Optics Communications xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Optics Communications

journal homepage: www.elsevier.com/locate/optcom

Analysis of phase encoding for optical encryption

David S. Monaghan [a,b,c], Guohai Situ [d], Unnikrishnan Gopinathan [d], Thomas J. Naughton [e,f], John T. Sheridan [a,b,c,*]

[a] School of Electrical, Electronic and Mechanical Engineering, College of Engineering, Mathematics and Physical Sciences, Ireland
[b] Optoelectronic Research Centre, University College Dublin, Ireland
[c] SFI-Strategic Research Cluster in Solar Energy Conversion, University College Dublin, Belfield, Dublin 4, Ireland
[d] Institut für Technische Optik, Universität Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany
[e] Department of Computer Science, National University of Ireland, Maynooth, Ireland
[f] University of Oulu, RFMedia Laboratory, Oulu Southern Institute, Vierimaantie 5, 84100 Ylivieska, Finland

ARTICLE INFO

Article history:
Received 25 April 2008
Accepted 9 October 2008
Available online xxxx

OCIS:
200.4740
100.2000
070.2580
000.4430

Keywords:
Optical processing
Digital image processing
Fourier optics
Numerical approximation and analysis

ABSTRACT

A phase encoded image is encrypted using the double random phase encoding technique, (DRPE). The effects of using a variable dynamic range of phase distribution during phase encoding (pre-encryption) are examined. We begin by phase encoding the input image using the full phase range, from $-\pi$ to π. We perform numerically perfect encryption and we then introduce errors into the decrypting phase-keys in the form of a pseudo-random distribution (position and phase) of incorrect pixels values. By quantifying the resulting error in the attempted decryptions, for increasing amounts of error in the decrypting phase-keys, we examine the effects of reducing the phase range to, $+/-(\pi - \Delta)$. In this way we attempt to improve the phase encoding procedure for use with the DRPE technique. When the pixel values calculated, during an attempted decryption, fall outside the phase range used to phase encode the assigned input image, we examine different methods of redistributing the value of that pixel to an assigned value within the allowed phase range. The effects of the phase quantisation used in the keys are also examined.

1. Introduction

Powerful desktop computers are becoming cheaper and more readily available with the emergence of high-speed, multi-core processors. This trend brings with it an increased demand for information security and a continuing search for stronger encryption algorithms. Two dimensional optical encryption, (or image encryption), has received much attention as an application of optical signal processing (OSP) [1–8]. The most apparent advantage of optical based encryption systems are due to the natural 2D imaging capabilities of optics and thus the inherent parallelism achievable with OSP. They also boast high-speed, but require sensitive digital techniques to recover the full wave-field information [9–12]. Another advantage of optical encryption is the potential for a large key-space [13], rendering brute-force methods of attack almost impossible using current digital technology. Several optical encryption techniques have been proposed [1] in the literature, however the one that has received the most attention, and which is of interest to us, is the random phase encoding (DRPE) technique [14].

Implementation of the DRPE technique involves the use of two random phase masks, one in the input-image domain and the other in the Fourier domain. If both phase masks are statistically independent white noises then the encrypted output image is also a stationary white noise [14]. In this study we will look at the DRPE technique when the 2D input information is carried on the phase front of the incident plane wave. In general it can be assumed that if you phase encode the image, input to the DRPE technique, the security of the system is improved [15]. To illustrate this point we note that some of the most widely studied techniques [16,17] for evaluating the security of DRPE exploit the redundancy of the first, image-plane, phase-key, since this phase-key is not necessary to the recovery of grayscale image intensity data. However, when using a phase encoded input it is necessary to know both the image-plane and the Fourier-plane phase-keys in order to access the encrypted image data [15].

Our objective is to study how the initial method of phase encoding the input image can affect the amount of error in the output

* Corresponding author. Address: SFI-Strategic Research Cluster in Solar Energy Conversion, University College Dublin, Belfield, Dublin 4, Ireland. Tel.: +353 (0) 1 716 1927; fax: +353 (0) 1 283 0921.
E-mail address: john.sheridan@ucd.ie (J.T. Sheridan).
URL: http://eleceng.ucd.ie (J.T. Sheridan).

0030-4018/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.optcom.2008.10.047

image, and determine if we can, in this way better understand and improve on the performance of the phase encoded DRPE technique. In a previous study on amplitude and phase encoding [15], the phase encoded input-image was encoded to the phase range [0 to +π] radians. Any so called '*noisy phase*' value, arising during an attempted decryption, that landed in the 3rd quadrant, i.e. (+π to +3π/2] rad, of the unit circle, would get assigned to the value π, and noisy phase value that landed in the 4th quadrant, i.e. (+3π/2 to +2π) rad, would get assigned to 0. In this paper we will examine the effects that a reduction in the phase range used has during phase encoding, when attempting to decrypt the phase encoded image, i.e. the quantity and type of the resulting noise in the decrypted image. We perform numerically flawless encryption and then pseudo-randomly alter the value of an increasing number of randomly positioned pixels in the decrypting phase-keys and note the associated amount of error in the decrypted data. Then we decrease the phase range used to phase encode the input-image and note the corresponding change in the error in the resulting decrypted image. Finally we look at three different methods of redistributing noisy phase and the effect of each on the error in an imperfectly decrypted image.

Important, in carrying out this study, is an understanding of the effects of the number of levels of phase quantisation used in the encryption keys. In order to examine these effects we vary the number of quantisation levels in the keys. We show that, for an image phase encoded and then encrypted using 256 phase levels, employing keys with more that 16 levels produce little significant effect in our results.

The paper is organised as follows in Section 2 we introduce the phase encoding technique used in conjunction with the DRPE algorithm. In Section 3 we present our error metric, the Normalised Root Mean Squared, (NRMS), error, which is used to quantify the difference between our decrypted image and our original data. In Section 4 we discuss the phase quantisation of the phase-keys and the effects, for different levels of noise, on the resulting NRMS error as a function of this quantisation. In Section 5 we then turn to the effects of the method of phase encoding used, (phase range, and re-assignment technique), prior to encryption. The NRMS is used to examine the phase encoding technique used and the effects of noise in the decryption keys. The form of the resulting imperfectly decrypted output image noise is also examined. Finally in Section 6 a brief conclusion, outlining results and future work, is presented.

2. Phase encoding and encryption

The primary test image used through out this study is the gray-scale Lena image [18]. It is a normalised 8-bit image with 256 gray levels. In the standard case of phase encoding, the input image is mapped to the phase range, i.e. where the normalised amplitude in the range [0, +1] is mapped to 256 discretely quantised levels in the phase range $[-\pi, +\pi)$. We phase encode the input-image, $f_A(x, y)$, as follows:

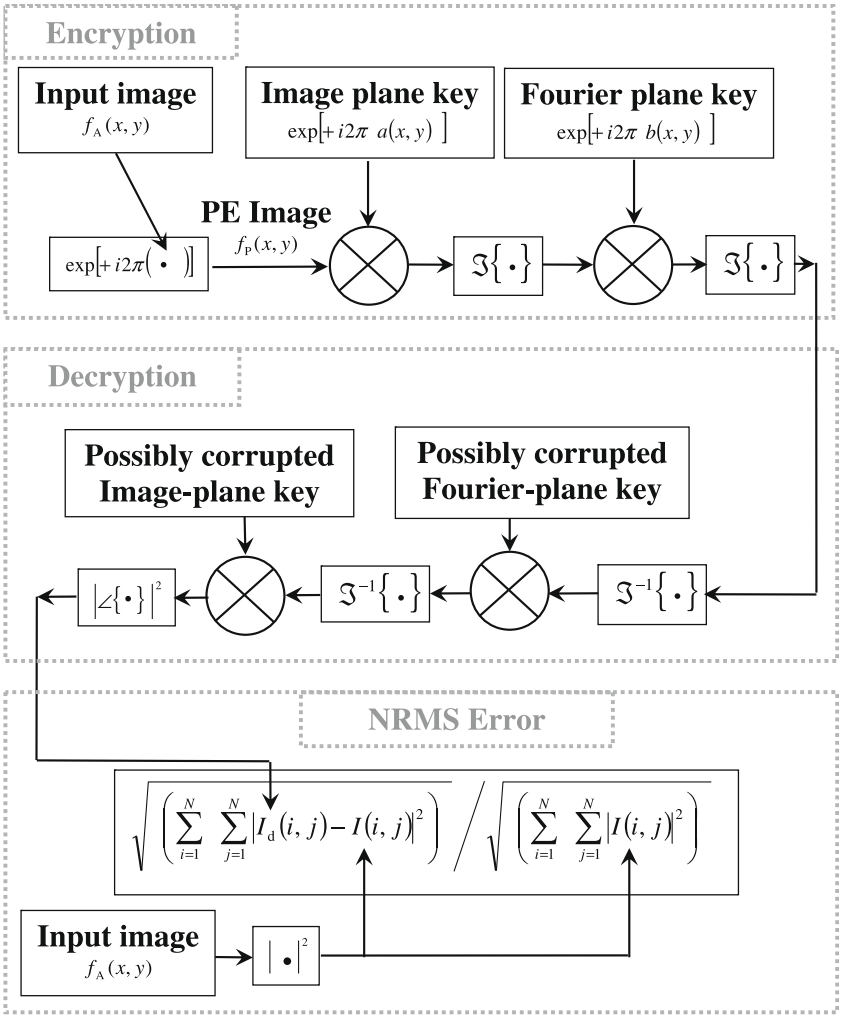

Fig. 1. A flow chart diagram of the encryption/decryption process and the NRMS calculation when using a phase encoded input image.

$$f_P(x,y) = \exp[+i2\pi f_A(x,y)], \quad (1)$$

where the subscript 'P' denotes that the input is a phase image, and where $-1/2 \leqslant f_A(x,y) \leqslant +1/2$. The phase-keys used, both the encryption and the decryption keys, are quantised to 16 phase levels, (see Section 4). The encryption and decryption processes are described in Eqs. (2) and (3) as follows:

$$\psi(x,y) = \Im\{\Im\{\exp[+i2\pi f_A(x,y)] \times \exp[+i2\pi a(x,y)]\} \times \exp[+i2\pi b(x,y)]\}, \quad (2)$$

and

$$|f_P(x,y)| = \left|Arg(\Im^{-1}\left\{\Im^{-1}\{\psi(x,y)\} \times \exp[-i2\pi b(x,y)]\right\} \times \exp[-i2\pi a(x,y)])\right|, \quad (3)$$

where $a(x, y)$ and $b(x, y)$ are the image-plane and Fourier-plane phase-keys and fall in the limits $0 \leqslant a \leqslant 1$ and $0 \leqslant b \leqslant 1$. The encryption and decryption processes are represented by a flow diagram in Fig. 1. We note that when the input-image is phase encoded its amplitude is uniform, which implies that each phase value lies on the unit circle in the complex plane, i.e. $|f_P(x,y)| = 1$.

3. Error metric: NRMS

In this paper the encryption/decryption process is performed numerically using a standard Matlab fast Fourier algorithm (FFT) [19]. Each pixel in the phase-keys and images is represented by a single complex value in a finite 2D array within the computer.

In doing so we neglect physical all modelling issues, e.g. polarisation and diffraction effects, Spatial Light Modulator (SLM) fill factor, etc. [12,20,21]. Such simplifying assumptions are justified because the aim of this study is to examine the nature of the DRPE technique, and as such we are not concerned with errors introduced due to the physical limitations of the optical system.

In order to calculate the Normalised Root Mean Squared (NRMS) error of the output phase image, following an attempted decryption, it is converted to a normalised amplitude image, where the entire output phase range, i.e. a range of [0, 2π), is mapped to the range [0, 1]. We note in this paper that no clipping or re-quantisation of this output image is performed prior to the calculation of the NRMS value. The resulting intensity at each pixel of this image is compared to the corresponding pixel value in the original normalised input image using the following equation:

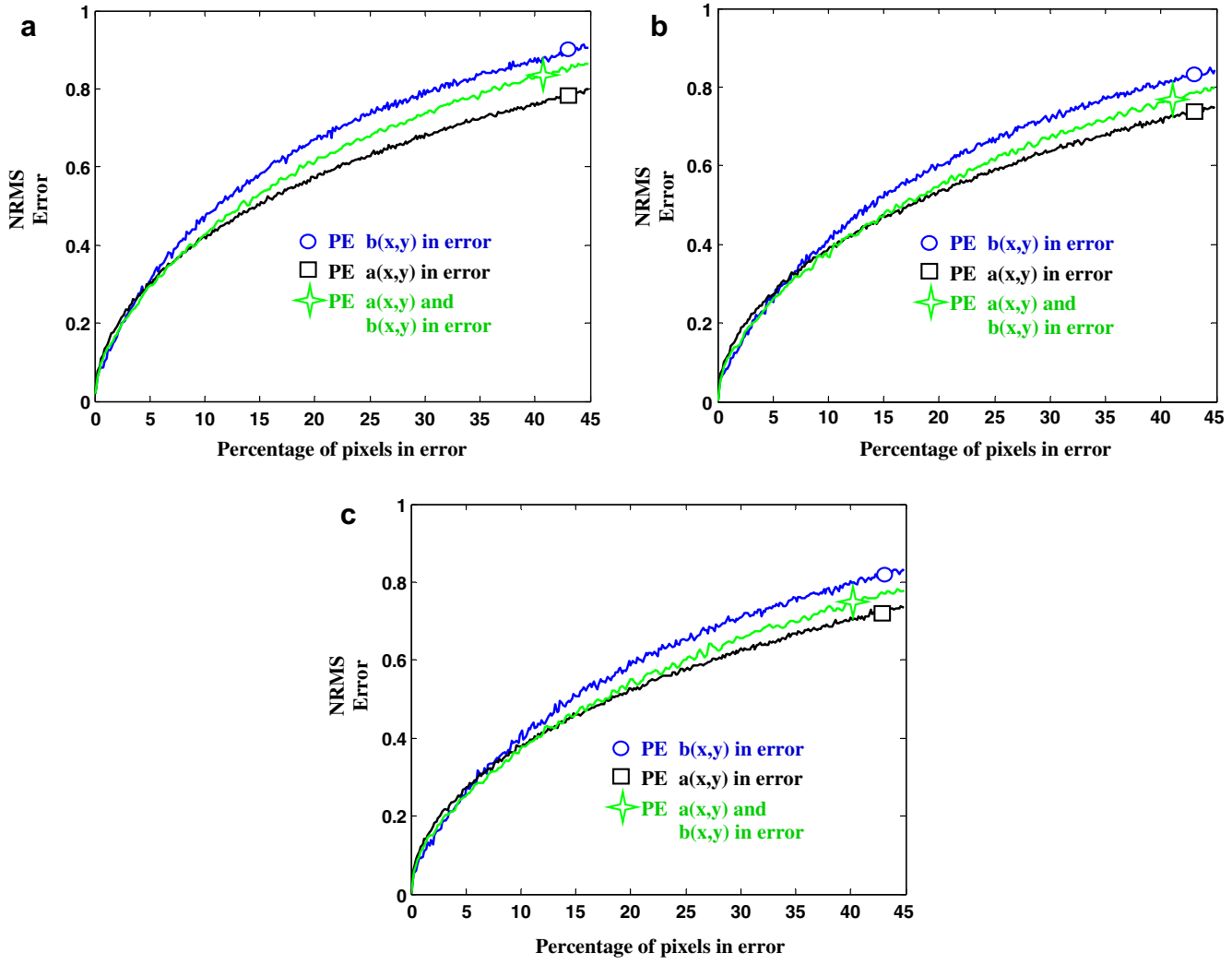

Fig. 2. (a) The three curves in each graph look at phase encoding for the cases when the phase-keys, $a(x, y)$ and $b(x, y)$, are in error separately and at the same time. 2(a) is for when both the encryption and the decryption keys are quantised to 4 phase levels, 2(b) is when they are quantised to 16 phase levels and in 2(c) they are both quantised to 64 phase levels.

$$\mathrm{NRMS} = \sqrt{\frac{\left(\sum_{i=1}^{N}\sum_{j=1}^{N}|I_d(i,j) - I(i,j)|^2\right)}{\left(\sum_{i=1}^{N}\sum_{j=1}^{N}|I(i,j)|^2\right)}}, \quad (4)$$

where $I_d(i,j)$ is the intensity of the (i,j) pixel in the decrypted image, and $I(i,j)$ is the corresponding pixel intensity in the original input image. The NRMS error is thus used to quantify the amount of error in an incorrectly decrypted image.

The NRMS error metric provides a direct measure of the Euclidean distances between intensity images. Other error metrics do exist which can be used to provide estimates of the effects of noise and key error [15]. Since it is the intensity of the decrypted data that is the quantity of primary interest, and since it is intensity values that are measured during experiments, we believe the NRMS provides useful insights. However it should be emphasised that all the conclusions presented in this paper, regarding the relative performance of the DRPE technique, have been arrived at based on the use of the NRMS metric.

4. Phase quantisation in the keys

Before examining the effects of the phase range used during phase encoding, we first study the effects of the number of quantised phase levels used in the encryption/decryption keys. Assuming that the normalised input image is phase encoded to 256 phase levels we now ask how many phase levels can meaningfully be applied during encryption and decryption in the phase-keys.

Ignoring physical system constraints, one might expect that the greater the number of phase levels available the greater the security of the encryption achieved, since any attempt at decryption will require that accurate information from a larger key-space is needed to correctly decrypt the encoded information. To test this hypothesis we simultaneously changed the number of phase levels used in all the phase-keys, during encryption and decryption, and attempted decryption with noise added to the decryption keys. While a large range of quantisation values were examined by us, in this paper we present results for three sample cases: (a) when 4 quantisation levels were used in both the image and Fourier-plane keys during both encryption and decryption, (b) when 16 levels were used, and (c) when 64 levels were used.

To examine this situation quantitatively using the NRMS we proceed as follows: Our normalised 256 graylevel Lena image is phase encoded to the phase range $[-\pi, +\pi)$, (later in Section 5 we refer to this as the $\Delta = 0$ case). Following perfect encryption, attempts are then made to decrypt the image using phase-keys containing random quantised phase perturbations at random positions. The results are presented in Fig. 2a–c.

In these figures the percentage of pixels that are in error are plotted along the horizontal-axis and the associated NRMS error is plotted along the vertical-axis. The three curves presented in each figure correspond: First to the case in which the Fourier-plane phase-key, $b(x,y)$, has been corrupted and the image-plane phase-key, $a(x,y)$, is assumed correct, (this curve is denoted in the graph

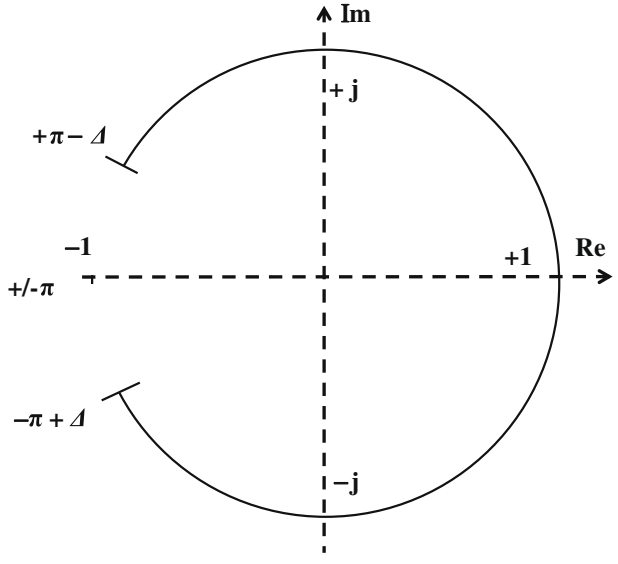

Fig. 3. The blanked out arc in this diagram is not used during phase encoding. If, following encryption and decryption, a pixel acquires a phase argument corresponding to this angle range (regardless of its radial magnitude), the pixel will be re-assigned a new value on the allowed part of the unit circle circumference by one of the three methods illustrated in Fig. 7.

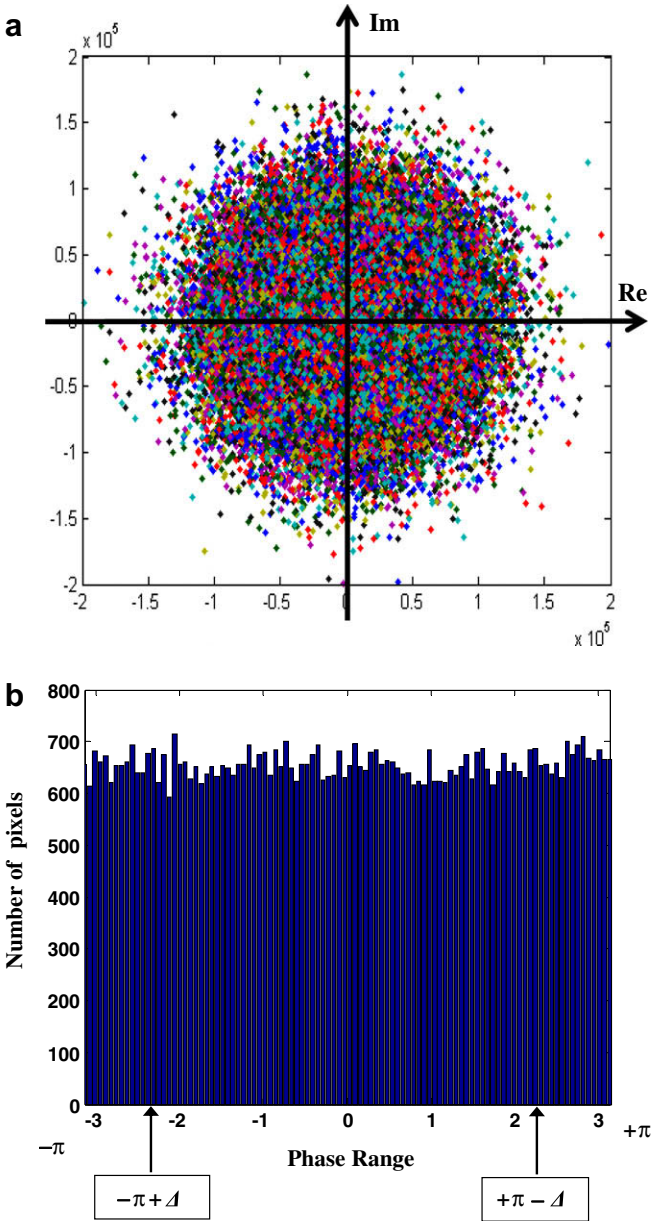

Fig. 4. (a) Shows the complex pixels values of an encrypted image for which $\Delta = \pi/4$ when phase encoding. An attacker, who acquired this image, would not be able to distinguish whether it had been encoded using the entire phase range or a reduced range. (b) Shows the corresponding pixel phase distribution.

by a circle). Second, when the image-plane phase-key has been corrupted and the Fourier-plane phase-key is assumed perfect, (denoted by the square), and lastly when both of the decrypting phase-keys have been corrupted, (the star). It should be noted that for the case when both of the decrypting phase-keys are corrupted that the percentage of pixels in error is taken as the sum of the percentage of pixels in error in both keys. For example, a point on the curve denoted by the star, which is at 20% pixels in error is for the case when 10% of the pixels are in error in both of the decrypting phase-keys. We note that in all cases the incorrect pixel values used during decryption were always quantised to the appropriate number of phase levels.

Several trends can be observed in these figures. First, it would appear that the decrypting Fourier-plane phase-key, $b(x, y)$, is more sensitive to incorrect pixel values as the NRMS error values tend to increase rapidly to a higher value than in the other two cases [22]. Secondly, comparing the NRMS error values for the three different key quantisation levels, i.e. 4, 16 and 64, we note that when the number of levels is increased beyond 16 levels there is very little change in the graphs, i.e. between Fig. 2b and c.

These calculations have been repeated for a number of different input images and also for different DRPE random keys. In all cases while the individual slope values and maximum values found differed slightly the trends observed were always identical. What this indicates is that increasing the number of phase levels above 16 does not significantly increase the security of the encryption process. Since there is no appreciable increase in the NRMS error metric for increasing numbers of phase levels an attacker, who does not know a priori the number of phase levels involved in encryption, can choose to assume 16 levels (4 bits per pixel) and will be able to attack the encrypted data confident that equivalent accuracy can be achieved as with a more numerically intense 64 level (6 bits per pixel) search.

For the interested reader we note that in a recent paper [22] we have examined the effects of decreasing the number of quantisation levels used in the decryption keys following a perfect encryption. The results suggested that it is advantageous, when attacking, to use an equal number of quantisation levels in both the image-plane phase-key, $a(x, y)$, and the Fourier-plane phase-key, $b(x, y)$. While these results are not theoretically definitive they are suffi-

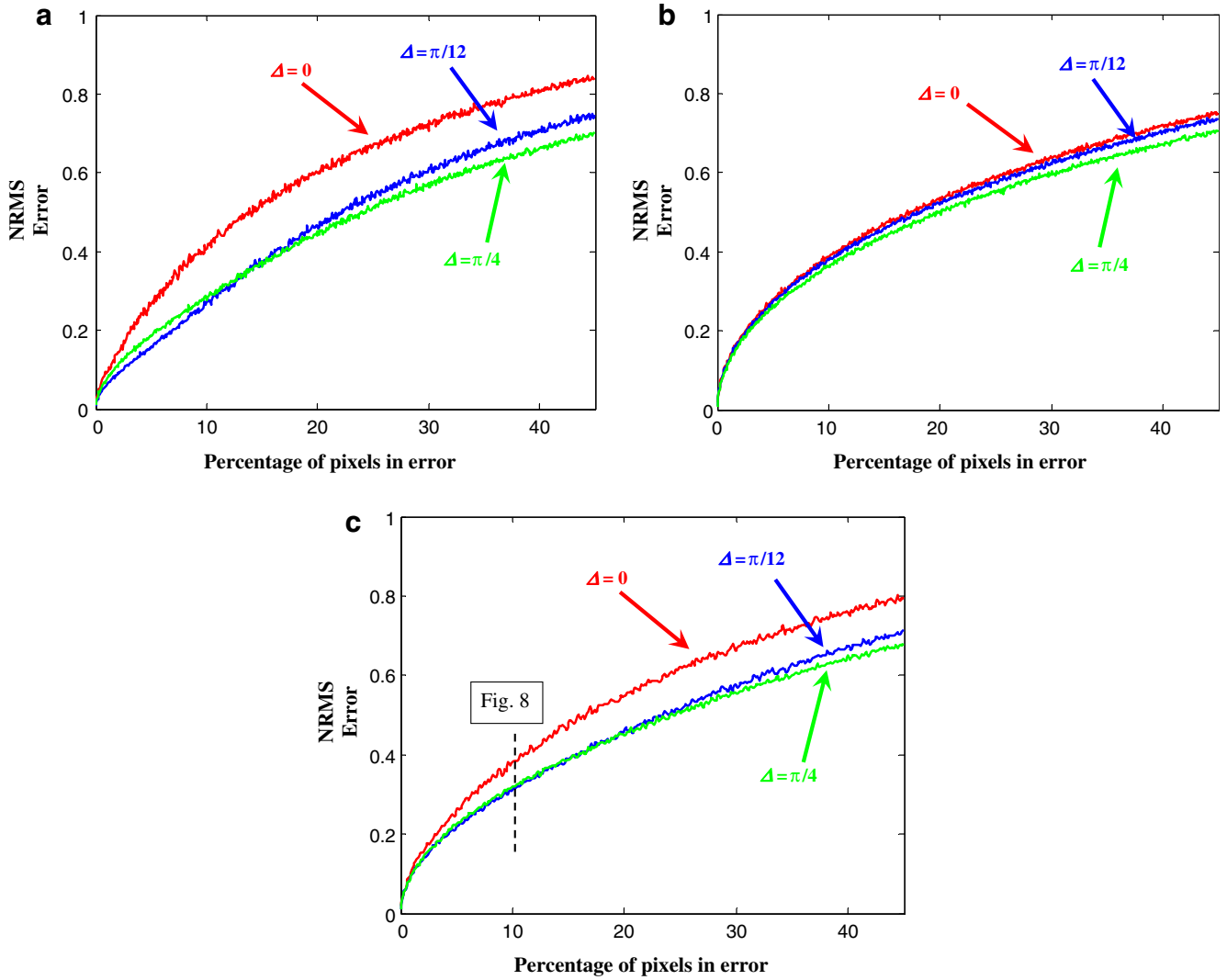

Fig. 5. These curves show the effect of variations in the size of the unused arc in phase encoding, shown in Fig. 3. To produce each curve, corresponding to different forbidden arc region, the simulation were run three times for the Lena input, with the same keys, but different noises for each run. In all cases reassignment of forbidden pixel values was performed using Method (iii). They differ as follows: (a) when only the decrypting Fourier phase-key, $b(x, y)$, is in error, (b) when only the decrypting image phase-key, $a(x, y)$ is in error, and (c) when both decrypting phase-keys are in error. The vertical dashed line in (c) corresponds to decrypted examples in Fig. 8.

Please cite this article in press as: D.S. Monaghan et al., Opt. Commun. (2008), doi:10.1016/j.optcom.2008.10.047

ciently numerically reproducible to be considered indicative for the cases studied.

5. Phase encoding results

In this section we study the effects of the phase encoding range used in conjunction with the DRPE technique. As in the previous section, we perform numerically perfect encryption and then we add increasing amounts of pseudo-random error to the two decrypting keys, in the image and the Fourier planes. The resulting NRMS values are then used to quantify the effects of the introduced errors.

In the simulation presented in Fig. 2b we have mapped the graylevel input image to 256 quantisation phase levels over the phase range $[-\pi, +\pi)$. We now reduce this phase encoding range symmetrically while maintaining the same number of phase-quantisation levels in the input image. Fig. 3 shows the unit circle where the blanked out arc indicates the part of the phase range that is not used in the phase encoding of the original input image. The input image is still phase encoded to 256 quantised levels but these are now spread between the new phase range limits.

Fig. 4 shows the complex pixels values for an encrypted image displayed in the complex plane. The associated input image was phase encoded using $\Delta = \pi/4$. Comparing phase encoded images encrypted with $\Delta = 0$ and $\Delta = \pi/4$, we note that there are no discernable differences that would indicate what value of Δ was used. Therefore an attacker, who acquired this encrypted image, would not be able to simply identify Δ. Furthermore, even if attackers are aware of the number of quantisation levels, i.e. 256, they will not be aware of the exact quantisation values.

If, after decryption using a corrupted decryption phase-key, the phase value at a pixel falls within the forbidden phase range, i.e. the range not used during phase encoding, we propose to map the value assigned to that pixel back to an allowed phase value. In general, following decryption with keys which are in error, the decrypted pixel values will no longer be at discrete quantised phase values. In applying all of our reassignment techniques we assume that during decryption it is known a priori what the quantisation values are and mapping takes place to these quantised levels.

We examine three different methods of re-assigning these new values. We are only interested in the phase information of the output image and this allows us to neglect any amplitude variations and map all the pixel values to unity whilst maintaining the phase angle.

In Fig. 5a–c we examine the effects on the calculated NRMS values, of the size of the forbidden phase region used for one method of reassignment. In all the cases examined, prior to encryption, the normalised input images are mapped to the phase ranges $[-\pi + \Delta, +\pi - \Delta]$, where Δ can have a value of $[0, \pi/12, \text{or } \pi/4]$ as indicated. During imperfect decryption pixels will acquire values within the corresponding forbidden phase region. We examine how the extent of this forbidden phase region affects the NRMS results when the decrypting phase-keys are corrupted. In Fig. 5a it is assumed that random errors only occur in the Fourier-plane key, $b(x, y)$. Fig. 5b is for the case when errors occur in the image-plane key, $a(x, y)$, and finally in Fig. 5c equal percentages of random phase errors are inserted into both keys simultaneously. In all cases, following our results in Section 4, 16 quantised phase levels are used in the encryption and decryption keys. Furthermore in all the cases examined in Fig. 5 the forbidden phase values are reassigned using an exponential probability function which we refer to as Method (iii). All three reassignment methods are discussed later in this section in detail, and are illustrated in Fig. 6. In all cases calculation of the NRMS error then takes place following re-assignment.

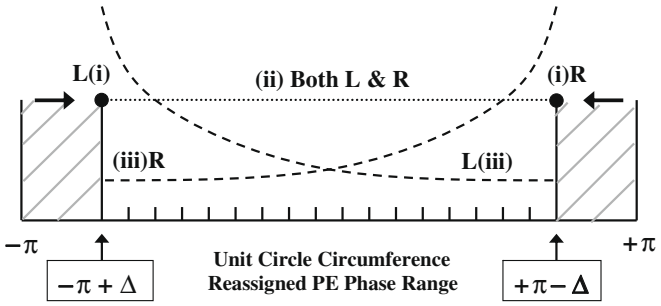

Fig. 6. The shaded region here relates to the forbidden phase range from Fig. 3. The values that are re-assigned from the left, (L) and right regions (R), are reassigned: Method (i) to the nearest limits, Method (ii) reassigned with uniform probability to a random quantisation value (dotted line), or using Method (iii) reassigned to a quantisation values based on an exponential probability distribution function (dashed line).

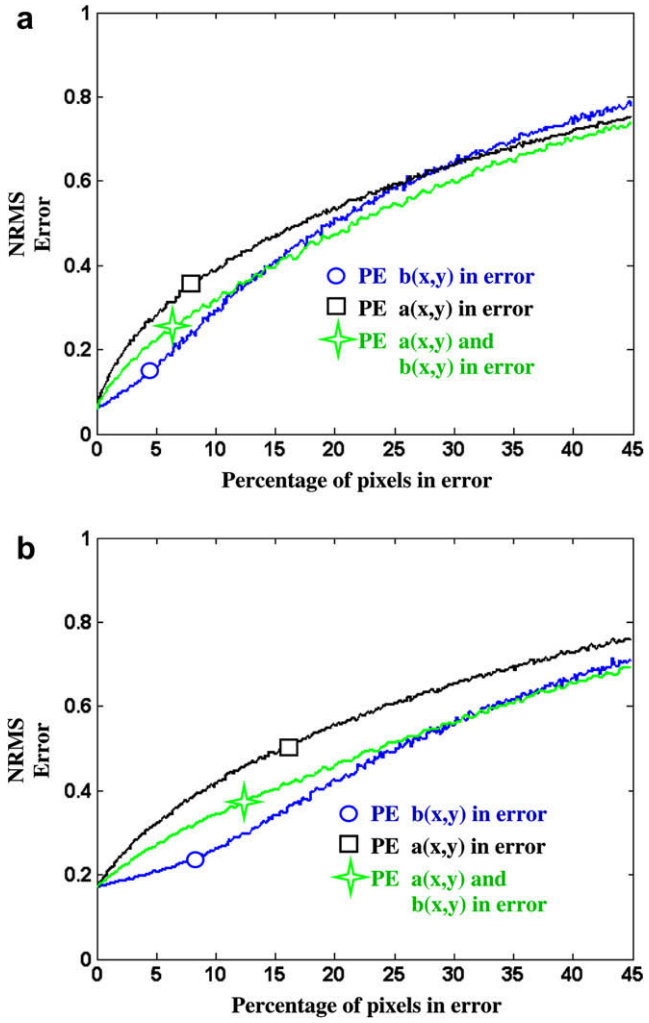

Fig. 7. Phase encoding is performed using Fig. 6a $\Delta = \pi/12$, and Fig. 6b $\Delta = \pi/4$. The decryption process is performed assuming $\Delta = 0$ in both (a) and (b). Note the non-zero NRMS value at 0%.

Fig. 7 is for the cases when phase encoding is performed using: (a) $\Delta = \pi/12$, and (b) $\Delta = \pi/4$. However in the decryption process we have assumed that $\Delta = 0$, (no reassignment method is used), which simulates the scenario of an attacker not being aware that a non-zero Δ value was used in the phase encoding process. The

results indicate that the NRMS error levels are higher for keys with small amounts of incorrect pixels, i.e. less than 10% pixels in error, which would make an attack using heuristics methods more difficult. Significantly, even if the correct keys are used but Δ is unknown, substantial error still occurs and for the cases examined this error increases as Δ increases.

The three methods of re-assigning phase values examined. Method (i) is the simplest method, and involved pixels with forbidden phase values, being reassigned to the nearest end points of the allowed phase range. These end points are labelled L(i) and (i)R in Fig. 6. Method (ii) involves forbidden pixel values being reassigned values with equal probability. (randomly to one of the known quantisation levels), on the allowed part of the circumference of the unit circle. This is denoted by the dotted horizontal line in Fig. 6. Method (iii), used above to produce Fig. 5, is also illustrated in Fig. 6. In this case the forbidden pixel value is assigned to a quantisation value based on a probability distribution function denoted by the dashed line. In this case the probability is assumed to decrease exponentially as the value is reassigned further away from its original value. In all cases assignment takes place with a total probability of unity.

Fig. 8 shows the resulting decrypted Lena images following an attempt to decrypt each of the three phase cases when 10% in total of the pixels (shared equally between both keys) are in error. This corresponds to the situation indicated by the dashed vertical line in Fig. 5c. In all cases re-assignment is performed using Method (i). In these images it can be seen that there are two distinct types of noise. One of these noises is in the form of shot noise referred to as 'salt and pepper' noise that is easily identified by its distinct black and white spots, and can be easily removed with median filtering [23]. The other is a Gaussian type noise and is more difficult to remove with post-processing [23]. Both of these types of noise

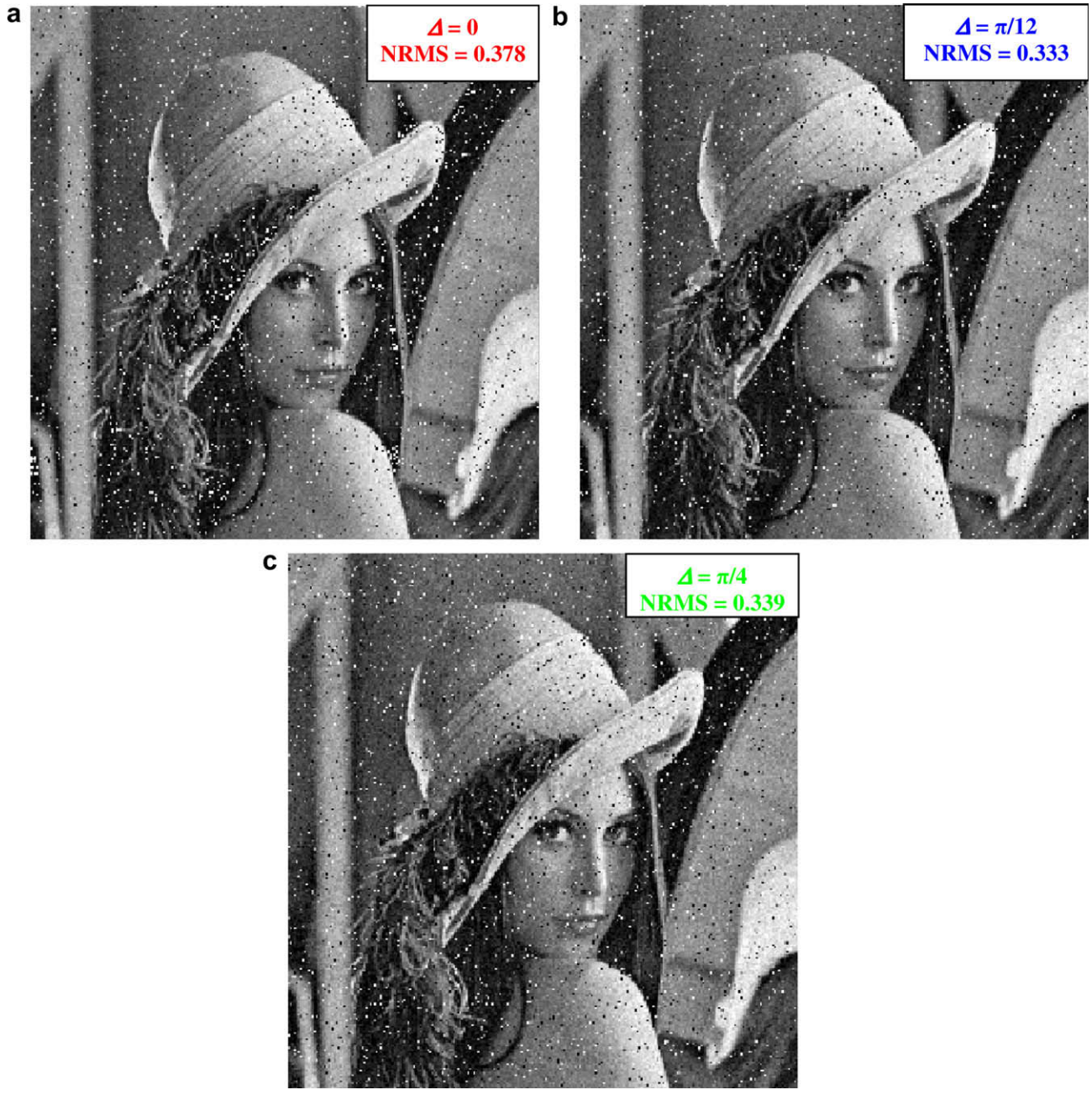

Fig. 8. With 10% of the pixels in error, as in indicated in Fig 5c. Noisy decrypted images for three cases: Δ = (a) 0, (b) $\pi/12$, and (c) $\pi/4$.

are present in Figs. 8a–c. Later we will examine cases when the two types of noise can be separately identified, for now we note that by appropriate post-processing the image quality in these cases can be improved.

In Fig. 9 we examine the effect on the NRMS of the three methods for re-assigning pixel values illustrated in Fig. 6. In all cases it is assumed that Δ and the number of quantisation levels are known during reassignment. Fig. 9a shows the three phase cases when pixels with values in the forbidden phase region are re-assigned to the value at the nearest limits, i.e. Method (i). Fig. 9b corresponds to the case when they are assigned randomly to an allowed quantised phase level, i.e. Method (ii), and Fig. 9c is the case when they are assigned to an allowed phase value with exponentially decreasing probability, i.e. Method (iii).

When incorrectly decrypted pixels acquire values that are in the forbidden phase region we do not know what their correct phase should be. Examining Fig. 9 it can be seen that re-assigning these pixels to values using Methods (ii) and (iii) will in general produce a lower NRMS error in the decrypted image than using Method (i). This difference is most marked when errors are introduced in the Fourier-plane key, $b(x, y)$.

Fig. 10 shows an example of attempted decryptions, using Method (i), when 10% of the decryption key pixels are in error. The images correspond to the results indicated by the dashed vertical line in Fig. 9a. The two different types of noise, commented upon in Fig. 8, are produced by errors in the two decryption keys. Fig. 10a, produced due to errors in the Fourier-plane key, contains predominantly Gaussian type noise. Fig. 10b produced due to errors in the image-plane key, contains predominantly shot or 'salt & pepper' type noise. Both types of noise, (as in Fig. 8), are present in Fig. 10c.

To further clarify our discussion of the effects of the phase encoding method used in conjunction with the DRPE technique and the types of noise observed, we now re-examine in a slightly different way, the effects on the decrypted output of errors of the decrypting Fourier key, $b(x, y)$, for the case when $\Delta = \pi/4$. This corresponds to the zero error case in Fig. 5a in which re-assignment is un-necessary. In Fig. 11a we show the resulting output pixel values plotted in the complex plane, and in Fig. 11b the distribution of phase values for the case when perfect, error free decryption is performed. The phase values are quantised and none appear in the forbidden region. In Fig. 12a

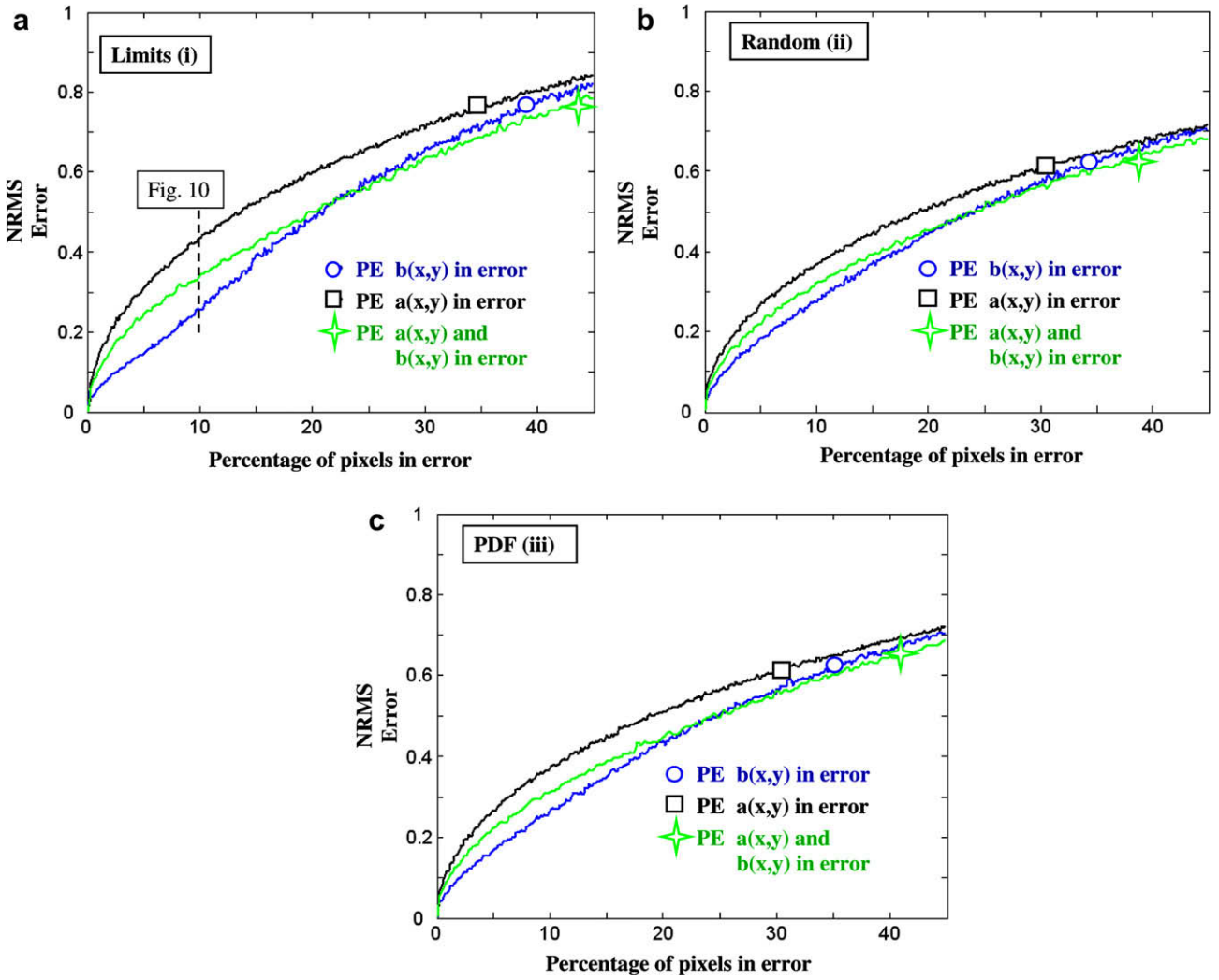

Fig. 9. These graphs relate to Fig. 6 where: (a) is when value in the forbidden phase region are assigned to the limits, (b) when they are assigned with uniform probability to random quantisation values, and (c) when they are assigned to a quantisation value based on an exponential probability distribution function. The circle denotes the case when the Fourier-plane decrypting phase-key is in error, the square denotes when the image-plane decrypting phase-key is in error and the star denotes when both the decrypting phase-keys are in error.

Fig. 10. These noisy decrypted images are for the three cases corresponding to the vertical dashed line in Fig. 9a. In each case 10% of the pixels are in error but in (a) Method (i), in (b) Method (ii), and in (c) Method (iii) are used to reassign pixels with forbidden phase angles.

and b we show the corresponding figures when 2.28% of the pixels in $b(x, y)$ are in error during decryption. This corresponds to the cases examined in Figs. 5a and 7b prior to re-assignment of the forbidden phase values. In Fig. 12a a variation in pixel amplitudes and a filling up of the circle can be observed. In Fig. 12b a smoothing of the distribution of phase values and the appearance of values in the forbidden region can be seen. Finally in Fig. 13a and b 10% of the pixels in $b(x, y)$ are in error. The trends observed continue, with further scrambling of the image information.

The progressive degradation of the image data, in going from Figs. 11–13, shows that, when the Fourier-plane phase-key is in error, the distribution of the pixel values tends to a Gaussian type distribution. This, as we have noted, produces predominantly Gaussian type noise in the decrypted Lena image, see Fig. 10a.

When the corresponding series of figures are generated assuming that only the image-plane key, $a(x, y)$, is in error two significant difference are observed. First, there is no change in the magnitudes of the pixel values observed in the complex plane. The introduction of the errors in the key does however result in a random filling in of the forbidden arc, i.e. the forbidden phase region in the circumference of the unit circle. Second, while increasing error smoothes the distribution of pixel phase values, significant variations between the number of pixel having adjoining phase values (similar to those in the perfectly decrypted case), can be observed. The presence of these sudden jumps is linked to the existence of the salt and pepper type noise, in the incorrectly decrypted Lena images discussed earlier, see Fig. 10b.

Errors in both keys then lead to a combination of both types of noise as observed in Figs. 8 and 10c.

Please cite this article in press as: D.S. Monaghan et al., Opt. Commun. (2008), doi:10.1016/j.optcom.2008.10.047

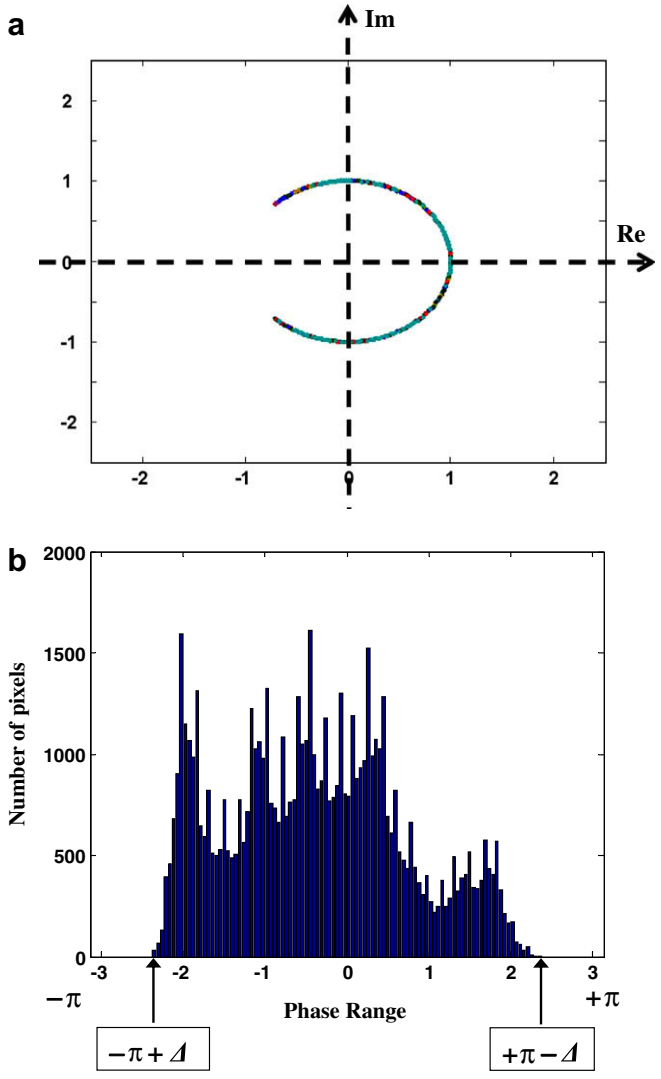

Fig. 11. (a) This is the output phase for perfect decryption when there are no errors present in the decrypting phase-keys. (b) Shows a histogram of the distribution of the pixels across the phase range. $\Delta = \pi/4$.

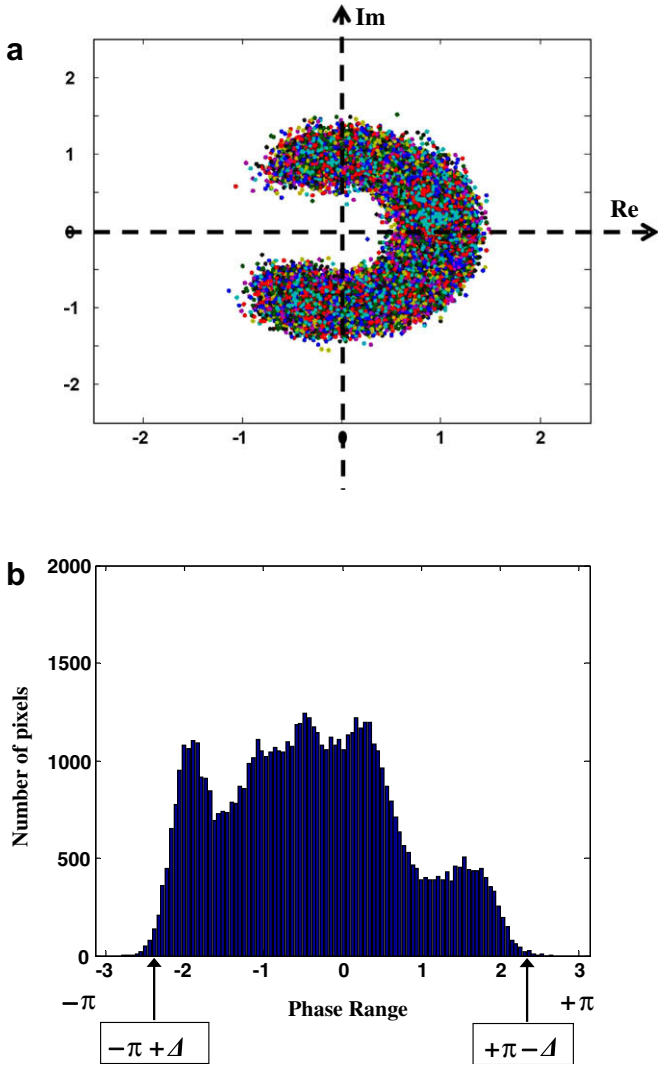

Fig. 12. (a) This is the output phase of a decrypted image 2.28% of the pixels in the Fourier-plane decrypting phase-key are in error. (b) Shows a histogram of the distribution of the pixels across the phase range. $\Delta = \pi/4$.

6. Conclusions

In this paper the DRPE technique, applied using phase encoded input images, is examined in detail. In particular we wish to more fully understand the effects of using a particular phase range during phase encoding and the form of the noise produced during incorrect decryption.

To perform this study we have shown that in this case increasing the number of phase-key phase levels above 16 does not significantly increase the security of the DRPE technique, since there is no appreciable increase in the NRMS error metric for increasing numbers of phase levels, see Fig. 2. Therefore an attacker, who does not know a priori the number of phase levels involved in encryption, can choose to assume 16 levels (4 bits per pixel) and will be able to attack the encrypted data confident that equivalent accuracy can be achieved as with a more numerically intense 64 level (6 bits per pixel) search.

We proceed by increasing the amounts of error in the decrypting phase-keys and quantifying the resulting increasing NRMS error in the output decrypted images. Using this procedure we examine the effects on the NRMS error of reducing the phase range from +/− π to +/− (π − Δ) during the phase encoding of the original input image. If identically encrypted phase encoded images are compared, one using $\Delta = 0$ and one using $\Delta = \pi/4$, it is observed that there is no discernable difference between the two random encrypted outputs, see Fig. 4. Since an attacker cannot simply identify the Δ value used, or the corresponding quantised phase levels, the introduction of Δ provides extra security. This point is emphasised as it is shown that even if both decryption keys are exactly known, if Δ is assumed to be zero by an attacker then the larger the actual value of Δ used, the greater NRMS error even if both keys are known. See Fig. 7.

Examining the cases presented in Fig. 5, it is clear that introducing a Δ value makes the system more robust to noise for a valid user. However, since changing the key values produce smaller changes in the resulting NRMS error it also appears to make the encryption system harder to crack by an invalid user.

We then proceed to examine three different methods of assigning acceptable phase values to pixels which, following incorrect decryption, have fallen outside the allowed phase range used to phase encode the original input image. We labelled these three re-assignment techniques Methods (i), (ii) and (iii), as illustrated in Fig. 6. It is shown that re-assignment to the limiting forbidden range values, as used in Ref. [15], produces higher NRMS values

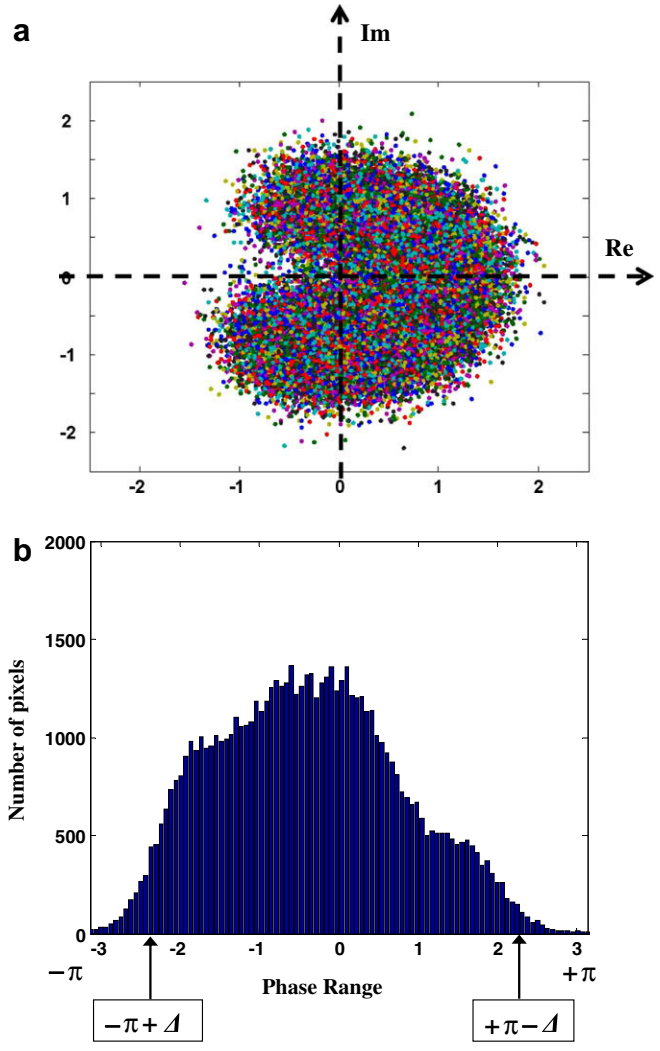

Fig. 13. (a) This is the output phase of a decrypted image 10% of the pixels in the Fourier-plane decrypting phase-key are in error. (b) Shows a histogram of the distribution of the pixels across the phase range. $\Delta = \pi/4$.

and therefore should not be used in any attempt to break into the system by an invalid user. See Fig. 9.

In Fig. 10 it is shown that the resultant noise appearing in the decrypted images arising due to errors in the Fourier-plane phase-key, $b(x, y)$, lead primarily to Gaussian noise, while errors in the image-plane phase-key, $a(x, y)$, lead to 'salt and pepper' noise. This is discussed in more detail using Figs. 11–13. It is indicated that shot noise introduced by errors in the image-plane key can be more easily eliminated using post-processing. It therefore seems reasonable to suggest that during an attack, elimination of shot noise in the decrypted image prior to calculation of the NRMS error might allow efforts to be focused on the Fourier-plane key.

From our results it is reasonable to make three recommendations. We suggest that it is advisable: (a) To use a Δ value when encrypting in order to improve on the overall security of the encrypted image; (b) If Δ is known it is advantageous to an attacker to use either Method (ii) or (iii) to reassign forbidden phase values; and (c) An examination of the form of noise exhibited during a known plain cipher text attack might be used to increase an attackers chances of success.

References

[1] B.M. Hennelly, J.T. Sheridan, Opt. Commun. 247 (2005) 291.
[2] T. Nomura, B. Javidi, Appl. Opt. 39 (2000) 4783.
[3] G. Unnikrishnan, J. Joseph, K. Singh, Opt. Lett. 25 (2000) 887.
[4] P.C. Mogensen, J. Gluckstad, Opt. Lett. 25 (2000) 566.
[5] Y. Frauel, A. Castro, T.J. Naughton, B. Javidi, Proc. SPIE 5986 (2005) 25.
[6] N.K. Nishchal, J. Joseph, K. Singh, Opt. Eng. 42 (2003) 1583.
[7] E. Tajahuerce, O. Matoba, S.C. Verrall, B. Javidi, Appl. Opt. 39 (2000) 2313.
[8] B.M. Hennelly, J.T. Sheridan, Optik 114 (2003) 251.
[9] E. Tajahuerce, B. Javidi, Appl. Opt. 39 (2000) 6595.
[10] T.J. Naughton, B. Javidi, Opt. Eng. 43 (2004) 2233.
[11] U. Gopinathan, G. Situ, T.J. Naughton, J.T. Sheridan, JOSA. A. 25 (2008) 108.
[12] U. Gopinathan, D.S. Monaghan, B.M. Hennelly, C.P. Mc Elhinney, D.P. Kelly, J.B. McDonald, T.J. Naughton, J.T. Sheridan, J. Display Technol. 4 (2008) 254.
[13] D.S. Monaghan, U. Gopinathan, T.J. Naughton, J.T. Sheridan, Appl. Opt. 46 (2007) 6641.
[14] P. Refregier, B. Javidi, Opt. Lett. 20 (1995) 767.
[15] B. Javidi, N. Towghi, N. Maghzi, S.C. Verrall, Appl. Opt. 39 (2000) 4117.
[16] U. Gopinathan, D.S. Monaghan, T.J. Naughton, J.T. Sheridan, Opt. Express 14 (2006) 3181.
[17] Y. Frauel, A. Castro, T.J. Naughton, B. Javidi, Opt. Express 15 (2007) 10253.
[18] Lena Test Image, <http://sipi.usc.edu/database/>.
[19] Matlab 7.0.1, <http://www.mathworks.com/>.
[20] G. Unnikrishnan, M. Pohit, K. Singh, Opt. Commun. 185 (2000) 25.
[21] U. Gopinathan, T.J. Naughton, J.T. Sheridan, Appl. Opt. 45 (2006) 5693.
[22] D.S. Monaghan, G. Situ, U. Gopinathan, T.J. Naughton, J.T. Sheridan, Appl. Opt. 47 (2008) 3808.
[23] W.K. Pratt, Digital Image Processing, fourth ed., Wiley-Interscience, 2007.