ALGEBRAS GROUPS AND GEOMETRIES 35 395 - 411 (2018)
DOI 10.29083/AGG.35.04.2018

ISONUMBERS AND RGB IMAGE ENCRYPTION

Mamadou I. Wade and Tepper L. Gill
EECS Howard University, Washington, DC 20059
mamadou.wade@howard.edu
tgill@howard.edu

Received May 12, 2020
Revised June 1, 2020

Abstract

In this paper we first discuss isonumber theory, which was developed for applications in physics by R. M. Santilli. We then use isonumbers of the first kind and the Paillier cryptographic system to add an additional security degree of freedom to the image encryption scheme, which we call the Paillier-Senegalese encryption process.

Following standard procedures we first separate a given RGB image into its constituent channel images, use public encryption keys in conjunction with the Paillier-Senegalese encryption function, to encrypt each of the R, G, and B-channel pixel intensity values. The encrypted channel images can be then combined and compressed if necessary before transmission through a possibly unsecured communication channel. The transmitted encrypted image is subsequently recovered by a decryption process which uses the Paillier-Senegalese decryption function in conjunction with the private decryption keys.

We are conducting a number of the performance and security analyses on the recovered and encrypted images in order to study and verify their robustness relative to all desirable security metrics. The complete results of our study will be reported later, but preliminary evidence shows that, the use of isonumbers provide an additional security degree of freedom for the production of encrypted images.

1 Introduction

A number of image encryption schemes have been proposed in the past. Image encryption schemes based on chaos in dynamical systems have been proposed in [3], [15] and [19]. Pareek et al. [19], have developed an encryption scheme which uses a 80-bit encryption key to initialize two chaotic maps. G. Ye et al. [13], have proposed an encryption scheme based on electrocardiography and autoblocking to generate the initial keys, which removes the need for manual assignment. Singh et al. [16] have developed an Elliptic Curve Cryptosystem, which one applies to a group of pixels to obtain a corresponding cipher-image. In another direction, P. P. Dang and P. M. Chau [20], have discussed an encryption scheme which uses the Discrete Wavelet Transform (DWT) for image compression and the block cipher Data Encryption Standard (DES) to produce a secure image. Image encryption using Multi-orders of the Fractional Fourier Transforms has been proposed by R. Tao, X. Meng, and Y. Wang [21]. In this case, the summation of different orders of the inverse discrete Fractional Fourier transform (FRFT) is applied to the interpolated sub-images to create the encrypted image.

This paper explores the application of isonumber theory to red green blue (RGB) image encryption.

Purpose

In this paper we introduce the Paillier-Senegalese image encryption process using the Paillier cryptographic scheme and Santilli's Isonumber Theory of the first kind (see [10]). The isonumber theory replaces the unit "one" of normal arithmetic by a new unit called an "isounit", which provides an additional degree of freedom for the encryption process. We apply our approach to the encryption and decryption of images in the visible range of the electromagnetic spectrum.

The introduction of an isounit may be seen as a shared private encryption and decryption key. We also note that our scheme possesses both public-key and private-key characteristics, so it can be seen as a hybrid public-key and private-key system. The major contribution of this paper is the introduction of an isounit as a private key and the development of this hybrid public-key and private-key cryptographic system, allowing the improvement of security by increasing the number of decryption keys required for an attacker to gain access to the transmitted images.

Content

The paper is organized as follows: In Section II, we introduce the Santilli's isonumbers theory of the first kind. In Section III, we combine the isonumbers with the Paillier cryptographic scheme to produce Paillier-Senegalese image cryptographic scheme. In Section IV we propose methods that will allow us to analyze the performance and security for our image encryption approach. Section V is devoted to our conclusion.

2 Santilli's Theory of Isonumbers

In this section, we provide a brief introduction to Santilli's Theory of Isonumbers. The origin of isonumbers can be traced at least back to 1978, when Santilli published the second volume of his fundamental work on the "Foundations of Theoretical Mechanics". He was studying the more realistic problem of local physical systems with dissipative forces as opposed to the mathematically beautiful but physically incorrect case of local systems without these forces. As first noted by Santilli, all of the nice mathematics and geometry developed in the twenth century and used in physics to study elementary particles fails. The failure follows because the special theory of relativity does not apply for studies inside the proton (or inside a star), where particles move in dissipative media (and not in a vacuum).

While looking for a way to extend the ideal mathematics, making it more useful for physics, he discovered that all mathematical structures could be preserved by replacing the normal unit by a strictly positive definite unit, which need not be commutative and may depend on any number of physical variables, change with time and have as many derivatives as needed to reproduce experimental reality. This new unit he called an "isotope" (see [5], [7] and [8]). The simplest case is that of the real and complex numbers, where the unit is the number one. Santilli began his study of this case with his seminal paper on "isonumbers" (see [5] and also [9]).

Our interest is limited to isonumbers. To make the transition easy, we consider the simplest example, which was used by Santilli to represent antiparticles as particles moving backwards in time, providing a rigorous foundation for an idea proposed by Feynman.
The real numbers is a field, which may be represented by a five-tuple $(\mathbb{R}, +, 0, \cdot, 1)$. Where zero is the unit for addition and one is the unit for multiplication. Santilli noticed that mathematicians implicitly use a sense of direction from left to right for $\mathbb{R}$, so that we should write it as a six-tuple $(\mathbb{R}, +, 0, \cdot, 1, \rightarrow)$. It now follows that, in practice, our real number system is not symmetric. Santilli's solution was to propose a new representation of the real numbers $(\hat{\mathbb{R}}, +, 0, *, \hat{1}, \leftarrow)$, where the unit for addition is unchanged, but the new unit for multiplication is now $\hat{1} = -1$. Thus,

$$\hat{a} * \hat{b} = (-a)(-1)(-b) = -ab = \widehat{ab}. \tag{2.1}$$

Santilli called $(\hat{\mathbb{R}}, +, 0, *, \hat{1}, \leftarrow)$ the isotopic dual representation of the real numbers. This is an example of Santilli's Isonumber Theory of the Second Kind, where the new unit, called isounit, is in the field (i.e. $(-1) = \hat{I} \in \mathbb{R}$). We are interested in Santilli's Isonumber Theory of the First Kind, where the new isounit $\hat{I}$ is not in the field. Let the field $F(a, +, .)$ be given (e.g., Z_p, with p prime) and let the isounit $\hat{I} = \hat{T}^{-1} \notin F$ be an invertible quantity. Let a new definition of multiplication be defined on F using $* = \hat{T}$ (see [10]). This allows us to define a new field $\hat{F}(\hat{a}, +, *)$ (called isofield of the first kind), with elements called isonumbers and rules given by:

$$\begin{gathered} \hat{a} = a\hat{I}, \quad \hat{a} * \hat{b} = (a\hat{I})\hat{T}(b\hat{I}) = ab\hat{I} = \widehat{ab} \\ \hat{a} + \hat{b} = (a\hat{I}) + (b\hat{I}) = (a+b)\hat{I} = \widehat{a+b}. \\ \hat{0} = 0\hat{I} = 0 \end{gathered} \tag{2.2}$$

3 The Paillier-Senegalese Algorithm

The construction of the Paillier-Senegalese cryptographic algorithm is accomplished in a three step process. The first step is the generation of the Paillier-Senegalese public and private isokeys required for encryption and decryption. The second step is the definition and use of the Paillier-Senegalese encryption function with the public isokeys to encrypt each of the pixel intensity value from a given set of R, G, and B-Channel images. The third step is the definition and use of the Paillier Senegalese decryption function with the private isokeys for the decryption phase, recovering the corresponding RGB images.

Let a RGB image be given with an 8-bit encoding for each channel. This means that the pixel intensity values are in the range $[0, (L-1)]$, where is $L = 256$. These pixel intensity values for each channel image can be assumed to belong to the finite prime field $Z_p = \{0, 1, 2, \cdots (p-1)\}$ of order $p = 257$ representing the closest prime number after $L = 256$,

The key generation, encryption and decryption phases are as follows:

3.1 Public and Private Keys Generation

We begin with the use of the Paillier cryptographic system for generating the public and private keys, which we use to generate our Paillier-Senegalese cryptographic system public and private isokeys.

3.1.1 Traditional Paillier public and private keys generation

We randomly choose two large prime numbers q and s, so that the greatest common divisor (gcd) of qs and $(q-1)(s-1)$ is 1, that is

$$gcd(qs, (q-1)(s-1)) = 1. \tag{3.1}$$

The condition is satisfied if q and s are primes with the same length. Our next step is to compute the public key N, and private key λ. This is done by defining $N = q \times s$ and, using the the least common multiple function lcm, to define λ by:

$$\lambda = lcm\left(q-1, s-1\right). \tag{3.2}$$

To compute the private decryption key μ, we generate a random integer $g \in Z^*_{N^2} = \{1, 2, \ldots, (N^2-1)\}$ such that the order l, of g is a multiple of N.

The value of g can be also computed as $g = 1 + N$ when p and s have the same length. The private decryption key μ is then defined using the modulo operator $\mod$, by:

$$\mu = \{[L(g^{\lambda} \mod (N^2))]^{-1}\} \mod (N), \tag{3.3}$$

where $L[U] = (U - 1)\, N^{-1}$. Hence, the privation key needed for decryption is (λ, μ), while the public key needed for encryption is (N, g).

3.1.2 Paillier-Senegalese Cryptographic System Public and Private keys Generation

The public and private isokeys for the Paillier-Senegalese cryptographic scheme are described in this section. First, we randomly generate two large prime numbers q and s, and randomly generate the isounit $\hat{I}$ outside Z_p. For proof of concept, we have randomly generated the 200 digit integer
$\hat{I} = 91446193624825680120842585289199005703827325309092129$
$498393744207361338371695567731788631147895697391520081713935620$
$650244629828082900156163404995075142190905435955078225770963O2$
2485699534833574025767.
We use it to compute the following parameters:

$$\hat{q} = q\hat{I} \tag{3.4}$$

$$\hat{s} = s\hat{I} \tag{3.5}$$

$$\hat{N} = \hat{q} * \hat{s} = q\hat{I}\left(\hat{I}^{-1}\right)s\hat{I} = (qs)\hat{I} = N\hat{I} \tag{3.6}$$

Given $\hat{N}$, the value of N can be computed as $N = \hat{N}/\hat{I}$. We can also compute $\hat{N^{\hat{2}}}$ needed for the encryption function using the isomultiplicative operation of the first kind $\hat{a}^{\hat{b}} = a^b\hat{I}$ as follows:

$$\hat{N}^{\hat{2}} = N^2\hat{I} \tag{3.7}$$

The values for $\hat{g}$ and $\hat{p}$ corresponding to g and p are given by

$$\hat{g} = \hat{I} + \hat{N} = \hat{I} + N\hat{I} = (1 + N)\hat{I} = g\hat{I} \tag{3.8}$$

$$\hat{p} = p\hat{I} \tag{3.9}$$

Note that given $\hat{g}$, the value of g can be computed as $g = \hat{g}/\hat{I}$. The function $L(U)$ becomes

$$\begin{aligned} L[\hat{U}] &= \left(\hat{U} - \hat{I}\right) * \hat{N}^{-1} = (U-1)\,\hat{I}(\hat{I}^{-1})\hat{I}N^{-1} \\ &= \left[(U-1)\,N^{-1}\right]\hat{I} = L[U]\hat{I}, \end{aligned} \tag{3.10}$$

The private decryption isokeys corresponding to μ and λ becomes

$$\hat{\mu} = \mu\hat{I} \tag{3.11}$$

$$\hat{\lambda} = lcm(\hat{q} - \hat{I}, \hat{s} - \hat{I}) \tag{3.12}$$

The private isokey $\hat{\lambda}$ is also given by

$$\hat{\lambda} = \lambda\hat{I} \tag{3.13}$$

When $\hat{\lambda}$ is known, λ can be obtained by $\lambda = \hat{\lambda}/\hat{I}$.
Hence, the pairs of public encryption isokey and private decryption isokey are $(\hat{g}, \hat{N})$ and $(\hat{\lambda}, \hat{\mu})$, respectively.

3.2 Paillier-Senegalese RGB Image Encryption Phase

This subsection introduces the traditional Paillier encryption scheme and the modification leading to the Paillier-Senegalese scheme for a given RGB image.
For the traditional Paillier encryption phase, let y_R, y_G, and y_B represent the pixel intensity values for the R, G, and B-channel images respectively. To encrypt each pixel intensity value in the Paillier scheme, one writes:

$$E(y_R) = g^{y_R} x_1^N \mod (N^2) \tag{3.14}$$

$$E(y_G) = g^{y_G} x_2^N \mod (N^2) \tag{3.15}$$

$$E(y_B) = g^{y_B} x_3^N \mod (N^2) \tag{3.16}$$

where $x_i, i = 1, 2, 3$ is a random number in $Z_N^* = \{1, 2, \ldots, (N-1)\}$. The encrypted values $E(y_R)$, $E(y_G)$, and $E(y_B)$ are outside of our range of $[0, (L-1)]$. To map them back into this range, we apply the $\mod [p]$ operation, where p is 257. These values belong to the encrypted channel images that can be transmitted or stored as described in [18], and can be computed as follows:

$$C_R = E(y_R) \mod (p) = \{[g^{y_R} x_1^N \mod (N^2)]\} \mod (p) \tag{3.17}$$

$$C_G = E(y_G) \mod (p) = \{[g^{y_G} x_2^N \mod (N^2)]\} \mod (p) \tag{3.18}$$

$$C_B = E(y_B) \mod (p) = \{[g^{y_B} x_3^N \mod (N^2)]\} \mod (p) \tag{3.19}$$

3.2.1 The Paillier-Senegalese Encryption Scheme

To obtain the Paillier-Senegalese encryption scheme, we replace the above equations by:

$$E(\hat{y}_R) = \hat{g}^{\hat{y}_R} * \hat{x}_1^{\hat{N}} mod(\hat{N}^{\hat{2}}) \tag{3.20}$$

$$E(\hat{y}_G) = \hat{g}^{\hat{y}_G} * \hat{x}_2^{\hat{N}} mod(\hat{N}^{\hat{2}}) \tag{3.21}$$

$$E(\hat{y}_B) = \hat{g}^{\hat{y}_B} * \hat{x}_3^{\hat{N}} mod(\hat{N}^{\hat{2}}) \tag{3.22}$$

where $\hat{x}_i = x_i \hat{I}, i = 1, 2, 3$ is a random isonumber in $\hat{Z}_N^*$, $\hat{g} = g\hat{I}$, etc. (see section 3.1).

Proposition 3.1. *Using the Paillier-Senegalese cryptographic system, we have*

$$E(\hat{y}) = E(y)\hat{I} \tag{3.23}$$

Proof. To encrypt y using Paillier encryption function, we can write

$$E(y) = g^y x^N \mod (N^2) \tag{3.24}$$

Let N^2 be a positive integer and $g^y x^N$ be a nonnegative integer. Using the division algorithm, we have

$$g^y x^N = qN^2 + E(y) \tag{3.25}$$

where $q = \lfloor \frac{g^y x^N}{N^2} \rfloor$, $0 \leq E(y) < N^2$,
where the symbol $\lfloor x \rfloor$ is the floor function which represents the largest integer less than or equal to x.
So using Eq. 3.25, Eq. 3.24 can be written as

$$E(y) = g^y x^N - \lfloor \frac{g^y x^N}{N^2} \rfloor \times N^2 \tag{3.26}$$

Multiplying both side of of Eq. 3.26 by $\hat{I}$ gives

$$E(y)\hat{I} = \left(g^y x^N - \lfloor \frac{g^y x^N}{N^2} \rfloor N^2 \right) \hat{I} \tag{3.27}$$

The right side of Eq. 3.23 can be written as

$$\begin{aligned} E(\hat{y}) &= \hat{g}^{(\hat{y})} * \hat{x}^{(\hat{N})} \quad \bmod \hat{N}^{\hat{2}} && (3.28a) \\ &= \left(g^{(y)} x^N\right) \hat{I} \quad \bmod (N^2 \hat{I}) && (3.28b) \\ &= \left(g^{(y)} x^N\right) \hat{I} - \lfloor \frac{g^y x^N \hat{I}}{N^2 \hat{I}} \rfloor (N^2 \hat{I}) && (3.28c) \\ &= \left[\left(g^{(y)} x^N\right) - \lfloor \frac{g^y x^N}{N^2} \rfloor N^2 \right] \hat{I} && (3.28d) \\ E(\hat{y}) &= E(y)\hat{I} && (3.28e) \end{aligned}$$

□

Using the result obtained from Eq. 3.28e, Eqs. 3.20 3.21, and 3.22 can be written as

$$E(\hat{y}_R) = E(y_R)\hat{I} \tag{3.29}$$

$$E(\hat{y}_G) = E(y_G)\hat{I} \tag{3.30}$$

$$E(\hat{y}_B) = E(y_B)\hat{I} \tag{3.31}$$

The computational cost of implementing Eqs. 3.29, 3.30, and 3.31 is less than the computational cost of Eqs. 3.20, 3.21, and 3.22.
Similar to the encryption of RGB images using the traditional Paillier encryption system as described on our previous papers [17], [18] the quantities

$E(\hat{y}_R)$, $E(\hat{y}_G)$, and $E(\hat{y}_B)$ are out of the range $[0, (L-1)]$. They must be mapped to this range by applying $\mod p$ to each as follows:

$$\hat{C}_R = E(\hat{y}_R) \mod (p) = [\hat{g}^{(\hat{y}_R)} * \hat{x}_1^{(\hat{N})} \mod (\hat{N})^{\hat{2}}] \mod (p) \tag{3.32}$$

$$\hat{C}_G = E(\hat{y}_G) \mod (p) = [\hat{g}^{(\hat{y}_G)} * \hat{x}_2^{(\hat{N})} \mod (\hat{N})^{\hat{2}}] \mod (p) \tag{3.33}$$

$$\hat{C}_B = E(\hat{y}_B) \mod (p) = [\hat{g}^{(\hat{y}_B)} * \hat{x}_3^{(\hat{N})} \mod (\hat{N})^{\hat{2}}] \mod (p) \tag{3.34}$$

The quantities $\hat{C}_R$, $\hat{C}_G$, and $\hat{C}_B$ represents the encrypted pixel intensity values that will be transmitted and/or stored.
Other parameter needed for the decryption are also computed as follows:

$$q_R = \left\lfloor \frac{E(\hat{y}_R)}{p} \right\rfloor \tag{3.35}$$

$$q_G = \left\lfloor \frac{E(\hat{y}_G)}{p} \right\rfloor \tag{3.36}$$

$$q_B = \left\lfloor \frac{E(\hat{y}_B)}{p} \right\rfloor \tag{3.37}$$

where the symbol $\lfloor \ \rfloor$ represents the floor function. The quantities given in Eqs. (3.35), (3.36), and (3.37) are not secret but can also be encrypted using other encryption methods to increase the security of the cipher images.

3.3 Paillier-Senegalese RGB Image Decryption Phase

This subsection presents the Traditional Paillier Cryptographic Decryption Approach and the Proposed Paillier-Senegalese Cryptographic Decryption Approach.

In our previous paper [17], the traditional Paillier Cryptographic scheme was used for the decryption of $E(y_R)$, $E(y_G,)$ and $E(y_B)$ to obtain y_R, y_G , and y_B.
In particular the encrypted pixel intensity values C_R, C_G, and C_B, and the quotients given by $qt_R = \left\lfloor \frac{E(y_R)}{p} \right\rfloor$, $qt_G = \left\lfloor \frac{E(y_G)}{p} \right\rfloor$, $qt_B = \left\lfloor \frac{E(y_B)}{p} \right\rfloor$ are

assumed available to the decryption function at the receiver side. To reconstruct $E(y_R)$, $E(y_G)$, and $E(y_B)$ before applying the Paillier decryption function, we can write.

$$E(y_R) = qt_R \times p + C_R = \alpha \tag{3.38}$$

$$E(y_G) = qt_G \times p + C_G = \beta \tag{3.39}$$

$$E(y_B) = qt_B \times p + C_B = \gamma \tag{3.40}$$

To recover the original pixels' intensity values y_R, y_G, and y_B, the Paillier decryption function is applied to $E(y_R) = \alpha$, $E(y_G) = \beta$, and $E(y_B) = \gamma$ in Eqs. (3.38), (3.39), and (3.40) as follows:

$$y_R = \frac{L[\alpha^\lambda \mod (N^2)]}{L[g^\lambda \mod (N^2)]} \mod (N) \tag{3.41}$$

$$y_G = \frac{L[\beta^\lambda \mod (N^2)]}{L[g^\lambda \mod (N^2)]} \mod (N) \tag{3.42}$$

$$y_B = \frac{L[\gamma^\lambda \mod (N^2)]}{L[g^\lambda \mod (N^2)]} \mod (N) \tag{3.43}$$

3.3.1 The Paillier-Senegalese Decryption Approach

Now, consider our proposed Paillier-Senegalese decryption method. Assume that the quantities q_R, q_G, and q_B are available to the decryption function in addition to $\hat{C}_R$, $\hat{C}_G$, and $\hat{C}_B$.

One must first reconstruct $E(\hat{y}_R)$, $E(\hat{y}_G)$, and $E(\hat{y}_B)$ from the receive cipher pixel values, $\hat{C}_R$, $\hat{C}_G$, and $\hat{C}_B$ before applying the decryption function as follows.

$$E(\hat{y}_R) = q_R \times p + \hat{C}_R = \hat{\alpha} \tag{3.44}$$

$$E(\hat{y}_G) = q_G \times p + \hat{C}_G = \hat{\beta} \tag{3.45}$$

$$E(\hat{y}_B) = q_B \times p + \hat{C}_B = \hat{\gamma} \tag{3.46}$$

The Paillier-Senegalese decryption function can be applied to $E(\hat{y}_R)$, $E(\hat{y}_G)$, and $E(\hat{y}_B)$, in Eqs. (3.44), (3.45), and (3.46) to recover the original pixels' intensity values $\hat{y}_R$, $\hat{y}_G$, and $\hat{y}_B$, and then y_R, y_G, and y_B belonging to our original channel images. They are computed as follows:

$$\hat{y}_R = \frac{L[\hat{\alpha}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})]}{L[\hat{g}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})]} \mod (\hat{N}) \tag{3.47}$$

$$\hat{y}_G = \frac{L[\hat{\beta}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})]}{L[\hat{g}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})]} \mod (\hat{N}) \tag{3.48}$$

$$\hat{y}_B = \frac{L[\hat{\gamma}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})]}{L[\hat{g}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})]} \mod (\hat{N}) \tag{3.49}$$

From equations 3.11 and 3.3, $L[\hat{g}^{\hat{\lambda}} \bmod (\hat{N}^{\hat{2}})] = \hat{\mu} = \mu\hat{I}$, so that these equations can be written as:

$$\hat{y}_R = \left\{L[\hat{\alpha}^{\hat{\lambda}} \bmod (\hat{N}^{\hat{2}})] * \hat{\mu}\right\} \bmod (\hat{N}) \tag{3.50}$$

$$\hat{y}_G = \left\{L[\hat{\beta}^{\hat{\lambda}} \bmod (\hat{N}^{\hat{2}})] * \hat{\mu}\right\} \bmod (\hat{N}) \tag{3.51}$$

$$\hat{y}_B = \left\{L[\hat{\gamma}^{\hat{\lambda}} \bmod (\hat{N}^{\hat{2}})] * \hat{\mu}\right\} \bmod (\hat{N}) \tag{3.52}$$

To find $L[\hat{\alpha}^{\hat{\lambda}} \mod (\hat{N}^{2})]$, let $\hat{U} = \hat{\alpha}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})$. The value of $\hat{U}$ can also be computed as $\hat{U} = (\alpha^{\lambda})\hat{I} \mod (\hat{N}^{2})$ and using Eq. 3.10 we have

$$L[\hat{\alpha}^{\hat{\lambda}} \mod (\hat{N}^{2})] = L(\hat{U}) = \frac{(\hat{U} - \hat{I})}{N} \tag{3.53}$$

Another method of computing $L[\hat{\alpha}^{\hat{\lambda}} mod(\hat{N}^{2})]$ is given by

$$L[\hat{\alpha}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})] = L(\hat{U}) = L(U)\hat{I} \tag{3.54}$$

where $L(U)$ is given in section 3.1.1 and $U = \alpha^{\lambda} mod N^{2}$, where $\alpha = \frac{\hat{\alpha}}{\hat{I}}$. Similarly, let $\hat{U_1} = \hat{g}^{\hat{\lambda}} \mod (\hat{N}^{2})$. The expression of $\hat{U_1}$ can also be written as $\hat{U_1} = (g^{\lambda})\hat{I} mod(\hat{N}^{2})$ and we can also write

$$L[\hat{g}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})] = L(\hat{U_1}) = \frac{(\hat{U_1} - \hat{I})}{N} \tag{3.55}$$

Furthermore, Eq. 3.55 is also given by

$$L[\hat{g}^{\hat{\lambda}} \mod (\hat{N}^{\hat{2}})] = L(\hat{U}_1) = L(U_1)\hat{I} \tag{3.56}$$

where $L(U_1)$ is given in section 3.1.1 and $U_1 = g^{\lambda} \mod (N^2)$.
Similar analysis from Eq. 3.50 through Eq. 3.56 can be also apply to further compute the values of $\hat{y}_G$ and $\hat{y}_B$ in Eqs. 3.48 and 3.49. Now, the values of y_R, y_G, and y_B are given by

$$y_R = \frac{\hat{y}_R}{\hat{I}}, \quad y_G = \frac{\hat{y}_G}{\hat{I}}, \quad y_B = \frac{\hat{y}_B}{\hat{I}} \tag{3.57}$$

For software implementation purposes, it is more efficient to use matrices of pixels' intensity values instead of individual pixels.

4 CONCLUSION

In this paper, we have introduced the Paillier-Senegalese cryptographic scheme which combines the Santilli's Isonumber Theory of the first kind with the Paillier Cryptographic System to provide an additional degree of freedom to the encryption-decryption of RGB images. This leads to a Hybrid public private-key cryptographic system with a substantial increase in image security, which also may be used in a variety of applications for secure systems that may not be related to image encryption and decryption.

References

[1] A. Soleymani, Md. J. Nordin, and Z. Md. Ali, *A Novel Public Key Image Encryption Based on Elliptic Curves over Prime Group Field,* Journal of Image and Graphics, Vol. 1, No. 1, March, 2013.

[2] A. Kanso, M. Ghebleh, *A novel image encryption algorithm based on a 3D chaotic map,* Commun Nonlinear Sci Numer Simulat 17 (2012) 2943–2959,
`www.elsevier.com/locate/cnsns`

[3] A. Daneshgar and B. Khadem, *A self-synchronized chaotic image encryption scheme,* Signal Processing: Image Communication 36 (2015) 106-114.
`www.elsevier.com/local/image`

[4] A. K. A. Hassan, *Reliable Implementation of Paillier Cryptosystem,* Iraqi Journal of Applied Physics, IJAP, Vol. 10, No. 4, October-December 2014, pp. 27-29

[5] R. M.Santilli-1, *Foundations of Theoretical Mechanics II,* Springer-Verlag New York, U.S.A, (1978), ISBN 0-387-08874-1.

[6] R. M.Santilli-2, *Isonumbers and Genonumbers of Dimensions 1, 2, 4, 8, their Isoduals and Pseudoduals, and "Hidden Numbers," of Dimension 3, 5, 6, 7,* Algebras, Groups and Geometries, Vol. 10, 273 (1993).

[7] R. M.Santilli-3, *Elements of Hadronic Mechanics I,* Ukraine Academy of Sciences, Kiev, (1995), Ukraine, ISBN 0-911767-68-1.

[8] R. M.Santilli-4, *Isorepresentation of the Lie-isotopic SU(2) Algebra with Application to Nuclear Physics and Local Realism,* Acta Applicandae Mathematicae Vol. 50, 177 (1998).

[9] C. Corda, *Introduction to Santilli's IsoNumbers,* AIP Conf. Proceed. 1479, 1013 (2012),

[10] C. X. Jiang, *Foundations of Santilli's Isonumber Theory*, Fundamental Open Problems in Science at the End of the Millennium, Proceeding of the Beijing Workshop, August 1997, Hadronic Press, Palm Harbor, FL 34682-1577, U.S.A, ISBN 1-57485-029-6, PP. 105-139, Edited by Tepper Gill (Co-author of this paper), K. Liu, and E. Trell.

[11] G. Zhang, and Q. Liu, *A novel image encryption method based on total shuffling scheme*, Optics Communications 284 (2011) 2775–2780, `www.elsevier.com/locate/optcom`

[12] G. Chen, Y. Mao, and C. K. Chui, *A symmetric image encryption scheme based on 3D chaotic cat maps*, Chaos, Solitons and Fractals 21 (2004) 749–761,
`www.elsevier.com/locate/chaos`

[13] G. Ye and X. Huang, *An Image Encryption Algorithm Based on Autoblocking and Electrocardiography*, Published by the IEEE Computer Society. April-June 2016.

[14] H. Liu, X. Wang, and A. kadir, *Image encryption using DNA complementary rule and chaotic maps*, Applied Soft Computing 12 (2012) 1457–1466,
`www.elsevier.com`

[15] H. S. Kwok, Wallace K. S. Tang, *A fast image encryption system based on chaotic maps with finite precision representation*, Chaos, Solitons and Fractals 32 (2007) 1518–1529,
`www.elsevier.com/locate/chaos`

[16] L. D. Singh and K. M. Singh, *Image Encryption using Elliptic Curve Cryptography*, Procedia Science 54 (2015) 475-481,
`www.sciencedirect.com`

[17] M. I. Wade, H. C. Ogworonjo, M. Gul, M. Ndoye, M. Chouikha, W. Paterson *Red Green Blue Image Encryption Based on Paillier Cryptographic Cryptographic Approach*, World Academy of Science, Engineering and Technology, International Journal of Electronics and Commu-

nication Engineering Vol:11, No:12, 2017,
https://waset.org/abstracts/79232

[18] M. I. Wade, *Distributed Image Encryption Based On a Homomorphic Cryptographic Approach*, 10th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, Columbia University, New York City, USA 10- 12 October 2019 (IEEE UEMCON 2019).

[19] N. K. Pareek, V. Patidar, and K. K. Sud, *Image Encryption Using Chaotic Logistic Map*, Image and Vision Computing 24 (2006) 926-934,
www.elsevier.com/locate/optlaseng

[20] P. P. Dang and P. M. Chau, *Image Encryption for Secure Internet Multimedia Applications*, IEEE Trans. On Consumer Electronics, Vol. 46. No. 3, August 2000.

[21] R. Tao, X. Meng, and Y. Wang, *Image Encryption With Multiorders of Fractional Fourier Transforms*, IEEE Trans. Inf. Forensics and Security, Vol. 5, No 4, Dec 2010.

[22] R. Rivest, Lecture Notes 15, Computer and Network Security: *Voting, Homomorphic Encryption*, October, 2002

[23] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.*,3rd ed. Person Education Inc., 2008.

[24] R. Rhouma, S. Meherzi, and S. Belghith, *OCML-based colour image encryption*, Chaos, Solitons and Fractals 40 (2009) 309–318,
www.elsevier.com/locate/chaos

[25] S. Mazloom and A. M. E-Moghadam, *Color image encryption based on Coupled Nonlinear Chaotic Map*, Chaos, Solitons and Fractals 42 (2009) 1745–1754,
www.elsevier.com/locate/chaos

[26] *University of Southern California, Signal and Image Processing Institute*,
http://sipi.usc.edu/database/

[27] Y. Zhou, L. Bao, C. L. P. Chen *A new 1D chaotic system for image encryption*, Signal Processing 97 (2014) 172–182, `www.elsevier.com/locate/sigpro`

[28] Yi. Xun, P. Russell, and B. Elisa, *Homomorphic Encryption and Applications*, 2014 XII, 126 p. 23 illus., `http://www.springer.com/978-3-319-12228-1`