Multimedia Tools and Applications (2019) 78:20609–20636
https://doi.org/10.1007/s11042-019-7378-x

An efficient scheme for secure domain medical image fusion over cloud

Lakshmi V. S.[1] · **Deepthi P. P.**[1]

Received: 1 July 2018 / Revised: 12 December 2018 / Accepted: 15 February 2019 /
Published online: 4 March 2019

Abstract

The exponential growth in the medical images is making the healthcare industry move towards cloud-based paradigm, which has vast storage and high end processing facilities. However, moving medical images containing highly sensitive data to third-party cloud servers brings in serious security threats. Even though encrypting medical images before outsourcing using traditional encryption schemes seem to be a feasible solution, that can not support encrypted domain processing. In this paper, we propose an affine Hill cipher based scheme for encrypted domain medical image fusion. The random vectors used in this scheme are carefully designed to preserve the randomness and security properties when operations are performed on the encrypted data. The proposed scheme offers data privacy and supports encrypted domain processing with no additional storage burden at the cloud side and very low computational burden at the healthcare provider side. The security of the proposed scheme is evaluated through extensive cryptanalysis in terms of resistance against various statistical attacks. The performance of the proposed scheme is analyzed by comparing various metrics of encrypted domain MR-CT/PET image fusion results with those of plaintext domain fusion. The values of structural similarity index, normalized correlation coefficient and structural content are 1 and the image quality index is 0.999, which show that the proposed encrypted domain image fusion provides same accuracy levels as that of plaintext domain image fusion.

Keywords Data privacy · Cloud storage · Hill cipher · Homomorphic encryption · Image fusion

✉ Lakshmi V. S.
lakshmivs23@gmail.com

Deepthi P. P.
deepthi@nitc.ac.in

[1] Department of Electronics and Communication Engineering, National Institute of Technology Calicut, Kozhikode, India

1 Introduction

Medical image fusion plays a crucial role in accurate diagnosis and treatment of diseases. Medical images of various modalities can provide different information. For example, computed tomography (CT) image give information on dense structures such as bones; and positron emission tomography (PET) image provide functional information while magnetic resonance (MR) imaging provide information on soft tissues. Image fusion helps the doctor for better diagnosis by combining the complimentary features in images of different modalities. As medical images of different modalities acquired by different devices are of high resolution for precise diagnosis, it impose huge storage cost to the health care providers. In addition, due to the exponential growth in medical images generated per day, the biggest challenge faced by the health care providers/industry is the storage, processing and management of this huge amount of medical images being generated.

In recent years, with the advent of cloud platform offering high end storage and computing facilities, health care industry is now shifting the medical data to the cloud environment [28]. Although cloud storage and computing relieves the healthcare providers from the burden of data storage, maintenance and processing cost, it also brings in several security threats since the outsourced data is stored in third party cloud servers. One of the major security challenges that need to be addressed is to protect privacy or confidentiality of data, when highly sensitive data like medical images are outsourced, especially when processing operations are to be performed on these data. Although traditional encryption schemes like AES provide data confidentiality, they will not support operations on the encrypted medical images.

Homomorphic encryption schemes are a special class of encryption schemes, which allow to perform some operations on the encrypted data without any knowledge of the decryption function. Image fusion techniques involve only linear operations and an additive homomorphic encryption scheme is sufficient for encrypting the medical images to be outsourced. The most popular additive homomorphic encryption schemes available in literature are Paillier [23] and secret sharing schemes [30]. Paillier scheme is computationally expensive due to the modular exponentiation operation over a finite field of large size. Moreover, the storage overhead of Paillier scheme is twice since the size of ciphertext is double that of plaintext. Secret sharing scheme (SSS) has lesser computational complexity compared to the Paillier scheme due to linear encryption and decryption operations. But in SSS [22, 34], different image shares generated from a single image need to be stored in storage devices of multiple non-colluding cloud service providers in order to provide adequate security. This brings in additional restrictions on storage. Furthermore, since size of each image share is same as the size of original image, the storage overhead increases with the number of image shares being stored.

As the encryption and decryption operations have to be done at the health service provider (client) side, it is desirable to have an additive homomorphic encryption scheme with low computational complexity. Moreover, since data outsourcing cost charged by the cloud depends on the amount of data stored in the cloud, it is required to minimize the storage overhead incurred due to the ciphertext expansion. This motivated us to design a low complex additive homomorphic encryption system based on Hill cipher [35] since it satisfies the requirements in terms of computational complexity and storage overhead. This is due to the fact that the encryption and decryption operations of Hill cipher are linear and as the size of ciphertext is same as that of plaintext, there is no storage overhead. However, original Hill cipher scheme is prone to known-plaintext attack. Hence, in order to use

Hill cipher for secure image fusion, it is necessary to modify it in such a way that, it is computationally infeasible for an adversary, who has access to all data stored in the cloud and some knowledge about the plaintext, to retrieve the plaintext. But since such an adversary cannot have access to encrypting machine, chosen plaintext attack need not be considered as a valid attack. This assumption will not cause any loss of generality, due to the fact that in all practical applications, the client machine will be geographically separated from the cloud servers. This paper attempts to enhance the security of the Hill cipher so that it can withstand the known-plaintext attack, preserving additive homomorphism.

The major contributions of this work are listed below.

1. A symmetric key additively homomorphic encryption scheme based on affine Hill Cipher is proposed to provide confidentiality of outsourced data and to support linear operations in the encrypted domain.
   (a) In order to reduce the computational complexity involved in finding the inverse of the key matrix during decryption, self invertible matrices are used in this work.
   (b) In the proposed scheme, different random vectors are generated for different data blocks through a novel design method based on a combination of linear feedback shift registers (LFSR) which helps to improve security of Hill cipher based encryption scheme.
   (c) The method for reseeding the LFSR is designed so that randomness and security properties will be preserved while homomorphically combining the blocks. Through mathematical analysis, it is proved that the linear combination of encrypted blocks neither nullifies the effect of individual random vectors nor destroys the randomness properties.
2. The suitability of the proposed encryption scheme for encrypted domain MR-CT image fusion is analyzed through comparing simulation results in encrypted domain with those of plaintext domain in terms of various subjective and objective performance metrics. It is verified that encrypted domain image fusion provides same accuracy levels as that of plaintext domain image fusion.
3. Security of the proposed encryption scheme against cipher text only attack and known plaintext attack is established through cryptanalysis and in terms of resistance against different statistical attacks.

The rest of the paper is organized as follows. Section 2 presents the summary of related works while Section 3 details the system model and adversary model considered. Section 4 gives the description of the proposed homomorphic encryption scheme and Section 5 provides the detailed performance analyses. The security analyses is described in Section 6 followed by concluding remarks in Section 7.

2 Related work

In this section, the recent works on homomorphic encryption schemes and methods for securing medical images are discussed in detail.

The idea of processing encrypted data was first addressed by Rivest et al. [26]. Fully homomorphic encryption schemes which allow any computations on encrypted data was introduced by Gentry [10]. Several schemes [4, 5, 11] were proposed following this work. Even though many schemes with optimizations were proposed, these schemes still

remain computationally complex and impractical. Whereas partially homomorphic encryption schemes support computations only for either multiplication or addition. RSA [27], Paillier [23] and Goldwasser-Micali [12] are some of the cryptosystems that support partial homomorphism. RSA is homomorphic over multiplication whereas the other two schemes are additively homomorphic which allow linear combination operations on the encrypted data.

Among the additive homomorphic encryption schemes, Paillier is the widely used for image processing works on encrypted domain [21, 25, 32]. The basic image scaling and cropping operations in encrypted domain is proposed in [21]. A reversible data hiding approach based on Paillier scheme, where the hidden data is directly embedded into the encrypted images is proposed in [32]. A digital watermarking scheme for secure transmission of medical images based on Paillier encryption is introduced in [25]. In this scheme, the encrypted watermark image is directly embedded into the encrypted image. However, the Paillier scheme suffers from the following issues. The storage overhead of Paillier scheme is twice as the size of ciphertext is double the plaintext size. Moreover, in Paillier scheme, to provide 128-bit security, the modulus required is a prime number of size 2048bits. Hence, this scheme has high computational complexity due to the modular exponentiation operations involved during encryption and decryption.

Some notable works for securing medical images during storage or/and transmission have been proposed in [8, 9, 31, 36]. A watermark embedding scheme based on Arnold transform for ensuring authentication of medical images is proposed in [31]. For confidentiality of medical images, the watermarked images are encrypted through chaos-based encryption in [36]. For secure transmission and storage of the medical images in the cloud server, the outsourced images are encrypted using elliptic curve cryptography, whose keys are chosen using hybrid swarm optimization in [9]. A hybrid encryption scheme developed from AES and RSA is used for securing medical data while transmission in IoT environment is presented in [8]. Even though all these works ensure secure storage or transmission of medical images, these schemes are not designed to support secure processing in encrypted domain.

The use of Arnold transform [31] to scramble the medical images to be fused suffers from known-plaintext attack (KPA) since the transform matrix used for encrypting the images to be fused need to be same to support fusion in encrypted domain. Similarly, chaos based encryption [36] is a one-time pad encryption, where the keystreams generated through chaotic map acts as a one-time pad. This method is also prone to KPA and the randomness properties of the keystream cannot be ensured while homomorphically combining the encrypted images. The elliptic curve cryptography adopted in [9] can be modified to support additive homomorphism as detailed in [33]. However, the storage overhead of this elliptic curve based scheme is double and has high computational complexity. The hybrid encryption scheme based on AES and RSA [8] cannot be used for encrypted domain processing since AES does not support homomorphism over encryption.

Hill cipher [35] is a potential candidate for additive homomorphic encryption system for encrypted domain image fusion due to the linearity in encryption and decryption operations. Moreover, it introduces no storage overhead and has less computational complexity. However, the original Hill cipher [35] is susceptible to KPA. A symmetric key additively homomorphic encryption scheme based on Hill cipher, namely iterated Hill cipher (IHC) is proposed in [6]. Although the authors claim it to be secure against KPA, later in [38], it is proved that IHC can be broken through KPA. Another version of Hill cipher that can withstand KPA is affine Hill cipher [37]. In this scheme, unique random vectors are added to the ciphertexts generated through original Hill cipher. Nevertheless, this scheme cannot be

directly converted to provide additive homomorphism as the randomness properties of the random vector cannot be ensured while homomorphically combining the data.

In the proposed affine Hill cipher based homomorphic encryption scheme, the linearity in encryption process is retained in order to preserve additive homomorphism. The random vectors used for encryption are keystreams generated using linear feedback shift registers (LFSR) due to the good randomness properties and low structural complexity of LFSR. At the same time, care has been taken to preserve the randomness and security properties of the random vectors, while homomorphically combining the data.

3 System model and adversary model

3.1 System model

The system model considered is an end to end secure cloud based medical image fusion scenario as shown in Fig. 1, where the encrypted medical images of different modalities are outsourced to the cloud; the cloud fuses the images in encrypted domain and the fused encrypted image is accessed by the healthcare provider, who further decrypts the fused image. Multiscale decomposition method is a widely used technique for multi-modal image fusion [7]. In this method, the images of different modalities are decomposed using wavelets and the wavelet coefficients corresponding to images of different modalities are combined using appropriate fusion rule to obtain the fused coefficients. Then the final fused image is obtained by taking the inverse wavelet transform of the decomposed image consisting of fused coefficients. The fusion rule generally used are averaging, weighted averaging etc. which are linear operations [16]. In the system model considered, the preprocessing of images of different modalities includes image decomposition using appropriate wavelets and rounding off the pixel values based on the requirements of the encryption scheme.

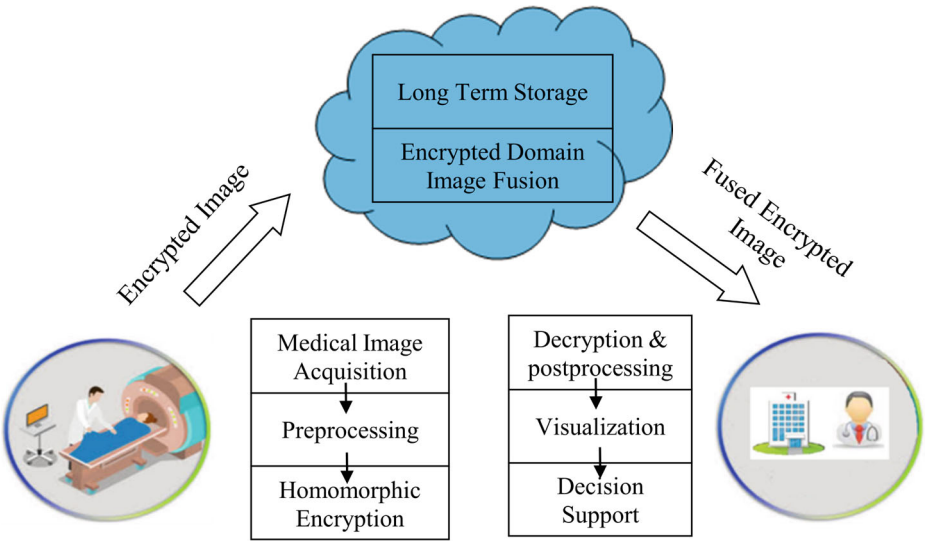

Fig. 1 System Model for secure medical image fusion

There is a need for additive homomorphic encryption scheme to facilitate image fusion in encrypted domain.

3.2 Adversary model

Cloud systems use distributed storage architecture for reliable data storage and securing data in distributed storage systems (DSS) assumes that the adversaries have access to only a subset of storage servers. The works on secure DSS [17, 24, 29] aims to resist ciphertext-only attack against adversaries who have access to only a limited number of servers in the DSS and is achieved through adding randomness to the data. But in practical DSS such as those with cloud storage, the storage servers may be distributed in the same geometrical environment. So in this work, we consider computationally bounded passive adversaries, who can eavesdrop on all the cloud servers. Also, based on the type of data stored, the adversary may have knowledge of some plaintexts with which he can try to mount a known-plaintext attack.

4 Proposed scheme for encrypted domain processing

The design of the proposed scheme for image fusion in encrypted domain is inspired from the Hill cipher [35] construction. In original Hill cipher, the encryption of a plaintext vector, m of length 'p' is done by multiplying it with a secret invertible random matrix, G of size $p \times p$ and the plaintext is decrypted by multiplying the ciphertext with the inverse of G matrix. The set of all possible keys or key space in the case of Hill cipher is the set of all possible invertible $p \times p$ matrices from the space of all $p \times p$ matrices. Iterated Hill cipher (IHC) [6] is proposed as a homomorphic encryption scheme and it extends the key space to the set of all $p \times p$ matrices. An initialization vector of length 'p' and the iteration number are kept secret in addition to G in the case of iterated Hill cipher. In IHC, the encryption and decryption are done using an iterative algorithm and the encoding results corresponding to the k^{th} and $(k-1)^{th}$ iteration form the ciphertext, where k is the number of iterations. Thus the size of ciphertext in the case of IHC will be double the size of plaintext, which will result in storage overhead. The authors claim that iterated Hill cipher can be made secure against known plaintext attack by changing the initialization vector during every encryption. But it is proved in [14] that this will spoil the homomorphic property offered by the encryption scheme. Furthermore, in [38], the authors showed that it can be broken through known plaintext attack even if the initialization vector is changed during every encryption. An improved version of the Hill cipher which can withstand known plaintext attacks is Affine Hill cipher [20, 37]. Affine Hill cipher construction relies on adding unique random vectors to the ciphertexts generated through original Hill cipher. But this scheme cannot be directly converted to support homomorphic operations since the randomness and security properties of the random vector cannot be ensured while homomorphically combining the image (data) blocks.

An affine Hill cipher based additive homomorphic encryption scheme is proposed in this paper to securely store the data in cloud and to support encrypted domain processing. While designing the scheme, care has been taken to preserve the randomness and security properties of the random vector, while homomorphically combining the data. In addition, in this work self invertible or involutory matrices are used as key matrix, G. This facilitates the use of same matrix, G for encryption and decryption and reduces the computational complexity involved in finding the inverse of G matrix during decryption.

The details of the proposed encryption scheme are as follows:

Encryption The ciphertext, $c_i \in F_q^p$ of length 'p' whose elements are chosen from finite field, F_q, corresponding to i^{th} plaintext, $m_i \in F_q^p$ is given by

$$c_i = E_K(m_i) = G \cdot m_i + r_i \tag{1}$$

where $G \in F_q^{p \times p}$ is a self-invertible matrix and $r_i \in F_q^p$ is a random vector which is different for each plaintext.

Decryption The plaintext, $m_i \in F_q^p$ corresponding to the ciphertext, $c_i \in F_q^p$ can be retrieved by

$$m_i = D_K(c_i) = G.(c_i - r_i) \tag{2}$$

Homomorphic property The encryption operation supports additivity and homogeneity properties which are the requirements for an additive homomorphic encryption scheme. Let $m_1, m_2 \in F_q^p$ represent two plaintext messages and $r_1, r_2 \in F_q^p$ represent the corresponding random vectors. Then homomorphic properties of the encryption scheme can be defined as

Additivity

$$\begin{aligned} E_K(m_1) + E_K(m_2) &= (G.m_1 + r_1) + (G.m_2 + r_2) \\ &= G.(m_1 + m_2) + (r_1 + r_2) \\ &= E_K(m_1 + m_2) \end{aligned} \tag{3}$$

Homogeneity

$$\begin{aligned} \beta.E_K(m_1) &= \beta.(G.m_1 + r_1) = G.\beta m_1 + \beta r_1 \\ &= E_K(\beta m_1) \end{aligned} \tag{4}$$

where $\beta \in F_q$ represents a scalar.

To ensure additive homomorphism as given in (3) and (4), it is essential to design random vectors 'r_i' properly. For (5) to hold, it is required that sum of random vectors $r_1 + r_2$ should yield a random vector with properties same as that of r_1 and r_2. Similarly for (4) to hold, the scalar multiple of random vector $\beta \cdot r_1$ should also have same randomness properties as r_1. During decryption operation to retrieve the plaintext, it is essential to have the knowledge of the effective resultant random vector. Since encryption and decryption are done at the client side, the client can remove the effect of the random vectors with the help of keys used for generating it.

Following section discusses how random vectors can be designed to satisfy these required properties.

4.1 Design of random vector

As the image fusion technique mainly involves averaging and weighted averaging operations, linear combination can be considered as the generalized operation that is required in the encrypted domain image fusion. In order to ensure homomorphism for linear operations, the set of random vectors used for encryption should be closed under linear combination operations. That means linear combinations of random vectors should yield random vectors of the same randomness properties. In order to satisfy these requirements, we are using a well-designed combination of LFSRs for generating random vectors. The proper

design of the random vectors is important as the randomness properties offered by the LFSR keystream will be spoiled if the linear combination of random vectors yield a null vector while linearly combining ciphertexts. Therefore, the secret initial states of LFSR used for generating different random vectors should also be derived properly to retain the randomness properties of the random vector in the linearly combined ciphertext.

It is well known that keystream constituting one period of the LFSR output, satisfy Golomb' s randomness properties [19]. Also linear combination of these output keystreams is a keystream generated from the linear combination of corresponding states.

Theorem 1 *The keystreams generated by the LFSR satisfies superposition property.*

Proof (1) Additivity property –The sum of the keystreams is a new keystream generated by an initial state which is the sum of initial states corresponding to individual keystreams.

Let $k(x) = k_0 + k_1x + k_2x^2 + \cdots + k_{L-1}x^{L-1}$ be the polynomial representation of initial state and $g(x) = g_0 + g_1x + g_2x^2 + \cdots + g_Lx^L$ be the feedback polynomial of LFSR, where $k_i, g_i \in F_q$. Then the state of the L-length shift register of LFSR initially consists of values, $k_0, k_1, k_2, \cdots, k_{L-1}$, which are coefficients of k(x) and the tap weights of the feedback connections of the LFSR are decided by $g_0, g_1, g_2, \cdots, g_L$, the coefficients of $g(x)$. Hence, the output sequence with period $q^L - 1$ generated by the LFSR can be represented as

$$a(x) = f(x)/g(x), wheref(x) = \sum_{i=0}^{L-1}\left(\sum_{j=0}^{i} k_j g_{i-j}\right)x^i \tag{5}$$

Suppose $k_1(x) = k_{10} + k_{11}x + k_{12}x^2 + \cdots + k_{1L-1}x^{L-1}$ and $k_2(x) = k_{20} + k_{21}x + k_{22}x^2 + \cdots + k_{2L-1}x^{L-1}$ are two different initial states of the LFSR with the same feedback polynomial $g(x)$. Let $a_1(x)$ and $a_2(x)$ represent the output sequences generated by LFSR corresponding to intial states, $k_1(x)$ and $k_2(x)$ and feedback polynomial, $g(x)$. Then using (5), $a_1(x)$ and $a_2(x)$ can be represented in terms of k_{ij} as

$$\begin{aligned} a_1(x) &= \left[\sum_{i=0}^{L-1}\left(\sum_{j=0}^{i} k_{1j} g_{i-j}\right)x^i\right]/g(x) \\ a_2(x) &= \left[\sum_{i=0}^{L-1}\left(\sum_{j=0}^{i} k_{2j} g_{i-j}\right)x^i\right]/g(x) \end{aligned} \tag{6}$$

The sum of output sequences or keystreams, $a_1(x) + a_2(x)$ of the LFSR can be expressed as

$$\begin{aligned} a_1(x) + a_2(x) &= \left[\sum_{i=0}^{L-1}\left(\sum_{j=0}^{i}(k_{1j} + k_{2j}) g_{i-j}\right)x^i\right]/g(x) \\ &= \left[\sum_{i=0}^{L-1}\left(\sum_{j=0}^{i} k_{3j} g_{i-j}\right)x^i\right]/g(x) \\ &= a_3(x) \end{aligned} \tag{7}$$

where $a_3(x)$ is a keystream generated by the initial state $k_3(x) = k_1(x) + k_2(x)$. Equation 7 shows that the sum of keystreams result in another keystream, $a_3(x)$, which can be generated

by an LFSR with initial state, $k_3(x)$, which is equal to the sum of the initial states, $k_1(x)$ and $k_2(x)$. □

Proof (2) Homogeneity property –The scalar multiple of a keystream is a new keystream generated by an initial state which is the scalar multiple of the initial state corresponding to original keystream.

Let $a(x)$ be the keystream generated by key, $k(x)$. Then $b(x) = \beta \cdot a(x)$ is the keystream generated by $k_\beta(x) = \beta \cdot k(x)$

$$\begin{aligned}
\beta \cdot a(x) &= \beta \cdot \left[\sum_{i=0}^{L-1} \left(\sum_{j=0}^{i} \left(k_j\right) g_{i-j} \right) x^i \right] / g(x) \\
&= \left[\sum_{i=0}^{L-1} \left(\sum_{j=0}^{i} \beta \cdot k_j g_{i-j} \right) x^i \right] / g(x) \\
&= \left[\sum_{i=0}^{L-1} \left(\sum_{j=0}^{i} k_{\beta j} g_{i-j} \right) x^i \right] / g(x) \\
&= b(x) \qquad (8)
\end{aligned}$$

□

Due to these properties of LFSR keystream, the random vectors for the proposed homomorphic encryption system are chosen as the output keystreams of an LFSR. The initial state of the LFSR, $k(x)$ and the feedback polynomial $g(x)$, which decide the feedback connections of the LFSR are kept secret and form part of the secret key of the proposed encryption scheme. The length of the random vector 'p' is to be chosen as a value close to an integral multiple of period of LFSR to ensure randomness properties, i.e., $p \cong c(q^L - 1)$, where 'L' is the length of LFSR and 'c' is a nonzero integer.

In the proposed scheme, it is required to generate different random vectors corresponding to different data blocks to retain homomorphism over linear operations. In order to facilitate generation of a distinct random vector corresponding to each distinct data block to be homomorphically combined, it is required to derive different initial states for the LFSR from the initial secret key through a proper design method. Therefore, next attempt in the design of proposed encryption scheme is to devise a method for generating multiple initial states from the initial secret key.

4.1.1 Properties of random vector

The random vector should be generated to satisfy the randomness and security properties.

Randomness property If N image blocks are to be linearly combined during image fusion, it is required to have at least N linearly independent random vectors for encryption. This is to ensure that, when ciphertext blocks are linearly combined, the corresponding random vectors obtained through linear combination operation will not yield a null vector so that the security of encryption operation is retained. From the previous discussions on LFSR theory it can be clearly seen that, linearly independent initial states of an LFSR will result in linearly independent random vectors. Also, since each state of an LFSR of length L forms a L-dimensional vector in a vector space V over F_q, there can be only L linearly independent

initial states. Therefore, the minimum possible length of LFSR 'L' has to be chosen as at least N ie, Length of LFSR, $L \geq N$, where N is the number of data blocks to be combined. Now, in order to ensure that the generated random vectors are linearly independent, the initial state s_i of LFSR corresponding to each message block m_i where $1 \leq i \leq N$, can be derived from the initial key $k = (k_0, k_1, k_2, \cdots, k_{L-1})$ by a cyclic shifting operation. To complete the design of random vector for encryption, it is required to arrive at the number of cyclic shifting operations that can be performed on a vector of length 'L' so that the set of shifted vectors remain linearly independent.

Theorem 2 *For an LFSR of length L, the set of L initial states generated by cyclically shifting an initial secret key $k(x)$ are linearly independent if gcd $(k(x), x^L - 1)$ is a polynomial of degree zero.*

Proof Let the initial state of LFSR which acts as the secret key be represented as $k = (k_0, k_1, k_2, \cdots, k_{L-1})$, where $k_i \in F_q$. Then define a shift operator, $T : V \mapsto V$ by

$$T(k_0, k_1, k_2, \cdots, k_{L-1}) = (k_{L-1}, k_0, k_1, \cdots, k_{L-2}) \tag{9}$$

If the initial secret key, k and its $L-1$ shifted versions are arranged as rows of a matrix, K, it will form a $L \times L$ circulant matrix as shown in (10).

$$K = \begin{bmatrix} k \\ Tk \\ \vdots \\ T^{L-2}k \\ T^{L-1}k \end{bmatrix} = \begin{bmatrix} k_0 & k_1 & \cdots & k_{L-2} & k_{L-1} \\ k_{L-1} & k_0 & \cdots & k_{L-3} & k_{L-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ k_2 & k_3 & \cdots & k_0 & k_1 \\ k_1 & k_2 & \cdots & k_{L-1} & k_0 \end{bmatrix} \tag{10}$$

All the 'L' initial states obtained by taking the cyclic shifted versions of $k(x)$ will be linearly independent if the circulant matrix is full rank.

Let U denote the circulant matrix whose entries are

$$U_{ij} = \begin{cases} 1, j - i \equiv 1(mod L) \\ 0, j - i \not\equiv 1(mod L) \end{cases} \tag{11}$$

Then the $L \times L$ circulant matrix K corresponding to k is given by $K = \sum_{i=0}^{L-1} k_i U^i$ where $U^0 = I$, the identity matrix of size $L \times L$ and U^1 is obtained by cyclically shifting each row of U^0 with shift operator T. Now, if K is invertible over F_q, then there exists a circulant matrix $M = \sum_{i=0}^{L-1} m_i U^i$, where $m_i \in F_q$ such that $K.M = I$.

If $k(x) = \sum_{i=0}^{L-1} k_i x^i$ be the polynomial representation of circulant matrix, K, then finding the inverse of K is equivalent to finding a polynomial $m(x) = \sum_{i=0}^{L-1} m_i x^i$ in $F_q[x]$ such that

$$k(x).m(x) \equiv 1[mod(x^L - 1)] \tag{12}$$

The congruence modulo $(x^L - 1)$ follows from the equality $U^L = I$. Using Extended Euclidean algorithm, (12) can be written as

$$k(x).m(x) + t(x).(x^L - 1) = 1 \tag{13}$$

Thus the $L \times L$ circulant matrix will be of full rank if gcd $(k(x), (x^L - 1)) = v$, where v is a non-zero integer in F_q. In general, if degree of gcd $(k(x), (x^L - 1))$ is 's' , then the rank of circulant matrix is ' $L - s$' . So if $s = 0$, then the matrix K will be full rank and all the rows are linearly independent. □

So for encrypting N image blocks, the length of LFSR should satisfy $L - s \geq N$ where s = degree of gcd $(k(x), (x^L - 1))$ for the initial secret key polynomial, $k(x)$.

Security property If the initial states of the LFSR for generation of random vectors are derived from initial secret key by simple linear shifting operations, it can cause security leakage as discussed below:

Let c_1 and c_2 be the ciphertexts corresponding to plaintexts m_1 and m_2, which are to be linearly combined in the encrypted domain. Then the data blocks m_1 and m_2 will be encrypted with the same self-invertible matrix, G and random vectors r_1 and r_2 respectively as

$$c_1 = G.m_1 + r_1 \tag{14}$$

$$c_2 = G.m_2 + r_2 \tag{15}$$

where $r_2 = T_R(r_1)$, T_R represents the right shift.

If c_2' represents the ciphertext obtained by left shifting c_2, then

$$c_2' = T_L(c_2) = T_L(G \cdot m_2) + T_L(r_2) \tag{16}$$

where T_L represents the left shift.

Therefore,

$$\begin{aligned} c_1 - c_2' &= G \cdot [m_1 - T_L(m_2)] + r_1 - T_L(r_2) \\ &= G \cdot [m_1 - T_L(m_2)], \end{aligned} \tag{17}$$

Thus the effect of random vector can be removed from ciphertext. So to enhance the security, the LFSR initial state needed for generating successive random vectors are derived by shifting the previous initial state and multiplying with a random element from F_q. This helps to prevent the security leakage while preserving linear independence and there by randomness properties.

Algorithm 1 For generating random vector.

Input: $k_1, \alpha_1, g(x)$
Output: r_i
1: **for** $i = 1 : N$ **do**
2: r_i = LFSR-PRNG(k_i)
3: $k_{i+1} = T_{Ri}(k_1) \cdot \alpha_i$
4: α_{i+1} = LFSR-State(α_i)
5: **end for**
6: **return** r_i

Algorithm 1 gives the procedure for generating random vectors. The inputs to the algorithm are initial secret key of LFSR, k_1 and feedback polynomial, $g(x)$ and initial seed α_1 to generate random multiplier α_i. Output is random vector, r_i used for encrypting each data block, m_i. In this algorithm, LFSR-PRNG refers to the pseudorandom number generator based on LFSR which outputs 'N' random elements to form random vectors, r_i used for encryption based on the initial state, k_1 and the feedback polynomial $g(x)$. LFSR-State refers to LFSR state updation with initial state α_1 and outputs a single random element. $T_{Ri}(k_1)$ in the algorithm indicates right shift of the initial state k_1 by 'i' bits.

4.2 Design of key matrix

Random invertible matrices are used as key matrix in original Hill cipher. However, the complexity in finding such a matrix increases with increase in the size of the matrix and field size, q. Moreover, during decryption the inverse of the matrix needs to be computed which will increase the decryption time and complexity. In order to overcome these problems, in this work we make use of self-invertible or involutory matrix [1] as key matrix. A matrix, G is said to be self-invertible if $G^{-1} = G$. Let G be a $p \times p$ involutory matrix, which can be written as $G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$, where $G_{ij}, 1 \leq i, j \leq 2$ are matrices of order $p/2 \times p/2$. An involutory matrix over any field will satisfy the following properties.

1. The determinant of an involutory matrix is ± 1, i.e., $|G| = \pm 1$.
2. The square of an involutory matrix is an identity matrix, i.e., $G^2 = I$

Assuming $|G| = -1$ and from $G^{-1} = G$, the involutory matrix G can be obtained by solving the equation, $G_{12}G_{21} = I - (G_{11})^2$, since $G_{22} = -G_{11}$.

Algorithm 2 For generating key matrix.

Input: p, γ
Output: G

1: Randomly choose a $p/2 \times p/2$ matrix as G_{11}
2: Obtain $G_{22} = -G_{11}$
3: Generate $G_{12} = \gamma(I - G_{11})$ or $\gamma(I + G_{11})$
4: Then obtain $G_{21} = (I + G_{11})/\gamma$ or $(I - G_{11})/\gamma$
5: **return** G

Algorithm 2 give the steps for generating involutory key matrix, G. The inputs to the algorithm are the size of matrix, p and a random element, γ, where $\gamma \in F_q$. In this algorithm, the subblocks, G_{12} and G_{21} of the matrix are generated from the factors of $I - (G_{11})^2$.

4.3 Key space

The secret key for the proposed encryption scheme consists of the self invertible matrix, $G \in F_q^{p \times p}$, the initial states, k_1, α_1 of two LFSRs and the feedback polynomial $g(x)$ of LFSR.

4.4 Proposed secure medical image fusion

The complete schematic of the proposed encrypted domain MR-CT/PET image fusion over cloud is shown in Fig. 2. As mentioned in Section 3.1, we have considered DWT based MR-CT/PET image fusion using averaging rule. The hospital (client) computes the DWT of captured MR and CT/PET images using Haar wavelet. Then the client generates the encrypted MR and CT/PET images from their decomposed images using proposed Hill cipher based encryption scheme. For encryption, these decomposed images are first divided into blocks, and passed through pre-processsing operation. Pre-processing of image is required to ensure that the pixel values are integers after decomposition and averaging during fusion. Encrypted image blocks are sent to cloud for long term storage and image fusion. The cloud performs the encrypted domain image fusion of the encrypted MR and CT/PET image vectors. In order to retrieve the final fused image, the health care provider (client) access the encrypted version of the fused MR-CT/PET image from the cloud. The client then performs the decryption of the fused image vectors and post-process the decrypted DWT

fused coefficients in order to match the results in the plaintext domain. The post processed fused image thus obtained after decryption is in DWT domain. Then the final fused MR-CT/PET image is generated by computing the inverse DWT of the post processed image.

The detailed steps of the proposed encrypted domain MR-CT image fusion with the required mathematical expressions are shown in Algorithm 3. Since single level Haar wavelet decomposition involves division by 2 and averaging the encrypted image blocks involves another division by 2, the original image pixels are preprocessed by multiplying with 4. The pixels of the corresponding encrypted MR and CT image vectors are added and multiplied with multiplicative inverse of 2 to obtain average of the encrypted image vectors. The effect of the preprocessing done before encryption is removed after decryption in the post processing step. In a similar manner, the steps for the proposed encrypted domain MR-PET image fusion can be obtained by replacing CT images with PET images.

Algorithm 3 For secure medical image fusion.

Input: MR image, I^{MR}, CT image, I^{CT}
Output: Fused MR-CT Image, I_F^{MR-CT}

Step1: DWT Computation
Obtain decomposed MR image, I_D^{MR} and CT image, I_D^{CT} using Haar wavelet
Step2: Image Division
Divide the decomposed MR and CT images into M blocks, I_{Dj}^{MR} and I_{Dj}^{CT}, $j = 1, 2, \cdots, M$
Step3: Preprocessing
Generate $\tilde{I}_{Dj}^{MR}(i,j) = I_{Dj}^{MR}(i,j) * 4$ and $\tilde{I}_{Dj}^{CT}(i,j) = I_{Dj}^{CT}(i,j) * 4$
Step4: Encryption
Arrange each MR and CT image block as a column vector represented by $\hat{I}_{Dj}^{MR}$ and $\hat{I}_{Dj}^{CT}$ respectively, each with size p
Multiply each image vector with G matrix of size $p \times p$ to obtain $G \cdot \hat{I}_{Dj}^{MR}$ and $G \cdot \hat{I}_{Dj}^{CT}$.
Add random vector, r_j^{MR} and r_j^{CT} of size $p \times 1$ to the output of the previous step to obtain encrypted MR and CT image vectors, $\hat{I}_{Ej}^{MR} = G \cdot \hat{I}_{Dj}^{MR} + r_j^{MR}$ and $\hat{I}_{Ej}^{CT} = G \cdot \hat{I}_{Dj}^{CT} + r_j^{CT}$ respectively.
Step5: Encrypted Domain MR-CT Image Fusion
Add encrypted image vectors of corresponding MR and CT images to generate, $\hat{I}_{Ej} = \hat{I}_{Ej}^{MR} + \hat{I}_{Ej}^{CT}(mod\ q)$
Multiply this with multiplicative inverse of 2 to generate, $\hat{I}_{Fj}$
Step6: Decryption
Remove the effect of added random vector from the fused image, $\tilde{I}_{Fj} = \hat{I}_{Fj} - (r_j^{MR} + r_j^{CT})(mod\ q)$
Multipy it with inverse of G matrix, G^{-1} to obtain the decrypted fused image vectors, $\tilde{I}_{FDj} = G^{-1} \cdot \tilde{I}_{Fj}$
Step7: Post-processing
$\hat{I}_{FDj}(x,y) = \begin{cases} \tilde{I}_{FDj}(x,y)/4; \tilde{I}_{FDj}(x,y) < (q+1)/2 \\ (\tilde{I}_{FDj}(x,y) - q)/4; \tilde{I}_{FDj}(x,y) > (q+1)/2 \end{cases}$
Rearrange the image vectors into blocks and form the fused image in decomposed form, $\hat{I}_{FD}$
Step8: IDWT Computation
Obtain the final fused image, I_F^{MR-CT} by performing inverse Haar DWT of $\hat{I}_{FD}$
return I_F^{MR-CT}

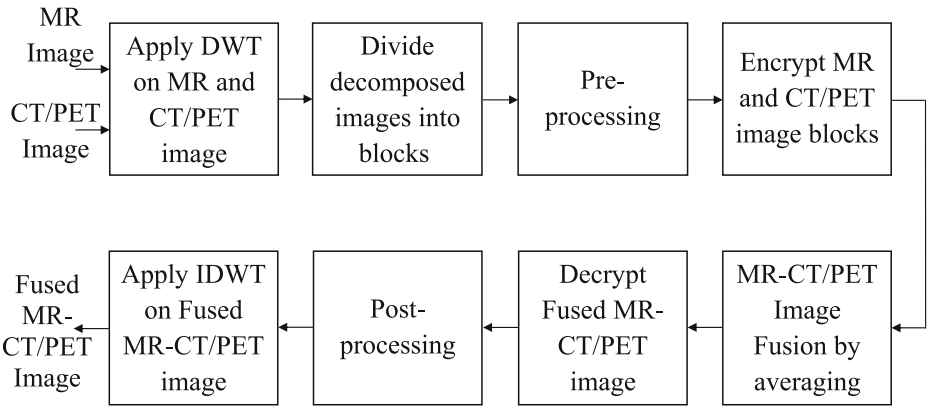

Fig. 2 Flow Diagram for proposed secure medical image fusion

5 Performance analysis

In this section, the accuracy of encrypted domain image fusion is analyzed in terms of subjective and objective performance metrics.

5.1 Simulation results and analysis

To evaluate the performance of the proposed encryption scheme, the fusion of medical images of different modalities are considered. Simulations of the proposed scheme are performed on PC with Intel(R) Xeon(R) CPU E3-1226 v3 3.3 GHz 16GB RAM running on Windows 10 Professional equipped with MATLAB R2015b environment.

For simulation, we have used standard MR, CT and PET image datasets from the Harvard university site which are available at http://www.med.harvard.edu/aanlib/home.html. MR and CT images of size 512×512 are first decomposed using single level 2D Haar wavelet. The decomposed MR and CT images to be encrypted are divided into 64 blocks and each block is encrypted by multiplying with self-invertible G matrix of size 4096×4096 which is followed by the addition of the random vector of length 4096. We have also considered the fusion of MR and PET images of size 512×512, where MR images are gray-scale and PET images are colour. As PET images provide functional information with low spatial resolution and MR images provide the tissue information with high resolution, the fusion of MR-PET image helps in better diagnosis.

The pixels of the image can be represented using an 8-bit integer since each pixel takes a value in the range 0 to 255. Hence the field size, q should be at least 257, which is the nearest prime ≥ 255. However, the field size should be large enough to accommodate the result of processing for preserving the accuracy of computation in encrypted domain. For example, if two encrypted image blocks are added, the resultant pixel can take value in the range 0 to 510. So in order to preserve this value on decryption, q should be a prime number ≥ 510. Since the final value of the pixels on decryption could be negative or positive, care should be taken to choose q such that it is at least twice greater than the range of values involved. Thus to ensure the correctness of the computation in encrypted domain, all the modulo operations have to be done using a prime number at least ≥ 4080 (=255*4*4). In

Table 1 Simulation parameters

Simulation parameters	
Image size	512×512
No. of decomposed image blocks	64
Size of G matrix	4096×4096
Length of random vector	4096
Field size	4091

this paper, all simulations for image fusion are done by choosing $q = 4091$. Table 1 shows the in-detail simulation parameters used in this paper.

The original MR, CT image, their decomposed images, encrypted images before and after fusion and decrypted fused image are shown in Fig. 3. From Fig. 3, it is clear that the encrypted images after fusion will not leak any information about the original images.

5.1.1 Subjective analysis

Since accuracy is an important parameter, the efficiency of the proposed homomorphic encryption scheme in fusing MR and CT images is evaluated by comparing the accuracy of the encrypted domain (ED) image fusion with that of the plaintext domain (PD) fusion. The subjective analysis of secure MR-CT image fusion for different data sets is shown in Fig. 4, which includes the MR and CT images and the corresponding fused images in plaintext domain (PD) and encrypted domain (ED). The subjective analysis of secure MR-PET image fusion for different data sets is shown in Fig. 5.

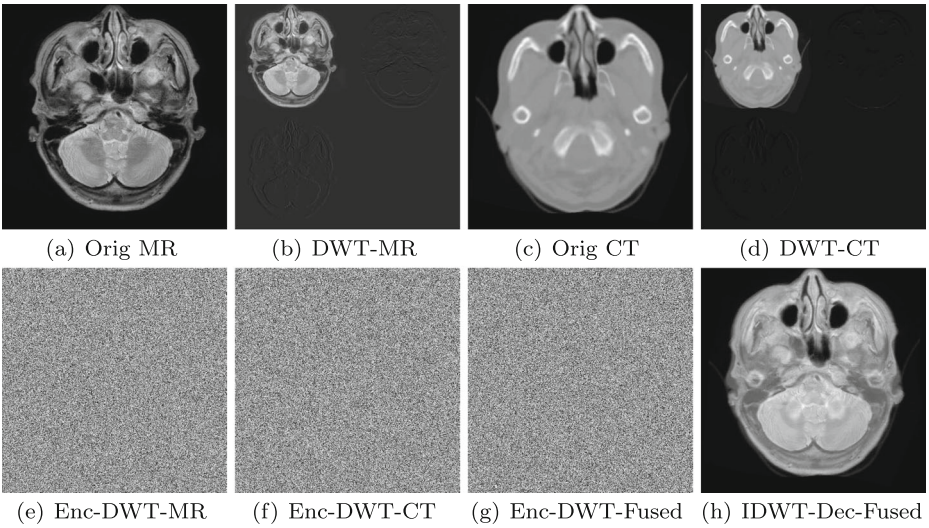

Fig. 3 Original MR and CT images, its decomposed and encrypted versions, encrypted and decrypted fused MR-CT images

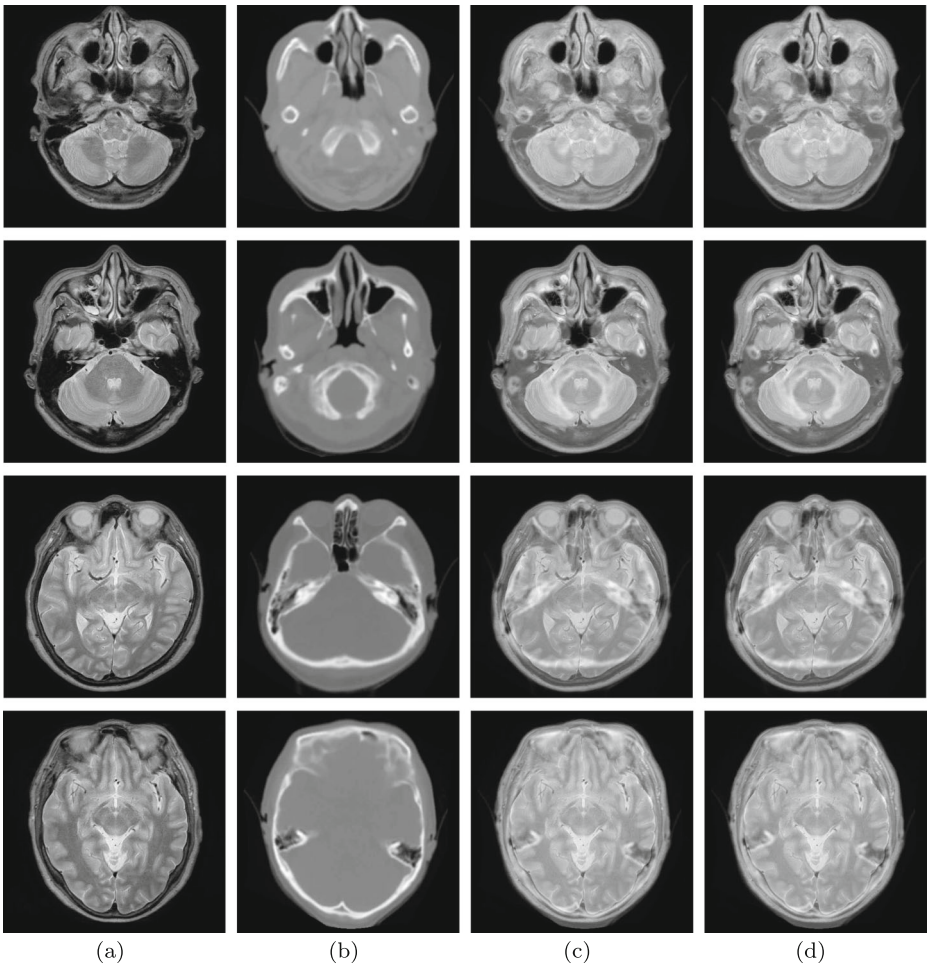

Fig. 4 Example images **a** MR image, **b** Registered CT image, **c** Fused image in PD, and **d** Fused image in ED

5.1.2 Objective analysis

The performance of the encrypted domain image fusion is also analyzed using objective performance measures. There are two classes of objective performance measures, one requires a reference image while the other one does not require reference image. As it is difficult to obtain an ideal fused image as a reference image, non-reference metrics such as entropy (H), Standard Deviation (SD) and mutual information (MI) are used to evaluate the quality of a fused image [3]. Feature Mutual Information (FMI) metric [13] is another non-reference metric which calculates the amount of information conducted from the source images to the fused image. Table 2 shows the H, SD, MI and FMI values of plaintext domain (PD) and encrypted domain (ED) image fusion for different datasets. It should be noted from Figs. 4, 5 and Table 2 that the encrypted domain results match with the plaintext domain results.

The fused image quality in ED is also analyzed using objective measures relying on reference image, considering PD fused image as the reference image. Maximum difference

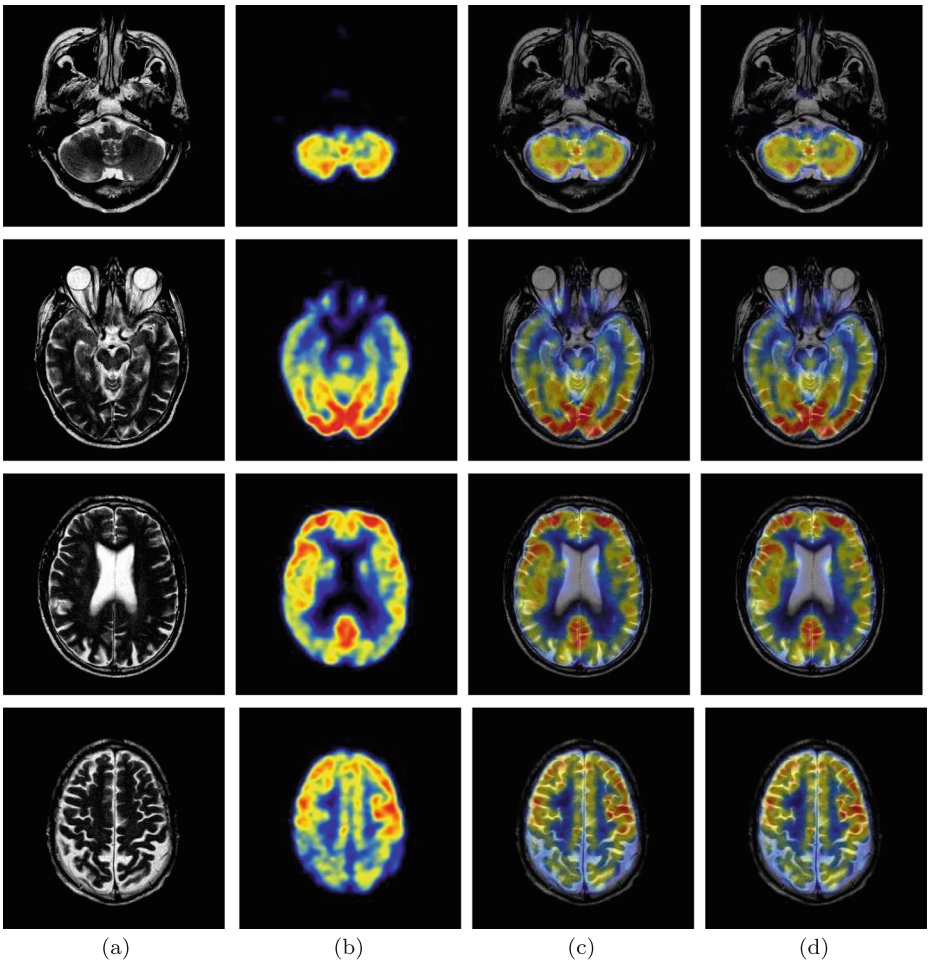

Fig. 5 Example images **a** MR image, **b** PET image, **c** Fused image in PD, and **d** Fused image in ED

(MD), Average difference (AD), Normalized absolute error (NAE) , Mean-square error (MSE) [2], Image quality index (IQI) [39], Normalized correlation coefficient (NCC), Structural content (SC) and Structural similarity index (SSIM) [3] are used to evaluate the quality of the fused image in ED with respect to PD. Table 3 shows the values of these parameters corresponding to different datasets. The very low values of MD, AD, NAE and MSE ranging from 10^{-13} to 10^{-28} indicates that the difference between ED fused image and PD fused image are negligibly small. Moreover NCC, SC and SSIM values corresponding to all datasets are equal to the maximum value, 1 and IQI is also close to 1. This indicates that ED and PD fused images are very close to each other.

6 Security analysis

In this section, the security of the proposed scheme is first evaluated through resistance against various statistical attacks, which is then followed by mathematical cryptanalysis.

Table 2 Quantitative Evaluation Results based on source images and fused image in PD and ED

Datasets	H		SD		MI		FMI	
	PD	ED	PD	ED	PD	ED	PD	ED
Dataset 1	6.2955	6.2953	103.44	103.44	2.1933	2.1933	0.8858	0.8858
Dataset 2	5.3307	5.3306	122.93	122.93	2.2972	2.2968	0.8894	0.8893
Dataset 3	5.5309	5.5309	122.11	122.11	2.4101	2.4097	0.8879	0.8880
Dataset 4	6.4180	6.4180	111.99	111.99	2.3919	2.3916	0.8835	0.8835
Dataset 5	6.0527	6.0527	119.26	119.26	2.6342	2.6341	0.8919	0.8919
Dataset 6	6.7761	6.7633	107.52	107.52	2.2600	2.2599	0.8860	0.8860
Dataset 7	6.2675	6.2674	112.54	112.54	2.2669	2.2668	0.8845	0.8845
Dataset 8	6.3635	6.3624	98.03	98.03	2.0766	2.0766	0.8757	0.8757
Dataset 9	6.1437	6.1437	103.44	103.44	2.1648	2.1647	0.8886	0.8886
Dataset 10	6.3275	6.3275	105.22	105.22	2.2012	2.2012	0.8832	0.8832
Dataset 11	6.4846	6.4824	117.88	117.88	2.1187	2.1185	0.8749	0.8749
Dataset 12	6.1361	6.1359	111.01	111.01	2.3024	2.3023	0.8859	0.8859
Dataset 13	6.5765	6.5765	106.93	106.93	2.2701	2.2701	0.8772	0.8772

6.1 Histogram analysis

A good encryption scheme should generate a uniformly distributed histogram corresponding to the encrypted image. This prevents the adversary from acquiring any information about the original image from the histogram of the encrypted image. The original MR and CT images, their encrypted versions and fused MR-CT image in ED along with their histograms are shown in Fig. 6. From Fig. 6, it is clear that the proposed encryption scheme completely randomizes the plaintext since the histogram of the encrypted image follows uniform distribution. Moreover, the uniform distribution of the histogram of the ED fused image also

Table 3 Quantitative Evaluation Results of the fused image in ED considering PD fused image as reference image

Datasets	MD	AD	NAE	MSE	IQI	NCC	SC	SSIM
Dataset 1	1.71E-13	1.69E-14	1.54E-16	9.57E-28	0.9997	1	1	1
Dataset 2	1.71E-13	1.65E-14	1.57E-16	1.14E-27	0.9997	1	1	1
Dataset 3	1.71E-13	1.83E-14	1.59E-16	1.26E-27	0.9998	1	1	1
Dataset 4	1.71E-13	2.31E-14	1.59E-16	1.48E-27	0.9998	1	1	1
Dataset 5	1.71E-13	2.49E-14	1.71E-16	1.70E-27	0.9997	1	1	1
Dataset 6	1.71E-13	2.03E-14	1.47E-16	1.25E-27	0.9998	1	1	1
Dataset 7	1.71E-13	1.97E-14	1.59E-16	1.23E-27	0.9997	1	1	1
Dataset 8	1.71E-13	1.50E-14	1.48E-16	7.92E-28	0.9992	1	1	1
Dataset 9	1.71E-13	1.67E-14	1.56E-16	9.58E-28	0.9993	1	1	1
Dataset 10	1.71E-13	1.69E-14	1.53E-16	9.68E-28	0.9997	1	1	1
Dataset 11	1.71E-13	1.95E-14	1.53E-16	1.28E-27	0.9997	1	1	1
Dataset 12	1.71E-13	2.07E-14	1.58E-16	1.26E-27	0.9998	1	1	1
Dataset 13	1.71E-13	2.01E-14	1.54E-16	1.20E-27	0.9997	1	1	1

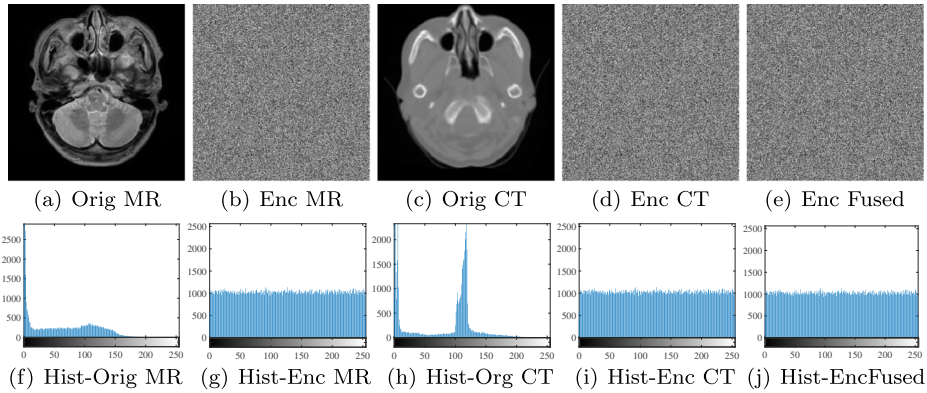

Fig. 6 Original MR and CT images, its encrypted versions and their corresponding histograms

indicates that no information is leaked after ED image fusion. Hence the proposed scheme is secure against histogram attack.

6.2 Correlation analysis

Usually the correlation of neighboring pixels of the plaintext image will be very high. In order to resist statistical attacks by exploiting the correlation between adjacent pixels, the encrypted image should have low correlation coefficients. The correlation analysis of the encrypted images using proposed scheme is performed by taking into consideration all possible adjacent cases (horizontal, vertical and diagonal). The correlation coefficient is computed using the following equations.

$$Corr_{x,y} = \frac{Cov(x, y)}{\sqrt{V(x)} \times \sqrt{V(y)}} \tag{18}$$

$$Cov(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}) \tag{19}$$

$$V(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 \tag{20}$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{21}$$

where x and y are values of two neighbouring pixels; and $\bar{x}$ and $\bar{y}$ are the mean values. $Cov(x, y)$ and $V(x)$ represent the covariance and variance respectively. Table 4 shows the correlation coefficients of adjacent pixels for original plaintext MR and CT/PET image; their encrypted versions before and after fusion corresponding to different datasets. It is clear from Table 4 that the correlation coefficients of encrypted MR and CT/PET images are very low compared to original MR and CT/PET images. This reveals that the proposed scheme completely randomizes the pixels and no statistical information is leaked from the encrypted image. Moreover, the very low values of correlation coefficients of the encrypted domain fused image indicate that the encrypted domain fusion will not leak any information about the original fused image, resisting correlation attacks.

Table 4 Correlation coefficients of MR and CT/PET images in plaintext domain (PD) and encrypted domain (ED); and fused MR-CT/PET image in ED

Datasets	Orientation	MR Image		CT/PET Image		Fused Image
		PD	ED	PD	ED	ED
Dataset 1	H	0.9912	0.0024	0.9984	−0.0009	0.0044
	V	0.9908	0.0012	0.9988	−0.0024	0.0012
	D	0.9844	0.0042	0.9972	−0.0023	−0.0026
Dataset 2	H	0.9942	−0.0001	0.9996	0.0020	0.0017
	V	0.9949	−0.0042	0.9995	0.0015	0.0001
	D	0.9905	−0.0009	0.9992	−0.0038	0.0039
Dataset 3	H	0.9943	0.0020	0.9993	0.0015	0.0000
	V	0.9951	0.0004	0.9994	0.0015	−0.0020
	D	0.9906	0.0025	0.9989	−0.0021	−0.0009
Dataset 4	H	0.9908	−0.0019	0.9979	0.0000	0.0044
	V	0.9916	−0.0002	0.9980	0.0013	0.0009
	D	0.9844	0.0022	0.9959	0.0022	−0.0033
Dataset 5	H	0.9933	−0.0032	0.9984	0.0038	−0.0002
	V	0.9935	−0.0015	0.9986	−0.0005	0.0006
	D	0.9884	0.0015	0.9969	−0.0003	0.0015
Dataset 6	H	0.9904	0.0008	0.9986	−0.0022	−0.0016
	V	0.9908	0.0019	0.9985	0.0002	−0.0009
	D	0.9828	0.0025	0.9971	0.0040	0.0020
Dataset 7	H	0.9916	−0.0028	0.9982	−0.0021	−0.0011
	V	0.9925	0.0017	0.9986	0.0011	0.0005
	D	0.9859	−0.0027	0.9967	−0.0035	0.0032
Dataset 8	H	0.9890	−0.0021	0.9982	−0.0022	−0.0022
	V	0.9864	−0.0015	0.9987	0.0016	−0.0048
	D	0.9787	−0.0019	0.997	0.0018	−0.0017
Dataset 9	H	0.9893	−0.0043	0.9981	−0.0003	−0.0001
	V	0.9879	0.0034	0.9986	−0.0012	0.0021
	D	0.9800	0.0000	0.9968	−0.0002	0.0024
Dataset 10	H	0.9844	−0.0015	0.9983	−0.0015	0.0004
	V	0.9908	0.0003	0.9988	−0.0002	0.0028
	D	0.9912	0.0007	0.9972	0.0020	0.0025
Dataset 11	H	0.9919	−0.0022	0.9986	0.0003	0.0002
	V	0.9924	0.0014	0.9988	0.0012	−0.0045
	D	0.9859	0.0010	0.9974	−0.0033	−0.0018
Dataset 12	H	0.9923	0.0002	0.9982	0.0009	−0.0058
	V	0.9932	0.0008	0.9981	0.0036	−0.0005
	D	0.9873	−0.0018	0.9965	−0.0017	0.0002
Dataset 13	H	0.9895	0.0018	0.9977	−0.0036	−0.0001
	V	0.9901	−0.0007	0.9983	0.0023	−0.0016
	D	0.9821	0.0020	0.9960	−0.0006	−0.0019

6.3 Key sensitivity analysis

Key sensitivity is an important parameter used to quantify the security of an encryption scheme. A good encryption scheme will provide totally different ciphertexts when encrypted with keys which are differing only in a few bits. The key sensitivity of the proposed scheme is first analyzed by decrypting the image with a key different from the encryption key by only 1 bit. Even though there is only one-bit change in key, it is observed that the decrypted image is completely random. The two commonly used parameters to analyze the key sensitivity are number of pixels change rate, NPCR and unified average changing intensity, UACI. The NPCR and UACI values are computed using the following expressions.

$$NPCR = \frac{\Sigma_{i,j} D(i,j)}{W \times H} \times 100\% \tag{22}$$

$$D(i,j) = \begin{cases} 0, c_1(i,j) = c_2(i,j) \\ 1, c_1(i,j) \neq c_2(i,j) \end{cases} \tag{23}$$

$$UACI = \frac{1}{W \times H} \left[\Sigma_{i,j} \frac{|c_1(i,j) - c_2(i,j)|}{255} \right] \times 100\% \tag{24}$$

Here c_1 and c_2 are two cipher images obtained by encrypting plain images with two different keys, k_1 and k_2 such that k_1 and k_2 differ in only one bit. W and H are width and length of the image. The results of NPCR and UACI values for different MR images are shown in Table 5. The encryption scheme provides high key sensitivity if NPCR value is above 99.5% and UACI value lies between 33.3% to 33.8% [40]. The results in Table 5 show that the proposed encryption scheme provides high key sensitivity.

6.4 Cryptanalysis

The secrecy of the data blocks stored in cloud, when the adversary gets access to all the ciphertexts, is mathematically analyzed in this section. It is also assumed that adversary can have knowledge about some data blocks, which is a valid assumption in scenarios where adversary knows the type of data being stored. Thus, the adversary knowing only the

Table 5 Key sensitivity analysis

Datasets	NPCR	UACI
Dataset 1	99.71%	33.54%
Dataset 2	99.71%	33.52%
Dataset 3	99.71%	33.56%
Dataset 4	99.71%	33.52%
Dataset 5	99.71%	33.45%
Dataset 6	99.71%	33.49%
Dataset 7	99.71%	33.54%
Dataset 8	99.71%	33.51%
Dataset 9	99.71%	33.54%
Dataset 10	99.71%	33.54%
Dataset 11	99.71%	33.43%
Dataset 12	99.71%	33.52%
Dataset 13	99.71%	33.51%

ciphertexts can mount a ciphertext only attack whereas the adversary having access to some plaintexts in addition to ciphertexts can mount a known-plaintext attack.

6.4.1 Ciphertext only attack

Since in the proposed homomorphic encryption scheme, the random vectors are designed properly to retain the randomness properties under linear combination operations, no attempt of the adversary on ciphertexts can remove the effect of random vectors. Also the random vectors are designed in such a way that for a message of block length 'p' , the random vectors have Hamming weight $\approx p/2$, so that no information about the matrix G chosen for encryption is leaked. Hence, the adversary is left with the only option of brute force attack to retrieve the data. The probability that the adversary retrieves the original data block through brute force attack can be evaluated from the keyspace of the proposed encryption scheme.

Theorem 3 *The adversary observing data stored in the DSS succeeds in retrieving the original data blocks with a probability* $1/\left(g_p \sum_{i=0}^{p} \frac{1}{g_i g_{p-i}}\right) \cdot \left(\prod_{j=1}^{r}(q^{d_j}-1)\right)$

Proof In order to extract the original message block, m_i from corresponding ciphertext block c_i, where the relationship between m_i and c_i is through $p \times p$ matrix G and $p \times 1$ vector, r_i as shown in (1), the adversary need to try all combinations of encrypting matrix, G and random vector r_i for a successful attack.

The key space for G is formed by all the $p \times p$ self-invertible invertible matrices over F_q. Finding the number of self-invertible or involutory matrices is equivalent to finding the number of matrices satisfying $G^2 - I = 0$ due to property 2 of involutory matrix mentioned in Section 4.2. The number of matrices that satisy the condition, $G^2 - I = 0$ over a finite field is given as Theorem 1 in [15]. Hence the number of $p \times p$ involutory matrices, $|G|$ over F_q is given by

$$|G| = g_p \sum_{i=0}^{p} \frac{1}{g_i g_{p-i}} \tag{25}$$

where $g_i = \prod_{k=0}^{i-1}(q^i - q^k)$, $0 < i < p$ and $g_0 = 1$.

As mentioned in Section 4.1, each random vector, r_i of length 'p' is generated using LFSR with an initial state consisting of 'L' symbols and feedback polynomial of degree L, where L is related to p and q as $p \cong c(q^L - 1)$, for a non zero integer c. The initial state and feedback polynomial form the part of the secret key. The key space corresponding to the random vector generation includes all possible combinations of feedback polynomial, $g(x)$ and initial state of LFSR, $k(x)$ from which consecutive states for reseeding the LFSR circuit are derived. It is well known from LFSR theory that for generating keystream with maximum length, the feedback polynomial should be primitive. The number of primitive polynomials of degree 'L' over finite field, F_q is $\phi(q^L - 1)/L$, where ϕ represents the Euler totient function. The initial state of LFSR which acts as the seed should satisfy the condition that the degree of polynomial corresponding to gcd $(k(x), x^L - 1)$ is zero (see Theorem 2). The number of such initial states, $k(x)$ is given by $\prod_{j=1}^{r}(q^{d_j} - 1)$, where d_j is the degree of $f_j(x)$ which corresponds to the irreducible factors of $x^L - 1$ [18]. Therefore, the key space corresponding to LFSR circuit which generates the random vectors is $\phi(q^L - 1)/L \cdot \prod_{j=1}^{r}(q^{d_j} - 1)$. It should be noted that it is only required to satisfy $L - s \geq N$, for linear

independence of random vectors where ' s' is the degree of polynomial which is gcd $(k(x), (x^L - 1))$. So it is possible to choose the full key space of $k(x)$ without any restriction for a sufficiently large value of L. Therefore, in general, the total keyspace is at least $|G| \cdot \phi(q^L - 1)/L \cdot \prod_{j=1}^{r}(q^{d_j} - 1)$. Hence the probability that an adversary succeeds in retrieving the key with brute force attack on keyspace is at least $1/|G| \cdot \phi(q^L - 1)/L \cdot \prod_{j=1}^{r}(q^{d_j} - 1)$. □

Considering a small field size, $q = 257$, block size, $p = 16$, length of LFSR, $L = 11$, the keyspace is approximately 2^{1119} and for a larger field size, $q = 4091$, the keyspace becomes approximately 2^{1677}. Therefore, the average computational complexity of the adversary in mounting a successful ciphertext only attack is at least $O(2^{1118})$ and $O(2^{1676})$ for $q = 257$ and $q = 4091$ respectively. The computational complexity of the adversary in mounting a ciphertext only attack further increases with increase in block size, p.

6.4.2 Known-plaintext attack

In scenario where an adversary knows the type of data and possess some plaintext blocks to be homomorphically combined, the attack boils down to known plaintext attack (KPA). In KPA, the attacker can retrieve the key, G with the help of 'p' plaintext blocks if he succeeds in removing the effect of random vector, r_i from the ciphertext. It is possible to separate the effect of G matrix and random vector, r_i by making the plaintext part zero through linear combination of known plaintext pieces. The cryptanalysis of the iterated Hill cipher detailed in [38] mounts a known-plaintext attack which uses the idea of generating an all zero plaintext by taking the linear combination of known plaintexts in such a way that the result of linear combination is zero. The steps in the cryptanalysis [38] can be summarized as follows.

1. Represent ciphertext, c in terms of plaintext vector, m and random vector, r as $c = \begin{bmatrix} A_1 & A_2 \end{bmatrix} . \begin{bmatrix} m \\ r \end{bmatrix} = A_1.m + A_2.r$, where A_1 and A_2 are $2p \times p$ matrices and c is of length $2p$.
2. If m is all zero vector, then ciphertexts can be considered as codeword, r generated by matrix A_2 and finding a parity check matrix, H such that $H.A_2 = 0$ will help to remove the random vector, r from ciphertext.
3. All zero plaintext vector, $m = 0$ can be obtained by taking the linear combination of any '$2p$' plaintexts. It is well known that by taking '$2p$' samples, 'p' linearly independent samples can be obtained with high probability.
4. Homomorphically create encryption of zero by taking the linear combination of corresponding ciphertexts using the same coefficients.
5. Arrange these ciphertexts as columns of a matrix, C and find H matrix such that $H \cdot C = 0$.
6. Using the H matrix, the random vector, r can be removed from ciphertext and the A_1 matrix can be found from the knowledge of any 'p' linearly independent plaintext message blocks.

Theorem 4 *If each message block is encrypted with different random vector, then the proposed homomorphic encryption scheme is secure to known-plaintext attack*

Proof If same random vector is used for encrypting every message block, then the adversary will be able to retrieve the G matrix and random vector, r using any $p + 1$ known plaintexts. In the proposed encryption scheme, since the random vector is changed during

each encryption, it will not be feasible to retrieve encryption matrix G in KPA. It can be proved that our encryption scheme is resistant to the known-plaintext attack on iterated Hill cipher. The arguments are as follows:

Using step 1, our encryption scheme can be represented as $c = \begin{bmatrix} G & I \end{bmatrix} \cdot \begin{bmatrix} m \\ r \end{bmatrix} = Gm + Ir$, where G is a $p \times p$ self-invertible matrix and I is a $p \times p$ identity matrix. For mounting the aforementioned cryptanalysis on our scheme, the attacker should be able to find a parity check matrix such that $H.I = 0$. Since the null space of Identity matrix is zero, the attacker cannot find such a matrix, H. As a result, the attacker will not succeed in decoupling the $G.m$ from random vector. Thus the proposed homomorphic encryption scheme resists known-plaintext attack given in [38]. □

Since mounting a known-plaintext attack by separating the contribution of the random vector r_i and matrix G is not possible, the key can be retrieved only by guessing the random vector and then finding the key matrix G by solving linear equations. Attacker can mount a KPA with 'p' plaintext-ciphertext pairs (m_i, c_i). From relationship between plaintext-ciphertext pairs 'p' linear equations can be formed with $(p^2 + p)$ unknowns, where p^2 unknowns are from G matrix and p unknowns are from random vector, r_i. Since with inclusion of additional plaintext-ciphertext pairs new unknowns are also added in the form of random vectors, increasing number of equations will not directly give a solution. Hence a successful attack need following steps.

1. Pick 'p' plaintext-ciphertext pairs (m_i, c_i).
2. Check whether the message matrix is invertible. If so, proceed to step (3); else return to step (1).
3. Choose one set of initial keys k_1, α_1 and feedback polynomial $g(x)$.
4. From p equations between pairs (m_i, c_i) solve for G matrix.
5. Repeat steps (3) and (4) for all possible sets of keys k_1, α_1 and $g(x)$ and form the table of possible solutions of G.
6. Pick a new plaintext-ciphertext pair (m_j, c_j).
7. Try with all possible solutions of G from the table and solve for G.

The number of possibilities of the key set k_1, α_1 and $g(x)$ are given by $\phi(q^L - 1)/L \cdot \prod_{j=1}^{r}(q^{d_j} - 1) \cdot (q-1)^{L-1}$, where $\phi(q^L - 1)/L$, $\prod_{j=1}^{r}(q^{d_j} - 1)$ and $(q-1)^{L-1}$ represent the number of possible feedback polynomials, $g(x)$, initial states, $k(x)$ and α_1 values for LFSR respectively. So in step (5) attacker has $\phi(q^L - 1)/L \cdot \prod_{j=1}^{r}(q^{d_j} - 1) \cdot (q-1)^{L-1}$ possible solutions for G and he has to try all these possibilities to solve for G in step (7). Therefore, the keyspace of the known-plaintext attack is $2 \cdot \phi(q^L - 1)/L \cdot \prod_{j=1}^{r}(q^{d_j} - 1) \cdot (q-1)^{L-1}$. Considering a small field size, $q = 257$, block size, $p = 16$, length of LFSR, $L = 11$, this keyspace is approximately 2^{175} and for a larger field size, $q = 4091$, the keyspace becomes approximately 2^{263} . Therefore, the average computational complexity of the adversary in mounting a successful known-plaintext attack boils down to $O(2^{174})$ and $O(2^{262})$ for $q = 257$ and $q = 4091$ respectively. It should be noted that an adversary can successfully mount this KPA only if he possesses $p + 1$ plaintext-ciphertext pairs. Since the number of data blocks, N is chosen to be at most equal to L in order to preserve the randomness properties in the homomorphically combined data blocks, it will be infeasible for an adversary to have so much information about the plaintext data. In MR-CT image fusion described in Section 4.4, $N = 2$ since the average of corresponding MR and CT image blocks are taken for fusion. Moreover, it is only required to use same G for encrypting data blocks to be homomorphically combined. So the aforementioned attack possibility can be prevented by using different keys for different sets of data blocks to be homomorphically combined.

6.5 Storage overhead and computational complexity

Storage overhead and computational complexity are two important parameters that determine the efficiency of the proposed encryption scheme. The storage overhead can be expressed in terms of the ciphertext expansion ratio, which is defined as the ratio of the size of ciphertext to the size of the plaintext. In the proposed encryption scheme, since the size of ciphertext is same as the size of plaintext, this scheme will not introduce any storage overhead.

The computational complexity of the encryption scheme can be expressed in terms of number of additions and multiplications. For encrypting a message block of size p, it is first multiplied with a G matrix of size $p \times p$, which is then followed by the addition of a random vector of size p. Multiplying the message block with matrix G involves p^2 multiplications and $(p^2 - p)$ additions in F_q while addition by the random vector involves p additions. So, the computational complexity for encrypting each message block includes p^2 modular multiplications and p^2 modular additions. Since the decryption operation also involves similar operations, the computational complexity of decryption is same as that of encryption. The bit complexity of modular multiplication and modular addition operation are $O((logq)^2)$ and $O(logq)$ respectively. Thus, the computational complexity of encryption and decryption process for the proposed scheme is $O(p^2(logq)^2)$ bit multiplications and $O(p^2.logq)$ bit additions.

6.6 Comparison with existing homomorphic encryption scheme

Paillier encryption [23] is the most popular additive homomorphic encryption scheme where addition of two plaintexts is realized by multiplying the corresponding ciphertexts. The plaintexts are represented as elements of Z_n, where Z_n denotes the set of integers modulo n and ciphertexts are represented as an integer modulo n^2, where n is a product of two large primes. So the size of the ciphertext will be double the size of plaintext and this will result in storage overhead. The security relies on the decisional composite residuosity assumption which in turn depends on the computational difficulty in integer factorization. So to provide 128-bit security, 'n' should be of 2048 bits, which also results in huge data expansion.

For the proposed encryption scheme, the field size, q required to provide 128-bit security is only 257. Even though $q = 257$ is sufficient for providing 128-bit security, the computational complexity of proposed scheme is compared using $q = 4091$ since this value is used for encrypted domain image fusion. The computational complexity of encryption and decryption of a message block of size, $p = 16$ using Paillier and proposed encryption scheme for providing 128-bit security is compared in Table 6. Since $q = 4091$, each element of message block can be represented using 12 bits. Message block of size $p = 16$ can be considered as a single message consisting of 192 bits in Paillier encryption scheme since message, $m \in Z_n$, where $n = 2048$. The encryption process and decryption process in Paillier scheme requires '1' exponentiation and '1' modular multiplication operations. Each exponentiation needs $2 \cdot (log_2 y)$ modular multiplications, where 'y' represents the exponent.

Table 6 Comparison of the computational complexity of Paillier and proposed encryption scheme for 128-bit security

Operation	Encryption		Decryption	
	Paillier	Proposed	Paillier	Proposed
No. of bit multiplications	2^{34}	2^{16}	2^{34}	2^{16}
No. of bit additions	0	2^{12}	0	2^{12}

In Paillier scheme, $y \in Z_n$ which implies $(log_2 y) = 2048$. So each exponentiation operation is equivalent to 4096 modular multiplications. The bit complexity of modular multiplication operation is $O((log_2 n)^2)$, where n is modulus value. In Paillier scheme, since modulus is taken with respect to n^2, the number of bit multiplications involved for single modular multiplication is $(2049)^2$. So the computational complexity of encryption and decryption for this scheme is $O(4097.(2049)^2)$. In the proposed scheme, modular operations are done with respect to q, where q is of 12 bits since $q = 4091$. Therefore, the number of bit multiplications is $12^2.p^2$ and the number of bit additions is $12.p^2$, considering the bit complexity of modular addition operation is $O(log_2 n)$. So the computational complexity of encryption and decryption for the proposed scheme is $O(12^2.16^2)$ bit multiplications and $O(12.16^2)$ bit additions. Thus computational complexity of the proposed encryption scheme is lesser compared to Paillier cryptosystem for comparable security levels; Also the storage overhead is double for Paillier encryption whereas there is no storage overhead for our proposed encryption scheme for comparable security which is an important advantage in cloud storage scenario.

7 Conclusion

In this paper, an affine Hill cipher based additive homomorphic encryption scheme is proposed to support encrypted domain image fusion over cloud. The security of the affine Hill cipher is enhanced through the addition of random vectors. We designed an algorithm for generating random vectors using a combination of LFSRs and verified that these vectors preserve the randomness and security properties while homomorphically combining the encrypted image blocks during fusion. Through mathematical analysis, it is established that the proposed scheme resists possible ciphertext only attack and known-plaintext attack at the cloud side. The security of the proposed scheme is also analyzed in terms of resistance against statistical attacks. The performance of the encrypted domain (ED) medical image fusion is analyzed in terms of various non-reference and reference based objective metrics. The reference based metrics: NCC, SC, SSIM values are equal to 1 and IQI values are 0.999; and MD, AD, NAE and MSE values varies from 10^{-13} to 10^{-28}. The closeness of the values of non-reference metrics: H, SD, MI and FMI corresponding to PD and ED results show that the proposed encrypted domain image fusion provides same accuracy levels as that of plaintext domain image fusion. Moreover, the proposed homomorphic encryption scheme does not introduce any storage overhead due to ciphertext expansion and it offers very low computational complexity. These desirable features make this scheme a suitable candidate for privacy preserving cloud storage and computing applications. Although our proposed homomorphic encryption scheme gives good performance in encrypted domain image fusion, it can support only block-level encryption operations. In our future work, we will focus on pixel level homomorphic encryption so that DWT can also be computed in encrypted domain, which helps to further reduce the client computational complexity.

References

1. Acharya B, Rath GS, Patra SK, Panigrahy SK (2007) Novel methods of generating self-invertible matrix for hill cipher algorithm. Int J of Secur 1(1):14–21
2. Bashir R, Junejo R, Qadri NN, Fleury M, Qadri MY (2019) Swt and pca image fusion methods for multi-modal imagery. Multimed Tools and Appl 78(2):1235–1263
3. Blum RS, Liu Z (2005) Multi-sensor image fusion and its applications. CRC Press, Boca Raton

4. Brakerski Z, Vaikuntanathan V (2011) Fully homomorphic encryption from ring-lwe and security for key dependent messages. In: Annual Cryptology Conference. Springer, Santa Barbara, pp 505–524
5. Brakerski Z, Vaikuntanathan V (2014) Efficient fully homomorphic encryption from (standard) lwe. SIAM J on Comput 43(2):831–871
6. Chan AF (2009) Symmetric-key homomorphic encryption for encrypted data processing. In: IEEE international conference on communication (ICC). IEEE, Dresden, pp 1-5
7. Du J, Li W, Lu K, Xiao B (2016) An overview of multi-modal medical image fusion. Neurocomput 215:3–20
8. Elhoseny M, Ramírez-González G, Abu-Elnasr OM, Shawkat SA, Arunkumar N, Farouk A (2018a) Secure medical data transmission model for iot-based healthcare systems. IEEE Access 6:20596–20608
9. Elhoseny M, Shankar K, Lakshmanaprabu S, Maseleno A, Arunkumar N (2018b) Hybrid optimization with cryptography encryption for medical image security in internet of things. Neural Comput and Appl, 1–15. https://doi.org/10.1007/s00521-018-3801-x
10. Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: 41st ACM symp. on the theory of comput. (STOC), Bethesda, Maryland, vol 9, pp 169-178
11. Gentry C, Halevi S (2011) Implementing gentry's fully-homomorphic encryption scheme. In: EUROCRYPT, Springer, Tallinn, Estonia, vol 6632, pp 129-148
12. Goldwasser S, Micali S (1984) Probabilistic encryption. J of computer and syst sci 28(2):270–299
13. Haghighat MBA, Aghagolzadeh A, Seyedarabi H (2011) A non-reference image fusion metric based on mutual information of image features. Comput Electr Eng 37(5):744–756
14. Haridas D, Venkataraman S, Varadan G (2012) Strengthened iterated hill cipher for encrypted processing. In: IEEE international conference on parallel distrib. and grid comput. (PDGC). IEEE, Solan, pp 491-496
15. Hodges JH (1958) The matrix equation x 2-i = 0 over a finite field. Am Math Monthly 65(7):518–520
16. James AP, Dasarathy BV (2014) Medical image fusion: a survey of the state of the art. Inf Fusion 19:4–19
17. Kadhe S, Sprintson A (2014) Weakly secure regenerating codes for distributed storage. In: IEEE international Symp. on netw. Coding (netcod), Aalborg Oest, Denmark, pp 1-6
18. MacWilliams FJ (1971) Orthogonal circulant matrices over finite fields, and how to find them. J of Comb Theory Series A 10(1):1–17
19. Menezes AJ, Van Oorschot PC, Vanstone SA (1996) Handbook of applied cryptography. CRC Press, Boca Raton
20. Mishra DC, Sharma RK, Ranjan R, Hanmandlu M (2015) Security of rgb image data by affine hill cipher over sln (fq) and mn (fq) domains with arnold transform. Optik-Int J for Light and Electron Optics 126(23):3812–3822
21. Mohanty M, Asghar MR, Russello G (2016a) 2dcrypt : Image scaling and cropping in encrypted domains. IEEE Trans Inf Forensics Secur 11(11):2542–2555
22. Mohanty M, Ooi WT, Atrey PK (2016b) Secret sharing approach for securing cloud-based preclassification volume ray-casting. Multimed Tools Appl 75(11):6207–6235
23. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT, Springer, vol 99, pp 223-238
24. Pawar S, El Rouayheb S, Ramchandran K (2011) Securing dynamic distributed storage systems against eavesdropping and adversarial attacks. IEEE Trans Inf Theory 57(10):6734–6753
25. Priya S, Varatharajan R, Manogaran G, Sundarasekar R, Kumar PM (2018) Paillier homomorphic cryptosystem with poker shuffling transformation based water marking method for the secured transmission of digital medical images. Personal and Ubiquitous Comput 22(5-6):1141–1151
26. Rivest RL, Adleman L, Dertouzos ML (1978a) On data banks and privacy homomorphisms. Foundations of secur comput 4(11):169–180
27. Rivest RL, Shamir A, Adleman L (1978b) A method for obtaining digital signatures and public-key cryptosystems. Commun of the ACM 21(2):120–126
28. Rosenthal A, Mork P, Li MH, Stanford J, Koester D, Reynolds P (2010) Cloud computing: a new business paradigm for biomedical information sharing. J of Biomed Inf 43(2):342–353
29. Shah NB, Rashmi KV, Kumar PV (2011) Information-theoretically secure regenerating codes for distributed storage. In: IEEE Global commun. conference (GLOBECOM), Houston, Texas, USA, pp 1-5
30. Shamir A (1979) How to share a secret. Commun of the ACM 22(11):612–613
31. Shehab A, Elhoseny M, Muhammad K, Sangaiah AK, Yang P, Huang H, Hou G (2018) Secure and robust fragile watermarking scheme for medical images. IEEE Access 6:10269–10278
32. Shi H, Liu D, Lu H, Zhou C (2017) A homomorphic encrypted reversible information hiding scheme for integrity authentication and piracy tracing. Multimed Tool Appl 77(16):20535–20567
33. Shim KA, Park CM (2015) A secure data aggregation scheme based on appropriate cryptographic primitives in heterogeneous wireless sensor networks. IEEE Tran Parallel Distrib Syst 26(8):2128–2139
34. Singh P, Raman B, Misra M (2017) Just process me, without knowing me: a secure encrypted domain processing based on shamir secret sharing and pob number system. Multimed Tool Appl 77(10):2581–12605

35. Stinson DR (2005) Cryptography: theory and practice. CRC Press, Boca Raton
36. Thakur S, Singh AK, Ghrera SP, Elhoseny M (2018) Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications. Multimed Tool Appl 78(3):3457–3470
37. Toorani M, Falahati A (2009) A secure variant of the hill cipher. In: IEEE symp. on computers and commun. (ISCC). IEEE, Sousse, pp 313-316
38. Vizár D, Vaudenay S (2015) Cryptanalysis of chosen symmetric homomorphic schemes. Stud Sci Math Hung 52(2):288–306
39. Wang Z, Bovik A (2002) A universal image quality index. IEEE Signal Process Lett 9(3):81–84
40. Wu Y, Noonan JP, Agaian S (2011) Npcr and uaci randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J of Sel Areas in Telecommun (JSAT) 1(2):31–38

Lakshmi V. S. Received B. Tech Degree in Electronics and Communication Engineering from Govt. Engineering College Barton Hill, Trivandrum (Kerala University) in 2004, M. Tech Degree in Digital Systems and Communication from National Institute of Technology Calicut in 2008. Currently doing Research at National Institute of Technology Calicut in the field of Secure Coding for Distributed Storage Applications. She has been working as Faculty in the Department of Electronics and Communication Engineering, Sree Chitra Thirunal College of Engineering, Trivandrum from 2008 onwards. Her current interests include Cryptography, Signal Processing in Encrypted domain, Error Control Coding, and Secure Distributed Storage.

Deepthi P. P. Received B. Tech Degree in Electronics and Communication Engineering from N.S.S. College of Engineering, Palakkad (Calicut University) in 1991, M. Tech Degree in Instrumentation from Indian Institute of Science, Bangalore in 1997 and Ph.D. from National Institute of Technology Calicut in 2009 in the field of Secure Communication. She has been working as Faculty in institutions under IHRD, Thiruvanthapuram from 1992 to 2001 and in the Department of Electronics and Communication Engineering, National Institute of Technology Calicut from 2001 onwards. Her current interests include Cryptography, Information Theory and Coding Theory, and Multimedia Security and Secure Signal Processing.