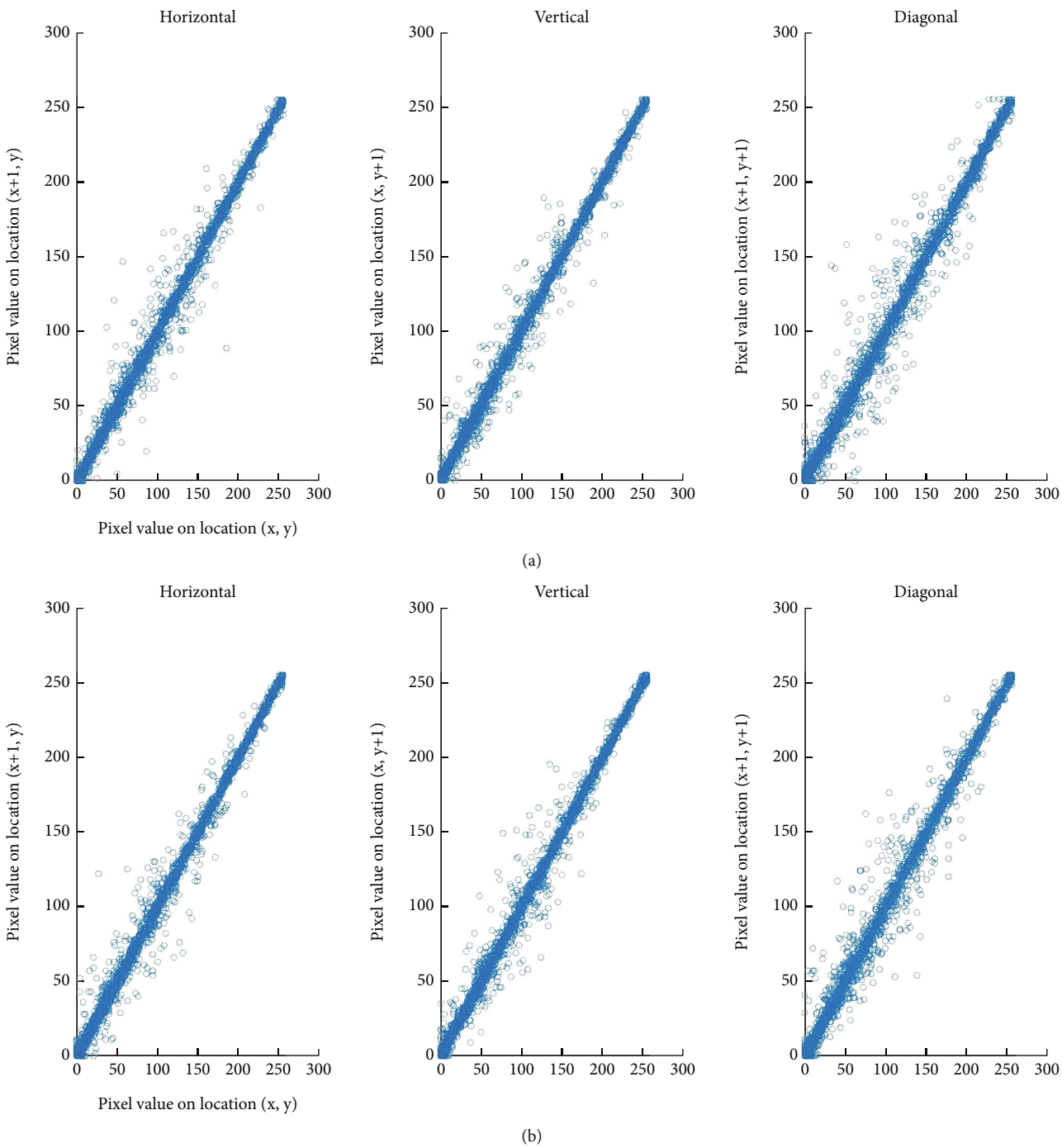
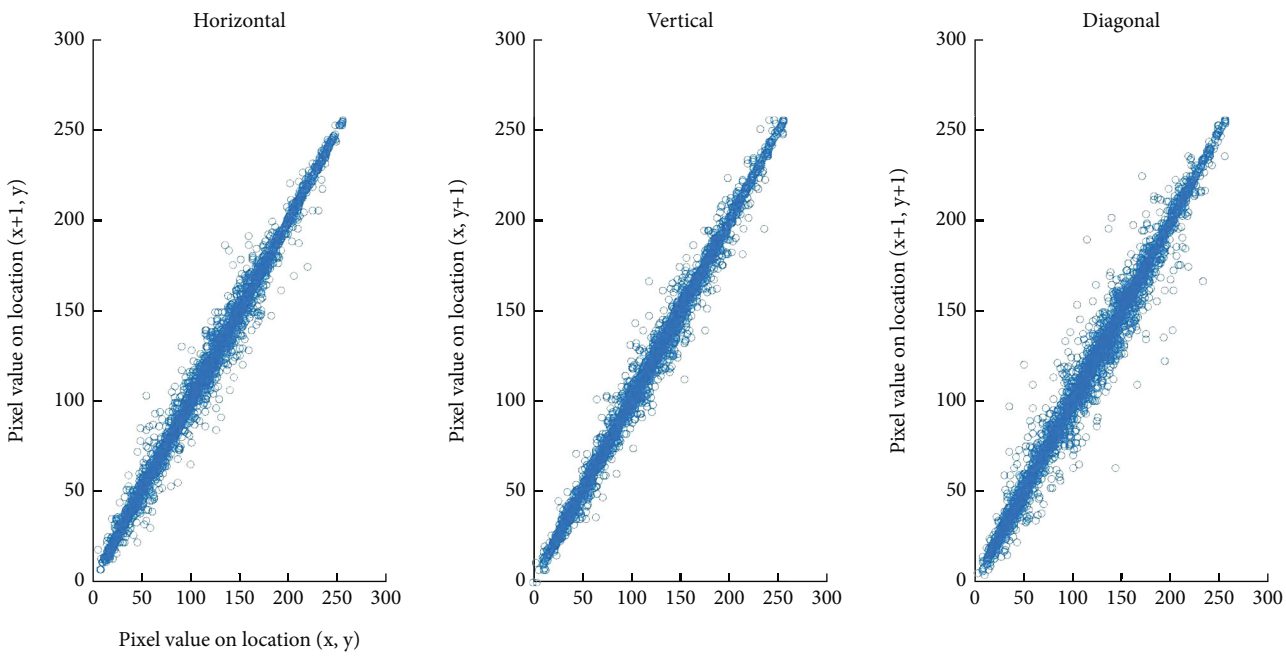


Figure 7: Pixel intensity for Figure 3: (a) cover image; (b) graph stego image.

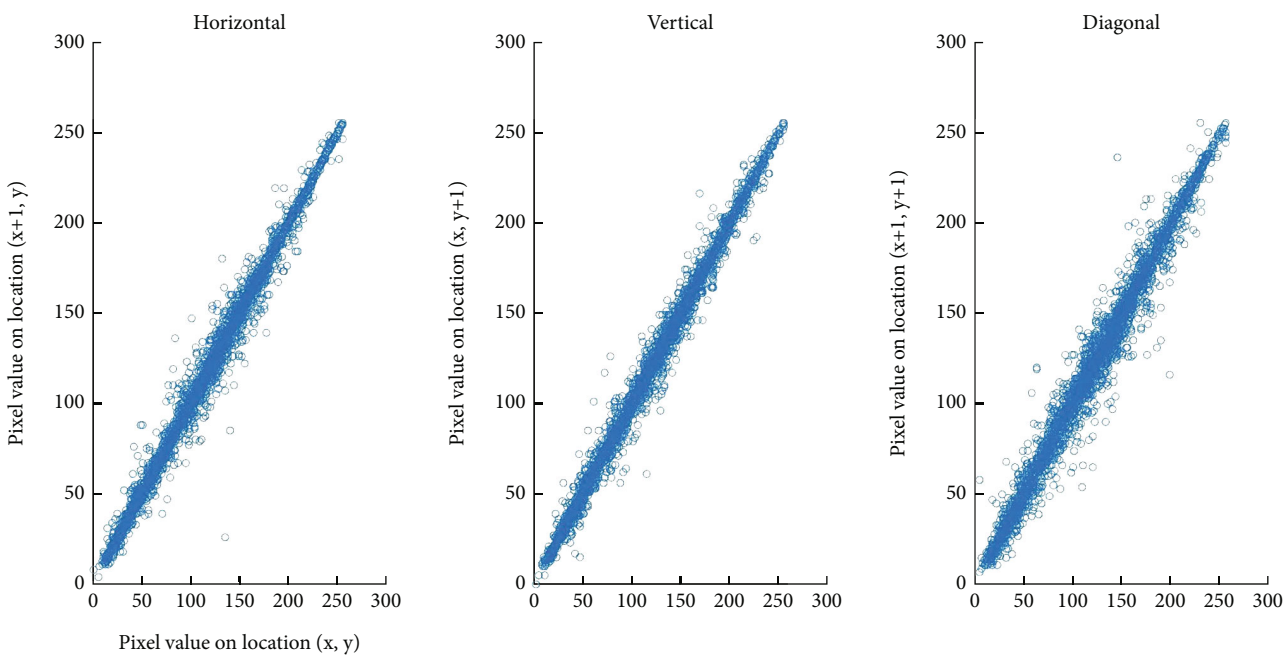
decryption key slightly alters, the output will be corrupted and damaged. When the key used for decryption and encryption is identical, the original data can be retrieved. Communication requirements often increase because the ciphertexts in these ciphers are substantially larger than the plaintexts.

5. Comparison

The methods described in [10–13, 24] mostly employ cryptographic techniques to render security. Liu et al. [10] developed a graph encryption strategy for an important graph query type known as top-k Nearest Keyword (kNK) searches by building a number of indexes to store the data needed to answer to queries. When performing kNK queries on an encrypted graph, privacy information would be leaked from both the graph and the queries. Xie and Eric introduced CryptGraph in [24], which uses ring-based homomorphic encryption techniques to encrypt graphs. Typically, homomorphic ciphers do not offer verified computing on their own. Communication requirements often increase because



(a)



(b)

Figure 8: Pixel intensity for Figure 4: (a) cover image; (b) graph stego image.

Table 1: Statistical analysis of cover image and graph stego image for Figure 3.

Component	NPCR	UACI	MSE	PNSR	CC
Red	9.3106	0.0479	0.1899	55.3447	1.0000
Green	13.5636	0.0563	0.1611	56.0603	1.0000
Blue	18.9633	0.1341	0.7612	49.3159	0.9999

Table 2: Statistical analysis of cover image and graph stego image for Figure 4.

Component	NPCR	UACI	MSE	PNSR	CC
Red	0.4921	0.1930	0.4921	101.2103	1.0000
Green	0	0	0	∞	1.0000
Blue	0	0	0	∞	1.0000

the ciphertexts in these ciphers are substantially larger than the plaintexts. The proposed graph crypto-stego methodology makes use of both cryptographic and steganographic methods to secure graph data. This technique uses association schemes over finite abelian group for encryption which is a new technique and has not been employed in graph encryption before. According to Tables 1 and 2, the embedding rate of the suggested technique is 0.99 using the PSNR value, and the data retrieval is lossless.

6. Conclusion

In this article, we present an association scheme-based cryptography method along with steganography for the secure transfer of graph data. Till now, only cryptographic methods are applied to graph data. Here, we use steganography to hide the encrypted graph behind a cover image. This method enables the safe and secure delivery of critical information over an insecure network. The bits of the secret data are preserved at the LSB of the cover image after being initially encrypted with a private key and an association scheme over an abelian group. The receiver uses the association scheme with a decryption key that is derived from the encryption key in order to recover the original information. Since the extracted data and the original data are same, there has apparently been no information loss. We have presented experimental data, statistical analysis, and error analysis for directed graph and complete graph that demonstrate the viability of our method for delivering data securely over any open network. This method can be used to secure any type of graphs and is likely to be crucial in developing new steganography and cryptography methods. Finding more uses of association schemes to conceal two-dimensional data in the form of video and audio in another multimedia file would be a worthwhile research endeavour.

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W.-S. Han, S. Lee, K. Park et al., "TurboGraph: a fast parallel graph engine handling billion-scale graphs in a single PC," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 2013.

[2] A. Kyrola and C. Guestrin, "GraphChi-DB: simple design for a scalable graph database system–on just a PC," 2014, https://arxiv.org/abs/1403.0701.

[3] Y. Low, J. E. Gonzalez, A. Kyrola, D. Bickson, C. E. Guestrin, and J. Hellerstein, "Graphlab: a new framework for parallel machine learning," 2014, https://arxiv.org/abs/1408.2041.

[4] G. Malewicz, M. H. Austern, A. J. C. Bik et al., "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, New York, 2010.

[5] M. Sarwat, S. Elnikety, Y. He, and G. Kliot, "Horton: online query execution engine for large distributed graphs," in *2012 IEEE 28th International Conference on Data Engineering*, Arlington, VA, USA, 2012.

[6] B. Shao, H. Wang, and Y. Li, "Trinity: a distributed graph engine on a memory cloud," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, New York, 2013.

[7] H. Mahalingam, T. Veeramalai, A. R. Menon, and R. Amirtharajan, "Dual-domain image encryption in unsecure medium—a secure communication perspective," *Mathematics*, vol. 11, no. 2, p. 457, 2023.

[8] H. Mahalingam, P. V. Meikandan, K. Thenmozhi et al., "Neural attractor-based adaptive key generator with DNA-coded security and privacy framework for multimedia data in cloud environments," *Mathematics*, vol. 11, no. 8, 2023.

[9] X. Meng, S. Kamara, K. Nissim, and G. Kollios, "Grecs: graph encryption for approximate shortest distance queries," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York, 2015.

[10] C. Liu, L. Zhu, and J. Chen, "Graph encryption for top-k nearest keyword search queries on cloud," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 371–381, 2017.

[11] Z. Xu, F. Zhou, J. Li, Y. Li, and Q. Wang, "Graph encryption for all-path queries," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 16, article e5362, 2020.

[12] C. Zhang, L. Zhu, C. Xu, K. Sharif, C. Zhang, and X. Liu, "PGAS: privacy-preserving graph encryption for accurate constrained shortest distance queries," *Information Sciences*, vol. 506, pp. 325–345, 2020.

[13] Q. Wang, K. Ren, M. Du, Q. Li, and A. Mohaisen, "SecGDB: graph encryption for exact shortest distance queries with efficient updates," in *International Conference on Financial Cryptography and Data Security*, New York, 2017.

[14] Z. Liu, L. Xu, T. Liu et al., "Color image encryption by using Arnold transform and color-blend operation in discrete cosine transform domains," *Optics Communications*, vol. 284, no. 1, pp. 123–128, 2011.

[15] D. C. Mishra, R. K. Sharma, M. Kumar, and K. Kumar, "Security of color image data designed by public-key cryptosystem associated with 2D-DWT," *Fractals*, vol. 22, no. 4, article 1450011, 2014.

[16] H. Sharma, D. C. Mishra, R. K. Sharma, and N. Kumar, "Multi-image steganography and authentication using crypto-stego techniques," *Multimedia Tools and Applications*, vol. 80, no. 19, pp. 29067–29093, 2021.

[17] H. Sharma, D. C. Mishra, R. K. Sharma, and N. Kumar, "Crypto-stego system for securing text and image data," *International Journal of Image and Graphics*, vol. 18, no. 4, article 1850020, 2018.

[18] J. Bai, C.-C. Chang, T.-S. Nguyen, C. Zhu, and Y. Liu, "A high payload steganographic algorithm based on edge detection," *Displays*, vol. 46, pp. 42–51, 2017.

[19] S. Sun, "A novel edge based image steganography with 2 k correction and Huffman encoding," *Information Processing Letters*, vol. 116, no. 2, pp. 93–99, 2016.

[20] R. C. Bose and T. Shimamoto, "Classification and analysis of partially balanced incomplete block designs with two associate

classes," *Journal of the American Statistical Association*, vol. 47, no. 258, pp. 151–184, 1952.

[21] P.-H. Zieschang, *An algebraic approach to association schemes*, Springer, 2006.

[22] A. Sabharwal, R. K. Sharma, and P. Yadav, *AS and Its Applications in Cryptography*, Association Schemes in Steganography, 2022.

[23] A. Sabharwal and P. Yadav, "Association schemes over some finite group rings," in *Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy*, pp. 13–23, Springer, 2022.

[24] P. Xie and X. Eric, "Cryptgraph: Privacy preserving graph analytics on encrypted graph," 2014, https://arxiv.org/abs/1409.5021.