Key-space analysis of double random phase encryption technique

David S. Monaghan,[1] Unnikrishnan Gopinathan,[1] Thomas J. Naughton,[2] and John T. Sheridan[1,*]

[1]College of Engineering, Mathematics, and Physical Sciences, School of Electrical, Electronic, and Mechanical Engineering, University College Dublin, Belfield, Dublin 4, Ireland

[2]Department of Computer Science, National University of Ireland, Maynooth, Ireland, and University of Oulu, RFMedia Laboratory, Oulu Southern Institute, Vierimaantie 5, 84100 Ylivieska, Finland

*Corresponding author: john.sheridan@ucd.ie

Received 20 November 2006; revised 2 May 2007; accepted 22 May 2007; posted 25 May 2007 (Doc. ID 77142); published 7 September 2007

We perform a numerical analysis on the double random phase encryption/decryption technique. The key-space of an encryption technique is the set of possible keys that can be used to encode data using that technique. In the case of a strong encryption scheme, many keys must be tried in any brute-force attack on that technique. Traditionally, designers of optical image encryption systems demonstrate only how a small number of arbitrary keys cannot decrypt a chosen encrypted image in their system. However, this type of demonstration does not discuss the properties of the key-space nor refute the feasibility of an efficient brute-force attack. To clarify these issues we present a key-space analysis of the technique. For a range of problem instances we plot the distribution of decryption errors in the key-space indicating the lack of feasibility of a simple brute-force attack.

OCIS codes: 200.4740, 100.2000, 070.2580, 000.4430.

1. Introduction

The importance of cryptography [1–4] and information security has been recognized by governments and individuals throughout history. However, major technological advances in both computer technology and global communications have occurred in the past 50 years. In the digital information age, access to powerful computers brings with it both increased demands for, and threats to, security. This demand has led to ever faster and more powerful encryption systems being continually developed. Optical encryption [5–11] is one such solution to this problem. Optical encryption is particularly interesting, as it offers the possibility of high-speed parallel encryption of two-dimensional image data. Newly available low cost technology such as high quality spatial light modulators (SLMs), high resolution digital cameras (CCDs), and powerful desktop computers (PCs) have made optical encryption physically realizable. One such method of optical encryption is double random phase encoding (DRPE) [5].

DRPE is what we believe to be a unique method of optically encoding an image (see Fig. 1). The primary input image X is encoded to stationary white noise by the use of two statistically independent random phase-keys and two Fourier transforms. One key is placed in the input domain and the other key is placed in the Fourier domain. (See Fig. 2 for an optical implementation of the DRPE.) The method can be numerically simulated by means of matrices of discrete values and the fast Fourier transform (FFT) [12]. In this study we concern ourselves only with the intensity of the output image, and the output phase can be discarded. Therefore, in this study of the DRPE system, the random key located at the Fourier plane serves as the only decryption key to the system.

In a physical implementation of this optical encryption system it is necessary to capture the full field information, amplitude, and phase. Since CCDs can capture only the intensity of a wave field, digital holographic [6,13–15] techniques need to be employed to extract the full complex wave field information at

0003-6935/07/266641-07$15.00/0

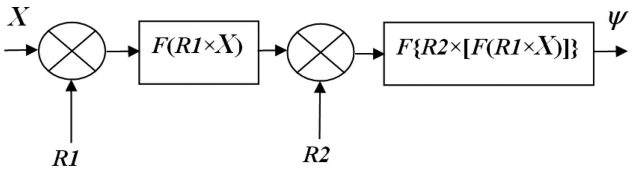

Fig. 1. Double random phase encoding (DRPE).

the camera. Digital holographic techniques can involve the use of a reference beam, with the resulting interference fringes being captured by the CCD and numerical phase retrieval techniques [16,17] being used to recover and unwrap the phase.

The novelty and advantage of the DRPE technique over digital encryption techniques such as the Diffie-Hellman [3] public key algorithm are that it has an optical implementation. The DRPE and similar optical systems have been primarily studied numerically [18] with some experimental studies [19].

Several other studies of the cryptographic strength of DRPE have been performed to date. Prior work has been done examining the effect of noise and errors in the input/output pair when the encryption/decryption keys are known perfectly [20]. The papers of Carnicer *et al.* [21], Frauel *et al.* [22], and Peng *et al.* [23] focused on retrieving exact solutions to the decryption key for various special-case chosen and known plaintext attacks. More relevant to the work presented in this paper, Gopinathan *et al.* [24] have analyzed DRPE in the context of attacks that seek to approximate the decryption key. However, the results of this study are specific to the particular heuristic used. Until now, there have been no analyses of the key-space independent of the type of attack methodology employed.

An encryption technique's key-space is a set of possible keys that can be used to encode data using that technique. For instance, a simple combination lock with three dials, each with 10 digits, has a key-space of 1000 keys, i.e., 10^3. The number of possible combinations therefore grows exponentially with the number of dials (equivalently, the number of pixels in our study). The size of the key-space determines the number of possible unique keys that can be used by the encryption technique. In an ideal encryption technique only one key would decrypt the encoded message and every other key would give a large error, i.e., the decryption would contain no useful information and be highly uncorrelated from the input image. However, this is not the case with the DRPE technique. As is generally known, there are typically several keys that will decrypt the encoded message with relatively low error, and as we demonstrate here there are in fact multiple keys that give a perfect decryption.

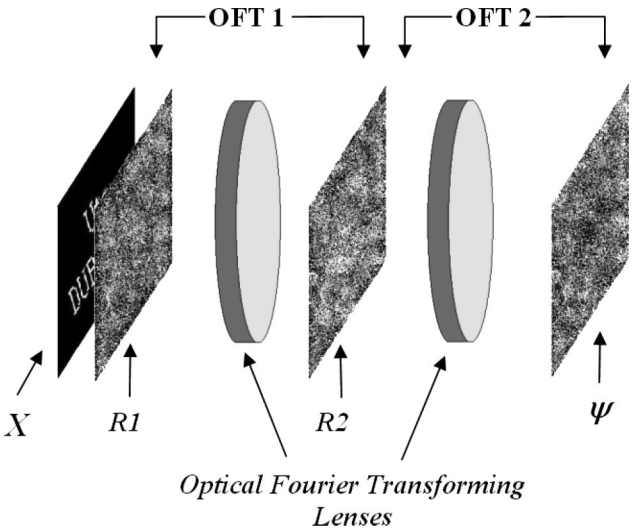

Fig. 2. Schematic of one possible optical implementation of DRPE.

In what follows we examine the technique's key-space using histograms showing the number of keys that decrypt an encoded message to given quantitative error levels. An analysis of the key-space for large image sizes (large number of pixels) is computationally intensive because of the large number of keys. We therefore carry out our analysis for small input image sizes and extrapolate from these results to predict the behavior for larger inputs under certain assumed conditions. By mapping the decrypting error across the entire key-space we can provide an analysis of the strength of the optical encryption technique.

While a review of the different existing optical encryption methods has been carried out in the past [18], no one, to our knowledge, has previously carried out this type of key-space analysis of the DRPE technique. This may be the case because it was assumed impractical given the very large size of the key-space involved. For instance, a random phase-key with four quantization levels between 0 and 2π and having 10×10 pixels has 1.6×10^{60} possible unique configurations. In this paper, by performing a complete analysis for small key-spaces, with pixel sizes of 3×3, 4×4, and 5×5, we attempt to identify trends in key-space, which, if consistent across all experiments, can reasonably be assumed to be consistent in larger random key-spaces. We then verify our conclusions with a statistical (rather than complete) analysis of a more practical sized image (256×256 pixel Lena [25]).

Our analysis is based on the key-space of the encryption system. In this system the key is regarded as phase-key **R2** (the phase-key in the Fourier plane—see Figs. 1 and 2), as we only require the output intensity of the image. Therefore the number of keys in the key-space is completely determined by the phase-key **R2** and depends on

(1) The key dimensions in pixels, and
(2) The number of phase quantization levels used in the phase-key.

A system with a phase-key that has $N \times M$ pixels, each with Q quantisation levels, has $Q^{(N \times M)}$ keys.

2. Error Analysis

In our analysis the encryption/decryption process is performed numerically. The FFT algorithm is used and each pixel is represented by a single complex value in the computer. Thus we neglect all physical modeling issues, e.g., SLM fill factor, polarization, and diffraction effects. Such simplifications are tolerated only because it is the nature of the DRPE technique, which is our primary consideration here and not the nonideality introduced by the physical limitations of the use of SLMs in physically implemented optical systems. However, it should be noted that although it is found that the immunity of DRPE is proportional to M, N, and Q, ultimately, Q-dependence will be limited by the signal-to-noise ratio, and the immunity dependence on M and N will be limited by the resolution. This implies that there is a physical upper limit to the size of the key-space. We assume that we have a known plain/cipher pair (a known input and the resulting encrypted image using an arbitrary phase key **R1** and the unknown key **R2**).

The metric we use to quantify the decrypting ability of each phase-key examined is the normalized root mean squared (NRMS) error in the resulting decrypted image. This is calculated using

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

where $I_d(\,)$ and $I(\,)$ are the intensities of the decrypted and original images, respectively; $0 \le \mathrm{NRMS}$ where $\mathrm{NRMS} = 0$ means perfect decryption. We define an acceptable decrypting phase-key as one that produces an NRMS error < 0.2. This is because in general the output at this level can be recognized by visual inspection. More specifically we have found, for the examples discussed in this paper (i.e., Fig. 6 in Section 3), that applying simple thresholding to the outputs of all the keys that result in $\mathrm{NRMS} < 0.2$ gave the correct binary input image.

We note that since we assume a lossless system, energy must be conserved between the input image and the output encrypted image. We use this as a necessary but insufficient test of the numerical accuracy and stability of our software.

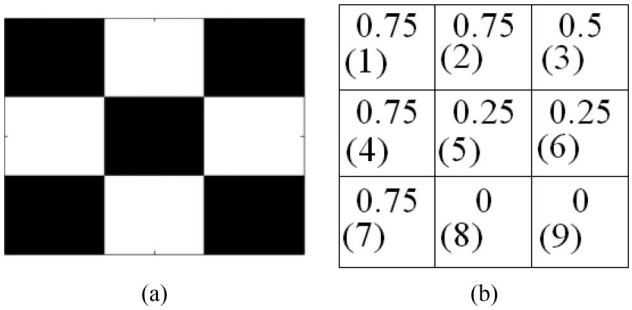

Fig. 3. (a) Original image is a binary real image and (b) **R2** phase values (multiply by 2π). The numbers in parentheses correspond to those in Fig. 11.

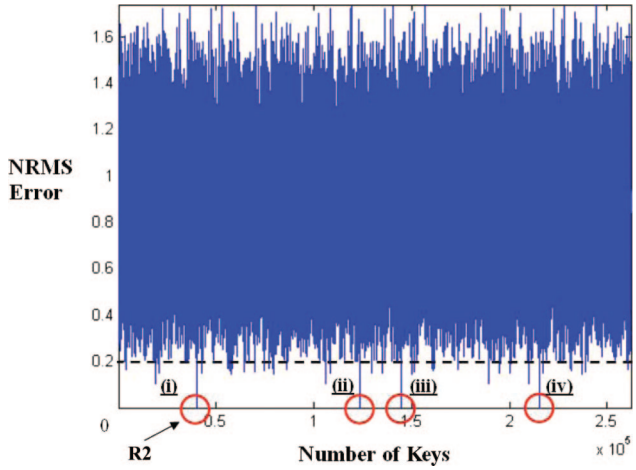

Fig. 4. (Color online) Error produced by each phase-key when used to decrypt the system. There are four exact, NRMS = 0, encircled solutions, and 56 cases with NRMS < 0.2. The four exact solutions are labeled as in Fig. 11.

3. Results

The analysis of the system presented here was carried out numerically on a PC (Pentium 4 CPU 3.2 GHz, 2 Gbytes RAM using Matlab 7.0.1). We present the results from a detailed series of tests carried out using a 3×3 image with four quantization levels. There are $4^{3\times3} = 262{,}144$ possible unique phase-keys in the key-space for this system. The four possible values for the phase-key levels are $2\pi \times [0, 0.25, 0.5, 0.75]$.

The input plaintext image is encrypted using a randomly chosen phase-key from the key-space. Attempts are then made to decrypt the output using every possible phase-key. The NRMS error associated with the use of every possible phase-key in the key-space is recorded. Figure 3 shows an input image and **R2**.

The resulting NRMS errors for the entire key-space of this system are given in Fig. 4. As we move along the x-axis of the graph we systematically try

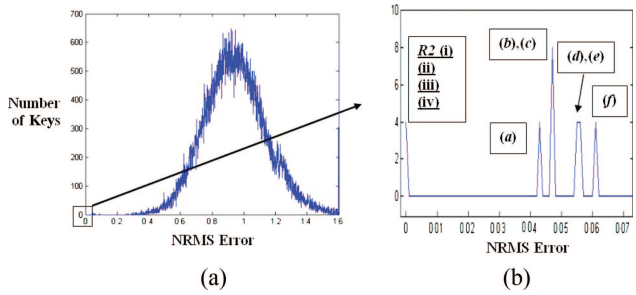

Fig. 5. (Color online) (a) Histogram of the NRMS error associated with every phase-key in key-space, which shows the number of phase-keys that decrypt to a certain error. (b) A zoomed-in plot of the section of (a) near the origin, showing there are four phase-keys that achieve exact decryption and 24 with $0.04 < \mathrm{NRMS} < 0.07$ (also see Fig. 11).

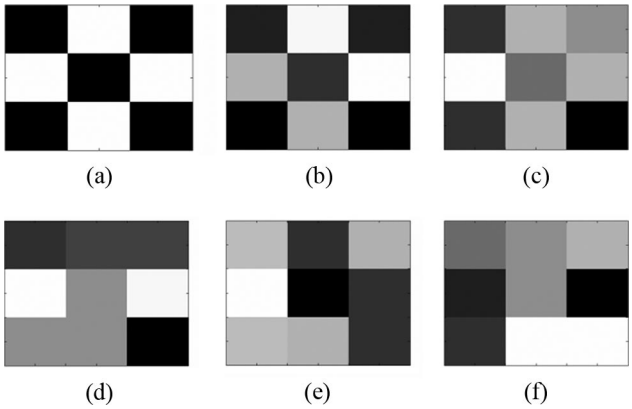

Fig. 6. Decrypted images with NMRS errors of: (a) 0, (b) 0.2, (c) 0.4, (d) 0.6, (e) 0.8, and (f) 1. The application of a threshold at 0.5, which sets each pixel to either 1 or 0, will mean output (a) and (b) give the correct solution and output (c) has only one incorrect pixel.

each phase-key and plot the corresponding error that each key produces. A single curve is then drawn connecting each of these points for visualization purposes. As is indicated by the four circles, there are exactly four phase-keys that perfectly decrypt the encrypted image, labeled (i)–(iv). This arises because there are four quantization levels and because we are interested only in the output intensity. Therefore only the relative phase between pixels matters and adding a constant phase to a decrypting phase-key will not affect the decrypted intensity. Thus in such a system, with $Q = 4$ phase levels, there are four keys that achieve perfect decryption. However, importantly, we also note that there are 56 phase-keys that decrypt the image with an NRMS error < 0.2 (below the dashed line in Fig. 4).

Figure 5 is a histogram of the spread of phase-keys in the key-space with respect to the NRMS error they produce. Figure 5(b) shows a zoomed-in area of 5(a) close to NRMS = 0. The four phase-keys that perfectly decrypt it are once again labeled **R2**, (i), (ii), (iii), and (iv). Figure 5(b) shows that 24 phase-keys decrypt the input with $0.04 <$ NRMS error < 0.07. These are labeled (a)–(f) and are discussed later in relation to Fig. 11.

Figure 6 shows an example of the output image when a particular phase-key is used for cases when the NRMS = (a) 0, (b) 0.2, (c) 0.4, (d) 0.6, (e) 0.8, and (f) 1. The input was binary and the resulting outputs are gray scale. However, based on the *a priori* knowledge that our input was binary, the output pixels could be set to either white (1) or black (0), and this postprocess yields more accurate results. For instance, in Fig. 6, following the application of a threshold of 0.5, both outputs (a) and (b) give the correct input image, while output (c) has one incorrect pixel value. This result indicates the significance for this case of the NRMS = 0.2 value.

So far the phase-key used contained 3×3 pixels, and this is too small to be considered statistically significant. We extended the experiments to larger key sizes and repeated all of the previously described tests for phase-keys with 4×4 pixels and 5×5 pixels with $Q = 2$. Increasing the size of the key-space, from $4^{(3\times3)} = 262{,}144$ to $2^{(5\times5)} = 33{,}554{,}432$ phase-keys, and using both nonuniform binary and gray scale inputs, we note that exactly the same trends we had reported are still observed.

Based on our simulations we note that the number of acceptable phase-keys, NRMS < 0.2, as a percentage of the total number of keys in the key-space falls very quickly as the size of the phase-key increases. This suggests that it is more secure to have phase-keys with a large number of quantization levels despite the resulting increase in the number of both exact solutions, Q, and solutions with NRMS < 0.2, Y. This trend is illustrated in Table 1. The results for Y, presented in Table 1, are average values found after 10 runs of each simulation.

Can our results for keys with 9, 16, and 25 pixels be extrapolated to keys with a larger number of pixels? We ran a simulation for a system with a 256×256 (65,536) pixel phase-key with $Q = 8$ quantization levels giving $8^{(256\times256)}$ keys. The input image for this simulation was a gray scale picture of Lena (256×256). We randomly generated 10^6 phase-keys and used them to decrypt an output. In Fig. 7 we plot the resulting histogram of the NRMS error values. None of the keys generated an NRMS error outside the 0.98–1.02 range. Thus, as in Fig. 5(a), most keys produce NRMS values centered at NRMS = 1. For the sake of thoroughness all eight exact phase-key solutions were tested and we confirmed that each key decrypted perfectly.

Next for this large key-space case we took the original decrypting key and added increasing amounts of error. To systematically examine key degradation we first introduced the error by randomly choosing a number of pixels and adding identical amounts of phase error to all the pixels chosen. Figure 8 shows the results of these simulations for various numbers of pixels and phases. Each point on the graph represents 100 simulations with the average of these results being plotted. Clearly, as the number of pixels in

Table 1. For a 3×3 Pixel System With $Q = 2$, 3, and 4 There Is a Comparison of the Number of Keys in the Key-Space to the Fraction of Keys That Produce an Output an Exact Solution and the Fraction of Keys That Produce an Output With NRMS < 0.02. The Increase in Exact Solutions and Solutions With NRMS < 0.2 Is Much Less Than the Increase in Key-Space

Size of Key-Space	$Q^{(N\times M)}$	$2^{(3\times3)} = 512$	$3^{(3\times3)} = 19{,}683$	$4^{(3\times3)} = 262{,}144$
Fraction That Are Exact Solutions	$Q/Q^{(N\times M)}$	0.0039	0.000152	0.0000153
Number With NRMS < 0.02	Y	4	11	58
Fraction With NRMS < 0.02	$Y/Q^{(N\times M)}$	0.00781	0.000559	0.000221

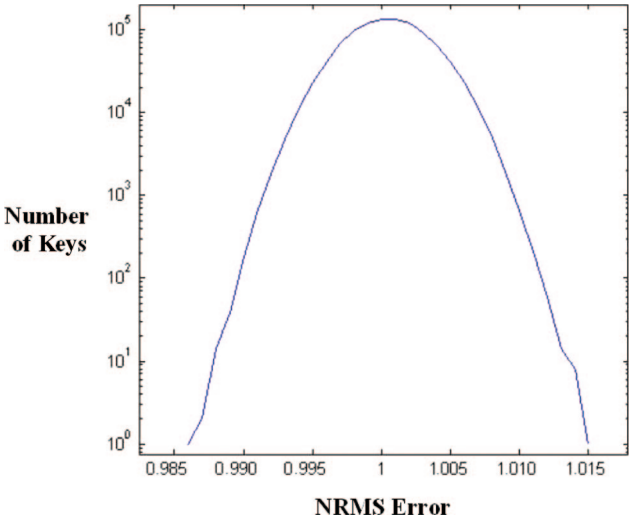

Fig. 7. (Color online) Distribution of 1×10^6 keys randomly generated in a 256×256 system with $Q = 8$. Note that the y-axis is a logarithmic scale. For comparison see Fig. 5(a).

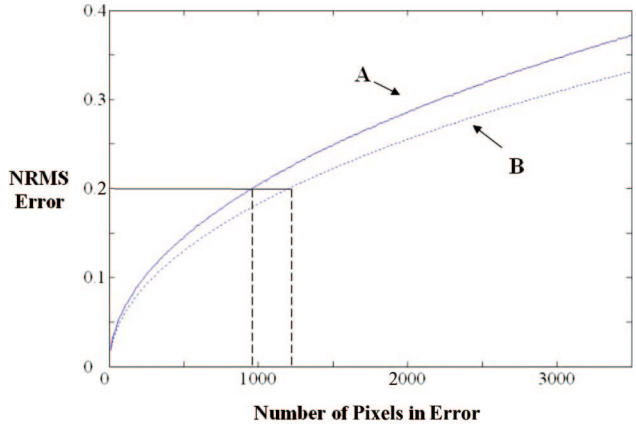

Fig. 9. (Color online) The exact phase-key **R2** is taken and seven equally likely phase values are chosen randomly and added to an increasing number of pixels to see how it affects the decryption (A, solid curve). The values from Fig. 8 are averaged and plotted (B, dotted curve). The random selection of phases produces a slightly higher NRMS error. NRMS < 0.2 when 1.5%–1.9% of the total number of pixels are in error.

error increases, the NRMS error increases. Furthermore, the largest error arises when the constant phase value added to all the pixels chosen is π. This is as expected since, as the phase-key is modulo 2π, a pixel will have the largest error when it is π radians away from its correct value. This also explains why the curve is symmetric about the phase error value of π. The phase-key used in this simulation had 65,536 pixels. We perturbed up to a maximum of 3,500 pixels, corresponding to 5.3% of the total number of pixels in the key.

We repeated this experiment but chose to add one of seven equally likely phase values, $\pi/4$, $\pi/2$, $3\pi/4$, π, $5\pi/4$, $3\pi/2$, and $7\pi/4$, to each of the randomly chosen pixels in error. Once again each simulation was repeated 100 times and the average result is plotted (A: solid curve) in Fig. 9. The constant phase values presented in Fig. 8 were averaged and also plotted (B: dotted curve). Comparing these results we note that in this case (i.e., A) we predict a slightly higher error than the average value of those presented in Fig. 8, (i.e., B). This would seem to indicate that random phase error positioned randomly among the pixels will in general, be more deleterious than constant errors randomly positioned. We repeated the "A" simulation for gray scale Lena images of 64×64, 128×128 and 256×256 pixels. We note that the shape of the curve is consistent for the different-sized keys. Furthermore, in all cases, it was observed that on average when 1%–2% of pixels were in error the key resulted in an NRMS error ~ 0.2.

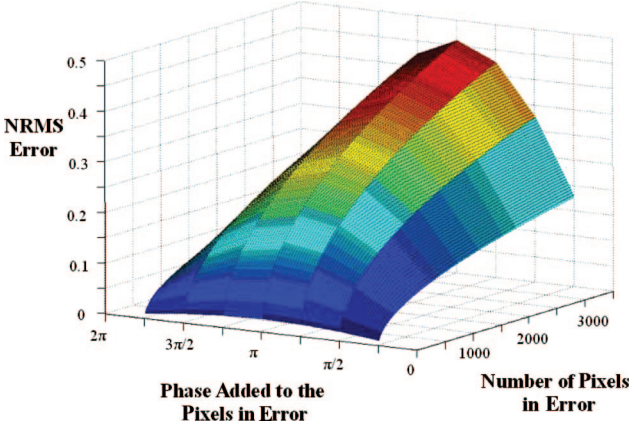

Fig. 8. (Color online) The exact phase-key is taken and a constant phase is added to an increasing number of pixels to see how it affects the decryption. There are 65,536 pixels in the phase-key and the maximum number of pixels changed is 3,500, which is 5.3% of the total number of pixels.

4. Aids in the Visualization of Key-Space

Our analysis of the DRPE technique from a key-space perspective allows users to evaluate the security of the system against a brute-force attack. However, by fully mapping the key-space in a systematic manner, it might then be possible to navigate the map, i.e., to find a solution without the need to check every key. One difficulty is to find a sensible method of representing a multidimensional key-space.

We now propose two graphical aids to help in the conceptualization of key-space. The first is a method of plotting key-space that emphasizes the pixilated nature of the key. If we take, for example, a phase-key with two pixels and $Q = 8$ quantization levels, the key-space of this system will contain 64 keys and eight exact solutions. Figure 10 shows a plot of the key-space for such a system with a randomly generated gray scale input image. The eight exact solutions, which have NRMS $= 0$ error, form a diagonal curve, and since the system is modulo 2π, this line is broken into two parallel segments. We can see that if we choose any key at random, fix the phase value of one pixel, and then vary the phase of the other pixel, we are guaranteed to hit one of the exact solutions. In this way we have reduced the dimension of the

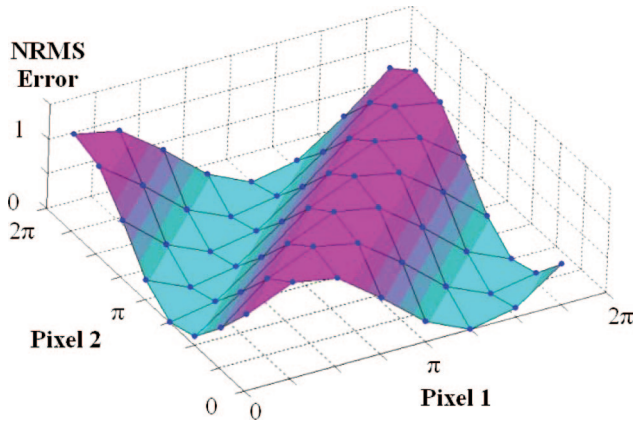

Fig. 10. (Color online) Visualization aid of the key-space for a system with two pixels and $Q = 8$.

search. While this graphical technique offers some insights when visualizing key-space for 1, 2, and 3 pixel systems, since each pixel requires an axis, keys with larger numbers of pixels defy simple representation.

The second graphical aid we propose involves mapping out individual keys as paths so that a visual comparison can be made between them. We illustrate this technique in Fig. 11, in which we reexamine the 3×3 pixel case with $Q = 4$ previously discussed in Section 3. Labeling the pixels of **R2** from 1–9, as shown in Fig. 3(b), we plot the quantized **R2** phase values as a function of pixel position. In this way keys can be drawn as paths on the grid. **R2** appears twice in Fig. 11 as two parallel piecewise linear curves (thick solid lines) separated by 2π radians. The other exact solution keys with NRMS = 0, labeled (ii)–(iv) (see Figs. 4 and 5), appear with identical path shapes

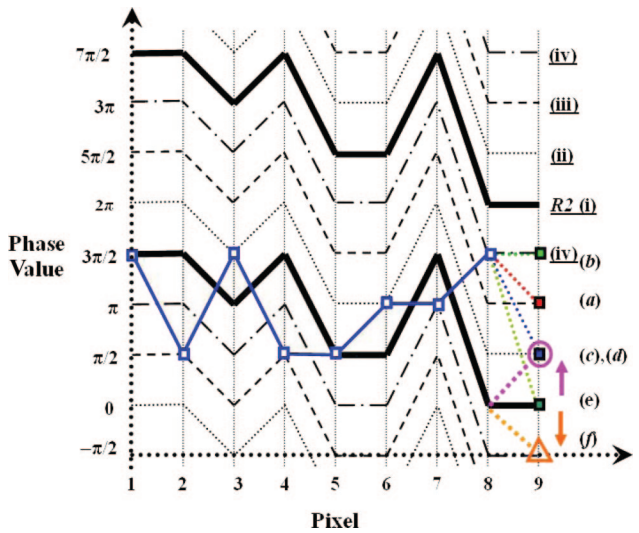

Fig. 11. (Color online) Visualization aid for mapping out the phase-keys for the cases examined in Figs. 4 and 5. The thick solid line is **R2** (i). The dashed lines (ii), (iii), and (iv) are the other correct keys. The labels (a)–(f) correspond to keys giving low NRMS error decryption.

to **R2**, separated by shifts of constant quantized phase value.

Before proceeding we first return to Fig. 5(b), where we see **R2** (i)–(iv) at NRMS = 0. We also see labeled, as (a)–(f), the $4 \times 6 = 24$ keys with NRMS error between 0.04 and 0.07. One of each of the four keys associated with (a)–(f) is now plotted on Fig. 11.

These keys naturally divide into two types: The first type involves cases (d) circle, NRMS = 0.0555, and (f) triangle, NRMS = 0.0606. Both these cases involve **R2**, with the phase value of the last pixel ("9") being incorrect by plus (arrow up, circle) or minus (arrow down, triangle) one quantization phase level. Thus, they clearly represent weak perturbations to the **R2** key, which, using the NRMS error function (cost function), produces a small perturbation of the error value.

The second type involves cases (a) NRMS = 0.0430, (b) NRMS = 0.0474, (c) NRMS = 0.0475, and (e) NRMS = 0.0559. They also correspond to a single path, differing from one another only in the phase value of the last pixel, 9. The common part of these keys is represented in Fig. 11 by squares joined by a thin solid line. As in the case of **R2**, four equivalent phase shifted versions of each path give the same NRMS error; these other versions of each path are not given in Fig. 11. Clearly, these keys provide almost exact decryption while simultaneously differing completely from the exact **R2** key. This indicates that the NRMS error function predicts the existence of keys (local minima) with little difference from **R2** (the global minimum, i.e., NRMS = 0). This also provides some explanation why low NMRS-based estimates of **R2**, found during plaintext attacks [23], do not on occasion then provide good decryption of other images encrypted using **R2**.

Therefore, based on our use of the NRMS error function, this implies that the DRPE technique is secure from brute-force attack, since good keys, as defined by the NRMS, are simply not identifiable. To further support this conclusion we examined the value of the NRMS error when other **R2** pixel phase values were changed by one phase level. In general, large errors, NRMS ~ 0.4, were observed. Thus, increasing or decreasing the difference between the keys (paths) in the key-space does not necessarily correspond to a simply related change in the NRMS error of the decrypted image.

5. Conclusion

In a desire to study the robustness of the DRPE technique to brute-force attack, we have examined the key-space assuming that insights gained by fully mapping small key-spaces can be extrapolated to large key-spaces. Comparing the full, yet statistically insignificant, small key-space results to the incomplete, but statistically significant, large key-space result provides evidence in support of this hypothesis.

We have observed that for image data a DRPE system with Q quantization levels has Q phase-keys that perfectly decrypt the system. This has been explained as a result of being interested only in the

output intensity. Since the size of the key-space depends on the number of quantization levels, i.e., $Q^{(N\times M)}$, any increase in the number of quantization levels will produce a much larger key-space whose size increases much more rapidly than the resulting increase in the number of exact solutions.

Defining an NRMS error metric, we have shown that as well as there being Q exact solutions there are always several phase-keys that will decrypt the system with low NRMS error. For the low dimensional cases examined, we have shown that for NRMS < 0.2, the decrypted outputs frequently yield the correct solution after a simple thresholding operation is performed. However, we also have demonstrated that the number of keys for which NRMS < 0.2 also decreases rapidly as a fraction of the total number keys in the key-space.

It is important to note that these results are not definitive, as a 5×5 pixel sized key, the largest key-space we fully mapped, is too small to be considered truly statistically significant. Therefore, we also have presented results for a gray scale Lena image with 256×256 pixels, and taking $Q = 8$. For such a large key-space any brute-force method of mapping the entire key-space currently appears to be unrealistic. The strength of the DRPE technique is, however, indicated by our observation (for both small and large key-spaces) that the majority of the phase-keys produce results centered on the NRMS error value of 1 and, furthermore, that the introduction of even a small number of random variations in the **R2** key will in general lead to large NRMS errors. However, while brute-force attack appears impractical, it should be noted that nonbrute-force attack techniques, based on heuristic approaches [24], exist and have been applied successfully.

To aid in the mapping of the key-space we have introduced and discussed two simple graphical representations. Examining small key-spaces, we have applied these to illustrate (i) a systematic reduction in the dimensionality of the key-space, and (ii) the relationship between deviations from the correct keys and the NRMS error function. These graphs clarify our observations regarding the robustness of the DRPE technique to brute-force attack and show the difficulty in systematically mapping (and thus searching) the key-space.

We acknowledge the support of Enterprise Ireland and Science Foundation Ireland through the Research Innovation and Proof of Concept Funds, the Basic Research and Research Frontiers Programs. We also acknowledge the support of the Irish Research Council for Science, Engineering, and Technology.

References

1. G. F. Gaines, *Cryptanalysis: a Study of Ciphers and Their Solution* (Dover, 1939).
2. H. O. Yardley, *The American Black Chamber* (U.S. Naval Institute Press, 1931).
3. W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theory **22,** 644–654 (1976).
4. C. A. Deavours, *Cryptology Yesterday, Today and Tomorrow* (Artech House, 1987).
5. P. Refregier and B. Javidi, "Optical-image encryption based on input plane and Fourier plane random encoding," Opt. Lett. **20,** 767–769 (1995).
6. E. Tajahuerce and B. Javidi, "Encrypting three-dimensional information with digital holography," Appl. Opt. **39,** 6595–6601 (2000).
7. B. Hennelly and J. T. Sheridan, "Optical image encryption by random shifting in fractional Fourier domains," Opt. Lett. **28,** 269–271 (2003).
8. B. M. Hennelly and J. T. Sheridan, "Optical encryption and the space bandwidth product," Opt. Commun. **247,** 291–305 (2005).
9. L. E. M. Brackenbury and K. M. Bell, "Optical encryption of digital data," Appl. Opt. **39,** 5374–5379 (2000).
10. G. Unnikrishnan, J. Joseph, and K. Singh, "Optical encryption by double-random phase encoding in the fractional Fourier domain," Opt. Lett. **25,** 887–889 (2000).
11. T. J. Naughton and B. Javidi, "Compression of encrypted three-dimensional objects using digital holography," Opt. Eng. **43,** 2233–2238 (2004).
12. B. M. Hennelly and J. T. Sheridan, "Generalizing, optimizing, and inventing numerical algorithms for the fractional Fourier, Fresnel, and linear canonical transforms," J. Opt. Soc. Am. A **22,** 917–927 (2005).
13. J. W. Goodman and R. W. Lawrence, "Digital image formation from electronically detected holograms," Appl. Phys. Lett. **11,** 77–79 (1967).
14. U. Schnars and W. Juptner, "Direct recording of holograms by a CCD target and numerical reconstruction," Appl. Opt. **33,** 179–181 (1994).
15. B. H. Zhu, H. F. Zhao, and S. T. Liu, "Image encryption based on pure intensity random coding and digital holography technique," Optik **114,** 95–99 (2003).
16. M. Liebling, T. Blu, and M. Unser, "Complex-wave retrieval from a single off-axis hologram," J. Opt. Soc. Am. A **21,** 367–377 (2004).
17. J. R. Fienup, "Phase retrieval algorithms—a comparison," Appl. Opt. **21,** 2758–2769 (1982).
18. B. M. Hennelly and J. T. Sheridan, "Image encryption and the fractional Fourier transform," Optik **114,** 251–265 (2003).
19. U. Gopinathan, T. J. Naughton, and J. T. Sheridan, "Polarization encoding and multiplexing of two-dimensional signals: application to image encryption," Appl. Opt. **45,** 5693–5700 (2006).
20. F. Goudail, F. Bollaro, B. Javidi, and P. Refregier, "Influence of a perturbation in a double phase-encoding system," J. Opt. Soc. Am. A **15,** 2629–2638 (1998).
21. A. Carnicer, M. Montes-Usategui, S. Arcos, and I. Juvells, "Vulnerability to chosen-cyphertext attacks of optical encryption schemes based on double random phase keys," Opt. Lett. **30,** 1644–1646 (2005).
22. Y. Frauel, A. Castro, T. J. Naughton, and B. Javidi, "Resistance of the double random phase encryption against various attacks," Opt. Express **15,** 10253–10265 (2007).
23. X. Peng, P. Zhang, H. Z. Wei, and B. Yu, "Known-plaintext attack on optical encryption based on double random phase keys," Opt. Lett. **31,** 1044–1046 (2006).
24. U. Gopinathan, D. S. Monaghan, T. J. Naughton, and J. T. Sheridan, "A known-plaintext heuristic attack on the Fourier plane encryption algorithm," Opt. Express **14,** 3181–3186 (2006).
25. http://sipi.usc.edu.database/.